



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

### Details

| Product Status             | Active  |
|----------------------------|---|
| Core Processor             | PIC   |
| Core Size                  | 8-Bit   |
| Speed                      | 20MHz   |
| Connectivity               | I <sup>2</sup> C, SPI   |
| Peripherals                | Brown-out Detect/Reset, LCD, POR, PWM, WDT                                |
| Number of I/O              | 25  |
| Program Memory Size        | 7KB (4K x 14)   |
| Program Memory Type        | ОТР   |
| EEPROM Size                | -   |
| RAM Size                   | 176 x 8   |
| Voltage - Supply (Vcc/Vdd) | 2.5V ~ 5.5V   |
| Data Converters            | A/D 5x10b   |
| Oscillator Type            | External  |
| Operating Temperature      | -40°C ~ 85°C (TA)   |
| Mounting Type              | Surface Mount   |
| Package / Case             | 64-TQFP   |
| Supplier Device Package    | 64-TQFP (10x10)   |
| Purchase URL               | https://www.e-xfl.com/product-detail/microchip-technology/pic16lc925-i-pt |

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

### FIGURE 2-4: REGISTER FILE MAP— DSTEMP

| Α  | ddress | A  | Address    |  | Address | Ac   | dre  |
|--|--------|--|------------|--|---------|--|------|
| ndirect addr. <b>(*)</b>                   | 00h    | Indirect addr.(*)                          | 80h        | Indirect addr.(*)                          | 100h    | Indirect addr.(*)                          | 180  |
| TMR0                                       | 01h    | OPTION                                     | 81h        | TMR0                                       | 101h    | OPTION                                     | 1811 |
| PCL  | 02h    | PCL  | 82h        | PCL  | 102h    | PCL  | 1821 |
| STATUS                                     | 03h    | STATUS                                     | 83h        | STATUS                                     | 103h    | STATUS                                     | 1831 |
| FSR  | 04h    | FSR  | 84h        | FSR  | 104h    | FSR  | 1841 |
| PORTA                                      | 05h    | TRISA                                      | 85h        |  | 105h    |  | 1851 |
| PORTB                                      | 06h    | TRISB                                      | 86h        | PORTB                                      | 106h    | TRISB                                      | 1861 |
| PORTC                                      | 07h    | TRISC                                      | 87h        | PORTF                                      | 107h    | TRISF                                      | 1871 |
| PORTD                                      | 08h    | TRISD                                      | 88h        | PORTG                                      | 108h    | TRISG                                      | 188  |
| PORTE                                      | 09h    | TRISE                                      | 89h        |  | 109h    |  | 1891 |
| PCLATH                                     | 0Ah    | PCLATH                                     | 8Ah        | PCLATH                                     | 10Ah    | PCLATH                                     | 18A  |
| INTCON                                     | 0Bh    | INTCON                                     | 8Bh        | INTCON                                     | 10Bh    | INTCON                                     | 18B  |
| PIR1                                       | 0Ch    | PIE1                                       | 8Ch        | PMCON1                                     | 10Ch    | PMDATA                                     | 18C  |
|  | 0Dh    |  | 8Dh        | LCDSE                                      | 10Dh    | PMADR                                      | 18D  |
| TMR1L                                      | 0Eh    | PCON                                       | 8Eh        | LCDPS                                      | 10Eh    | PMDATH                                     | 18E  |
| TMR1H                                      | 0Fh    |  | 8Fh        | LCDCON                                     | 10Fh    | PMADRH                                     | 18F  |
| T1CON                                      | 10h    |  | 90h        | LCDD00                                     | 110h    |  | 190  |
| TMR2                                       | 11h    |  | 91h        | LCDD01                                     | 111h    |  | 1911 |
| T2CON                                      | 12h    | PR2  | 92h        | LCDD02                                     | 112h    |  | 1921 |
| SSPBUF                                     | 13h    | SSPADD                                     | 93h        | LCDD03                                     | 113h    |  | 193  |
| SSPCON                                     | 14h    | SSPSTAT                                    | 94h        | LCDD04                                     | 114h    |  | 1941 |
| CCPR1L                                     | 15h    |  | 95h        | LCDD05                                     | 115h    |  | 1951 |
| CCPR1H                                     | 16h    |  | 96h        | LCDD06                                     | 116h    |  | 196  |
| CCP1CON                                    | 17h    |  | 97h        | LCDD07                                     | 117h    |  | 197  |
|  | 18h    |  | 98h        | LCDD08                                     | 118h    | · · · · · · · · · · · · · · · · · · ·      | 198  |
|  | 19h    |  | 99h        | LCDD09                                     | 119h    |  | 1991 |
|  | 1Ah    |  | 9Ah        | LCDD10                                     | 11Ah    | · · · · · · · · · · · · · · · · · · ·      | 19A  |
|  | 1Bh    |  | 9Bh        | LCDD11                                     | 11Bh    | · · · · · · · · · · · · · · · · · · ·      | 19B  |
|  | 1Ch    |  | 9Ch        | LCDD12                                     | 11Ch    | · · · · · · · · · · · · · · · · · · ·      | 19C  |
|  | 1Dh    |  | 9Dh        | LCDD13                                     | 11Dn    | · · · · · · · · · · · · · · · · · · ·      | 19D  |
| ADRESH                                     | 1Eh    | ADRESL                                     | 9Eh        | LCDD14                                     |         |  | 19E  |
| ADCON0                                     | 1Fh    | ADCON1                                     | 9Fh        | LCDD15                                     | 11FN    |  | 19FI |
|  | 20h    |  | A0h        |  | 120h    |  | 1A0  |
| General<br>Purpose<br>Register<br>96 Bytes |        | General<br>Purpose<br>Register<br>80 Bytes | BFh<br>C0h | General<br>Purpose<br>Register<br>80 Bytes |         | General<br>Purpose<br>Register<br>80 Bytes |      |
|  |        |  | EFh        |  | 16Fh    |  | 1EF  |
|  |        | accesses<br>70h - 7Fh                      | F0h        | accesses<br>70h - 7Fh                      | 170h    | accesses<br>70h - 7Fh                      | 1F0  |
| Darah 0                                    | 7Fh    | Bank 1                                     | FFh        | Bank 2                                     | 17Fh    | Bank 3                                     | 1FF  |

| Address | Name    | Bit 7         | Bit 6          | Bit 5                   | Bit 4          | Bit 3        | Bit 2            | Bit 1          | Bit 0    | Value on<br>Power-on<br>Reset | Details on page |
|---------|---------|---------------|----------------|-------------------------|----------------|--------------|------------------|----------------|----------|-------------------------------|-----------------|
| Bank 1  |         |               |                |                         |                |              |                  |                |          |                               |                 |
| 80h     | INDF    | Addressing    | this location  | uses conten             | ts of FSR to a | address data | memory (not      | t a physical r | egister) | 0000 0000                     | 26              |
| 81h     | OPTION  | RBPU          | INTEDG         | TOCS                    | T0SE           | PSA          | PS2              | PS1            | PS0      | 1111 1111                     | 20              |
| 82h     | PCL     | Program Co    | ounter (PC) L  | east Signific           | ant Byte       |              |                  |                |          | 0000 0000                     | 25              |
| 83h     | STATUS  | IRP           | RP1            | RP0                     | то             | PD           | Z                | DC             | С        | 0001 1xxx                     | 19              |
| 84h     | FSR     | Indirect Dat  | a Memory Ad    | ddress Pointe           | er             |              |                  |                |          | XXXX XXXX                     | 26              |
| 85h     | TRISA   | —             | —              | PORTA Dat               | a Direction R  | egister      |                  |                |          | 11 1111                       | 29              |
| 86h     | TRISB   | PORTB Dat     | a Direction F  | Register                |                |              |                  |                |          | 1111 1111                     | 31              |
| 87h     | TRISC   | _             | —              | PORTC Da                | ta Direction F | Register     |                  |                |          | 11 1111                       | 33              |
| 88h     | TRISD   | PORTD Dat     | a Direction F  | Register                |                |              |                  |                |          | 1111 1111                     | 34              |
| 89h     | TRISE   | PORTE Dat     | a Direction F  | Register                |                |              |                  |                |          | 1111 1111                     | 36              |
| 8Ah     | PCLATH  | _             | —              | —                       | Write Buffer   | for the uppe | er 5 bits of the | e PC           |          | 0 0000                        | 25              |
| 8Bh     | INTCON  | GIE           | PEIE           | TMR0IE                  | INTE           | RBIE         | TMR0IF           | INTF           | RBIF     | 0000 000x                     | 21              |
| 8Ch     | PIE1    | LCDIE         | ADIE           | _                       | —              | SSPIE        | CCP1IE           | TMR2IE         | TMR1IE   | 00 0000                       | 24              |
| 8Dh     | —       | Unimpleme     | nted           |                         |                |              |                  |                |          | —                             | —               |
| 8Eh     | PCON    | —             | —              | —                       | —              | —            | —                | POR            | BOR      | 0-                            | 24              |
| 8Fh     | —       | Unimpleme     | nted           |                         |                |              |                  |                |          | —                             | —               |
| 90h     | —       | Unimpleme     | nted           |                         |                |              |                  |                |          | —                             | —               |
| 91h     | —       | Unimpleme     | nted           |                         |                |              |                  |                |          | —                             | —               |
| 92h     | PR2     | Timer2 Peri   | od Register    |                         |                |              |                  |                |          | 1111 1111                     | 51              |
| 93h     | SSPADD  | Synchronou    | is Serial Port | (I <sup>2</sup> C mode) | Address Reg    | ister        |                  | -              | -        | 0000 0000                     | 69, 72          |
| 94h     | SSPSTAT | SMP           | CKE            | D/A                     | Р              | S            | R/W              | UA             | BF       | 0000 0000                     | 59              |
| 95h     | _       | Unimpleme     | nted           |                         |                |              |                  |                |          | _                             | _               |
| 96h     | —       | Unimpleme     | nted           |                         |                |              |                  |                |          | —                             | —               |
| 97h     | _       | Unimpleme     | nted           |                         |                |              |                  |                |          | _                             | _               |
| 98h     | _       | Unimpleme     | nted           |                         |                |              |                  |                |          | _                             | _               |
| 99h     | _       | Unimpleme     | nted           |                         |                |              |                  |                |          | —                             | -               |
| 9Ah     | _       | Unimpleme     | nted           |                         |                |              |                  |                |          | _                             | _               |
| 9Bh     | _       | Unimpleme     | nted           |                         |                |              |                  |                |          | _                             | _               |
| 9Ch     | —       | Unimplemented |                |                         |                |              |                  |                |          | -                             | —               |
| 9Dh     | —       | Unimpleme     | nted           |                         |                |              |                  |                |          | —                             | —               |
| 9Eh     | ADRESL  | A/D Result    | Register Low   | 1                       |                |              |                  |                |          | xxxx xxxx                     | 79              |
| 9Fh     | ADCON1  | —             | _              |                         | —              | _            | PCFG2            | PCFG1          | PCFG0    | 000                           | 76              |

#### SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED) **TABLE 2-1:**

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0'. Shaded locations are unimplemented, read as '0'. Note 1: These pixels do not display, but can be used as general purpose RAM.

#### 2.3.3 INTCON REGISTER

The INTCON Register is a readable and writable register which contains various enable and flag bits for the TMR0 register overflow, RB Port change and external RB0/INT pin interrupts.

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

### **REGISTER 2**

| R/W-0                  | R/W-0                            | R/W-0                          | R/W-0                          | R/W-0              | R/W-0         | R/W-0       |
|------------------------|----------------------------------|--------------------------------|--------------------------------|--------------------|---------------|-------------|
| GIE                    | PEIE                             | TMR0IE                         | INTE                           | RBIE               | TMR0IF        | INTF        |
| bit 7                  |                                  |                                |                                |                    |               |             |
| GIE: Glob              | oal Interrupt E                  | nable bit                      |                                |                    |               |             |
| 1 = Enab<br>0 = Disab  | les all unmas<br>les all interru | ked interrup<br>pts            | ts                             |                    |               |             |
| PEIE/GE                | L: Periphera                     | Interrupt Er                   | nable bit                      |                    |               |             |
| 1 = Enab<br>0 = Disab  | les all unmas<br>les all periph  | ked periphe<br>eral interrup   | ral interrupts<br>ts           | 5                  |               |             |
| TMR0IE:                | TMR0 Overfl                      | ow Interrupt                   | Enable bit                     |                    |               |             |
| 1 = Enab<br>0 = Disab  | les the TMR0<br>les the TMR0     | overflow int<br>overflow in    | errupt<br>terrupt              |                    |               |             |
| INTE: RB               | 0/INT0 Exter                     | nal Interrupt                  | Enable bit                     |                    |               |             |
| 1 = Enab<br>0 = Disab  | les the RB0/I<br>les the RB0/I   | NT external<br>NT external     | interrupt<br>interrupt         |                    |               |             |
| RBIE: RE               | B Port Change                    | e Interrupt E                  | nable bit                      |                    |               |             |
| 1 = Enab<br>0 = Disab  | les the RB po<br>les the RB po   | ort change in<br>ort change ir | terrupt<br>nterrupt            |                    |               |             |
| TMR0IF:                | TMR0 Overfl                      | ow Interrupt                   | Flag bit                       |                    |               |             |
| 1 = TMR(<br>0 = TMR(   | ) register has<br>) register did | overflowed<br>not overflow     | (must be cle                   | eared in soft      | ware)         |             |
| INTF: RB               | 0/INT0 Exter                     | nal Interrupt                  | Flag bit                       |                    |               |             |
| 1 = The F<br>0 = The F | RB0/INT exter<br>RB0/INT exter   | nal interrupt                  | t occurred (r<br>t did not occ | nust be clea<br>ur | red in softwa | are)        |
| RBIF: RE               | Port Change                      | e Interrupt Fl                 | ag bit                         |                    |               |             |
| 1 = At lea             | st one of the                    | RB7:RB4 pi                     | ns changed                     | state (must        | be cleared i  | n software) |

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

### 6.0 TIMER1 MODULE

Timer1 is a 16-bit timer/counter consisting of two 8-bit registers (TMR1H and TMR1L), which are readable and writable. The TMR1 Register pair (TMR1H:TMR1L) increments from 0000h to FFFFh and rolls over to 0000h. The TMR1 Interrupt, if enabled, is generated on overflow, which is latched in interrupt flag bit, TMR1IF (PIR1<0>). This interrupt can be enabled/disabled by setting/clearing TMR1 interrupt enable bit, TMR1IE (PIE1<0>).

Timer1 can operate in one of two modes:

- As a timer
- As a counter

The operating mode is determined by the clock select bit, TMR1CS (T1CON<1>).

In Timer mode, Timer1 increments every instruction cycle. In Counter mode, it increments on every rising edge of the external clock input.

Timer1 can be turned on and off using the control bit TMR1ON (T1CON<0>).

Timer1 also has an internal "RESET input". This RESET can be generated by the CCP module (Section 8.0). Register 6-1 shows the Timer1 control register.

When the Timer1 oscillator is enabled (T1OSCEN is set), the RC1/T1OSI and RC0/T1OSO/T1CKI pins become inputs, regardless of the TRISC<1:0>. RC1 and RC0 will be read as '0'.

| U-0             | U-0  | R/W-0  | R/W-0   | R/W-0  | R/W-0   | R/W-0   | R/W-0   |
|-----------------|--|--|---|--|---|---|---|
|                 | —  | T1CKPS1  | T1CKPS0   | T1OSCEN  | T1SYNC  | TMR1CS  | TMR10N  |
| bit 7           |  |  |   |  |   |   | bit 0   |
| Unimplem        | ented: Rea   | 0' as '0'  |   |  |   |   |   |
| TICKDO          |  | Timor1 Input   | Clock Proce   | ala Salaat hiti  |   |   |   |
| 11 = 1.8 P      | rescale vali   |  |   |  | 5   |   |   |
| 10 = 1:4 P      | rescale valu   | ue   |   |  |   |   |   |
| 01 = 1:2 P      | rescale valu   | ue   |   |  |   |   |   |
| 00 = 1:1 P      | rescale valu   | ue   |   |  |   |   |   |
| T10SCEN         | l: Timer1 Os   | scillator Enab   | le Control bit  |  |   |   |   |
| 1 = Oscillar    | ator is enab   | off  |   |  |   |   |   |
|                 | alor is shul-  | -011   |   |  |   |   |   |
| Note: The       | oscillator in  | overter and fe   | edback resist   | or are turned  | off to elim   | inate powe  | r drain.  |
| T1SYNC:         | Timer1 Exte  | ernal Clock In   | put Synchron  | ization Contr  | ol bit  |   |   |
| TMR1CS =        | <u>= 1:</u>  |  |   |  |   |   |   |
| 1 = Do no       | t synchroni  | ze external cl   | ock input   |  |   |   |   |
| 0 = Synch       | nronize exte   | ernal clock inp  | out   |  |   |   |   |
| <u>TMR1C5 =</u> | <u>= U:</u><br>iaporod Tin   | nor1 upon the  | internal alaa   |  | 100 - 0   |   |   |
|                 | Timor1 Clo   |  | loct hit  |  | 100 = 0.  |   |   |
| 1 - Evtern      | al clock from  | n nin T1CKL  | on the rising   | (anha  |   |   |   |
| 0 = Interna     | al clock (Fo:  | sc/4)  | on the namy   | euge)  |   |   |   |
| TMR10N:         | Timer1 On  | bit  |   |  |   |   |   |
| 1 = Enabl       | es Timer1  |  |   |  |   |   |   |
| 0 = Stops       | Timer1   |  |   |  |   |   |   |
| Legend.         |  |  |   |  |   |   | ]   |
| R = Reada       | ahle hit   | W = W  | ritable bit   | LI = Unimpl  | emented h   | it read as '  | n'  |
| - n = Value     | e at POR   | '1' = Ri   | it is set   | '0' = Bit is c   | leared  | x = Bit is u  | hknown  |
|                 | U-0<br>bit 7<br>Unimplem<br>T1CKPS1<br>11 = 1:8 P<br>10 = 1:4 P<br>01 = 1:2 P<br>00 = 1:1 P<br>T1OSCEN<br>1 = Oscill<br>0 = Oscill<br>Note: The<br>T1SYNC:<br>TMR1CS:<br>1 = Do nc<br>0 = Synch<br>TMR1CS:<br>1 = Do nc<br>0 = Synch<br>TMR1CS:<br>1 = Extern<br>0 = Interna<br>TMR1ON:<br>1 = Enabl<br>0 = Stops<br>Legend:<br>R = Reada<br>- n = Value | U-0U-0 $ -$ bit 7Unimplemented: ReaTICKPS1:TICKPS0:11 = 1:8 Prescale value10 = 1:4 Prescale value01 = 1:2 Prescale value00 = 1:1 Prescale value00 = 1:1 Prescale valueTIOSCEN: Timer1 Os1 = Oscillator is enable0 = Oscillator is enable0 = Oscillator is shuteNote: The oscillator inTISYNC: Timer1 Os1 = Do not synchroni0 = Synchronize exterTMR1CS = 0:This bit is ignored. TimerTMR1CS = 0:This bit is ignored. TimerTMR1CS = 0:TMR1CS: Timer1 Clo1 = External clock (ForTMR1ON: Timer1 On1 = Enables Timer10 = Stops Timer10 = Stops Timer1Legend:R = Readable bit- n = Value at POR | U-0U-0R/W-0T1CKPS1bit 7Unimplemented: Read as '0'T1CKPS1:T1CKPS0: Timer1 Input11 = 1:8 Prescale value10 = 1:4 Prescale value10 = 1:4 Prescale value01 = 1:2 Prescale value00 = 1:1 Prescale valueT1OSCEN: Timer1 Oscillator Enabled1 = Oscillator is enabled0 = Oscillator is shut-offNote: The oscillator inverter and feT1SYNC: Timer1 External Clock InTMR1CS = 1:1 = Do not synchronize external clock inpTMR1CS = 0:This bit is ignored. Timer1 uses theTMR1CS = 0:This bit is ignored. Timer1 uses theTMR1CS: Timer1 Clock Source Se1 = External clock (Fosc/4)TMR1ON: Timer1 On bit1 = Enables Timer10 = Stops Timer1Legend:R = Readable bitW = W- n = Value at POR'1' = Bit | U-0U-0R/W-0R/W-0 $-$ T1CKPS1T1CKPS0bit 7Unimplemented: Read as '0'T1CKPS1:T1CKPS0: Timer1 Input Clock Presca11 = 1:8 Prescale value10 = 1:4 Prescale value01 = 1:2 Prescale value00 = 1:1 Prescale valueT1OSCEN: Timer1 Oscillator Enable Control bit1 = Oscillator is enabled0 = Oscillator is enabled0 = Oscillator is shut-offNote: The oscillator inverter and feedback resistTISYNC: Timer1 External Clock Input SynchronTMR1CS = 1:1 = Do not synchronize external clock input0 = Synchronize external clock inputTMR1CS = 0:This bit is ignored. Timer1 uses the internal clockTMR1CS: Timer1 Clock Source Select bit1 = External clock (FOSC/4)TMR1ON: Timer1 On bit1 = Enables Timer10 = Stops Timer1 | U-0       U-0       R/W-0       R/W-0       R/W-0         —       —       T1CKPS1       T1CKPS0       T1OSCEN         bit 7         Unimplemented: Read as '0'         T1CKPS1:T1CKPS0: Timer1 Input Clock Prescale Select bits         11 = 1:8 Prescale value       10 = 1:4 Prescale value       11 = 1:2 Prescale value         00 = 1:1 Prescale value       00 = 1:1 Prescale value       11 = 0 coscillator is enabled         01 = 0scillator is enabled       0       - Oscillator is shut-off         Note: The oscillator inverter and feedback resistor are turned       T1SYNC: Timer1 External Clock Input Synchronization Contration Contration         TMR1CS = 1:       1       = Do not synchronize external clock input       0         0       = Synchronize external clock input       10 = Synchronize external clock input         11 = Do not synchronize external clock input       11 = External clock from pin T1CKI (on the rising edge)         0       = Internal clock (Fosc/4)       11 = External clock (Fosc/4)         TMR1ON: Timer1 On bit       1       = Enables Timer1         0       = Stops Timer1       12 = Bit is set       '0' = Bit is contracted by the set of the set o | U-0       U-0       R/W-0       R/W-0       R/W-0       R/W-0 | U-0       U-0       R/W-0       R/W-0       R/W-0       R/W-0       R/W-0       R/W-0         -       -       T1CKPS1       T1CKPS0       T1OSCEN       T1SYNC       TMR1CS         bit 7         Unimplemented: Read as '0'         T1CKPS0: Timer1 Input Clock Prescale Select bits         11 = 1:8 Prescale value       0       = 1:4 Prescale value       0       = 1:12 Prescale value         00 = 1:1 Prescale value       0       = 1:1 Prescale value       0       = 1:1 Prescale value         00 = 1:1 Prescale value       0       = 0 = 0 colllator is enabled       0       = 0 colllator is enabled         0 = 0 = Oscillator is shut-off       Note: The oscillator inverter and feedback resistor are turned off to eliminate power         TISYNC: Timer1 External Clock Input Synchronization Control bit         TMR1CS = 1:         1 = Do not synchronize external clock input         0 = Synchronize external clock input         TMR1CS = 0:         TMR1CS = 0:         TMR1CS: Timer1 Clock Source Select bit         1 = External clock (Fosc/4)         TMR1CN: Timer1 On bit         1 = Enables Timer1         0 = Stops Timer1 <td< td=""></td<> |

### REGISTER 6-1: T1CON: TIMER1 CONTROL REGISTER (ADDRESS 10h)

### 9.1 SPI Mode

The SPI mode allows 8-bits of data to be synchronously transmitted and received simultaneously. To accomplish communication, typically three pins are used:

- Serial Data Out (SDO) RC5/SDO
- Serial Data In (SDI) RC4/SDI
- Serial Clock (SCK) RC3/SCK

Additionally, a fourth pin may be used when in a Slave mode of operation:

• Slave Select (SS) RA5/AN4/SS

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits in the SSPCON register (SSPCON<5:0>) and SSPSTAT<7:6>. These control bits allow the following to be specified:

- Master mode (SCK is the clock output)
- Slave mode (SCK is the clock input)
- Clock Polarity (Idle state of SCK)
- Clock Edge (output data on rising/falling edge of SCK)
- Clock Rate (Master mode only)
- · Slave Select mode (Slave mode only)

The SSP consists of a transmit/receive shift register (SSPSR) and a buffer register (SSPBUF). The SSPSR shifts the data in and out of the device, MSb first. The SSPBUF holds the data that was written to the SSPSR. until the received data is ready. Once the 8-bits of data have been received, that byte is moved to the SSPBUF register. Then, the buffer full detect bit, BF (SSPSTAT<0>), and interrupt flag bit, SSPIF (PIR1<3>), are set. This double buffering of the received data (SSPBUF) allows the next byte to start reception before reading the data that was just received. Any write to the SSPBUF register during transmission/reception of data will be ignored, and the write collision detect bit, WCOL (SSPCON<7>), will be set. User software must clear the WCOL bit so that it can be determined if the following write(s) to the SSPBUF register completed successfully. When the application software is expecting to receive valid data, the SSPBUF should be read before the next byte of data to transfer is written to the SSPBUF. Buffer full bit, BF (SSPSTAT<0>), indicates when SSPBUF has been loaded with the received data (transmission is complete). When the SSPBUF is read, bit BF is cleared. This data may be irrelevant if the SPI is only a transmitter. Generally, the SSP interrupt is used to determine when the transmission/reception has completed. The SSPBUF must be read and/or written. If the interrupt method is not going to be used, then software polling can be done to ensure that a write collision does not occur. Example 9-1 shows the loading of the SSPBUF (SSPSR) for data transmission. The MOVWF RXDATA instruction (shaded) is only required if the received data is meaningful.

### EXAMPLE 9-1: LOADING THE SSPBUF (SSPSR) REGISTER

|      | BCF<br>BSF | STATUS,<br>STATUS,  | RP1<br>RP0  | ;Select Bank1<br>;   |
|------|------------|---|---|--|
| LOOP | BTFSS      | SSPSTAT,  | BF  | ;Has data been<br>;received<br>;(transmit<br>;complete)?   |
|      | GOTO       | LOOP  |   | ;NO  |
|      | BCF        | STATUS,   | RP0   | ;Select Bank0  |
|      | MOVF       | SSPBUF,   | W   | ;W reg = contents<br>;of SSPBUF  |
|      | MOVWF      | RXDATA  |   | ;Save in user RAM  |
|      | MOVF       | TXDATA,   | W   | ;W reg = contents<br>; of TXDATA   |
|      | MOVWF      | SSPBUF  |   | ;New data to xmit  |
|      | LOOP       | BCF<br>BSF<br>LOOP BTFSS<br>GOTO<br>BCF<br>MOVF<br>MOVWF<br>MOVWF | BCF STATUS,<br>BSF STATUS,<br>LOOP BTFSS SSPSTAT,<br>GOTO LOOP<br>BCF STATUS,<br>MOVF SSPBUF,<br>MOVWF RXDATA<br>MOVF TXDATA,<br>MOVWF SSPBUF | BCF STATUS, RP1<br>BSF STATUS, RP0<br>LOOP BTFSS SSPSTAT, BF<br>GOTO LOOP<br>BCF STATUS, RP0<br>MOVF SSPBUF, W<br>MOVWF RXDATA<br>MOVF TXDATA, W<br>MOVWF SSPBUF |

The block diagram of the SSP module, when in SPI mode (Figure 9-1), shows that the SSPSR is not directly readable or writable, and can only be accessed from addressing the SSPBUF register. Additionally, the SSP status register (SSPSTAT) indicates the various status conditions.

FIGURE 9-1:

### SSP BLOCK DIAGRAM (SPI MODE)



### 9.3.1.2 Reception

When the R/W bit of the address byte is clear and an address match occurs, the R/W bit of the SSPSTAT register is cleared. The received address is loaded into the SSPBUF register.

When the address <u>byte</u> overflow condition exists, then no Acknowledge (ACK) pulse is given. An overflow condition is defined as either bit BF (SSPSTAT<0>) is set, or bit SSPOV (SSPCON<6>) is set.

An SSP interrupt is generated for each data transfer byte. Flag bit SSPIF (PIR1<3>) must be cleared in software. The SSPSTAT register is used to determine the status of the byte.





### 9.3.1.3 Transmission

When the R/W bit of the incoming address byte is set and an address match occurs, the R/W bit of the SSPSTAT register is set. The received address is loaded into the SSPBUF register. The ACK pulse will be sent on the ninth bit, and pin RC3/SCK/SCL is held low. The transmit data must be loaded into the SSPBUF register, which also loads the SSPSR register. Then, pin RC3/SCK/SCL should be enabled by setting bit CKP (SSPCON<4>). The master must monitor the SCL pin prior to asserting another clock pulse. The slave devices may be holding off the master by stretching the clock. The eight data bits are shifted out on the falling edge of the SCL input. This ensures that the SDA signal is valid during the SCL high time (Figure 9-18). An SSP interrupt is generated for each data transfer byte. Flag bit SSPIF must be cleared in software, and the SSPSTAT register is used to determine the status of the byte. Flag bit SSPIF is set on the falling edge of the ninth clock pulse.

As a slave-transmitter, the  $\overline{ACK}$  pulse from the master-receiver is latched on the rising edge of the ninth SCL input pulse. If the SDA line was high (not  $\overline{ACK}$ ), then the data transfer is complete. When the  $\overline{ACK}$  is latched by the slave, the slave logic is reset and the slave then monitors for another occurrence of the START bit. If the SDA line was low ( $\overline{ACK}$ ), the transmit data must be loaded into the SSPBUF register, which also loads the SSPSR register. Then, pin RC3/SCK/SCL should be enabled by setting bit CKP.







### 10.5 A/D Operation During SLEEP

The A/D module can operate during SLEEP mode. This requires that the A/D clock source be set to RC (ADCS<1:0> = 11). When the RC clock source is selected, the A/D module waits one instruction cycle before starting the conversion. This allows the SLEEP instruction to be executed, which eliminates all digital switching noise from the conversion. When the conversion is completed, the GO/DONE bit will be cleared and the result loaded into the ADRES register. If the A/D interrupt is enabled, the device will wake-up from SLEEP. If the A/D interrupt is not enabled, the ADON bit will remain set.

When the A/D clock source is another clock option (not RC), a SLEEP instruction will cause the present conversion to be aborted and the A/D module to be turned off, though the ADON bit will remain set.

Turning off the A/D places the A/D module in its lowest current consumption state.

| Note: | For the A/D module to operate in SLEEP,   |
|-------|---|
|       | the A/D clock source must be set to RC    |
|       | (ADCS < 1:0 > = 11). To allow the conver- |
|       | sion to occur during SLEEP, ensure the    |
|       | SLEEP instruction immediately follows the |
|       | instruction that sets the GO/DONE bit.    |

### 10.6 Effects of a RESET

A device RESET forces all registers to their RESET state. This forces the A/D module to be turned off, and any conversion is aborted. All A/D input pins are configured as analog inputs.

The value that is in the ADRESH:ADRESL registers is not modified for a Power-on Reset. The ADRESH:ADRESL registers will contain unknown data after a Power-on Reset.

| Address | Name   | Bit 7                         | Bit 6       | Bit 5      | Bit 4                        | Bit 3    | Bit 2       | Bit 1    | Bit 0  | POR,<br>BOR | MCLR,<br>WDT |
|---------|--------|-------------------------------|-------------|------------|------------------------------|----------|-------------|----------|--------|-------------|--------------|
| 0Bh     | INTCON | GIE                           | PEIE        | TMR0IE     | INTE                         | RBIE     | TMR0IF      | INTF     | RBIF   | 0000 000x   | 0000 000u    |
| 0Ch     | PIR1   | LCDIF                         | ADIF        | (1)        | (1)                          | SSPIF    | CCP1IF      | TMR2IF   | TMR1IF | r0rr 0000   | r0rr 0000    |
| 8Ch     | PIE1   | LCDIE                         | ADIE        | (1)        | (1)                          | SSPIE    | CCP1IE      | TMR2IE   | TMR1IE | r0rr 0000   | r0rr 0000    |
| 1Eh     | ADRESH | A/D Result Register High Byte |             |            |                              |          |             |          |        | xxxx xxxx   | uuuu uuuu    |
| 9Eh     | ADRESL | A/D Resu                      | ılt Registe | r Low Byte | e                            |          |             |          |        | xxxx xxxx   | uuuu uuuu    |
| 1Fh     | ADCON0 | ADCS1                         | ADCS0       | CHS2       | CHS1                         | CHS0     | GO/DONE     | —        | ADON   | 0000 00-0   | 0000 00-0    |
| 9Fh     | ADCON1 | ADFM                          | _           | —          | —                            | PCFG3    | PCFG2       | PCFG1    | PCFG0  | 0- 0000     | 0- 0000      |
| 85h     | TRISA  | _                             | _           | PORTA D    | ORTA Data Direction Register |          |             |          |        | 11 1111     | 11 1111      |
| 05h     | PORTA  | _                             | _           | PORTA D    | ata Latch                    | when wri | tten: PORTA | pins whe | n read | 0x 0000     | 0u 0000      |

### TABLE 10-2: REGISTERS/BITS ASSOCIATED WITH A/D

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used for A/D conversion. **Note 1:** These bits are reserved; always maintain these bits clear.

### 11.1 LCD Timing

The LCD module has 3 possible clock source inputs and supports static, 1/2, 1/3, and 1/4 multiplexing.

### 11.1.1 TIMING CLOCK SOURCE SELECTION

The clock sources for the LCD timing generation are:

- Internal RC oscillator
- Timer1 oscillator
- System clock divided by 256

The first timing source is an internal RC oscillator which runs at a nominal frequency of 14 kHz. This oscillator provides a lower speed clock which may be used to continue running the LCD while the processor is in SLEEP. The RC oscillator will power-down when it is not selected or when the LCD module is disabled. The second source is the Timer1 external oscillator. This oscillator provides a lower speed clock which may be used to continue running the LCD while the processor is in SLEEP. It is assumed that the frequency provided on this oscillator will be 32 kHz. To use the Timer1 oscillator as a LCD module clock source, it is only necessary to set the T1OSCEN (T1CON<3>) bit.

The third source is the system clock divided by 256. This divider ratio is chosen to provide about 32 kHz output when the external oscillator is 8 MHz. The divider is not programmable. Instead the LCDPS register is used to set the LCD frame clock rate.

All of the clock sources are selected with bits CS1:CS0 (LCDCON<3:2>). Refer to Register 11-1 for details of the register programming.



FIGURE 11-6: LCD CLOCK GENERATION

### 12.0 SPECIAL FEATURES OF THE CPU

What sets a microcontroller apart from other processors are special circuits to deal with the needs of real time applications. The PIC16CXXX family has a host of such features, intended to maximize system reliability, minimize cost through elimination of external components, provide power saving operating modes and offer code protection. These are:

- Oscillator Selection
- RESET
  - Power-on Reset (POR)
  - Power-up Timer (PWRT)
  - Oscillator Start-up Timer (OST)
  - Brown-out Reset (BOR)
- Interrupts
- Watchdog Timer (WDT)
- SLEEP
- Code Protection
- ID Locations
- In-Circuit Serial Programming

The PIC16CXXX has a Watchdog Timer which can be shut-off only through configuration bits. It runs off its own RC oscillator for added reliability.

There are two timers that offer necessary delays on power-up. One is the Oscillator Start-up Timer (OST), intended to keep the chip in RESET until the crystal oscillator is stable. The other is the Power-up Timer (PWRT), which provides a fixed delay of 72 ms (nominal) on power-up only, designed to keep the part in RESET while the power supply stabilizes. With these two timers on-chip, most applications need no external RESET circuitry.

SLEEP mode is designed to offer a very low current power-down mode. The user can wake-up from SLEEP through external RESET, Watchdog Timer Wake-up, or through an interrupt.

Several oscillator options are also made available to allow the part to fit the application. The RC oscillator option saves system cost, while the LP crystal option saves power. A set of configuration bits are used to select various options.

### 12.1 Configuration Bits

The configuration bits can be programmed (read as '0'), or left unprogrammed (read as '1'), to select various device configurations. These bits are mapped in program memory location 2007h.

The user will note that address 2007h is beyond the user program memory space and can be accessed only during programming.

### 12.2 Oscillator Configurations

### 12.2.1 OSCILLATOR TYPES

The PIC16CXXX can be operated in four different oscillator modes. The user can program two configuration bits (FOSC1 and FOSC0) to select one of these four modes:

- LP Low Power Crystal
- XT Crystal/Resonator
- HS High Speed Crystal/Resonator
- RC Resistor/Capacitor

## 12.2.2 CRYSTAL OSCILLATOR/CERAMIC RESONATORS

In XT, LP, or HS modes a crystal or ceramic resonator is connected to the OSC1/CLKIN and OSC2/CLKOUT pins to establish oscillation (Figure 12-1). The PIC16CXXX oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may give a frequency out of the crystal manufacturers specifications. When in XT, LP, or HS modes, the device can have an external clock source to drive the OSC1/CLKIN pin (Figure 12-2).

### FIGURE 12-1: CRYSTAL/CERAMIC RESONATOR OPERATION (HS, XT OR LP OSC CONFIGURATION)



### FIGURE 12-2: EXTERNAL CLOCK INPUT OPERATION (HS, XT OR LP OSC



### TABLE 12-1: CERAMIC RESONATORS

| Ranges Tested: |                |                        |             |  |  |  |  |  |  |
|----------------|----------------|------------------------|-------------|--|--|--|--|--|--|
| Mode           | Freq.          | C1                     | RC21        |  |  |  |  |  |  |
|                | 455 kHz        | 687 100 pE             | 68 - 100 pF |  |  |  |  |  |  |
| XT             | 2.0 MH2 \      | (\  <b> 1</b> 5\}68`pF | 15 - 68 pF  |  |  |  |  |  |  |
| (              | 4.0 MHz        | 15 - 68 pF             | 15 - 68 pF  |  |  |  |  |  |  |
| THS C          | 80 MHz         | 10 - 68 pF             | 10 - 68 pF  |  |  |  |  |  |  |
| These valu     | ues are for de | sian auidance          | only        |  |  |  |  |  |  |

See notes following Table 12-2.

## TABLE 12-2:CAPACITOR SELECTION FOR<br/>CRYSTAL OSCILLATOR

| Osc Type    | Crystal<br>Freq. | Cap. Range<br>C1 | Cap.<br>Range<br>C2 |
|-------------|------------------|------------------|---------------------|
| ID          | 32 kHz           | 33 pF            | 33.pf               |
| LI          | 200 kHz          | 15 pF            | HttpH               |
|             | 200 kHz          | 47-68 pt         | ≫47-68 pF           |
| XT          | _1 MHz[]         | NU5pF            | 15 pF               |
|             | AMHZ             | 15 pF            | 15 pF               |
| RA          | 4 MHz            | 15 pF            | 15 pF               |
| 1913        | 8 MHz            | 15-33 pF         | 15-33 pF            |
| Those value | os aro for d     | docian auidanco  | only                |

These values are for design guidance only. See notes following this table.

### Note 1: Recommended ranges of C1 and C2 are depicted in Table 12-1.

- 2: Higher capacitance increases the stability of the oscillator, but also increases the start-up time.
- 3: Since each resonator/crystal has its own characteristics, the user should consult the resonator/crystal manufacturer for appropriate values of external components.
- **4:** Rs may be required in HS mode, as well as XT mode, to avoid overdriving crystals with low drive level specification.

### 13.0 INSTRUCTION SET SUMMARY

Each PIC16CXXX instruction is a 14-bit word divided into an OPCODE which specifies the instruction type and one or more operands which further specify the operation of the instruction. The PIC16CXXX instruction set summary in Table 13-2 lists **byte-oriented**, **bitoriented**, and **literal and control** operations. Table 13-1 shows the opcode field descriptions.

The instruction set is highly orthogonal and is grouped into three basic categories:

- Byte-oriented operations
- Bit-oriented operations
- Literal and control operations

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the W register. If 'd' is one, the result is placed in the file register specified in the instruction.

For **bit-oriented** instructions, 'b' represents a bit field designator which selects the number of the bit affected by the operation, while 'f' represents the address of the file in which the bit is located.

For **literal and control** operations, 'k' represents an eight or eleven bit constant or literal value.

### FIGURE 13-1: GENERAL FORMAT FOR INSTRUCTIONS

| Byte-oriented file               | regis              | ster or      | berat | tions       |   |  |
|----------------------------------|--------------------|--------------|-------|-------------|---|--|
| 13                               | 8                  | 7            | 6     |             | 0 |  |
| OPCODE                           |                    | d            |       | f (FILE #)  |   |  |
| d = 0 for dest<br>d = 1 for dest | tinati<br>tinati   | on W<br>on f |       |             |   |  |
| f = 7-bit file r                 | egis               | ter ad       | dres  | S           |   |  |
| <b>Bit-oriented</b> file re      | aiste              | or one       | ratio | ine         |   |  |
| 13                               | 10 10              | 9<br>9       | 7     | 6           | 0 |  |
| OPCODE                           |                    | b (Bl        | T #)  | f (FILE #)  |   |  |
| Literal and contro<br>General    | ol op              | eratio       | ns    |             |   |  |
| 13                               |                    | 8            | 7     |             | 0 |  |
| OPCODE                           |                    |              |       | k (literal) |   |  |
| k = 8-bit imm                    | nedia              | ite val      | ue    |             |   |  |
| CALL and GOTO in                 | struc              | tions        | only  |             |   |  |
| 13 11                            | 10                 |              |       |             | 0 |  |
| OPCODE                           | OPCODE k (literal) |              |       |             |   |  |
| k = 11-bit im                    | medi               | ate va       | alue  |             |   |  |

# TABLE 13-1: OPCODE FIELD DESCRIPTIONS

| Field         | Description   |
|---------------|---|
| f             | Register file address (0x00 to 0x7F)  |
| W             | Working register (accumulator)  |
| b             | Bit address within an 8-bit file register   |
| k             | Literal field, constant data or label   |
| x             | Don't care location (= 0 or 1).<br>The assembler will generate code with $x = 0$ .<br>It is the recommended form of use for com-<br>patibility with all Microchip software tools. |
| d             | Destination select; $d = 0$ : store result in W,<br>d = 1: store result in file register f.<br>Default is d = 1.  |
| label         | Label name  |
| TOS           | Top-of-Stack  |
| PC            | Program Counter   |
| PCLATH        | Program Counter High Latch  |
| GIE           | Global Interrupt Enable bit   |
| WDT           | Watchdog Timer/Counter  |
| TO            | Time-out bit  |
| PD            | Power-down bit  |
| dest          | Destination either the W register or the<br>specified register file location  |
| []            | Options   |
| ()            | Contents  |
| $\rightarrow$ | Assigned to   |
| < >           | Register bit field  |
| E             | In the set of   |
| italics       | User defined term (font is courier)   |

All instructions are executed within one single instruction cycle, unless a conditional test is true, or the program counter is changed, as a result of an instruction. In this case, the execution takes two instruction cycles, with the second cycle executed as a NOP. One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1  $\mu$ s. If a conditional test is true, or the program counter is changed, as a result of an instruction, the instruction execution time is 2  $\mu$ s.

Table 13-2 lists the instructions recognized by the MPASM<sup>TM</sup> assembler.

Figure 13-1 shows the general formats that the instructions can have.

| Note: | To maintain upward compatibility with |
|-------|---------------------------------------|
|       | future PIC16CXXX products, do not use |
|       | the OPTION and TRIS instructions.     |

All examples use the format `0xnn' to represent a hexadecimal number.

### TABLE 13-2: PIC16CXXX INSTRUCTION SET

| Mnemonic, |          | Description                  | Cycles | 14-Bit Opcode |         |          | Status   |          | Notes |
|-----------|----------|------------------------------|--------|---------------|---------|----------|----------|----------|-------|
| Opera     | nds      | Description                  | Cycles | MSb           |         |          | LSb      | Affected | NOLES |
| BYTE-OR   | ENTED I  | FILE REGISTER OPERATIONS     |        |               |         |          |          |          |       |
| ADDWF     | f, d     | Add W and f                  | 1      | 00            | 0111    | dfff     | ffff     | C,DC,Z   | 1,2   |
| ANDWF     | f, d     | AND W with f                 | 1      | 00            | 0101    | dfff     | ffff     | Z        | 1,2   |
| CLRF      | f        | Clear f                      | 1      | 00            | 0001    | lfff     | ffff     | Z        | 2     |
| CLRW      | -        | Clear W                      | 1      | 00            | 0001    | 0xxx     | xxxx     | Z        |       |
| COMF      | f, d     | Complement f                 | 1      | 00            | 1001    | dfff     | ffff     | Z        | 1,2   |
| DECF      | f, d     | Decrement f                  | 1      | 00            | 0011    | dfff     | ffff     | Z        | 1,2   |
| DECFSZ    | f, d     | Decrement f, Skip if 0       | 1(2)   | 00            | 1011    | dfff     | ffff     |          | 1,2,3 |
| INCF      | f, d     | Increment f                  | 1      | 00            | 1010    | dfff     | ffff     | Z        | 1,2   |
| INCFSZ    | f, d     | Increment f, Skip if 0       | 1(2)   | 00            | 1111    | dfff     | ffff     |          | 1,2,3 |
| IORWF     | f, d     | Inclusive OR W with f        | 1      | 00            | 0100    | dfff     | ffff     | Z        | 1,2   |
| MOVF      | f, d     | Move f                       | 1      | 00            | 1000    | dfff     | ffff     | Z        | 1,2   |
| MOVWF     | f        | Move W to f                  | 1      | 00            | 0000    | lfff     | ffff     |          |       |
| NOP       | -        | No Operation                 | 1      | 00            | 0000    | 0xx0     | 0000     |          |       |
| RLF       | f, d     | Rotate Left f through Carry  | 1      | 00            | 1101    | dfff     | ffff     | С        | 1,2   |
| RRF       | f, d     | Rotate Right f through Carry | 1      | 00            | 1100    | dfff     | ffff     | С        | 1,2   |
| SUBWF     | f, d     | Subtract W from f            | 1      | 00            | 0010    | dfff     | ffff     | C,DC,Z   | 1,2   |
| SWAPF     | f, d     | Swap nibbles in f            | 1      | 00            | 1110    | dfff     | ffff     |          | 1,2   |
| XORWF     | f, d     | Exclusive OR W with f        | 1      | 00            | 0110    | dfff     | ffff     | Z        | 1,2   |
| BIT-ORIEN | NTED FIL | E REGISTER OPERATIONS        | •      |               |         |          |          |          |       |
| BCF       | f, b     | Bit Clear f                  | 1      | 01            | 00bb    | bfff     | ffff     |          | 1,2   |
| BSF       | f, b     | Bit Set f                    | 1      | 01            | 01bb    | bfff     | ffff     |          | 1,2   |
| BTFSC     | f, b     | Bit Test f, Skip if Clear    | 1 (2)  | 01            | 10bb    | bfff     | ffff     |          | 3     |
| BTFSS     | f, b     | Bit Test f, Skip if Set      | 1 (2)  | 01            | 11bb    | bfff     | ffff     |          | 3     |
| LITERAL   | AND CO   | NTROL OPERATIONS             |        | 1             |         |          |          |          |       |
| ADDLW     | k        | Add literal and W            | 1      | 11            | 111x    | kkkk     | kkkk     | C,DC,Z   |       |
| ANDLW     | k        | AND literal with W           | 1      | 11            | 1001    | kkkk     | kkkk     | Z        |       |
| CALL      | k        | Call subroutine              | 2      | 10            | 0kkk    | kkkk     | kkkk     |          |       |
| CLRWDT    | -        | Clear Watchdog Timer         | 1      | 0.0           | 0000    | 0110     | 0100     | TO,PD    |       |
| GOTO      | k        | Go to address                | 2      | 10            | 1 k k k | kkkk     | kkkk     |          |       |
| IORLW     | k        | Inclusive OR literal with W  | 1      | 11            | 1000    | kkkk     | kkkk     | Z        |       |
| MOVLW     | k        | Move literal to W            | 1      | 11            | 00xx    | kkkk     | kkkk     |          |       |
| RETFIE    | -        | Return from interrupt        | 2      | 0.0           | 0000    | 0000     | 1001     |          |       |
| RETLW     | k        | Return with literal in W     | 2      | 11            | 01xx    | kkkk     | kkkk     |          |       |
| RETURN    | -        | Return from Subroutine       | 2      | 0.0           | 0000    | 0000     | 1000     |          |       |
| SLEEP     | -        | Go into standby mode         | 1      | 00            | 0000    | 0110     | 0011     | TO,PD    |       |
| SUBLW     | k        | Subtract W from literal      | 1      | 11            | 110×    | kkkk     | kkkk     | C,DC,Z   |       |
| XORLW     | k        | Exclusive OR literal with W  | 1      | 11            | 1010    | kkkk     | kkkk     | Z        |       |
| 1         |          |                              |        | - <u>-</u> -  | TOTO    | 12121212 | 12121212 | 1        | 1     |

Note 1: When an I/O register is modified as a function of itself (e.g., MOVF PORTE, 1), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

2: If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 Module.

3: If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

| BCF   | Bit Clear f   |                         |                 |                       |  |  |  |
|---|---|-------------------------|-----------------|-----------------------|--|--|--|
| Syntax:   | [ <i>label</i> ]BCF f[,b]   |                         |                 |                       |  |  |  |
| Operands:   | $0 \le f \le 127$<br>$0 \le b \le 7$  |                         |                 |                       |  |  |  |
| Operation:  | $0 \rightarrow (f < b >)$   |                         |                 |                       |  |  |  |
| Status Affected:  | None  |                         |                 |                       |  |  |  |
| Encoding:   | 01  | 00bb                    | bfff            | ffff                  |  |  |  |
| Description:  | Bit 'b' in register 'f' is cleared.   |                         |                 |                       |  |  |  |
| Words:  | 1   |                         |                 |                       |  |  |  |
| Cycles:   | 1   |                         |                 |                       |  |  |  |
| Q Cycle Activity:   | Q1  | Q2                      | Q3              | Q4                    |  |  |  |
|   | Decode  | Read<br>register<br>'f' | Process<br>data | Write<br>register 'f' |  |  |  |
| Example   | BCF   | FLAG_                   | REG, 7          |                       |  |  |  |
| Before Instruction:<br>FLAG_RE<br>After Instruction:<br>FLAG_RE | Before Instruction:<br>FLAG_REG = 0xC7<br>After Instruction:<br>FLAG_REG = 0x47 |                         |                 |                       |  |  |  |

| BTFSC   | Bit Test,  | Skip if Cl           | ear                |                 |  |
|---|--|----------------------|--------------------|-----------------|--|
| Syntax:   | [ <i>label</i> ] E   | STFSC f              | [,b]               |                 |  |
| Operands:   | $\begin{array}{l} 0 \leq f \leq 127 \\ 0 \leq b \leq 7 \end{array}$  |                      |                    |                 |  |
| Operation:  | skip if (f<  | b>) = 0              |                    |                 |  |
| Status Affected:  | None   |                      |                    |                 |  |
| Encoding:   | 01   | 10bb                 | bfff               | ffff            |  |
| Description:  | If bit 'b' in register 'f' is '1', then the<br>next instruction is executed.<br>If bit 'b' in register 'f' is '0', then the<br>next instruction is discarded, and a<br>NOP is executed instead, making this a<br>2Tcy instruction. |                      |                    |                 |  |
| Words:  | 1  |                      |                    |                 |  |
| Cycles:   | 1(2)   |                      |                    |                 |  |
| Q Cycle Activity:   | Q1   | Q2                   | Q3                 | Q4              |  |
|   | Decode   | Read<br>register 'f' | Process<br>data    | No<br>Operation |  |
| If Skip:  | (2nd Cyc<br>Q1   | le)<br>Q2            | Q3                 | Q4              |  |
|   | No<br>Operation  | No<br>Operation      | No<br>Operation    | No<br>Operation |  |
| Example   | HERE<br>FALSE<br>TRUE  | BTFSC<br>GOTO<br>•   | FLAG,1<br>PROCESS_ | CODE            |  |
| Before Instruc<br>PC  | tion:<br>= addre   | SS HERE              |                    |                 |  |
| After Instruction:<br>if FLAG<1> = 0,<br>PC = address TRUE<br>if FLAG<1> = 1, |  |                      |                    |                 |  |

PC = address FALSE

| BSF   | Bit Set f                            |                           |      |    |  |  |  |
|---|--------------------------------------|---------------------------|------|----|--|--|--|
| Syntax:   | [ label ] B                          | [ <i>label</i> ]BSF f[,b] |      |    |  |  |  |
| Operands:   | $0 \le f \le 127$<br>$0 \le b \le 7$ |                           |      |    |  |  |  |
| Operation:  | $1 \rightarrow (f < b;$              | >)                        |      |    |  |  |  |
| Status Affected:  | None                                 |                           |      |    |  |  |  |
| Encoding:   | 01 01bb bfff ffff                    |                           |      |    |  |  |  |
| Description:  | Bit 'b' in register 'f' is set.      |                           |      |    |  |  |  |
| Words:  | 1                                    |                           |      |    |  |  |  |
| Cycles:   | 1                                    |                           |      |    |  |  |  |
| Q Cycle Activity:   | Q1                                   | Q2                        | Q3   | Q4 |  |  |  |
|   | Decode                               | Write<br>register 'f'     |      |    |  |  |  |
| Example   | BSF                                  | FLAG_RE                   | G, 7 |    |  |  |  |
| Before Instruction:<br>FLAG_REG = 0x0A<br>After Instruction:<br>FLAG_REG = 0x8A |                                      |                           |      |    |  |  |  |

| Q4 |  |
|----|--|

| XORLW   | Exclusiv  | ve OR Li            | teral wit       | h W           |  |  |  |
|---|---|---------------------|-----------------|---------------|--|--|--|
| Syntax:   | [ label ]   | XORLW               | k               |               |  |  |  |
| Operands:   | $0 \le k \le 2$   | $0 \le k \le 255$   |                 |               |  |  |  |
| Operation:  | (W) .XOR. $k \rightarrow (W)$   |                     |                 |               |  |  |  |
| Status Affected:  | Z   |                     |                 |               |  |  |  |
| Encoding:   | 11  | 1010                | kkkk            | kkkk          |  |  |  |
| Description:  | The contents of the W register are XOR'ed with the eight-bit literal 'k'. The result is placed in the W register. |                     |                 |               |  |  |  |
| Words: 1  |   |                     |                 |               |  |  |  |
| Cycles:   | 1   |                     |                 |               |  |  |  |
| Q Cycle Activity:   | Q1  | Q2                  | Q3              | Q4            |  |  |  |
|   | Decode  | Read<br>literal 'k' | Process<br>data | Write to<br>W |  |  |  |
| Example:<br>Before Instructi<br>W = 0<br>After Instruction<br>W = 0 | XORLW<br>ion:<br>xB5<br>n:<br>x1A   | 0xAF                |                 |               |  |  |  |

| XORWF                        | Exclusive OR W with f  |   |  |   |  |  |
|------------------------------|--|---|--|---|--|--|
| Syntax:                      | [label] X  | ORWF                                      | f [,d]   |   |  |  |
| Operands:                    | $\begin{array}{l} 0 \leq f \leq 127 \\ d \in [0,1] \end{array}$                |   |  |   |  |  |
| Operation:                   | (W) .XOR. (f) $\rightarrow$ (destination)                                      |   |  |   |  |  |
| Status Affected:             | Z  |   |  |   |  |  |
| Encoding:                    | 00   | 0110                                      | dfff   | ffff  |  |  |
| Description:                 | Exclusive 0<br>register wit<br>result is sto<br>'d' is 1, the<br>register 'f'. | OR the<br>h regist<br>pred in<br>result i | contents<br>ter 'f'. If 'c<br>the W re<br>s stored | of the W<br>d' is 0, the<br>gister. If<br>back in |  |  |
| Words:                       | 1  |   |  |   |  |  |
| Cycles:                      | 1  |   |  |   |  |  |
| Q Cycle Activity:            | Q1   | Q2  | Q3   | Q4  |  |  |
|                              | Decode   | Read<br>egister 'f'                       | Process<br>data                                    | Write to destination                              |  |  |
| Example                      | XORWF R  | EG  | 1  |   |  |  |
| Before Instruc<br>REG<br>W s | tion:<br>= 0xAF<br>= 0xB5  |   |  |   |  |  |
| After Instructio<br>REG =    | on:<br>= 0x1A  |   |  |   |  |  |

W = 0xB5

### 14.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK object linker combines relocatable objects created by the MPASM assembler and the MPLAB C17 and MPLAB C18 C compilers. It can also link relocatable objects from pre-compiled libraries, using directives from a linker script.

The MPLIB object librarian is a librarian for precompiled code to be used with the MPLINK object linker. When a routine from a library is called from another source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications. The MPLIB object librarian manages the creation and modification of library files.

The MPLINK object linker features include:

- Integration with MPASM assembler and MPLAB C17 and MPLAB C18 C compilers.
- Allows all memory areas to be defined as sections to provide link-time flexibility.

The MPLIB object librarian features include:

- Easier linking because single libraries can be included instead of many smaller files.
- Helps keep code maintainable by grouping related modules together.
- Allows libraries to be created and modules to be added, listed, replaced, deleted or extracted.

### 14.5 MPLAB SIM Software Simulator

The MPLAB SIM software simulator allows code development in a PC-hosted environment by simulating the PIC series microcontrollers on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a file, or user-defined key press, to any of the pins. The execution can be performed in single step, execute until break, or trace mode.

The MPLAB SIM simulator fully supports symbolic debugging using the MPLAB C17 and the MPLAB C18 C compilers and the MPASM assembler. The software simulator offers the flexibility to develop and debug code outside of the laboratory environment, making it an excellent multiproject software development tool.

### 14.6 MPLAB ICE High Performance Universal In-Circuit Emulator with MPLAB IDE

The MPLAB ICE universal in-circuit emulator is intended to provide the product development engineer with a complete microcontroller design tool set for PIC microcontrollers (MCUs). Software control of the MPLAB ICE in-circuit emulator is provided by the MPLAB Integrated Development Environment (IDE), which allows editing, building, downloading and source debugging from a single environment.

The MPLAB ICE 2000 is a full-featured emulator system with enhanced trace, trigger and data monitoring features. Interchangeable processor modules allow the system to be easily reconfigured for emulation of different processors. The universal architecture of the MPLAB ICE in-circuit emulator allows expansion to support new PIC microcontrollers.

The MPLAB ICE in-circuit emulator system has been designed as a real-time emulation system, with advanced features that are generally found on more expensive development tools. The PC platform and Microsoft<sup>®</sup> Windows environment were chosen to best make these features available to you, the end user.

### 14.7 ICEPIC In-Circuit Emulator

The ICEPIC low cost, in-circuit emulator is a solution for the Microchip Technology PIC16C5X, PIC16C6X, PIC16C7X and PIC16CXXX families of 8-bit One-Time-Programmable (OTP) microcontrollers. The modular system can support different subsets of PIC16C5X or PIC16CXXX products through the use of interchangeable personality modules, or daughter boards. The emulator is capable of emulating without target application circuitry being present.

### **15.0 ELECTRICAL CHARACTERISTICS**

| Absolute Maximum Ratings †  |                                    |
|---|------------------------------------|
| Ambient temperature under bias  | 55°C to +125°C                     |
| Storage temperature   | 65°C to +150°C                     |
| Voltage on any pin with respect to Vss (except VDD, MCLR, and RA4)  | 0.3V to (VDD + 0.3V)               |
| Voltage on VDD with respect to Vss  | 0V to +7.5V                        |
| Voltage on MCLR with respect to Vss   | 0V to +13.25V                      |
| Voltage on RA4 with respect to Vss  | 0V to +8.5V                        |
| Voltage on VLCD2, VLCD3 with respect to Vss   | 0V to +10V                         |
| Total power dissipation (Note 1)  | 1.0 W                              |
| Maximum current out of Vss pin  | 300 mA                             |
| Maximum current into VDD pin  | 250 mA                             |
| Input clamp current, Iικ (VI < 0 or VI > VDD)   | ± 20 mA                            |
| Output clamp current, Ioк (Vo < 0 or Vo > VDD)  | ± 20 mA                            |
| Maximum output current sunk by any I/O pin  | 25 mA                              |
| Maximum output current sourced by any I/O pin   | 25 mA                              |
| Maximum current sunk by all Ports combined  | 200 mA                             |
| Maximum current sourced by all Ports combined   | 200 mA                             |
| <b>Note 1:</b> Power dissipation is calculated as follows: PDIS = VDD x {IDD - $\sum$ IOH} + $\sum$ {(VDD - | VOH) x IOH} + $\Sigma$ (VOI x IOL) |

† NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

### **15.1 DC Characteristics**

| PIC16LC925/926<br>(Commercial, Industrial) |           |  |               | $\begin{array}{llllllllllllllllllllllllllllllllllll$   |              |        |   |  |
|--|-----------|--|---------------|--|--------------|--------|---|--|
| PIC16C925/926<br>(Commercial, Industrial)  |           |  |               | Standard Operating Conditions (unless otherwise stated)Operating temperature $-40^{\circ}C \leq TA \leq +85^{\circ}C$ for industrial $0^{\circ}C \leq TA \leq +70^{\circ}C$ for commercial |              |        |   |  |
| Param<br>No.                               | Sym       | Characteristic   | Min           | Тур†   | Max          | Units  | Conditions  |  |
|  | Vdd       | Supply Voltage   |               |  |              |        |   |  |
| D001<br>D001A                              |           | PIC16LC925/926   | 2.5<br>4.5    | _  | 5.5<br>5.5   | V<br>V | LP, XT and RC osc configuration<br>HS osc configuration                 |  |
| D001<br>D001A                              |           | PIC16C925/926  | 4.0<br>4.5    | _  | 5.5<br>5.5   | V<br>V | XT RC and LP osc configuration<br>HS osc configuration                  |  |
| D002                                       | Vdr       | RAM Data Retention<br>Voltage (Note 1)                           |               | 1.5  |              | $\sum$ | Device in SLEEP mode  |  |
| D003                                       | VPOR      | VDD Start Voltage<br>to ensure internal<br>Power-on Reset signal | -             | Vss  | $\mathbb{N}$ |        | See Power-on Reset section for details                                  |  |
| D004                                       | Svdd      | VDD Rise Rate<br>to ensure internal<br>Power-on Reset signal     | 0.05          | 11A  | $\swarrow$   | Ƴ/ms   | See Power-on Reset section for details (Note 6)                         |  |
| D005                                       | VBOR      | Brown-out Reset<br>voltage trip point                            | 3.65          | $\searrow$   | 4.35         | V      | BODEN bit set   |  |
|  | Idd       | Supply Current (Note 2)  |               | $\checkmark$   |              |        |   |  |
| D010                                       |           | PIC16LC925/926   | $\mathcal{A}$ | .6   | 2.0          | mA     | XT and RC osc configuration<br>Fosc = 4 MHz, VDD = 3.0V (Note 4)        |  |
| D011                                       |           |  | ~_            | 225  | 48           | μA     | LP osc configuration<br>FOSC = 32 kHz, VDD = 3.0V, WDT disabled         |  |
| D010                                       |           | PIC16C925/926  | —             | 2.7  | 5            | mA     | XT and RC osc configuration<br>FOSC = 4 MHz, VDD = 5.5V (Note 4)        |  |
| D011                                       |           |  | —             | 35   | 70           | μΑ     | LP osc configuration<br>FOSC = $32 \text{ kHz}$ , VDD = $4.0 \text{ V}$ |  |
| D012                                       | $\langle$ | $\bigcirc$   | _             | 7  | 10           | mA     | HS osc configuration<br>Fosc = 20 MHz, VDD = 5.5V                       |  |

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

- 2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, 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

- **3:** 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 hi-impedance state and tied to VDD and VSS.
- 4: For RC osc configuration, current through REXT is not included. The current through the resistor can be estimated by the formula Ir = VDD/2REXT (mA) with REXT in kOhm.
- 5: The  $\Delta$  current is the additional current consumed when this peripheral is enabled. This current should be added to the base IDD or IPD measurement.
- **6:** PWRT must be enabled for slow ramps.
- 7:  $\Delta$ ILCDT1 and  $\Delta$ ILCDRC includes the current consumed by the LCD Module and the voltage generation circuitry. This does not include current dissipated by the LCD panel.

# PIC16C925/926

### FIGURE 15-8: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS



| Param<br>No. | Symbol                     |                                   | Characterist                           | ic  | Min                                       | Тур†         | Max                   | Units | Conditions                         |
|--------------|----------------------------|-----------------------------------|--|---|---|--------------|-----------------------|-------|------------------------------------|
| 40           | Tt0H                       | T0CKI High P                      | ulse Width                             | No Prescaler  | 0.5TCY + 20                               | —            | —                     | ns    | Must also meet                     |
|              |                            |                                   |  | With Prescaler  | 10  | —            | —                     | ns    | parameter 42                       |
| 41           | Tt0L T0CKI Low Pulse Width |                                   | ulse Width                             | No Prescaler  | 0.5TCY + 20                               | —            |                       | ns    | Must also meet                     |
|              |                            |                                   |  | With Prescaler  | 10  |              | H                     | ∕ns   | parameter 42                       |
| 42           | Tt0P                       | T0CKI Period                      |  | No Prescaler  | Tcy + 40                                  | $\mathbb{N}$ | 77                    | ns    |                                    |
|              |                            |                                   |  | With Prescaler  | Greater of:                               |              | $\bigvee \rightarrow$ | ns    | N = prescale value                 |
|              |                            |                                   |  |   | 20 pr <del>TCX + 40</del><br>N            |              |                       |       | (2, 4,, 256)                       |
| 45           | Tt1H                       | T1CKI High                        | Synchronous, P                         | $\operatorname{Prescaler}(A) = A A A A A A A A A A A A A A A A A A$ | 0.5Tcy + 20                               | —            |                       | ns    | Must also meet                     |
|              |                            | Time                              | Synchronous,                           | PIC16 <b>C</b> 925(926  | 15  | —            | -                     | ns    | parameter 47                       |
|              |                            |                                   | Prescaler =<br>2,4,8                   | PIC16LC925/926  | 25  | -            | _                     | ns    |                                    |
|              |                            | (D)                               | Asynchronous                           | PIC16 <b>C</b> 925/926  | 30  | -            | —                     | ns    |                                    |
|              |                            | $) \land \frown \land$            |  | PIC16 <b>LC</b> 925/926   | 50  | -            |                       | ns    |                                    |
| 46           | Tt1L                       | Ť1CKI Low                         | Synchronous, Prescaler = 1             |   | 0.5TCY + 20                               | —            |                       | ns    |                                    |
|              |                            | Time                              | Synchronous,                           | PIC16 <b>C</b> 925/926  | 15  | —            | —                     | ns    |                                    |
|              |                            |                                   | Prescaler =<br>2,4,8                   | PIC16 <b>LC</b> 925/926   | 25  | -            | —                     | ns    | Must also meet<br>parameter 47     |
|              |                            |                                   | Asynchronous                           | PIC16 <b>C</b> 925/926  | 30  | —            | Ι                     | ns    |                                    |
|              |                            |                                   |  | PIC16 <b>LC</b> 925/926   | 50  | —            | Ι                     | ns    |                                    |
| 47           | Tt1P                       | T1CKI Input<br>Period             | Synchronous                            | PIC16 <b>C</b> 925/926  | Greater of:<br>30 or <u>Tcy + 40</u><br>N | _            | _                     | ns    | N = prescale value<br>(1, 2, 4, 8) |
|              |                            |                                   |  | PIC16 <b>LC</b> 925/926   | Greater of:<br>50 or <u>Tcy + 40</u><br>N |              |                       |       | N = prescale value<br>(1, 2, 4, 8) |
|              |                            |                                   | Asynchronous                           | PIC16 <b>C</b> 925/926  | 60  | —            | —                     | ns    |                                    |
|              |                            |                                   |  | PIC16 <b>LC</b> 925/926   | 100                                       | —            | Ι                     | ns    |                                    |
|              | Ft1                        | Timer1 oscilla<br>(oscillator ena | tor input frequen<br>bled by setting b | cy range<br>it T1OSCEN)   | DC  | -            | 200                   | kHz   |                                    |
| 48           | TCKEZtmr1                  | Delay from ex                     | ternal clock edge                      | e to timer increment  | 2Tosc                                     | —            | 7Tosc                 | —     |                                    |

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.



### FIGURE 15-13: SPI SLAVE MODE TIMING (CKE = 1)



# PIC16C925/926

| POR .  |  | 2   |
|--|--|---|
| 0  | scillator Start-up Timer (OST)   | 2   |
| P  | OR Status (POR Bit)24  | 1   |
| P  | ower Control Register (PCON)102  | 2   |
| P  | ower-on Reset (POR)  | 1   |
| P  | ower-up Timer (PWRT)97, 102  | 2   |
| R  | ESET Condition for Special Registers   | 3   |
| Ti   | me-out Sequence102   | 2   |
| Ti   | me-out Sequence on Power-up106   | 3   |
| Т  |  | l   |
| Port R   | 3 Interrupt 108  | 3   |
| PORTA  | A Contraction of the second seco |   |
| A  | ssociated Registers  | )   |
| In   | itialization29   | 9   |
| In   | itialization States104   | 1   |
| Pi   | n Functions  | )   |
| R  | A3:RA0 and RA5 Port Pins29   | )   |
| R  | A4/T0CKI Pin   | 9   |
| R  | egister29  | )   |
| TI   | RISA Register  | 9   |
| PORTE  | 3  |   |
| A  | ssociated Registers32  | 2   |
| In   | itialization   | I   |
| In   | itialization States104   | 1   |
| Pi   | n Functions  | 2   |
| R  | 30/INT Edge Select (INTEDG Bit)  | )   |
| R  | 31:RB0 Port Pins   |   |
| R  | 37:RB4 Port Pins   |   |
| R  | egister  | I   |
| TI   | RISB Register  | I   |
| PORTO  |  |   |
| A  | analistad Decistera  |   |
|  |  | 3   |
| BI   | ock Diagram (Peripheral Output Override)   | 3   |
| BI   | ock Diagram (Peripheral Output Override)   | 3   |
| BI<br>In<br>In   | ock Diagram (Peripheral Output Override)   | 3<br>3<br>3<br>1  |
| Bi<br>In<br>In<br>Pi   | ock Diagram (Peripheral Output Override)   | 3<br>3<br>3<br>1<br>3   |
| Bl<br>In<br>In<br>Pi<br>R  | socialed Registers   | 3<br>3<br>3<br>4<br>3   |
| BI<br>In<br>In<br>Pi<br>Ri<br>TI   | soctated Registers   | 333433  |
| Bi<br>In<br>Pi<br>Ri<br>TI<br>PORTI  | ock Diagram (Peripheral Output Override)   | 3<br>3<br>3<br>4<br>3<br>3<br>3<br>3<br>3   |
| BI<br>In<br>Pi<br>Ri<br>TI<br>PORTI  | ock Diagram (Peripheral Output Override)   | 3<br>3<br>3<br>4<br>3<br>3<br>3<br>5<br>5   |
| BI<br>In<br>Pi<br>Ri<br>PORTI<br>As  | ock Diagram (Peripheral Output Override)   | 333433351   |
| BI<br>In<br>Pi<br>R<br>TI<br>PORTI<br>As<br>In   | sock Diagram (Peripheral Output Override)       33         itialization       33         itialization States       104         n Functions       33         egister       33         RISC Register       33         o       ssociated Registers       33         o       ssociated Registers       35         o       ssociated Registers       36         otitalization       34       34         itialization       34   | 3 3 4 3 3 5 4 4   |
| Bi<br>In<br>Pi<br>Ri<br>PORTI<br>As<br>In<br>In<br>Pi  | sock Diagram (Peripheral Output Override)       33         itialization       33         itialization States       104         n Functions       33         egister       33         RISC Register       33         o       ssociated Registers       35         o       ssociated Registers       35         o       ssociated Registers       36         o       ssociated Register       36         o       ssociated Register       36         o       ssociated Register       37         o       ssociated Register       36  | 3 3 4 3 3 5 4 4 5   |
| Bi<br>In<br>Pri<br>R<br>R<br>TI<br>PORTI<br>As<br>In<br>Pi<br>Pi   | sock Diagram (Peripheral Output Override)         33           itialization         33           itialization States         104           n Functions         33           egister         33           RISC Register         33           o         35           ssociated Registers         35           o         35           n Functions         35           o         35           o         35           o         36           o         37           o         36           o         37           o         36           o         37           o         37           o         36           o         37           o         37           o         38           o         37           o         36           o         37           o         37           o         38           o         39           o         39           o         39           o         39           o         34   | 3 3 4 3 3 4 5 4 5 4 5 4   |
| Bi<br>In<br>Pi<br>R<br>TI<br>PORTI<br>As<br>In<br>Pi<br>Pi<br>Pi   | sock Diagram (Peripheral Output Override)       33         itialization       33         itialization States       104         n Functions       33         egister       33         RISC Register       33         o       ssociated Registers       35         o       ssociated Registers       35         o       ssociated Registers       35         o       ssociated Registers       36         n Functions       34       104         n Functions       36       36         n s < 4:0>       34       34         n s < 7:5>       34       34   | 3331333 51151   |
| Bi<br>In<br>Pri<br>PORTI<br>PORTI<br>A:<br>In<br>Pi<br>Pi<br>R   | sock Diagram (Peripheral Output Override)       33         itialization       33         itialization States       104         n Functions       33         egister       33         RISC Register       33         o       ssociated Registers       35         o       ssociated Registers       35         o       ssociated Registers       35         o       ssociated Registers       36         n Functions       34       34         ns <4:0>       34       34         ns <7:5>       34       34         egister       34       34  | 3334333 544544  |
| Bi<br>In<br>Pri<br>PORTI<br>PORTI<br>A:<br>In<br>PI<br>PI<br>R<br>TI   | sock Diagram (Peripheral Output Override)       33         itialization       33         itialization States       104         n Functions       33         egister       33         RISC Register       33         o       ssociated Registers       35         o       ssociated Registers       35         o       ssociated Registers       35         o       ssociated Registers       36         n Functions       36       36         ns <4:0>       34       34         ns <7:5>       34       34         Qister       34       34         Qister       34       34         AISD Register       34       34  | 3 3 3 1 3 3 3 5 1 1 5 1 1 1 1   |
| Bi<br>In<br>Pri<br>PORTI<br>PORTI<br>A:<br>In<br>PORTI<br>PORTI  | sock Diagram (Peripheral Output Override)       33         itialization       33         itialization States       104         n Functions       33         egister       33         RISC Register       33         ossociated Registers       35         ossociated Registers       35         ossociated Registers       35         n Functions       34         itialization       34         n Functions       35         ns <4:0>       34         ns <7:5>       34         egister       34         RISD Register       34  | 3331333 5115111   |
| Bi<br>In<br>Pri<br>PORTI<br>PORTI<br>PORTI<br>PORTI<br>PORTI   | sock Diagram (Peripheral Output Override)       33         itialization       33         itialization States       104         n Functions       33         egister       33         RISC Register       33         ossociated Registers       35         ossociated Registers       35         ossociated Registers       35         ossociated Registers       36         n Functions       36         ns <4:0>       34         ns <7:5>       34         egister       34         RISD Register       34         ssociated Registers       36         ssociated Registers       36         ssociated Registers       36  | 3331333 5115111 6   |
| Bi<br>In<br>Pi<br>R<br>TI<br>PORTI<br>PORTI<br>PORTI<br>PORTE<br>A:<br>B   | sociated Registers   | 3331333 51151111 65   |
| Bi<br>In<br>Pi<br>PORTI<br>PORTI<br>PORTI<br>PORTI<br>PORTI<br>A:<br>BI  | sock Diagram (Peripheral Output Override)       33         itialization       33         itialization States       104         n Functions       33         egister       33         RISC Register       33         ossociated Registers       35         ossociated Registers       35         ossociated Registers       35         ossociated Registers       36         ossociated Registers       36         ossociated Registers       36         ossociated Registers       36         n Functions       35         ns <4:0>       34         ns <7:5>       34         egister       34         Sociated Register       34         Sociated Register       34         Gister       34         Sociated Register       34         Sociated Register       34         Sociated Registers       36         ock Diagram       36         oth Diagram       36         oth Diagram       36         oth Diagram       36         oth Diagram       36   | 3331333 5115111 666   |
| Bi<br>In<br>Pi<br>PORTI<br>PORTI<br>PORTI<br>PORTI<br>PORTI<br>BI<br>BI<br>In<br>In  | sociated Registers   | 3         3         3         1   |
| Bi<br>In<br>Pi<br>PORTI<br>PORTI<br>PORTI<br>PORTI<br>PORTI<br>PORTI<br>Bi<br>In<br>PORTI  | sociated Registers   |   |
| Bi<br>In<br>Pi<br>PORTI<br>PORTI<br>PORTI<br>PORTE<br>PORTE<br>Bi<br>In<br>In<br>R   | sock Diagram (Peripheral Output Override)       33         itialization       33         itialization States       104         n Functions       33         egister       33         RISC Register       33         ossociated Registers       35         ossociated Registers       35         ossociated Registers       35         ossociated Registers       36         ossociated Registers       36         ossociated Registers       36         ossociated Registers       36         n Functions       35         ns <4:0>       34         ns <7:5>       34         egister       34         ISD Register       34         Essociated Registers       36         ock Diagram       36         ock Diagram       36         otalization       36         otalister       36   | 3<br>3<br>3<br>3<br>3<br>3<br>4<br>3<br>3<br>3<br>3<br>3<br>3<br>3<br>3<br>3<br>3<br>3<br>3<br>3  |
| Bi<br>In<br>PR<br>TI<br>PORTI<br>PORTI<br>PORTI<br>PORTI<br>PORTI<br>BI<br>In<br>PORTI<br>R<br>TI  | sociated Registers   | 3<br>3<br>3<br>3<br>3<br>4<br>3<br>3<br>3<br>4<br>5<br>4<br>4<br>5<br>4<br>4<br>4<br>4<br>5<br>5<br>6<br>6<br>6<br>6<br>6<br>6<br>6<br>6<br>6<br>6<br>6<br>6<br>6   |
| PORTE<br>PORTE<br>PORTE<br>PORTE<br>PORTE<br>BI<br>In<br>PORTE<br>BI<br>In<br>PORTE  | sock Diagram (Peripheral Output Override)       33         itialization       33         itialization States       104         n Functions       33         egister       33         RISC Register       33         ossociated Registers       35         ossociated Registers       35         ossociated Registers       35         ossociated Registers       36         itialization       34         itialization States       104         n Functions       35         ns <4:0>       34         ns <7:5>       34         egister       34         RISD Register       34         RISD Register       36         ock Diagram       36         ock Diagram       36         itialization States       104         egister       36         ock Diagram       36         ock Diagram       36         itialization States       104         egister       36         ock Diagram       36         ock Register       36         ock Register       36         itialization States       104   | 3         3         1   |
| PORTE<br>PORTE<br>PORTE<br>PORTE<br>PORTE<br>AS<br>BI<br>In<br>PORTE<br>PORTE  | sockated Registers       33         ock Diagram (Peripheral Output Override)       33         itialization       33         itialization States       104         n Functions       33         egister       33         RISC Register       33         ossociated Registers       35         ossociated Registers       35         ossociated Registers       35         ossociated Registers       36         itialization       34         itialization States       104         n Functions       35         ns <4:0>       34         egister       34         egister       34         egister       34         RISD Register       34         essociated Registers       36         ock Diagram       36         itialization       36         itialization States       104         n Functions       36         egister       36         itialization States       104         n Functions       36         egister       36         egister       36         itialization States       104  | 3         3         4         3         5         1         1         5         1         1         5         5         1         1         5         5         1         1         5         5         5         7         1         1         1         5         5         5         7         1         1         1         5         5         5         5         7         1         1         1         5   |
| PORTE<br>PORTE<br>PORTE<br>A:<br>In<br>PORTE<br>A:<br>BI<br>In<br>PORTE<br>A:<br>BI<br>PORTE<br>A:<br>BI<br>BI<br>A:<br>BI<br>BI<br>BI<br>BI<br>BI<br>BI<br>BI<br>BI<br>BI<br>BI<br>BI<br>BI<br>BI | sock Diagram (Peripheral Output Override)       33         itialization       33         itialization States       104         n Functions       33         egister       33         RISC Register       33         ossociated Registers       35         ossociated Registers       35         ossociated Registers       35         ossociated Registers       35         ossociated Registers       36         itialization       34         itialization States       104         n Functions       35         ns <4:0>       34         egister       34         egister       34         egister       34         RISD Register       34         egister       36         ock Diagram       36         itialization States       104         n Functions       36         ock Diagram       36         itialization States       104         n Functions       36         egister       36         egister       36         egister       36         egister       36         egister  | 3                                         |
| PORTE<br>PORTE<br>PORTE<br>A:<br>In<br>PORTE<br>A:<br>BI<br>PORTE<br>A:<br>BI<br>PORTE<br>BI<br>BI<br>BI<br>BI<br>BI<br>BI   | sociated Registers   | 3         3         3         3         3         1 <th1< th=""> <th1< th=""> <th1< th=""> <th1< th=""></th1<></th1<></th1<></th1<> |
| PORTE<br>PORTE<br>A:<br>PORTE<br>A:<br>PORTE<br>A:<br>BI<br>PORTE<br>A:<br>BI<br>PORTE<br>BI<br>BI<br>DORTE<br>BI<br>DORTE   | sock Diagram (Peripheral Output Override)       33         itialization       33         itialization States       104         n Functions       33         egister       33         cock Diagram (Peripheral Output Override)       33         itialization States       104         n Functions       33         egister       33         ssociated Registers       33         o       35         ssociated Registers       35         o       36         itialization States       104         n Functions       35         ns <4:0>       34         ns <4:0>       34         egister       34         egister       34         RISD Register       34         egister       36         ock Diagram       36         itialization States       104         n Functions       36         egister       36         ock Diagram       36         ssociated Registers       37         ock Diagram       37         ock Diagram       37         ock Diagram       37         ock Diagram   |   |
| PORTE<br>PORTE<br>A:<br>PORTE<br>A:<br>PORTE<br>A:<br>BI<br>PORTE<br>A:<br>BI<br>PORTE<br>A:<br>BI<br>DORTE<br>A:<br>BI<br>PORTE   | sock Diagram (Peripheral Output Override)       33         itialization       33         itialization       33         itialization       33         egister       33         RISC Register       33         obscience       33         segister       33         cost       33         egister       33         cost       33         cost       33         cost       33         cost       33         egister       33         cost       33         cost       33         cost       33         cost       33         cost       33         cost       34  | 3         3         3         4         3         5         4         4         5         4         4         5         4         4         5         4         4         5         5         4         4         5         5         4         4         5         5         4         4         5         5         1         4         5         5         5         7         7         5         7         7         5         7         7         5         7   |
| PORTE<br>PORTE<br>A:<br>PORTE<br>A:<br>PORTE<br>A:<br>BI<br>PORTE<br>A:<br>BI<br>PORTE<br>A:<br>BI<br>R<br>PORTE<br>R<br>R   | sociated Registers   | 3<br>3<br>3<br>3<br>3<br>3<br>4<br>3<br>3<br>3<br>5<br>4<br>4<br>5<br>4<br>4<br>5<br>4<br>4<br>5<br>4<br>5  |
| PORTE<br>PORTE<br>PORTE<br>A:<br>In<br>PORTE<br>A:<br>BI<br>PORTE<br>A:<br>BI<br>PORTE<br>A:<br>BI<br>In<br>PORTE<br>A:<br>TI<br>PORTE   | sociated Registers   | 3         3         3         3         3         3         3         1         3         3         1         3         3         1         3         3         1         3         3         3         1         3         3         1         3         3         3         1         3         3         3         1         3         3         1         3         3         1         3         3         1         3         3         1         3         3         1         3         3         1         3         3         1         3         3         1         3         3         1         3         3         3         1         3         3         1         3         3         1         3         3         1         3         3         1         3         3         1         3         3         3         1         3         3         3         1   |

| PORTG   |
|---|
| Associated Registers                          |
| Block Diagram                                 |
| Initialization                                |
| Initialization States105                      |
| Pin Functions                                 |
| Register                                      |
| TRISG Register                                |
| Postscaler, WDT                               |
| Assignment (PSA Bit)                          |
| Rate Select (PS2:PS0 Bits)                    |
| Power-down Mode (SLEEP)                       |
| Power-on Reset. See POR.                      |
| PR2   |
| Prescaler, Timer0                             |
| Assignment (PSA Bit)20                        |
| Rate Select (PS2:PS0 Bits) 20                 |
| Switching Between Timer0 and WDT 45           |
| PRO MATE II Universal Device Programmer 135   |
| Product Identification System 177             |
| Program Counter                               |
| RESET Conditions 103                          |
| Program Memory                                |
| Associated Registers                          |
| Operation During Code Protect                 |
| PMADR Register                                |
| PMCON1 Register                               |
| Program Read (Code Example)                   |
| Read  |
| Program Memory and Stack Maps11               |
| PUSH  |
| PWM Mode (CCP)56                              |
| Associated Registers                          |
| Block Diagram                                 |
| Example Frequencies/Resolutions               |
| Example Period and Duty Cycle Calculations 57 |
| D   |
| π.  |

| - |   |
|---|---|
| E |   |
| Г | ٦ |
| - |   |
|   |   |

| R/W bit                                 | 1 |
|---|---|
| RBIF bit                                | 8 |
| RC Oscillator 99, 100, 102              | 2 |
| RCV_MODE                                | 3 |
| Read-Modify-Write                       | 9 |
| Register File                           | 2 |
| Register File Map                       |   |
| PIC16C925                               | 3 |
| PIC16C92614                             | 4 |
| Registers                               |   |
| ADCON0 (A/D Control 0)                  | 5 |
| ADCON1 (A/D Control 1)                  | ô |
| CCP1CON (CCP Control)                   | 3 |
| Flag                                    | 3 |
| Initialization Conditions               | 5 |
| INTCON (Interrupt Control)21            | 1 |
| LCDCON (LCD Control)                    | 3 |
| LCDD (LCD Pixel Data, General Format)   | 2 |
| LCDPS (LCD Prescale)                    | 4 |
| LCDSE (LCD Segment Enable)              | 4 |
| OPTION REG                              | 0 |
| PCON (Power Control)                    | 4 |
| PIE2 (Peripheral Interrupt Enable 1) 22 | 2 |
| PIR1 (Peripheral Interrupt Request)     | 3 |
| PMCON1 (Program Memory Control)         | 7 |
| SSPCON (Svnc Serial Port Control)       | 0 |
| SSPSTAT (Sync Serial Port Status)       | 9 |
| STATUS                                  | 9 |
|   |   |