INTRODUCTION

PIC16C64/74 microcontrollers from Microchip Technology Inc. can be interfaced with ease into a multi-microprocessor environment using its built-in Parallel Slave Port (PSP). With their very high operating speeds (cycle times as low as 200 ns with a clock rate of 20 MHz), and an array of on-chip peripherals, these microcontrollers make ideal smart interfaces to the real world.

IMPLEMENTATION

PORTD operates as an 8-bit wide Parallel Slave Port, with PORTE providing the control signals. In parallel slave mode, PORTD is asynchronously readable and writable by the external world through the chip select (RE2/CS), Read (RE0/RD), and Write (RE1/WR) control inputs.

In order to use the Parallel Slave Port, the data direction bits in the TRISE register corresponding to RD, WR, and CS (TRISE<2:0>) must be configured as inputs (set = 1) and control bit PSPMODE (TRISE<4>) must be set.

The port pins are connected to two 8-bit latches, one for data output (from the PIC16CXXX) and one for data input. The PIC16CXXX sends data by writing to the output latch, and receives data by reading the input latch (note that the input and output latches are at the same address). In PSP mode the TRISD register is ignored, since the external device connected to the slave port controls the direction of data flow.

When the external device performs either a read or a write operation to the PIC16CXXX, interrupt flag, PSPIF (PIR1<7>), will be set and the processor interrupted if bit PSPIE (PIE1<7>) is set and interrupts are enabled (enable bits GIE and PEIE, (INTCON<7:6>) set). When the interrupt is serviced, bit PSPIF must be cleared by software.

The read-only status flag bit IBF, Input Buffer Full (TRISE<7>), is set if a received word is waiting to be read. Bit IBF is cleared upon read of the input buffer latch. If another word is received prior to the first being read, status flag bit IBOV (TRISE<5>) is set. Bit IBOV can be cleared by software.

The Output Buffer Full status bit, OBF (TRISE<6>), is set if a word written to PORTD latch is waiting to be read by the external bus.

When not in Parallel Slave Port mode the IBF and OBF bits are cleared. If flag bit IBOV was previously set, however, it must be cleared by software.

The following registers are for a PIC16C74 and not all peripherals are available on the PIC16C64.

<table>
<thead>
<tr>
<th>Register Name</th>
<th>Function</th>
<th>Address</th>
<th>Power-on Reset Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PORTD</td>
<td>Parallel slave port Read/Write Data</td>
<td>08h</td>
<td>xxxxx xxxxx</td>
</tr>
<tr>
<td>TRISD</td>
<td>PORTD data direction register</td>
<td>88h</td>
<td>1111 1111</td>
</tr>
<tr>
<td>PORTE</td>
<td>Read/Write/Chip Select signals</td>
<td>09h</td>
<td>------ -xxx</td>
</tr>
<tr>
<td>TRISE</td>
<td>Control bits for PORTD slave port</td>
<td>89h</td>
<td>0000 -111</td>
</tr>
<tr>
<td>INTCON</td>
<td>peripheral and global interrupt enable bits</td>
<td>08h</td>
<td>0000 -000</td>
</tr>
<tr>
<td>PIR1</td>
<td>Interrupt register (PSPIF bit)</td>
<td>0Ch</td>
<td>0000 000x</td>
</tr>
<tr>
<td>PIE1</td>
<td>Interrupt Enable register (PSPIE bit)</td>
<td>8Ch</td>
<td>0000 0000</td>
</tr>
</tbody>
</table>

TABLE 1: SUMMARY OF PARALLEL SLAVE PORT REGISTERS
### TABLE 2: PORTE FUNCTIONS

<table>
<thead>
<tr>
<th>Name</th>
<th>Bit#</th>
<th>Buffer Type</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>RE0/ RD/ AN5</td>
<td>bit0</td>
<td>ST/TTL (^{(1)})</td>
<td>Input/output port pin or read control input in parallel slave port mode or analog input:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(RD)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 = Not a read operation</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 = Read operation. Reads PORTD register (if chip selected)</td>
</tr>
<tr>
<td>RE1/ WR/ AN6</td>
<td>bit1</td>
<td>ST/TTL (^{(1)})</td>
<td>Input/output port pin or write control input in parallel slave port mode or analog input:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(WR)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 = Not a write operation</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 = Write operation. Writes PORTD register (if chip selected)</td>
</tr>
<tr>
<td>RE2/ CS/ AN7</td>
<td>bit2</td>
<td>ST/TTL (^{(1)})</td>
<td>Input/output port pin or chip select control input in parallel slave port mode or analog input:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(CS)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 = Device is not selected</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 = Device is selected</td>
</tr>
</tbody>
</table>

Legend: ST = Schmitt Trigger input TTL = TTL input

Note 1: Input buffers are Schmitt Triggers when in I/O mode and TTL buffers when in Parallel Slave Port Mode.

### FIGURE 1: TRISE REGISTER

<table>
<thead>
<tr>
<th>R-0</th>
<th>R-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>U-0</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBF</td>
<td>OBF</td>
<td>IBOV</td>
<td>PSPMODE</td>
<td>—</td>
<td>Bit2</td>
<td>Bit1</td>
<td>Bit0</td>
</tr>
</tbody>
</table>

Legend: 
- \(R\) = Readable bit
- \(W\) = Writable bit
- \(U\) = Unimplemented bit, read as ‘0’
- \(-n\) = Value at POR reset

- **bit 7**: IBF: Input Buffer Full Status bit
  - 1 = A word has been received and waiting to be read by the CPU
  - 0 = No word has been received

- **bit 6**: OBF: Output Buffer Full Status bit
  - 1 = The output buffer still holds a previously written word
  - 0 = The output buffer has been read

- **bit 5**: IBOV: Input Buffer Overflow Detect bit (in microprocessor mode)
  - 1 = A write occurred when a previously input word has not been read (must be cleared in software)
  - 0 = No overflow occurred

- **bit 4**: PSPMODE: Parallel Slave Port Mode Select bit
  - 1 = Parallel slave port mode
  - 0 = General purpose I/O mode

- **bit 3**: Unimplemented: Read as ‘0’

- **bit 2**: Bit2: Direction control bit for pin RE2/ CS/ AN7
  - 1 = Input
  - 0 = Output

- **bit 1**: Bit1: Direction control bit for pin RE1/ WR/ AN6
  - 1 = Input
  - 0 = Output

- **bit 0**: Bit0: Direction control bit for pin RE0/ RD/ AN5
  - 1 = Input
  - 0 = Output
### FIGURE 2: PIE1 REGISTER

<table>
<thead>
<tr>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSPIE</td>
<td>ADIE</td>
<td>RCIE</td>
<td>TXIE</td>
<td>SSPIE</td>
<td>CCP1IE</td>
<td>TMR2IE</td>
<td>TMR1IE</td>
<td></td>
</tr>
</tbody>
</table>

bit 7: **PSPIE**: Parallel Slave Port Read/Write Interrupt Enable bit  
1 = Enables the PSP read/write interrupt  
0 = Disables the PSP read/write interrupt

bit 6: **ADIE**: A/D Converter Interrupt Enable bit  
1 = Enables the A/D interrupt  
0 = Disables the A/D interrupt

bit 5: **RCIE**: USART Receive Interrupt Enable bit  
1 = Enables the USART receive interrupt  
0 = Disables the USART receive interrupt

bit 4: **TXIE**: USART Transmit Interrupt Enable bit  
1 = Enables the USART transmit interrupt  
0 = Disables the USART transmit interrupt

bit 3: **SSPIE**: Synchronous Serial Port Interrupt Enable bit  
1 = Enables the SSP interrupt  
0 = Disables the SSP interrupt

bit 2: **CCP1IE**: CCP1 Interrupt Enable bit  
1 = Enables the CCP1 interrupt  
0 = Disables the CCP1 interrupt

bit 1: **TMR2IE**: TMR2 to PR2 Match Interrupt Enable bit  
1 = Enables the TMR2 to PR2 match interrupt  
0 = Disables the TMR2 to PR2 match interrupt

bit 0: **TMR1IE**: TMR1 Overflow Interrupt Enable bit  
1 = Enables the TMR1 overflow interrupt  
0 = Disables the TMR1 overflow interrupt

- **R** = Readable bit  
- **W** = Writable bit  
- **U** = Unimplemented bit, read as '0'  
- **n** = Value at POR reset
### FIGURE 3: PIR1 REGISTER

<table>
<thead>
<tr>
<th>Bit 7: PSPIF(1)</th>
<th>Bit 6: ADIF</th>
<th>Bit 5: RCIF</th>
<th>Bit 4: TXIF</th>
<th>Bit 3: SSPIF</th>
<th>Bit 2: CCP1IF</th>
<th>Bit 1: TMR2IF</th>
<th>Bit 0: TMR1IF</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>R/W-0</strong></td>
<td><strong>R/W-0</strong></td>
<td><strong>R-0</strong></td>
<td><strong>R-0</strong></td>
<td><strong>R/W-0</strong></td>
<td><strong>R/W-0</strong></td>
<td><strong>R/W-0</strong></td>
<td><strong>R/W-0</strong></td>
</tr>
<tr>
<td>PSPIF(1)</td>
<td>ADIF</td>
<td>RCIF</td>
<td>TXIF</td>
<td>SSPIF</td>
<td>CCP1IF</td>
<td>TMR2IF</td>
<td>TMR1IF</td>
</tr>
<tr>
<td><strong>bit 7</strong></td>
<td><strong>bit 6</strong></td>
<td><strong>bit 5</strong></td>
<td><strong>bit 4</strong></td>
<td><strong>bit 3</strong></td>
<td><strong>bit 2</strong></td>
<td><strong>bit 1</strong></td>
<td><strong>bit 0</strong></td>
</tr>
</tbody>
</table>

- **bit 7:** PSPIF(1): Parallel Slave Port Read/Write Interrupt Flag bit
  - 1 = A read or a write operation has taken place (must be cleared in software)
  - 0 = No read or write has occurred

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

- **bit 5:** RCIF: USART Receive Interrupt Flag bit
  - 1 = The USART receive buffer is full (cleared by reading RCREG)
  - 0 = The USART receive buffer is empty

- **bit 4:** TXIF: USART Transmit Interrupt Flag bit
  - 1 = The USART transmit buffer is empty (cleared by writing to TXREG)
  - 0 = The USART transmit buffer is full

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

- **bit 2:** CCP1IF: CCP1 Interrupt Flag bit
  - **Capture Mode**
    - 1 = A TMR1 register capture occurred (must be cleared in software)
    - 0 = No TMR1 register capture occurred
  - **Compare Mode**
    - 1 = A TMR1 register compare match occurred (must be cleared in software)
    - 0 = No TMR1 register compare match occurred
  - **PWM Mode**
    - Unused in this mode

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

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

**Note 1:** PIC16C73/73A/76 devices do not have a Parallel Slave Port implemented, this bit location is reserved on these devices, always maintain this bit clear.

---

Interrupt flag bits get set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>). User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.
**TABLE 3: INTCON REGISTER**

<table>
<thead>
<tr>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-x</th>
</tr>
</thead>
<tbody>
<tr>
<td>GIE</td>
<td>PEIE</td>
<td>T0IE</td>
<td>INTE</td>
<td>RBIE</td>
<td>T0IF</td>
<td>INTF</td>
<td>RBIF</td>
</tr>
</tbody>
</table>

- **R = Readable bit**
- **W = Writable bit**
- **U = Unimplemented bit, read as '0'**
- **n = Value at POR reset**

**bit 7:** **GIE:** Global Interrupt Enable bit
1 = Enables all un-masked interrupts
0 = Disables all interrupts

**bit 6:** **PEIE:** Peripheral Interrupt Enable bit
1 = Enables all un-masked peripheral interrupts
0 = Disables all peripheral interrupts

**bit 5:** **T0IE:** TMR0 Overflow Interrupt Enable bit
1 = Enables the TMR0 interrupt
0 = Disables the TMR0 interrupt

**bit 4:** **INTE:** RB0/INT External Interrupt Enable bit
1 = Enables the RB0/INT external interrupt
0 = Disables the RB0/INT external interrupt

**bit 3:** **RBIE:** RB Port Change Interrupt Enable bit
1 = Enables the RB port change interrupt
0 = Disables the RB port change interrupt

**bit 2:** **T0IF:** TMR0 Overflow Interrupt Flag bit
1 = TMR0 register has overflowed (must be cleared in software)
0 = TMR0 register did not overflow

**bit 1:** **INTF:** RB0/INT External Interrupt Flag bit
1 = The RB0/INT external interrupt occurred (must be cleared in software)
0 = The RB0/INT external interrupt did not occur

**bit 0:** **RBIF:** RB Port Change Interrupt Flag bit
1 = At least one of the RB7:RB4 pins changed state (must be cleared in software)
0 = None of the RB7:RB4 pins have changed state
APPENDIX A: PIC16C64/74 Parallel Slave Port

MPASM 01.40 Released          PSP64.ASM  1-16-1997  17:03:44    PAGE 1

LOC OBJECT CODE     LINE SOURCE TEXT
VALUE

00001 ;*********************************************************************
00002 ;* 16C64/74 Parallel Slave Port
00003 ;* This program demonstrates the Parallel Slave Port function of
00004 ;* the PIC16C64/74. The program is interrupt driven, when the PIC
00005 ;* is either read from or written to, an interrupt is generated. If
00006 ;* the interrupt was caused by a read, a register is incremented, and
00007 ;* the new count is placed in an output queue. If the interrupt was
00008 ;* caused by a write, the data is put on the Port B pins
00009 ;
00010 ; Program:          PSP64.ASM
00011 ; Revision Date:   1-15-97      Compatibility with MPASWIN 1.40
00012 ;
00013 ;
00014 ;;********************************************************************
00015         list p=16c64
00016         ERRORLEVEL -302
00017 ;
00018         include "p16c64.inc"
00019 LIST
00020 ; P16C64.INC Standard Header File, Ver. 1.01 Microchip Technology, Inc.
00021 LIST
00022;Register definitions
00023          00024 FLAGREG equ     20h             ;Flag bit register
00024          00025 OUTDATA equ     21h             ;Output data
00025          00026 INDATA  equ     22h             ;Input data
00026          00027 COUNT   equ     23h             ;Count of times output register read
00027          00028
go
00028;Bit definitions for flag register
00029          00030 err     equ     00h             ;Error flag bit
00030          00031 OUTRDY  equ     01h             ;Output data ready flag
00031          00032 INFULL  equ     02h             ;Input data received flag
00032          00033
go
00033;Reset Vector
00034          00035 org     0000h        ;Reset Vector
00035          00036 goto    Start
00036          00037
go
00037;Interrupt Vector
00038          00039 org     0005h        ;Interrupt Vector
00039          00040 goto    Service_Int
00040          00041
go
00041_Start
00042          00043 clrf    OUTDATA         ;Clear data registers
00043          00044 clrf    INDATA
00044          00045 bsf     STATUS,RP0     ;Select register Bank1
00045          00046 movlw   b'00010111'    ;Set RD, WR, and CS as
00046          00047 movwf    TRISE          ;inputs, Enable Parallel Slave port
00047          00048 movlw   0FFh
00048          00049 movwf    TRISB
00049          00050 movlw   b'10000000'    ;Set Port_B to all outputs
00050          00051 movlw   b'10000000'    ;
00051          00052 movwf    PIE1           ;Enable Parallel Slave Port interrupt
00052          00053 movwf    STATUS,RP0  ;Select register Bank0
00053          00054 movlw   b'11000000'    ;
00054          00055 movwf    PORTD
00055          00056 movwf    OUTDATA,W    ;Set output Data in PORTD
00056          00057 movwf    PORTD
00057          00058 movlw   b'11000000'    ;Set GIE, PEIE (enable interrupts)
0013 008B 00052 movwf INTCON
00053
0014 00054 Loop
0014 1920 00055 btfsc FLAGREG,INFULL ;Check if input data received
0015 2819 00056 goto Checkout ;No data ready, check output
0016 1120 00057 bcf FLAGREG,INFULL ;Clear input data ready flag
0017 0822 00058 movf INDATA,W ;Get Input data
0018 0086 00059 movwf PORTB ;Output input data to Port_B
0019 00060 Checkout
0019 18A0 00061 btfsc FLAGREG,OUTRDY ;Check if data output already
001A 0062 goto Loop ;Not output yet, loop
001B 0063 incf COUNT,F ;Increment output data
001C 0086 00064 movwf PORTB ;Output input data to Port_B
001D 0065 bcf FLAGREG,OUTRDY ;Set flag for interrupt routine
001E 0067 goto Loop
001F 0068
001F 0069 ;*********************************************************************
0020 00089 Service_Int
0020 1F8C 00090 btfss PIR1,PSPIF ;Test for Peripheral interrupt
0021 2832 00091 goto Intout ;Not a Peripheral interrupt, exit
0022 138C 00092 bcf PIR1,PSPIF ;Clear Peripheral interrupt
0023 1683 00093 bcf STATUS,RP0 ;Select Bank1
0024 1F89 00094 btfss TRISE,IFB ;Check if input data ready
0025 283A 00095 goto Notinput ;No input, check output
0026 1283 00096 bcf STATUS,RP0 ;Input ready, select Bank0
0027 1520 00097 bcf FLAGREG,INFULL ;Set flag for main routine
0028 0808 00098 movf PORTD,W ;Get input data
0029 00A2 00099 movwf INDATA ;Put byte in input queue
002A 00100 Notinput
002A 1B09 00101 btfss TRISE,OBF ;Check if output data read
002B 2832 00102 goto Intout ;Not read, exit
002C 1283 00103 bcf STATUS,RP0 ;Select Bank0
002D 1CA0 00104 btfss FLAGREG,OUTRDY ;Check if data in output queue
002E 2832 00105 goto Intout ;Output not read, exit
002F 0821 00106 movf OUTDATA,W ;Get data from queue
0030 0888 00107 movf PORTD,F ;Put data in output buffer
0031 10A0 00108 bcf FLAGREG,OUTRDY ;Clear flag for main routine
0032 00109 Intout
0032 1683 00110 bcf STATUS,RP0 ;Select Bank1
0033 1A89 00111 btfss TRISE,IBOV ;Check input buffer overflow flag
0034 2837 00112 bcf STATUS,RP0 ;Select Bank0
0035 1283 00113 bcf STATUS,RP0 ;Select Bank0
0036 0009 00114 retfie ;Re-enable GIE and return
0037 00115 Interror
0037 1283 00116 bcf STATUS,RP0 ;Select Bank0
0038 1420 00117 bcf FLAGREG,ERR ;Set error flag for main routine

© 1997 Microchip Technology Inc.  AN579

DS00579B-page 7
0039 0009 00118 retfie ;Re-enable GIE and return
00119 00120 end

MEMORY USAGE MAP ('X' = Used, '-' = Unused)

0000 : X----XXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXX------

All other memory blocks unused.

Program Memory Words Used: 54
Program Memory Words Free: 1994

Errors : 0
Warnings : 0 reported, 0 suppressed
Messages : 0 reported, 6 suppressed
Information contained in this publication regarding device applications and the like is intended for suggestion only and may be superseded by updates. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip’s products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights. The Microchip logo and name are registered trademarks of Microchip Technology Inc. in the U.S.A and other countries. All rights reserved. All other trademarks mentioned herein are the property of their respective companies.