PIC Microcontroller and Embedded Systems
Muhammad Ali Mazidi, Rolin McKinlay and Danny Causey

Eng. Husam Alzaq
The Islamic Uni. Of Gaza
Chapter 4: PIC I/O Port Programming

- I/O Port Programming in PIC18
- I/O Bit Manipulation Programming
Objective

- List all the ports of the PIC18
- Describe the dual role of PIC18 pins
- Code Assembly to use ports for input or output
- Code PIC instructions for I/O handling
- Code I/O bit-manipulation Programs for PIC
- Explain the bit addressability of PIC ports
I/O Port Programming in PIC18

- PIC18 has many ports
  - Depending on the family member
  - Depending on the number of pins on the chip
  - Each port can be configured as input or output.
    - Bidirectional port
  - Each port has some other functions
    - Such as timer, ADC, interrupts and serial communication
  - Some ports have 8 bits, while others have not
Figure 4-1. PICF458 Pin Diagram

The PIC uC's
<table>
<thead>
<tr>
<th>Pins</th>
<th>Add 18-pin</th>
<th>28-pin</th>
<th>40-pin</th>
<th>64-pin</th>
<th>80-pin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chip</td>
<td>PIC18F1220</td>
<td>PIC18F2220</td>
<td>PIC18F458</td>
<td>PIC18F6525</td>
<td>PIC18F8525</td>
</tr>
<tr>
<td>PORT A</td>
<td>F80H</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>PORT B</td>
<td>F81H</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>PORT C</td>
<td>F82H</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>PORT D</td>
<td>F83H</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>PORT E</td>
<td>F84H</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>PORT F</td>
<td>F85H</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>PORT G</td>
<td>F86H</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>PORT H</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PORT J</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PORT K</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PORT L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
I/O SFR

- Each port has three registers for its operation:
  - TRIS register (Data Direction register)
    - If the corresponding bit is 0 → Output
    - If the corresponding bit is 1 → Input
  - PORT register (reads the levels on the pins of the device)
  - LAT register (output latch)
- The DataLatch (LAT) register is useful for read-modify-write operations on the value that the I/O pins are driving.
I/O SFR

- PIC18F458 has 5 ports
- Upon reset, all ports are configured as input
- TRISx register has OFFH

<table>
<thead>
<tr>
<th>Pins</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>PORT A</td>
<td>F80H</td>
</tr>
<tr>
<td>PORT B</td>
<td>F81H</td>
</tr>
<tr>
<td>PORT C</td>
<td>F82H</td>
</tr>
<tr>
<td>PORT D</td>
<td>F83H</td>
</tr>
<tr>
<td>PORT E</td>
<td>F84H</td>
</tr>
<tr>
<td>LATA</td>
<td>F89H</td>
</tr>
<tr>
<td>LATB</td>
<td>F8AH</td>
</tr>
<tr>
<td>LATC</td>
<td>F8BH</td>
</tr>
<tr>
<td>LATD</td>
<td>F8CH</td>
</tr>
<tr>
<td>LATE</td>
<td>F8DH</td>
</tr>
<tr>
<td>TRISA</td>
<td>F92H</td>
</tr>
<tr>
<td>TRISB</td>
<td>F93H</td>
</tr>
<tr>
<td>TRISC</td>
<td>F94H</td>
</tr>
<tr>
<td>TRISD</td>
<td>F95H</td>
</tr>
<tr>
<td>TRISE</td>
<td>F96H</td>
</tr>
</tbody>
</table>
Figure 4-2. CMOS States for P and N Transistors

The PIC uCs
Figure 4-3. Outputting (Writing) 0 to a Pin in the PIC18
Figure 4-4. Outputting (Writing) 1 to a Pin in the PIC18
Figure 4-5. Inputting (Reading) 0 from a Pin in the PIC18

The PIC uCs
Figure 4-6. Inputting (Reading) 1 from a Pin in the PIC18
Port A

- PORTA is a 7-bit wide, bidirectional port.
  - Sometimes A6 is not available. [why?]
- The corresponding Data Direction register is TRISA.
- Setting a TRISA bit (= 1) will make the corresponding PORTA pin an input
- Clearing a TRISA bit (= 0) will make the corresponding PORTA pin an output
- On a Power-on Reset, these pins are configured as inputs and read as '0'.
Example 1

```
BACK
MOVLW 0x55
MOVWF PORTA
CALL DELAY
MOVLW 0xAA
MOVWF PORTA
CALL DELAY
GOTO BACK
```

```
MOVLW B’00000000’
MOVWF TRISA
MOVLW 0x55
MOVWF PORTA
CALL DELAY
MOVLW 0xAA
MOVWF PORTA
CALL DELAY
GOTO BACK
```
Example 2

MYREG EQU 0x20
MOVLW B'11111111'
MOVWF TRISA
MOVF PORTA, w
MOVWF MYREG
PORT B, PORT C, PORT D and PORT E

- PORTB is 8 pins
- PORTC is 8 pins
- PORTD is 8 pins
- PORTE is 3 pins
Read followed by write operation

- Be careful
  - Don’t have a two I/O operations one right after the other.

- Data Dependency
  - A NOP is needed to make that data is written into WREG before it read for outputting to PortB.

```assembly
CLRF TRISB
SETF TRISC
L4 MOVF PORTC,W
MOVWF PORTB
BRA L4
```
The PIC uCs

**Figure 4-7. Pipeline for Read Followed by Write I/O**

The RAW (Read - After - Write) for two consecutive instructions.
### Two Solutions

#### Solution 1

<table>
<thead>
<tr>
<th>CLRF</th>
<th>TRISB</th>
</tr>
</thead>
<tbody>
<tr>
<td>SETF</td>
<td>TRISC</td>
</tr>
<tr>
<td>L4</td>
<td>MOVF</td>
</tr>
<tr>
<td>NOP</td>
<td>PORTC,W</td>
</tr>
<tr>
<td>MOVWF</td>
<td>PORTB</td>
</tr>
<tr>
<td>BRA L4</td>
<td></td>
</tr>
</tbody>
</table>

#### Solution 2

<table>
<thead>
<tr>
<th>CLRF</th>
<th>TRISB</th>
</tr>
</thead>
<tbody>
<tr>
<td>SETF</td>
<td>TRISC</td>
</tr>
<tr>
<td>L4</td>
<td>MOVFF</td>
</tr>
<tr>
<td>NOP</td>
<td>PORTC, PORTB</td>
</tr>
<tr>
<td>BRA L4</td>
<td></td>
</tr>
</tbody>
</table>

MOVFF is 4-byte instruction.
Example 4-1

Write a test program for the PIC18 chip to toggle all the bits of PORTB, PORTC and PORTD every 0.25 of a second. (suppose that there is a 4 MHz)

```assembly
list P= PIC18F458
#include P18F458.INC
R1 equ 0x07
R2 equ 0x08
ORG 0
  CLRF TRISB
  CLRF TRISC
  CLRF TRISD
  MOVLW 0x55
  MOVWF PORTB
  MOVWF PORTC
  MOVWF PORTD
```

The PIC uCs
Solution

L3

COMF PORTB,F

COMF PORTC,F

COMF PORTD,F

CALL QDELAY

BRA L3

QDELAY

MOVLW D'200'

MOVWF R1

D1

MOVLW D'250'

MOVWF R2

D2

NOP

NOP

DECF R2, F

BNZ D2

DECF R1, F

BNZ D1

RETURN

END

The PIC uCs
I/O Bit Manipulation Programming

- I/O ports and bit-addressability
- Monitoring a single bit
- Reading a single bit

Section 4-2
## I/O ports and bit-addressability

<table>
<thead>
<tr>
<th>PORT A</th>
<th>PORT B</th>
<th>PORT C</th>
<th>PORT D</th>
<th>PORT E</th>
<th>PORT Bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>RA0</td>
<td>RB0</td>
<td>RC0</td>
<td>RD0</td>
<td>RE0</td>
<td>D0</td>
</tr>
<tr>
<td>RA1</td>
<td>RB1</td>
<td>RC1</td>
<td>RD1</td>
<td>RE1</td>
<td>D1</td>
</tr>
<tr>
<td>RA2</td>
<td>RB2</td>
<td>RC2</td>
<td>RD2</td>
<td>RE2</td>
<td>D2</td>
</tr>
<tr>
<td>RA3</td>
<td>RB3</td>
<td>RC3</td>
<td>RD3</td>
<td></td>
<td>D3</td>
</tr>
<tr>
<td>RA4</td>
<td>RB4</td>
<td>RC4</td>
<td>RD4</td>
<td></td>
<td>D4</td>
</tr>
<tr>
<td>RA5</td>
<td>RB5</td>
<td>RC5</td>
<td>RD5</td>
<td></td>
<td>D5</td>
</tr>
<tr>
<td></td>
<td>RB6</td>
<td>RC6</td>
<td>RD6</td>
<td></td>
<td>D6</td>
</tr>
<tr>
<td></td>
<td>RB7</td>
<td>RC7</td>
<td>RD7</td>
<td></td>
<td>D7</td>
</tr>
</tbody>
</table>
# Bit Oriented Instruction for PIC18

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>BSF</td>
<td>fileReg, bit Bit Set File Register</td>
</tr>
<tr>
<td>BCF</td>
<td>fileReg, bit Bit Clear File Register</td>
</tr>
<tr>
<td>BTG</td>
<td>fileReg, bit Bit Toggl File Register</td>
</tr>
<tr>
<td>BTFSC</td>
<td>fileReg, bit Bit Test File Register, skip if</td>
</tr>
<tr>
<td>BTFSS</td>
<td>fileReg, bit Bit Test File Register, skip if</td>
</tr>
</tbody>
</table>

The PIC uCs 4-25
Example 4-2

A LED is connected to each pin of port D. Write a program to turn on each LED from pin D0 to D4.

- CLRF TRISD
- BSF PORTD,0
- CALL DELAY
- BSF PORTD,1
- CALL DELAY
- BSF PORTD,2
- CALL DELAY
- BSF PORTD,3
- CALL DELAY
- BSF PORTD,4
- CALL DELAY
Example 4-3

- Write the following programs
  A. Create a square wave of 50% duty cycle on bit 0 of C

Solution 1

```c
BCF TRISC,0
HERE
BSF PORTC,0
CALL DELAY
BCF PORTC,0
CALL DELAY
BRA HERE
```

How many byte are used?

The PIC uCs
Example 4-3

Write the following programs

A. Create a square wave of 50% duty cycle on bit 0 of C

Solution 2

```
BCF TRISC,0
BACK
BTF PORTC,0
CALL DELAY
BRA BACK
```

How many byte are used?
Example 4-4

Write a program to perform the following:

a) Keep monitoring the RB2 bit until it becomes HIGH (1)

b) When RB2 becomes HIGH, write value 45H to portC and send a HIGH to LOW plus to RD3

```
BSF TRISB,2
CLRF TRISC
BCF PORTD,3
MOVLW 0x45
AGAIN
BTFSS PORTB,2
BRA AGAIN
MOVWF PORTC
BSF PORTD,3
CALL DELAY
BCF PORTD,3
```
Example 4-5

- Bit RB3 is an input and represents the condition of a door alarm.
- Whenever it goes LOW, send a HIGH-to-LOW pulse to RC5 to turn on a buzzer.
Solution

BSF TRISB,3
BCF TRISC,5
HERE
BTFSC PORTB,3
BRA HERE
BSF PORTC,5
BCF PORTC,5
CALL DELAY
BRA HERE
Reading a single bit
Example 4-8

- A switch is connected to pin RB0 and a LED to pin RB7. Write a program to read the status of SW and send it to the LED.

```
BSF TRISB,0
BCF TRISB,7
AGAIN
BTFSS PORTB,0
GOTO OVER
BSF PORTB,7
GOTO AGAIN
OVER
BCF PORTB,7
GOTO AGAIN
```
Reading input pins VS. LATx port

- There are two possibilities to read port's value
  - Through reading the status of the input pin
  - Through reading the internal latch of the LAT register.
    - Some instructions do that
    - The action is
      1. The instruction read the latch instead of the pin
      2. Execute the instruction
      3. Write back the result to the Latch
      4. The data on the pins are changed only if the TRISx bits are cleared.

The PIC uCs
<table>
<thead>
<tr>
<th>Instruction</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDWF</td>
<td>fileReg,d Add WREG from f</td>
</tr>
<tr>
<td>BSF</td>
<td>fileReg,bit Bit Set fileReg</td>
</tr>
<tr>
<td>BCF</td>
<td>fileReg,bit Bit Clear fileReg</td>
</tr>
<tr>
<td>COMF</td>
<td>fileReg,d Complement f</td>
</tr>
<tr>
<td>INCF</td>
<td>fileReg,d Increment F</td>
</tr>
<tr>
<td>SUBWF</td>
<td>fileReg,d Subtract WREG from f</td>
</tr>
<tr>
<td>XORWF</td>
<td>fileReg,d Exclusive-OR WREG with f</td>
</tr>
</tbody>
</table>
Figure 4-8. LATx Register Role in Reading a Port or Latch
Chapter 4: Summary

- We focused on the I/O Ports of the PIC.
- These ports used for input or output.
- We discussed Bit manipulation instructions.

Next: Arithmetic, logic Instruction and programs