ENGR 270 Lab #2 Online – Input/output and Basic Operations

Objectives

Apply input/output and basic operations of a processor to implement a 4-bit adder.

Preparation

Complete the following steps before starting to work on the experiments in this lab:

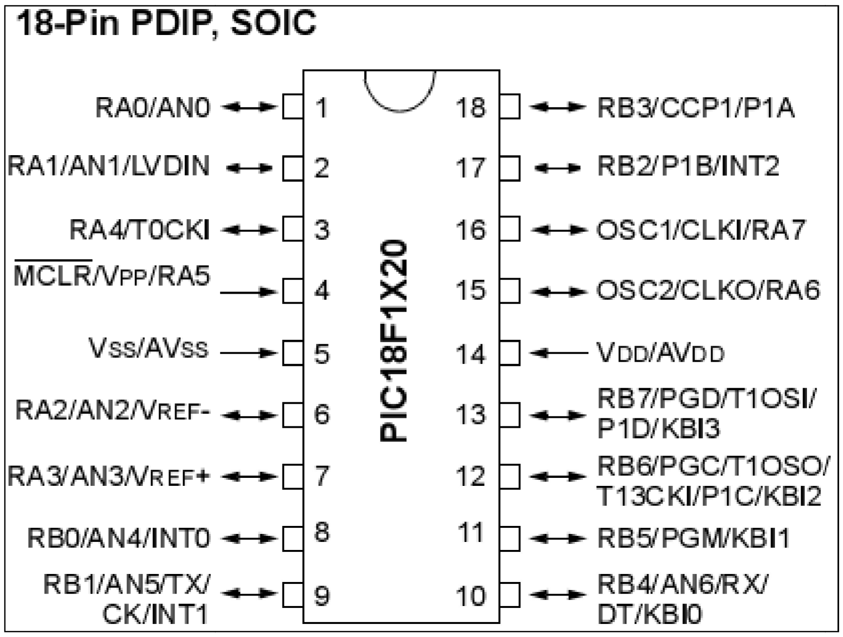
1) Complete Lab1 and associated report.
2) Complete lecture, homework and videos in Chapter 2 “architecture and Instruction Type” and Chapter 3 “Input/Output Organization”.

In this lab and all future labs, you are expected to write assembly programs (PICmicro Assembly) that accept input, process the data and output the results.

Input and Output operations are fully described in Chapter 3 of Computer Organization and Microprocessors textbook. Even more information is available in the PICmicro Data Sheet.

Specifically, you need to be familiar with the following concepts:

Pin out and Packaging for PICmicro (PIC18f1220)



Each pin can be configured to perform a variety of functions, for example Pin 8 may be an I/O port (RB0), I/O port (AN4), external Interrupt 0 (INT0). This type of multi-use is common in microcontroller with high level of functionality but it is less common in general purpose microprocessors.

The two pins whose definition is constant are pins 5 and 14 which are power and ground:

Pin 5	Ground (0 V)
Pin 14	Power (5 V)

❖ External Pin Set up as general purpose I/O Ports
External I/O ports are multiplexed with an alternate function from other modules on the PICmicro. In general, when a peripheral is enabled, that pin may not be used as a general purpose I/O pin. Each port has three registers for its operation. These registers are:

- TRIS register (data direction register)
  TRISA address is 0xF92
  TRISB address is 0xF93

- PORT register (reads the levels on the pins of the device)
  PORTA address is 0xF80
  PORTB address is 0xF81

- ADCON1 register (output latch)
  ADCON1 address is 0xFC1

➢ PORTA, TRISA and LATA Registers
PORTA is an 8-bit wide, bidirectional port. Reading the PORTA register reads the status of the pins, whereas writing to it will write to the port latch.

	RA7	**RA6**	**RA5**	**RA4**	**RA3**	**RA2**	**RA1**	**RA0**
PORTA Register	b7	b6	b5	b4	b3	b2	b1	b0
I/O Pins:	p1	p1	p4	p3	p7	p6	p2	p1
Alternative Uses:	"Each I/O pin may be configured for multiple uses, refer to pin definitions earlier in the chapter for a list of Alternative uses for each pin"							

The corresponding data direction register is TRISA. Setting a TRISA bit (= 1) will make the corresponding PORTA pin an input. When the pin is set to input it will be in a high-impedance mode. Clearing a TRISA bit (= 0) will make the corresponding PORTA pin an output. In this mode the contents of the output latch will be available on the selected external I/O pin.

Any instruction that specifies a file register as part of the instruction performs a Read-Modify-Write (R-M-W) operation. The register is read, the data is modified and the result is stored according to either the instruction or the destination designator 'd'. A read operation is performed on a register even if the instruction writes to that register. It is important to consider the impact of a read on the configuration before using read-modify-write instruction.

- Example of initializing PortA

```
CLRF    PORTA       ; Initialize PORTA by clearing output data latches
MOVLW   0x7F
MOVWF   ADCON1      ; Configure A/D for digital I/O
MOVLW   0xF0        ; Value used to initialize data direction
MOVWF   TRISA       ; Set RA<3:0> as outputs RA<7:4> as inputs
```

➢ PortB, TRISB and LATB Registers
PORTB is an 8-bit wide, bidirectional port. Reading the PORTB register reads the status of the pins, whereas writing to it will write to the port latch.

	RB7	**RB6**	**RB5**	**RB4**	**RB3**	**RB2**	**RB1**	**RB0**
Port B Register	b7	b6	b5	b4	b3	b2	b1	b0
I/O Pins:	p1	p1	p1	p1	p1	p1	p9	p8
Alternating Uses:	"Each I/O pin may be configured for multiple uses, refer to pin definitions earlier in the chapter for a list of Alternative uses for each pin"							

The corresponding data direction register is TRISB. Setting a TRISB bit (= 1) will make the corresponding PORTB pin an input. When the pin is set to input it will be in a high-impedance mode. Clearing a TRISB bit (= 0) will make the corresponding PORTB pin an output. In this mode the contents of the output latch on the selected pin.

Any instruction that specifies a file register as part of the instruction performs a Read-Modify-Write (R-M-W) operation. The register is read, the data is modified and the result is stored according to either the instruction or the destination designator 'd'. A read operation is performed on a register even if the instruction writes to that register. It is important to consider the impact of a read on the configuration before using read-modify-write instruction.

- Example of initializing PortB

```
CLRF    PORTB       ; Initialize PORTB by clearing output data latches
MOVLW   0x7F
MOVWF   ADCON1       ; Configure A/D for digital I/O
MOVLW   0xCF        ; Value used to initialize data direction
MOVWF   TRISB       ; Set RB<3:0> as inputs, RB<5:4> as outputs and; RB<6:7> as
input
```

❖ RB5, Pin 11 (PortB, Bit 5)
Pin 11 is also used for Low Voltage Programming (LVP) and must be explicitly disabled in order to

make pin 11 a digital I/O pin. LVP can be disabled at the start of the program by writing 0 to LVP, bit 2 of CONFIG4L register (located at address 0x300006). This can be done either by using table write or from MPLAB menu Configure > Configure Bits.

- Watch Dog Timer (WDT)
  Watch Dog Timer resets the processor if it is not cleared once every 400 msec. You can clear the WDT by using the instruction "CLRWDT". Another option is to disable the WDT at the start of the program by writing 0 to WDT, bit 0 of CONFIG2H register (located at address 0x300003). This can be done either by using table write or from MPLAB menu Configure > Configure Bits.

Experiment 1. 4-bit Adder

Write a PICmicro Assembly code that accept 4-bit operand 1 from the least significant bits of RB<3:0> and 4-bit operand 2 from the most significant bits of RB <7:4>. Next the opernad 1 and 2 are added and placed in PORTA or RA<4:0>.

The program shall wait until value of operands are changes and only then update the sum stored in PORTA or RA<4:0>.

Report Requirements

This lab and associated report must be completed individually. All reports must be computer printed (formulas and diagrams may be hand drawn) and at minimum:

For each experiment include:

- Clear problem statement in your words.
- Answer to any specific experiment questions (if any)
- Pseudo code which may be written in C-like syntax
- Disassembled code available after successful assembling
- Test plan which describes the input values and expect output/memory values for a successful design.
- Simulator output which shows the stimuli, relevant memory locations values showing validation based on test plan.

For the whole report include:

- A Cover sheet with your name, class, lab and completion date.
- A Lessons Learned section which summarizes your learning from this lab.
- A New Experiment section that has description of a new experiment and the experiment's results. Experiment should be related to material covered in class but not similar to one of the experiments in this lab.