

ENGR 270 LAB #1 - Introduction to Microprocessors

Objective

Gain an understanding of the role of assembly language and introduction to the PIC assembler, linker, simulator and use of microprocessor in system design.

Related Principles

- ❖ Computer Organization and Design
- ❖ Microprocessors
- ❖ Hardware and Software Interface
- ❖ Digital Design
- ❖ Assembly language

Equipment

- ❖ Windows-based PC with MPLAB Simulation Solutions Software
- ❖ USB hard disk or other removable drives
- ❖ Microchip PICkit programmer

Supplies

- ❖ PC Board
- ❖ PICmicro (18F1220)
- ❖ Four LEDs
- ❖ One 8-DIP Switch
- ❖ Three 1K resistors

Preparation/Background

- 1) Installation.
If you have not already installed MPLAB X, visit <http://www.microchip.com/pagehandler/en-us/family/mplabx/> for MPLAB X download and installation instruction. MPLAB X is Java-based, and should work well under Windows, Linux or OS-X. However, if you have a 64-bit machine, you may need to enable 32-bit Java for MPLAB-X to function properly.
- 2) MPLAB start-up and interface.
It would be helpful to refer to MPLAB Help section for a detailed description of MPLAB interface and functionality.

Experiment #1

In this experiment, student is expected to practice creating a new project, compile it, simulate it, and program a PICmicro chip and test it in a circuit.

❖ Create a new project for a simple counter.

- 1) In MPLAB-X, select File/New Project... from the menu. This will start the MPLAB-X Project Wizard which will walk you through creating a project
 - a) In Step 1 of the Project Wizard, just click "Next>" (the defaults: "Microchip Embedded"/"Standalone Project" should be fine);
 - b) Next, under "Families:" select "Advanced 8-bit MCUs (PIC18)"; then under "Device" select "PIC18F1220." Then click "Next>"
 - c) Under "Select Tool" choose "Simulator " then click "Next>"
 - (i) The wizard then skips to Step 6:
 - d) Under "Select Compiler" choose the item under "mpasm" (currently this says "mpasm (v5.68) [C:/Program Files (X86)...]" but this may be different on your machine);
 - e) On the final window, under "Select Project Name and Folder," first enter a Project Location. This is the parent of the directory where your project will be stored. Then enter a Project Name (we'll use "Lab1" in this example; avoid spaces and special characters). Under "Project Folder" you should see a new folder named "projectname.X" (where projectname is whatever you entered). **Make sure you use a separate project name and folder for each program you write.** One of the most frequent issues people run into is editing a new program Y but accidentally including an older file X. You can minimize the risk of this by always starting a new project in a new folder for each program you write (even if they're part of the same assignment). Also, **it's recommended that you keep a copy** of your code outside of MPLAB-X. Make frequent backups, and make sure you keep copies of your code on a flash drive etc. Remember that the lab computers remove user files when you log out. Click "Finish" when done.
- 2) Once you have finished the Project Wizard, you should see a window with different sub-windows inside it. In the upper-left area of the window should be a window with the name of your project ("Lab1"), followed by "Header Files," "Important Files," and other folder names. Note that if you were previously working on a different project, you may see that project listed here as well. Find the name of the project you're creating now, and *right-click* on the project name, then select "New/AssemblyFile.asm"
- 3) Enter a "File Name" of your choice. main.asm is always a good choice, but you can choose something more descriptive if you like. Don't enter anything in the Folder: area. You should see a full path under "Created File:" which should end with "\Lab1.X\main.asm" Hit "Finish"

Note: On the upper-middle/right side of the window should be a tab labeled "main.asm" This is where you'll enter your program's source code.

- 4) Copy the code from next page into your newly created main.asm file. Save your code by selecting File/Save" or by just hitting CTRL-S. Note that some of the indenting may not match what's shown below. You can indent entire sections of text in MPLAB-X by selecting the text and hitting TAB (you can un-indent by hitting SHIFT-TAB).
- 5) Select "Run/Build Project" to assemble the code
Notes:
 - i) *Tab formatting should be maintained.*
 - ii) *Select "Absolute" if prompted with "Absolute or Relocatable" code dialog box.*
 - iii) The lower-right window should have a tab labeled "Output - Lab1(Build, Load)" and the bottom of that window should include the messages "BUILD SUCCESSFUL" and "Loading Complete." Any **messages in red** indicate a likely error in your source code.

```

;-----
; FILE: Intro.asm
; DESC: Experiment 1 – Simple Counter
; DATE: 5-18-06
; AUTH: Class
; DEVICE: PICmicro (PIC18F1220)
;-----

list           p=18F1220           ; Set processor type
radix          hex                 ; Sets the default radix for data exp.
config         WDT=OFF, LVP=OFF, OSC = INTIO2 ; Disable Watchdog timer, Low V. Prog, and RA6 as a clock

#define        PORTA                0xF80
#define        PORTB                0xF81
#define        TRISA                0xF92
#define        TRISB                0xF93
#define        ADCON1              0xFC1

COUNT        equ                 0x080
LASTIN        equ                 0x081
INPUT         equ                 0x082
TEMP          equ                 0x083

org           0x000                ; Set the program origin (start) to absolute 0x000

; Initialize all I/O ports
CLRF          PORTA                ; Initialize PORTA
CLRF          PORTB                ; Initialize PORTB
MOVLW        0x7F                  ; Set all A/D Converter Pins as
MOVWF        ADCON1              ; digital I/O pins
MOVLW        0x00                  ; Value used to initialize data direction
MOVWF        TRISB                ; Set Port B RB<7:0> as outputs
MOVLW        0x01                  ; Value used to initialize data direction
MOVWF        TRISA                ; Set Port A Pin 0 RA<0> as input

MOVLW        0x00                  ; W = 0
MOVWF        COUNT                ; COUNT = WREG
MOVWF        LASTIN               ; LASTIN = WREG

Loop:
MOVFF        PORTA, INPUT          ; INPUT = PORTA
MOVF         INPUT, 0              ; W = PORTA
XORWF        LASTIN, 0            ; W = W XOR LASTIN
ANDLW        0x1                  ; W = W AND 0x1
MOVFF        INPUT, LASTIN        ; LASTIN = PORTA
MOVWF        TEMP                 ; TEMP = W
BTFSF        TEMP, 0              ; If TEMP<0> = 0 Then Skip Next Command
CALL        IncL
GOTO        Loop

IncL:
MOVF         COUNT, 0              ; W = COUNT
ADDLW        1                    ; W = W + 1
MOVWF        COUNT                ; COUNT = W
MOVWF        PORTB                ; PORTB = W
RETURN

end                                ; Indicates the end of the program.

```

❖ Simulation

Use the simulation feature of MPLAB-X to test the functionality of your counter by following the steps shown:

- 1) In your source code listing, click on the line number next to the code line that says

```
"CLRF   PORTA       ; Initialize PORTA"
```

The entire line should turn red. This indicates that you've created a *breakpoint* at that line. Each time the simulator is about to execute a line with a breakpoint, the simulator will pause and return control to you.

- 2) Select "Debug/New Watch..." and under "Enter or Select Watch Expression" enter PORTA and click OK (remember that names like "PORTA" are case-sensitive: "PORTA" is different from "PortA").
- 3) Again select "Debug/New Watch..." but this time enter PORTB and click OK. Repeat for WREG (note that these 3 items (which are called "Special Function Registers") can also be selected from the drop-down box).
- 4) Select Window/Simulator/Stimulus" You'll see a new tab labeled "Stimulus." Click on the area beneath "Pin" and select RA0 (which specifies bit 0 of PORTA). Click on the box under Action and select "Toggle."
- 5) Select "Debug/Debug Project(Lab1)." You'll see a box saying "PICkit2 not Found." Select "Simulator" under "Hardware Tools" and click "OK"
- 6) Your main dialog box should now say something like:
 - Launching
 - Initializing simulator
 - User program running
 - User program stopped
 - Breakpoint hit at line 21 in file ...
- 7) Here you can enter a few more watch locations. Click on the Variables tab, *double-click* under "Name" (where it says "<Enter new watch>"), enter "0x80" and hit Tab. Repeat for 0x81, 0x82 and 0x83.

At this point you're ready to begin simulating your code. The F7 key will single-step your code, one instruction at a time. As the code executes, the current instruction will be highlighted in your source-code window. Begin hitting F7 and observe what happens. You should see the program advance line by line, until it reaches the GOTO Loop" statement, at which point it will return to a previous instruction. Your variables should all show a Value of 0x00. This pattern will repeat indefinitely, until you change the value of your simulated input. Proceed as follows:

1. In the Stimulus window, click the arrow under the "Fire" column;
2. Click on the "Variables" tab to examine your variables;
3. Continue hitting F7 and observe what happens. The PORTA line should turn red and show a value of 0x01 (this reflects the fact that you just toggled bit 0 of that port). Keep hitting F7 and observe what happens. After several steps, the program should reach the line labeled "Incl:"
4. After 3 more steps, you should see the value of address 0x80 change to 0x01. After one more step, PORTB should also change to 0x01.
5. The program will now return to looping indefinitely, until you switch to the Stimulus window and again click Fire. Then go back to Variables and hit F7 and you'll again observe PORTB incrementing after 10-20 instructions.

NOTE: You can hold down F7 instead of repeatedly pressing it, but changes in the Variables window may not appear immediately.

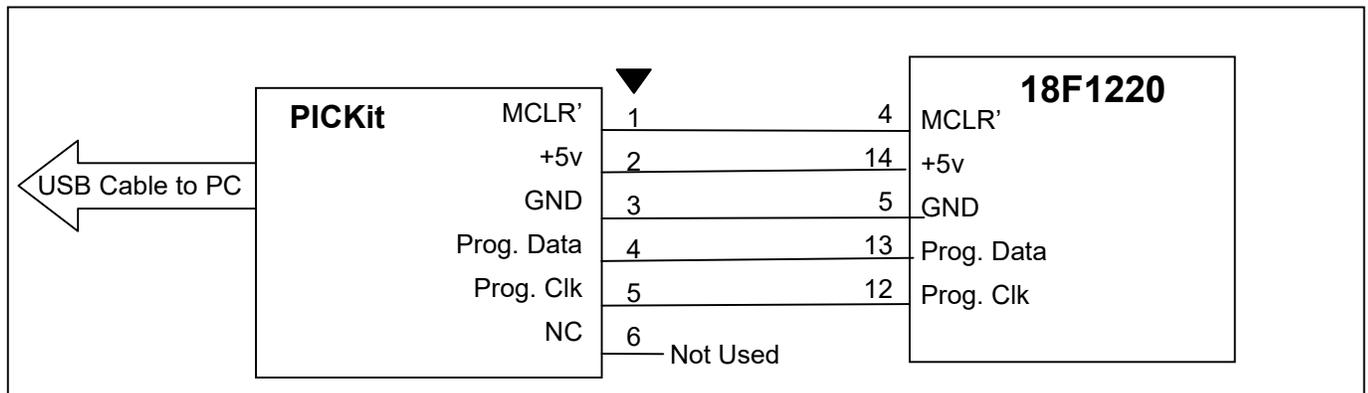
When you're finished, you can select "Debug/Finish Debugger Session."

❖ PICmicro Programming.

At this point you have a program that has been tested using a simulation, so now it is time for programming the PICmicro so that it could be used in a later design. The following steps enable you to program the PICmicro:

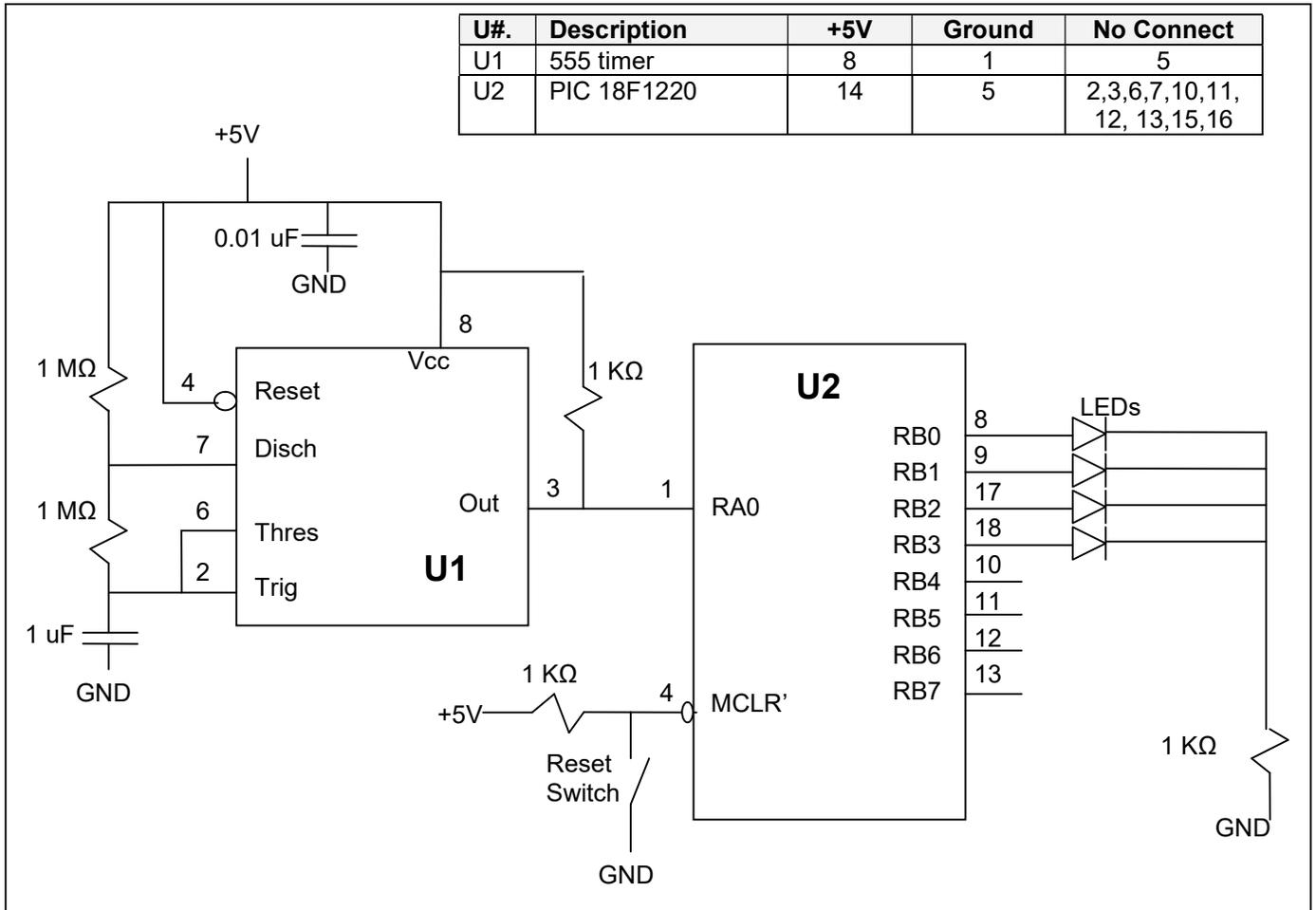
- Connect the PICKit USB to your PC using the USB cable;
- Start MPLAB-X and open your project (if it's not already opened);
- In the upper-left window, *right-click* on the project name ("Lab1") and select Properties.
- Under "Hardware Tools" select "PICKit" and click OK
- Right-click on the project name again, and select "Make and Program Device"

PICKit can be directly connected to the circuitry using the following connections in order to program the PICmicro without removing it from the circuit. PICKit is able to power your circuit so no need for power supply.



Experiment #2

The objective of this experiment is to build a circuit to test the functionality of the PICmicro programmed in the previous experiment. Build the following circuit and develop a test to ensure that your counter is working properly.



Using your implementation based on the above schematics and earlier code, confirm that it performs as a 4-bit binary up counter. This experiment requires that you demonstrate your system to the instructor upon completion. Include the approval signature in your report.

Team Members: • • • •	LAB1 Demo Instructor Approval Signature & Date:
-----------------------------------	---

Report Requirements

All reports must be computer printed (formulas and diagrams may be hand drawn) and at minimum include:

For each experiment:

- a) Clear problem statement; specify items given and to be found.
- b) Specific responses to each question asked in the experiment.
- c) Documentation of resulting high level design, disassembled code, system diagram, schematics and any other supporting material.

Note:

Disassembly listing can be found at "Window>Debugging>Output>Disassembly Listing File" menu item.

If it is grayed out (not available) then access project property, select "Loading" and check "Load symbols...". Make sure to click on the Apply button! You may also have to rebuild the project to generate the disassembled listing.

For the report as a whole

- a) Cover sheet with your name, course, lab title, date of completion and your teammates' name.
- b) Lessons learned from this lab.
- c) A new experiment and expected results which provide additional opportunity to practice the concepts in this lab.