# ECE 376 - Homework #2

Assembler, Flow Charts, Binary Inputs. Due Wednesday, September 8th, 2021
Please make the subject "ECE 376 HW#2" if submitting homework electronically to Jacob_Glower@yahoo.com (or on blackboard)

## Assembler Coding

1) Convert the following C code to assembler (8-bit operations)

```
% unsigned char A, B, C;

A      equ      0
B      equ      1
C      equ      2

%  C = 2*A + 3*B + 4;

       movf     A,W
       addwf    A,W
       addwf    B,W
       addwf    B,W
       addwf    B,W
       addlw    4
       movwf    C
```

2) Convert the following C code to assembler: (16-bit operations)

One method: 22 insructions

```
%  unsigned int A, B, C;

A      equ 0
B      equ 2
C      equ 4

% C = 2*A + 3*B + 4;

       movff    A,C
       movff    A+1,C+1

       movf     A,W
       addwf    C,F
       movf     A+1,W
       addwfc   C+1,F

       movf     B,W
       addwf    C,F
       movf     B+1,W
       addwfc   C+1,F

       movf     B,W
       addwf    C,F
       movf     B+1,W
       addwfc   C+1,F

       movf     B,W
       addwf    C,F
       movf     B+1,W
       addwfc   C+1,F

       movlw    4
       addwf    C,F
       movlw    0
       addwfc   C+1,F
```

Another solution uses the MUL command (15 instructions)

```
%  unsigned int A, B, C;

A      equ 0
B      equ 2
C      equ 4

% C = 2*A + 3*B + 4;

       movlw    4
       movwf    C
       clrf     C+1

       movf     A,W
       mullw    2
       movf     PRODL
       addwf    C,F
       movf     PRODH
       addwfc   C+1,F

       movf     B,W
       mullw    3
       movf     PRODL
       addwf    C,F
       movf     PRODH
       addwfc   C+1,F
```

3) Convert the following C code to assembler

```
% unsigned char A, B, C;

A      equ      0
B      equ      1
C      equ      2

% if( B > 10 )
%   C = A + 2;
%else
%   C = A + 5;

       movl     10
       cpfsgt   B
       goto     If
       goto     Else
If:    movf     A,W
       addlw    2
       movwf    C
       goto     End
Else:   movlw   5
       addwf    A,W
       movwf    C
End:   nop
```

4) Convert the following C code in to assembler

```
% unsigned char A, B, C;
A      equ      0
B      equu     1
C      equ      2

%while( B > 0) {
%   if(B > 10) {
%      C = A + 2;
%   else
%      C = A + 5;
%   }

While:
       movlw    0
       cpfsgt   B
       goto     End

       movlw    10
       cpfsgt   B
       goto     Else
If:
       movf     A,W
       addlw    2
       movwf    C
       goto     While
Else:
       movf     A,W
       addlw    5
       movwf    C
       goto     While
End:
       nop
```

## Flow Charts & Counters

5) The flow chart below turns your PIC into a rigged voting machine

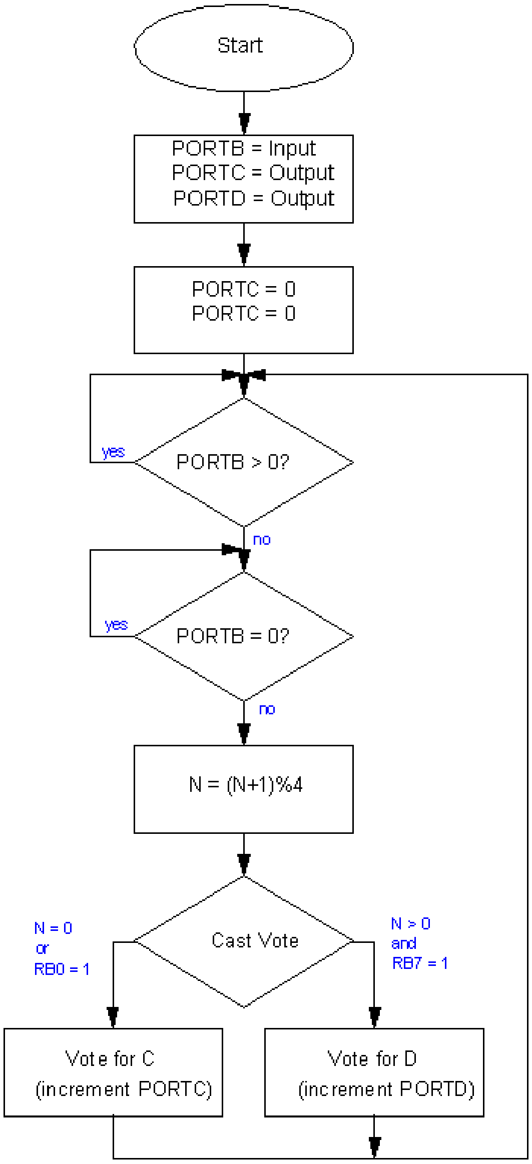
- When you press RB0, one vote is cast for Candidate C
- When you press RB7, one vote is cast for Candiadate D
- Every 4th vote always goes to Candidate C

Write the corresponding assembler code.

```
       org      0x800

       movlw    0xFF
       movwf    TRISB
       clrf     TRISC
       clrf     TRISD
       movlw    0x0F
       movwf    ADCON1

       clrf     PORTC
       clrf     PORTD
L1:
       movlw    0
       cpfsgt   PORTB
       goto     L2
       goto     L1
L2:
       movlw    0
       cpfseq   PORTB
       goto     L3
       goto     L2
L3:
       incf     N,W
       andlw    0x03
       movwf    N
L4:
       movlw    0
       cpfseq   N
       goto     Test2
       goto     VoteC
       btfsc    PORTB,0
       goto     VoteC
VoteD
       incf     PORTD,F
       goto     L1
VoteC
       incf     PORTC,F
       goto     L1
```



Problem #5

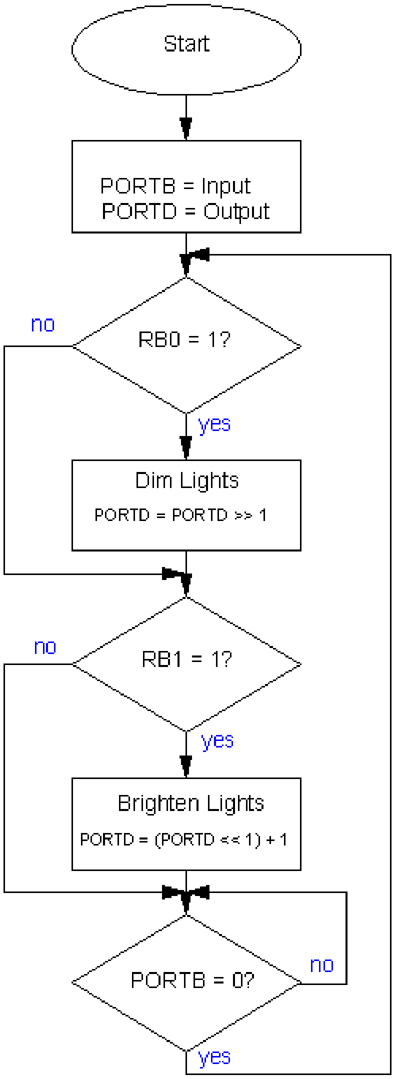
6) The flow chart below turns your PIC into an electronic flashlight

- RB0: Make the light dimmer
- RB1: Make the light brighter

Write the corresponding assembler code

```
        org     0x800

        movlw   0xFF
        movwf   TRISB
        clrf    TRISD
        movlw   0x0F
        movwf   ADCON1
L1:
        btfss   PORTB,0
        goto    L2
        rrncf   PORTD,W
        andlw   0x7F
        movwf   PORTD
L2:
        btfss   PORTB,1
        goto    L3
        rlncf   PORTD,W
        iorlw   1
        movwf   PORTD
L3:
        movlw   0
        cpfseq  PORTB
        goto    L1
        goto    L3
```



Problem #6

## Binary Inputs

A thermistor has the following temperature - resistance relationship:

$$R = 1000 \exp\left(\frac{3905}{T+273} - \frac{3905}{278}\right)\ \Omega$$

where T is the temperature in degrees C.

7) Design a circuit which outputs

- 0V when T < 5C
- 5V when T > 5C

5 degrees C gives 2566.7 Ohms. Using a voltage divider with a 2k resistor gives

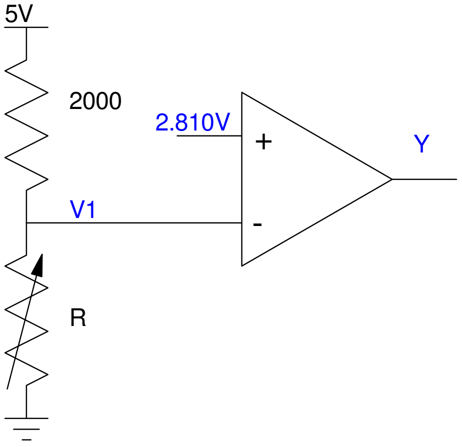
$$V_1 = \left(\frac{2566.7}{2566.7+2000}\right) 5V = 2.810V$$

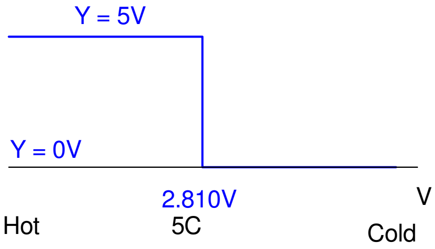
2.810V corresponds to 5C.

When it's really hot

- R goes to zero
- V1 goes to zero
- Vout goes to 5V

so connect to the minus input





8) Design a circuit which outputs

- 0V when T < 0C
- 5V when T > 5C
- No change for 0C < T < 5C

0C gives

- R = 3320.12 Ohms
- V1 = 3.120V
- Y = 0V

5C gives

- R = 2567.0 Ohms
- V1 = 2.810V
- Y = 5V

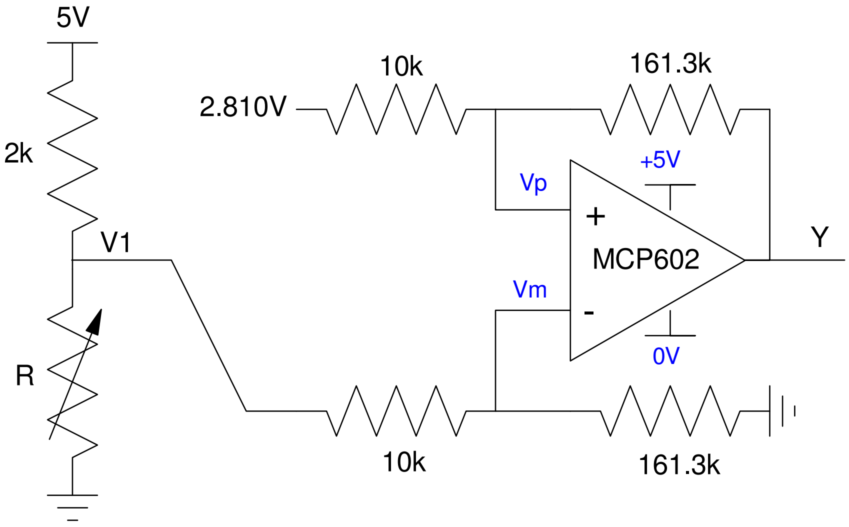
As V1 goes down, Y goes up. Connect to the minus input

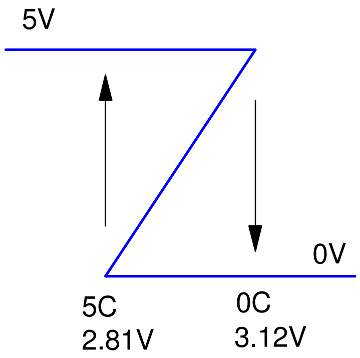
When Y=0, it switches at V1 = 2.810V. Make the offset 2.810V

The gain needed is

$$gain = \left(\frac{\text{change in output}}{\text{change in input}}\right) = \left(\frac{5V-0V}{3.120V-2.810V}\right) = 16.13$$

Make the resistors 16.13 : 1


5V
2k
V1
R
2.810V
10k
161.3k
Vp
+5V
+
MCP602
-
Vm
0V
Y
10k
161.3k


5V
0V
5C
2.81V
0C
3.12V