ECE 376 - Homework #2

Assembler, Flow Charts, Binary Inputs. Due Wednesday, September 6th
Please submit as a hard copy, submit on BlackBoard, or email

Assembler Programming

1) Determine the contents of registers W, A, and B after each assembler command:

Command	W	A	B
; Start	7	6	5
addlw 9	**16**	6	5
addwf A,F	16	**22**	5
sublw 3	**-13 (243)** 3 - 16 = -13	22	5
subwf B,W	**18** 5 - (-13) = 18	22	5
andwf A,F	18	**18** W = 0001 0010 = 18 A = 0001 0110 = 22 ----------------------------- A = 0001 0010 = 18	5
iorwf B,F	18	18	**23** W = 18 = 0001 0010 B = 5 = 0000 0101 ----------------------------- OR = 23 = 0001 0111

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

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

A equ 0
B equ 1
C equ 2

; A = B + 2*C + 3;

        movlw     3
        addwf     B,W
        addwf     C,W
        addwf     C,W
        movwf     A
```

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

Opton #1: Best if multiplying by small numbers

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

A equ 0
B equ 2
C equ 4

; A = B + 2*C + 3;

      movlw     3              ; 3
      movwf     A
      clrf      A+1

      movf      B,W            ; 3 + B
      addwf     A,F
      movf      B+1,W
      addwfc    A+1,F

      movf      C,W            ; 3 + B + C
      addwf     A,F
      movf      C+1,W
      addwfc    A+1,F

      movf      C,W            ; 3 + B + 2C
      addwf     A,F
      movf      C+1,W
      addwfc    A+1,F
```

Option #2: Best when multiplying by large numbers

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

A equ 0
B equ 2
C equ 4

; A = B + 2*C + 3;

      movlw     3              ; 3
      movwf     A
      clrf      A+1

      movf      B,W            ; 3 + B
      addwf     A,F
      movf      B+1,W
      addwfc    A+1,F

      movf      C,W            ; 3 + B + 2*(0:CL)
       mullw    2
      movf      PRODL,W
      addwf     A
      movf      PRODH,W
      addwfc    A+1

      movf      C+1,W          ; 3 + B + 2*(CH:0)
      mullw     2
      movf      PRODL,W
      addwf     A+1
```

4) Convert the following C code to assembler (if-statements)

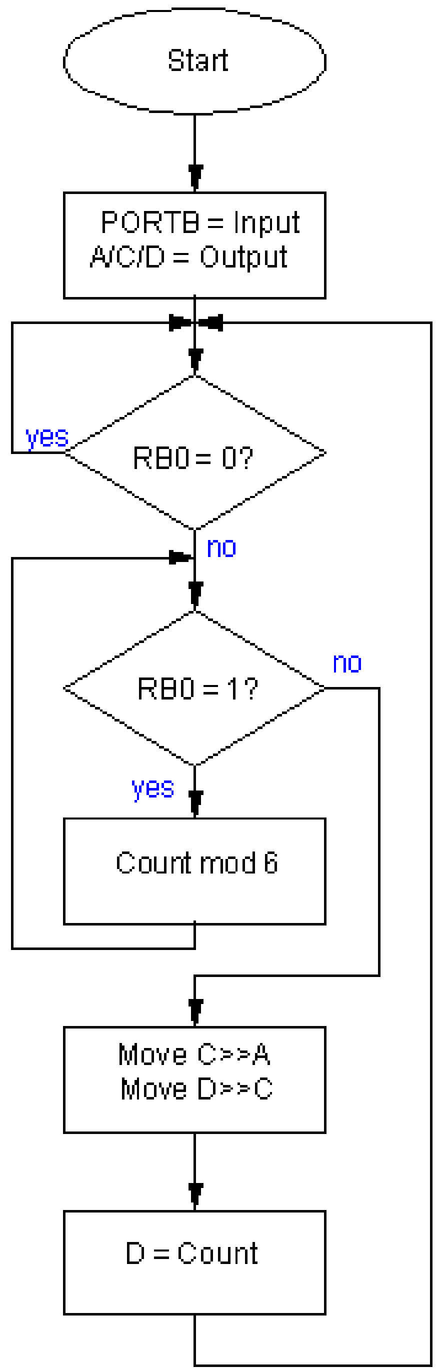
```
; unsigned char A, B;

A equ 0
B equ 1

;A = A & 0x0F;
       movlw     0x0F
       andwf     A,F

;if(A == 0) B = 0;
       movlw     0
       cpfseq    A
       goto      L1
       clrf      B

;if(A == 1) B = 1;
L1:
       movlw     1
       cpfseq    A
       goto      L2
       movlw     1
       movwf     B

; if(A == 2) B = 3;
L2:
       movlw     2
       cpfseq    A
       goto      L3
       movlw     3
       movwf     B

;if(A == 3) B = 7;
L3:
       movlw     3
       cpfseq    A
       goto      L4
       movlw     7
       movwf     B
L4:
       nop
```

5) The flow chart below rolls three six-sided dice, one at a time

- Press RB0 three times to roll the dice
- The values are displayed on PORTA, PORTC, and PORTD

Write the corresponding assembler code.

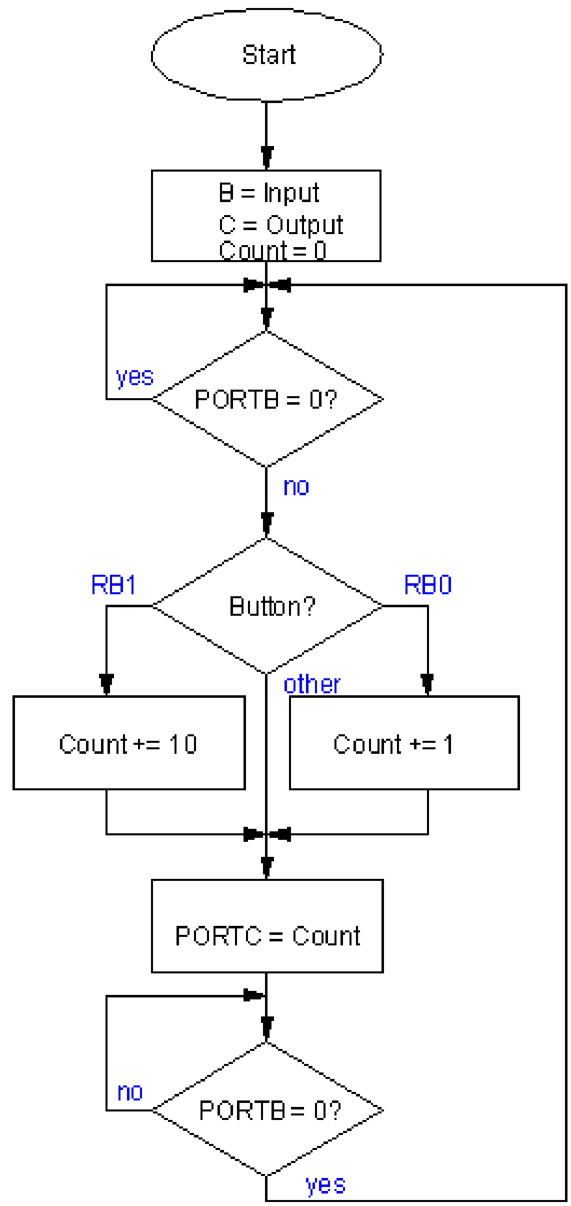
```
COUNT equ 0

        movlw     0xFF
        movwf     TRISB
        clrf      TRISA
        clrf      TRISC
        clrf      TRISD


L1:
        btfss     PORTB,0
        goto      L1


L2:
        btfss     PORTB,0
        goto      L3


        incf      COUNT,F
        movlw     6
        cpfslt    COUNT
        clrf      COUNT
        goto      L2


L3:
        movff     PORTC, PORTA
        movff     PORTD, PORTC


        movff     COUNT, PORTD


        goto      L1
```



Problem 5

6) The flow chart below counts by one (RB0) or ten (RB1) each time you press the button. Write the corresponding assembly code

```
COUNT equ 0

        movlw     0xFF
        movwf     TRISB
        clrf      TRISC
        clrf      COUNT

L1:
        btfss     PORTB,0
        goto      L1

        btfsc     PORTB,1
        goto      L2
        btfsc     PORTB,0
        goto      L3
        goto      L4

L2:
        movlw     10
        addwf     COUNT,F
        goto      L4

L3:
        incf      COUNT,F
        goto      L4


L4:
        movff     COUNT,PORTC

L5:
        btfsc     PORTB,0
        goto      L5

        goto      L1
```



Problem 6

Binary Inputs (hardware)

Assume a thermistor has a resistance-temperature relationship of

$$R = 1000 \cdot \exp\left(\frac{3905}{T+273} - \frac{3905}{298}\right)\Omega$$

7) Design a circuit which outputs

- 0V when T < 10C
- 5V when T > 10C

At 10C, R = 2002.827 Ohms

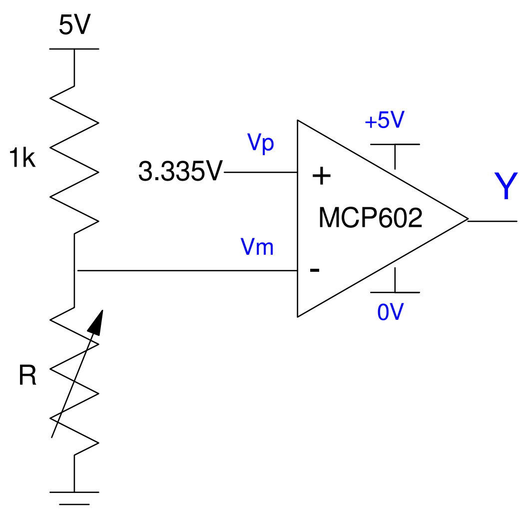
Assuming a voltage divider with R1 = 1000 Ohms

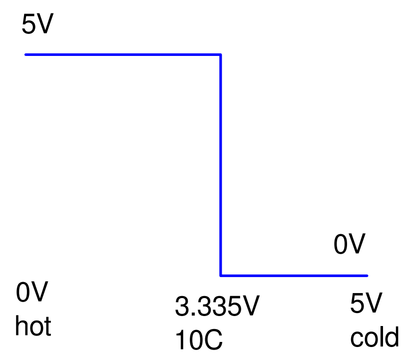
$$V_{in} = \left(\frac{R}{R+1000}\right)5V = 3.335V$$

When T goes to infinity

- R goes to zero
- Vin goes to zero
- Vout goes to +5V

Connect to the minus input





8) Design a circuit which outputs

- 0V when T < 10C
- 5V when T > 15C

Assume a votlage divider with R1 = 1k

At 10C

- R = 2002.28 Ohms
- Vin = 3.335V
- Vout = 0V

At 15C

- R = 1576.17 Ohms
- Vin = 3.059V
- Vout = 5V

V(on) < V(off)

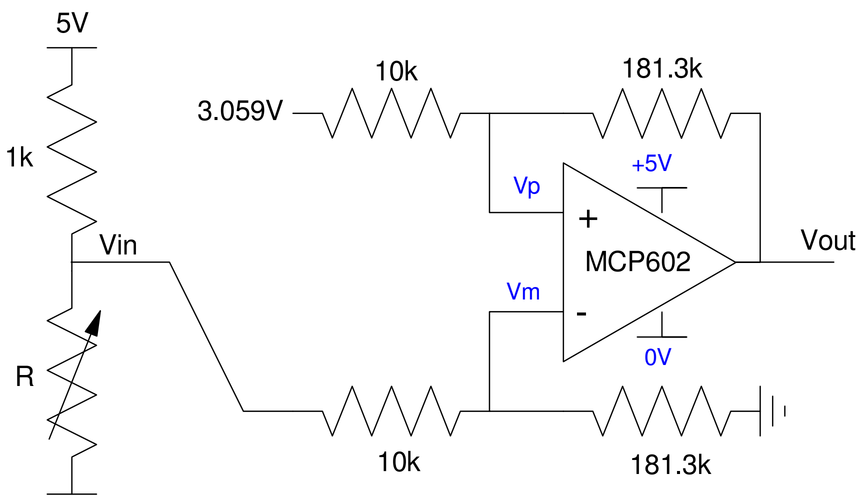
- connect to the minus input

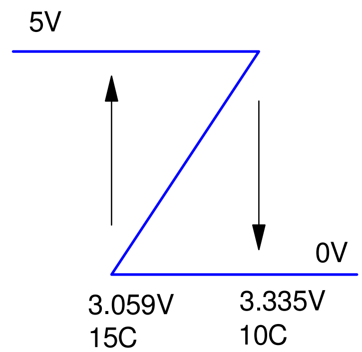
V(on) = 3.059V

- make the offset 3.059V

The gain required is

$$gain = \left(\frac{\text{change in output}}{\text{change in input}}\right) = \left(\frac{5V - 0V}{3.335V - 3.059V}\right) = 18.13$$



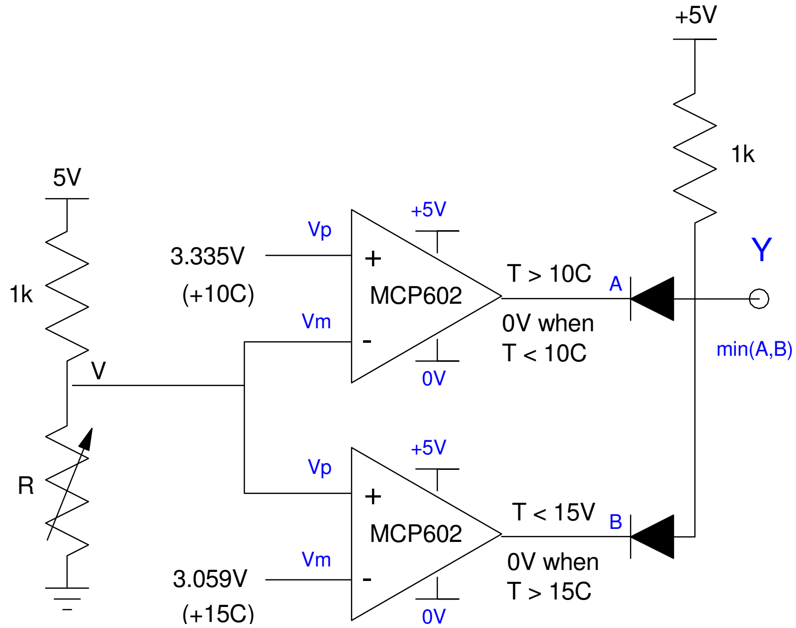


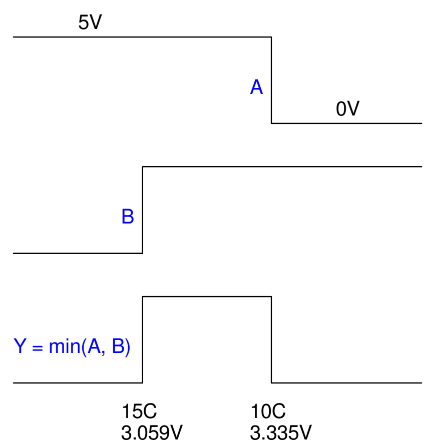
9) Design a circuit which outputs

- 5V when 10C < T < 15C
- 0V otherwise

Option #1: Use some trickery from Electronics

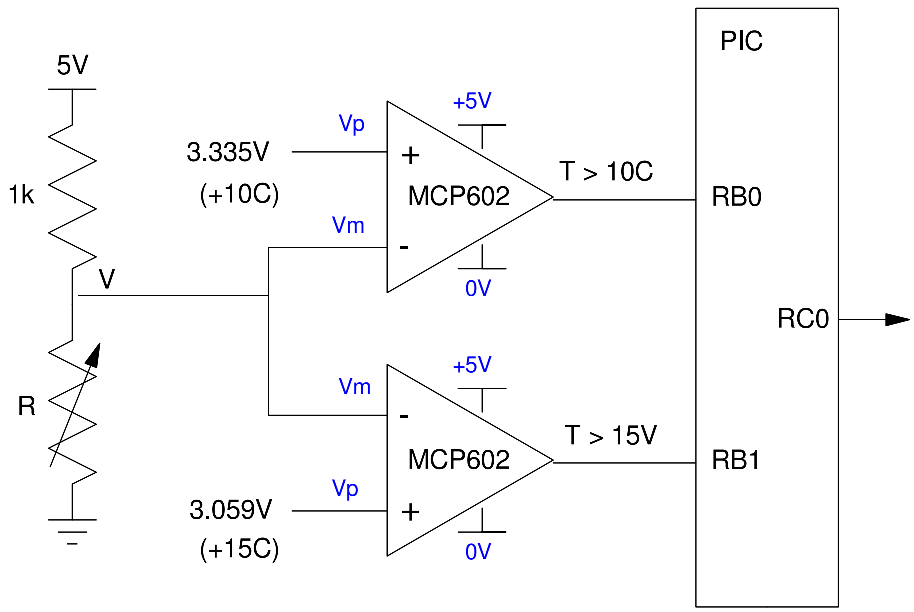
- Lecture #9: Max / Min Circuits





Option #2: Use a microprocessor along with some coding

- anything you can do in hardware you can also do in software



Code for 10 < T < 15

```
ON equ 0

       movlw     0xFF
       movwf     TRISB
       clrf      TRISC

L1:
       btfss     PORTB,0
       goto      OFF

       btfsc     PORTB,1
       goto      OFF
ON:
       bsf       PORTC,0
       goto      END
OFF:
       bcf       PORTC,0
END:
       goto      L1
```