

# ECE 376 - Homework #2

Assembler & Flow Charts - Due Monday, January 27th

## Assembler Programming

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

Command	W	A	B
<code>; Start</code>	15	11	6
<code>incf B,W</code>	7	11	6
<code>iorwf A,W</code>	15	11	6
<code>decf A,F</code>	15	10	6
<code>andwf B,F</code>	15	10	6
<code>movlw 17</code>	17	10	6
<code>subwf A,W</code>	-7	10	6

### Explanation

`iorwf A, W`

A = 11 = b 0000 1011

W = 15 = b 0000 1111

-----  
or = b 0000 1111

`andwf B, F`

B = 6 = b 0000 0110

W = 15 = b 0000 1111

-----  
and = b 0000 0110

## 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 = 2*B + 3*C + 4;
```

```
    movlw    4
    addwf    B,W
    addwf    B,W
    addwf    C,W
    addwf    C,W
    addwf    C,W
    movwf    A
```

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

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

```
A equ 0
```

```
B equ 2
```

```
C equ 4
```

```
A =    AH(1)    :    AL (0)
```

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

```
    movlw    4
    movwf    A
    movlw    0
    movwf    A+1
    movf     B,W
    addwf    A,F
    movf     B+1,W
    addwfc   A+1,F
    movf     B,W
    addwf    A,F
    movf     B+1,W
    addwfc   A+1,F
    movf     C,W
    addwf    A,F
    movf     C+1,W
    addwfc   A+1,F
    movf     C,W
    addwf    A,F
    movf     C+1,W
    addwfc   A+1,F
    movf     C,W
    addwf    A,F
    movf     C+1,W
    addwfc   A+1,F
```

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

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

A equ 0
B equ 0
C equ 0

; if(A > B) C = 5
    movf      B,W
    cpfsgt   A
    goto     elseif
    movlw    5
    movwf   C
    goto     end

; else if (A < B) C = 6
elseif:
    movf      B,W
    cpfslt   A
    goto     else
    movlw    6
    movwf   C
    goto     end

; else C = 7
else:
    movlw    7
    movwf   C

end:
    nop
```

5) The flow chart on the left is for turning your PIC into a 20-sided die for D&D

- Each time you press and release RB0, a random number from 1..20 is displayed on PORTC
- If the roll is a 20, all of the lights on PORTD turn on

Write the corresponding assembly code

```

#include <pic18f4620.inc>

        org      0x800
        movlw   0xFF
        movwf   TRISB
        clrf    TRISC
        clrf    TRISD
        movlw   0x0F
        movwf   ADCON1

L1:      btfss   PORTB, 0
        goto    L1

L2:      movlw   19
        cmfseq  PORTC
        goto    L2a
        clrf    PORTC
        goto    L3
L2a:     incf    PORTC, F

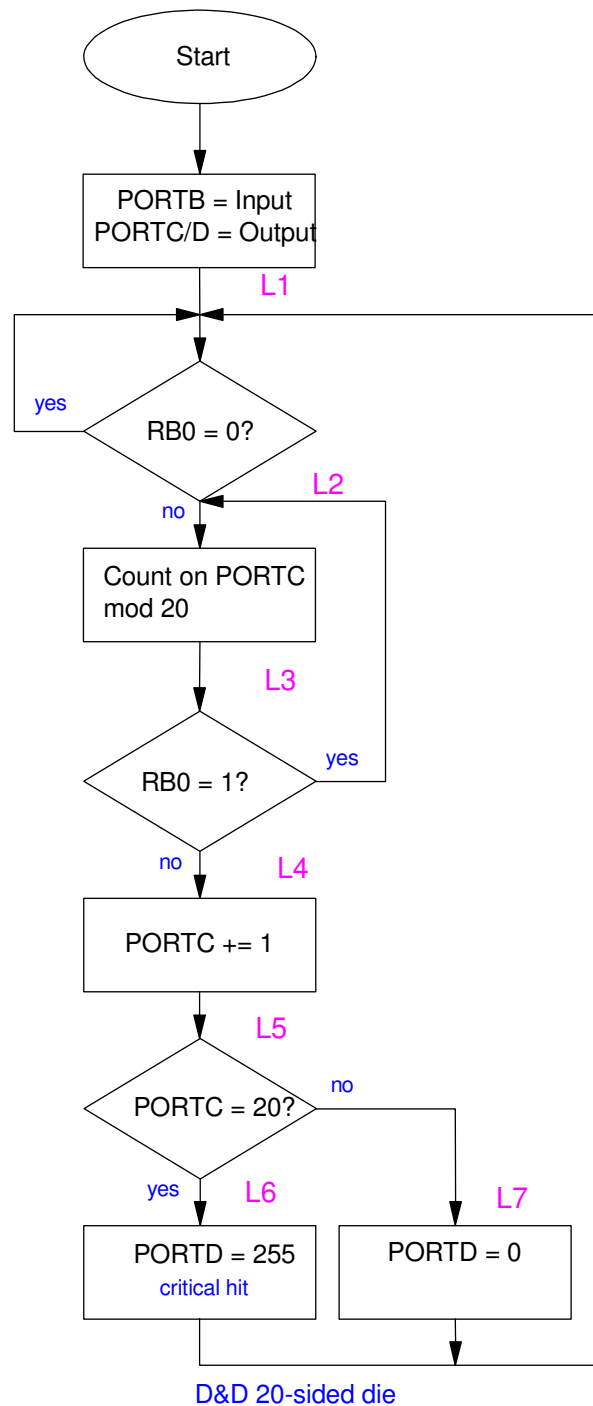
L3:      btfsc   PORTB, 0
        goto    L2

L4:      incf    PORTC, F

L5:      movlw   20
        cpfseq  PORTC
        goto    L7

L6:      movlw   0xFF
        movwf   PORTD
        goto    L1

L7:      clrf    PORTD
        goto    L1
    
```



6) The flow chart to the right generates a three-level flash light

- RB0: Lights turn off
- RB1: Half of the lights on PORTC and PORTD turn on
- RB2: All of the lights on PORTC and PORTD turn on

Write the corresponding assembly code

```
#include <pic18f4620.inc>

    org     0x800
    movlw  0xFF
    movwf  TRISB
    clrf   TRISC
    clrf   TRISD
    movlw  0x0F
    movwf  ADCON1

L1:   btfss PORTB, 0
      goto  L3

L2:   clrf   PORTC
      clrf   PORTD

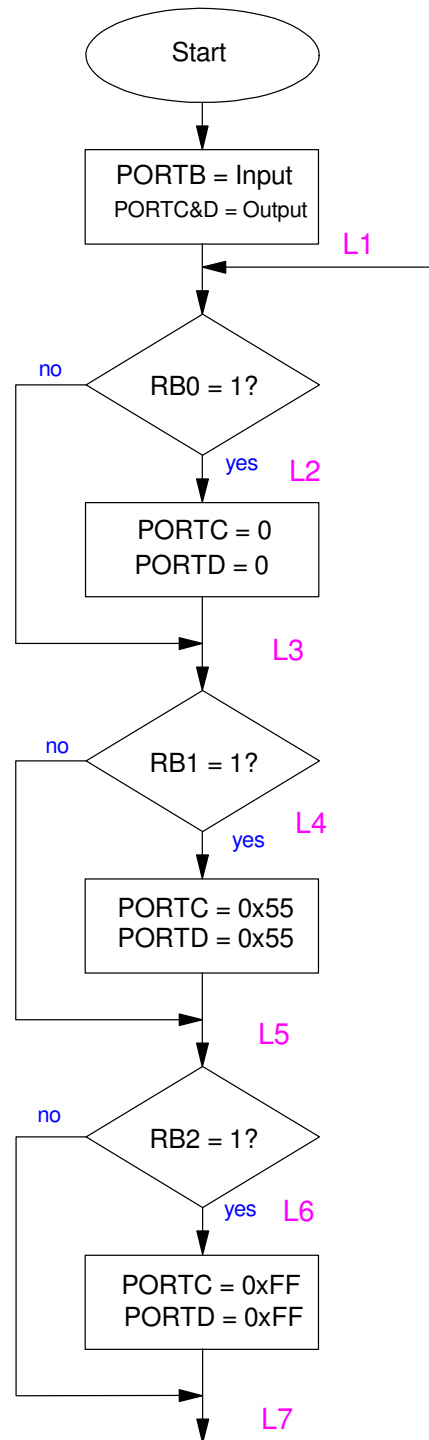
L3:   btfss PORTB, 1
      goto  L5

L4:   movlw  0x55
      movwf  PORTC
      movwf  PORTD

L5:   btfss PORTB, 2
      goto  L7

L6:   movlw  0xFF
      movwf  PORTC
      movwf  PORTD

L7:   goto  L1
```



Problem #6: 3-Level Flashlight