EECE.3170: Microprocessor Systems Design I

Fall 2019

Homework 5 Solution

Show the result of each PIC 16F1829 instruction in the sequences below. Be sure to show not only the state of updated registers, but also the carry (C) and zero (Z) bits. You should assume that the carry bit is initially 0 for each sequence.

Each sequence is worth 25 points.

a.
```
cblock 0x20
    x
endc
```

movlw	0x05	**W = 0x05**
sublw	0x15	**W = 0x15 – W = 0x15 – 0x05 = 0x10**
clrf	x	**x = 0x00**
comf	x, F	**x = ~x = ~0x00 = 0xFF**
xorwf	x, F	**x = x XOR W = 0xFF XOR 0x10 = 0xEF**
swapf	x, W	**W = value in x with nibbles swapped = 0xFE**
btfsc	x, 7	**Test bit 7 of x and skip next instruction if bit is 0** **→ x = 0xEF = $1110\ 1111_2$ → bit 7 = 1 → do not skip**
bsf	x, 0	**Set bit 0 of x → x = $1110\ 1111_2$ before set** **→ No change, since bit 0 already is 1**

b.

```
cblock      0x20
    A
    B
endc

clrf        A           A = 0x00

movlw       0x11        W = 0x11

movwf       B           B = W = 0x11

addlw       0x34        W = W + 0x34 = 0x11 + 0x34 = 0x45

subwf       A, F        A = A – W = 0x00 – 0x45 =  0xBB

comf        A, W        W = ~A = ~ 0xBB = 0x44

swapf       A, F        Swap nibbles of A → A = 0xBB
```

c.

```
cblock 0x40
    var1
endc

movlw       0x1E        W = 0x1E

movwf       var1        var1 = W = 0x1E

rrf         var1, F     Rotate var1 1 bit right through carry
                                → (var1, C) = 0001 1110 0 rotated right
                                → (var1, C) = 0000 1111 0
                                → var1 = 0x0F, C = 0

xorwf       var1, W     W = var1 XOR W = 0x0F XOR 0x1E
                                = 0000 1111 XOR 0001 1110
                                = 0001 0001 = 0x11

btfss       var1, 4     Test bit 4 of var1; skip next instruction if bit = 1
                                → var1 = 0x0F = 0000 1111
                                → Bit is 0 → do not skip

iorlw       0x06        W = W OR 0x06 = 0x11 OR 0x06
                                = 0001 0001 OR 0000 0110
                                = 0001 0111 = 0x17

andwf       var1, F     var1 = var1 AND W = 0x0F AND 0x17
                                = 0000 1111 AND 0001 0111
                                = 0000 0111 = 0x07

bcf         var1, 0     Clear bit 0 of var1
                                → var1 = 0x07 = 0000 0111
                                → After clear, var1 = 0000 0110 = 0x06
```

d.
```
cblock 0x70
    num1, num2
endc

movlw    0xAA          W = 0xAA
andlw    0x0F          W = W AND 0x0F = 0xAA AND 0x0F = 0x0A
movwf    num1          num1 = W = 0x0A
xorlw    0xFF          W = W XOR 0xFF = 0x0A XOR 0xFF = 0xF5
movwf    num2          num2 = W = 0xF5
asrf     num2, F       num2 = num2 >> 1 (keep sign intact)
                               = 0xF5 >> 1 = 0x1111 0101 >> 1
                               = 0x1111 1010 = 0xFA
                       C = bit shifted out = 1
lslf     num1, W       W = num1 << 1 = 0x0A << 1
                          = 0000 1010 << 1 = 0001 0100 = 0x14
                       C = bit shifted out = 0
xorwf    num2, F       num2 = num2 XOR W = 0xFA XOR 0x14 = 0xEE
comf     num2, W       W = ~num2 (flip all bits of num2)
                          = ~0xEE = 0x11
```