# 16.317: Microprocessor Systems Design I

Fall 2015

Homework 4
Due **2:00 PM, Friday, 10/30/15**

**Notes:**
- While typed solutions are preferred, handwritten solutions are acceptable
- All solutions <u>must</u> be legible and contained in one file. Archive files are not acceptable.
- Electronic submissions should be e-mailed to Dr. Geiger at Michael_Geiger@uml.edu.
- This assignment is worth 100 points.

1. (40 points) Write the following subroutine in x86 assembly:

```
int fib(int n)
```

Given a single integer argument, $n$, return the $n$th value of the Fibonacci sequence—a sequence in which each value is the sum of the previous two values. The first 15 values are shown below—note that the first value is returned if n is 0, not 1.

| n      | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7  | 8  | 9  | 10 | 11 | 12  | 13  | 14  |
|--------|---|---|---|---|---|---|---|----|----|----|----|----|-----|-----|-----|
| fib(n) | 0 | 1 | 1 | 2 | 3 | 5 | 8 | 13 | 21 | 34 | 55 | 89 | 144 | 233 | 377 |

Recall that:

- Subroutine arguments are passed on the stack, and can be accessed within the body of the subroutine starting at address EBP+8.

- At the start of each subroutine:

  i.   Save EBP on the stack

  ii.  Copy the current value of the stack pointer (ESP) to EBP

  iii. Create space within the stack for each local variable by subtracting the appropriate value from ESP. For example, if your function uses four integer local variables, each of which contains four bytes, subtract 16 from ESP.

  iv.  Local variables can then be accessed starting at the address EBP-4.

- A subroutine's return value is typically stored in EAX.

See Lectures 15-18 for more details on subroutines, the x86 architecture, and the conversion from high-level concepts to low-level assembly.

2. (60 points) 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.

a.  cblock 0x20
        x
    endc

    movlw     0x05
    sublw     0x15
    clrf      x
    comf      x, F
    xorwf     x, F
    swapf     x, W
    btfsc     x, 7
    bsf       x, 0


b.  cblock    0x20
        A
        B
    endc

    clrf      A
    movlw     0x11
    movwf     B
    addlw     0x34
    subwf     A, F
    comf      A, W
    swapf     A, F


c.  cblock 0x40
            var1
    endc

    movlw     0x1E
    movwf     var1
    rrf       var1, F
    xorwf     var1, W
    btfss     var1, 4
    iorlw     0x06
    andwf     var1, F
    bcf       var1, 0


d.  cblock 0x70
            num1, num2
    endc

    movlw     0xAA
    andlw     0x0F
    movwf     num1
    xorlw     0xFF
    movwf     num2
    asrf      num2, F
    lslf      num1, W
    xorwf     num2, F
    comf      num2, W