16.317: Microprocessor Systems Design I

Summer 2013

Homework 7
Due **Tuesday, 8/13/13—NO LATE ASSIGNMENTS ACCEPTED**

Note that this assignment is for extra credit—your score on this assignment can replace your lowest grade from Homeworks 1-4 (not Homework 5 or 6). The solution to this assignment will be posted early on Wednesday, 8/14/13, so no late submissions will be accepted.

1. *(25 points)* For each of the following x86 instructions, write a sequence of PIC 16F684 instructions that performs an equivalent operation.

Assume each instruction is in a program that starts with the following cblock directive:

```
cblock 0x20
        AL, AH, EAL, EAH  ; All bytes of register EAX
                          ; EAH = most significant byte, AL = least significant
        BL, BH, EBL, EBH  ; All bytes of register EBX
        CL, CH, ECL, ECH  ; All bytes of register ECX
        DL, DH, EDL, EDH  ; All bytes of register EDX
endc
```

In other words, variables are defined for all bytes of x86 registers so that you can write operations that cover any data size, from 8 bits to 32 bits. While register names like "EAL" and "EAH" are not officially part of the x86 architecture, they are defined to provide a way to refer to the upper 16 bits of the 32-bit x86 registers.

__NOTE: each problem is worth 5 points, for a total of 25 points on this problem.__

   a. ADD   EAX, EBX

   b. CMP   AX, BX

   c. JLE   label

   d. SAR   AX, 10

   e. ROR   EAX, 5

2. **(25 points)** Configure Port A so that all bits are outputs, and configure Port C so that all bits are inputs. Repeatedly read a byte from Port C, doing the following operations based on the input values:

   o If bit 0 is set, increment the working register.

   o If bit 0 is clear and bit 1 is set, decrement the working register.

   o If bits 0-1 are clear and bit 2 is set, clear the working register.

   o If bits 0-2 are clear and bit 3 is set, output the current value of the working register to Port A.

   o If bits 0-3 are clear and bit 4 is set, end the program.

Your program should ensure that the working register is set to 0 before you start repeatedly reading from Port C.