ICS 51: Introductory Computer Organization **[Fall 2015]**

Lab #5

In this lab, you will learn to use the following Intel Assembly instructions: MOV, ADD/INC, SUB/DEC, (I)MUL, (I)DIV. After completing and answering each question, show it to your TA.

- **Notes**
- **For this lab, you work individually**
- **You must place all answers in a file named Lab5.asm.**
- **After completing, check your answers with your TA, zip your Lab5.asm file and submit to EEE Dropbox – Lab 5**

Place the answers to the following questions in a file named Lab5.asm.

Q1)
Start Lab5.asm with the assembly program below. Replace the ?? in the comment of each line with the value of the destination operand after each of the following instructions executes:

```
.data
      myByte BYTE 0FFh, 0
.code
      ; QUESTION 1
      mov al,myByte    ; AL =??
      mov ah,[myByte+1]     ; AH = ??
      dec ah           ; AH = ??
      inc al           ; AL = ??
      dec ax           ; AX = ??
```

Q2)
Add the comment & assembly instructions below to Lab 5.asm. For each of the entries marked with ??, show the values of the destination operand and the Sign, Zero, and Carry flags:

```
      ; QUESTION 2
      mov ax,00FFh
      add ax,1   ; AX= ?? SF= ??  ZF= ??  CF= ??
      sub ax,1   ; AX= ?? SF=??  ZF=??  CF=??
      add al,1   ; AL= ??   SF=??  ZF=??  CF=??
      mov bh,6Ch
      add bh,95h ; BH= ??   SF=??  ZF=??  CF=??
      mov al,2
      sub al,3   ; AL= ??   SF=??  ZF=??  CF=??
```

Q3)
Add the comment & assembly instructions below to Lab 5.asm. For each of the entries marked with ??, show the values of the given flags after each operation?

```
; QUESTION 3
mov al,-128
neg al     ; CF = ??    OF = ??

mov ax,8000h
add ax,2   ; CF = ??  OF =  ??
mov ax,0
sub ax,2   ; CF = ??  OF = ??

mov al,-5
sub al,+125     ; OF = ??
```

Q4)
Add the comment & assembly instructions below to Lab 5.asm. For each of the entries marked with ??, show the hexadecimal values of DX and AX after the following instructions execute? Or, if divide overflow occurs, you can indicate that as your answer:

```
; QUESTION 4
mov  ax,0FDFFh  ; -513
cwd
mov  bx,100h
idiv bx ;After idiv: DX = ??,  AX = ??

mov eax,00128765h
mov ecx,10000h
mul ecx ;After mul: EDX = ??, EAX = ??,CF = ??
```

Q5)
Translate the following expression into assembly language. Do not permit `Rval`, `Xval`, `Yval`, and `Zval` to be modified:

Rval = Xval - (-Yval + Zval)

Modify your Lab5.asm `.data` segment with declarations for the variables `Rval`, `Xval`, `Yval`, and `Zval.` Assume that all values are signed doublewords. and initialize their values to ?, `25`, -5, `20` respectively. Modify your `.code` segment with the separation label, **`; QUESTION 5`** and instructions that translate the above high-level expression into assembly instructions.

Q6)
Implement the following expression using signed 32-bit integers:

eax = (ebx * 20) / ecx

Modify your `.code` segment with the separation label, **`; QUESTION 6`** and instructions that translate the above high-level expression into assembly instructions. Initialize `ebx` to `10` and `ecx` to 2.

Q7)
Modify your Lab5.asm `.data` segment with declarations for the variables `var1`, `var2`, and `var3.` Assume that all values are signed doublewords and initialize their values to 5, `20`, -5 respectively. Initialize `ebx` to `10`. Modify your `.code` segment with the separation label, **`; QUESTION 7`** and instructions that translate the above high-level expression into assembly instructions.

Implement the following expression using signed 32-bit integers. Do not modify any variables other than var3:

var3 = (var1 * -var2) / (var3 – ebx)