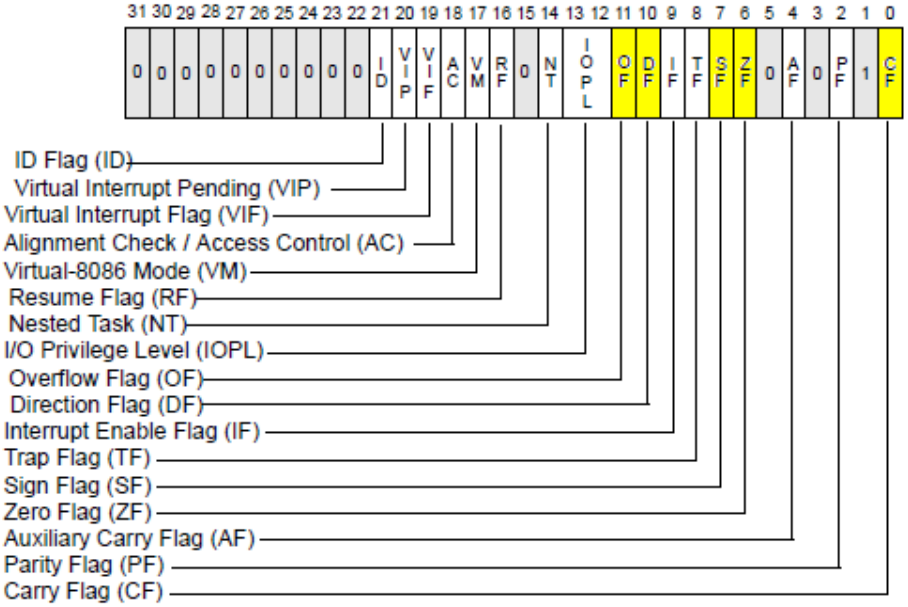
(100 pts) CS3843 Computer Organization Exam #3 Name/abc123:_____
(50 pts) Part 2

Coding: (50 pts)

1. (15 pts) The EFlags register in x86 has bit assignments for the primary flags as shown below. Assume the value of the EFlags register has been put in eax (shown how by code below), write the assembly code to effect the following changes to the flags: Set DF, clear ZF and CF, toggle SF, and set IOPL (bits 12,13) to 3. Do not use built-in assembly instructions such as "cld".

```
// Puts EFlags register on the stack
pushf          ; save eflags register to stack
pop eax        ; put contents of eflags register, from stack, into eax
```




```
// Code to alter the flags here: Leave all other flags unchanged.
```

```
// Puts the altered flags back into the EFlags register
push eax       ; put altered eax on to the stack
popf           ; put values into flags register
```

2. (10 pts) Analyze and explain the following assembly code – what ends up in eax?

```
   call  loc_HERE
loc_HERE:
   pop eax
```

3. (10 pts) Write an inline assembly function that will take the first 4-byte parameter, rotate it right by the number in the 2nd parameter, set the upper 16 bits to zero, and store the result in a local variable at ebp-4. ASSUME the standard stack frame setup – no need to show that code.

4. (15 pts) Write a program to iterate through a data set and alternately xor the data with 0xA5 and 0x5A. The pointer to the data is the 1st parameter and the data length is the 2nd parameter. The first byte of data should be xor'd first, but instead of starting the index at zero, start at –length. So if data is at address 0x1000 and has a length of 0x50, set the starting index to -50 and increment to zero. (Hint: 0x1050 + -0x50 = 0x1000)

```
55          push  ebp
8b ec       mov   ebp, esp
```

```
8b e5       mov   esp, ebp
5d          pop   ebp
c3          ret
```