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

Lab #4

In this lab, you will learn to use the VisualStudio IDE to assemble & debug assembly programs. You will complete this lab in one session. 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 Lab4.txt.**
- **After completing, check your answers with your TA, zip your Lab4.txt file and submit to EEE Dropbox – Lab 4**

The Q&A portion of Lab 4 is designed to be completed after you have finished following the tutorial for Assembling & Debugging programs in Visual Studio using the AddTwo.asm and AddSubTwo.asm programs in the Lab4.zip folder.

1. Once you have completed the tutorial, create an assembly file named, Lab4.asm that contains the following code:

```
;// Lab 4(Lab4.asm)
;// Program Description:
;// Author:
;// Creation Date:
;// Revisions:
;// Date:                  Modified by:

.386
.model flat, stdcall
.stack 4096
ExitProcess PROTO, dwExitCode : DWORD

.data
u_Byte        BYTE    1
s_Byte        SBYTE  -1
u_Word        WORD   0FFFFh
s_Word        SWORD  0FFFFh
u_DWord       DWORD  12345678h
s_DWord       SDWORD 09ABCDEFh

.code
main PROC

MOV    AL,    u_Byte
ADD    AL,    s_Byte

MOV EAX, 0
MOV EBX, 0

MOV    AX, u_Word
MOV    BX, s_Word
ADD AX, BX

MOV EAX, u_Dword
MOV EBX, s_DWord

INVOKE ExitProcess, 0

main ENDP

END main
```

2. Follow the instructions for opening, building, running & debugging an assembly project.
3. Press F10 to begin debugging the program.
4. Open the registers, memory, and disassembly windows.

Place the answers to the following questions in a file named Lab4.txt.
Q1) What is the value in the EIP register?

Copy the value of the EIP register into the memory window.
Q2) What values do you see for the first 5-bytes?

Set a breakpoint at the line: `MOV AL,      u_Byte`
Q3) What is in EAX? AX? AH? AL?

Press F10 to move over the line.
Q4) What is in EAX? AX? AH? AL?

Press F10 to step over: `ADD      AL,      s_Byte`
Q5) What is in AH? AL?
Q6) For EAX, why didn't adding u_Byte + s_Byte = 1 + (-1) = 0?

Step over the lines:

```
MOV EAX, 0
MOV EBX, 0

MOV     AX, u_Word
MOV     BX, s_Word
```

Q7) What value does EAX contain? EBX?

In the watch window, add u_Word and s_Word.
Q8) What value is reported for u_Word? s_Word?
Q9) Both u_Word & s_Word were initialized with the value **`0FFFFh`**, why are their values reported differently?

Step over the following two lines of code:

```
MOV EAX, u_Dword
MOV EBX, s_DWord
```

In the disassembly window, copy the address for u_Dword (ex- highlighted in yellow: `001C1051   mov eax,dword ptr ds:[001C4006h]`). Paste the address into the Memory1 Window.
Q10) What values do you see for the first 4-bytes starting at that address?
Q11) What value is present in EAX?
Q12) Why do the two values in the answers above differ from one another?