**Andrew Tran**

**Computer Science 2253 Lab Experience Five**

Objectives:

1. Understanding common assembler instructions.
2. Utilizing predefined macros from the irvine32.inc library.

The purpose of this lab is for you to create your own assembly language instructions. I recommend you load a project solution into Visual C++ express/studio so you do not have to reset the environment variables each time you are to create an assembler program.

NOTE: If you have changed the location of the Irvine libraries, you will have to correct the project properties settings of the project so Visual Studio can locate these files. The authors website contains detail instructions on this is accomplished. The website is <http://kipirvine.com/asm/gettingStarted/index.htm>

Pay close attention to the status flags since we will be using this to control the branching instructions, i.e. looping and decision.

**What you are to do:**

**Exercise 1:**

The following problems pertain to chapter four in your textbook. Place your answers to each question below the question your word document.

Use the following data segment definitions below to answer questions 1 - 4

.data

var1 SBYTE -4,-2,3,1

var2 WORD 1000h,2000h,3000h,4000h

var3 SWORD -16,-42

var4 DWORD 1,2,3,4,5

1. For each of the following statements, state whether or not the instruction is valid:

a. mov ax,var1?

Invalid.

b. mov ax,var2

Valid.

c. mov eax,var3

Invalid.

d. mov var2,var3

Invalid.

e. movzx ax,var2

Invalid.

f. movzx var2,al

Invalid.

g. mov ds,ax

Valid.

h. mov ds,1000h

Invalid.

1. What will be the hexadecimal value of the destination operand after each of the following instructions execute in sequence?

mov al,var1

FCh

mov ah,[var1+3]

01h

1. What will be the value of the destination operand after each of the following instructions execute in sequence?

mov ax,var2

1000h

mov ax,[var2+4]

3000h

mov ax,var3

FFF0h

mov ax,[var3-2]

4000h

1. What will be the value of the destination operand after each of the following instructions execute in sequence?

mov edx,var4

00000001h

movzx edx,var2

00001000h

mov edx,[var4+4]

00000002h

movsx edx,var1

FFFFFFFCh

1. Write code using byte operands that adds two negative integers and causes the Overﬂow ﬂag to be set.

a BYTE 100000

b BYTE 111111

.code

Mov ax, a

Mov ax,b

1. Write a sequence of two instructions that use addition to set the Zero and Carry ﬂags at the same time.

Mov al FFh

Add al, 1

1. Write a sequence of two instructions that set the Carry ﬂag using subtraction.

Mov al, 3

Sub al, 4

1. Implement the following arithmetic expression in assembly language: EAX = –val2 + 7 – val3 + val1. Assume that val1, val2, and val3 are 32-bit integer variables.

a DWORD 8

b DWORD 6

c DWORD 3

.code

Move ax,7

Sub eax,b

Mov ebx,a

Sub ebx,c

Add eax,ebx

1. Implement the following expression in assembly language: AX = (val2 + BX) –val4. Assume that val2 and val4 are 16-bit integer variables.

a WORD 20

b WORD 10

.code

Add eax,a

Sub eax,b

Mov ax,eax

1. Write a sequence of two instructions that set both the Carry and Overﬂow ﬂags at the same time.

Mov al,FFh

Add al,-128

1. Write the pseudocode algorithm to perform the following task.

Write a program with a loop and indexed addressing that calculates the sum of all the gaps between successive array elements. The array elements are doublewords, sequenced in non-decreasing order. So, for example, the array {0, 2, 5, 9, 10} has gaps of 2, 3, 4, and 1, whose sum equals 10.

Declare dword array

Declare dword gap

Declare dword sum

Declare dword arraysize

Code

Move 0 to esi

Move ecx to arraysize

Decrement ecx

L1:

Move eax into array[esi+1]

Subtract eax from array[esi]

Increment esi

Loop L1

Move 0 to sum

Move 0 to esi

Move arraysize to ecx

Decrement ecx

L2:

Add gap and sum

Increment esi

Loop L2

end

**What to hand in:**

1. Save your word document as {yourName}Lab5.docx. For example TimWrennLab5.docx.
2. Place the file into the D2L DropBox labeled Lab Five.
3. Hand-in a hard copy of the word document.

Due Date: One week after your lab session.