**NUST SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE**

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
| Faculty Member:\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Semester:\_\_\_\_\_\_\_\_\_\_\_\_\_ | Section: \_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Department of Electrical Engineering

EE- 222: Microprocessor Systems

**LAB 02: Declaration and Manipulation of Variables in Assembly Language**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Student name | Reg. No. | Lab Report Marks / 10 | Viva Marks / 5 | Total/15 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**LAB 02:** **Declaration and Manipulation of Variables in Assembly Language**

**Objective:** The aim of this lab is to practice declaring and manipulating variables in assembly language programs and verifying the outputs.

Exercise 1: Write the value of register in blank given in front of each mov and arithmetic instruction.

TITLE Add and Subtract, (AddSub2.asm)

;This program adds and subtracts 32-bit unsigned

;integers and stores the sum in a variable. INCLUDE Irvine32.inc

.data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| var1 | DWORD 10000h | | ;val1 declared as a variable of type DWORD and initialized | |
| var2 | DWORD 40000h | |  |  |
| var3 | DWORD 20000h | |  |  |
| finalVal DWORD ? | |  |  |  |
| Val1 | WORD | 100H |  |  |
| Val2 | WORD | 200H |  |  |
| arrayB | BYTE |  | 10H, 20H, 30H, 40H | |
| arrayW | WORD | | 100h, 200h, 300h, 400h | |
| arrayD | DWORD | | 10000H, 20000H | |
| .code |  |  |  |  |
| main PROC | |  |  |  |
| mov | eax,var1 |  | ; | eax=\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| add | eax,var2 |  | ; | eax=\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| sub | eax,var3 |  | ; | eax=\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| mov | finalVal,eax | ; | | |
| call | DumpRegs | ; | | |
| mov | eax, 0 |  |  |  |
| mov | ebx, 0 |  |  |  |
| mov | ecx, 0 |  |  |  |
| mov | ax, val1 |  | ; | eax=\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| call dumpregs | |  |  |  |
| Mov | bx, val2 |  | ; bx=\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Mov | cl, arrayB |  | ; cl=\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Mov | cl, [arrayB+1] |  | ; cl=\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Mov | cl, [arrayB+2] |  | ; cl=\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Mov | ax, arrayW |  | ; ax=\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Mov | ax, [arrayW+2] | | ; ax=\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Mov | bx, [arrayW+4] | | ; bx=\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Mov | ecx, arrayD |  | ; ecx=\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Mov | ecx, [arrayD+4] | | ; ecx=\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Call dumpregs  exit |  |  |  |  |

main ENDP END main

**Exercise 2:** Once you are done Assemble and run the following program to verify you answer. Note down the contents of registers EAX, EBX and ECX as displayed by the program. Do the registers contents match the expected results?

**Exercise 3:** Write code to achieve the following:

1. Define two 8 bit variables var1, and var2, and initialize these to 20, and 30.
2. Swap the contents of var1 and var2 variables using registers.
3. Display the contents of the registers.

Write down the source code below:

**Exercise 4:** Write codes to evaluate the arithmetic expression “**5+(6-2)**”, by:

1. Using one register only
2. Using two registers only
3. Using three registers

Write down the source codes below:

**Exercise 5**: Explain why each of the following MOV statements are invalid?

.data

bVal BYTE 100

bVal2 BYTE ?

wVal WORD 2

dVal DWORD 5

.code

mov ds,45

mov esi,wVal

mov eip,dVal

mov 25,bVal

mov bVal2,bVal