CSC 3210

Computer Organization and Programming

Lab 6

Answer Sheet

Student Name: **Rafid Shaon**

Section: 06 / CRN: 18119 - Bikram

Lab 6(a)

Fix the errors in the provided code.

Build and Attach screenshot showing the code and “build succeeded” message.

Graphical user interface

Description automatically generated

Lab 6(b)

Debug through each line of instructions.

Take screenshot that includes code and register window.

Record the register content.

and explain the register contents.

Line number: 17

Instruction: mov bx, 0A69Bh

Register values: EBX = 00A1A69B , EIP = 00031014

Screenshot: A screenshot of a computer

Description automatically generated

Explanation: the bx is 16-bit register, and given data is 20 bits, so only 16 bits LSB's are moved to bx. So bx contains A69B h.

Line number: 18

Instruction: movzx eax, bx

Register values: EAX = 0000A69B, EIP = 00031017

Screenshot: A screenshot of a computer

Description automatically generated

Explanation: movzx is moved the data of bx to eax with 32 bits by extending the bits from 16-32 by 0 because we use zero extension here.

Line number: 20

Instruction: movzx eax, myByte1

Register values: EAX = 0000009B , EIP = 0003101E

Screenshot: A screenshot of a computer

Description automatically generated

Explanation: the instruction moved the data of Byte1 i.e., 9B h to eax but with extension to 32 bits with zero.

Line number: 22

Instruction: mov bx, 0A69Bh

Register values: EIP = 00031022

Screenshot: A screenshot of a computer

Description automatically generated

Explanation: Same as line 17…

Line number: 23

Instruction: movsx eax, bx

Register values: EAX = FFFFA69B , EIP = 00031025

Screenshot: A screenshot of a computer

Description automatically generated

Explanation: The instruction is same as 1st. But here instruction movsx is used that extends the bits to 32 by 1 as unsigned number.

So, 16 to 32 bits are set to 1 as

1111 1111 1111 1111 i.e., FFFF in hexadecimal.

(Copy paste this format if you need more)

Lab 6(c)

Debug through each line of instructions.

Take screenshot that includes code and register window.

Record the register content.

and explain the register contents.

Line number: 19

Instruction: mov eax, arrayD

Register values: EAX = 00010000 , EIP = 00C61015

Screenshot: A screenshot of a computer

Description automatically generated with medium confidence

Explanation: EAX register is 32-bit long with a signed integer variable. This register is updated with arrayD (10000h) in its contents by mov.

Line number: 20

Instruction: mov ebx, [arrayD+4]

Register values: EBX = 00020000 , EIP = 00C6101B

Screenshot: A screenshot of a computer

Description automatically generated

Explanation: EBX register is 32-bit long with an unsigned integer variable. This register is updated with 200000h. By adding 4 to arrayD, we are accessing the array element that is 4 bytes offset to the first one.

Line number: 21

Instruction: mov edx, [arrayD+8]

Register values: EDX = 00030000 , EIP = 00C61021

Screenshot: A screenshot of a computer

Description automatically generated

Explanation: EDX register is updated with 300000h. By adding 8 to arrayD, we are accessing the array element that is 8 bytes away from the first byte.

Lab 6(d)

Create a new project to run the following program.

Declare an array in the data segment: arrayB WORD 1,2,3,4

Write code to Rearrange the array as follows: 4,3,1,2

Add the screenshot of your code here.

Graphical user interface

Description automatically generated