CSC 3210

Computer Organization and Programming

Lab 3 (b)

Answer Sheet

Student Name: **Asrar Syed**

Section: **022 / CRN: 17915**

Debug through each line of code and explain the register content and flags.

(We already answered line 10 to 13 for your reference. Start writing your answer from Line 14)

Line: 10

Instruction: mov eax, 12345678h

Register value: EAX = 12345678

Explanation: 12345678 is a hexadecimal value which is 32-bit in binary. EAX register is also 32-bit.

Line 11:

Instruction: mov ax, 1122h

Register value: EAX = 12341122h

Explanation: 1122 is hexadecimal and it is 16-bit in binary. this mov instruction only updates AX (16 bit) register, a part of EAX register. That’s why you can see that the upper portion of EAX register is NOT updated.

Line 12:

Instruction: mov bl, al

Register value: EBX = \_ \_ \_ \_ \_ \_ 22

Explanation: AL register is 8-bit long. When you mov the content of al register (22) to BL register, it only updates the first 8-bit of the EBX register. The rest contains the garbage value.

Line 13:

Instruction: mov bl, ah

Register value: EBX = \_ \_ \_ \_ \_ \_ 11

Explanation: Ah register is 8-bit long. When you mov the content of AH register (11) to BL register, it only updates the first 8-bit of the EBX register. The rest contains the garbage value.

Line 14:

Instruction: mov al, 89h

**Register value:** EAX = 12341189 or EAX = \_ \_ \_ \_ \_ \_ 89

**Explanation:** The AL register is 8-bits long, and this instruction moves (89) which is in hexadecimal form to the AL register and only updates the first 8-bit of the EAX register. The rest contains the garbage value that do not change.

Line 15:

Instruction: add al, 10h

Do you see any change in flags? YES

Show the step of the hexadecimal addition. SHOWN AT BOTTOM

**Register value:** EAX = 12341199 or EAX = \_ \_ \_ \_ \_ \_ 99

**Explanation:** The AL register is 8-bits long and this instruction adds (10) in hexadecimal form to whatever was originally in the AL register. This again only updated the first 8-bits of the EAX register. The rest contains the garbage value that do not change.

Line 16:

Instruction: sub al, al

Do you see any change in flags? YES

Show the step of the hexadecimal addition. SHOWN AT BOTTOM

**Register value:** EAX = 12341100 or EAX = \_ \_ \_ \_ \_ \_ 00

**Explanation:** The AL register is 8-bits long and this instruction subtracts the AL register by itself, resulting in the 00 that we see in the AL register. This again only updated the first 8-bits of the EAX register. The rest contains the garbage value that do not change.

Line 17, 18:

Instruction:

mov al, 98h

add al, 89h

Do you see any change in flags? NO

Show the step of the hexadecimal addition. SHOWN AT BOTTOM

**Register value:** EAX = 12341198 or EAX = \_ \_ \_ \_ \_ \_ 98 (17th Line)

**Register value:** EAX = 12341121 or EAX = \_ \_ \_ \_ \_ \_ 21 (18th Line)

**Explanation (17th Line):** This instruction moves (98) which is in hexadecimal form to the AL register and this only updates the first 8-bit of the EAX register. The rest contains the garbage value that do not change.

**Explanation (18th Line):** This instruction adds (89) which is in hexadecimal form to the AL register and this only updates the first 8-bit of the EAX register. The rest contains the garbage value that do not change.

