CSC 3210

Computer Organization and Programming

Lab 3 (b)

Answer Sheet

Student Name: **Rafid Shaon**

Section: 06 / CRN: 18119 - Bikram

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

(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 of EAX register, after executing line 14.

Explain the content of the EAX register.

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

Explanation: al is an 8-bit register. When we mov 89h to al register, it updates only the least significant byte of the EAX register with 89h at al position.

Line 15:

Instruction: add al, 10h

Register value of EAX, after executing line 15:

Show the step of the hexadecimal addition.

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

Explanation: Here al is an 8-bit register. When we add 10h to al register, it updates only the least significant byte of the EAX register with sum of 89h + 10h = 99h at al position.

8 9h

+1 0h

----------

9 9h = 99h (it is like byte-by-byte addition. In hexadecimal 0,1,2,3,4,5,6,7,8,9 and then in hexadecimal 10 written as A, 11 as B, 12 as C, 13 as D, 14 as E, 15 as F. [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F]

Line 16:

Instruction: sub al, al

What Register value of EAX, after executing line 15?

Show the step of the hexadecimal subtraction.

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

Explanation: Here al is an 8-bit register. When we sub 99h from 99h from al register, it updates only the least significant byte of the EAX register with sub of 99h - 99h = 00h at al position.

9 9h

- 9 9h

---------

0 0h = 00h (it is like byte-by-byte subtraction 9h - 9h = 0)

Line 17, 18:

Instruction:

mov al, 98h

add al, 89h

Register value of EAX, after executing line 17 and 18:

Show the step of the hexadecimal addition.

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

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

There will be flag CF = 1 will be set as 98h + 89h overflow the byte with carry 1

Explanation: al is an 8-bit register. When we mov 98h to al register, it updates only the least significant byte of the EAX register with 98h at al position. (17th line)

Explanation: Here al is an 8-bit register. When we add 89h to al register, it updates only the least significant byte of the EAX register with sum of 98h + 89h = 21h at al position. (18th line)

9 8h

+ 8 9h

~~1~~ ----------

8h + 9h = 17h, 17h – 16h = 1, now carry the 1

9h + 8h = 17h, 17h + the 1 carry = 18h – 16h = 2, now carry the 1 … which is an overflow now

2 1h = 21h (It results extra carry 1 CF = 1) (it is like byte-by-byte addition. In hexadecimal 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 and then in hexadecimal 10 written as A, 11 as B, 12 as C, 13 as D, 14 as E, 15 as F. [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F]

for 16 in hexadecimal, it is 10h.

for 17 in hexadecimal, it is 11h.

for 18 in hexadecimal, it is 12h.