CSC 3210

Computer Organization and Programming

Lab 3 (b)

Answer Sheet

**Student Name: Hemant Kosaraju**

**Section: 020**

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

What Register value of EAX register, after executing line 14. **Register value: EAX = 12341189h**

Explain the content of the EAX register.

**After executing line 14 the content of the EAX register is equal to 12341189h because with binary each digit or bit represents 1 bit so 4-bit register will have four 0s or 1s with range -8 to 7, though for the hexadecimal each digit represents 4-bits and therefore a 32-bit register can store 8 hexadecimal values which in this situation of code were assigned to be 1 2 3 4 1 1 8 9 h which only are set to the EAX register because the instruction did not add or sub anything and no modification instruction was placed only a mov instruction was placed which assigns EAX register, where many calculations happen, to 12341189h**

Line 15:

Instruction: add al, 10h

What Register value of EAX, after executing line 15? **Register Value: EAX = 12341199h**

Do you see any change in flags? **Flags PL changed from 0 to 1 and ZR changed from 1 to 0**

Show the step of the hexadecimal addition.

|  |
| --- |
| 1 2 3 4 1 1 8 9 h |
| 0 0 0 0 0 0 1 0 h |
| 1 2 3 4 1 1 9 9 h |

**The al register represents the 8-bit storage register from the EAX 32-bit general purpose register and it is adding 10h to the last two digits of the initial value (because they are the 8-bit hexadecimal values that are represented by al register) that was moved to the eax and ax registers**

Line 16:

Instruction: sub al, al

What Register value of EAX, after executing line 16? **Register Value: EAX = 12341100h**

Do you see any change in flags? **Flags PL changed from 1 to 0 and ZR changed from 0 to 1**

Show the step of the hexadecimal subtraction.

|  |
| --- |
| 1 2 3 4 1 1 9 9 h |
| 0 0 0 0 0 0 9 9 h |
| 1 2 3 4 1 1 0 0 h |

**The 99h represented in hexadecimal is subtracted from the added value in the previous step because sub al, al would represent the smallest storage of 8-bits therefore the two digits in the 8-bit representation are the two values at the end being 99**

Line 17, 18:

Instruction:

mov al, 98h

add al, 89h

What Register value of EAX, after executing line 17 and 18? **Register Value: EAX = 12341121h**

Do you see any change in flags? **Flags OV changed from 0 to 1, ZR changed from 1 to 0, AC changed from 0 to 1, and CY changed from 0 to 1**

Show the step of the hexadecimal addition.



|  |
| --- |
| 1 2 3 4 1 1 9 8 h |
| 0 0 0 0 0 0 8 9 h |
| 1 2 3 4 1 1 2 1 h |

**The explanation for why the hexadecimal value changes the last two digits to 21h for the al 8 - bit register after adding 98h and 89h from the al registers is because a basic understanding of hexadecimal value is that each hexadecimal value represents 4 bits of storage and the values are base 16 (start at 0; end at F, where F represents 15 in decimal), therefore, when mov al, 98h was executed on line 17 from the EAX 32-bit register it selected the section containing al 8-bit register which are the last two digits and set them to 98h making the EAX register store the value 12341198h and then on line 18 it adds 89h to the al register which is represented by add al, 89h after the calculation takes place 8 + 9 = 17 however that is an error since hexadecimal is maxed out at F (1111 in binary; 15 in decimal) so we can directly do 17 MOD 16 which would give one. Though, to understand the carry do 17 / 16 = 1.0625 then subtract the 1 as a carry and with the remaining value 0.0625 × 16 = 1 which gets placed as a result. Now, 9 + 8 = 17 + 1 (increment on top) = 18 / 16 = 1.125 then subtract 1, with the remaining value 0.125 × 16 = 2 which gets placed as a result. The final EAX 32-bit register value becomes 12341121h.**