Assignment

COEN 311

Computer Organization and Software

Assignment #4

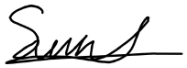
Sunil Kublalsingh

ID: 40212432

Professor: Anjali Agarwal

"I certify that this submission is my original work and meets the Faculty's Expectations of Originality"

Sunil Kublalsingh
Student ID: 40212432



March 28, 2022

Sunil Kublalsingh
Student ID: 40212432
Course: COEN 311 W – Computer Organization and Software
Professor: Anjali Agarwal

Question 1

1- What operation is performed by the instruction sequence that follows? **(10 points)**
Assume (DS) = 0100, (SI) = 0200 H

```
MOV AH, [SI]
SAHF
```

AH=value at address 0200 H

SAHF loads the flags with the value of AH which is at address 0100:0200 H.

Question 2

2- Assume the status of registers and memory are given as below:

(AX) = 8001 H, (SI)= 0200 H, (DI) = 0300 H, (DS:100H) = F0 H, (DS:200H) = F0 H, (DS:201H) = 01 H, (DS: 300) =34H, (DS:301) = 12 H

Describe the operation of the following instructions: **(10 points)**

Explain what is the result of each instruction.

a) CMP [0200H], AL
b) CMP WORD PTR [DI], 1234H

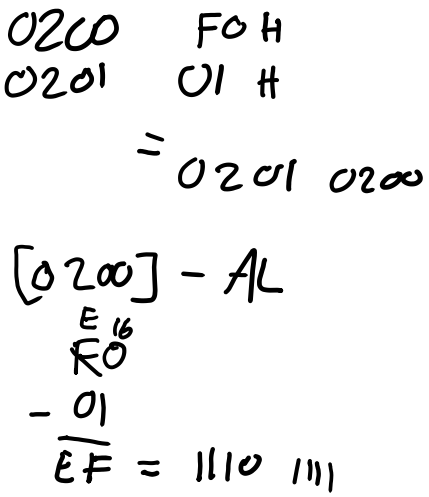
a)

Compares the value at 0100:0200 with the value of AL

[0200]-AL = F0 –01 = EF = 1110 1111

SF=1 because our answer was EF which is a negative value

AF=1 because our answer had a carry over the low to high nibble.



b)

Compares the value at offset 0300 with 1234H

Sunil Kublalsingh
Student ID: 40212432
Course: COEN 311 W – Computer Organization and Software
Professor: Anjali Agarwal

We use DS:300 + DS:301 since it's a word

1234-1234=0

ZF=1 because our answer was a 0

PF=1 because there is an even number of 1s in 0 (0 ones)

Question 3

3- Assume the values are unsigned, what happens to ZF and CF status flags after the following instructions are executed. Assume that they are both initially cleared. **(10 points)**

```
MOV BX, 1111H
MOV AX, BBBB H
CMP BX, AX
```

BX=1111

AX=BBBB

BX-AX=1111-BBBB

ZF=0 because they aren't equal

CF=1 because we need a borrow

Question 4

4- Identify the type of the jump and the type of the operand in the following instructions: **(18 points)**

a) JMP 10H
b) JMP 1000H
c) JMP WORD PTR [SI]
d) JNC 20 H
e) JNP 1000H
f) JO DWORD PTR [BX]

a) Unconditional jump, short label
b) Unconditional jump, near label
c) Unconditional jump, Memptr 16
d) Conditional jump, short label
e) Conditional jump, near label
f) Conditional jump, Memptr 32

Sunil Kublalsingh
Student ID: 40212432
Course: COEN 311 W – Computer Organization and Software
Professor: Anjali Agarwal

Question 5

5- Assume:
(CS) = 1075 H, (IP) = 0300 H, (SI) = 0100H , (DS:100H)= 00H, (DS:101H) = 10H

To what address is program control passed in each of the following instructions: (**12 points**)

a) JMP 10H
b) JMP 1000H
c) JMP WORD PTR [SI]

a) IP=0010H, address = 1075x10+0010 = 10750+0010=$10760
b) IP=1000 H, address = 1075x10+1000=10750+1000=$11750
c) IP=1000 H, address =1075x10+1000 = $11750

Question 6

6- Encode the following instructions using the formats given in class notes. Note that there may be special cases that do not follow the general format: **(10 points)**

a) INC CX
b) MOV [SI], BX
c) MOV CL, 20
d) ADD AX, [DI] +1234H
e) MOV AX, [0100H]

a) 01000 001 = 0100 0001 = $41
b) 100010 0 1 00 011 100 = 1000 1001 0001 1100 = $891C
c) 1011 0 001 0010 0000 = 1011 0001 0010 0000 = $B120
d) 000000 1 1 10 000 101 (34 12) = 0000 0011 1000 0101 (34 12) = $03853412
e) 100010 1 1 00 000 110 (00 01) = 1000 1011 0000 0110 (00 01) = $8B060001

Question 7

7- Write a program that compares two arrays, A and B. Each array contains 100 8-bit signed numbers. Compare corresponding elements of the two arrays until either two elements are found to be unequal or all elements of the arrays have been compared and found to be equal. Update the data segment to the new DATASEGADDR value, and assume the arrays start at offsets A000 H and B000 H in this segment, respectively. If the two arrays are found to be unequal save the address of the first unequal element of A array in the memory location with the offset address FOUND in this data segment. If the two arrays are equal, write 0 in this memory location FOUND. **(30 points)**

```
_start:
        mov esi, 0 ; the index we start with
        mov cx, 100 ; the number of elements in the array
myloop:
        mov ax, DATASEGADDR
        mov ds, ax
        mov al, [A000+esi]
        mov ah, [B000+esi]
        cmp al, ah
        jne notequal
        ; if we end up incrementing si, it means the items were equal
```

```
        inc si
        cmp si,cx
        je exactArray ; jump to exactArray if all elements in the arrays are equal
        jmp myloop ; otherwise, jump back to myloop to compare the next element
notequal:
        mov BYTE [FOUND], al ; place the value of first array in memory location FOUND
        jmp _exit
exactArray:
        mov BYTE [FOUND], $0
_exit:
        mov eax,1
        mov ebx,0
        int 80h
```