**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

# Question 1

Graphical user interface, text

Description automatically generated

AH=value at address 0200 H

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

# Question 2

Text

Description automatically generated

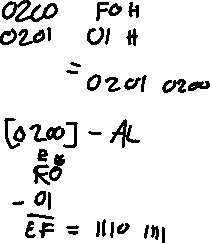
a)

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

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

Text

Description automatically generated

BX=1111

AX=BBBB

BX-AX=1111-BBBB

ZF=0 because they aren’t equal

CF=1 because we need a borrow

# Question 4

Text

Description automatically generated

1. Unconditional jump, short label
2. Unconditional jump, near label
3. Unconditional jump, Memptr 16
4. Conditional jump, short label
5. Conditional jump, near label
6. Conditional jump, Memptr 32

# Question 5

Graphical user interface, text

Description automatically generated

1. IP=0010H, address = 1075x10+0010 = 10750+0010=$10760
2. IP=1000 H, address = 1075x10+1000=10750+1000=$11750
3. IP=1000 H, address =1075x10+1000 = $11750

# Question 6

Text

Description automatically generated

1. 01000 001 = 0100 0001 = $41
2. 100010 0 1 00 011 100 = 1000 1001 0001 1100 = $891C
3. 1011 0 001 0010 0000 = 1011 0001 0010 0000 = $B120
4. 000000 1 1 10 000 101 (34 12) = 0000 0011 1000 0101 (34 12) = $03853412
5. 100010 1 1 00 000 110 (00 01) = 1000 1011 0000 0110 (00 01) = $8B060001

# Question 7

A screenshot of a computer

Description automatically generated with medium confidence

\_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