

Computer Organization and Architecture (EET2211)

LAB VIII: Calculate average of N 16-bit numbers

Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar

Branch:			Section:
S. No.	Name	Registration No.	Signature
52	Saswat Mohanty	1941012407	<i>Saswat Mohanty</i>

Marks: ____/10

Remarks:

Teacher's Signature

I. OBJECTIVE:

1. Write a program to calculate average of N 16-bit numbers

II. PRE-LAB

For Obj. 1:

- a. Calculate average of N 16-bit numbers.

[1500h] = 03h

[1501h] = 10h

[1502h] = 10h

[1503h] = 10h

Output: 10h

- b. Write the assembly code.

```
org 100h
mov ax,0000h
mov ds,ax
mov si,1500h
mov di,1510h
mov ax,0000h
mov cl,[si]
mov bl,cl
inc si
loop: add al,[si]
      adc ah,00
      inc si
      dec cl
      jnz loop
div bl
mov [di],ax
```

hlt

ret

III. LAB:

Assembly Program:

For Obj. 1:

```
; SASWAT MOHANTY
; 1941012407

; Write a program to calculate average of N 16-bit numbers

org 100h

mov ax,0000h
mov ds,ax
mov si,1500h      ; total number of elements = 07
mov di,1510h      ; address value to store the output = average of the elements
mov ax,0000h
mov cl,[si]       ; move the value at 1500 to cl
mov bl,cl         ; move the value of cl to bl
inc si            ; increment the value of si by 1
loop: add al,[si]  ; add and store the sum of elements in al
adc ah,00
inc si            ; move to the next element in the array i.e. 1502
dec cl           ; decrease the count of cl by 1
jnz loop         ; repeat the loop until cl becomes 0
div bl           ; avg = (sum of elements) / (total no. of elements)
mov [di],ax      ; the average value is stored in the specified memory location
hlt

ret
```

Observations (with screen shots):

For Obj. 1:

The screenshot displays the COA Emulator interface. The main window shows the assembly code for a program to calculate the average of 16-bit numbers. The code is as follows:

```
01 ; SASWAT MOHANTY
02 ; 1941012407
03
04 ; Write a program to calculate average of N 16-bit numbers
05
06 org 100h
07
08 mov ax,0000h
09 mov ds,ax
10 mov si,1500h ; total number of elements = 07
11 mov di,1510h ; address value to store the output = average of the elements
12 mov ax,0000h
13 mov cl,[si] ; move the value at 1500 to cl
14 mov bl,cl ; move the value of cl to bl
15 inc si ; increment the value of si by 1
16 loop: add al,[si] ; add and store the sum of elements in al
17 adc ah,00
18 inc si ; move to the next element in the array i.e. 1502
19 dec cl ; decrease the count of cl by 1
20 jnz loop ; repeat the loop until cl becomes 0
21 div bl ; avg = (sum of elements) / (total no. of elements)
22 mov [di],ax ; the average value is stored in the specified memory location
23 hlt
24
25 ret
```

The bottom window shows the Random Access Memory (RAM) dump, displaying the memory contents at addresses 0100:0500 to 0100:0560. The dump shows the following data:

Address	0100:0500	0100:0510	0100:0520	0100:0530	0100:0540	0100:0550	0100:0560
0100:0500	07 01 02 03 04 05 06 07-00 00 00 00 00 00 00 00	04 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00

Conclusion:

It can be concluded to determine the largest number in an array when dry run and executed in system found to be same. Thus, the program to determine the largest number in an array was executed.

IV. POST LAB:

1. What is the maximum internal clock frequency of 8086?

The maximum internal clock frequency of 8086 is 5MHz.

2. List few applications of microprocessor-based system.

The use of microprocessor in toys, entertainment equipment and home applications is making them more entertaining and full of features. The use of microprocessors is more widespread and popular. Now the Microprocessors are used in:

- Calculators
- Accounting system
- Games machine
- Complex Industrial Controllers
- Traffic light Control
- Data acquisition systems

3. Briefly explain the following instructions of 8086:

a) **JMP** b) **JZ** c) **JNZ** d) **JC** e) **JNC**

- a. **JMP:** - Used to jump to the provided address to proceed to the next instruction.
- b. **JZ:** - Used to jump if equal/zero flag $ZF = 1$
- c. **JNZ:** - Used to jump if not equal/zero flag $ZF = 0$
- d. **JC:** - Used to jump if carry flag $CF = 1$
- e. **JNC:** - Used to jump if no carry flag ($CF = 0$)

