



**KHULNA UNIVERSITY OF ENGINEERING AND TECHNOLOGY,
KUET**

SESSIONAL REPORT

Course No: CSE 2204

Department of: Computer Science and Engineering

Experiment No: 04

Name of the Experiment: Developing a program that finds the maximum and minimum number from some given number by using registers

Remarks

Date of Performance: 19.04.21

Name: Rifat Arefin

Date of Submission: 21.04.21

Roll: 1807117

Year: 2nd

Semester: 2nd

No. of experiments 04

Name of experiments Develop a program that finds the maximum and minimum digit from some given number by using registers. in assembly language. (zero should be ignored).

Objectives:

1. To obtain the use of registers in various purposes.
2. To get clear knowledge about jump instruction.
3. To know how to handle zero to ignore in this program.
4. To find the maximum and minimum numbers from some given number.

Introduction: There are eight general purpose registers in 8086 microprocessor.

These registers store data elements for processing different functions. The eight registers are respectively AX, BX, CX, DX, SP, BP, SI, DI.

In this program, we use these registers for storing some data from those data, our objective is to find the maximum and minimum numbers. Point to be noted, AX is divided into 2 parts, AL and AH; BX is divided into BL and BH, CX is divided into 2 parts, CH and CL. DX is also divided into 2 parts DH and DL.

To find the minimum number, we keep a temporary ^{register} variable which is used to store minimum number. We should update the variable

registers after each comparison if new minimum value is found. In the same way we have to take a value which is maximum and should store in another register.

Apparatus Required: emu 8086, laptop.

Methodology:

Code:

org 100h

; Here we take some numbers in some registers from which we have to find out the maximum and minimum numbers and zero should be ignored.

mov bl, 01h

mov bh, 03h

mov cl, 00h

mov ch, 07h

mov dh, 00h; to compare is the minimum is

or not? so 0 is kept here

mov al, 0ah; to minimum number is temporarily stored here and initially it is assigned with '0ah'

; find smallest number

c1:

mov ah, bl; keep first element in 'ah' register

cmp dh, ah; compare with 'dh'

jge c2; if the value of ah is zero then go to next label c2.

cmp al, ah; compare al with ah

jge w

c2:

mov ah, bh; move next element stored in 'bh' register in 'ah' register.

cmp dh, ah; compare is it zero or not

jge c3; if zero go to next label c3.

cmp al, ah; comparing with ah

jge x

C3:

mov ah, cl; move the next value stored in cl register to ah register.

cmp dh, ah; comparing zero or not.

jge c4; if zero ignore next instructions

cmp al, ah; comparing with ah

jge y

C4:

mov ah, ch; moving next element to ah

cmp dh, ah.

jge c5

cmp al, ah

jge z

C5:

mov dl, al; the minimum value of the numbers which was temporarily stored in 'al' is kept dl finally

; To find larger number we perform similar
; process except jump instruction. Here
; we change the code
; we use jbe instead of using jge

mov ~~ah~~ al, 00h; initially al is assigned with
0h.

a1:

mov ah, bl

cmp dh, ah

jge a2

cmp al, ah

jbe p

a2:

mov ah, bh

cmp dh, ah

jge a3

cmp ~~ah~~ al, ah

jbe q

a3:

mov ah, cl

cmp dh, ah

jge a4

cmp al, ah

jbe n.

a4:

mov ah, ch

cmp dh, ah

jge a5

cmp al, ah

jbe s

a5:

mov dh, al ; keep the maximum number

ret

among the given numbers to
dh register which was initially
stored in al register.

w: mov al, ah

jmp c2

x: mov al, ah

jmp c3

Y: mov al, ah
jmp e4

Z: mov al, ah
jmp e5

P: mov al, ah
jmp a2

Q: mov al, ah
jmp a3

R: mov al, ah
jmp a4

S: mov al, ah
jmp a5

Result and discussion: For finding out the maximum and minimum numbers from some numbers, we used jump instructions. We used both conditional and unconditional jump instructions. The experiment was to find out the smallest and largest numbers from some given

numbers and we got as expected result everytime
so we can ensure that our experiment had
done properly and our program worked well for
every input.

Conclusion: Through this experiment, we are able to
build our own logic to find out the
largest and smallest number from some given numbers.
We used jump instructions and in this program
we used both ~~jump~~ conditional and unconditional
jump instructions. So we have enriched our understanding
in using jump instructions from this
experiment.

References:

1. Microprocessors and Interfacing - by D.V. Hall
2. [emu8086/documentation/index.html](http://emu8086.com/documentation/index.html).