Terna Engineering College Computer Engineering Department

Program: Sem V

Course: Microprocessor Lab

Faculty: ARATHI BOYANAPALLI

LAB Manual

PART A

(PART A: TO BE REFERRED BY STUDENTS)

Experiment No. 5

A.1 Aim:

Write an assembly program to find minimum and maximum no. from a given array. [Use BIOS/DOS interrupts to read input and display results.]

A.2 Prerequisite:

Basic knowledge of 8086 instruction set and interrupts

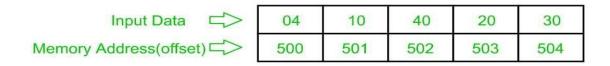
A.3 Outcome:

After successful completion of this experiment, students will be able to

- 1. Use appropriate instructions to program microprocessors to perform various tasks.
- 2. Develop the program in assembly/ mixed language for Intel 8086 processor
- 3. Demonstrate the execution and debugging of assembly/ mixed language program

A.4 Theory:

Finding maximum and minimum number from a given array of n numbers or list elements according to a comparison operator on the elements. The comparison operator helps in deciding whether the number taken in registers is small or large. In the given diagram shows the example finding the largest number from a given array. A random set of numbers are given in an array and is stored in memory location from 2000:500 to memory location 2000:504. The largest among the given array is stored in memory location 2000:600.





A.5 Algorithm:

- **1.** Start, Define memory model
- **2.** Initialize data segment.
- **3.** Initialize the code segment
- **4.** set the counter
- **5.** Initialize the array base pointer.
- **6.** Get the numbers on MAX and MIN.
- **7.** Compare the numbers
- **8.** If num in AL> MAX, Max = num in AL & increment pointer. If num in AL< MIN, Min = num in AL & increment pointer
- **9.** Decrement the counter.
- **10.** If count = 0 stop or else repeat steps 6, 7, 8, 9.
- **11.**Store maximum and minimum number.
- **12.**Stop.

PART B

(PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the ERP or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no ERP access available)

Roll No.: 50	Name: Amey Thakur
Class: TE-Comps B	Batch: B3
Date of Experiment: 13/08/2020	Date of Submission: 13/08/2020
Grade:	

B.1 Observations and learning:

(Software Code written by a student and output of the program)

Input to find Minimum number -

DATA SEGMENT

ARR DB 5,3,7,1,9,2,6,8,4

LEN DW \$-ARR

MIN DB ?

MAX DB?

DATA FNDS

CODE SEGMENT

ASSUME DS:DATA CS:CODE

MEGA:

MOV AX, DATA

MOV DS,AX

LEA SI, ARR

MOV AL, ARR[SI]

MOV MIN, AL

MOV MAX,AL

MOV CX,LEN

FILLY:

MOV AL, ARR[SI]

CMP MIN,AL

JL HITMAN

MOV MIN,AL

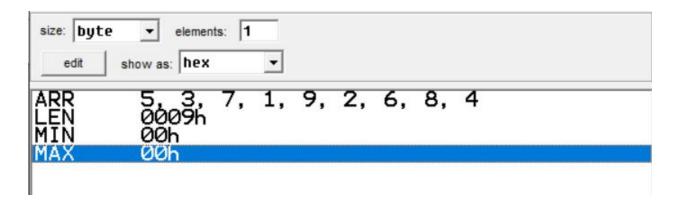
HITMAN:

CMP MAX.AL

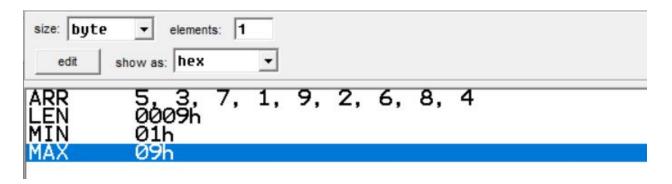
JG SAAKSHI MOV MAX,AL SAAKSHI: INC SI LOOP FILLY MOV AH,4CH INT 21H CODE ENDS

Output

A. Before Execution



B. After Execution



B.2 Conclusion:

(Students must write the conclusion as per the attainment of individual outcome listed above and learning/observation noted in section B.1)

We successfully learned assembly language programs to find minimum and maximum numbers from a given array.

B.5 Question of Curiosity

Q1. List out and explain any 5 different types of JMP instruction with example **Ans:**

- **1.** Jump Instructions The jump instruction transfers the program sequence to the memory address given in the operand based on the specified flag.
- **2.** Jump instructions are 2 types:
 - **A.** Unconditional Jump Instructions
 - **B.** Conditional Jump Instructions.
- **A.** Unconditional Jump Instructions: Transfers the program sequence to the described memory address.

OPCODE	OPERAND	EXPLANATION	EXAMPLE
JMP	address	Jumps to the address	JMP 2050

B. Conditional Jump Instructions: Transfers the program sequence to the described memory address only if the condition is satisfied.

OPCODE	OPERAND	EXPLANATION	EXAMPLE
JC	address	Jumps to the address if carry flag is 1	JC 2050
JNC	address	Jumps to the address if carry flag is 0	JNC 2050
JZ	address	Jumps to the address if zero flag is 1	JZ 2050
JNZ	address	Jumps to the address if zero flag is 0	JNZ 2050
JPE	address	Jumps to the address if parity flag is 1	JPE 2050
JPO	address	Jumps to the address if parity flag is 0	JPO 2050
JM	address	Jumps to the address if sign flag is 1	JM 2050
JP	address	Jumps to the address if sign flag 0	JP 2050

Q2.Write an assembly language program to find factorial of a number **Ans:**

• Input to find factorial of a number -

```
DATA SEGMENT
    NUM DB?
    FACT DB 1H
    RES DB 10 DUP ('$')
    ARCHIT DB "ENTER NUMBER: $"
    MEGA DB 10,13,"FACTORIAL : $"
DATA ENDS
CODE SEGMENT
ASSUME DS:DATA,CS:CODE
    FILLY:
    MOV AX, DATA
    MOV DS,AX
    LEA DX, ARCHIT
    MOV AH,9
    INT 21H
    MOV AH,1
    INT 21H
    SUB AL,30H
    MOV NUM,AL
    MOV AH,0
    MOV AL, FACT
    MOV CH,O
    MOV CL, NUM
    HASAN: MUL CL
    LOOP HASAN
    LEA SI,RES
    CALL AA
    LEA DX,MEGA
    MOV AH,9
    INT 21H
    LEA DX,RES
    MOV AH,9
    INT 21H
    MOV AH,4CH
    INT 21H
CODE ENDS
    AA PROC NEAR
    MOV CX,0
    MOV BX,10
    RUGVED: MOV DX,0
```

DIV BX
ADD DL,30H
PUSH DX
INC CX
CMP AX,9
JG RUGVED
ADD AL,30H
MOV [SI],AL
MAKAD: POP AX
INC SI
MOV [SI],AL
LOOP MAKAD
RET
AA ENDP
END FILLY

• Output (Factorial) -

