

# **MICROPROCESSOR LAB**

## **EXERCISE 7**

GAIS AMEER  
S6 CSE  
ROLL 33

## 7. Bit manipulation to check whether the data is positive or negative

### Program

```
DATA SEGMENT
    NUM DB 10
    POSITIVE DB 1H
    NEGATIVE DB 2H
    RES DB ?
DATA ENDS
CODE SEGMENT
ASSUME DS:DATA CS:CODE
START:
    MOV AX,DATA
    MOV DS,AX
    MOV AL,NUM
    ROL AL,01
    JC DN
    MOV DL,POSITIVE
    JMP EXIT
DN: MOV DL,NEGATIVE
EXIT:
    MOV RES,DL
    MOV AH,4CH
    INT 21H
CODE ENDS
END START
```

## Output

emulator: noname.exe\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	4C	28
BX	00	00
CX	00	2E
DX	00	01
CS	F400	
IP	0204	
SS	0710	
SP	FFFA	
BP	0000	
SI	0000	
DI	0000	
DS	0710	
ES	0700	

F400:0204

F4200: FF 255 RES  
F4201: FF 255 RES  
F4202: CD 205 =  
F4203: 21 033 ?  
F4204: CF 207 ±  
F4205: 00 000 NULL  
F4206: 00 000 NULL  
F4207: 00 000 NULL  
F4208: 00 000 NULL  
F4209: 00 000 NULL  
F420A: 00 000 NULL  
F420B: 00 000 NULL  
F420C: 00 000 NULL  
F420D: 00 000 NULL  
F420E: 00 000 NULL  
F420F: 00 000 NULL  
F4210: 00 000 NULL  
F4211: 00 000 NULL  
F4212: 00 000 NULL  
F4213: 00 000 NULL  
F4214: 00 000 NULL  
F4215: 00 000 NULL

F400:0204

BIOS DI  
INT 021h  
IRET  
ADD [BX + SI], AL  
ADD [BX + SI], AL  
ADD [BX + SI], AL  
ADD [BX + SI], AL  
ADD [BX + SI], AL  
ADD [BX + SI], AL  
ADD [BX + SI], AL  
ADD [BX + SI], AL  
ADD [BX + SI], AL  
ADD [BX + SI], AL  
ADD [BX + SI], AL  
ADD [BX + SI], AL  
ADD [BX + SI], AL  
ADD [BX + SI], AL  
ADD [BX + SI], AL  
...

screen source reset aux vars debug stack flags

## Before running code

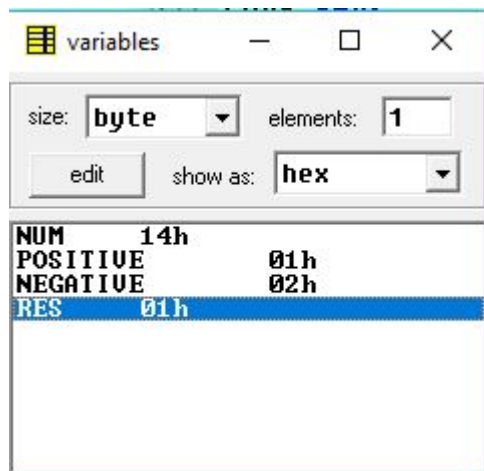
variables

size: byte elements: 1

edit show as: hex

NUM	14h
POSITIVE	01h
NEGATIVE	02h
RES	00h

After running code



NUM	14h
POSITIVE	01h
NEGATIVE	02h
RES	01h

## Result

Executed program for Bit manipulation to check whether the data is positive or negative in emu8086 and obtained the outputs