Experiment 04 : Finding Negative Numbers from given array

Learning Objective: Student should be able to Identify negative numbers from a given sign array using Assembly language.

Tools: TASM/MASM

Theory:

In computing, signed number representations are required to encode negative numbers in binary number systems. The early days of digital computing were marked by a lot of competing ideas about both hardware technology and mathematics technology (numbering systems). One of the great debates was the format of negative numbers, with some of the era's most expert people having very strong and different opinions. One camp supported two's complement, the system that is dominant today. Another camp supported ones' complement, where any positive value is made into its negative equivalent by inverting all of the bits in a word. A third group supported "sign & magnitude" (sign–magnitude), where a value is changed from positive to negative simply by toggling the word's sign (high–order) bit. In mathematics, negative numbers in any base are represented by prefixing them with a − sign. However, in computer hardware, numbers are represented in bit vectors only, without extra symbols. The four best–known methods of extending the binary numeral system to represent signed numbers are: sign–and–magnitude, ones' complement, two's complement, and excess–K. Some of the alternative methods use implicit instead of explicit signs, such as negative binary, using the base −2. Corresponding methods can be devised for other bases, whether

positive, negative, fractional, or other elaborations on such themes. There is no definitive criterion by which any of the representations is universally superior. The representation used in most current computing devices is two's complement. To check whether the number is negative, perform AND operation with 80H, if the number is negative, the result will be Non-zero (i.e. MSB is 1) and if number is positive then the result will be Zero (i.e. MSB is 0).

5. Procedure/Algorithm:

Algorithm:

1. Initialize index register with the offset of array of signed numbers
2. Initialize CX with array element count
3. Initialize negative number count to zero
4. Perform MSB test of array element
5. If set jump to step 6
6. Increment negative number count and continue
8. Point index register to the next element
9. Decrement the array element count from CX, if not zero jump to step 4, else continue
10. Display Negative number message and then display negative number count
12. EXIT

Application: Use of array in the Assembly Language programming to write modular program.

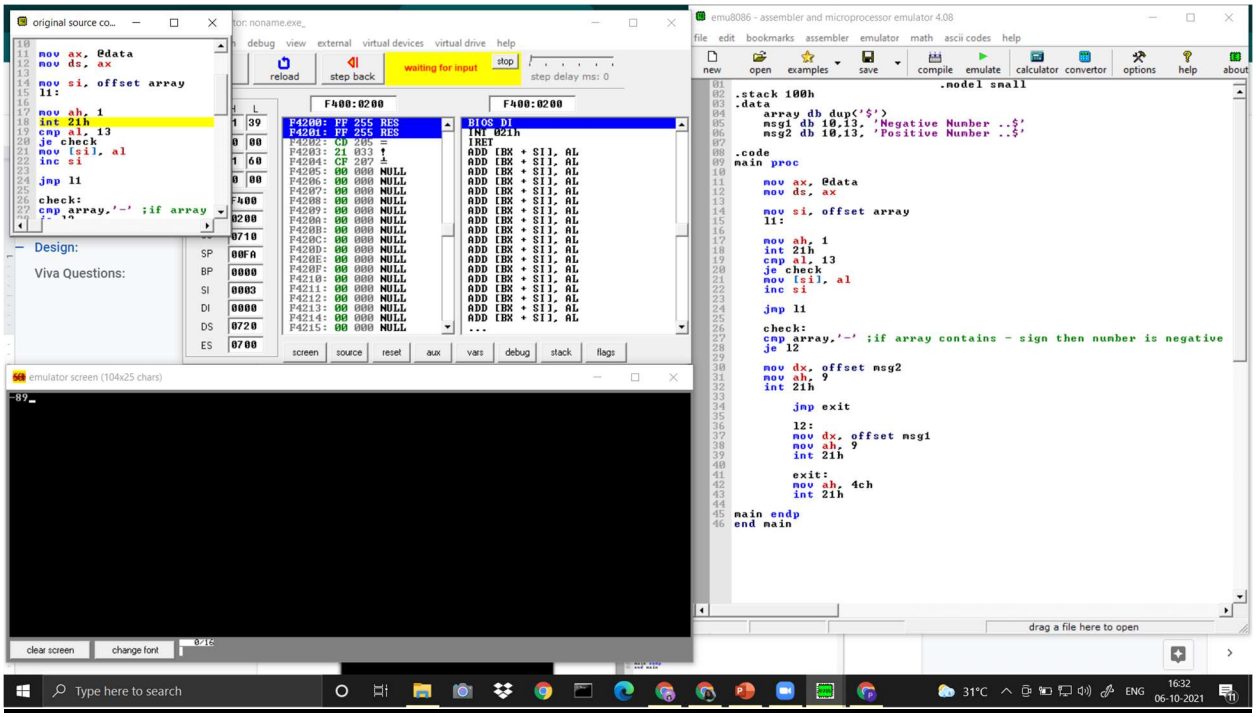
Design:

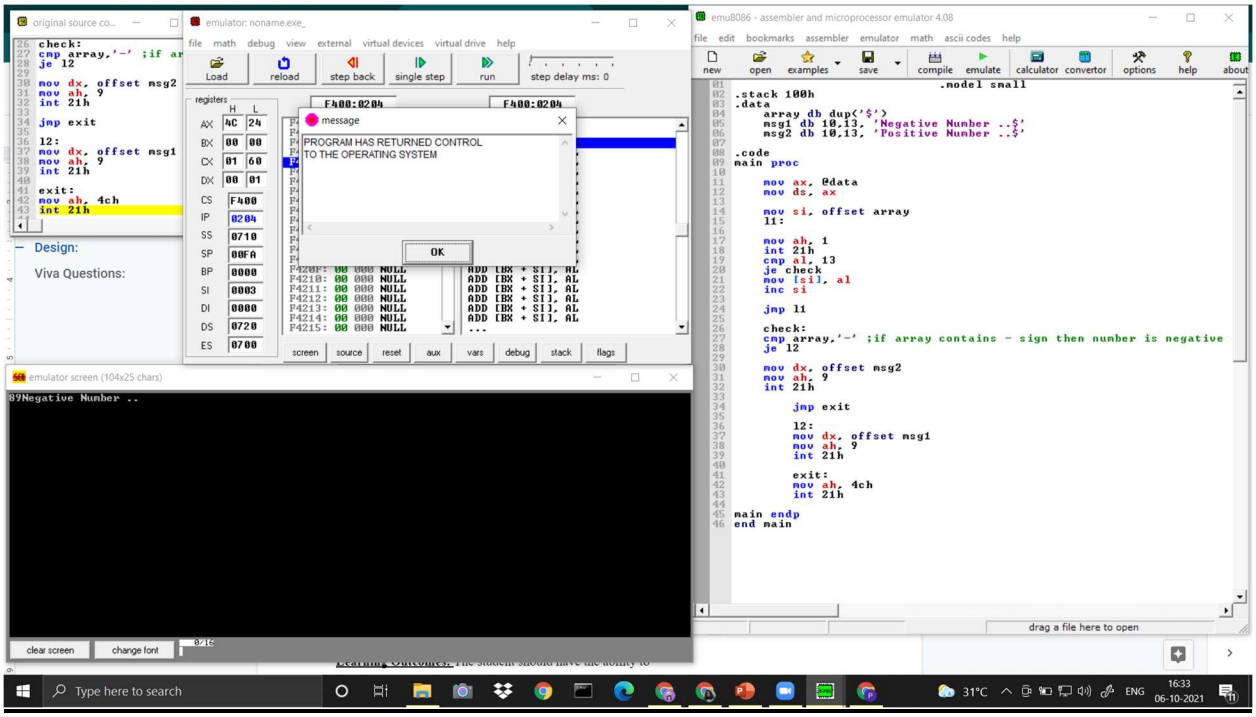
```
                .model small
.stack 100h
.data
   array db dup('$')
   msg1 db 10,13, 'Negative Number ..$'
```

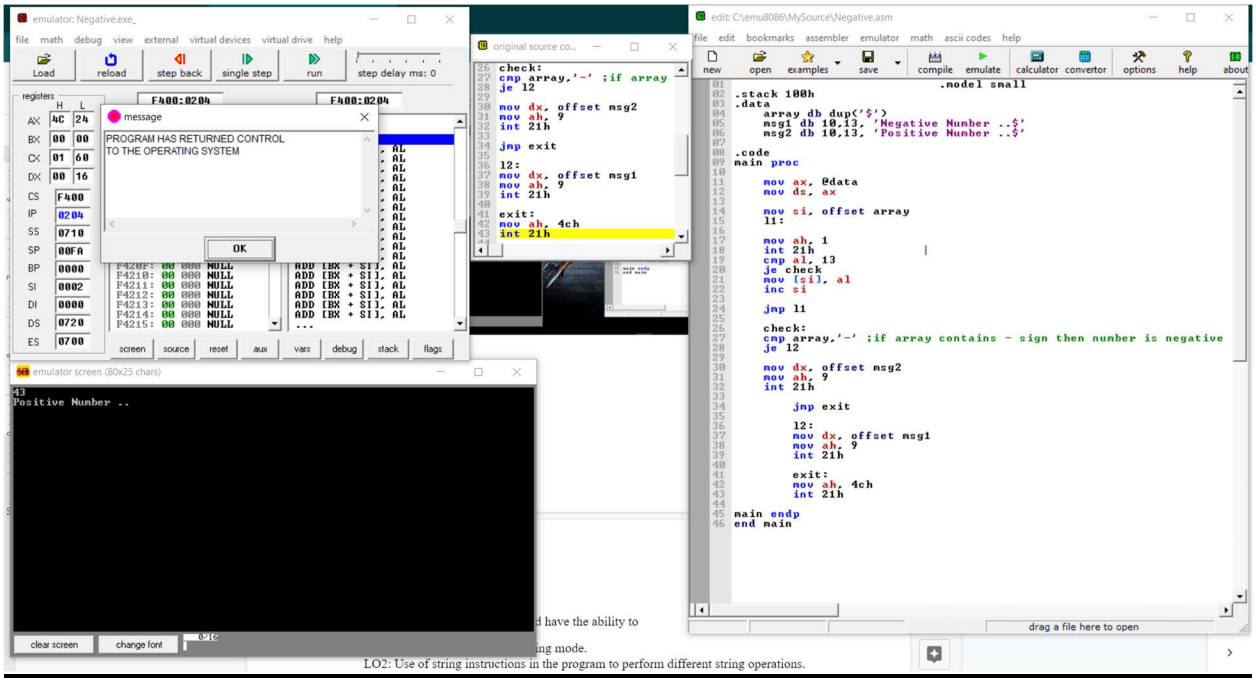
```
    msg2 db 10,13, 'Positive Number ..$'

.code
main proc

    mov ax, @data
    mov ds, ax

    mov si, offset array
    l1:

    mov ah, 1
    int 21h
    cmp al, 13
    je check
    mov [si], al
    inc si

    jmp l1

    check:
    cmp array,'-' ;if array contains - sign then number is negative
    je l2
```

```
    mov dx, offset msg2
    mov ah, 9
    int 21h

       jmp exit

       l2:
       mov dx, offset msg1
       mov ah, 9
       int 21h

       exit:
       mov ah, 4ch
       int 21h

main endp
end main
```

Result and Discussion:




emulator: noname.exe_
message
PROGRAM HAS RETURNED CONTROL
TO THE OPERATING SYSTEM
OK
emulator screen (104x25 chars)
89Negative Number ..
emu8086 - assembler and microprocessor emulator 4.08
.stack 100h
.data
array db dup('$')
msg1 db 10,13, 'Negative Number ..$'
msg2 db 10,13, 'Positive Number ..$'
.code
main proc
mov ax, @data
mov ds, ax
mov si, offset array
l1:
mov ah, 1
int 21h
cmp al, 13
je check
mov [si], al
inc si
jmp l1
check:
cmp array,'-' ;if array contains - sign then number is negative
je l2
mov dx, offset msg2
mov ah, 9
int 21h
jmp exit
l2:
mov dx, offset msg1
mov ah, 9
int 21h
exit:
mov ah, 4ch
int 21h
main endp
end main
drag a file here to open
Design:
Viva Questions:


emulator: Negative.exe_
message
PROGRAM HAS RETURNED CONTROL
TO THE OPERATING SYSTEM
OK
emulator screen (80x25 chars)
43
Positive Number ..
edit: C:\emu8086\MySource\Negative.asm
drag a file here to open
LO2: Use of string instructions in the program to perform different string operations.

Learning Outcomes: The student should have the ability to

LO1: Describe the string addressing mode.
LO2: Use of string instructions in the program to perform different string operations.
LO3: Write a program using array to find negative numbers from given signed array.

Course Outcomes: Upon completion of the course students will be able to make use of instructions of 8086 to build assembly and Mixed language programs.

Conclusion: In this experiment, we learned how to determine whether a number is positive or negative using the assembly level language in an 8086 microprocessor.

Viva Questions:

1. How negative Numbers are represented in Computer System?
2. Explain how array is defined in ALP.
3. Explain the procedure to find negative numbers from given array.

For Faculty Use

Correction Parameters	**Formative Assessment [40%]**	**Timely completion of Practical [40%]**	**Attendance / Learning Attitude [20%]**	
Marks Obtained				