

**Name : Khushi Nitinkumar Patel**

**PRN : 2020BTECS00037**

**Experiment 13 :** Write X86/64 ALP to count number of positive and negative numbers from the array.

### **Algorithm**

1. Move the contents present at 00H to BL register .
2. Move the contents of 0AH into DL register.
3. LEA (Load effective address) will sets SI to 2000H memory address.
4. Move the contents at SI to AL register .
5. SHL instruction performs logical shift on the destination.
6. Jump if there is no carry else increment value at BL.
7. Jump loop 2 .In loop 1 , increment in CL by 1 and in loop 2 , increment in SI by 1 .
8. Decrement by 1 unit in content of DL register .

9. Jump when not zero and move the contents present at CL to memory location 1234H .

10. Move the content of BL register to 1235H memory location .

11. Stop .

### **Source Code :**

```
MOV BL,00H
```

```
MOV CL,00H
```

```
MOV DL,0AH
```

```
LEA SI,[2000H]
```

```
LOOP:MOV AL, [SI]
```

```
SHL AL, 01
```

```
JNC L1
```

```
INC BL
```

```
JMP L2
```

L1:INC CL

L2:INC SI

DEC DL

JNZ LOOP

MOV [1234H], CL

MOV [1235H], BL

HLT

## Snap shots

The screenshot displays the emu8086 interface with three main windows:

- Random Access Memory:** Shows a memory dump starting at address 0100:2000. The data is organized in a grid with columns for addresses and hex values. The address 0100:0036 is highlighted.
- Registers:** A table showing the state of various registers. The SI register is highlighted with a value of 0000. The DS register is highlighted with a value of 0100.
- Source Code:** A window showing the assembly code being executed. The code includes instructions like MOV, INC, DEC, JNZ, and HLT. The instruction MOV [1235H], BL is highlighted in yellow.

The assembly code in the Source Code window is as follows:

```
01 MOV BL,00H
02 MOV CL,00H
03 MOV DL,00H
04 LEA SI,[2000H]
05 LOOP:MOV AL,[SI]
06 SHL AL,01
07
08 JNC L1
09 INC BL
10 JMP L2
11
12 L1:INC CL
13 L2:INC SI
14 DEC DL
15 JNZ LOOP
16 MOV [1234H],CL
17 MOV [1235H],BL
18
19
20
```

**Conclusion :** Performed count of positive and negative numbers from the array using 8086 emulator .