Roll No: 11

Name: Soham Desai Xavier ID: 202003021

Date: 23/3/22

EXPERIMENT 7

Aim: Program to move set of numbers from one memory block to another.

LO: 4

LO STATEMENT: Develop the assembly level programming using 8086 loop

instruction set

Software and Hardware Requirements: TASM Software

Theory:

1. MOV Instruction

The MOV instruction is the most important command in the 8086 because it moves data from one location to another. It also has the widest variety of parameters; so the assembler programmer can use MOV effectively, the rest of the commands are easier to understand. MOV copies the data in the source to the destination. The data can be either a byte or a word. Sometimes this has to be explicitly stated when the assembler cannot determine from the operands whether a byte or word is being referenced.

Syntax:

Move Destination, Source

Example:

MOV Ax, Bx

2. MOVSB Instruction

MOVSB instruction will perform all the actions in repeat unit loop. The MOVSB instruction will copy a byte from the location pointed to by the Direct Index Register.It will then automatically increment SI to point to the next source location. If you add a special prefix called the repeat prefix in front of the MOVSB instruction, the MOVSB instruction will be repeated and CX decremented until CX is counted down to zero.

Syntax:

Move Destination, Source

Example:

MOVSB Ax, Bx

3. LEA Instruction

LEA is Used to load the address of operand into the provided register. LES – Used to load ES register and other provided register from the memory. The lea instruction places the address specified by its first operand into the register specified by its second operand. Note, the contents of the memory location are not loaded, only the effective address is computed and placed into the register.

Roll No: 11

Name: Soham Desai Xavier ID: 202003021

Date: 23/3/22

Code:

```
assume cs:code,ds:code,es:extra
```

data segment

blk1 db 10H,20H,30H,40H,50H

data ends

extra segment

blk2 db 05 dup(?)

extra ends

code segment

start:

mov Ax,data

mov Ds,Ax

mov Ax, extra

mov es,Ax

lea si,blk1

lea di,blk2

mov Cx,05H

std

back:movsb

loop back

mov AH,4CH

int 21H

code ends

end start

Roll No: 11

Name: Soham Desai Xavier ID: 202003021

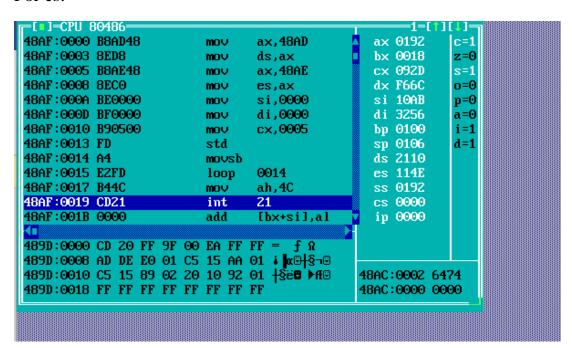
Date: 23/3/22

Output:

For ds:

```
[ ]=CPU 80486
                                                   ax 0192
48AF:0000 B8AD48
                                  ax,48AD
                                                              c=1
                          MOV
48AF:0003 8ED8
                                 ds,ax
                                                   bx 0018
                                                              z=0
                          MOV
48AF:0005 B8AE48
                                                   cx 092D
                                  ax,48AE
                                                              s=1
                                                  d× F66C
                                                              0=0
48AF:0008 8ECO
                                 es,ax
                          MOU
48AF:000A BE0000
                                 si,0000
                                                  si 10AB
                          MOV
                                                              p=0
48AF:000D BF0000
                                                  di 3256
                                                              a=0
                          MOV
                                 di,0000
                                 cx,0005
                                                  bp 0100
48AF:0010 B90500
                                                              i=1
                          MOV
48AF:0013 FD
                                                   sp 0106
                                                              d=1
                          std
                                                  ds 2110
48AF:0014 A4
                          movsb
48AF:0015 E2FD
                                 0014
                                                   es 114E
                          loop
                                                  ss 0192
48AF:0017 B44C
                          MOV
                                  ah,4C
48AF:0019 CD21
                                                  cs 0000
                          int
                                 21
                                 [bx+si],al
                                                   ip 0000
48AF:001B 0000
                          add
48AD:0000 10 20 30 40 50 00 00 00 ▶ 0@P
48AD:0008 00 00 00 00 00 00 00 00
48AD:0010 10 00 00 00 00 00 00 0 ▶
                                                 48AC<mark>:</mark>0002 6474
48AD:0018 00 00 00 00 00 00 00 00
                                                 48AC:0000 0000
```

For es:



Conclusion: From this experiment we learn how to move numbers from one memory block to another.