Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

A.Y. 2022-2023

PROCESSOR ORGANIZATION AND ARCHITECTURE

AYUSH JAIN

COMPUTER ENGINEERING | TE – B2 | 60004200132

EXPERIMENT – 7

Aim: Assembly Program to transfer n blocks of data from one segment to another segment

Theory:

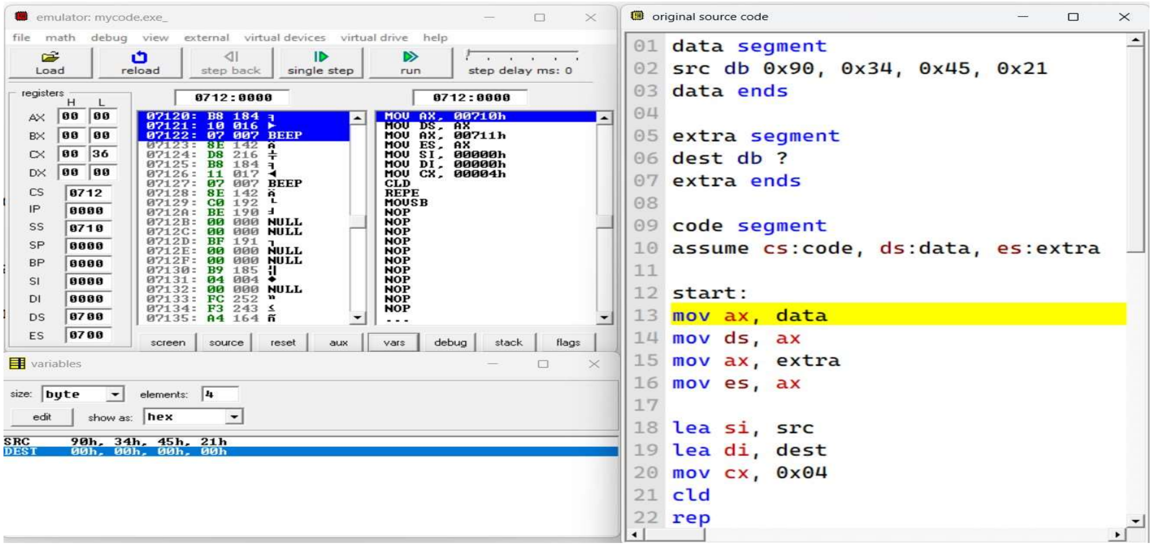
String Instructions in 8086-
String is a series of data bytes or words available in memory at consecutive locations. It is either referred to as a byte string or a word string. Their memory is always allocated in a sequential order. Instructions used to manipulate strings are called string manipulation instructions.

Code:

```
data segment src db 0x90,
0x34, 0x45, 0x21
data ends

extra segment
dest db ? extra
ends

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

```
start: mov ax,
   data mov
   ds, ax mov
   ax, extra
   mov es, ax

   lea si, src
   lea di, dest
   mov cx,
   0x04 cld
   rep movsb

end start
code ends
```

Output:



emulator: mycode.exe_
file math debug view external virtual devices virtual drive help
Load
reload
step back
single step
run
step delay ms: 0
screen source reset aux vars debug stack flags
variables
edit
original source code

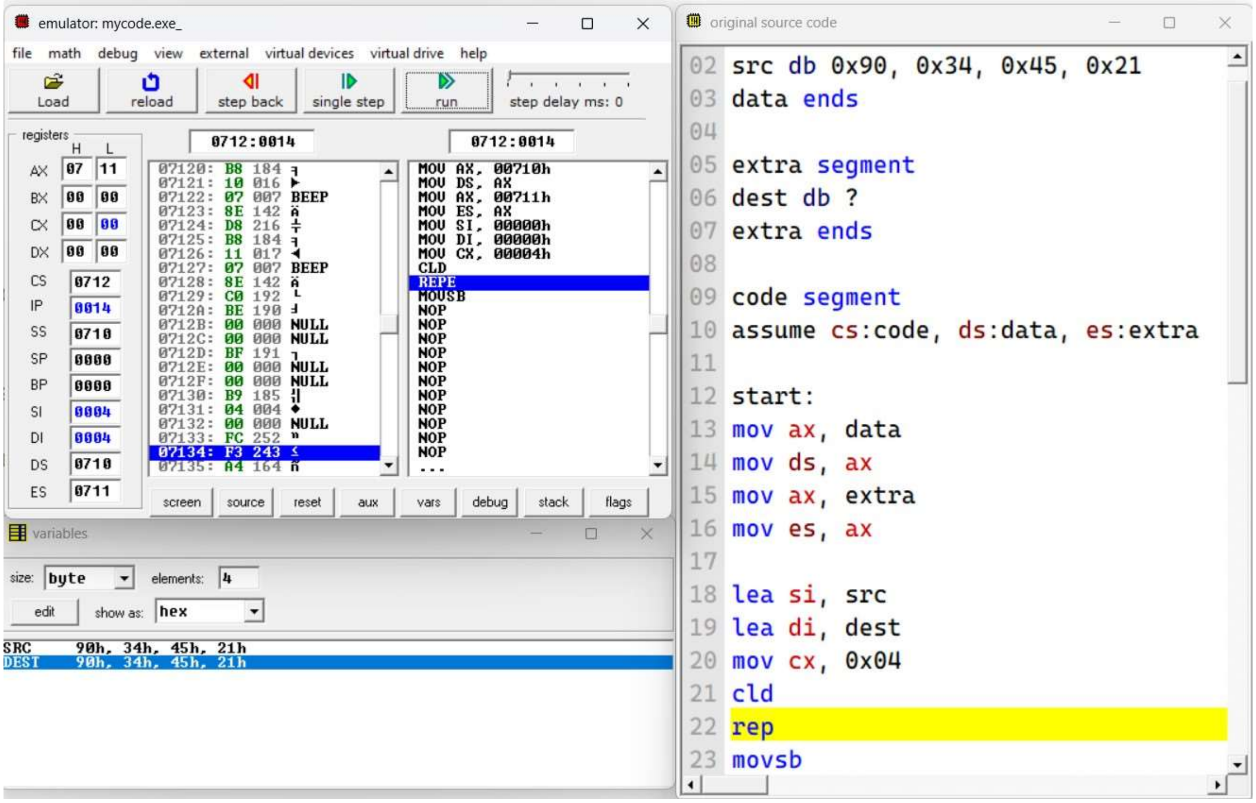
```
registers
     H   L
AX   00  00
BX   00  00
CX   00  36
DX   00  00
CS   0712
IP   0000
SS   0710
SP   0000
BP   0000
SI   0000
DI   0000
DS   0700
ES   0700
```

```
0712:0000
07120: B8 184 ╕
07121: 10 016 ►
07122: 07 007 BEEP
07123: 8E 142 Ä
07124: D8 216 ╪
07125: B8 184 ╕
07126: 11 017 ◄
07127: 07 007 BEEP
07128: 8E 142 Ä
07129: C0 192 └
0712A: BE 190 ╛
0712B: 00 000 NULL
0712C: 00 000 NULL
0712D: BF 191 ┐
0712E: 00 000 NULL
0712F: 00 000 NULL
07130: B9 185 ╣
07131: 04 004 ♦
07132: 00 000 NULL
07133: FC 252 ⁿ
07134: F3 243 ≤
07135: A4 164 ñ
```

```
0712:0000
MOV AX, 00710h
MOV DS, AX
MOV AX, 00711h
MOV ES, AX
MOV SI, 00000h
MOV DI, 00000h
MOV CX, 00004h
CLD
REPE
MOVSB
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

```
size: byte    elements: 4
show as: hex
SRC     90h, 34h, 45h, 21h
DEST    00h, 00h, 00h, 00h
```

```
data segment
src db 0x90, 0x34, 0x45, 0x21
data ends

extra segment
dest db ?
extra ends

code segment
assume cs:code, ds:data, es:extra

start:
mov ax, data
mov ds, ax
mov ax, extra
mov es, ax

lea si, src
lea di, dest
mov cx, 0x04
cld
rep
```



```
registers
      H    L
AX    07   11
BX    00   00
CX    00   00
DX    00   00
CS    0712
IP    0014
SS    0710
SP    0000
BP    0000
SI    0004
DI    0004
DS    0710
ES    0711
```

```
0712:0014

07120: B8 184 ╕
07121: 10 016 ►
07122: 07 007 BEEP
07123: 8E 142 Ä
07124: D8 216 ╪
07125: B8 184 ╕
07126: 11 017 ◄
07127: 07 007 BEEP
07128: 8E 142 Ä
07129: C0 192 └
0712A: BE 190 ╛
0712B: 00 000 NULL
0712C: 00 000 NULL
0712D: BF 191 ┐
0712E: 00 000 NULL
0712F: 00 000 NULL
07130: B9 185 ╣
07131: 04 004 ♦
07132: 00 000 NULL
07133: FC 252 ⁿ
07134: F3 243 ≤
07135: A4 164 ñ
```

```
0712:0014

MOV AX, 00710h
MOV DS, AX
MOV AX, 00711h
MOV ES, AX
MOV SI, 00000h
MOV DI, 00000h
MOV CX, 00004h
CLD
REPE
MOVSB
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

size: byte   elements: 4   show as: hex

```
SRC     90h, 34h, 45h, 21h
DEST    90h, 34h, 45h, 21h
```

```
src db 0x90, 0x34, 0x45, 0x21
data ends

extra segment
dest db ?
extra ends

code segment
assume cs:code, ds:data, es:extra

start:
mov ax, data
mov ds, ax
mov ax, extra
mov es, ax

lea si, src
lea di, dest
mov cx, 0x04
cld
rep
movsb
```

Conclusion:

Hence, we implemented string instructions to transfer n blocks of data