

Name : Kreena Shah
Div / Batch : C32
Sapid : 60004210243
Subject : Processor Organization and Architecture (POA)

Experiment 7

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

Theory : Transferring data between memory segments is a common operation in low-level programming. This experiment focuses on writing an assembly program to efficiently copy a specified number of data blocks (n blocks) from one segment to another.

Code :

```
DATA SEGMENT
    N1 DB 45H, 32H, 36H
DATA ENDS
```

```
EXTRA SEGMENT
    N2 DB ?
EXTRA ENDS
```

```
CODE SEGMENT
    START:
```

```
    MOV AX,DATA
    MOV DS,AX
```

```
    MOV AX,EXTRA
    MOV ES,AX
```

```
    LEA SI, N1
    LEA DI, N2
```

```
    MOV CL,04H
```

```
    L1:
```

```
    MOV AX,DS:[SI]
    MOV ES:[DI],AX
```

```
    INC SI
    INC DI
    DEC CL
```

```
    JNZ L1
```

```
CODE ENDS
END START
```

emu8086 - assembler and microprocessor emulator 4.08

file edit bookmarks assembler emulator math ascii codes help

new open examples save compile emulate calculator converter options help about

```

01 DATA SEGMENT
02 N1 DB 63H, 72H, 18H
03 DATA ENDS
04
05 EXTRA SEGMENT
06 M2 DB ?
07 EXTRA ENDS
08
09
10 CODE SEGMENT
11 START:
12
13 MOV AX,DATA
14 MOV DS,AX
15
16
17 MOV AX,EXTRA
18 MOV ES,AX
19
20 LEA SI, N1
21 LEA DI, N2
22
23 MOV CL,04H
24
25 LI:
26 MOV AX,DS:[SI]
27 MOV ES:[DI],AX
28
29 INC SI
30 INC DI
31 DEC CL
32
33 JNZ LI
34
35 CODE ENDS
36 END START

```

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	00	00
BX	00	00
CX	00	00
DX	00	00
CS	0712	
IP	0032	
SS	0710	
SP	0000	
BP	0000	
SI	0000	
DI	0000	
DS	0710	
ES	0711	

0712:0032

0714B: 90 144 E	NOP
07149: 90 144 E	NOP
07140: 90 144 E	NOP
0714B: 90 144 E	NOP
0714C: 90 144 E	NOP
0714D: 90 144 E	NOP
0714E: 90 144 E	NOP
0714F: 90 144 E	NOP
07150: 90 144 E	NOP
07151: 90 144 E	NOP
07152: F4 244	INC
07153: 00 000 NULL	NOP
07154: 00 000 NULL	NOP
07155: 00 000 NULL	NOP
07156: 00 000 NULL	NOP
07157: 00 000 NULL	ADD [BX + SI], AL
07158: 00 000 NULL	ADD [BX + SI], AL
07159: 00 000 NULL	ADD [BX + SI], AL
0715A: 00 000 NULL	ADD [BX + SI], AL
0715B: 00 000 NULL	ADD [BX + SI], AL
0715C: 00 000 NULL	...
0715D: 00 000 NULL	

screen source reset aux vars debug stack flags

variables

size: byte elements: 1

edit show as: hex

NI 63h
NZ 63h

original source

```

19 LEA SI, N1
20 LEA DI, N2
21
22 MOV CL,04H
23
24 LI:
25
26 MOV AX,DS:[SI]
27 MOV ES:[DI],AX
28
29 INC SI
30 INC DI
31 DEC CL
32
33 JNZ LI
34

```

stack

0710:0020	00BE
0710:0028	C0BE
0710:0026	0711
0710:0024	0B08
0710:0022	0B07
0710:0020	10B8
0710:001E	0000
0710:001C	0000
0710:001A	0000
0710:0018	0000
0710:0016	0000
0710:0014	0000
0710:0012	0018
0710:0010	7263
0710:000E	0000
0710:000C	0000
0710:000A	0000
0710:0008	0000
0710:0006	0000
0710:0004	0000
0710:0002	0018
0710:0000	7263 <

line: 2 col: 23 drag a file here to open

31°C Smoke

ENG IN

18:23 04-11-2023