

ANUSHKA GUPTA
19075088
B.TECH CSE

LAB ASSIGNMENT 5

TASK : Write a Program to Move a Block of Data Using 8085 & verify

Algorithm:

- 1) Initialize count in C as 6. This is the size of our data block
- 2) Initialize H-L as source memory pointer
- 3) Initialize D-E as the destination memory pointer
- 4) While the count is not equal to zero :
 - a) Get the byte from the source memory block
 - b) Store the byte in the destination memory block
 - c) Increment source memory pointer
 - d) Increment destination memory pointer
 - e) Decrement count

Code:

```
; <Program title>
jmp start
; data
; code
start: nop
MVI D,06
LXI H, 004H
LXI B, 00CH
up: MOV A,M
STAX B
INX H
INX B
DCR D
JNZ up
hlt
```

Before execution:

Data
Stack
Keypad
Memory
I/O Ports

Start
OK

Address (Hex)	Address	Data
0003	3	0
0004	4	2
0005	5	4
0006	6	6
0007	7	8
0008	8	10
0009	9	12
000A	10	0
000B	11	0
000C	12	0
000D	13	0
000E	14	0

Line No	Assembler Message
0	Program assembled successfully









After execution:

Data
Stack
Keypad
Memory
I/O Ports

Start
OK

Address (Hex)	Address	Data
000A	10	0
000B	11	0
000C	12	2
000D	13	4
000E	14	6
000F	15	8
0010	16	10
0011	17	12
0012	18	0
0013	19	0
0014	20	0
0015	21	0

Line No	Assembler Message
0	Program assembled successfully

File Reset Assembler Debug H				
       				
Registers			Flag	
<i>A</i>	0C		<i>S</i>	0
<i>BC</i>	00	12	<i>Z</i>	1
<i>DE</i>	00	00	<i>AC</i>	0
<i>HL</i>	00	0A	<i>P</i>	1
<i>PSW</i>	00	00	<i>C</i>	0
<i>PC</i>	42	15		
<i>SP</i>	FF	FF		
<i>Int-Reg</i>	00			

Input: (address in hex format)

0004 -> 2
 0005 -> 4
 0006 -> 6
 0007 -> 8
 0008 -> 10
 0009 -> 12

Output: (address in hex format)

000C -> 2
 000D -> 4
 000E -> 6
 000F -> 8
 0010 -> 10
 0011 -> 12

Verification: Our source address was 0004 and destination was 000C . The values stored in memory locations from 000C to 0011 are the same as the values stored in 0004 to 0009. Hence verified that moving of block data is successful.