**Microprocessor Lab  
Lab Experiment No. 10**

Name: Ninad Rao Roll No. 53

**Aim**: Perform block transfer from one location to other without overlap.

**Instructions on how to use TASM**:

Steps for creating the program:

1. TASM is loaded
2. TASM < Edit - We will get an edit window
3. Type the program here
4. Save the file as <filename>.asm

Steps for running the program:

1. c:\tasm> Type here tasm filename

**c:\tasm> tasm <filename>.asm**

This will save the program, and the edit window with this file name will be seen.

1. c:\tasm> Linking the program

**c:\tasm> tlink <filename>.obj**

This will create an object file after linking.

1. c:\tasm> Now to execute the program and get to the result window

**c:\tasm> td <filename>.exe**

After execution, all the window options are present to check all registers, all memory locations and so on.

**Program to block transfer without overlap**:

**Explanation**: Here we are initially setting up the source index register with the source of data blocks, then set the destination index register to store into another block. Then set the Data segment register and Extra Segment register to 0000H. By using MOVSB instruction, the entire block is transferred from one location to another. As the size is 4-byte, we have set the counter register (CX) as 04H. Until the CX register turns to 0, the data will be shifted.

**Algorithm**:

**Step I:** Define block of data.

**Step II:** Save memory for block transfer as block2.

**Step III:** Load block1 into SI.

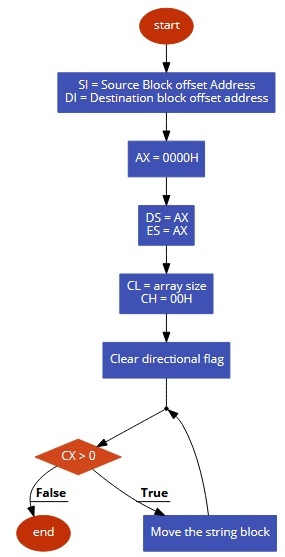
**Step IV:** Load block2 into DI.

**Step V:** Initialize counter.

**Step VI:** Move first data into DI.

**Step VII:** Repeat step 6 until the counter is zero.

**Flowchart**:



**Code**:

.model small

.data

src\_blk db 01, 02, 03, 04, 05, 06, 07, 08, 09, 0Ah

dest\_blk db 10 dup(?)

count dw 0Ah

.code

mov ax, @data ; Data inter defined

mov ds, ax ; Data segment initiation

mov es, ax ; Extra segment initiation

mov si, offset src\_blk ; Source pointer

mov di, offset dest\_blk ; Destination pointer

mov cx, count ; Counter CX initialized for 10 transfers

cld ; Auto-increment mode

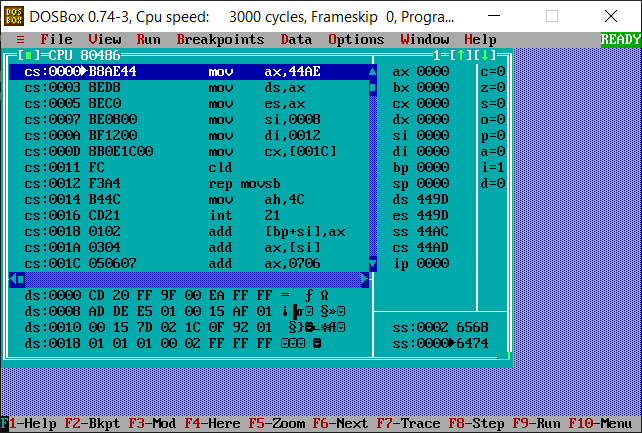
again: rep movsb ; Repeat till 10 transfers

mov ah, 4ch ; Terminate the program

int 21h

end

**Output**:



**Conclusion**: Thus, we have studied and understood the program to block transfer from one location to other without overlap.