



Microprocessors and Interfaces: 2021-22
Lecture 9
8086 Instructions Set : Part-3

By Dr. Sanjay Vidhyadharan

Data Transfer Instructions

- General Purpose Data Transfer
(MOV, XCHG, XLAT, PUSH, POP)
- Input / Output Data Transfer
(IN, OUT)
- Address Object Data Transfer
(LEA, LDS, LES)
- Flag Transfer Data Transfer
(LAHF, SAHF, PUSHF, POPF)

STRING DATA TRANSFERS

- String Instructions allow multiple data transfer operations on series of data residing in consecutive memory locations.
- The data transfers can be single byte, word, or double word.

We can use them to perform memory-to-memory block transfers.

STRING DATA TRANSFERS

- String Operations:

mnemonic	meaning	operand(s) required
LODS	LOaD String	source
STOS	STORe String	destination
MOVS	MOVE String	source & destination
CMPS	CoMPare Strings	source & destination
SCAS	SCAn String	destination

Source and Destination Address

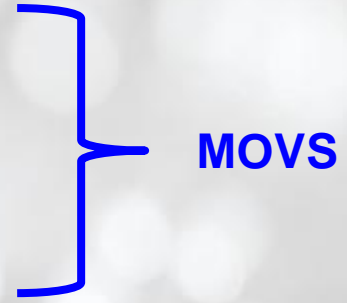
- Source : Memory given by DS:SI
- Destination : Memory given by ES:DI

SI and DI is auto incremented/decremented as per D flag

- String Operations used along with rep command for Bulk Transfer

String Operations used along with rep: C is used as Counter. Loaded with the required Count. Loop is stopped when C becomes zero (Detected by Zero flag).

- Operating in 32-bit mode EDI and ESI registers are used in place of DI and SI.



The Direction Flag

Flag Register



The direction of string operations depends on the value of the direction flag.

If the direction flag (DF) is clear (i.e., $DF = 0$), string operations proceed in the auto-increment mode.

CLD

(clear direction flag) Resets $DF = 0$

If the direction flag (DF) is Preset (i.e., $DF = 1$), string operations proceed in the auto-decrement mode.

STD

(SET direction flag) Sets $DF = 1$

Both of these instructions do not require any operands. Each instruction is encoded using a single byte and takes two clock cycles to execute.

MOVS

- During MOVS Flags are not
- MOVSB : SI and DI auto increment or decrement by 1
- MOVSW: SI and DI auto increment or decrement by 2
- MOVSD: SI and DI increment or decrement by 4

MOVS with REP

- C is auto decremented by 1

COPY A BLOCK OF 100 Bytes of DATA FROM ONE MEMORY AREA TO ANOTHER MEMORY AREA

Using MOV

.data

Array1 db 0ah,bch,deh,0f5h,11h, 56h,78h,0ffh,0ffh ,23h4ah, ...

Array2 db 100 dup(0)

.code

startup

MOV CX, 64H

LEA SI, Array1

LEA DI, Array2

Next: MOV AL, [SI]

MOV [DI], AL

INC SI

INC DI

LOOP Next

EXIT

END

COPY A BLOCK OF 100 Bytes of DATA FROM ONE MEMORY AREA TO ANOTHER MEMORY AREA

Using MOVSB

.data

Array1 db 0ah,bch,deh,0f5h,11h, 56h,78h,0ffh,0ffh ,23h4ah, ...

Array2 db 100 dup(0)

.code

startup

MOV CX, 64H

LEA SI, Array1

LEA DI, Array2

CLD

Next: MOVSB

LOOP Next

EXIT

END

COPY A BLOCK OF 100 Bytes of DATA FROM ONE MEMORY AREA TO ANOTHER MEMORY AREA

Using MOVSB with Rep

.data

Array1 db 0ah,bch,deh,0f5h,11h, 56h,78h,0ffh,0ffh ,23h4ah, ...

Array2 db 100 dup(0)

.code

startup

MOV CX, 64H

LEA SI, Array1

LEA DI, Array2

CLD

REP MOVSB

EXIT

END

COPY A BLOCK OF 100 Bytes of DATA FROM ONE MEMORY AREA TO ANOTHER MEMORY AREA

Using MOVSW with Rep

.data

Array1 db 0ah,bch,deh,0f5h,11h, 56h,78h,0ffh,0ffh ,23h4ah, ...

Array2 db 100 dup(0)

.code

startup

MOV CX, 32H

LEA SI, Array1

LEA DI, Array2

CLD

REP MOVSW

EXIT

END

COPY A BLOCK OF 100 Bytes of DATA FROM ONE MEMORY AREA TO ANOTHER MEMORY AREA

Using MOVSB with Rep in decreasing mode

.data

Array1 db 0ah,bch,deh,0f5h,11h, 56h,78h,0ffh,0ffh ,23h4ah, ...

Array2 db 100 dup(0)

.code

startup

MOV CX, 64H

LEA SI, Array1+ 63H

LEA DI, Array2+63H

STD

REP MOVSB

EXIT

END

COPY A BLOCK OF 100 Bytes of DATA FROM ONE MEMORY AREA TO ANOTHER MEMORY AREA in Reverse Order

Using MOVSB with Rep in decreasing mode

.data

Array1 db 0ah,bch,deh,0f5h,11h, 56h,78h,0ffh,0ffh ,23h4ah, ...

Array2 db 100 dup(0)

.code

startup

MOV CX, 64H

LEA SI, Array1

LEA DI, Array2+63H

CLD

Next: MOVSB

SUB DI, 2

LOOP Next

EXIT

END

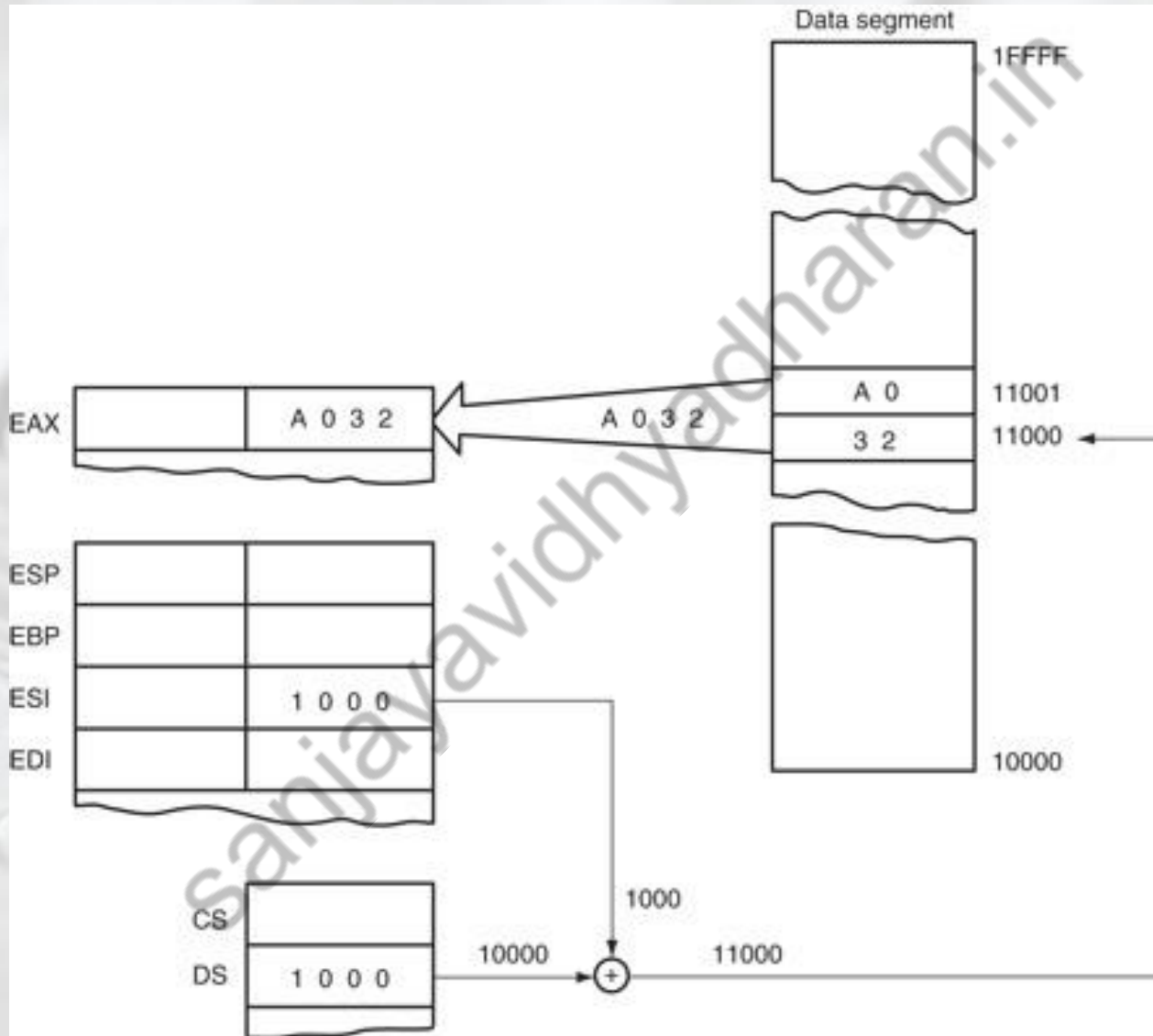
LODS/LODSB/LODSW/LODSD

Source : Memory given by DS:SI

Destination : AL or AX or EAX

- LODSB : SI auto increment or decrement by 1
- LODSW: SI auto increment or decrement by 2
- LODSD: SI auto increment or decrement by 4
- Based on D Flag
- FLAGS are not Affected by LODS

Example of LODSW Instruction



STOS /STOSB/STOSW

Source : AL or AX or EAX

Destination : Memory given by ES:DI

- STOSB : DI auto increment or decrement by 1
- STOSW: DI auto increment or decrement by 2
- STOSD: DI auto increment or decrement by 4
- Based on D Flag
- FLAGS are not Affected by STOS

Write an ALP to fill a set of 100 memory locations starting at displacement 'DAT1' with the value F6H

```
.DATA  
DAT1      DB    100 DUP(?)  
.CODE  
.STARTUP  
MOV DI, OFFSET DAT1  
MOV AL, 0F6H  
MOV CX, 64H  
CLD  
REP STOSB  
.EXIT  
END
```

Thank you