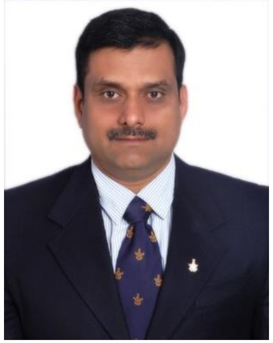
Microprocessors and Interfaces: 2021-22
Lecture 10
8086 Instructions Set : Part-4

By Dr. Sanjay Vidhyadharan



Data Transfer Instructions

MOVS with a REP

- The repeat prefix (REP) is added to any string data transfer instruction except LODS.
  - REP prefix causes CX to decrement by 1 each time the string instruction executes; after CX decrements, the string instruction repeats

- If CX reaches a value of 0, the instruction terminates and the program continues.

EX: If CX is loaded with 100 and a REP MOVSB instruction executes, the microprocessor automatically repeats the MOVSB 100 times.

Example of MOVS with a REP

Write an ALP that transfers a block of 50 bytes of data. The source and destination memory blocks start at 8000 H and 9000 H memory locations respectively. The data segment register value is 3000H.

Solution:

```
2000 MOV AX, 3000H        : Move initial address of DS register into AX.
2003 MOV DS, AX           : DS loaded with AX
2005 MOV ES , AX          : ES loaded with AX
2007 MOV SI, 8000 H       : Source address put into SI.
200A MOV DI, 9000 H       : Destination address put into DI.
200D MOV CX, 0032 H       : Count value for number of bytes put into CX register
2011 CLD
2012 REP MOVSB
2013 HLT
```

If at 2011 STD then ???

COPY A BLOCK OF DATA FROM ONE MEMORY AREA TO ANOTHER MEMORY AREA-50 DATA

```
.data
Array1   db0ah,bch,deh,0f5h,11h, 56h,78h,0ffh,0ffh ,23h4ah, …
Array2   db   50 dup(0)
.code
startup
MOV     CX, 32H
LEA      SI, array1
LEA     DI, array2
CLD
REP   MOVSB
.EXIT
END
```

LODS/LODSB/LODSW /LODSD

Loads AL or AX or EAX with the data stored at the data segment

•Offset address indexed by SI register

•After loading contents of SI INC if D = 0 & DEC if D = 1

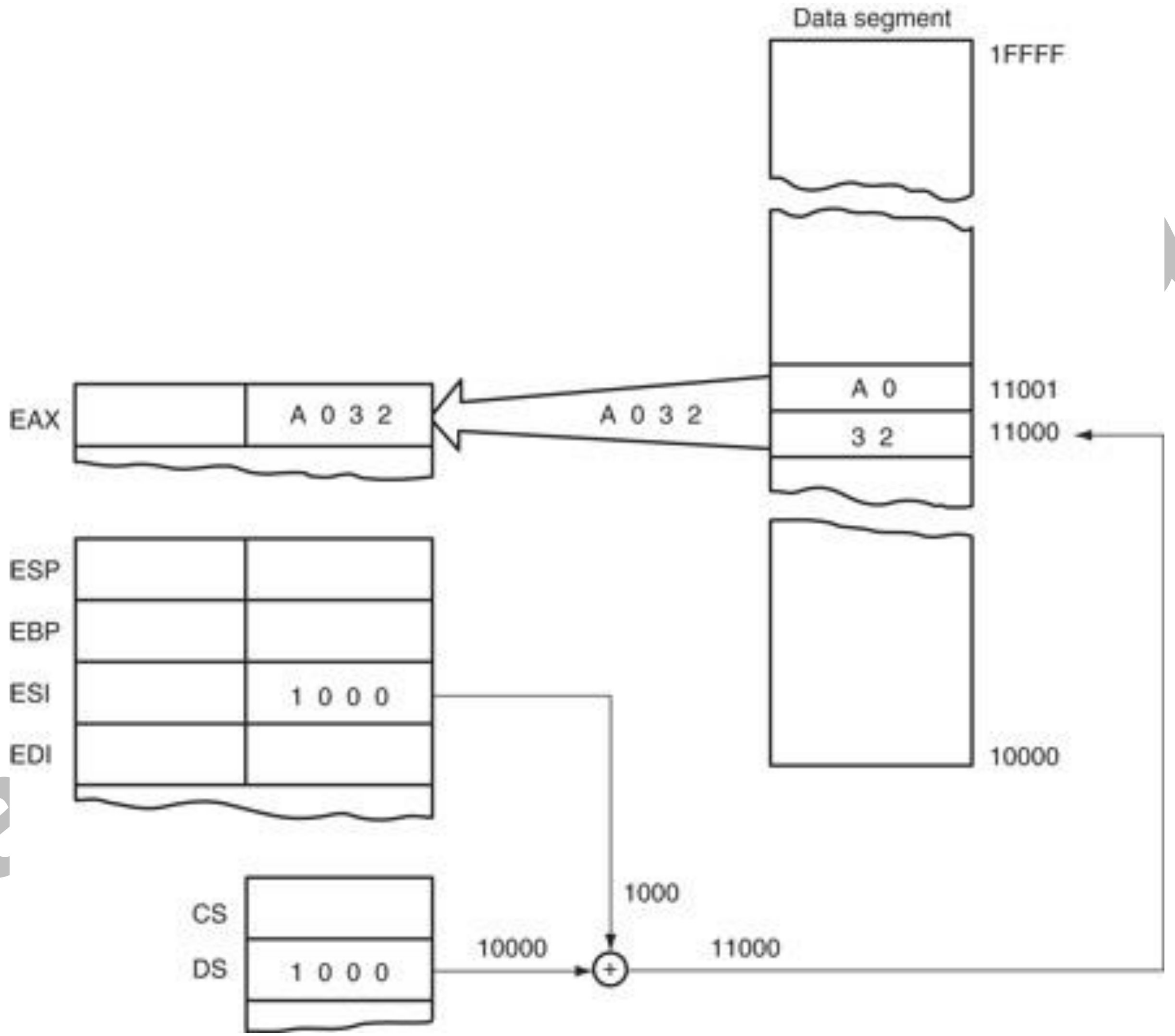
LODSB ; AL = DS:[SI]; SI = SI ±1

LODSW ; AX = DS:[SI]; SI = SI ±2

LODSD; EAX = DS:[SI]; SI = SI±4

•**LODS** affects no FLAGs

The operation of the LODSW instruction if DS=1000H, D=0,11000H,=32 11001H = A0. This instruction is shown after AX is loaded from memory, but before SI increments by 2.



STOS /STOSB/STOSW

Stores AL or AX or EAX into the Extra segment ES memory at Offset address indexed by DI register

•After storing contents in DI, INC if D = 0 & DEC if D = 1

STOSB ; ES:[DI]=AL; DI = DI ±1

STOSW ; ES:[DI]=AX; DI = DI ±2

STOSD; ES:[DI]=EAX; DI = DI ±4

STOS affects no FLAGs

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
```

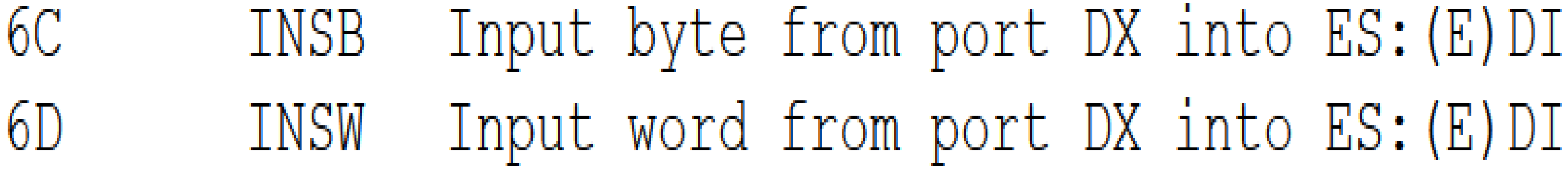
INS

- Transfers a byte or word of data from an I/O device into the extra segment memory location addressed by the DI register.
  - I/O address is contained in the DX register
- Useful for inputting a block of data from an external I/O device directly into the memory.
- Ex : One application transfers data from a disk drive to memory.
  - disk drives are often considered and interfaced as I/O devices in a computer system

THREE basic forms of the INS.

- INSB inputs data from an 8-bit I/O device and stores it in a memory location indexed by DI.
- INSW instruction inputs 16-bit I/O data and stores it in a word-sized memory location.
- INSD instruction inputs 32-bit I/O data and stores it in a word-sized memory location.
- These instructions can be repeated using the REP prefix
  - allows an entire block of input data to be stored in the memory from **an I/O device**

INS Examples



6C	INSB	Input byte from port DX into ES:(E)DI
6D	INSW	Input word from port DX into ES:(E)DI

OUTS

- Transfers a byte or word data from the data segment memory location address indexed by SI to an I/O device.
  - I/O device addressed by the DX register as with the INS instruction

Thank you