UNIT 2

The 8086 Microprocessor: Architecture, Register organization, 8086 signal description, Physical memory organization, Minimum and Maximum mode system and timing diagrams, Addressing modes, 8086 Instruction Set and Assembler Directives, Assembly Language example programs, Stack structure of 8086, Interrupt structure of 8086, Interrupt vector table, Procedures and macros.

Why study instruction set?

- Study of instruction set is required to write instructions in machine code that can be recognized and executed by a central processing unit.
- The knowledge will help you write lines of code or program by which you will be able to tell the processor, the sequence of operations to be performed.

Instruction set of 8086

1. Data Copy/Transfer instructions
2. Arithmetic Instructions
3. Logical instructions
4. Shift & Rotate instructions
5. String manipulation instructions
6. Control Transfer or Branching instructions
7. Machine Control instructions
8. Flag Manipulation instructions

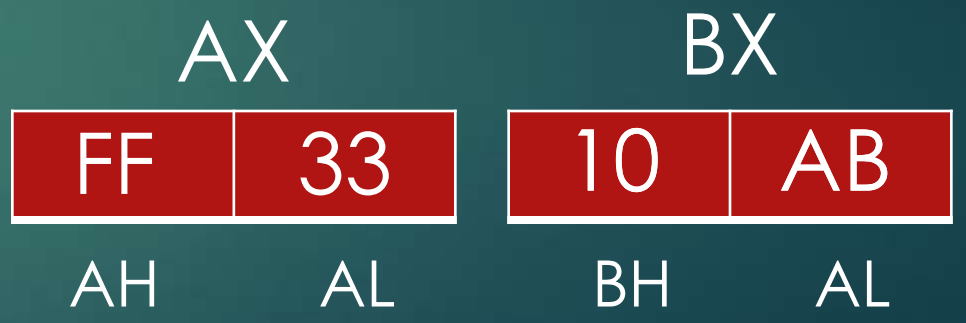
1: Data Copy/Transfer instructions

MOV Destination, Source

- There will be transfer of data from source to destination.

- **Source** can be register, memory location or immediate data.
- **Destination** can be register or memory operand.
- **Both Source and Destination cannot be memory location** or segment registers at the same time.

Eg: MOV AX,BX



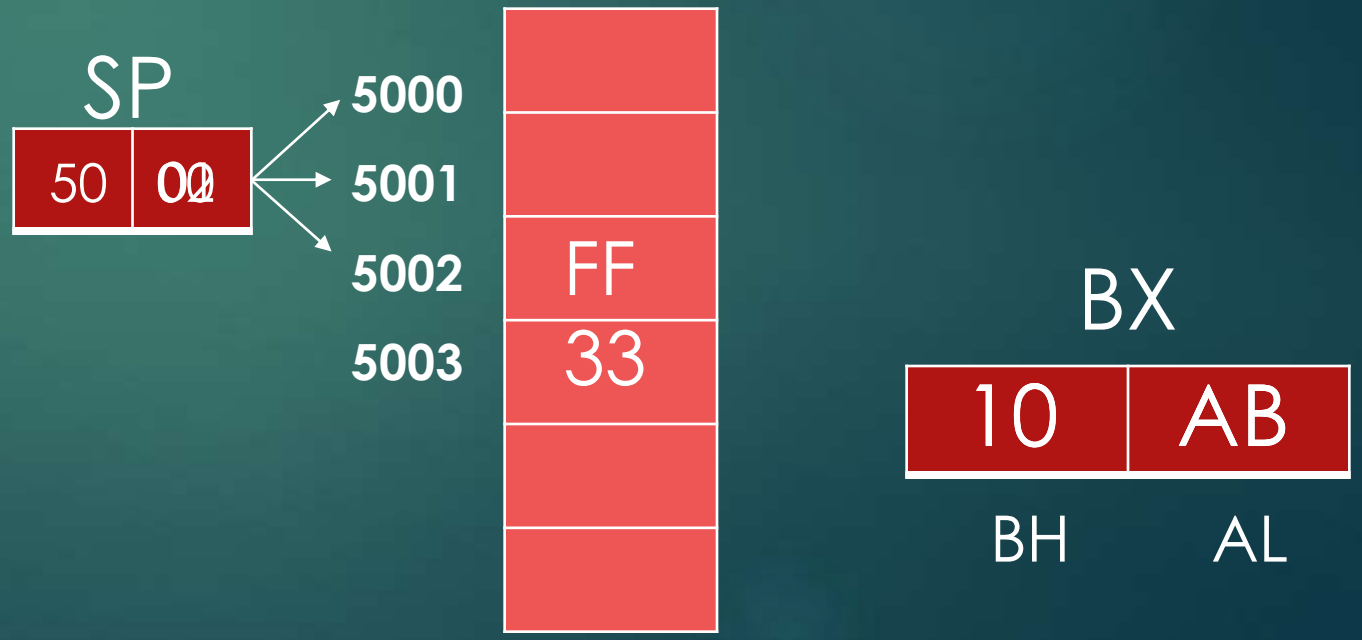
1: Data Copy/Transfer instructions

PUSH Source

- Pushes the 16 bit content specified by source in the instruction on to the stack

- **Pushing** operation decrements stack pointer.
- **Stack pointer** is a 16-bit register, contains the address of the data item currently on top of the stack.

Eg: PUSH BX



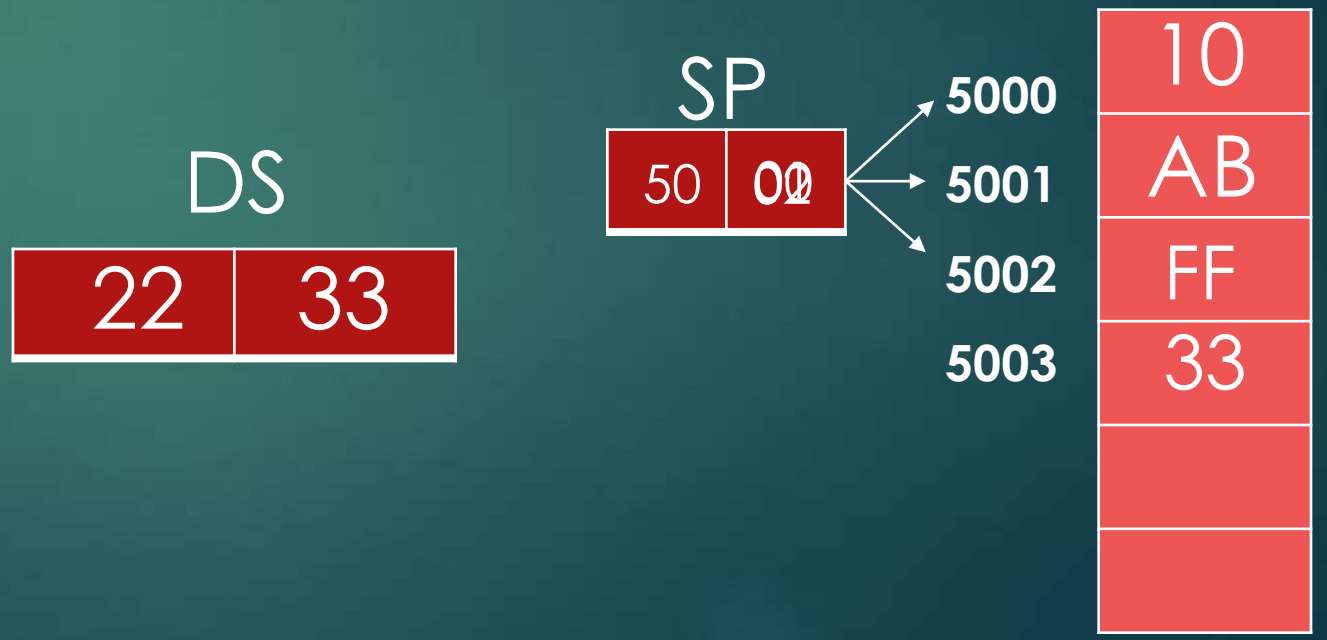
1: Data Copy/Transfer instructions

POP Destination

- Pops the 16 bit content from stack to destination specified in instruction.

- **Popping** operation increments stack pointer.

Eg: POP DS



1: Data Copy/Transfer instructions

XCHG Destination, Source

- This instruction exchanges contents of Source with destination.

- It cannot exchange two memory locations directly.

Eg: XCHG BX,AX

1: Data Copy/Transfer instructions

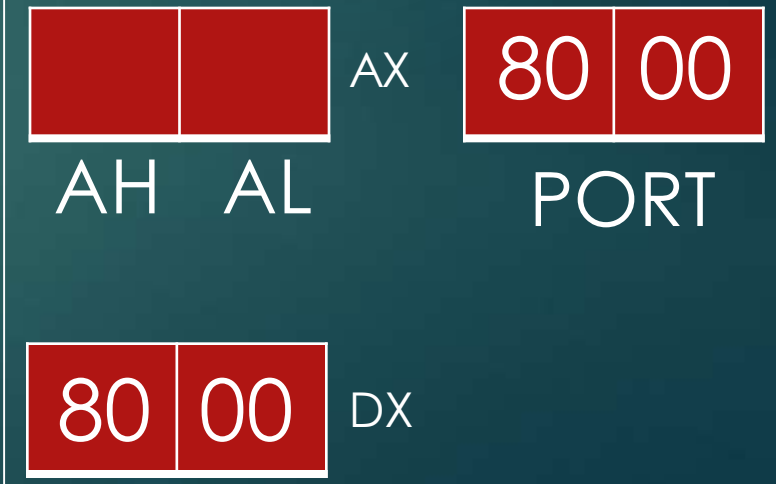
IN AL/AX, 8-bit/16-bit port address

- It copies data to accumulator from a port with 8-bit or 16-bit address.

- DX is the only register is allowed to carry port address.

Eg: IN AL, 80H

Eg: IN AX, DX



1: Data Copy/Transfer instructions

OUT 8-bit/16-bit port address, AL/AX

- It is used for writing to an 8-bit or 16-bit output port.

- DX is the only register is allowed to carry port address.

Eg: OUT 03H, AL

Eg: OUT DX, AX

1: Data Copy/Transfer instructions

XLAT

- Translate instruction is used to find out codes in case of code conversion.

Digit	Display	gfedcba	abcdefg	a	b	c	d	e	f	g
0	0	0×3F	0×7E	on	on	on	on	on	on	off
1	1	0×06	0×30	off	on	on	off	off	off	off
2	2	0×5B	0×6D	on	on	off	on	on	off	on
3	3	0×4F	0×79	on	on	on	on	off	off	on
4	4	0×66	0×33	off	on	on	off	off	on	on
5	5	0×6D	0×5B	on	off	on	on	off	on	on
6	6	0×7D	0×5F	on	off	on	on	on	on	on
7	7	0×07	0×70	on	on	on	off	off	off	off
8	8	0×7F	0×7F	on	on	on	on	on	on	on
9	9	0×6F	0×7B	on	on	on	on	off	on	on
A	A	0×77	0×77	on	on	on	off	on	on	on
b	b	0×7C	0×1F	off	off	on	on	on	on	on
C	C	0×39	0×4E	on	off	off	on	on	on	off
d	d	0×5E	0×3D	off	on	on	on	on	off	on
E	E	0×79	0×4F	on	off	off	on	on	on	on
F	F	0×71	0×47	on	off	off	off	on	on	on

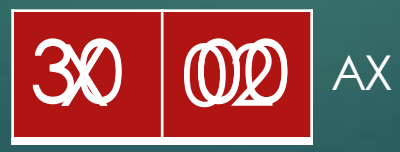
Eg: MOV AX, TABLE SEGMENT ADDRESS

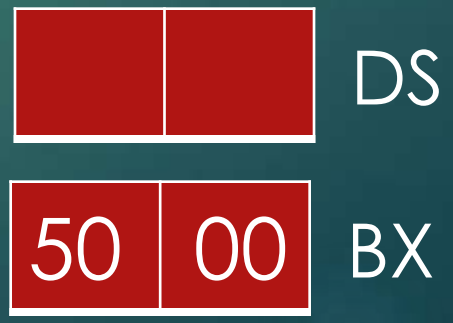
MOV DS, AX

MOV AL, DISPLAY CODE

MOV BX, OFFSET TABLE

XLAT





1: Data Copy/Transfer instructions

- **LEA Register, Src:**
  - It loads a 16-bit register with the offset address of the data specified by the Src.
  - E.g.: LEA BX, ADR
    - This instruction loads the effective address of label ADR(offset of ADR) will be transferred to BX register.

Data Transfer Instructions

- **LDS/LES Des, Src:**
  - This instruction loads the DS or ES register and the specified destination register in the instruction with the content of memory location specified as source in the instruction.

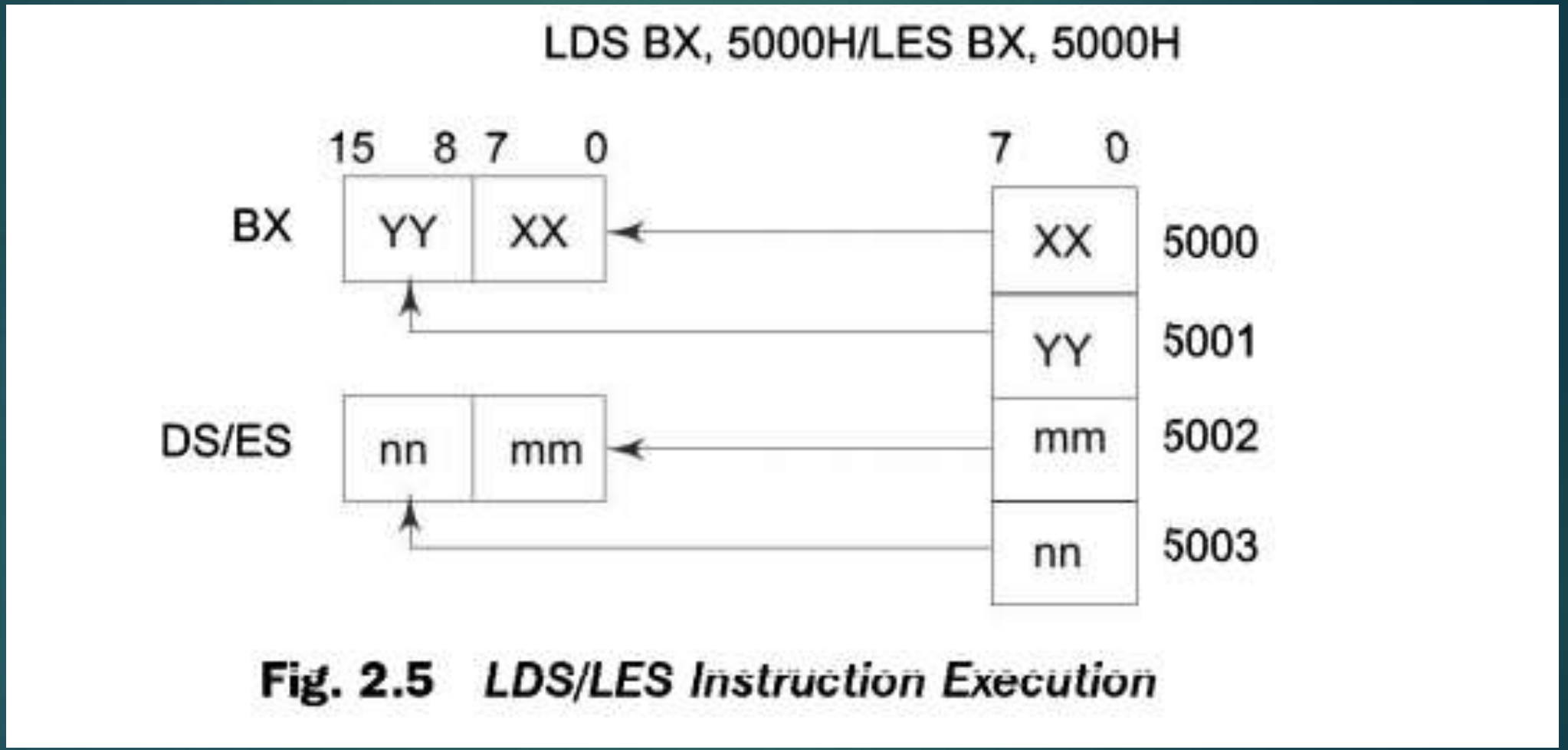


Fig. 2.5 *LDS/LES Instruction Execution*

Data Transfer Instructions

- **LAHF:**
  - It copies the lower byte of flag register to AH.
- **SAHF:**
  - It copies the contents of AH to lower byte of flag register.
- **PUSHF:**
  - Pushes flag register to top of stack.
- **POPF:**
  - Pops the stack top to flag register.