Instruction Set of 8086µP

There are 117 basic instructions in the instruction set of 8086µP. The instruction set of 8086µP can be divided into the following groups :

1. Data Transfer Instructions 2. Arithmetic and Logic Instructions

3. Shift and Rotate Instructions 4. Flag Control Instructions

5. Compare Instructions 6. Jump Instructions

7. Loop Instructions 8. Stack Instructions

9. Strings Instructions 10. Subroutines Instructions

Data Transfer Instructions

The data transfer instructions include:

1. **MOV** (byte or word) Instruction. 2. **XCHG** (Exchange byte or word)

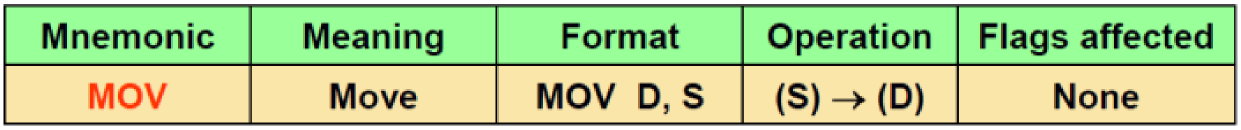
3. **XLAT** (Translate byte) 4. **LEA** (Load effective address)

5. **LDS** (Load register and DS) 6. **LES** (Load register and ES)

The MOV Instruction:

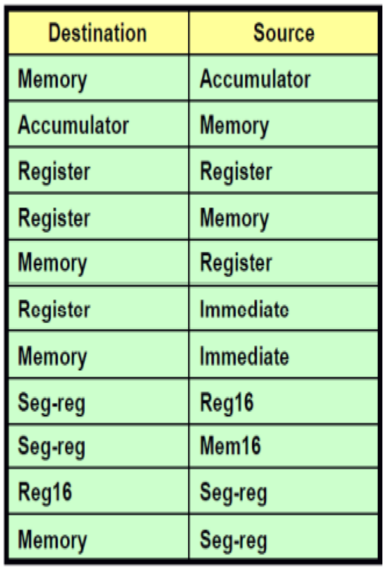
The MOV instruction is used to transfer (copy) a byte or word of data from a source operand to a destination operand. The general form of MOV

instruction is as shown below:



Mnemonic	Meaning	Format	Operation	Flags affected
MOV	Move	MOV D, S	(S) → (D)	None

The allowed operands for the source and destination are listed below:



Destination	Source
Memory	Accumulator
Accumulator	Memory
Register	Register
Register	Memory
Memory	Register
Register	Immediate
Memory	Immediate
Seg-reg	Reg16
Seg-reg	Mem16
Reg16	Seg-reg
Memory	Seg-reg

Examples :

MOV AX , BX
MOV AL , FFH
MOV [BX] , CX
MOV [SI] , FF87H
MOV BH , [PB+DI]
MOV DS , AX
MOV AL , 'A'
MOV [BX] , [SI] is not allowed, why?
MOV DS , FA34H is not allowed, why?
MOV DS , CS is not allowed, why?
MOV CS, AX is not allowed, why ?

The Exchange XCHG Instruction

The XCHG (exchange) instruction exchanges the contents of a register a memory with the contents of any other register or memory. The general form of this instruction is as shown below:

Mnemonic	Meaning	Format	Operation	Flags affected
XCHG	Exchange	XCHG D, S	(D) ↔ (S)	None

The allowed operands for the source and destination are listed below:

Destination	Source
Accumulator	Reg16
Memory	Register
Register	Register
Register	Memory

Example
(AX)=1000H and (BX)=2000H
XCHG AX , BX
After execution :
(AX)=2000H and (BX)=1000H

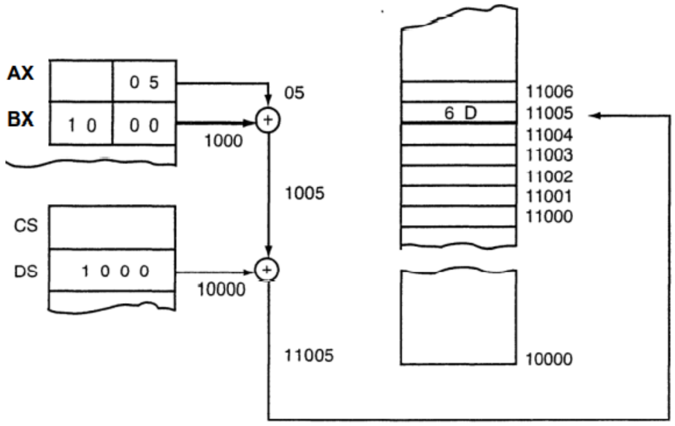
H.W. Repeat the example by using MOV instructions only.

The XLAT Instruction

This instruction used to simplify implementation of the lookup table operation. The general form of this instruction is as shown below:

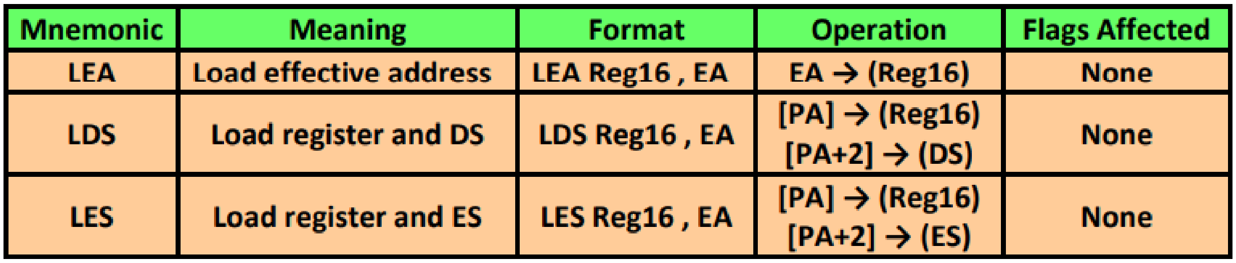
Mnemonic	Meaning	Format	Operation	Flags affected
XLAT	Translate	XLAT	((AL)+(BX)+(DS)0) → (AL)	None

Ex: Assume (DS) = 1000H, (BX)=1000H, and (AL)=05H. Execution of XLAT replaces the contents of AL by the contents of memory location with physical address : PA = (DS)0 + (BX) + (AL) = 10000H + 1000H + 05H = 11005H Thus: (AL) = 6DH (the old byte 05H in AL is replaced by 6DH).



Load-Effective Address (LEA, LDS, and LES) Instructions:

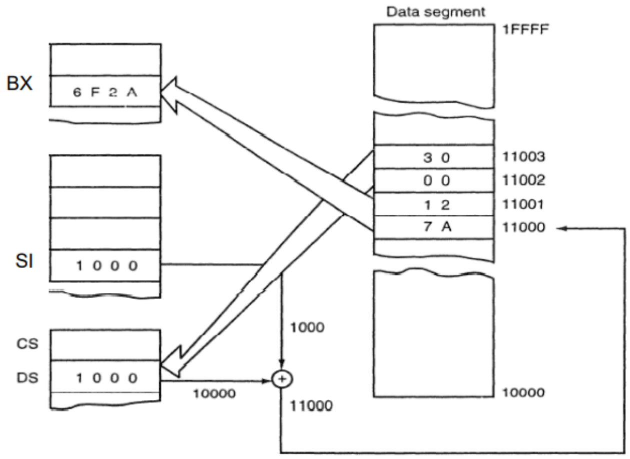
These instructions load a segment and general purpose registers with an address directly from memory. The general forms of these instructions are as shown below:



Mnemonic	Meaning	Format	Operation	Flags Affected
LEA	Load effective address	LEA Reg16 , EA	EA → (Reg16)	None
LDS	Load register and DS	LDS Reg16 , EA	[PA] → (Reg16) [PA+2] → (DS)	None
LES	Load register and ES	LES Reg16 , EA	[PA] → (Reg16) [PA+2] → (ES)	None

Ex: What does BX contain after executing: LEA BX , [SI+100H] If (SI)=1000H Ans. The effective address EA=1000H + 100H = 1100H → (BX) = 1100H Ex : If (SI)=1000H and (DS)=1000H , what do the registers BX and DS contain after execution LDS BX , [SI] instruction?

Ans. PA = 10000H + 1000H = 11000H The execution of LDS BX , [SI] loads BX from addresses 11000H and 11001H and DS from addresses 11002H and 11003H. (BX) = 127AH and (DS) = 3000H

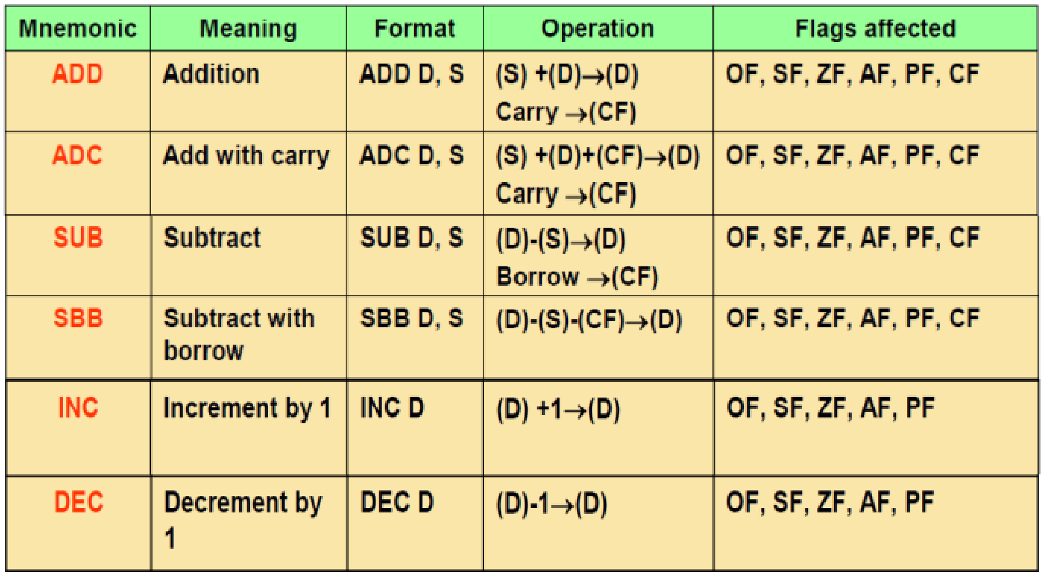


Note: All data transfer instructions do not affect the status flags.

Arithmetic and Logic Group

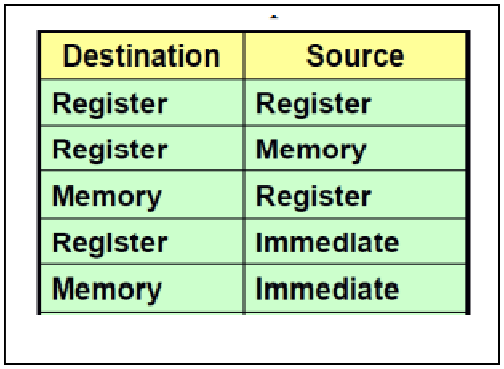
The arithmetic group includes instructions for the addition, subtraction, multiplication, division as well as increment and decrement operations.

Addition and Subtraction Instructions



Mnemonic	Meaning	Format	Operation	Flags affected
ADD	Addition	ADD D, S	(S) +(D)→(D) Carry →(CF)	OF, SF, ZF, AF, PF, CF
ADC	Add with carry	ADC D, S	(S) +(D)+(CF)→(D) Carry →(CF)	OF, SF, ZF, AF, PF, CF
SUB	Subtract	SUB D, S	(D)-(S)→(D) Borrow →(CF)	OF, SF, ZF, AF, PF, CF
SBB	Subtract with borrow	SBB D, S	(D)-(S)-(CF)→(D)	OF, SF, ZF, AF, PF, CF
INC	Increment by 1	INC D	(D) +1→(D)	OF, SF, ZF, AF, PF
DEC	Decrement by 1	DEC D	(D)-1→(D)	OF, SF, ZF, AF, PF

The allowed operands:



Destination	Source
Register	Register
Register	Memory
Memory	Register
Reglster	Immedlate
Memory	Immediate

Examples

```
ADD AL , BL
ADD AX , DI
ADC DX , 1234H
SUB AX , [BX]
SBB [BX+DI] , CL
INC AL
 DEC [DI]
```

Ex: Write an ALP that subtracts 1234H existing in DX from the word beginning at memory location 64200.

Ans. :

```
MOV AX , 6000H
MOV DS , AX
MOV BX , 4200H
MOV DX , 1234H
SUB [BX] , DX
HLT
```

Ex: Write a piece of code to add two 32-bit numbers stored at 82000H and 84000H and store the result at 86000H.

Ans. :

```
MOV AX , 8000H
MOV DS , AX
MOV AX , [2000]
 MOV DX , [2002]
ADD AX , [4000]
ADC DX , [4002]
MOV [6000] , AX
MOV [6002] , DX
HTL
```

Example:

$(\boldsymbol{AX})=\mathbf{1000}\boldsymbol{H}\ \boldsymbol{and}\ (\boldsymbol{BX})=\mathbf{2000}\boldsymbol{H}$

$XCHG\ AX,BX$

$\boldsymbol{After\ execution}$: $(AX)=2000H\ and\ (BX)=1000H$

Using MOV instruction

```
MOV CX,AX
MOV AX,BX
MOV BX,CX
```