Computer Science Dept.

Department	Computer Science	القسم:
Subject Name:	Microprocessors - (7)	أسم المادة :
Year of Study:	2023-2024	السنة الدراسية:
Term:	Second Term	الفصل الدراسي:
Email	ali.sadoon@uoa.edu.iq	Email
Instructor Name:	Ali Saadoon AHMED	أسم التدريسي:

OUTLINE

✓*Instruction Set of 8086µP*

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

Instruction Set of 8086µP

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)

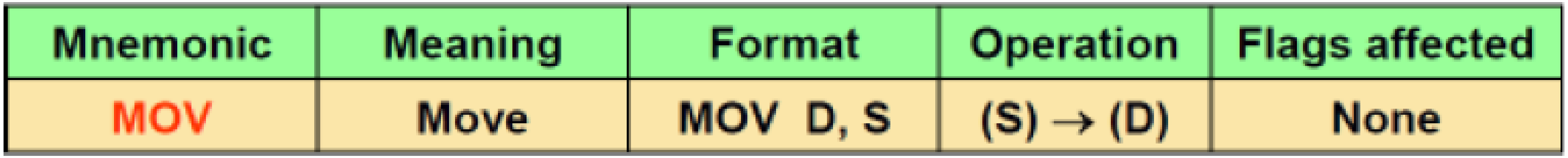
5. LDS (Load register and DS)

6. LES (Load register and ES)

Instruction Set of 8086µP

The MOV Instruction:

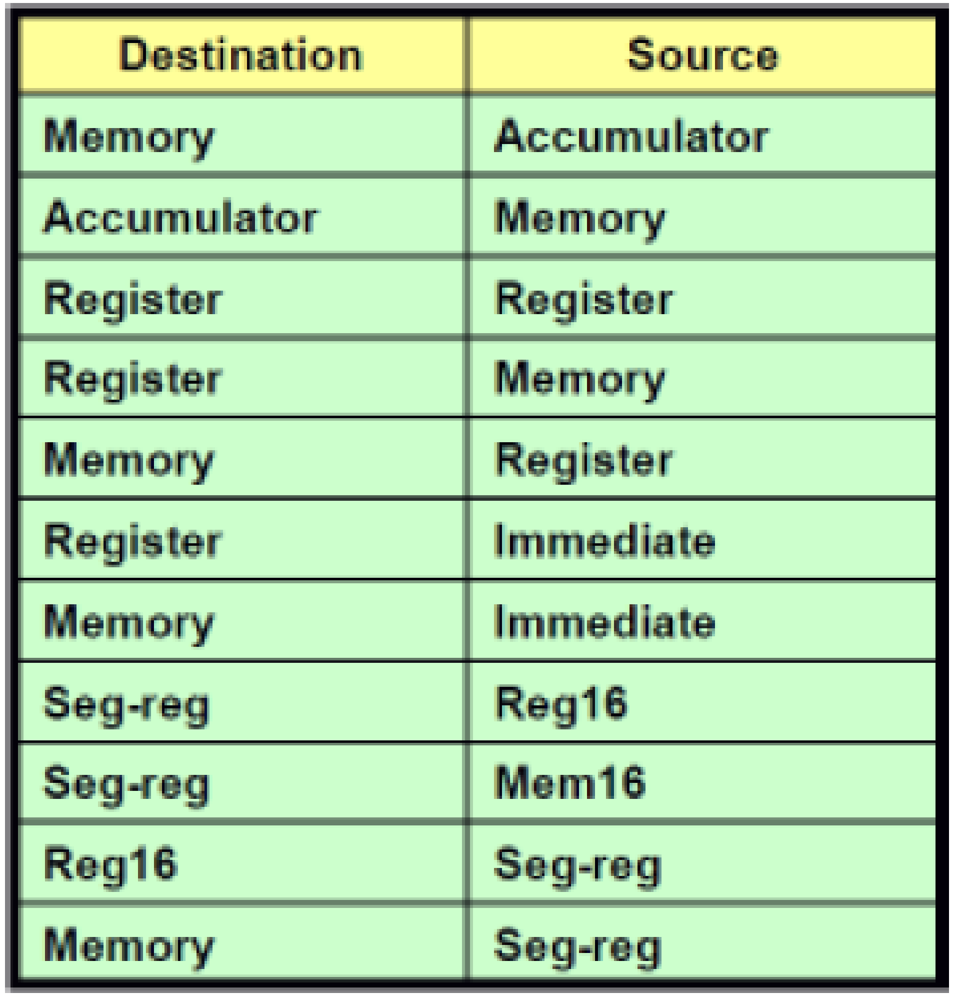
The MOV instruction is used to transfer (copy) a byte or word of data from a source operand to a destination operand. The general form of MOV instruction is as shown below:



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

Instruction Set of 8086µP

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

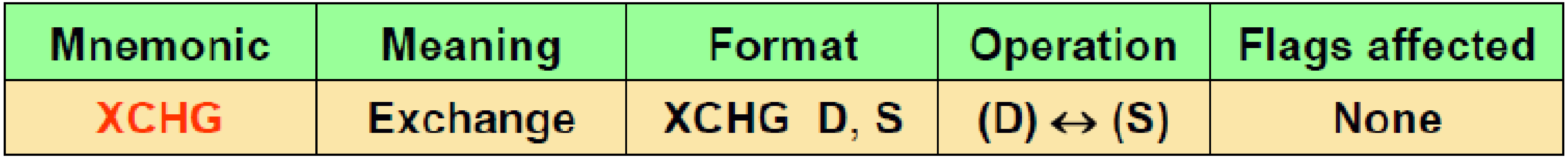


Destination	Source
Memory	Accumulator
Accumulator	Memory
Register	Register
Register	Memory
Memory	Register
Register	Immediate
Memory	Immediate
Seg-reg	Reg16
Seg-reg	Mem16
Reg16	Seg-reg
Memory	Seg-reg

Instruction Set of 8086µP

The Exchange XCHG Instruction

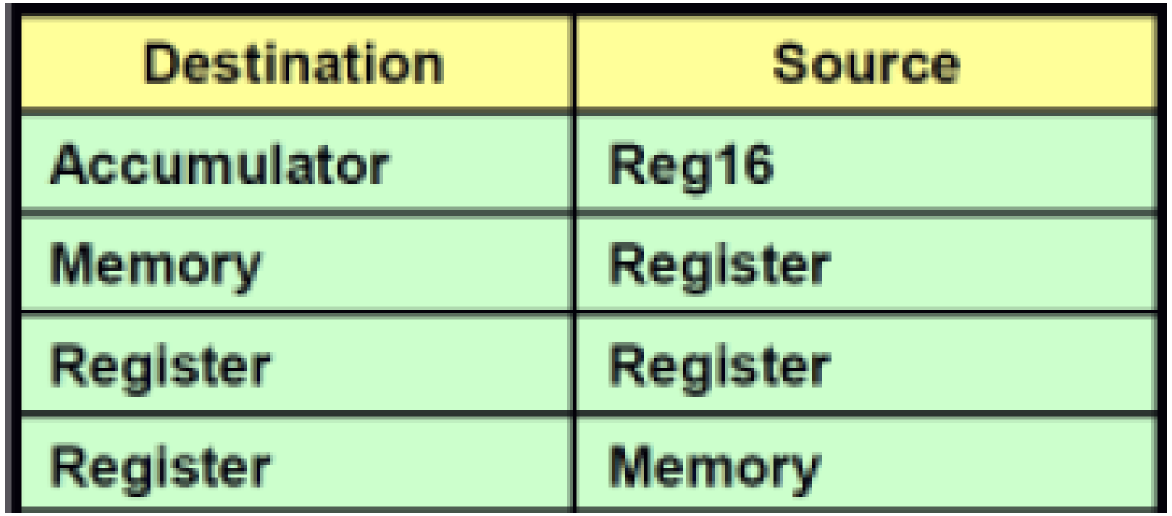
The XCHG (exchange) instruction exchanges the contents of a register a memory with the contents of any other register or memory. The general form of this instruction is as shown below:



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

Instruction Set of 8086µP

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



Destination	Source
Accumulator	Reg16
Memory	Register
Register	Register
Register	Memory

Instruction Set of 8086µP

Example:

(AX)=1000H and (BX)=2000H

XCHG AX , BX

After execution :

(AX)=2000H and (BX)=1000H

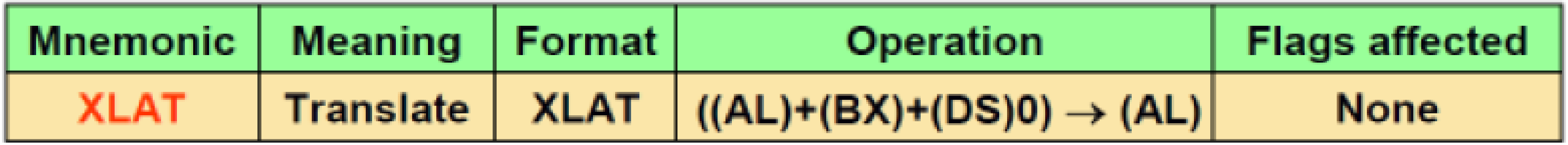
H.W. Repeat the example by using MOV instructions only.

Instruction Set of 8086µP

The XLAT Instruction:

This instruction used to simplify implementation of the lookup table operation.

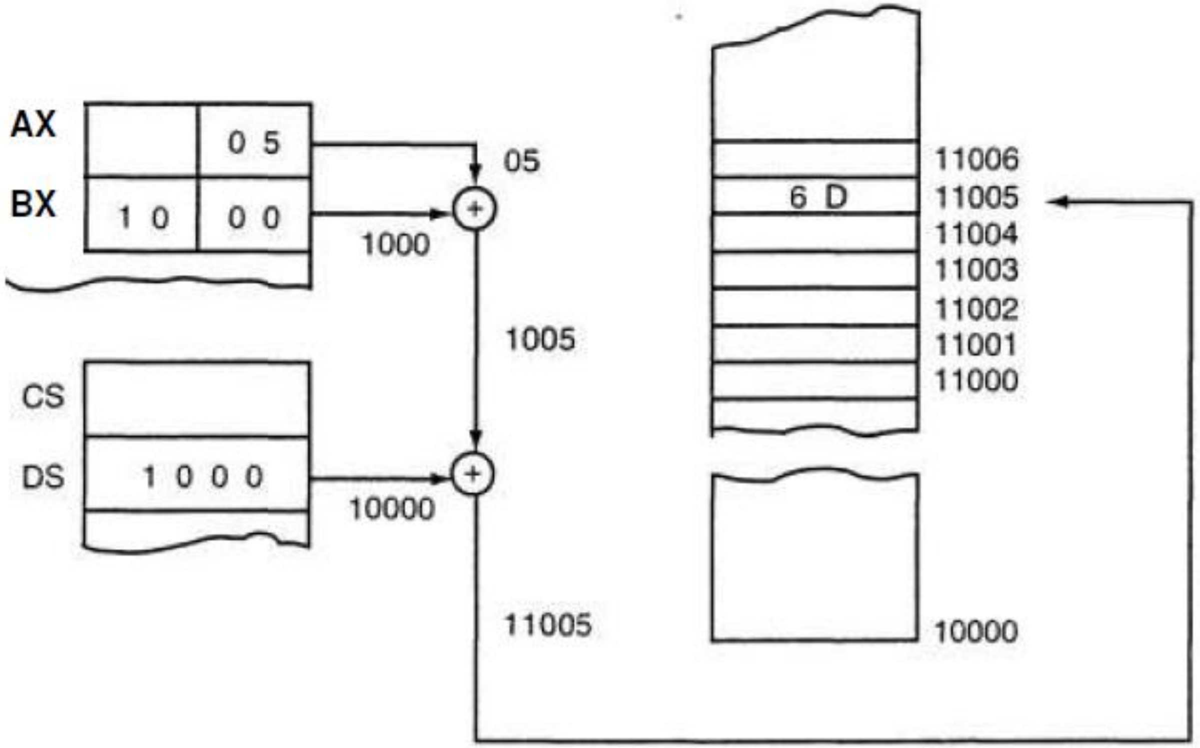
The general form of this instruction is as shown below:



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

Instruction Set of 8086µP

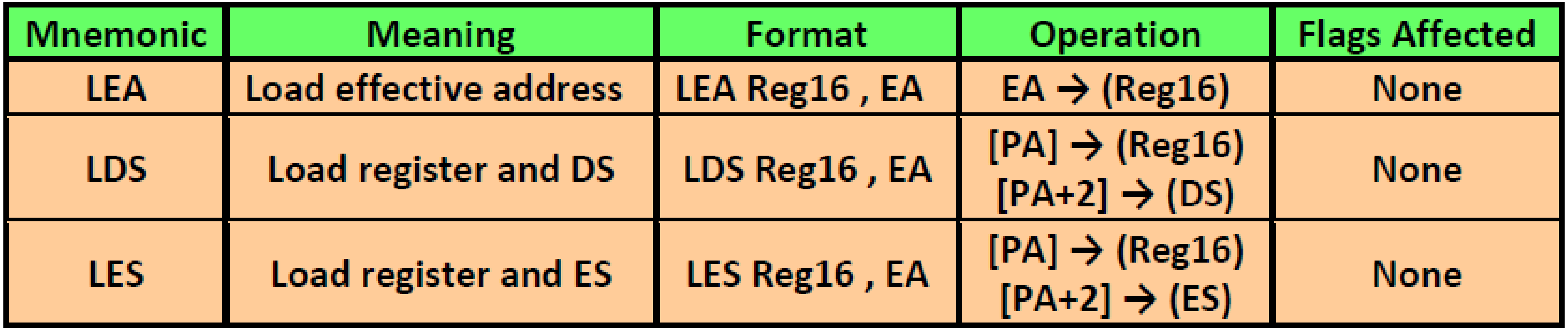
Ex: Assume (DS) = 1000H, (BX)=1000H, and (AL)=05H. Execution of XLAT replaces the contents of AL by the contents of memory location with physical address **: PA = (DS)0 + (BX) + (AL) = 10000H + 1000H + 05H = 11005H** Thus: (AL) = 6DH (the old byte 05H in AL is replaced by 6DH)



Instruction Set of 8086μP

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

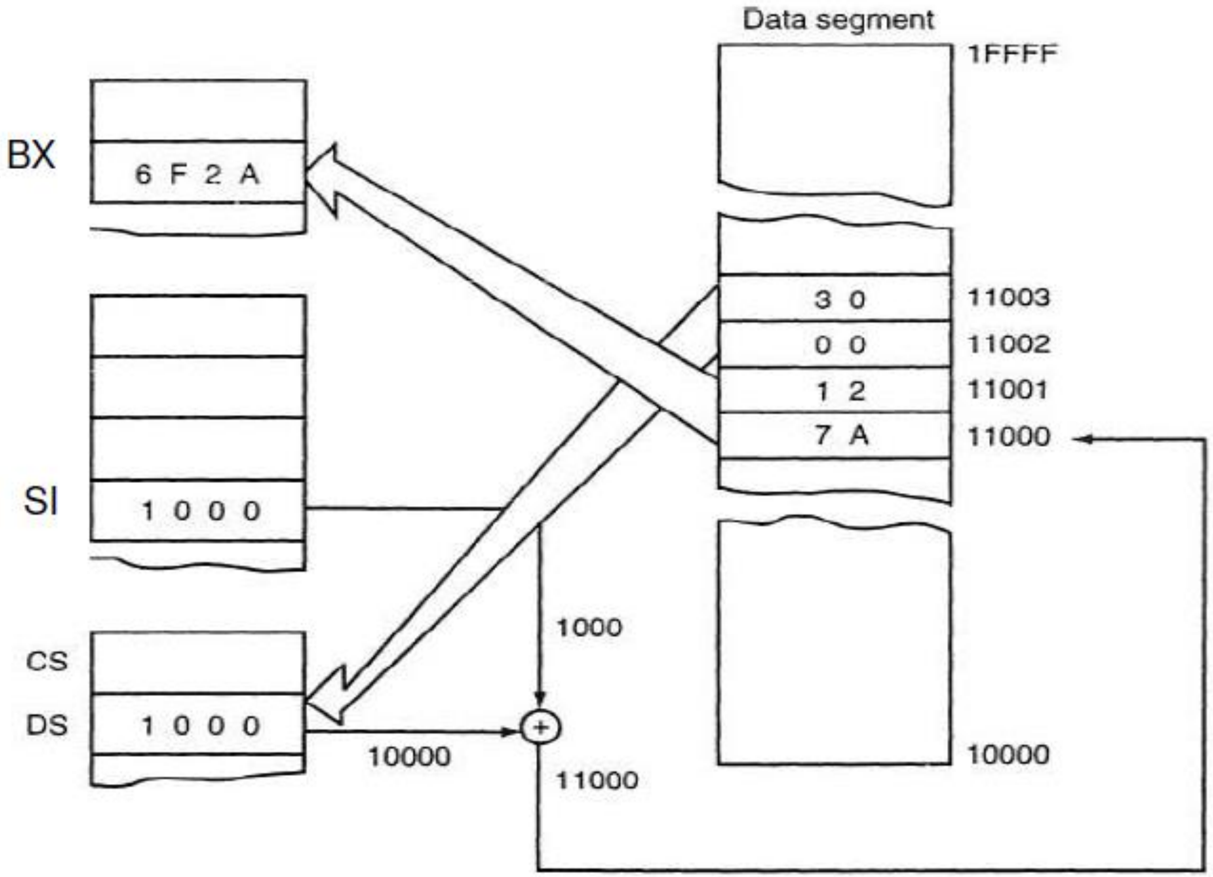
Instruction Set of 8086µP

Ex : If (SI)=1000H and (DS)=1000H , what do the registers BX and DS contain after execution

LDS BX , [SI]

Instruction Set of 8086µP

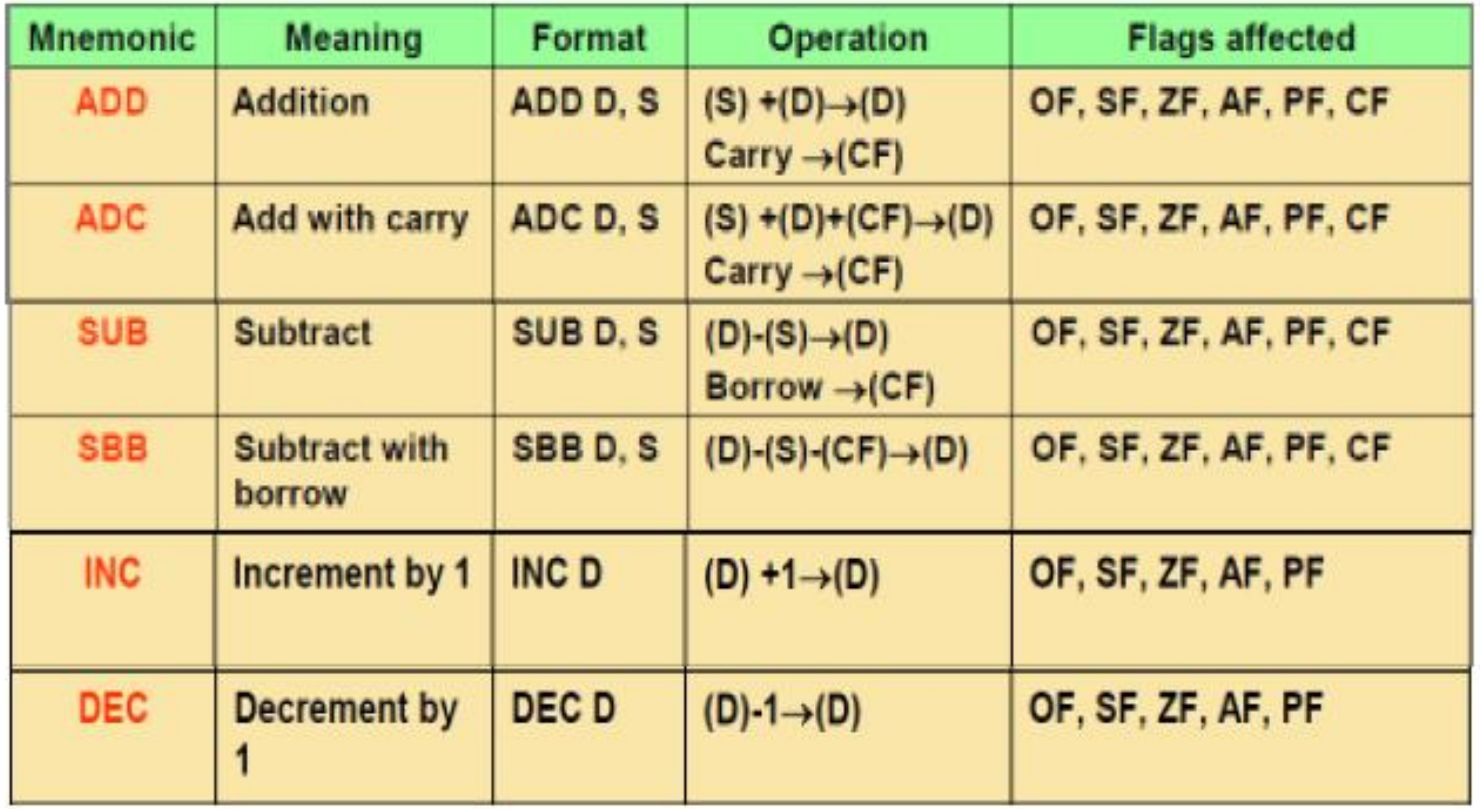
PA = 10000H + 1000H = 11000H The execution of **LDS BX , [SI]** loads BX from addresses 11000H and 11001H and DS from addresses 11002H and 11003H. (BX) = 127AH and (DS) = 3000H



Instruction Set of 8086µP

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

Instruction Set of 8086µP

ADD AL , BL

ADD AX , DI

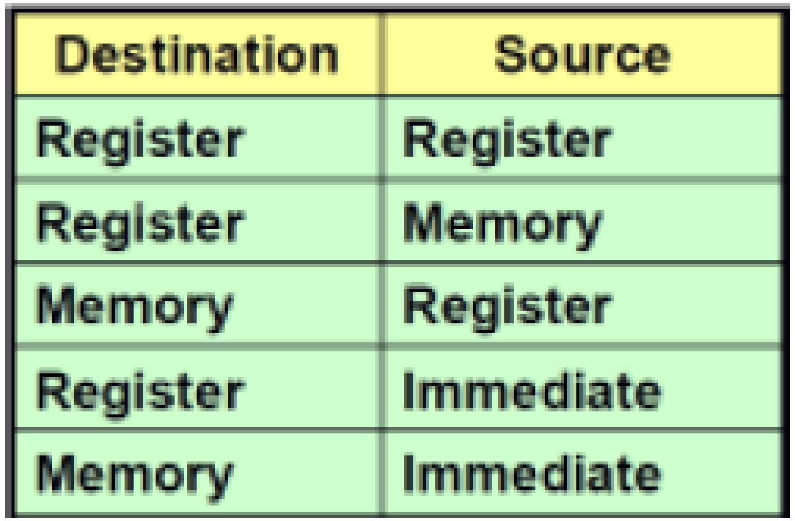
ADC DX , 1234H

SUB AX , [BX]

SBB [BX+DI] , CL

INC AL

DEC [DI]



Destination	Source
Register	Register
Register	Memory
Memory	Register
Register	Immediate
Memory	Immediate

Instruction Set of 8086µP

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

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

Instruction Set of 8086µP

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

Ans. :

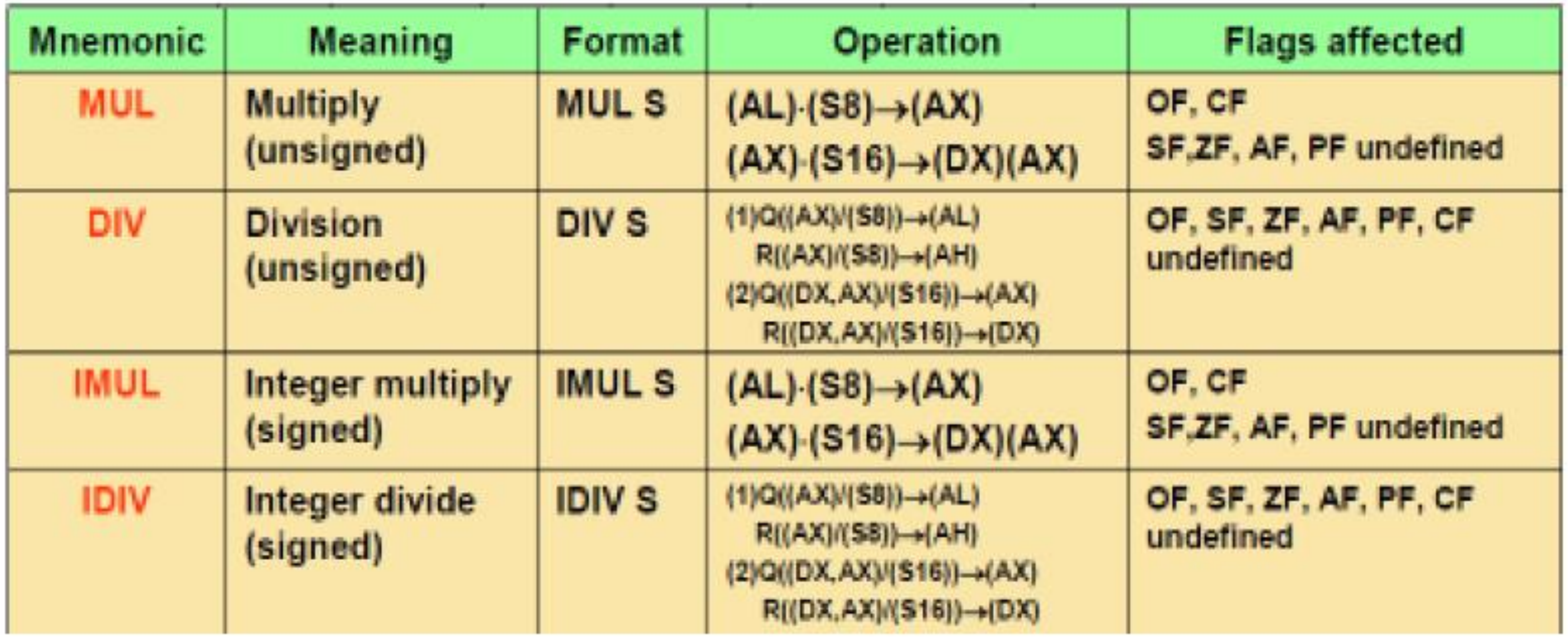
- MOV AX , 8000H
- MOV DS , AX
- MOV AX , [2000]
- MOV DX , [2002]
- ADD AX , [4000]
- ADC DX , [4002]
- MOV [6000] , AX
- MOV [6002] , DX

- HTL

Instruction Set of 8086µP

Multiplication and Division Instructions:

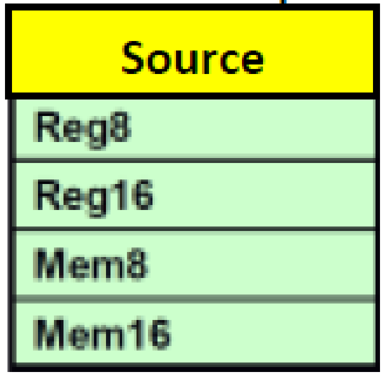
The 8086µP has instructions for multiplication and division of binary, BCD numbers, and signed or unsigned integers. Multiplication and division are performed on bytes or on words.



Mnemonic	Meaning	Format	Operation	Flags affected
MUL	Multiply (unsigned)	MUL S	$(AL)\cdot(S8)\rightarrow(AX)$ $(AX)\cdot(S16)\rightarrow(DX)(AX)$	OF, CF SF,ZF, AF, PF undefined
DIV	Division (unsigned)	DIV S	$(1) Q((AX)/(S8))\rightarrow(AL)$ $R((AX)/(S8))\rightarrow(AH)$ $(2) Q((DX,AX)/(S16))\rightarrow(AX)$ $R((DX,AX)/(S16))\rightarrow(DX)$	OF, SF, ZF, AF, PF, CF undefined
IMUL	Integer multiply (signed)	IMUL S	$(AL)\cdot(S8)\rightarrow(AX)$ $(AX)\cdot(S16)\rightarrow(DX)(AX)$	OF, CF SF,ZF, AF, PF undefined
IDIV	Integer divide (signed)	IDIV S	$(1) Q((AX)/(S8))\rightarrow(AL)$ $R((AX)/(S8))\rightarrow(AH)$ $(2) Q((DX,AX)/(S16))\rightarrow(AX)$ $R((DX,AX)/(S16))\rightarrow(DX)$	OF, SF, ZF, AF, PF, CF undefined

Instruction Set of 8086µP

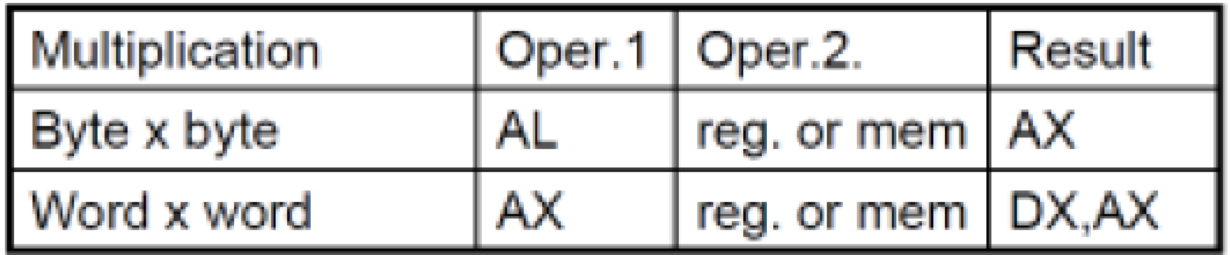
The allowed operand:



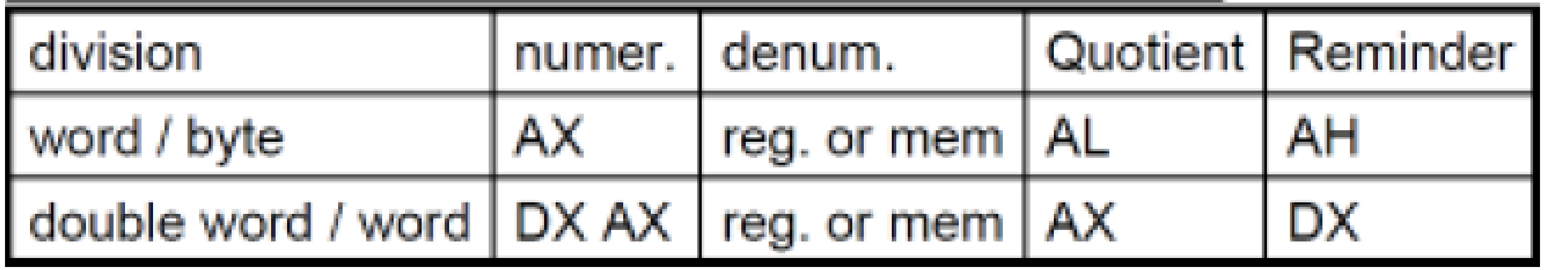
Source
Reg8
Reg16
Mem8
Mem16

Examples

MUL BL
DIV [SI]
IMUL BX
IDIV [DI+1000H]



Multiplication	Oper.1	Oper.2.	Result
Byte x byte	AL	reg. or mem	AX
Word x word	AX	reg. or mem	DX,AX



division	numer.	denum.	Quotient	Reminder
word / byte	AX	reg. or mem	AL	AH
double word / word	DX AX	reg. or mem	AX	DX

Instruction Set of 8086µP

Ex: Write an ALP for dividing 1234H by 34H.

Ans.

```
MOV AX, 1234H
MOV CL, 34H
DIV CL
HLT
```

After Execution : (AX)= 2059H

Quotient in AL = 59H and

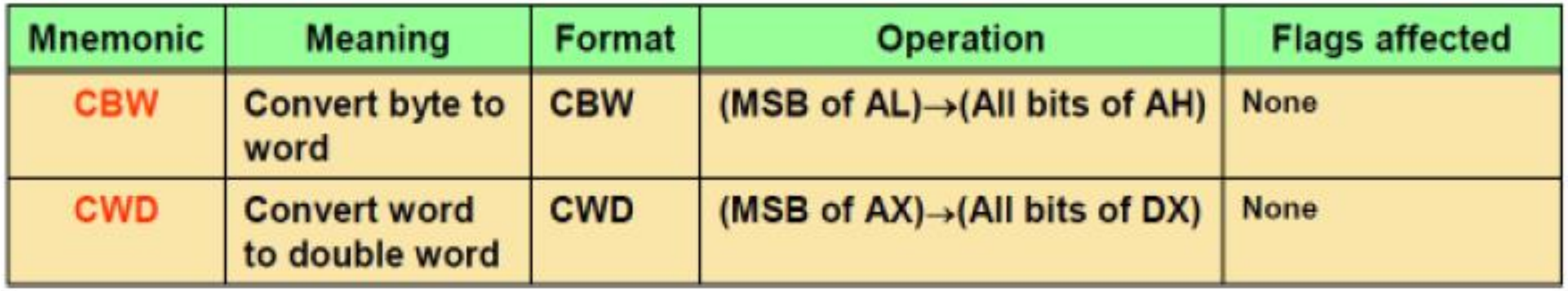
Remainder in AH = 20H.

Instruction Set of 8086μP

CBW(Convert Signed Byte to Word) & CWD(Convert Signed Word to Double Word) instructions:

The division instruction can also be used to divide a sign 8-bit dividend in AL by an 8-bit divisor. For this purpose we use (CBW) instruction. When (CBW) instruction is executed the value of AX register is as shown below:

AH=0 if the number is positive & AH=1 if the number is negative.



Mnemonic	Meaning	Format	Operation	Flags affected
CBW	Convert byte to word	CBW	(MSB of AL)→(All bits of AH)	None
CWD	Convert word to double word	CWD	(MSB of AX)→(All bits of DX)	None

Instruction Set of 8086µP

Ex: What is the result of executing the following piece of code?

MOV AL , A1H

CBW

CWD

The first instruction loads AL with

(AL)=A1H=10100001

Executing the second instruction gives:

(AH)=11111111 = FF

(AX)=1111111110100001= FFA1

Executing the third instruction gives:

(AX)=FFA1

(DX)=FFFF

Instruction Set of 8086µP

Ex: Write an ALP that divide a signed byte stored in 5600AH by the content of BL.

Ans.

```
MOV AX , 5000H
MOV DS , AX
MOV SI , 6000H
MOV AL , [SI + 0AH]
CBW
IDIV BL
MOV [SI + 0AH] , AX
HLT
```

THANK YOU

By:

Ali Saadoon Ahmed