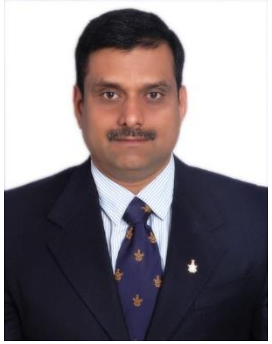
Microprocessors and Interfaces: 2021-22
Lab 4
Addressing Modes in 8086

By Dr. Sanjay Vidhyadharan



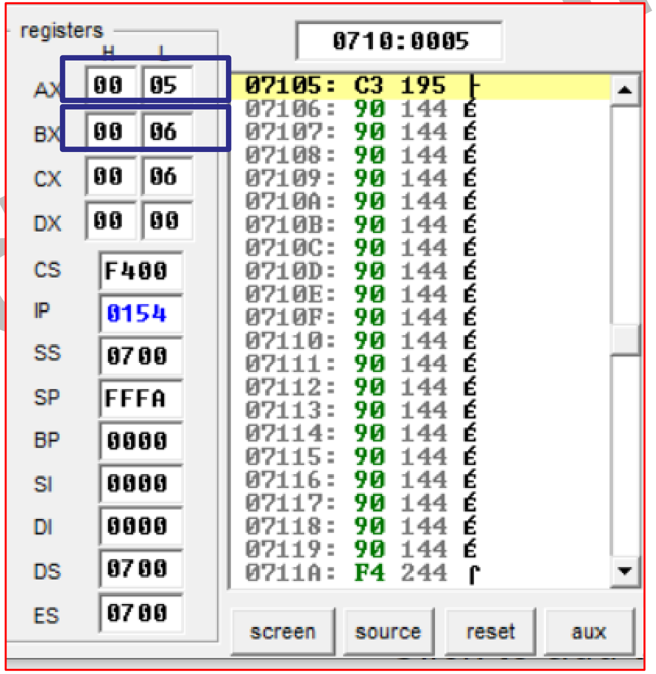
1.1 Immediate Addressing Mode

In this, immediate data is part of instruction, and appears in the form of successive byte or bytes

```
org 100h

MOV AX,0005H
MOV BL,06H

ret
```



Here, 0005H and 06H are the immediate data to be loaded in AX and BL registers, respectively.

1.2 Direct Addressing Mode

In this, a 16 bit memory address or an IO address is directly specified in the instruction

Example

```
ORG 100h
MOV AX, 2162H
MOV DS, AX
MOV CX, 24
MOV [481], CX
MOV BX, [481]
ret
```

Address Location Determination

$(481)_{10} \rightarrow (1E1)_{16}$

DS: 2162_{16}

Offset: 481_{10}

Logical Address: 2162:01E1

Physical Address: DS × 10_{16} + offset = $(21801)_{16}$

1.3 Register Addressing Mode

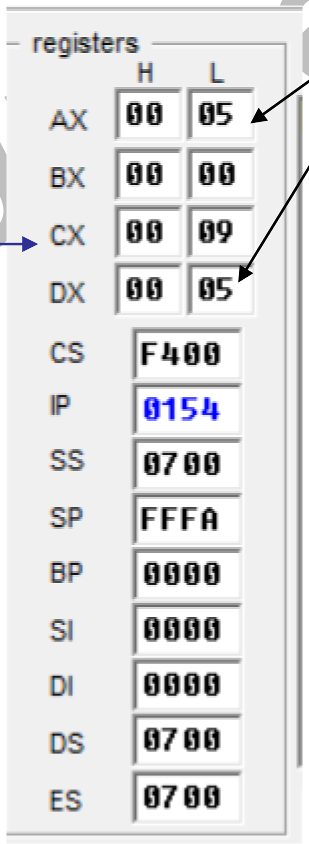
Operands are stored in registers. Transfer of operands are extremely fast since no memory access is required.

All except IP register can be used in this mode.

```
org 100h
MOV AX,0005H
MOV CX,0004H
MOV DX, AX
ADD CX, DX
ret
```

Move data content of Register A to Register D

Result



Content of Register A copied to D

1.4 Base Addressing Mode

- The BX or BP-register is used to hold a base value for effective address (EA).
- A signed 8-bit or unsigned 16-bit displacement will be specified in the instruction.
- The displacement is added to base value in BX or BP to obtain the EA.
  - When BX is used to hold base value for EA, the 20-bit physical address of memory is calculated by multiplying the content of DS by 16_{10} adding to EA.
  - When BP is used to hold base value for EA, the 20-bit physical address of memory is calculated by multiplying the content of SS by 16_{10} and adding to EA.

Example :

MOV AX, [BX+08H]

0008_H 08_H ; EA = (BX) + 0008_H

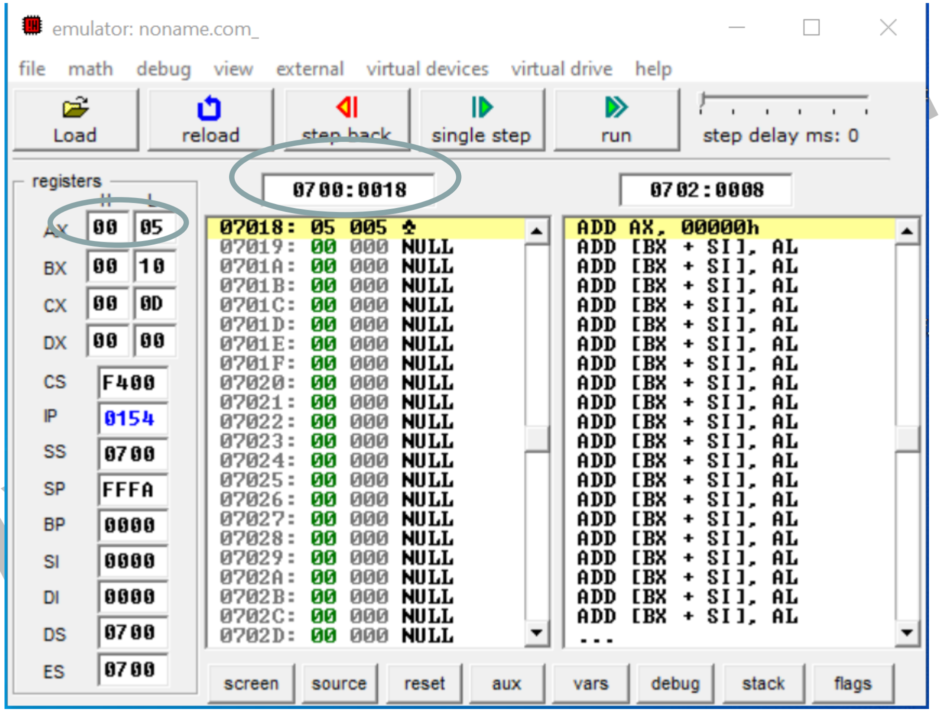
BA = (DS) $\times$ 16_{10} ; MA = BA + EA

(AX) $\leftarrow$ (MA) or (AL) $\leftarrow$ (MA)

(AH) $\leftarrow$ (MA+1)

1.4 Base Addressing Mode

```
org 100h
MOV BX,0010H
MOV [0018H],0005H
MOV AX, [BX+08H]
ret
```



DS: 0700h

BX: 0010h

EA: (BX)+08h = 0018h

Logical Address: 0700:0018

Physical Address: 7018h

1.5 Indexed Addressing Mode

- SI or DI-register is used to hold an index value for memory data.
- A signed 8-bit or unsigned 16-bit displacement will be specified in the instruction.
- The displacement is added to index value in SI or DI-register to obtain the Effective Address (EA)

MOV CX, [SI+0A2H]

$$FFA2_H \xleftarrow{\text{sign extend}} A2_H \quad ; \quad EA = (SI) + FFA2_H$$

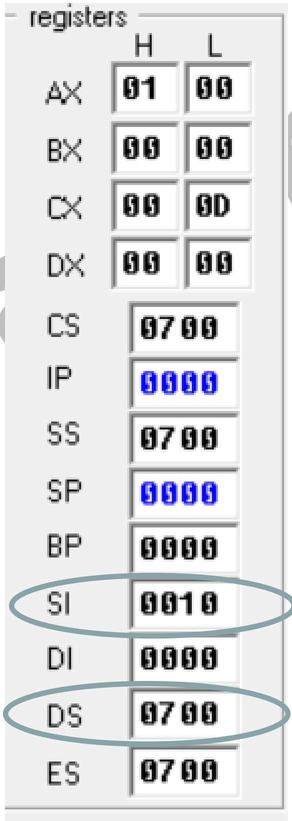
$$BA = (DS) \times 16_{10} \quad ; \quad MA = BA + EA$$

$$(CX) \leftarrow (MA) \quad \text{or} \quad (CL) \leftarrow (MA)$$

$$(CH) \leftarrow (MA + 1)$$

1.5 Indexed Addressing Mode

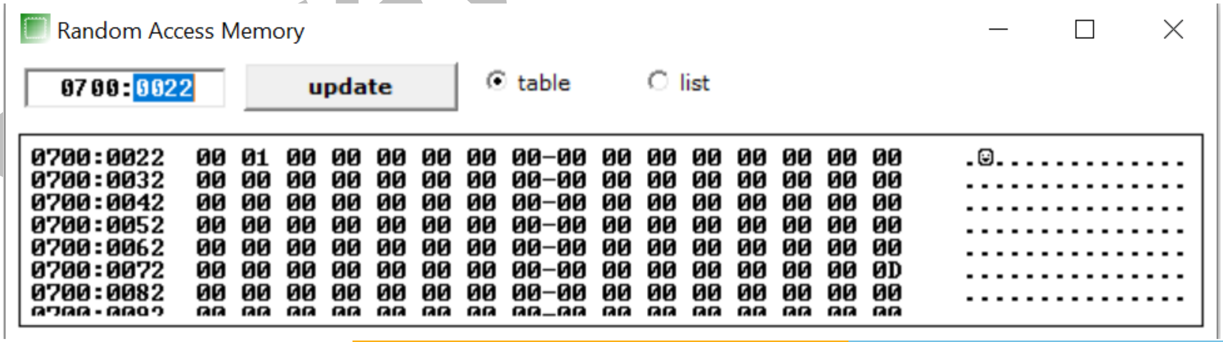
```
org 100h
MOV [0022H], 0100H
MOV SI, 0010H
MOV AX, [SI+012H]
ret
```



DS = 0700h; SI = 0010h

EA = (SI)+012h; BA = DS × 16_{10}

MA = BA + EA; AX ← MA; AL ← MA; AH ← MA + 1; Logical Address: 0700:0022;



1.6 Base Indexed Addressing Mode

- EA is given by sum of base value, index value and an 8-bit or 16-bit displacement specified in the instruction.
- The base value is stored in BX or BP-register.
- The index value is stored in SI or DI-register.

MOV DX, [BX+SI+0AH]

$000A_H \xleftarrow{\text{sign extend}} 0A_H$; $EA = (BX) + (SI) + 000A_H$

$BA = (DS) \times 16_{10}$; $MA = BA + EA$

$(DX) \leftarrow (MA)$ or $(DL) \leftarrow (MA)$

$(DH) \leftarrow (MA + 1)$

CORRECT SYNTAX FORMAT

✓ MOV AX, [BX+SI+20]
✓ MOV AX, [SI+BX+20]
✓ MOV AX, [SI][BX]+20

ILLEGAL SYNTAX FORMAT

× MOV AX, [BX][BP]+20
× MOV AX, [SI][DI]+20

1.6 Base Indexed Addressing Mode

```
org 100h
MOV [0033H], 0110H
MOV SI, 0010H
MOV BX, 0011H
MOV AX, [BX+SI+012H]
ret
```

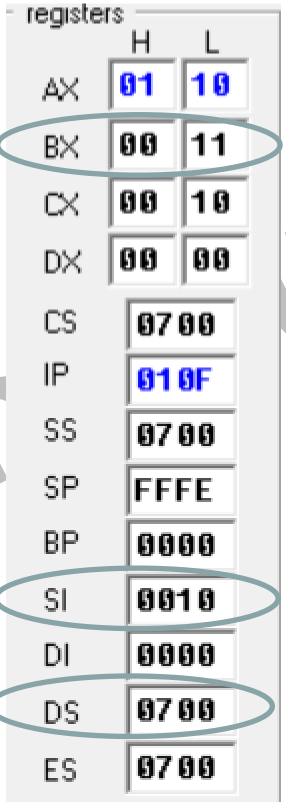
DS = 0700h; SI = 0010h; BX = 0011h

EA = (BX)+(SI)+012h;

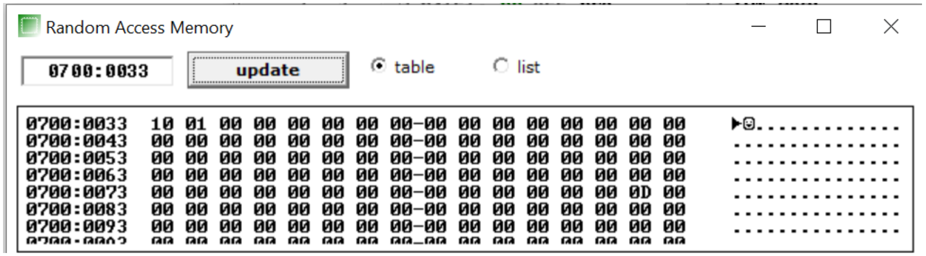
BA = DS × 16_{10}

Logical Address: 0700:0033;

Phycsial Address: 7033h and 7034h



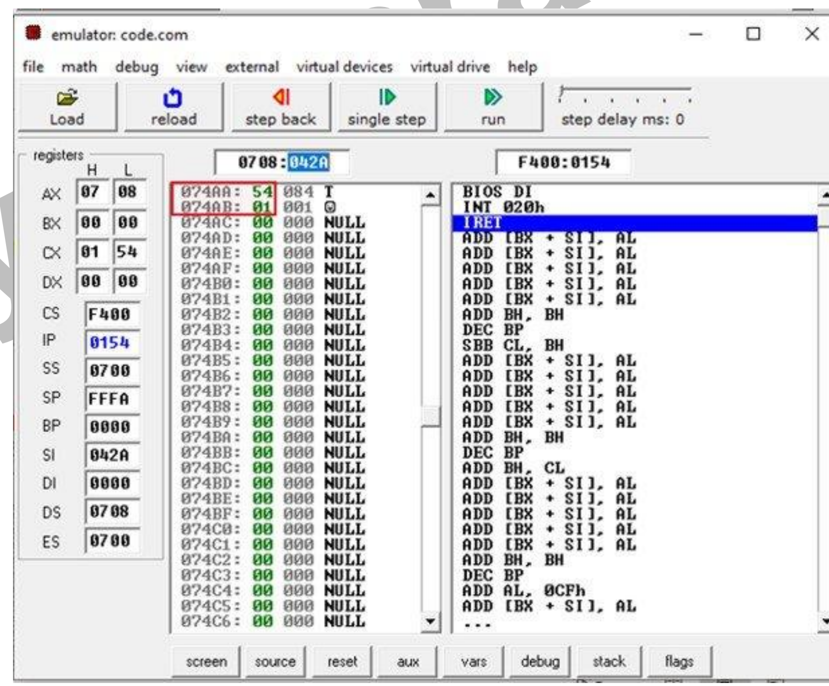
Contents of memory location 0018H moved to AX



1.7 Register Indirect Addressing Mode

- The offset address resides in one of these three registers i.e., BX, SI, DI.
- The sum of offset address and the DS value shifted by one position generates a physical address.

```
ORG 100h
MOV AX, 0708h
MOV DS, AX
MOV CX, 0154h
MOV SI, 42Ah
MOV [SI], CX
RET
```



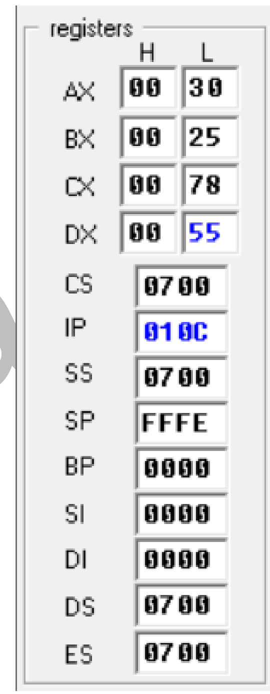
Physical Address: 0708:042Ah

1.8 Implied Addressing Mode

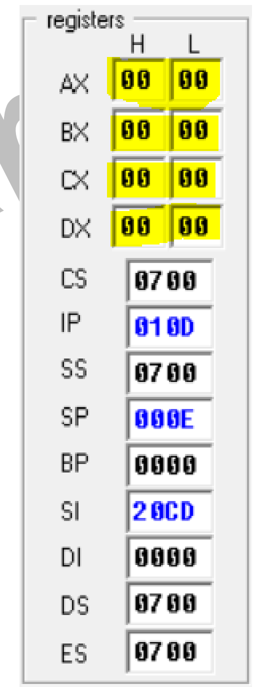
- The instruction itself will specify the data to be operated by the instruction.

```
org 100h
MOV BX , 25h
MOV AX,30h
MOV CX,78h
MOV DX,55h
POPA
RET
```

POPA command will pop all the general purpose registers.



Before POPA



After POPA

1.9 String Addressing Mode

- This addressing mode is related to string instructions.
- In this the value of SI and DI are auto incremented and decremented depending upon the value of directional flag.
- In string addressing mode, the EA of source data is stored in SI-register and the EA of destination data is stored in DI-register

Example :

MOVS BYTE

$EA = (SI)$; $BA = (DS) \times 16_{10}$; $MA = BA + EA$

$EA_E = (DI)$; $BA_E = (ES) \times 16_{10}$; $MA_E = BA_E + EA_E$

$(MA_E) \leftarrow (MA)$

If $DF = 1$, then $(SI) \leftarrow (SI) - 1$ and $(DI) \leftarrow (DI) - 1$

If $DF = 0$, then $(SI) \leftarrow (SI) + 1$ and $(DI) \leftarrow (DI) + 1$

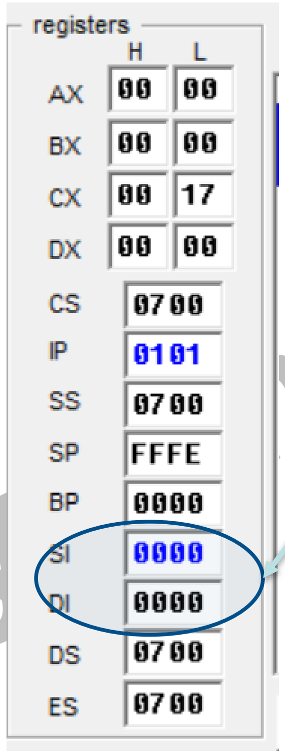
1.9 String Addressing Mode

```
ORG 100h

CLD
LEA SI, a1
LEA DI, a2
MOV CX, 5
REP MOVSB

RET

a1 DB 1,2,3,4,5
a2 DB 5 DUP(0)
```



Check the content of SI and DI after each step.

2.0 IO Port Addressing

- It can be either direct or indirect.
- In direct IO port addressing, the IO address is specified in the instruction.
- In indirect IO port addressing, the instruction will specify the name of the register which holds the port address

Direct IO Port Addressing
IN AL,80H; IN AX,80H; OUT 80H,AL; OUT 80H,AX;

Indirect IO Port Addressing
MOV DX,2080H IN AL,DX; IN AX,DX; OUT DX,AL; OUT DX,AX;

Problems

- Load the operand $(16AC)_{16}$ into register BX using immediate, register addressing mode.
- Using direct addressing mode load the data $(4ECB)_{16}$ in the memory location 3000:171E.
- Let DS = 0300h; SI = 3126h and CX = 4A3Ch. What will be the physical address of the memory location?

  MOV [SI], CX

  Justify your answer with an ALP.
- In the example of string addressing mode, replace the command 'MOVSB' by 'MOVSW'. Show the change in SI and DI after each step of execution. Does it still support string addressing mode?
- Assume DS = 3000h; BX = 1234h, ALPHA = 0012h, SI = 1233h. Determine the addressing mode and calculate and verify the physical addressing of the memory location for the following instructions:

  MOV [BX] + ALPHA, AH

  MOV [SI] + ALPHA, AH

Problems

- Assume: DS = 3000h, BX = 1000h, SI = 1234h and ALPHA = 0012h. Find the physical address for the following instruction:

  MOV AL, [BX] [SI] + ALPHA

  Verify your result with an ALP.

Thankyou