

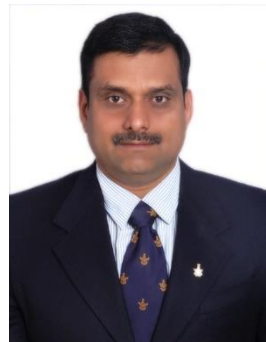


Microprocessors and Interfaces: 2021-22

Lab 4

Addressing Modes in 8086

By Dr. Sanjay Vidhyadharan



Carry, Sign and Overflow Flags



MOV AL,7FH (127₁₀ 01111111₂)

ADD AL,00H **07H** (127₁₀ 01111111₂) CF=0 SF=0 OF=0

MOV AL,7FH

ADD AL,01H **80H** (-128₁₀ 10000000₂) CF=0 SF=1 OF=1

MOV AL,0FFH (-1₁₀ 11111111₂)

ADD AL,02H **01H** (1₁₀ 00000001₂) CF=1 SF=0 OF=0

MOV AL,0FEH (-2₁₀ 11111110₂)

ADD AL,01h **FFH** (-1₁₀ 11111111₂) CF=0 SF=1 OF=0

MOV AL,0FEH (-2₁₀ 11111110₂)

ADD AL,0FFH **FDH** CF=1 SF=1 OF=0

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
```

The screenshot shows a debugger interface with the following components:

- Registers:** A table with columns H and L. CS is 0700 and IP is 0100, both circled in red. Other registers (AX, BX, CX, DX, SS, SP) are also shown.
- Memory:** A list of memory addresses and their contents. The first instruction at 07100: B8 184 7 is highlighted in green.
- Instruction List:** A list of instructions starting with MOV AX, 0005h, MOV BL, 06h, RET, and several NOP instructions.

BYTE 1				BYTE 2		BYTE 3	
		1 BIT	3 BITS	LOW DISP.		HIGH DISP.	
OPCODE		W	REG				
1011		1	000	05		00	

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
```

The screenshot shows a debugger interface with the following components:

- Registers:** AX (H: 00, L: 05), BX (00, 00), CX (00, 06), DX (00, 00), CS (0700), IP (0103, circled in red), SS (0700).
- Memory Window (0700:0103):**
 - 07100: B8 184 7
 - 07101: 05 005 *
 - 07102: 00 000 NULL
 - 07103: B3 179 |
 - 07104: 06 006 ↑
 - 07105: C3 195 ↓
 - 07106: 90 144 É
 - 07107: 90 144 É
 - 07108: 90 144 É
 - 07109: 90 144 É
 - 0710A: 90 144 É
 - 0710B: 90 144 É
 - 0710C: 90 144 É
- Instruction List:**
 - MOV AX, 00005h
 - MOV BL, 06h
 - RET
 - NOP
 - NOP
 - NOP
 - NOP
 - NOP
 - NOP
 - NOP
 - NOP

BYTE 1					BYTE 2		BYTE 3	
			1 BIT	3 BITS	LOW DISP.	HIGH DISP.		
OPCODE			W	REG				
1011			0	011	06			

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: $DS \times 10_{16} + \text{offset} = (21801)_{16}$

Random Access Memory

2162:01E1

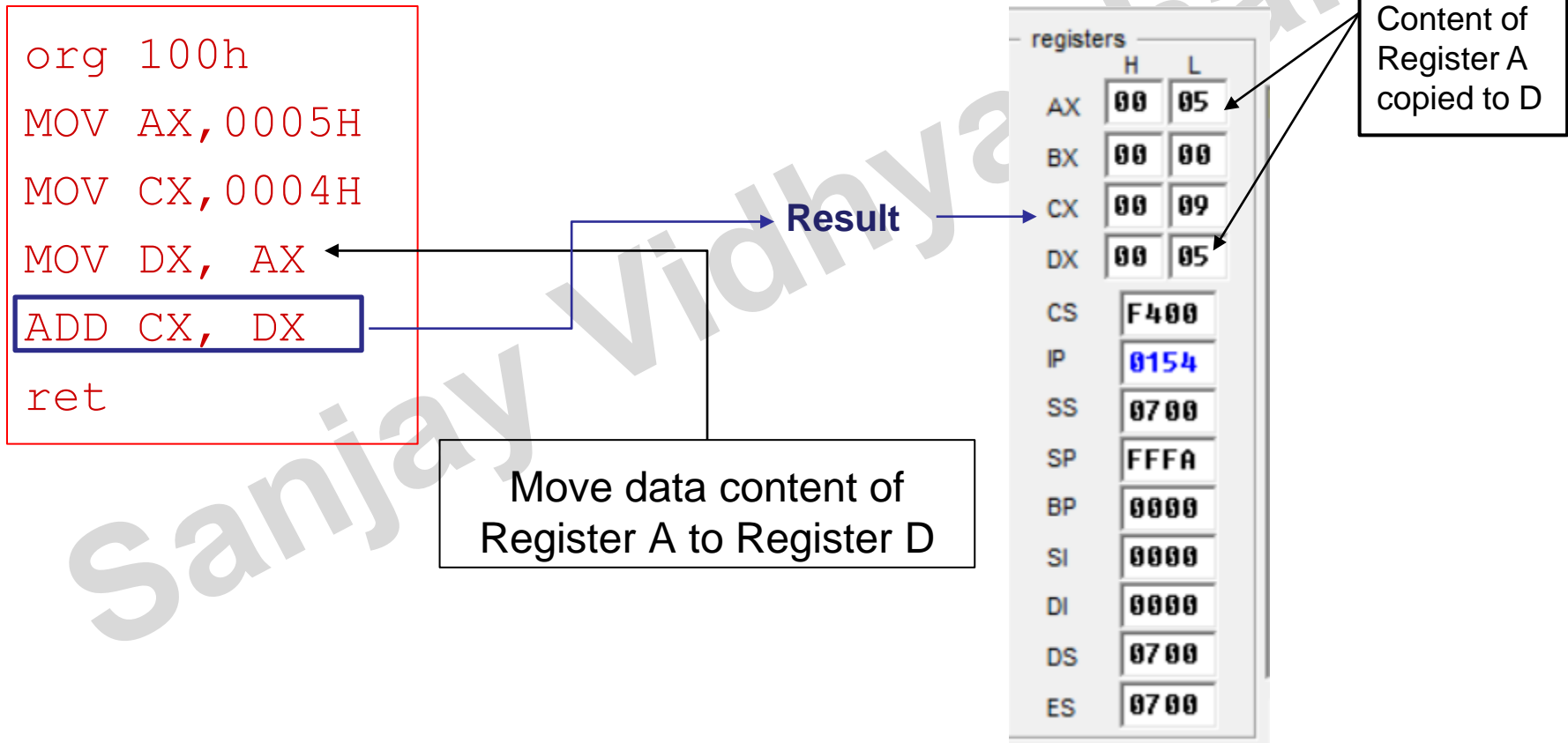
update

2162:01E1	18	00	00	00	00	00
2162:01E1	00	00	00	00	00	00
2162:0201	00	00	00	00	00	00
2162:0201	00	00	00	00	00	00

1.3 Register Addressing Mode

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

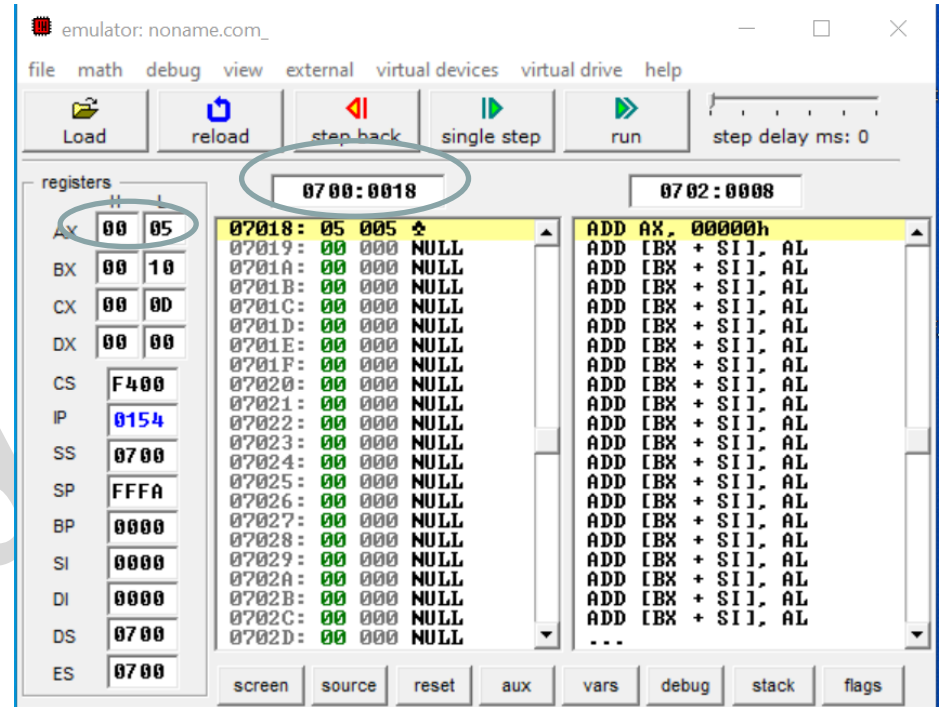
All except IP register can be used in this mode.



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

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

```
DS = 0700h; SI = 0010h
EA = (SI)+012h; BA = DS × 1610
MA = BA + EA; AX ← MA; AL ← MA; AH ← MA
+ 1; Logical Address: 0700:0022;
```

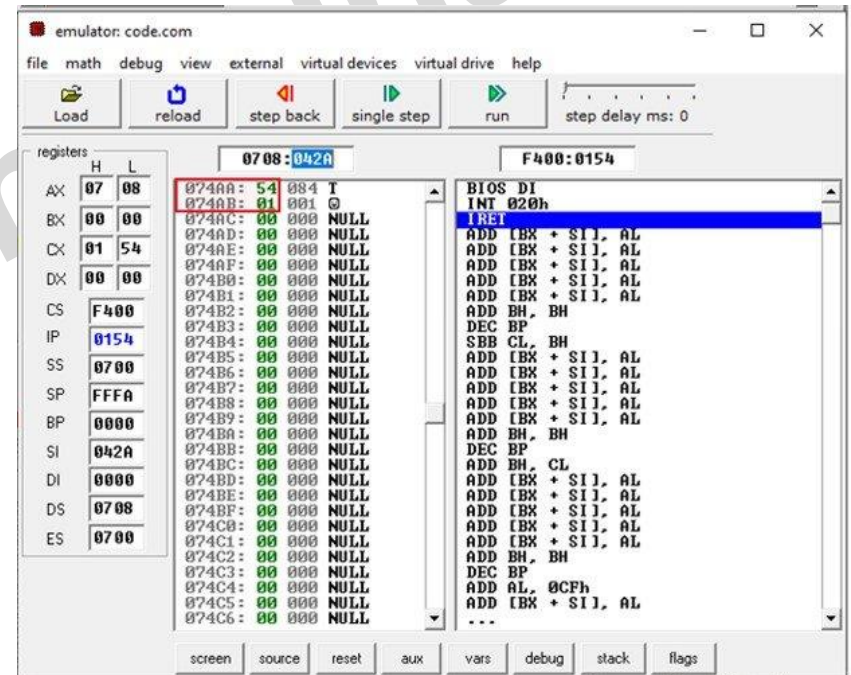
registers		
	H	L
AX	01	00
BX	00	00
CX	00	00
DX	00	00
CS	0700	
IP	0000	
SS	0700	
SP	0000	
BP	0000	
SI	0010	
DI	0000	
DS	0700	
ES	0700	

Random Access Memory																
0700:0022 update table list																
0700:0022	00	01	00	00	00	00	00	00-00	00	00	00	00	00	00	00
0700:0032	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00
0700:0042	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00
0700:0052	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00
0700:0062	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00
0700:0072	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00
0700:0082	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00
0700:0092	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00

• 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.

registers		
	H	L
AX	00	30
BX	00	25
CX	00	78
DX	00	55
CS	0700	
IP	010C	
SS	0700	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

registers		
	H	L
AX	00	00
BX	00	00
CX	00	00
DX	00	00
CS	0700	
IP	010D	
SS	0700	
SP	000E	
BP	0000	
SI	20CD	
DI	0000	
DS	0700	
ES	0700	

Before POPA

After POPA

• 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)
```

registers		
	H	L
AX	00	00
BX	00	00
CX	00	17
DX	00	00
CS	07 00	
IP	01 01	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

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.

Sanjay Vidhyadharan

-
- Thankyou

Sanjay Vidhyadharan