Addressing Modes

Lab Activity 4

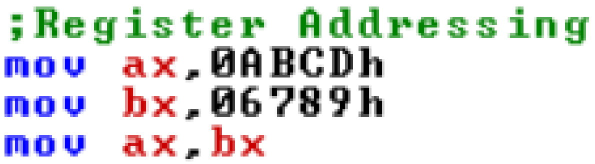
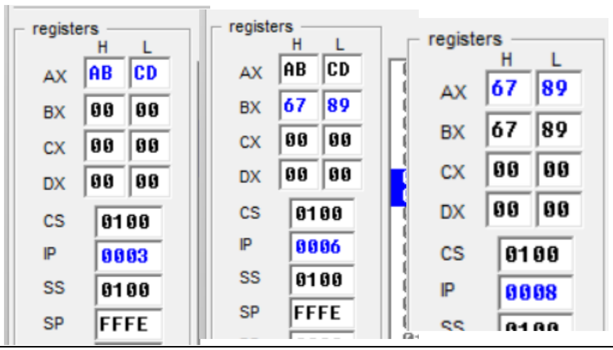
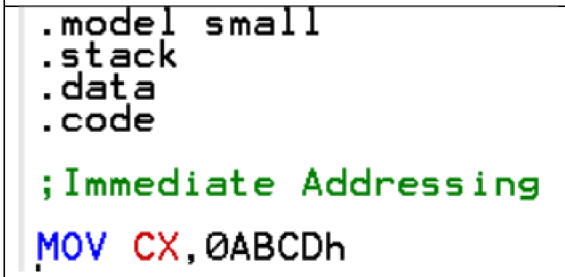
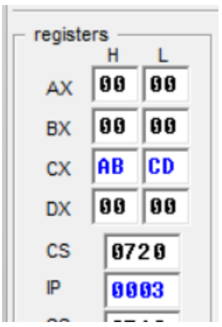
Dave Jhared G. Paduada **March 11, 2025**

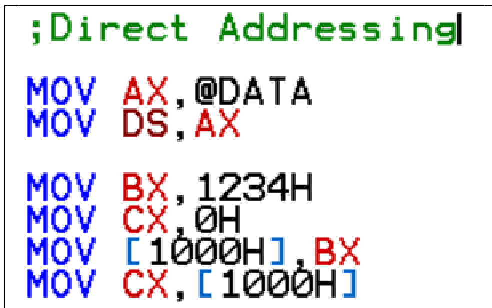
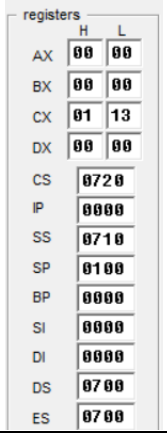
BSCpE III - GF

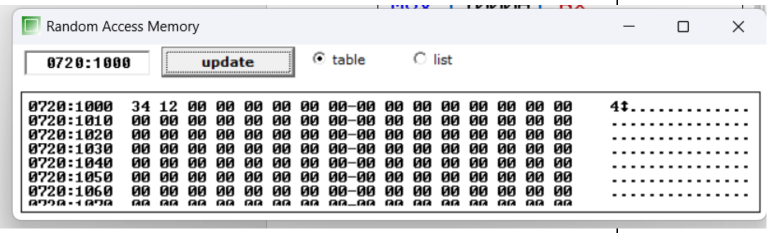
Run the following codes.

Make a screenshot of the following after executing the instructions.

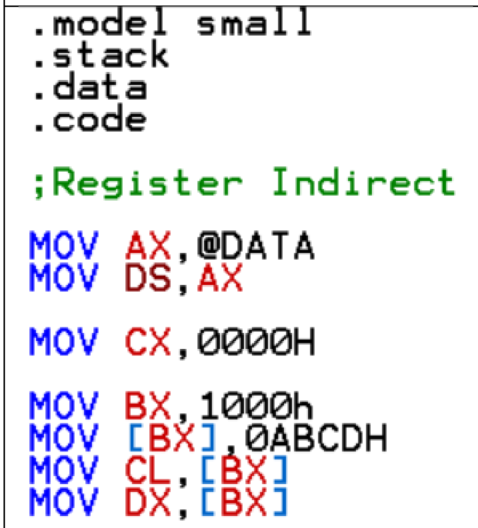
a. registers used
b. memory location being accessed
   i. memory
   ii. variable

Addressing Mode	Screenshot
 ;Register Addressing mov ax,0ABCDh mov bx,06789h mov ax,bx	Register  registers H L AX AB CD BX 00 00 CX 00 00 DX 00 00 CS 0100 IP 0003 SS 0100 SP FFFE registers H L AX AB CD BX 67 89 CX 00 00 DX 00 00 CS 0100 IP 0006 SS 0100 SP FFFE registers H L AX 67 89 BX 67 89 CX 00 00 DX 00 00 CS 0100 IP 0008 SS 0100
 .model small .stack .data .code ;Immediate Addressing MOV CX,0ABCDh	Register  registers H L AX 00 00 BX 00 00 CX AB CD DX 00 00 CS 0720 IP 0003

 ;Direct Addressing MOV AX,@DATA MOV DS,AX MOV BX,1234H MOV CX,0H MOV [1000H],BX MOV CX,[1000H]	Register and Memory  registers H L AX 00 00 BX 00 00 CX 01 13 DX 00 00 CS 0720 IP 0000 SS 0710 SP 0100 BP 0000 SI 0000 DI 0000 DS 0700 ES 0700

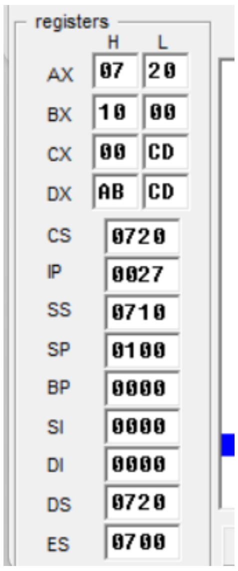


```
0720:1000  34 12 00 00 00 00 00 00-00 00 00 00 00 00 00 00    4‡..............
0720:1010  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00    ................
0720:1020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00    ................
0720:1030  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00    ................
0720:1040  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00    ................
0720:1050  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00    ................
0720:1060  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00    ................
0720:1070  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```

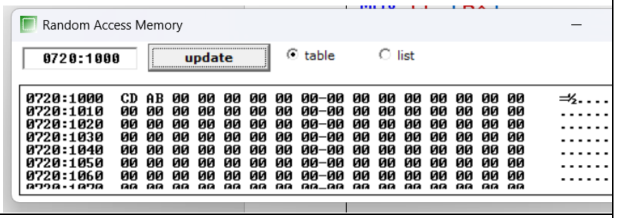


```
.model small
.stack
.data
.code

;Register Indirect

MOV AX,@DATA
MOV DS,AX

MOV CX,0000H

MOV BX,1000h
MOV [BX],0ABCDH
MOV CL,[BX]
MOV DX,[BX]
```

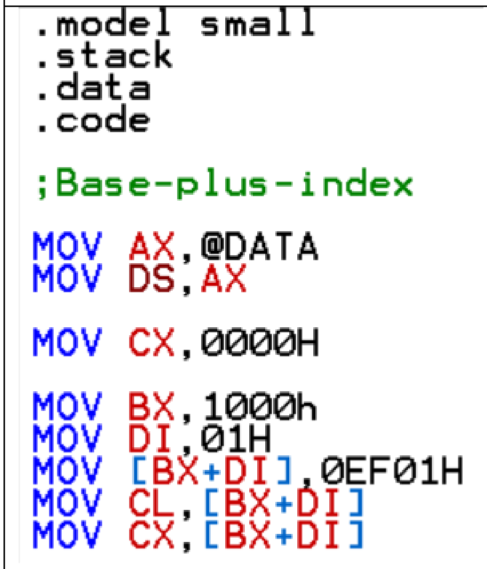
Register and Memory



```
AX  07 20
BX  10 00
CX  00 CD
DX  AB CD
CS  0720
IP  0027
SS  0710
SP  0100
BP  0000
SI  0000
DI  0000
DS  0720
ES  0700
```

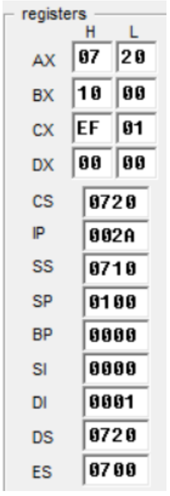


```
0720:1000  CD AB 00 00 00 00 00 00-00 00 00 00 00 00 00 00    ⇒½....
0720:1010  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00    ......
0720:1020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00    ......
0720:1030  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00    ......
0720:1040  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00    ......
0720:1050  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00    ......
0720:1060  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00    ......
0720:1070  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```

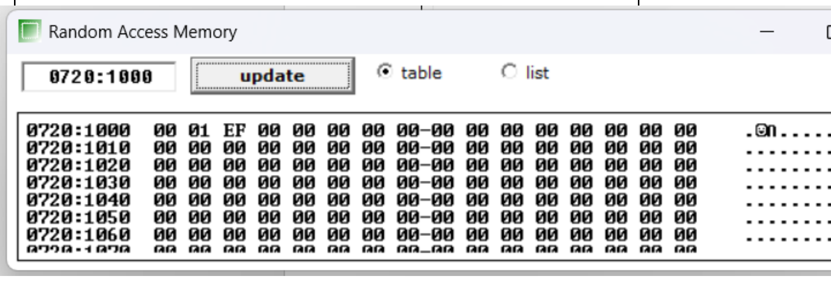


```
.model small
.stack
.data
.code

;Base-plus-index

MOV AX,@DATA
MOV DS,AX

MOV CX,0000H

MOV BX,1000h
MOV DI,01H
MOV [BX+DI],0EF01H
MOV CL,[BX+DI]
MOV CX,[BX+DI]
```

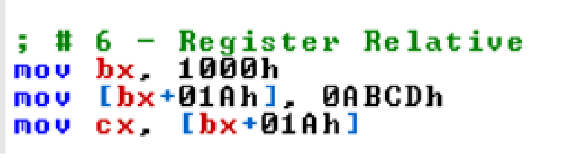
Register and Memory



```
AX  07 20
BX  10 00
CX  EF 01
DX  00 00
CS  0720
IP  002A
SS  0710
SP  0100
BP  0000
SI  0000
DI  0001
DS  0720
ES  0700
```

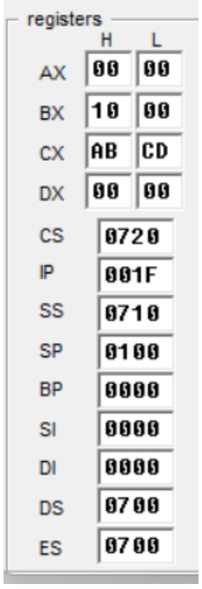


```
0720:1000  00 01 EF 00 00 00 00 00-00 00 00 00 00 00 00 00    .☺n......
0720:1010  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00    .........
0720:1020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00    .........
0720:1030  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00    .........
0720:1040  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00    .........
0720:1050  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00    .........
0720:1060  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00    .........
0720:1070  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```

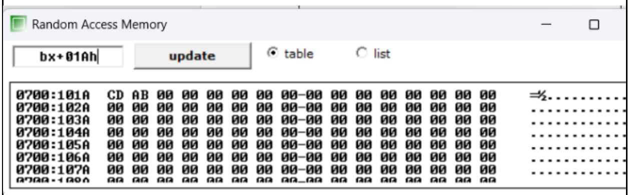


```
; # 6 - Register Relative
mov bx, 1000h
mov [bx+01Ah], 0ABCDh
mov cx, [bx+01Ah]
```

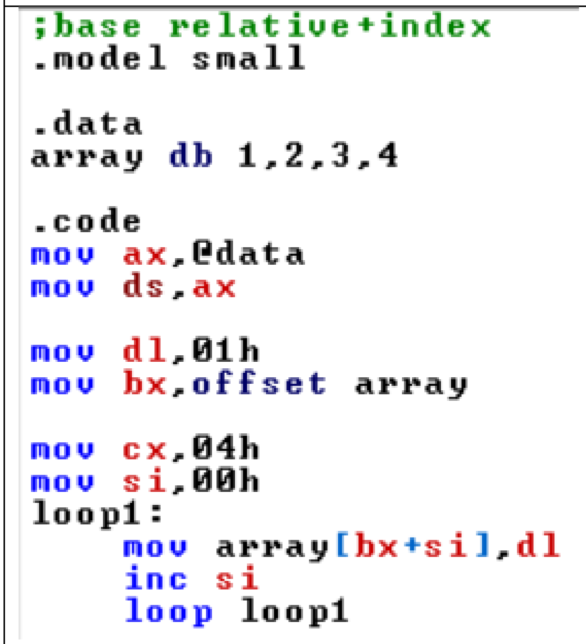
Register and Memory



```
AX  00 00
BX  10 00
CX  AB CD
DX  00 00
CS  0720
IP  001F
SS  0710
SP  0100
BP  0000
SI  0000
DI  0000
DS  0700
ES  0700
```



```
bx+01Ah

0700:101A  CD AB 00 00 00 00 00 00-00 00 00 00 00 00 00 00   =½..............
0700:102A  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0700:103A  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0700:104A  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0700:105A  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0700:106A  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0700:107A  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0700:108A  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```

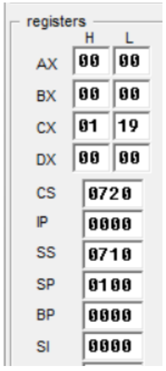


```
;base relative+index
.model small

.data
array db 1,2,3,4

.code
mov ax,@data
mov ds,ax

mov dl,01h
mov bx,offset array

mov cx,04h
mov si,00h
loop1:
      mov array[bx+si],dl
      inc si
      loop loop1
```

Register, Memory and Array/Variable **(Before and After)**

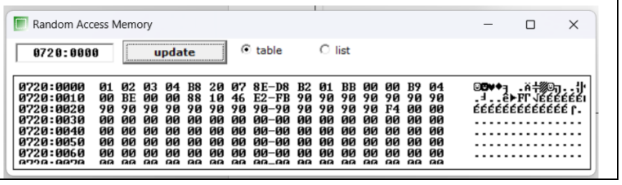
Before:

Register:



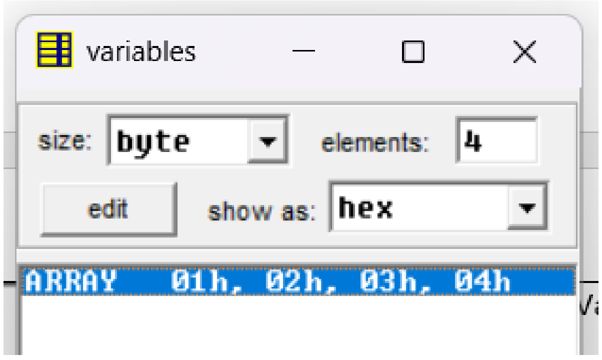
```
AX  00 00
BX  00 00
CX  01 19
DX  00 00
CS  0720
IP  0000
SS  0710
SP  0100
BP  0000
SI  0000
```

Memory:



```
0720:0000

0720:0000  01 02 03 04 B8 20 07 8E-D8 B2 01 BB 00 00 B9 04   ☺☻♥♦╕ .Ä╪▓☺╗..╣♦
0720:0010  00 BE 00 00 88 10 46 E2-FB 90 90 90 90 90 90 90   .╛..ê►FΓ√ÉÉÉÉÉÉÉ
0720:0020  90 90 90 90 90 90 90 90-90 90 90 90 90 F4 00 00   ÉÉÉÉÉÉÉÉÉÉÉÉÉ⌠..
0720:0030  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0720:0040  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0720:0050  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0720:0060  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0720:0070  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```

Array:



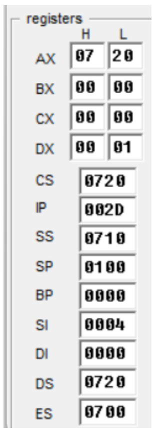
size: byte elements: 4

show as: hex

```
ARRAY    01h, 02h, 03h, 04h
```

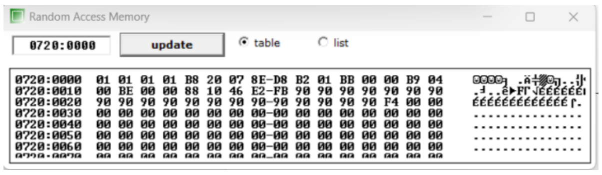
After:

Register:



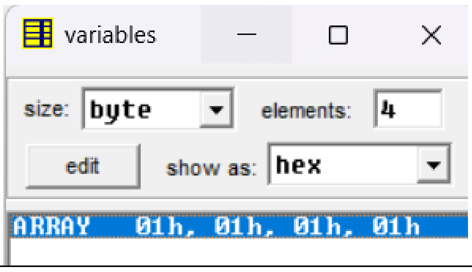
```
     H  L
AX  07 20
BX  00 00
CX  00 00
DX  00 01
CS  0720
IP  002D
SS  0710
SP  0100
BP  0000
SI  0004
DI  0000
DS  0720
ES  0700
```

Memory:



```
0720:0000
0720:0000  01 01 01 01 B8 20 07 8E-D8 B2 01 BB 00 00 B9 04   ☺☺☺☺╕ .Ä╪▒☺╗..╣♦
0720:0010  00 BE 00 00 88 10 46 E2-FB 90 90 90 90 90 90 90   .╛..ê►FΓ√ÉÉÉÉÉÉÉ
0720:0020  90 90 90 90 90 90 90 90-90 90 90 90 90 F4 00 00   ÉÉÉÉÉÉÉÉÉÉÉÉÉ⌠..
0720:0030  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0720:0040  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0720:0050  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0720:0060  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0720:0070  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```

ARRAY



size: byte elements: 4

show as: hex

```
ARRAY    01h, 01h, 01h, 01h
```