

Режими адресації МП 8086:

1. Адресація регістрів
2. Безпосередня адресація
3. Пряма
4. Непряма регістрова
5. Адресація за базою
6. Пряма адресація з індексуванням
7. Адресація за базою з індексуванням

Адресація регістрів

В цьому випадку операндою є вміст визначеного
регістру.

MOV AL, BH

BH



AL

MOV AX, CX

CX



AX

; Приклад

INC CH ;Плюс 1 до вмісту CH

PUSH DS ;DS зберігається в стеку

XCHG BX, BP ;BX і BP обмінюються вмістом

MOV ES, AX ;вміст AX пересилається в ES

Безпосередня адресація

В цьому випадку замість операнда джерела використовується безпосередньо **константа**:

```
MOV AX, 60
```

```
60
```

```
→
```

```
AX
```

```
MOV CL, -50
```

```
-50
```

```
→
```

```
AX
```

```
L EQU 256
```

```
.....
```

```
MOV CX,L
```

```
256
```

```
→
```

```
AX
```

Пряма адресація

Як відмічалось, у покажчику команди **IP** зберігається відносна адреса команди в сегменті, тобто кількість байт відносно його початку чи **виконавча адреса**.

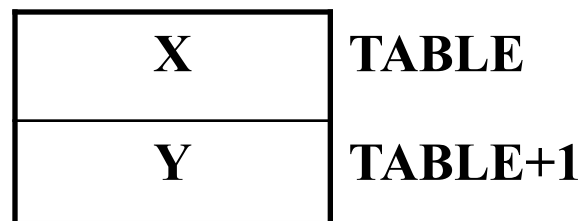
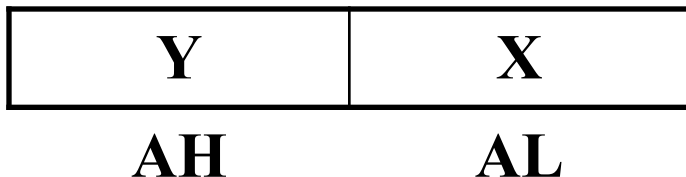
Для прямої адресації виконавча адреса відмічається **безпосередньо в команді**.

TABLE DW 1560;	в комірку TABLE записано 1560
INDEX DB -126;	в байт INDEX записано -126

Тоді можна записати:

MOV AX, TABLE ; переслати вміст TABLE в акумулятор

Відмітимо особливості такої пересилки
Коли в пам'яті було записано



Тобто, байти помінялися місцями
(зміщення записується до
команди)

Непряма адресація регістрів

Даний спосіб адресації використовує базовий регістр **BX**, покажчик **BP** і індексні регістри **SI**, **DI**, де записана адреса операнда:

```
MOV AX,[BX];
```

А як в регістр **BX** занести адресу комірки, наприклад, TABLE? Це можна зробити за допомогою операції OFFSET (зміщення).

```
MOV BX, OFFSET TABLE  
;або  
LEA BX, TABLE
```

Порівняйте: MOV BX, TABLE.



Адресація за базою

```
MOV AX, [BP] + N
```

N - кількість байт від початку масиву (зміщення)

Еквівалентні форми:

```
MOV AX, 2 [BP]  
MOV AX, [BP + 2]  
MOV AX, [BP - 6+4]
```

Тобто, обравши необхідне зміщення, можна довільно адресувати елементи масиву.



Пряма адресація з індексуванням

Якщо зафіксувати базову адресу елементів даних визначеною міткою, тоді, добратися до інших елементів даних можна за допомогою індексних реєстрів **SI**, **DI**.

```
MOV DI, 2  
MOV AX, TABLE
```

еквівалентне :

```
MOV AX, [TABLE]+2
```

Адресація за базою з індексуванням

Для обробки двомірних масивів зручно використовувати адресацію за базою з індексуванням, коли виконавча адреса дорівнює сумі значень базового реєстру, індексного реєстру та зміщення.

```
MOV AX, VALUE [BX] [DI]
```

Тут **VALUE** – ім'я змінної, а не адреса комірки. Замість імені змінної, можна задавати адресну константу. Наприклад:

```
MOV AX, 2[BP] [SI]
```

Операнди в дужках можна записувати по-різному:

```
MOV AX [BP + 2 + SI];  
MOV AX [SI + BP + 2];  
MOV AX [BP] [SI + 2].
```