



## 2.2 8088/8086的寻址方式



# 本节学习内容

- ◇ 熟悉8088/8086汇编语言指令格式
  - ◇ 掌握操作数的寻址方式
- 为8088/8086指令系统的学习做好准备

# 指令的基本格式



操作码

操作数

指令由操作码和操作数两部分组成

- ◇ 操作码定义了计算机要执行的操作，如传送、运算、移位、跳转等，它是指令中不可缺少的组成部分。
- ◇ 操作数是指令操作的对象。
- ◇ 有些指令不需要操作数，通常的指令都有一个或两个操作数。

# 8086指令助记符格式（微软宏汇编，Intel风格）

10:58<sup>AM</sup><sub>33</sub>

操作码    操作数1， 操作数2    ; 注释

**ADD AX, BX** ; 将**AX**与**BX**中的数据相加，结果存入**AX**

- ◇ 操作数2，称为**源操作数src**，它给出一个参与指令操作的对象（数据本身或储存数据的位置信息）
- ◇ 操作数1，称为**目的操作数dest**，运算开始时提供指令的一个操作对象，运算完成后用来指明结果存放的位置
- ◇ 分号及其后的内容是注释

# 指令的操作码和操作数

## ◇ 操作码

- ◆ 用一个助记符表示（指令功能的英文缩写）
- ◆ 对应着机器指令的一个或多个二进制编码（助记符相同，机器内部操作不同，如**call**和**ret**）

## ◇ 操作数

- ◆ 可以是一个具体的数值      `add ax,5`
- ◆ 可以是存放数据的寄存器      `add bx,cx`
- ◆ 或数据在主存的地址信息      `add dx,wvar`

# 操作数的寻址方式

- ◇ 操作数的寻址方式是指计算操作数地址的方法
- ◇ 理解操作数的寻址方式是理解指令功能从而能够合理、灵活运用指令编程的前提
- ◇ 采用合理的寻址方式有助于提高程序的效率
- ◇ 操作数可以编码在指令中，也可以存储在通用寄存器或主存中，接下来我们分类学习8086的各种寻址方式。

# 1.立即数寻址

在立即数寻址方式中，指令地址码字段直接给出操作数。在8086系统中，立即数可以为8位或16位。

例如：

MOV AL,80H ; 将十六进制数80H送入AL

MOV AX,1090H ; 将1090H送AX，AH中为10H，  
AL中为90H

演示

# 1.立即数寻址(续)

- ◇ 应用：主要用来对寄存器赋值。
- ◇ 特点：速度快。

注意：

- ①立即数可以为8位，也可以为16位；
- ②规定立即数只能是整数，不能是小数、变量或者其他类型的数据；
- ③立即数只能作为源操作数。



## 2. 寄存器寻址

采用寄存器寻址方式时，操作数从指令地址码指定的通用寄存器中取得。

16位操作数: AX, BX, CX, DX, SI, DI, SP, BP

8位操作数: AH, AL, BH, BL, CH, CL, DH, DL

例如:

INC CX ; 将CX的内容加1

ROL AH, 1 ; 将AH中的内容循环左移一位

演示

## 2. 寄存器寻址(续)

特点:

执行速度快。采用寄存器寻址方式的指令在执行时,操作完全在**CPU**内部进行,不需要启动总线周期。

应用:

编程时可对源、目的操作数中的任意一个或者二者同时使用寄存器寻址方式。

# 存储器中操作数的寻址

- ◇ 操作数在主存中，指令中对主存地址进行描述
- ◇ 8086汇编语言程序中采用逻辑地址表示主存地址
  - ◆ 段地址在默认的或用段超越前缀指定的段寄存器中
  - ◆ 指令中一般只需给出操作数的偏移地址（有效地址EA）
- ◇ 8086设计了多种存储器寻址方式
  - 1、直接寻址方式
  - 2、寄存器间接寻址方式
  - 3、寄存器相对寻址方式
  - 4、基址变址寻址方式
  - 5、相对基址变址寻址方式

## 8088地址变换

### ◇ 逻辑地址

格式：段基址:段内偏移地址

例：1234H:5678H

### ◇ 从逻辑地址计算物理地址的方法

$$\begin{array}{r} 12340H \\ + 5678H \\ \hline 179B8H \end{array}$$

物理地址

### 3.直接寻址

数据位于存储器中，指令地址码字段直接给出操作数的有效地址。

例如：

**MOV AX, [1070H] ; (MOV AX,WVAR)**

；将存储在**DS**段**1070H**和**1071H**两单元的字数据取到**AX**中

**注意：**采用直接寻址方式时，如果指令中没有用段超越前缀指明操作数在哪一个段，则默认在数据段，段寄存器是**DS**。

演示

### 3. 直接寻址：段超越

如果要对默认的段之外的存储区进行直接寻址，则指令中必须用段超越前缀指出数据所在的段。

例如：

**MOV BX, CS:[3000H]** ; 表示数据在代码段

设**CS**为**5100H**，则本指令在执行时，将**54000H**和**54001H**两单元的内容取出送**BX**。

## 4. 寄存器间接寻址

- ◇ 操作数所在存储单元的有效地址存储在指令地址码字段给定的寄存器中。
- ◇ 可作为间址寄存器的通用寄存器包括BX，BP，SI和DI。

例：

**MOV AX,[BX]**

演示

## 4. 寄存器间接寻址:物理地址的计算

- ◆ 当间址寄存器为**BX**、**SI**、**DI**时，段地址在数据段寄存器**DS**中。

物理地址的计算方法，以**BX**作间址寄存器为例：

$$\text{物理地址} = 16 \times (\text{DS}) + (\text{BX})$$

间址寄存器为**SI**、**DI**时，计算方法相同。

- ◆ 当间址寄存器为**BP**时，段地址在堆栈段寄存器**SS**中。物理地址的计算方法为：

$$\text{物理地址} = 16 \times (\text{SS}) + (\text{BP})$$



## 4. 寄存器间接寻址:例题

[例] 若  $(DS) = 2100H$ ,  $(DI) = 2000H$ ,  
求指令 `MOV AX, [DI]` 源操作数的物理地址。

物理地址  $= 16 \times (DS) + (DI) = 16 \times 2100H + 2000H = 23000H$

指令执行结果是将内存  $23000H$  和  $23001H$  单元的字数据送入寄存器 **AX** 中。

注意:

如果操作数不在数据段中, 需要在指令中用段超越前缀表明其所在的段。

## 5. 寄存器相对寻址方式

操作数在存储器中，将指令地址码字段指定的寄存器的内容与指令指定的位移量（**DISP**）之和作为操作数所在存储单元的有效地址。

寄存器可以是基址寄存器**BX**、**BP**，也可以是变址寄存器**SI**、**DI**。位移量是一个8位（**DISP8**）或16位（**DISP16**）的带符号二进制数。

**MOV AX, [SI+06H] ; AX ← DS:[SI+06H]**

**MOV AX, 06H[SI] ; AX ← DS:[SI+06H]**

演示

## 5. 寄存器相对寻址方式:举例

**[例]** 已知 (DS) = 2000H, (SI) = 1000H, DA1 = 2000H,  
则指令:

MOV BX, DA1[SI]

或 MOV BX, [DA1+SI]

源操作数的物理地址 =  $16 \times (\text{DS}) + (\text{SI}) + \text{DISP16}$

= 20000H + 1000H + 2000H = 23000H

## 6. 基址变址寻址方式

- ◇ 操作数在存储器中，指令将基址寄存器（**BX**或**BP**）与变址寄存器（**SI**或**DI**）内容之和作为操作数所在存储单元的有效地址**EA**。
- ◇ 当使用基址寄存器**BX**时，段寄存器为**DS**，物理地址计算方法为 物理地址=  $16 \times (\text{DS}) + (\text{BX}) + (\text{SI})$ （使用寄存器**DI**类同）
- ◇ 当使用基址寄存器**BP**时，段寄存器为**SS**，物理地址计算方法为物理地址=  $16 \times (\text{SS}) + (\text{BP}) + (\text{SI})$ （使用寄存器**DI**类同）

演示

**MOV AX, [BX+SI] ; AX ← DS:[BX+SI]**

**MOV AX, [BX][SI] ; AX ← DS:[BX+SI]**

## 6. 基址变址寻址方式(续)

[例] 已知  $(DS) = 3200H$ ,  $(BX) = 1234H$ ,  $(SI) = 3456H$ ,

计算指令  $MOV\ AL, [BX][SI]$

或  $MOV\ AL, [BX+SI]$

源操作数的有效地址和物理地址。

有效地址  $EA = (BX) + (SI) = 1234H + 3456H = 468AH$

物理地址  $= 16 \times (DS) + EA = 32000H + 468AH = 3668AH$

指令执行结果是将3668AH单元的内容送入寄存器AL中。

## 7. 相对基址变址寻址方式

◇ 将指令地址码字段给定的基址寄存器 (**BX**或**BP**) 的内容与变址寄存器 (**SI**或**DI**) 的内容及位移量 (**8位**或**16位**) 三者之和作为操作数的有效地址。

◇ 当使用基址寄存器**BX**时，默认段寄存器为**DS**，物理地址计算方法为

$$\text{物理地址} = 16 \times (\text{DS}) + (\text{BX}) + (\text{SI}) + \text{DISP}$$

◇ 当使用基址寄存器**BP**时，默认段寄存器为**SS**，物理地址计算方法为

$$\text{物理地址} = 16 \times (\text{SS}) + (\text{BP}) + (\text{SI}) + \text{DISP}$$

演示

## 7. 相对基址变址寻址方式:地址变换

[例] 已知 (DS) = 2000H, (BX) = 1000H, (SI) = 0500H, MK = 1120H, 计算如下指令源操作数的物理地址。

MOV AX, MK [BX] [SI]

或 MOV AX, MK [BX+SI]

或 MOV AX, [MK + BX + SI]

物理地址 =  $16 \times (\text{DS}) + (\text{BX}) + (\text{SI}) + \text{DISP}$

= 20000H + 1000H + 0500H + 1120H = 22620H

指令执行结果是将22620H、22621H单元的内容送入寄存器AX中。

# 寻址方式综合应用举例

关于DEC公司的一条记录(1982年)如下

- ◇ 公司名称: DEC
- ◇ 总裁姓名: Ken Olsen
- ◇ 排名: 137
- ◇ 收入: 40亿美元
- ◇ 著名产品: PDP

之后公司信息发生如下变化:

- ◇ 排名升至38
- ◇ 收入增加了70亿美元
- ◇ 著名产品变为VAX

要求编程将变化的部分进行修改。

返回



# 寻址方式综合应用

```
struct company
{
    char name[10];
    char host[10];
    int rank;
    int revenue;
    char prod[10];
}
```

```
company dec={"DEC","Ken Olsen",137,40,"PDP"};

int main(void)
{
    int i;
    dec.rank = 38;
    dec.revenue = dec.revenue + 70;
    i = 0;
    dec.prod[i]='V';
    i++;
    dec.prod[i]='A';
    i++;
    dec.prod[i]='X';
    return 0;
}
```

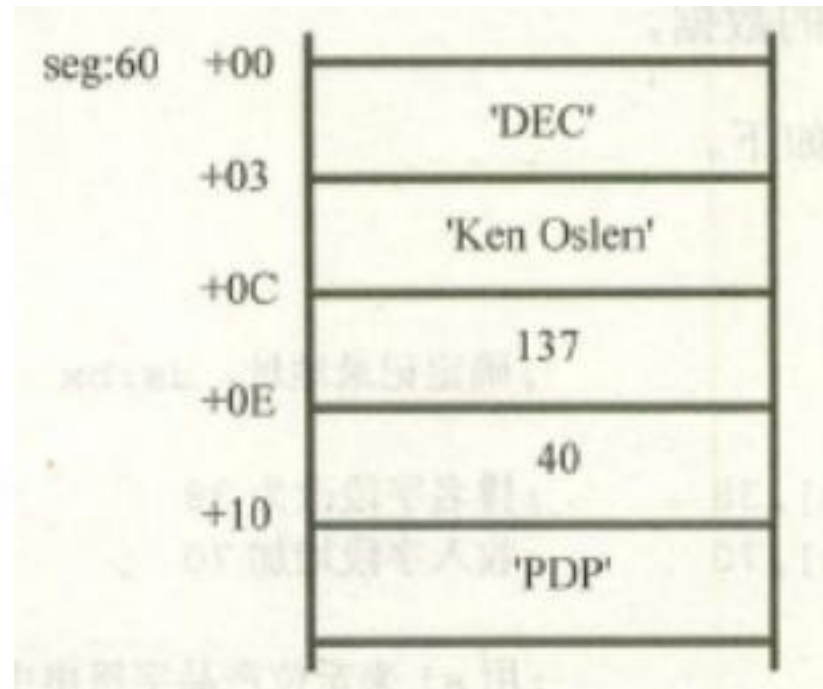
[返回](#)

# 寻址方式综合应用

用汇编来描述

Dec.asm

```
mov ax,seg
mov ds,ax
mov bx,60h
mov word prt [bx].0ch,38
add word prt [bx].0eh,70
mov si,0
mov [bx].10h[si],'V'
inc si
mov [bx].10h[si],'A'
inc si
mov [bx].10h[si],'X'
```



题目

# 与转移指令有关的寻址方式

◇ 留待结合程序转移指令学习

# 参考资料



# MOV指令的功能

MOV dest,src

dest



src



播放



## 立即数寻址方式



**MOV AX,0102H**



## 寄存器寻址方式

**AX**

--	--

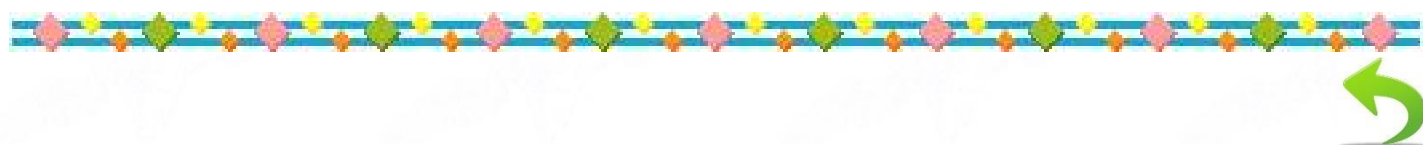
  
**AH AL**

**BX**

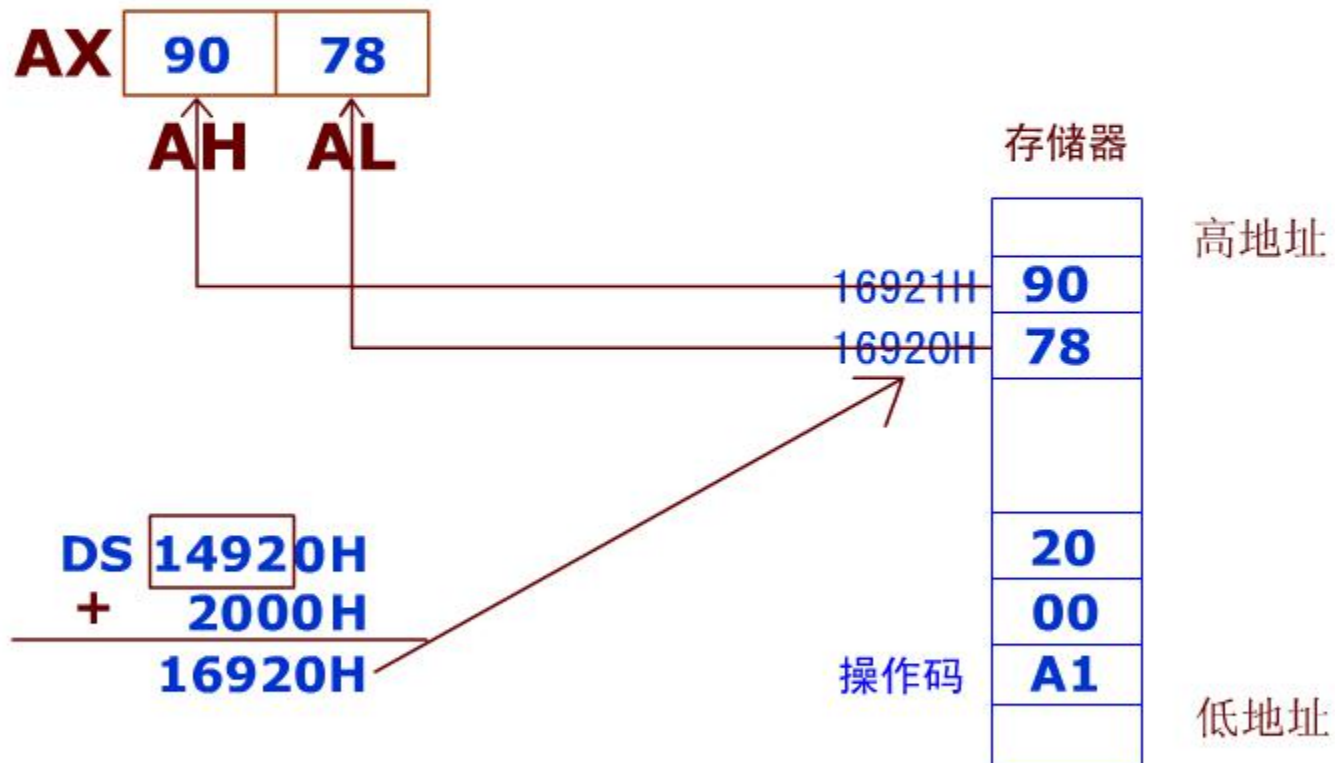
<b>1234</b>
-------------

  
**BH BL**

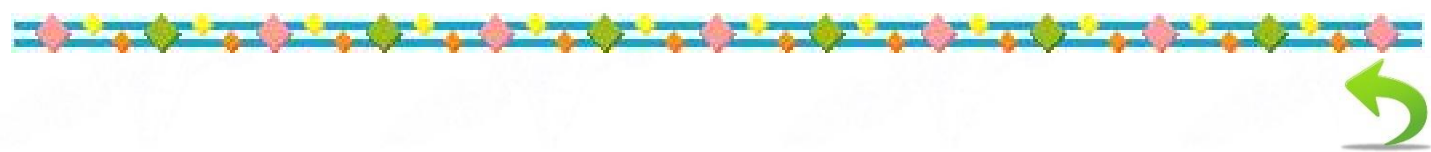
**MOV AX,BX**



## 直接寻址方式

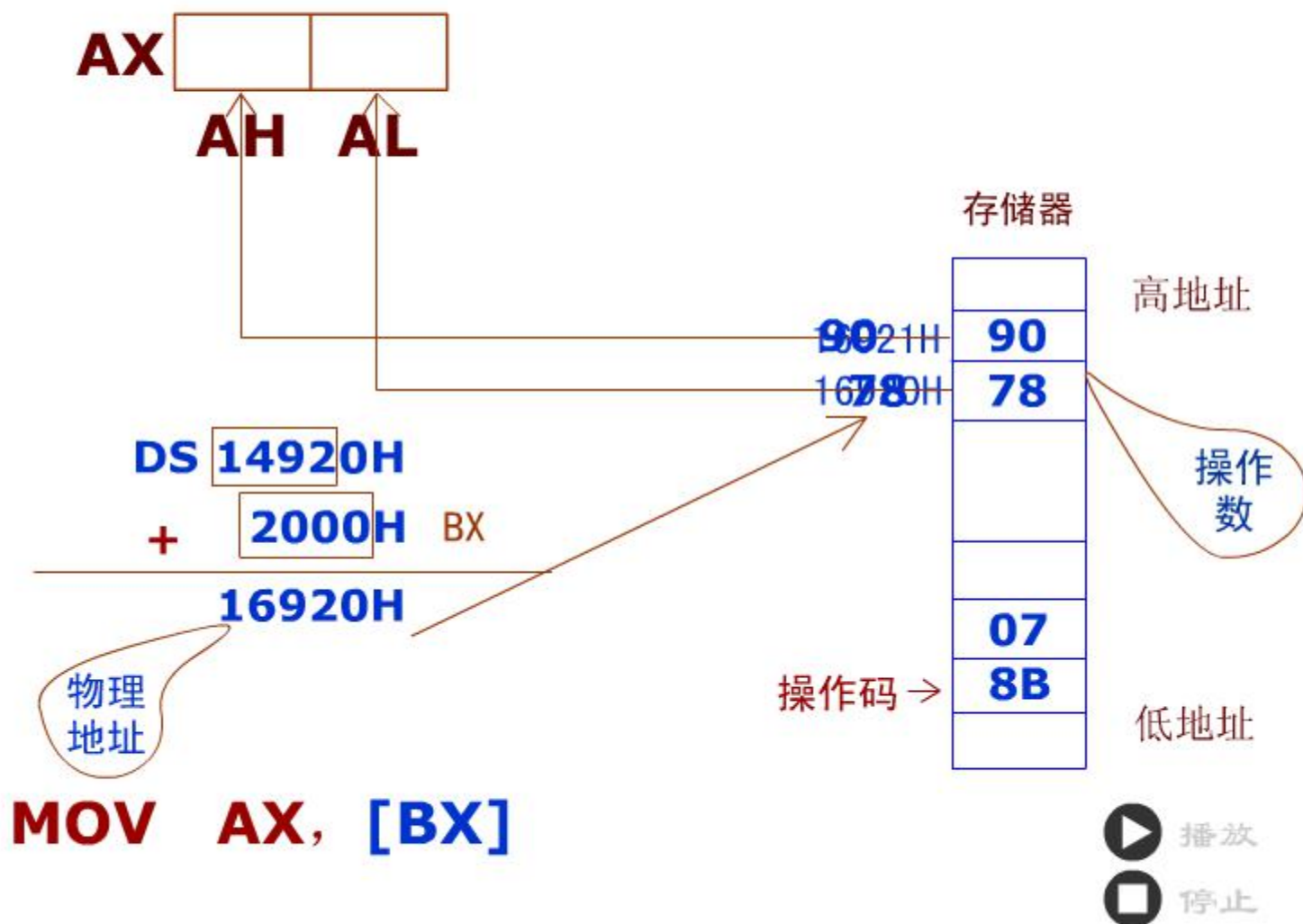


**MOV AX, [2000H]**





## 寄存器间接寻址方式



## 寄存器相对寻址方式

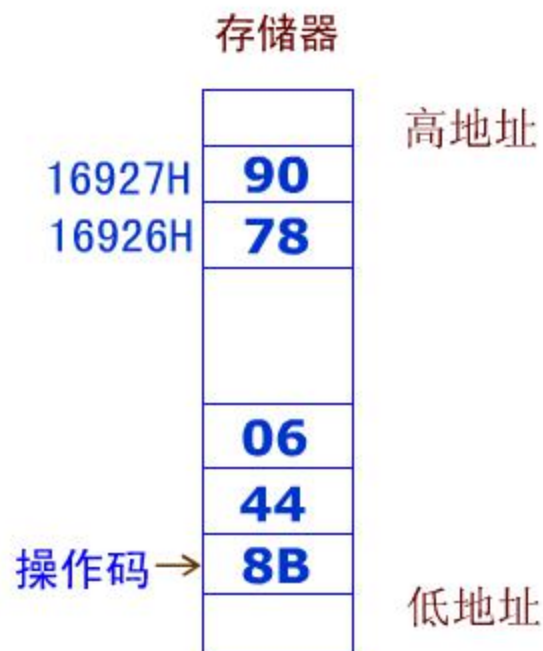
**AX**

--	--

  
**AH AL**

2000H
-------

**SI**



**MOV AX, [SI+06H]**



## 基址变址寻址方式

**AX**

--	--

  
**AH AL**

0006H	
2000H	

**SI**  
**BX**

**MOV AX, [BX+SI]**



## 相对基址变址寻址方式

**AX**

--	--

  
**AH AL**

<b>1000H</b>	
<b>1000H</b>	

**BX**  
**DI**

存储器

16927H  
16926H

<b>90</b>
<b>78</b>
<b>06</b>
<b>41</b>
<b>8B</b>

高地址

操作码→

低地址



播放



停止

**MOV AX, [BX+DI+06H]**



# 8086编程结构

