



# 第八章 指令系统(一)

---

秦磊华 计算机学院

---

8.1 指令系统概述

8.2 指令格式

8.3 寻址方式

CONTENT





## 8.1 指令系统概述

### 1. 基本概念

- ◆ 机器指令（指令）

指挥计算机执行某种操作的命令，计算机能直接识别；

- ◆ 指令系统（指令集）

一台计算机中所有机器指令的集合

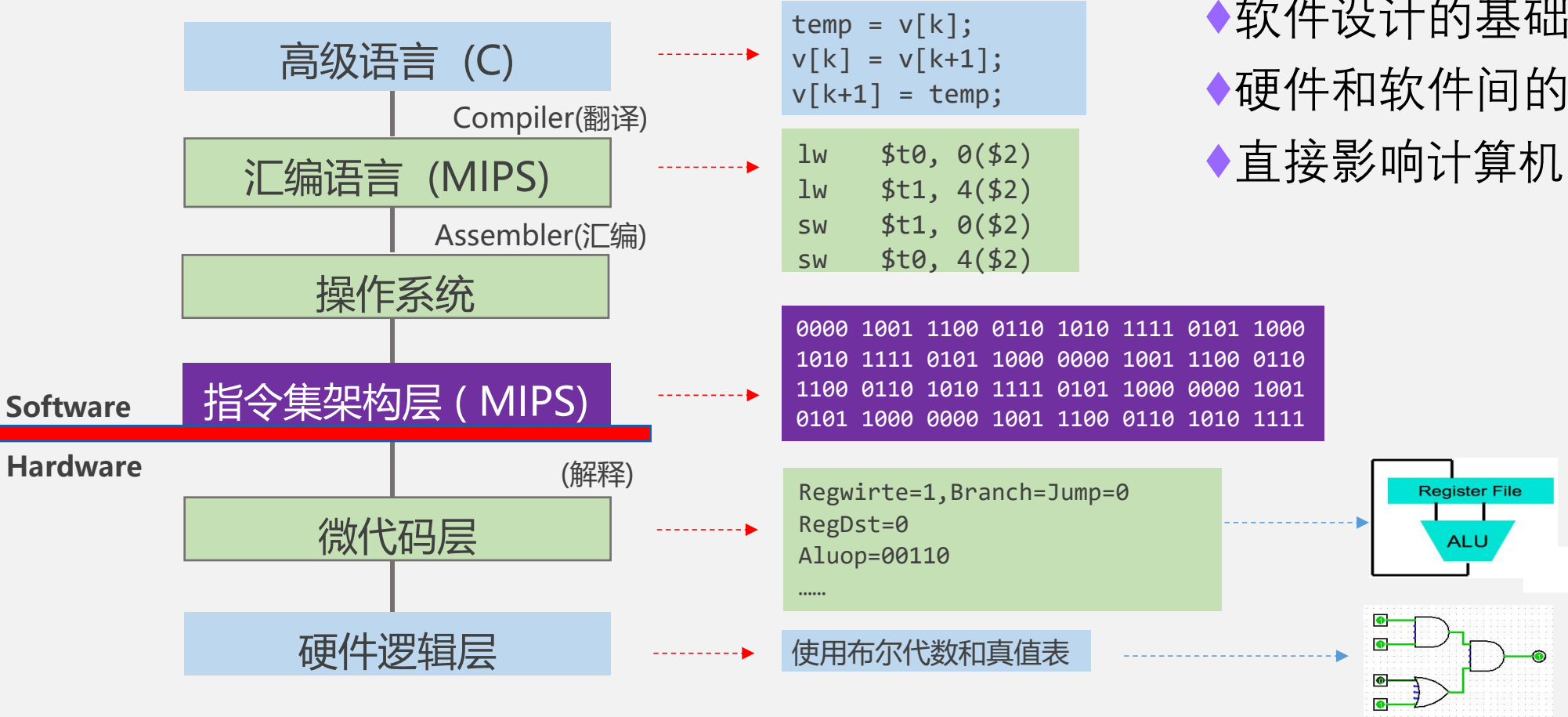
- ◆ 系列机（解决软件兼容问题）

基本指令系统相同，基本系统结构相同、同一厂家在不同时期生产的计算机，如PDP-11，VAX-11，Intel-x86



# 8.1 指令系统概述

## 2. 计算机指令系统层次



- ◆指令系统是CPU设计的依据
- ◆软件设计的基础
- ◆硬件和软件间的界面
- ◆直接影响计算机系统性能

### 3. 指令系统设计的基本原则

- ◆完备性：指令丰富，功能齐全，使用方便
- ◆有效性：程序占空间小，执行速度快
- ◆规整性：
  - 对称性 （对不同寻址方式的支持）
  - 匀齐性 （对不同数据类型的支持）
  - 一致性 （指令长度和数据长度的一致性）
- ◆兼容性：系列机软件兼容

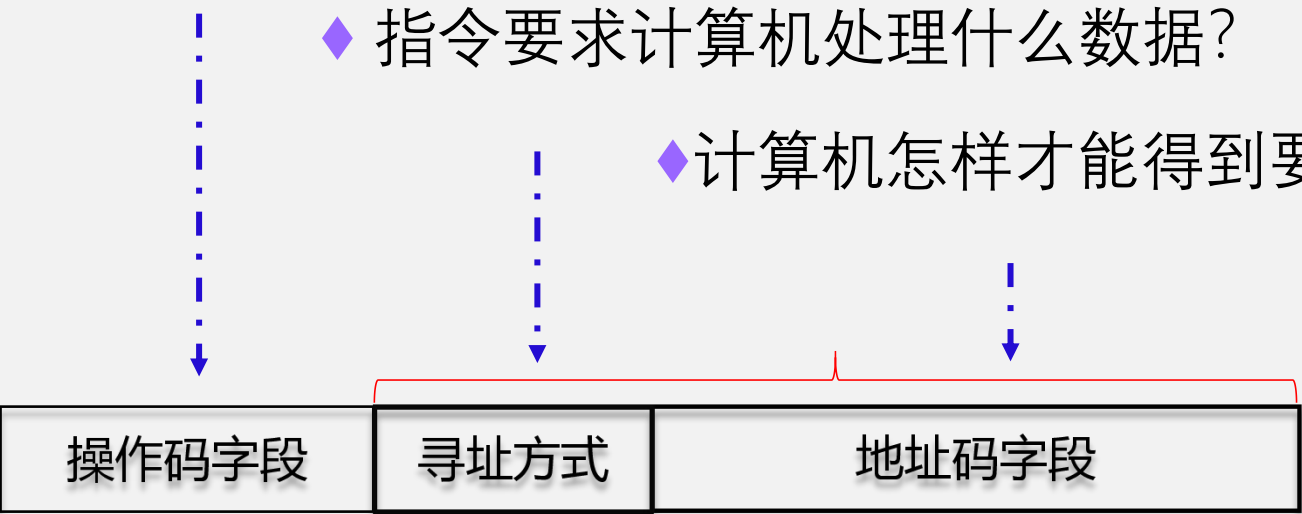
## 1. 指令格式的基本概念

用二进制代码表示的指挥计算机执行某种操作的命令形式及组织。

◆指令要求计算机对数据做什么处理？

◆ 指令要求计算机处理什么数据？

◆计算机怎样才能得到要处理的数据？



### 2. 指令格式的基本形式



◆ 操作码字段长度影响指令系统规模

定长操作码  $\text{Length}_{\text{OP}} = \lceil \log_2 n \rceil$

◆ 寻址方式字段长度影响寻址方式种类(对于有寻址方式字段而言)

◆ 操作数/地址字段长度及个数决定操作数个数、范围大小等(与寻址方式配合)

### 2. 指令格式的基本形式

三地址指令



$(A1) \text{ OP } (A2) \rightarrow A3$

二地址指令



$(A1) \text{ OP } (A2) \rightarrow A1$

单地址指令



$(AC) \text{ OP } (A1) \rightarrow AC$

零地址指令

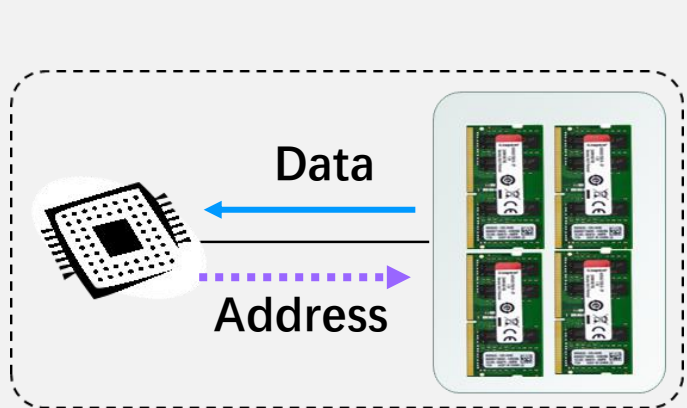


停机、空操作、开关中断等



### 1. 寻址方式的概念

根据冯诺依曼计算机的工作原理，需根据物理地址从内存中取指令和数据。如何获得指令和数据的物理地址？！



寻址方式

寻找指令和操作数有效地址的方法



## 8.3 寻址方式

### 2. 指令寻址方式

◆ 计算指令有效地址的方法(?);

◆ 指令的寻址方式只有两种

顺序寻址

跳跃寻址



### 2. 指令寻址方式

#### (1) 指令的顺序寻址

◆ 程序的指令序列在主存顺序存放。执行时从第一条指令开始(!)，逐条取出并执行，这种程序的顺序执行过程，称为顺序寻址方式。

◆ CPU中设置 程序计数器 (PC) 对指令执行顺序进行记录。开始存放程序首地址，执行一条指令，PC 加“1”，指出下条指令地址，直到程序结束。

## 8.3 寻址方式

### 2. 指令寻址方式

#### ◆ 指令的顺序寻址



需要深刻理解 “+1”



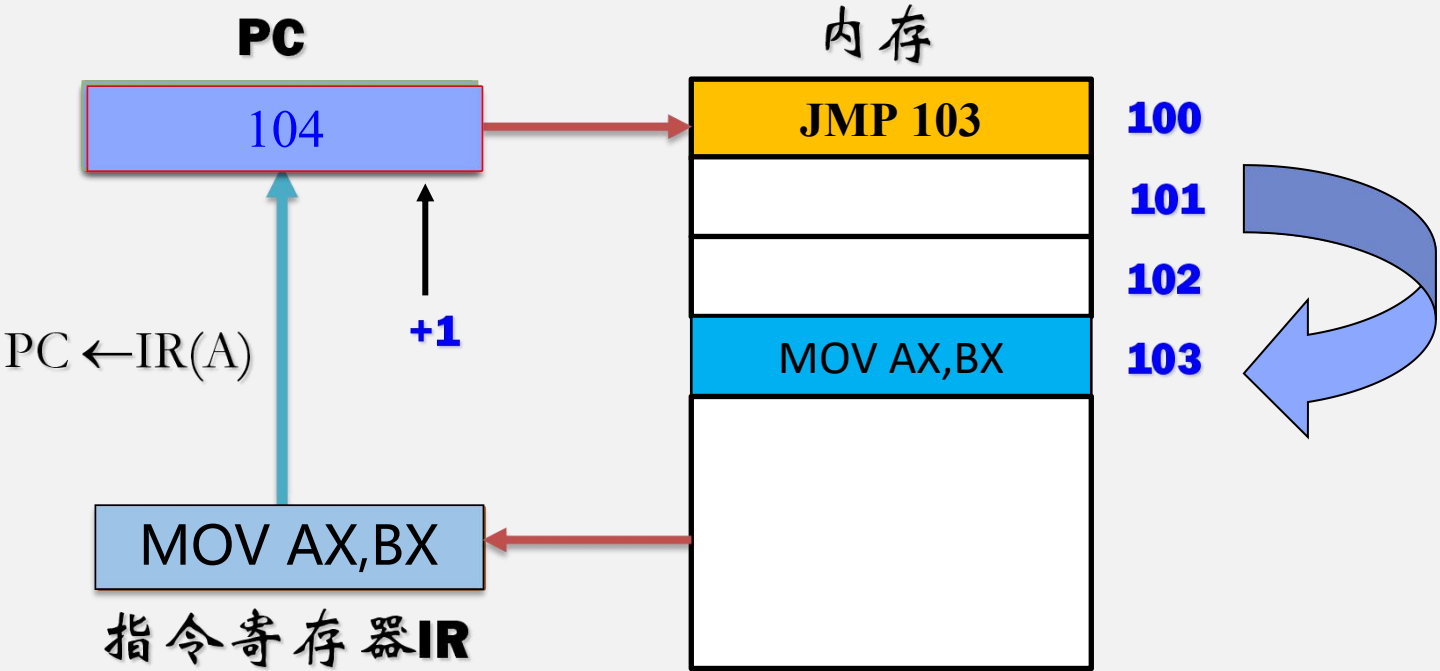
存储1条指令占用的字节单元数  
与存储字长有关!



# 8.3 寻址方式

## 2. 指令寻址方式

◆ 指令的跳跃寻址





## 8.3 寻址方式

### 3. 操作数寻址方式

- ◆ 形成操作数有效地址的方法
- ◆ 当数据在主存中时， 需要计算其有效地址E
- ◆  $S = (E)$

### 3. 操作数寻址方式

#### (1) 立即数寻址

指令地址码字段是操作数本身

MOV

000

200H

例 MOV AX,200H (AX ← 200H)

S=D

特点:

- ◆ 取指操作将数据与指令一并读入CPU内部寄存器，指令执行速度快
- ◆ 便于程序设计（变量赋初值）
- ◆ 数据大小受字段位数限制

### 3. 操作数寻址方式

#### (2) 寄存器寻址

操作数在地址码字段指定的寄存器中

MOV

000

001

例 MOV AX, BX (AX  $\leftarrow$  (BX))

特点:

- ◆ 操作数在寄存器中，指令执行速度快；
- ◆ 能访问数据大小一般与计算机字长相同；
- ◆ 地址字段的位数影响能访问的CPU内通用寄存器及其数量；



## 8.3 寻址方式

### 3. 操作数寻址方式

#### (3) 直接寻址

地址码字段直接给出操作数地址.  $E=D, S=(D)$

MOV AX, [200H]



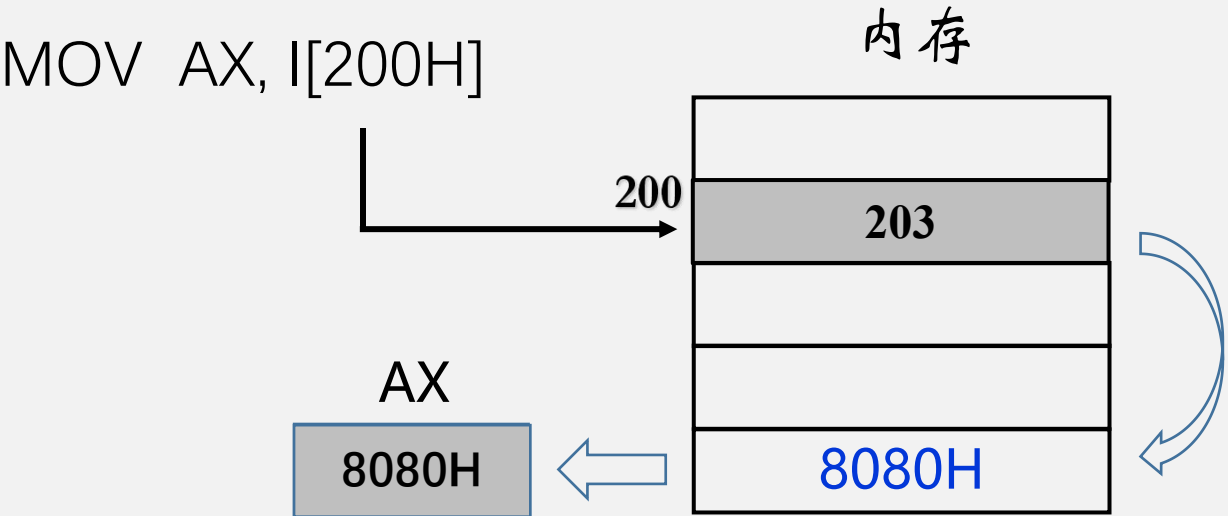
特点:

- ◆ 提供访问主存的操作;
- ◆ 获得数据要访问主存, 指令执行速度慢;
- ◆ 指令格式中地址字段位数决定了访存空间大小;

## 3. 操作数寻址方式

### (4) 间接寻址

地址码字段给出的是操作数地址的地址.  $E=(D), S=((D))$



特点:

- ◆ 解决了直接寻址方式下地址字段位数限制访存范围的问题
- ◆ 获得数据要访问主存2次，指令执行速度太慢

## 3. 操作数寻址方式

## (5) 寄存器间接寻址

操作数地址存放在指令地址码字段指定的寄存器中

$$E=(R), \quad S=((R))$$

例 MOV AX, [BX]



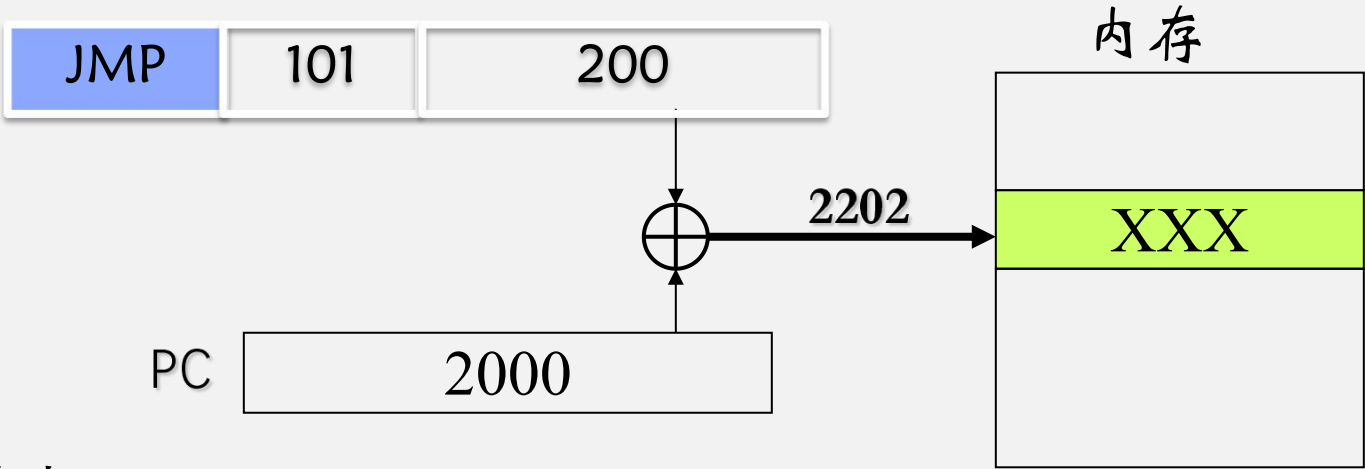
特点:

- ◆ 解决了直接寻址方式地址字段位数限制访存范围大小和间接寻址速度慢等问题
- ◆ 获得数据只需访问主存1次

## 3. 操作数寻址方式

(6)相对寻址(设机器字长为16位)

$$E = D + (PC), \text{ D为指令中地址字段的值}$$



特点:

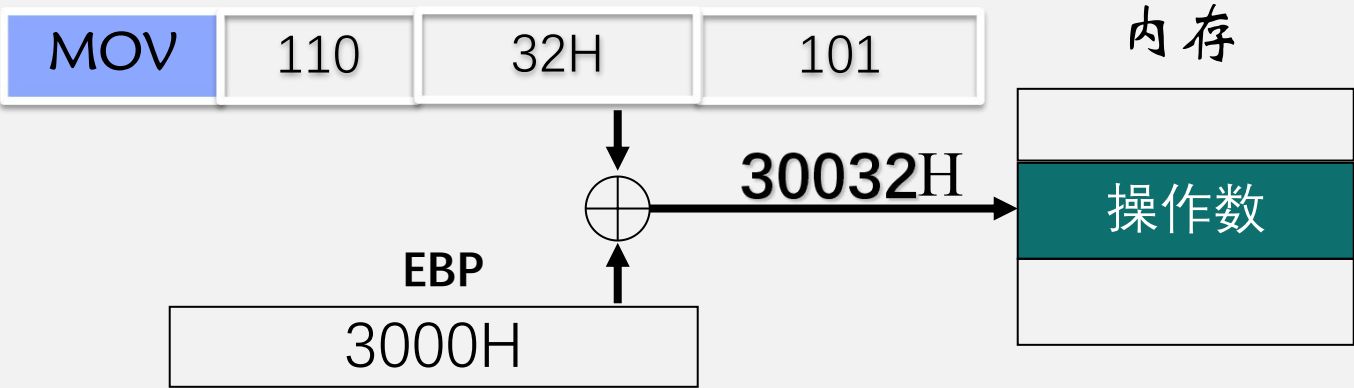
◆注意PC的改变对计算E的影响, 本例中 $E = 200 + 2000 + 2$

## 3. 操作数寻址方式

### (7)基址寻址

假定基址寄存器为EBP, MOV DX, D[EBP], D为偏移值

$$E = (EBP) \times 16 + D$$



特点:

- ◆使用基址寄存器可访问更大的主存空间
- ◆基址值设定后不变，故要访问不同数据需修改D

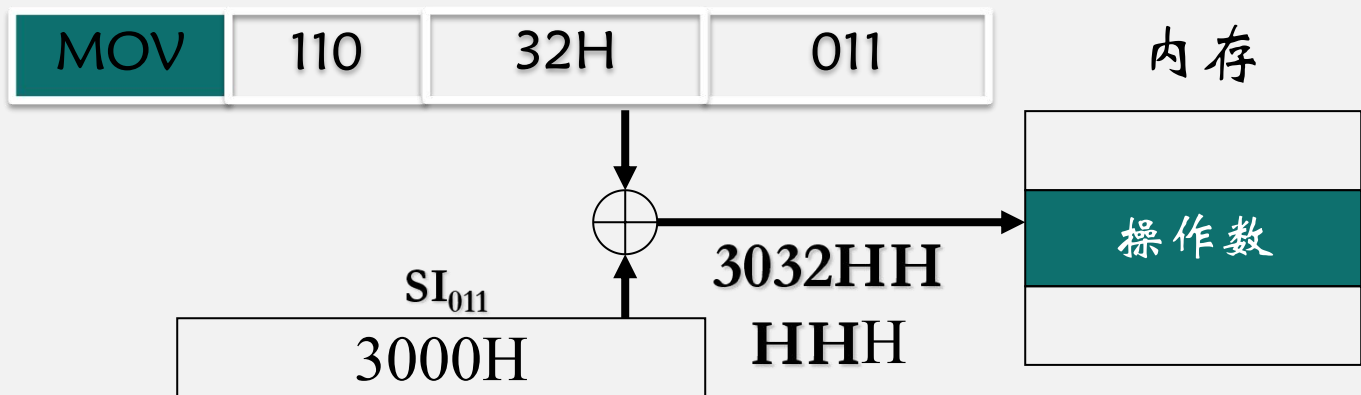
## 8.3 寻址方式

### 3. 操作数寻址方式

#### (8) 变址寻址

假定变址寄存器SI/DI,  $E = D + (SI/DI)$ , D为偏移值

MOV AX, 32[SI]



特点:

- ◆ 不改变指令即可改变数据有效地址
- ◆ 在字符串处理，向量运算等成批数据处理中非常有用



# 8.3 寻址方式

## 4. 不同寻址方式比较

	5bits	3bits	8bits	便利	速度	范围
	操作码	寻址模式	形式地址D			
立即寻址	MOV	000	38H	√	√	×
寄存器寻址	MOV	001	00	√	√	√
直接寻址	MOV	010	200	√	—	×
间接寻址	MOV	011	200	×	×	√
寄存器间接	MOV	100	01	×	—	√
相对寻址	JMP	101	20	√	—	√
变址寻址	MOV	110	32H011	√	—	√
基址寻址	MOV	110	32101	√	—	√



# 8.3 寻址方式

## 5. 寻址方式应用举例

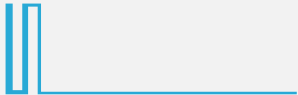
某计算机指令字长16位，格式、有关寄存器和主存内容如下，X为寻址方式，D为形式地址，请在下表中填入有效地址E及操作数的值。

OP	X	D=100	PC=1000	R <sub>变</sub> =2000
----	---	-------	---------	----------------------

100	200
200	500
500	800
1100	100
1102	350
2100	200

寻址方式	X	有效地址E	操作数2
立即	0	-----	100
直接	1	E=D=100	200
间接	2	E=(D)=200	500
相对	3	E=(PC)+D=1100	100
变址	4	E=(R)+D=2100	200
变址间址	5	E=((R)+D)=200	500





# 第一部分完