



汇编语言与微机原理

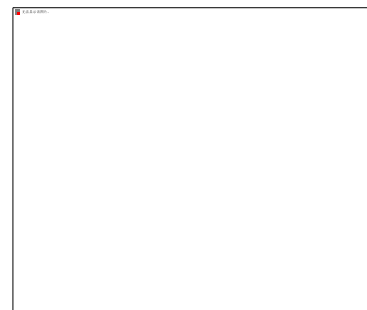
熊迎军

xyj@njau.edu.cn

13776655525

第3章

寻址方式



指令组成

[标号]: 指令助记符 [目的操作数],[源操作数];[注释]

- 标号表示该指令在主存中的逻辑地址
- 每个指令助记符就代表一种指令
- 目的和源操作数表示参与操作的对象
- 注释是对该指令或程序段功能的说明

寻找操作数的过程就是操作数的寻址



- **每种指令的操作码：**
 - 用一个唯一的助记符表示（指令功能的英文缩写）
 - 对应着机器指令的一个二进制编码
- **指令中的操作数：**
 - 可以是一个具体的数值
 - 可以是存放数据的寄存器
 - 或指明数据在主存位置的存储器地址



➤ 操作数可能的来源或存放处：

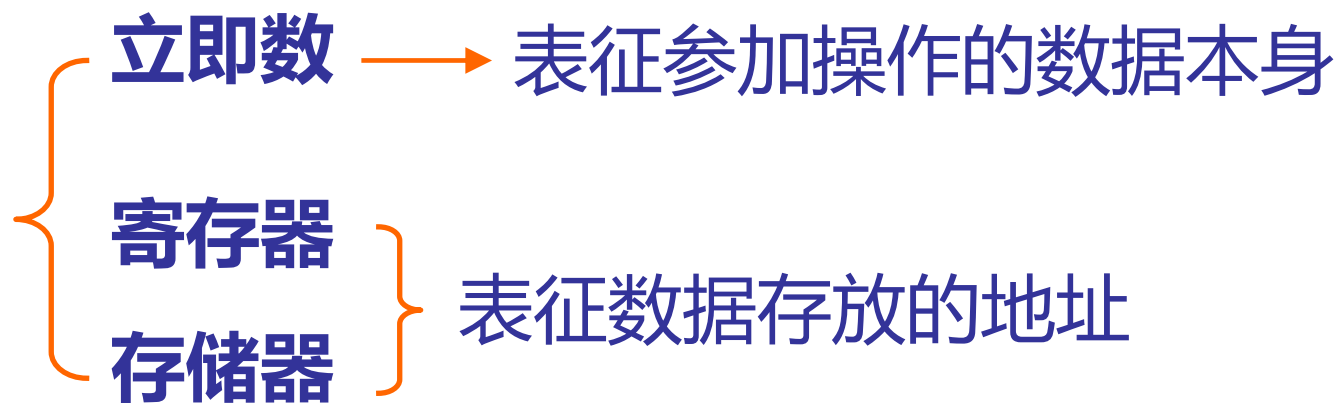
- 由指令直接给出
- 寄存器
- 内存单元

➤ 寻找操作数所在地址的方法可以有三种大类型

- 指令直接给出的方式
- 存放于寄存器中的寻址方式
- 存放于存储器中的寻址方式



指令中的操作数



立即数操作数

- 立即数本身是参加操作的数据，可以是8位或16位，只能作为源操作数。

例：MOV AX, 1234H

MOV BL, 22H

- 立即数无法作为目标操作数
- 立即数可以是10进制或者2进制的数,但指令运行后CPU自动把该数转换为16进制数,然后完成给定功能的操作，其中，二进制在数字后面加B区分。

MOV AL,57 ; 指令执行后将AL 39H

MOV AL,01010111B ; 指令执行后将AL 57H

- 立即数可以是无符号或带符号数，其数值应在可取值范围内。



寄存器操作数

- 参加运算的数存放在指令给出的寄存器中，可以是16位或8位。
- 例：
 - **MOV AX, BX**
 - **MOV DL, CH**

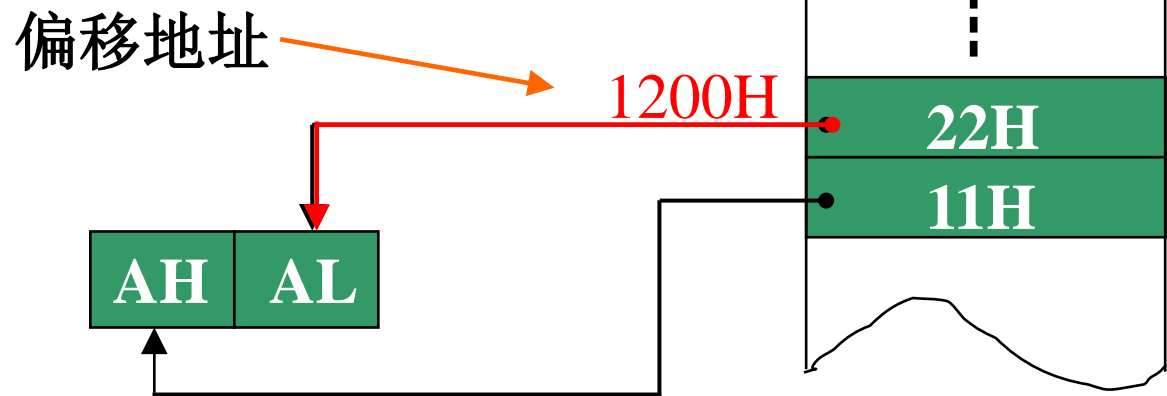
存储器操作数

- 参加运算的数存放在存储器的某一个或某两个单元中。
- 表现形式：[]

立即数或寄存器

[]中的内容是存放所寻找数据的单元的偏移地址

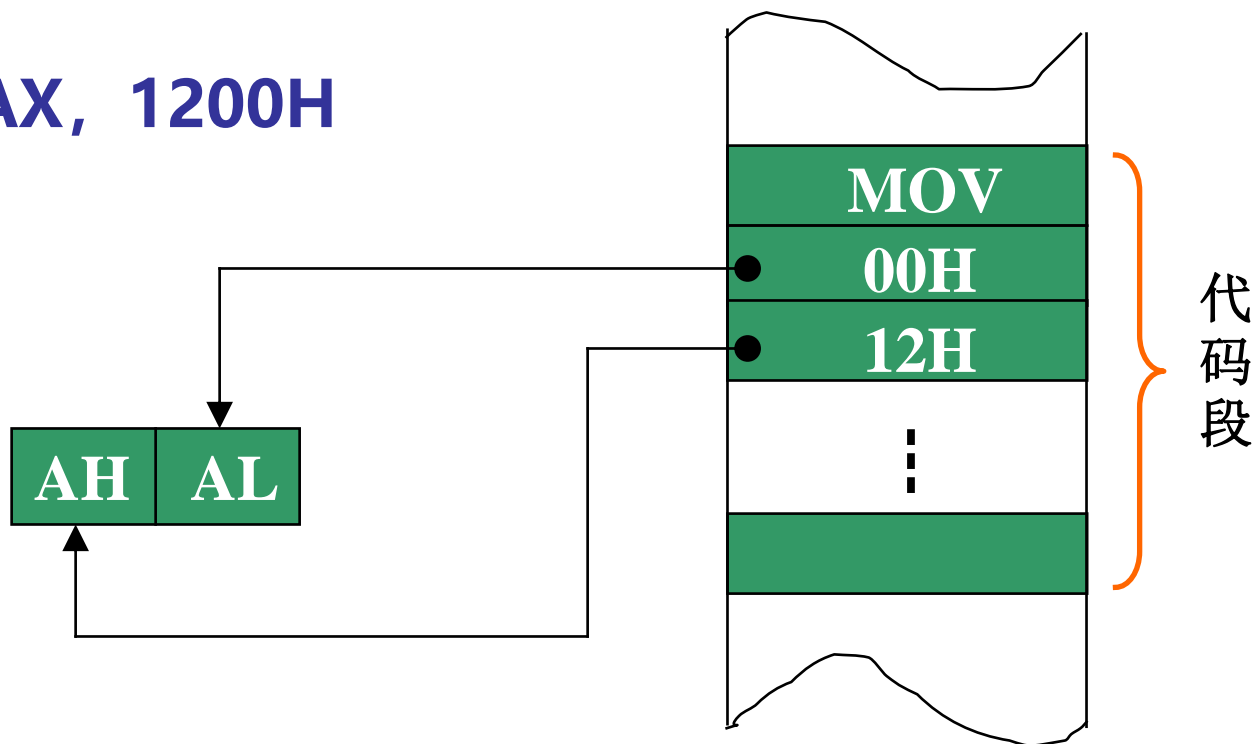
- 例: MOV AX, [1200H]
MOV AL, [1200H]



2.1 立即寻址

立即寻址仅适合于源操作数

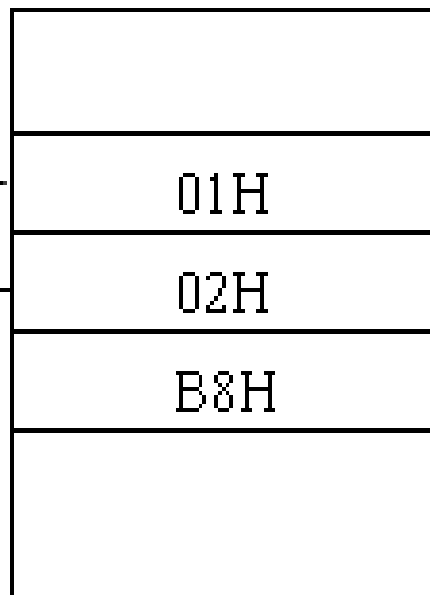
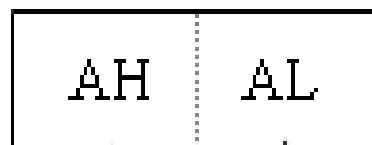
- 指令中的源操作数是立即数，即源操作数是参加操作的数据本身
- 例：MOV AX, 1200H



MOV AX,0102H ;
AX←0102H

存储器

高地址



← 立即数高字节
← 立即数低字节
← 操作码

MOV AX, 0102H

低地址

2.2 寄存器寻址

➤ 参加操作的操作数在CPU的通用寄存器中。

- AH、AL、BH、BL、CH、CL、DH、DL
- AX、BX、CX、DX、SI、DI、BP、SP
- CS、DS、SS、ES

例：MOV AX, BX





2.3 存储器寻址方式

- 指令中给出操作数的主存地址信息（偏移地址，称之为有效地址EA），而段地址在默认的或用段超越前缀指定的段寄存器中
- 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

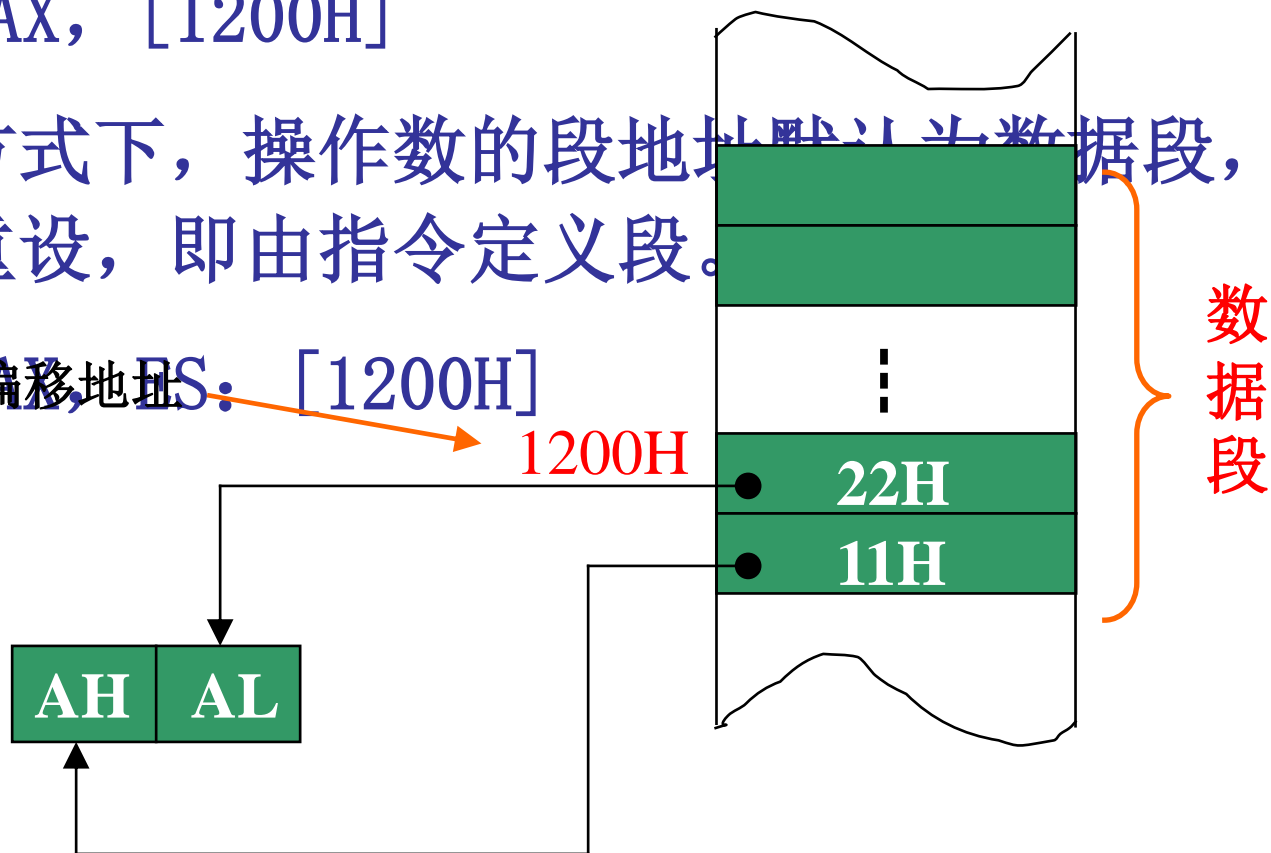
2.3.1 直接寻址

- 指令中直接给出操作数的偏移地址

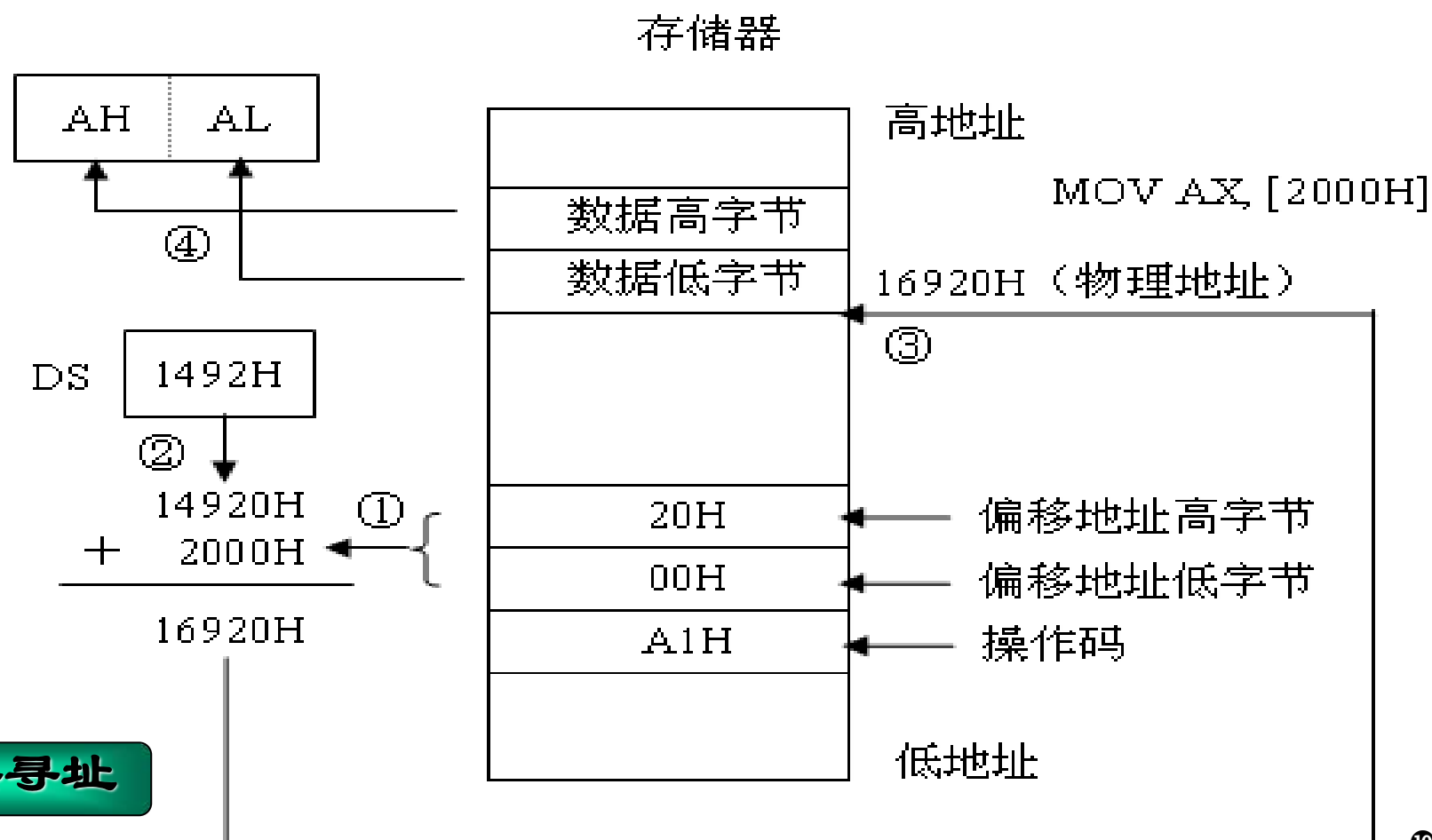
例：MOV AX, [1200H]

- 直接寻址方式下，操作数的段地址默认为数据段，但允许段重设，即由指令定义段。

例：MOV 偏移地址 AX, ES: [1200H]



MOV AX,[2000H] ; AX←DS:[2000H]

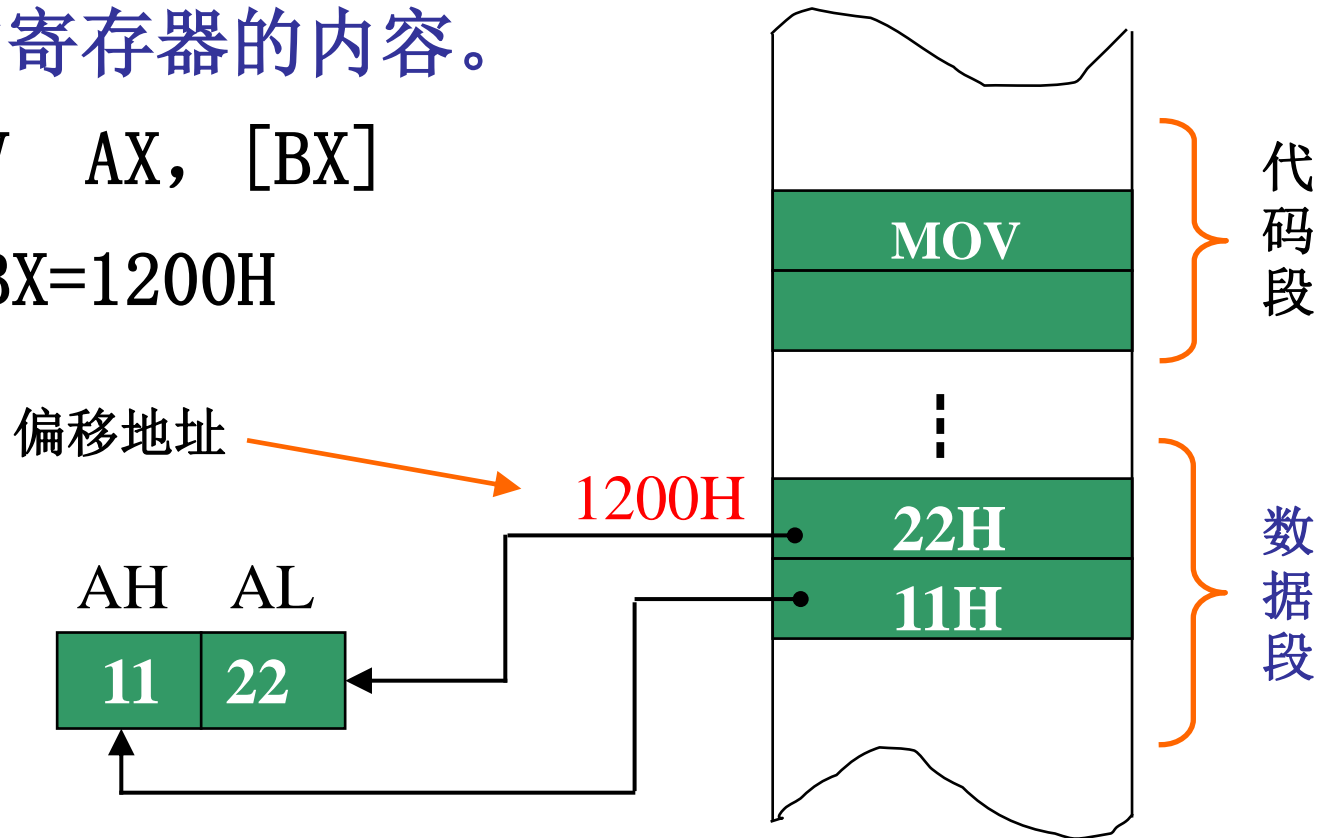


2.3.2 寄存器间接寻址

- 参与操作的操作数存放在内存中，其偏移地址为指令中的寄存器的内容。

例：MOV AX, [BX]

设BX=1200H





- 由寄存器间接给出操作数的偏移地址;
- 存放偏移地址的寄存器称为**间址寄存器**, 它们是: **BX, BP, SI, DI**
- 操作数的段地址 (数据处于哪个段) 取决于选择哪一个间址寄存器:

BX, SI, DI	→	默认在数据段
BP	→	默认在堆栈段



寄存器 间接寻址

基址寻址:

称该间址寄存器为**基址寄存器**BX, BP

变址寻址:

称该间址寄存器为**变址寄存器**SI, DI

2.3.3 寄存器间接相对寻址

- 操作数的偏移地址为寄存器的内容加上一个**位移量**

例：

- **MOV AX, [BX+DATA]**

设：DS=2000H, BX=0220H, DATA=05H

- 则：AX=[20225H]

2.3.4 基址-变址寻址

➤ 操作数的偏移地址为

- 一个基址寄存器的内容 + 一个变址寄存器的内容;

➤ 操作数的段地址由选择的基址寄存器决定

- 基址寄存器为BX，默认在数据段
- 基址寄存器为BP，默认在堆栈段

➤ 基址变址寻址方式与相对寻址方式一样，主要用于一维数组操作。

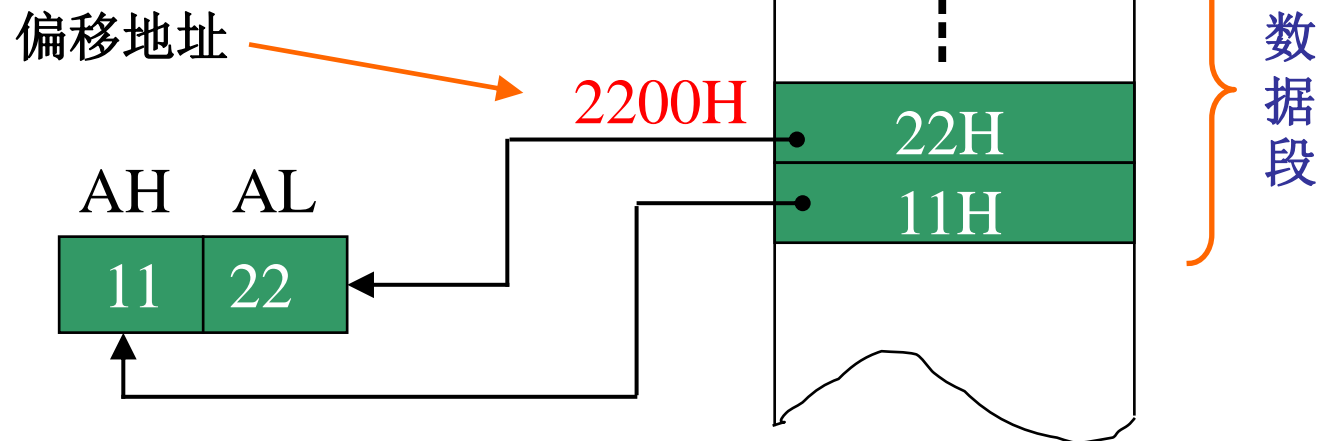
不允许将两个基址寄存器或两个变址寄存器组合在一起寻址，即指令中不允许同时出现两个基址寄存器或两个变址寄存器

例：执行下列指令：

MOV SI, 1100H

MOV BX, SI

MOV AX, [SI+BX]





例：判断以下指令是否正确：

MOV AX, [BX][BP]; 错误，不能同时出现两个基址寄存器

MOV AX, [SI][DI]; 错误，不能同时出现两个变址寄存器

2.3.5 基址-变址-相对寻址

- 操作数的偏移地址为：

基址寄存器内容 + 变址寄存器内容 + 位移量

或：BX/BP（基址寄存器） + SI/DI（变址寄存器） + 位移量

- 操作数的段地址由选择的基址寄存器（BX对应DS，BP对应SS）决定。

基址变址相对寻址方式主要用于二维表格操作。
不允许将两个基址寄存器或两个变址寄存器组合在一起寻址，即指令中不允许同时出现两个基址寄存器或两个变址寄存器

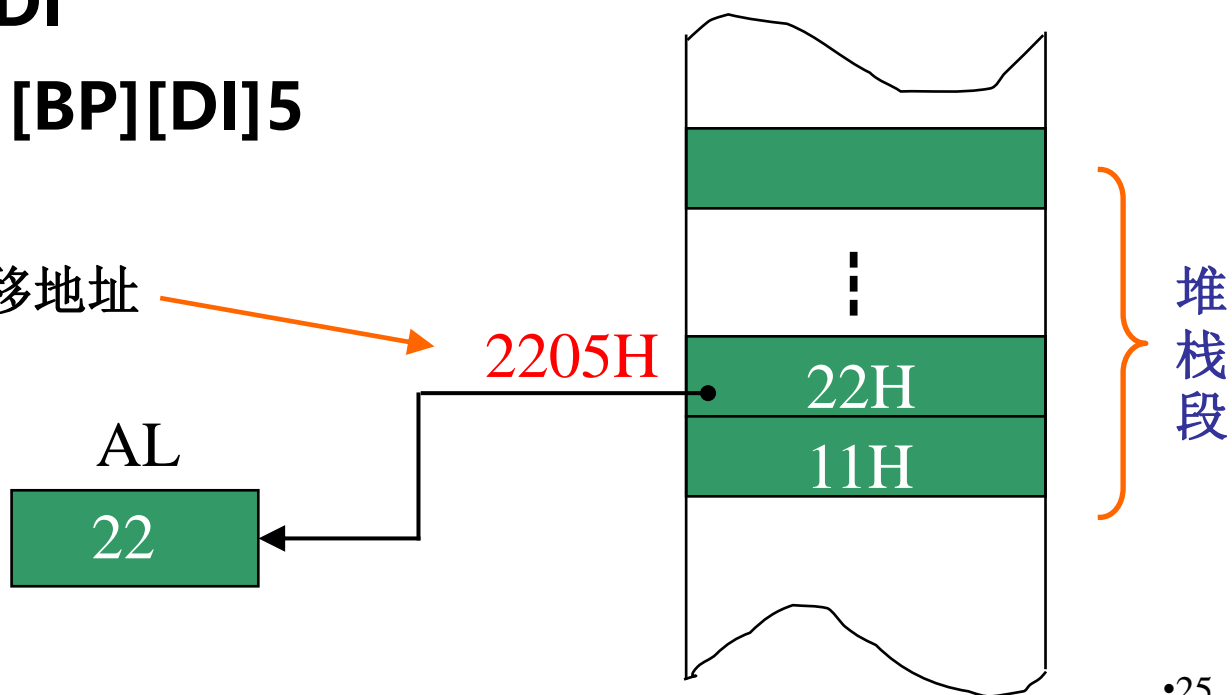
例：执行以下程序段：

MOV DI, 1100H

MOV BP, DI

MOV AL, [BP][DI]5

偏移地址





例： 判断以下指令是否正确：

MOV AX, DATA[SI][DI]; 非法

MOV AX, [BX][BP]DATA; 非法

2.4 隐含寻址

- 指令中隐含了一个或两个操作数的地址，即操作数在默认的地址中。

例：

- **MUL BL**

- 指令执行：

- **AL × BL → AX**



寄存器操作数的表达

- **r8——任意一个8位通用寄存器**

AH AL BH BL CH CL DH DL

- **r16——任意一个16位通用寄存器**

AX BX CX DX SI DI BP SP

- **reg——代表r8或r16**

- **seg——段寄存器**

CS DS ES SS



存储器操作数的表达

- m8——一个8位存储器操作数单元（所有主存寻址方式）
- m16——一个16位存储器操作数单元（所有主存寻址方式）
- mem——代表m8或m16



立即数的表达

- i8——一个8位立即数
- i16——一个16位立即数
- imm——代表i8或i16
- dest——目的操作数
- src——源操作数



本章结束，谢谢大家！