《微型计算机原理与接口技术》
第6版

第2章　8086 CPU

§2.3 8086的存储器组织

CPU的工作方式

- 8086/8088 只能工作于**实模式**，仅能访问 2^{20} = 1MB存储器

- 80286及以上CPU可工作于实模式和**保护模式**。在保护模式下，寻址范围为

  80286 ：寻址2^{24}=16MB内存

  80386：寻址2^{32}= 4GB内存

2.3.1 段地址和偏移地址

2.3.2 8086存储器的分体结构

2.3.1 段地址和偏移地址

1.段地址和偏移地址组合成物理地址

- 8086/8088有20根地址线，寻址2^{20}=1MB单元，地址范围00000~FFFFFH。
- 每个单元有1个绝对地址，即物理地址，CPU应先确定物理地址，才能存取该单元。
- 1MB内存空间分成多个逻辑段，每段最大2^{16}=64KB，段内地址连续。
- 各段相互独立，可连续排列，也可部分重叠或完全重叠。

- 在编制程序时，要把存储器划分成段，每个段最大可达64KB，这样段内地址可以用16位来表示。
- PC机对段的起始地址有限制，段不能起始于任意地址，而必须从任一小段的首地址开始。
- 机器规定：从0地址开始，每16个字节为一小段
- 00000，00001，00002，------，0000E，0000F；
- 00010，00011，00012，------，0001E，0001F；
- 00020，00021，00022，------，0002E，0002F
- 在十六进制表示地址中，最低位为0，（20位地址的最低4位为0）

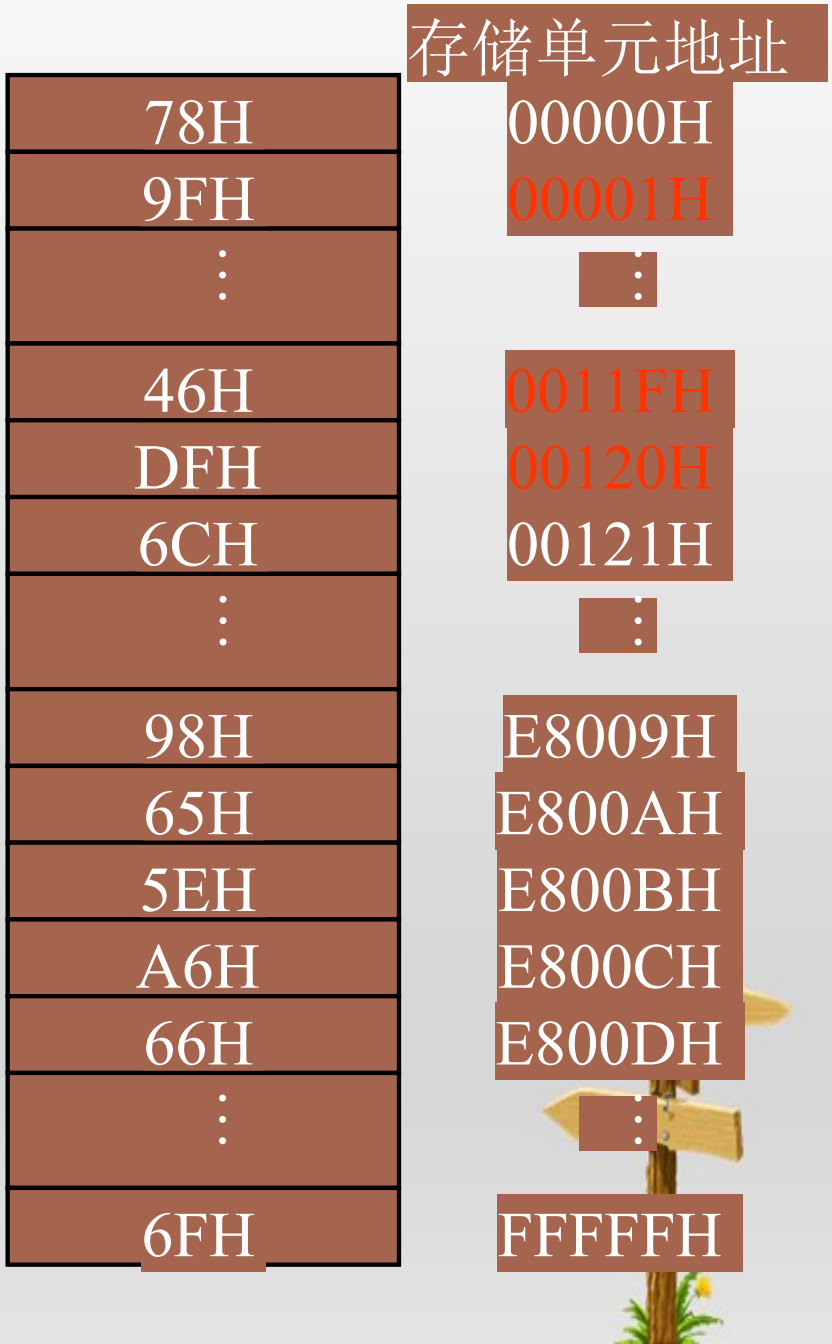
一个存储单元中存放的信息称为该存储单元的内容。

如00001H单元的内容为9FH，记为：(00001H)=9FH。

```
mov al, [0001H]
```

如从地址0011FH开始的两个连续单元中存放一个字型数据，则该数据为DF46H，记为：(0011FH)=DF46H。

```
mov ax, [011FH]
```



段地址和偏移地址

- 段地址：表示一段的开始
- 偏移地址，在段内相对于段起始地址的偏移值。如当偏移量为‘0’时，就在这个段的起始单元，当偏移量为0FFFFH时，就是这个段（最大）最末一个字节单元。
- CPU访问主存必须传送出物理地址，而用户编程则使用逻辑地址，每个存储单元都有两种形式的址：**物理地址**和**逻辑地址**
- 每个存储单元的**物理地址是唯一的**，就是这个单元的地址编码
- 程序中，程序员使用的是逻辑地址，而不是物理地址这样有利于程序的开发，且对存储器的动态管理也是有利的，一个逻辑地址是由段地址和偏移地址两部分组成，而且都是无符号16位二进制数。

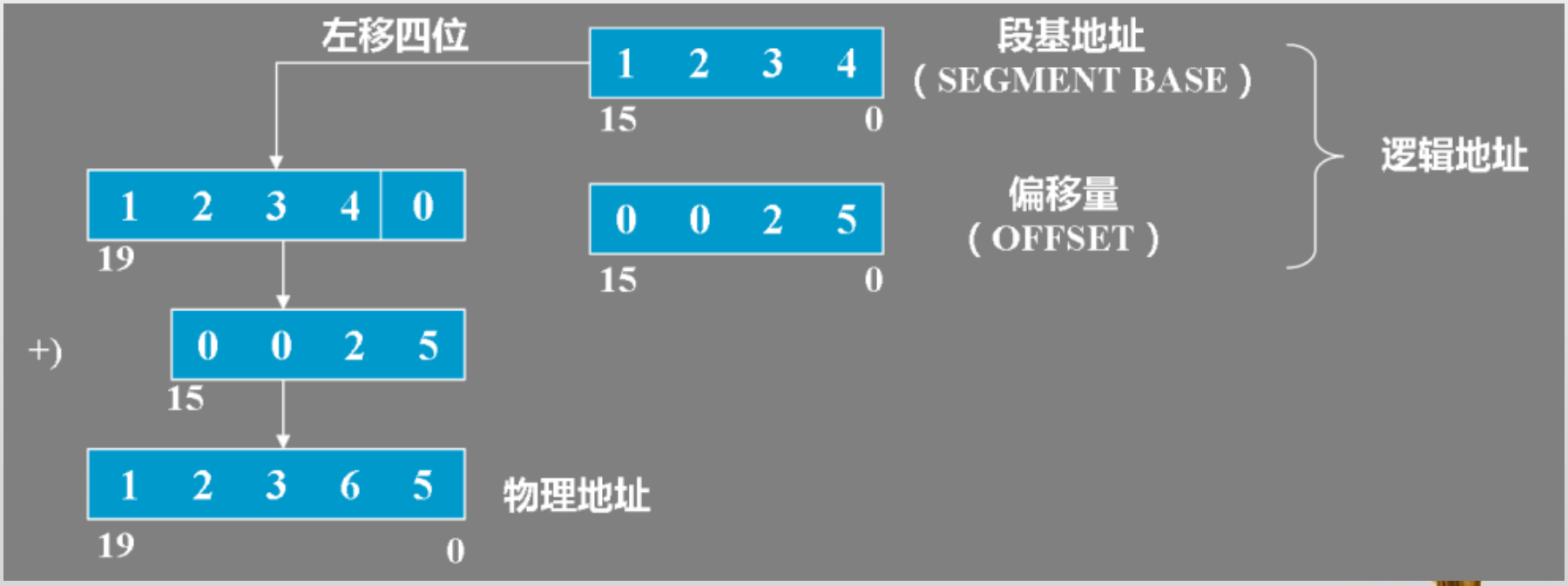
- 用两个16位寄存器来形成20位地址，形式为
  段地址：偏移量
  这也称为逻辑地址，段地址也称为段基地址。
- 段基地址定义任何64KB存储器的起始地址，偏移量在64KB存储器中选择任一单元。
- 由逻辑地址转换为物理地址的公式：

  20位物理地址=段基地址×16+16位偏移量

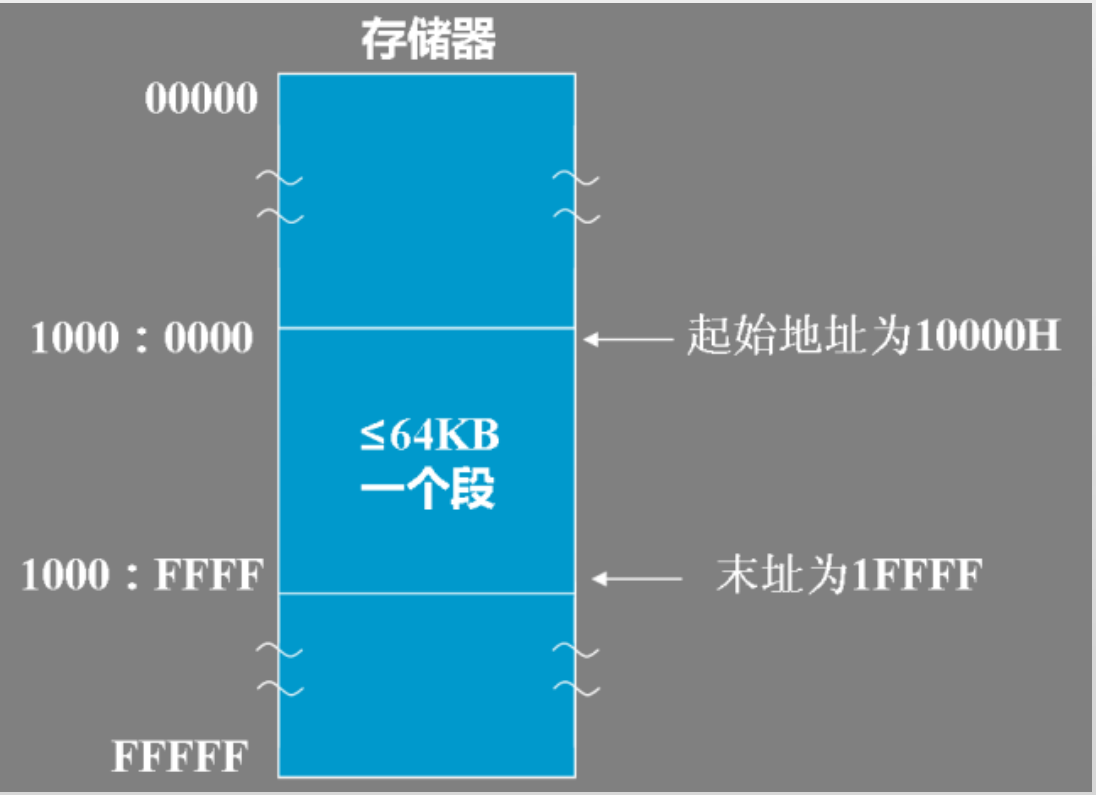
即：段寄存器中的16位数自动左移4位✚16位偏移量就形成20位物理地址。

- 由BIU的地址加法器Σ来口算物理地址。

- 设段地址：偏移地址=1234：0025H，形成20位物理地址12365H的过程：



- 如何用段基地址和偏移地址形成一个段，由偏移地址来选择段中的一个存储单元。



- 段基址=1000H，该段始址=1000H×16=10000H
- 段内偏址范围为0000~FFFFH，即段开始到所选单元的距离
- 段长=64KB，该段末址=1000H×16+FFFFH=1FFFFH

即段始址+FFFFH=该段的结束地址。

例2.4

- 设某个段寄存器的内容为3000H，则该段的起始地址和末地址各是什么？如果偏移地址OFFSET = 500H，则该单元的物理地址是多少？
- 根据物理地址的形成方法可知：
- 段起始地址为3000H×16=30000H
- 段结束地址为3000H×16+FFFFH=3FFFFH
- 偏移地址OFFSET=500H时，该单元的物理地址=3000H×16+500H=30500H

- 实模式下，只能从能被16整除的那些单元开始分段。
- 一个物理地址可以由不同的逻辑地址来形成。

例2.5

- 一个存储单元的物理地址为12345H，它可以由哪些逻辑地址形成?
- 解答： 1200：0345H

  1234：0005H

  1232：0025H

  …

这说明从12000H单元偏移345H单元和从12340H偏移5个单元等，均指向同一个内存单元。

2. 默认段寄存器和偏移地址寄存器

- CS和IP组合寻址下一条要执行指令的字节单元；
- SS和SP、BP组合寻址存储器堆栈段中的数据；
- DS和BX、SI、DI组合寻址数据段中的8位或16位数据；
- ES和DI组合寻址目的串地址。

◈ 通过段超越前缀可以对某些隐含规则进行修改。

段地址与段寄存器

- 实模式下，代码段的地址放在CS中，数据段的段地址放在DS中，堆栈段的段地址存放在SS中，附加段的段地址存放在ES中。
- 除非专门指定，如用户自行设定之外，在一般情况下，各段在存储器中的分配是由操作系统负责的。每个段可以独立地占用64KB存储区。各段也允许部分重叠或者完全重叠

3. 堆栈的设置和操作

- 什么是堆栈？
  - 堆栈是在内存中开辟的一个特定数据区域
  - 堆栈存放需暂时保存的数据，如调用子程序时的返回地址、中断处理时的断点及现场信息等。

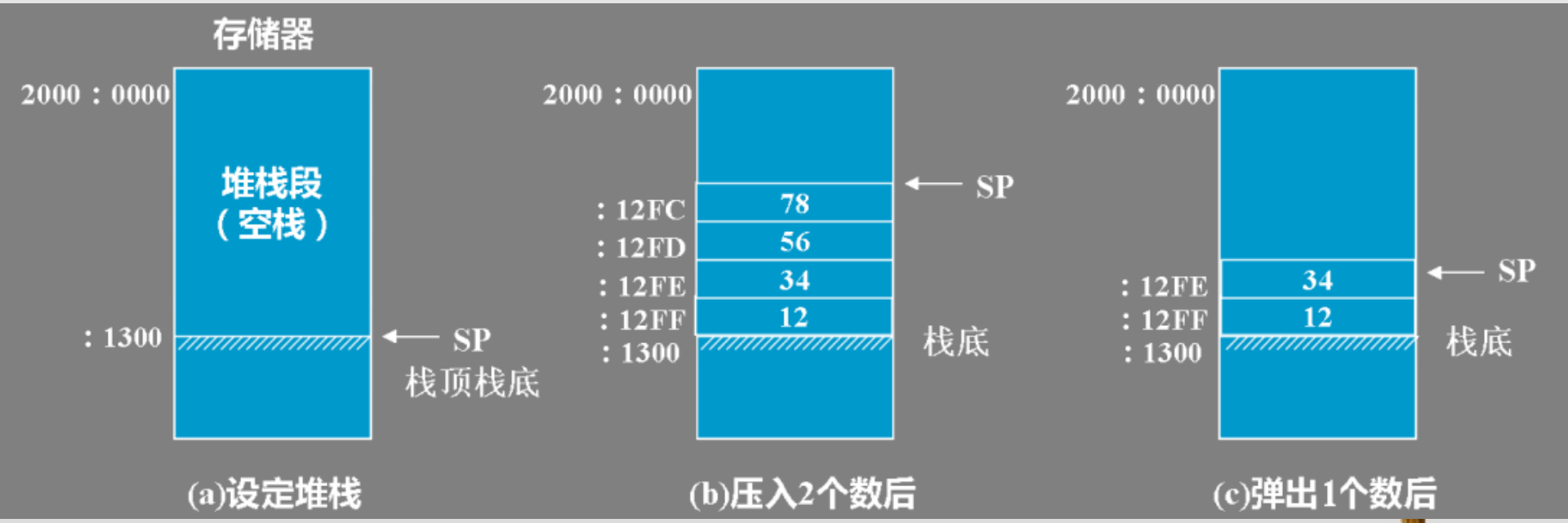
- 如何设置堆栈？
  - 堆栈位置和长度由SS：SP来设定
  - 可设置的堆栈最大容量为64KB

堆栈设置和操作举例

a）设置堆栈

令SS=2000H，SP=1300H。堆栈范围为2000:0000H ~ 2000:（1300H-1），即20000H ~ 212FFH。

SS=堆栈的段基址2000H，SP=栈顶地址，见图(a)



- 用PUSH/POP指令进行堆栈操作，遵循先进后出的原则。

b）PUSH操作将1个字压入栈，并使SP←SP-2

设AX=1234H，BX=5678H

执行PUSH AX，PUSH BX指令

4字节先后压入堆栈，并使SP=12FCH，如图(b)。

c） 执行POP DX指令

从栈里取出2字节，送入DX，DX=5678H，SP=12FEH，如图(c)。

- 通过BP指针也可从堆栈中获取数据，或向堆栈存入数据。

4.段加偏移量寻址机制允许重定位

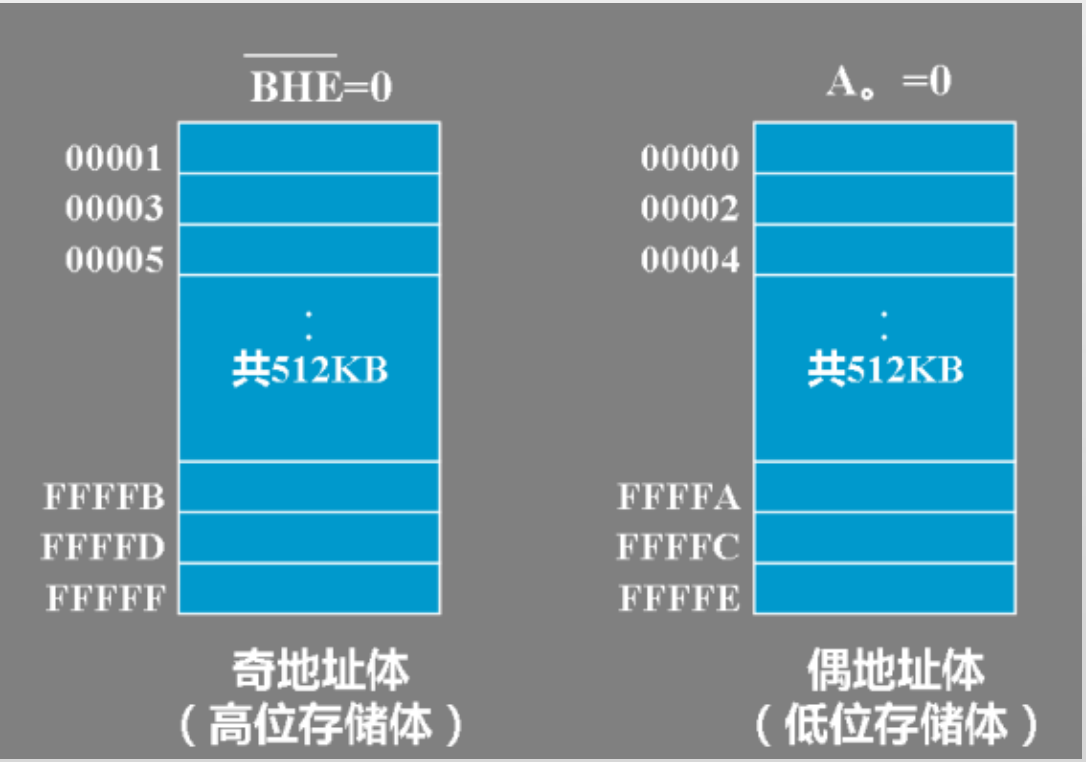
- **可重定位程序**，是指一个可以存放在存储器的任何区域，不加修改就可以执行的程序。
- **可重定位数据**，是指可以存放在存储器的任何区域，不用修改就可以被程序引用的数据。
- 由于存储器**采用偏移地址在段内寻址**，因此一个程序段或数据块，在内存中搬移时，可以**保持其偏移地址不变，只改变段寄存器的内容**，因此搬到哪里都只要修改段寄存器内容后就可以执行，即它们具有了重定位的特点。

2.3.1 段地址和偏移地址

2.3.2 8086存储器的分体结构

2.3.2 8086存储器的分体结构

1. 8086的奇偶存储体



- 1MB存储空间分成两个8位的存储体：偶地址体和奇地址体，各占512K字节。
- 偶地址体包含所有地址偶数的存储单元，奇地址体包含所有地址奇数的存储单元。

◆□构如图

8086的1M存储空间实际上分为两个512KB的存储体，又称存储库，分别叫高位库和低位库。

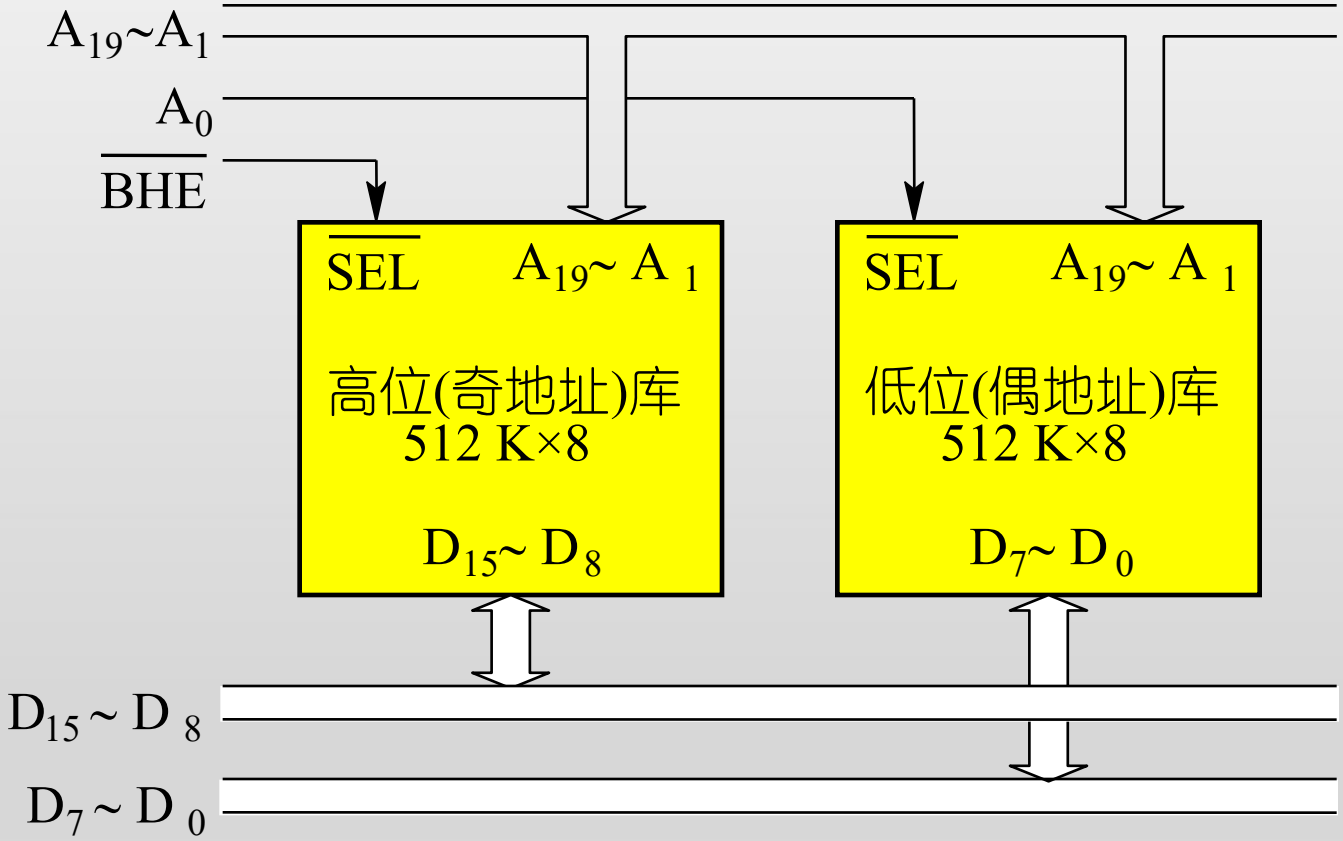


图3.13　8086存储器高低位库的连接

8086的1M存储空间实际上分为两个512KB的存储体，又称存储库，分别叫高位库和低位库。

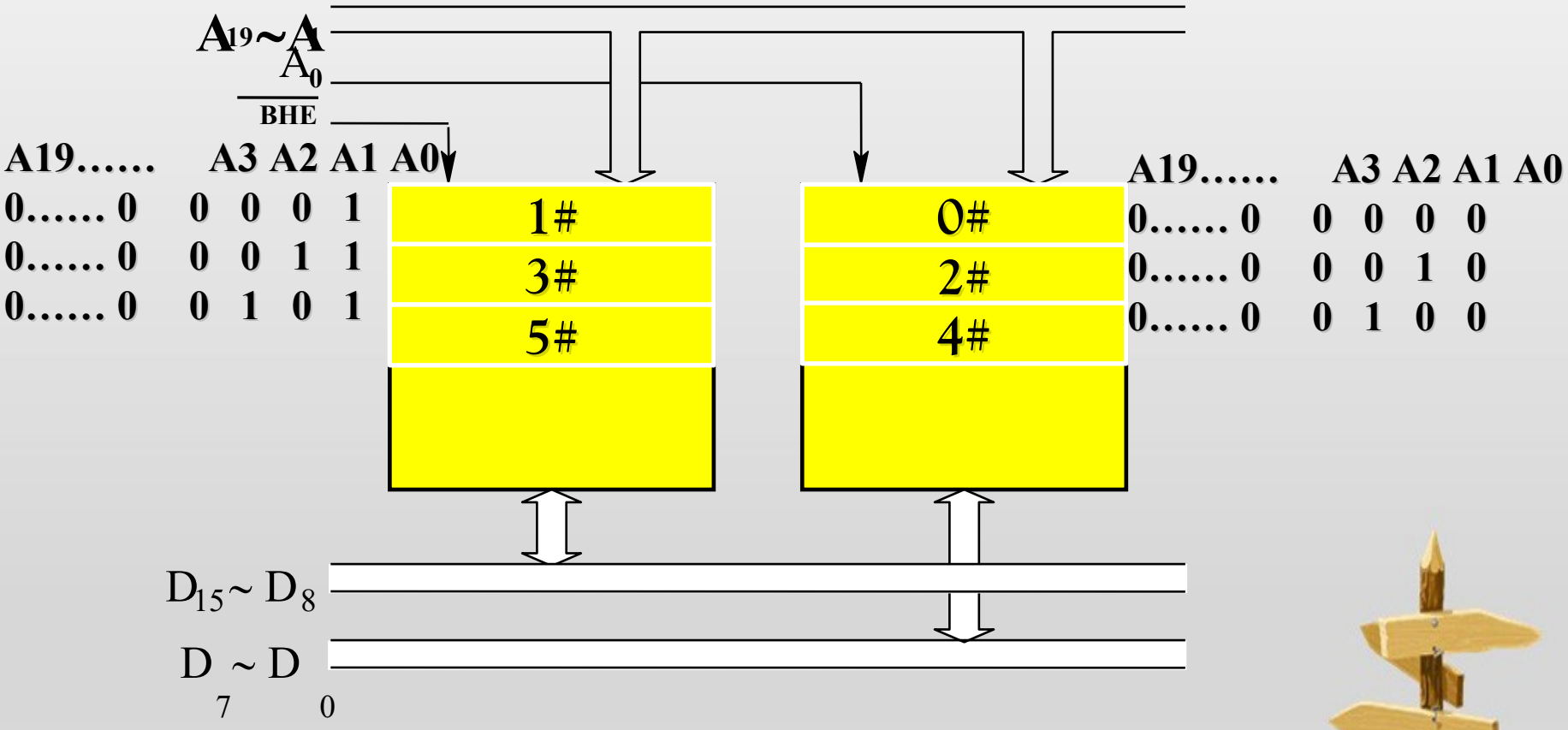


图3.13　8086存储器高低位库的连接

8086的1M存储空间实际上分为两个512KB的存储体，又称存储库，分别叫高位库和低位库。

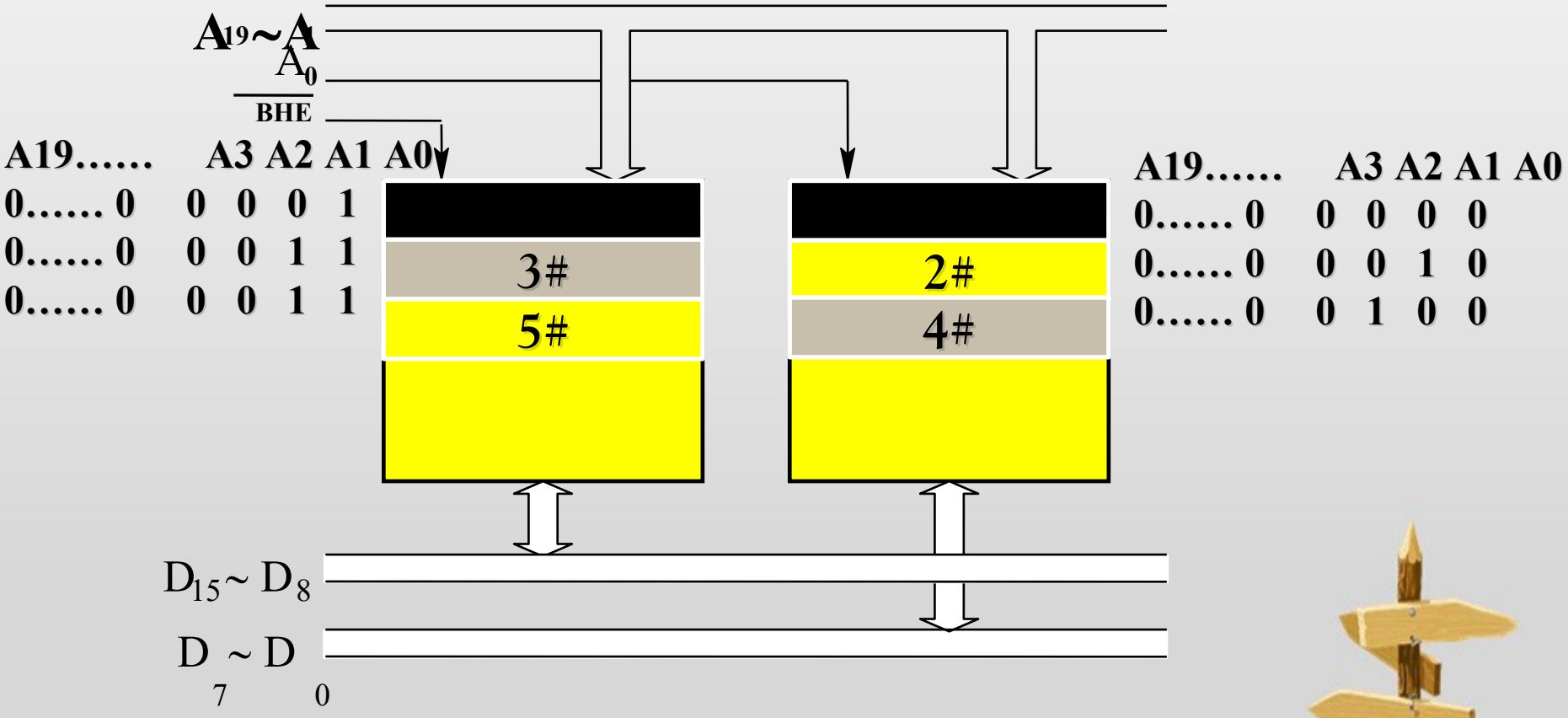


图3.13　8086存储器高低位库的连接

$\overline{BHE}$

- 用8086 CPU的 引脚信号和地址线A_0来选择一个或两个存储体进行数据传送，组合功能如表2.4。

表 2.4　使用$\overline{BHE}$和 A_0 选择存储体

$\overline{BHE}$	A_0	功　　能	使用数据总线
0	0	选中两个存储体，传送一个字	D_{15}～D_0
0	1	选中奇地址体，传送高字节	D_{15}～D_8
1	0	选中偶地址体，传送低字节	D_7～D_0
1	1	无效	

- A_0=0访问偶地址体，偶体数据线与数据总线低8位D_7~D_0连，传送低8位数据；
- $\overline{BHE}$ =0访奇地址体，奇体数据线与数据总线高8位D_{15}~D_8连，传送高8位数据；
- $\overline{BHE}$和A_0都为0，同时选中两个存储体，可传送16位数据。

2. 8086 CPU对存储器的存取操作

- 存取都从偶体开始。从偶地址单元开始存取一个字只要1次操作，从奇地址开始要2次操作。

- 从偶地址1000H开始，一次操作就可读取字数据5D7FH。
- 若要读取奇地址1001H开始的1个字345DH，要先从1000H开始读1个字5D7FH，取5DH为结果低字节，舍弃7FH，再从1002H单元读取1234H，取34H作结果高字节，舍弃12H。

1000	7F
1001	5D
1002	34
1003	12

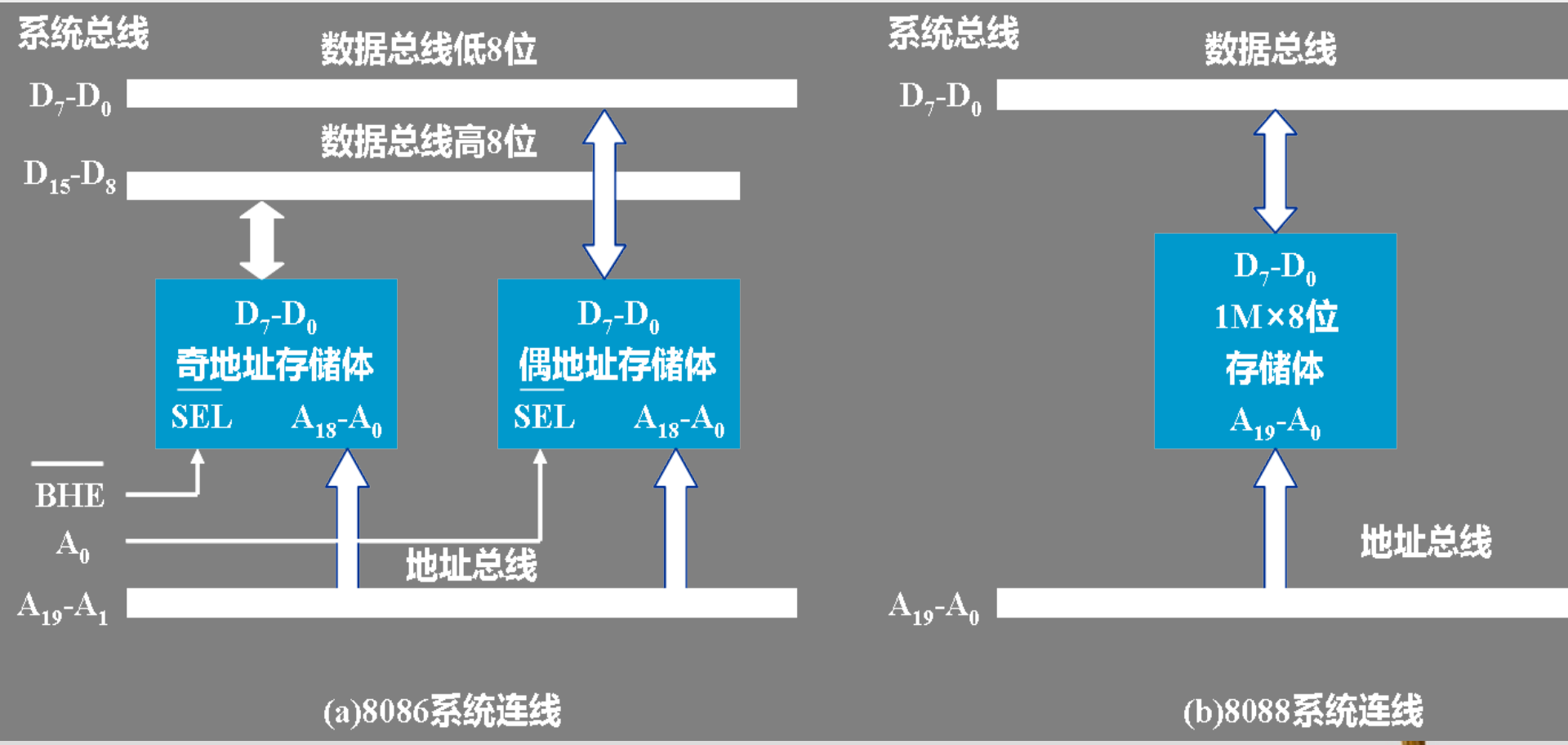
- 因此，存放字数据时，应放在偶地址开始单元中。对准伪指令EVEN能自动完成这种操作。

3．8088 CPU对存储器的存取操作

- 8088 的外部数据总线为8位，它每次访问存储器只读/写1个字节，读/写1个字要分2次完成。
- 1MB存储器被看作一个存储体，由A_{19}~A_0直接寻址，系统运行速度要慢些。

4. 8086/8088系统中存储器与总线的连接

- 连线如图2.9，左侧的系统总线是连到CPU的总线信号。



4. 8086/8088系统中存储器与总线的连接

图a)是8086系统存储器，分奇/偶地址体。

- 选择信号 $\overline{SEL}$ $\overline{BHE}$ 和A_0相连，选中1个存储体或两个都选中。
- 奇/偶地址体的8位数据线，分别与数据总线的高/低8位相连，传送高/低8位数据。
- 19根地址线A_{18}～A_0与地址总线的A_{19}～A_1相连，用来选择存储体内512KB单元中的某一个单元。

图b)是8088系统的1MB存储体。

- 8位数据线直接与低8位数据总线相连，20位地址线直接与20根地址总线相连。