

# 第6讲 从学生宿舍管理看存储器 ——基于编码的管理思维

---

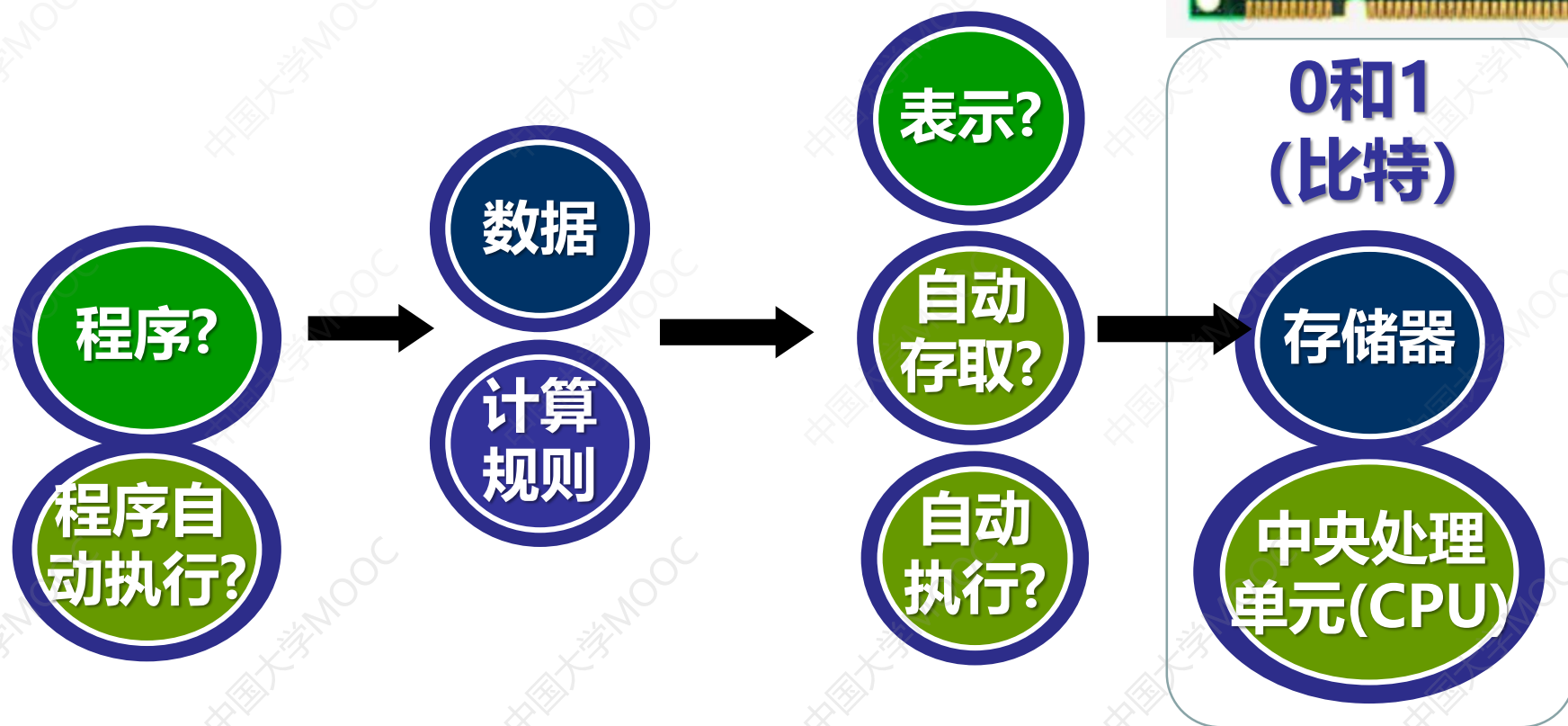
**战 德 臣**

哈尔滨工业大学计算学部教学委员会主任  
国家教学名师

18686783018, [dechen@hit.edu.cn](mailto:dechen@hit.edu.cn)

# 计算机器

回顾计算机器的基本思维

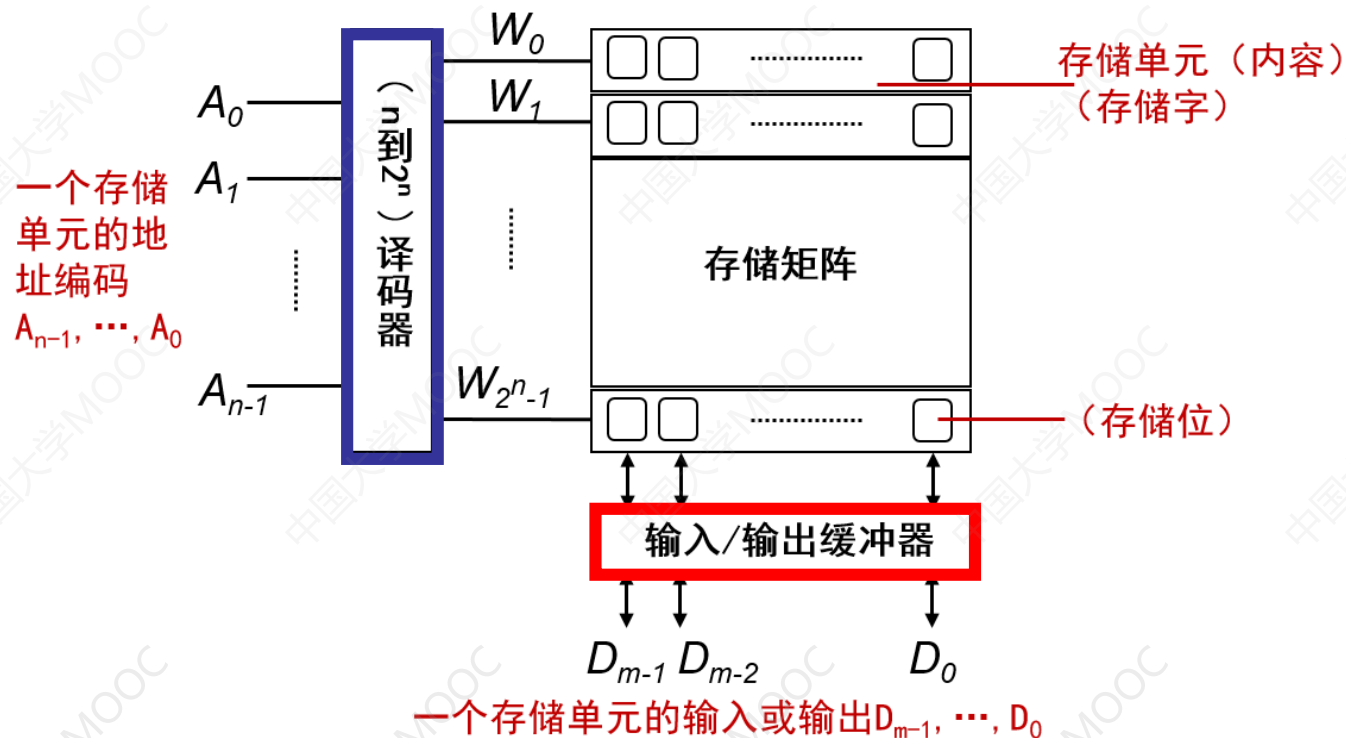


# 存储器的基本概念

3

## 什么是存储器？

- **【存储器】** 能够按地址读或写每一个存储单元的部件。
- **【存储单元】** 一个存储单元可以保存一个 $m$ 位的数据，通常记为 $D_{m-1} \dots D_0$ 。
- **【地址】** 是一个 $n$ 位的0/1编码，每一个编码指向一个存储单元，通常记为 $A_{n-1} \dots A_0$ 。



# 存储器的基本概念

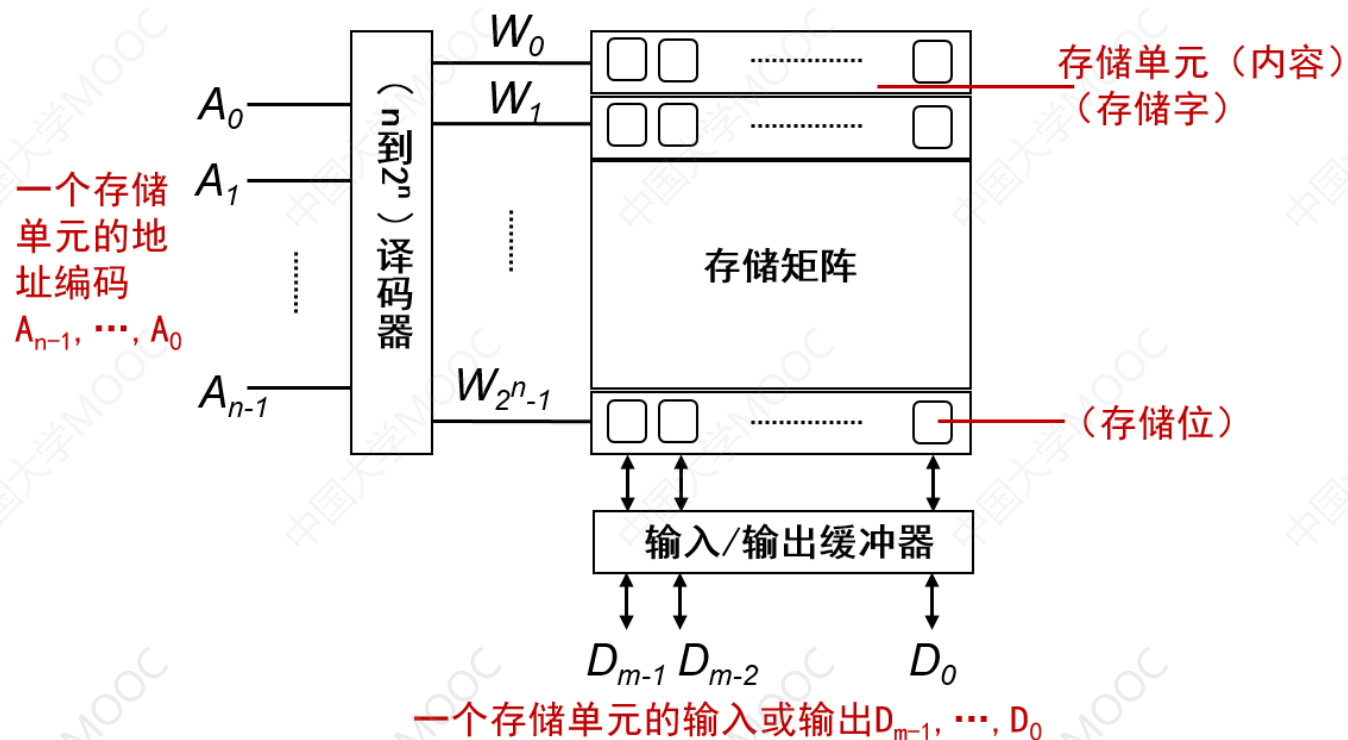
对比宿舍楼，理解存储器的相关概念

## 概念映射

存储器	宿舍楼
存储单元	房间
存储位(存0或存1)	床位(住人/不住人)
地址编码 $A_{n-1} \dots A_0$	房间号
单元控制线 $W_i$	房间钥匙
输出缓冲器	公共的走廊及大门
...	...

从存储器与宿舍楼的概念对比中，你能发现什么异同吗？

同一时刻只能有一个 $W$ 有效（接通存储单元）



# 存储器的基本概念

## 存储器的容量

地址编码n位:  $A_{n-1} \dots A_0$

地址空间:

00000000...00000000

00000000...00000001

... ..

11111111...11111111

2<sup>n</sup>个存储单元

存储字长m位:  $D_{m-1} \dots D_0$

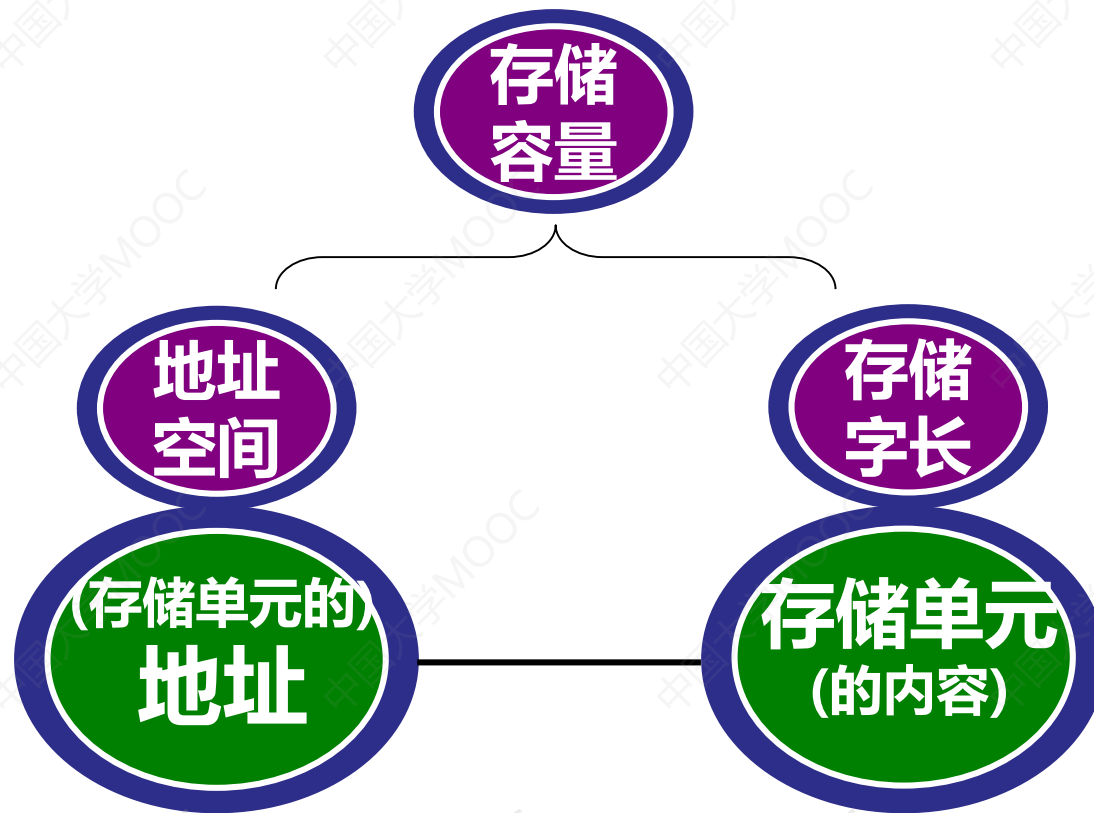
每个存储单元都是m位

存储容量 =  $2^n \times m$  Bit(位)

=  $2^n \times m/8$  Byte (字节)

存储容量的单位: 2<sup>10</sup>为换算单位

DB、NB、BB、YB、ZB、EB、PB、TB、GB、MB、KB、Byte

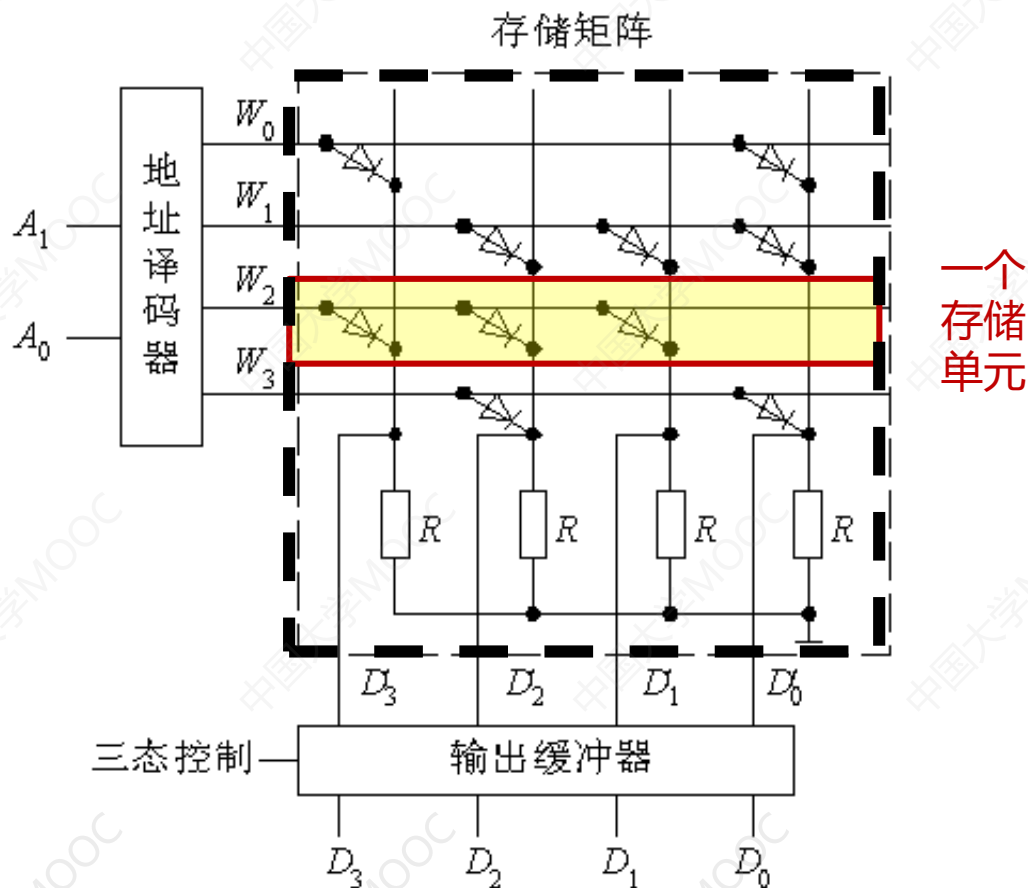


# 自动存取：存储器的工作原理

存储器是怎样存储0和1的？又是怎样控制存取的？

## 存储器内部的实现示例

◆当单元控制线W和数据线D间连接有二极管时，则存储的是1，否则，存储的是0。这是只读存储器（ROM，只能读出不能写入）示例。



二极管ROM结构示例  
(2位地址控制4个存储单元, 每个存储单元是4位0/1)

# 自动存取：存储器的工作原理

7

存储器是怎样存储0和1的？又是怎样控制存取的？

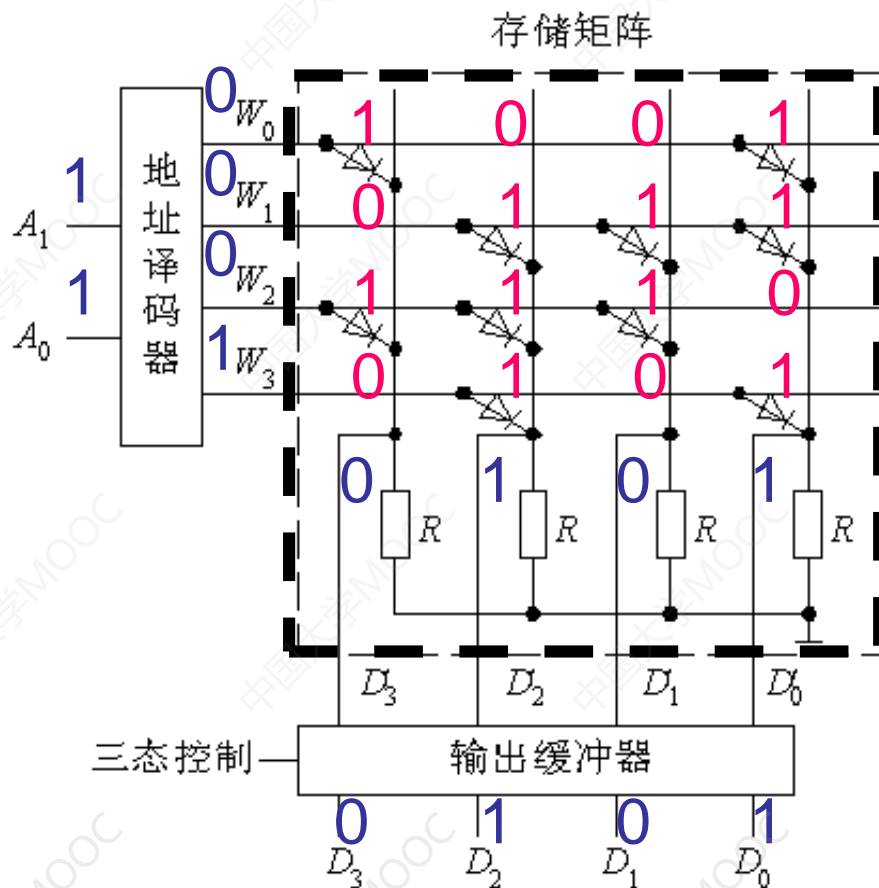
## 存储器内部的实现示例

◆当单元控制线W和数据线D间连接有二极管时，则存储的是1，否则，存储的是0。这是只读存储器（ROM，只能读出不能写入）示例。

■当单元控制线W和数据线D间连接有二极管时，由单元控制线决定其是输出1或0，即：当单元控制线为1（高电压）时，则输出1（高电压），而当单元控制线为0（低电压）时，则输出0（低电压）。没有连接的，则不受单元控制线影响，始终输出0（低电压）。

■ $W_3, W_2, W_1, W_0$ 随着 $A_1 A_0$ 的值同时只能有一个为1（高电压）其它为0（低电压），即控制一个存储单元所有位的读写。

■尽管所有存储单元的第i位都连接到 $D_i$ ，但只有将读取存储单元的第i位对 $D_i$ 产生作用。



二极管ROM结构示例

(2位地址控制4个信息单元, 每个信息单元是4位0/1码)

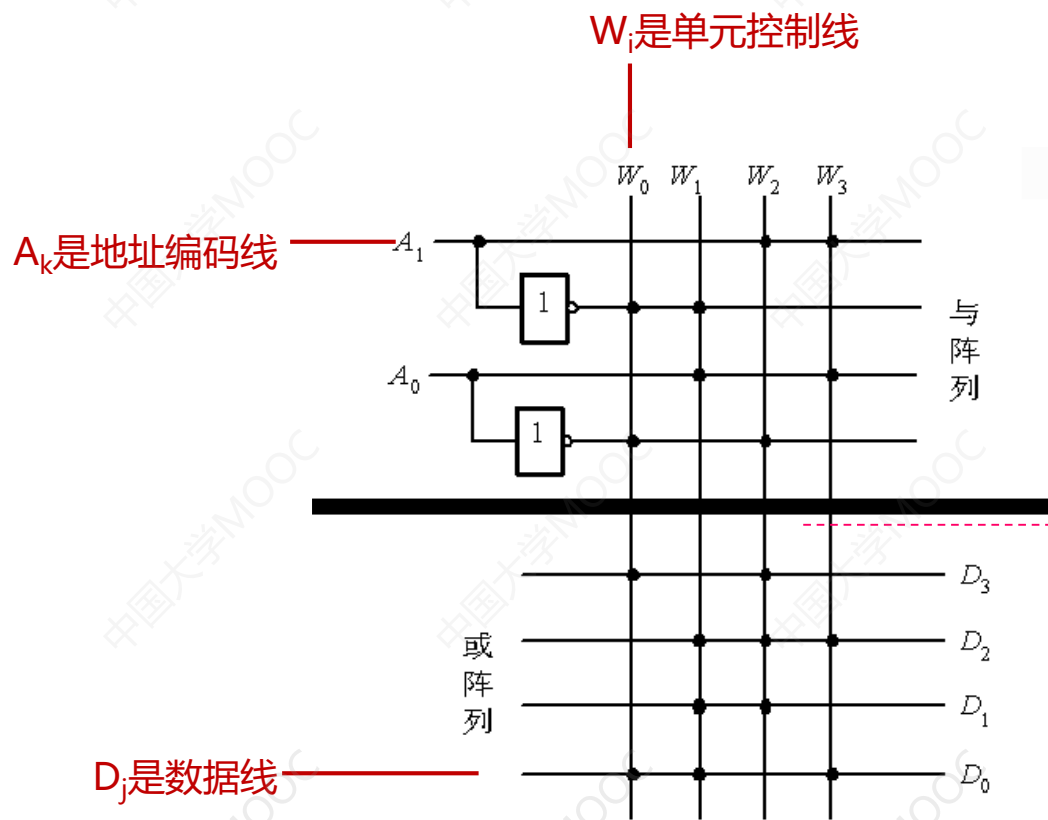


# 存储器实现的逻辑控制

## 存储矩阵的逻辑控制关系

### 存储矩阵：一种【与或逻辑】阵列

- 【地址编码线】与【单元控制线】有黑点则连接，无黑点则不连接。
- 【单元控制线】与【数据线】有黑点则连接，无黑点不连接。
- 高/低电压信号，即0,1，通过连接点相互传递。
- 上半部是【与】阵列，下半部是【或】阵列。





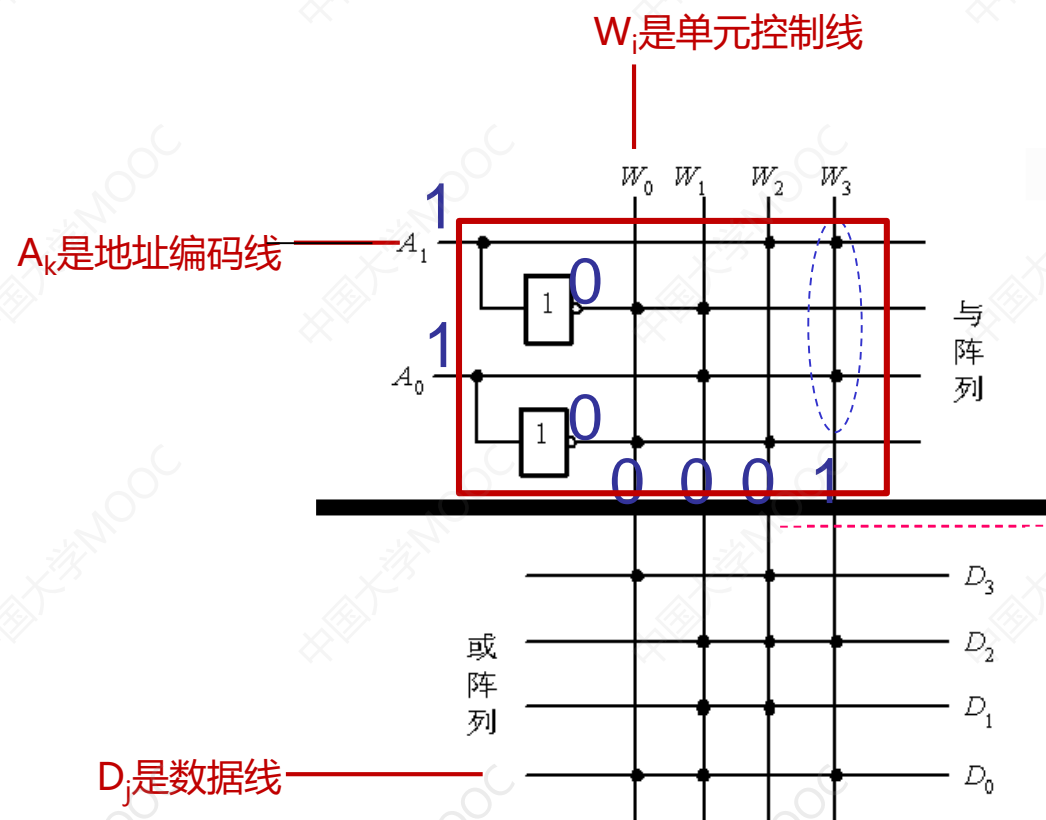
# 存储器实现的逻辑控制

## 存储矩阵的逻辑控制关系

译码器：将 $A_1A_0$ 地址编码翻译到只有一条单元控制线 $W_i$ 为1,其它为0。

- 【与】阵列，表述了如何由 $A_1A_0$ 的值产生 $W_0, W_1, W_2, W_3$ 的值（只能有一条线为1），由横线向纵线传输信号——上半部。
- 同一【单元控制线】上各连接点之间是“与”关系。即只有各连接点都为1时，该单元控制线的信号为1，否则为0。

$$\begin{aligned}W_0 &= (\text{NOT } A_0) \text{ AND } (\text{NOT } A_1) \\W_1 &= A_0 \text{ AND } (\text{NOT } A_1) \\W_2 &= (\text{NOT } A_0) \text{ AND } A_1 \\W_3 &= A_0 \text{ AND } A_1\end{aligned}$$



# 存储器实现的逻辑控制

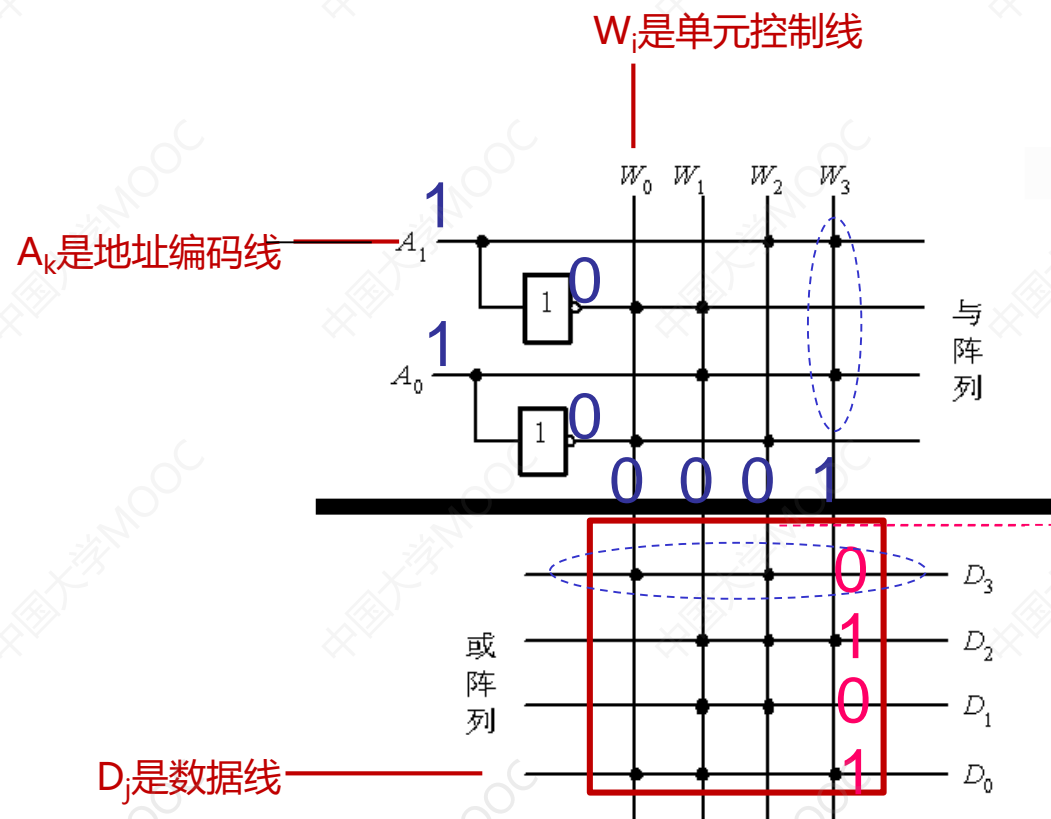
10

## 存储矩阵的逻辑控制关系

输出矩阵：将 $W_i$ 为1的单元控制线控制的存储单元的值进行输出。

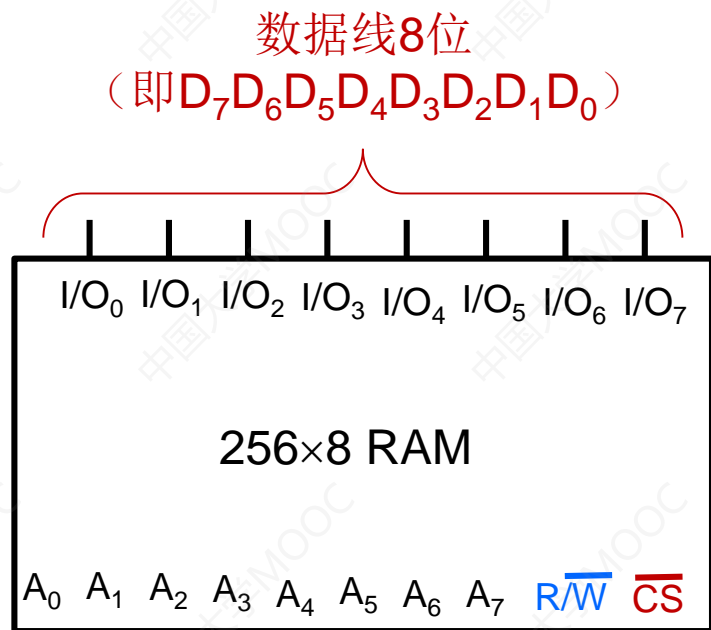
- 【或】阵列，表述了如何由 $W_3W_2W_1W_0$ 的值产生 $D_3, D_2, D_1, D_0$ 的值。单元控制线与数据线之间有黑点连接的，表示存储的是1，其能否输出取决于单元控制线是1还是0。由纵线向横线传输信号一下半部。
- 同一【数据线】上各连接点之间是“或”关系。即各连接点只要有一个为1时，该数据线的信号为1，否则为0。

$$\begin{aligned} D_3 &= W_0 \text{ OR } W_2 \\ D_2 &= W_1 \text{ OR } W_2 \text{ OR } W_3 \\ D_1 &= W_1 \text{ OR } W_2 \\ D_0 &= W_0 \text{ OR } W_1 \text{ OR } W_3 \end{aligned}$$



# 存储器容量不够怎么办？

## 存储器芯片 vs. 标准宿舍楼

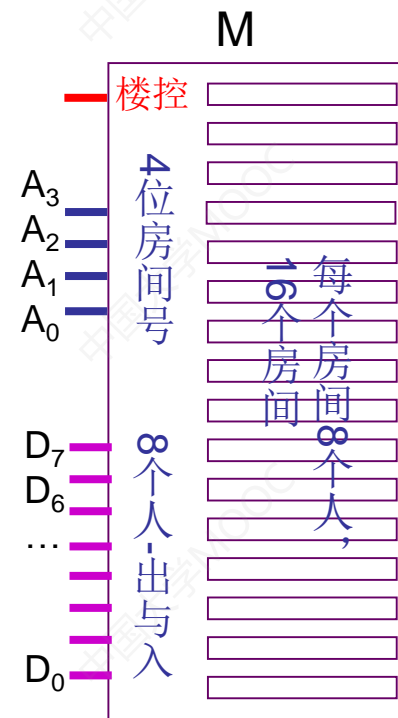


地址编码线8位  
(编码256个地址)

芯片是否工作的控制。  
当加高电压(1)时不工作，  
当加低电压(0)时工作。

控制存储单元的读或写  
当加高电压(1)时读出，当  
加低电压(0)时写入。

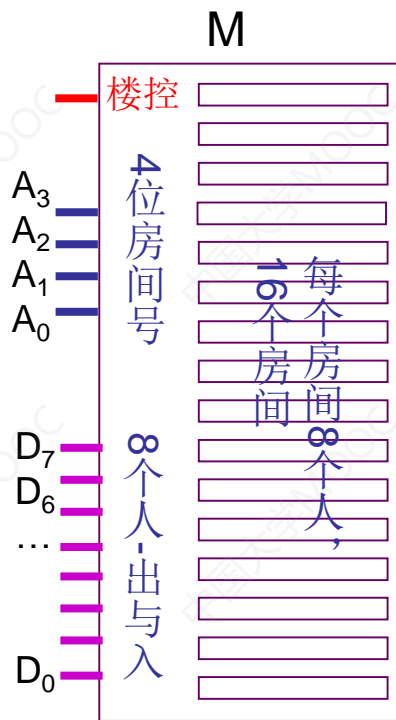
$A_3A_2A_1A_0$   
0000  
0001  
...  
0111  
1000  
1001  
...  
1111



16×8 标准宿舍楼

# 存储器容量不够怎么办？

一个宿舍楼不够怎么办？



16×8 标准宿舍楼

一个标准宿舍楼：16个房间，每个房间住8人  
 $A_3 A_2 A_1 A_0$        $D_7 \dots D_0$

住宿需求：64个大房间，每个大房间住16人？  
 $B_5 B_4 B_3 B_2 B_1 B_0$        $E_{15} \dots E_8 E_7 \dots E_0$

# 存储器容量不够怎么办？

多个宿舍楼拼接在一起使用

住宿需求：

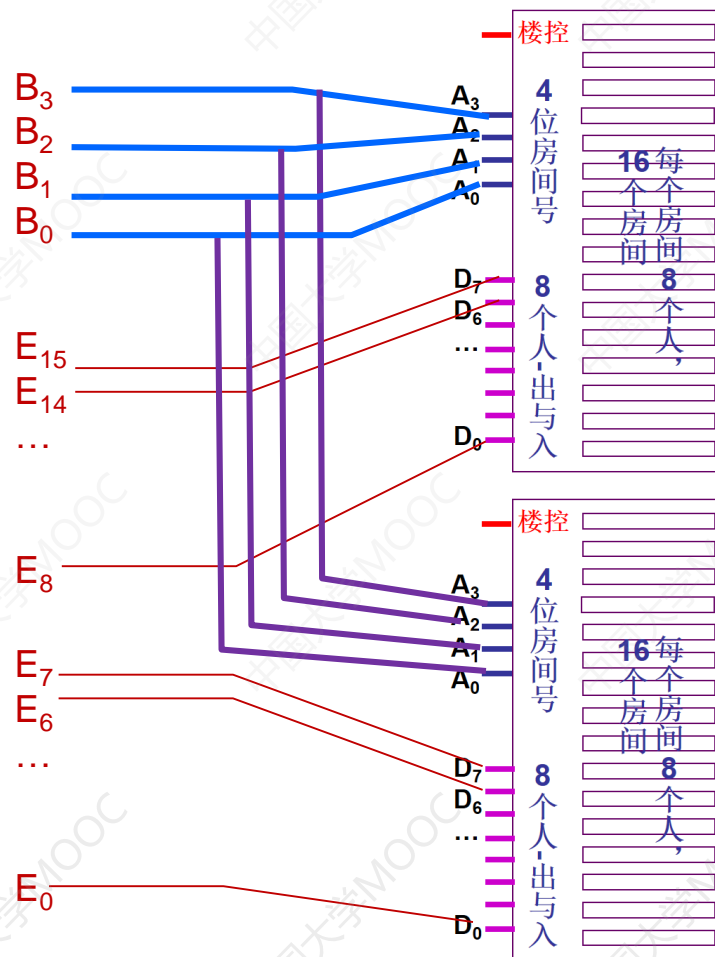
64个大房间，每个大房间住16人？

$B_5 B_4 B_3 B_2 B_1 B_0$

$E_{15} \dots E_8 E_7 \dots E_0$

解决方案第1步：

2个标准宿舍楼对应相同地址 $B_3 B_2 B_1 B_0$ 的房间构成一个大房间，其中编号为 $E_{15} \dots E_8$ 住上面的楼，而 $E_7 \dots E_0$ 住下面的楼，确保他们的地址 $A_3 A_2 A_1 A_0$ 在同一时刻是相同的，即同时进出。



### 住宿需求:

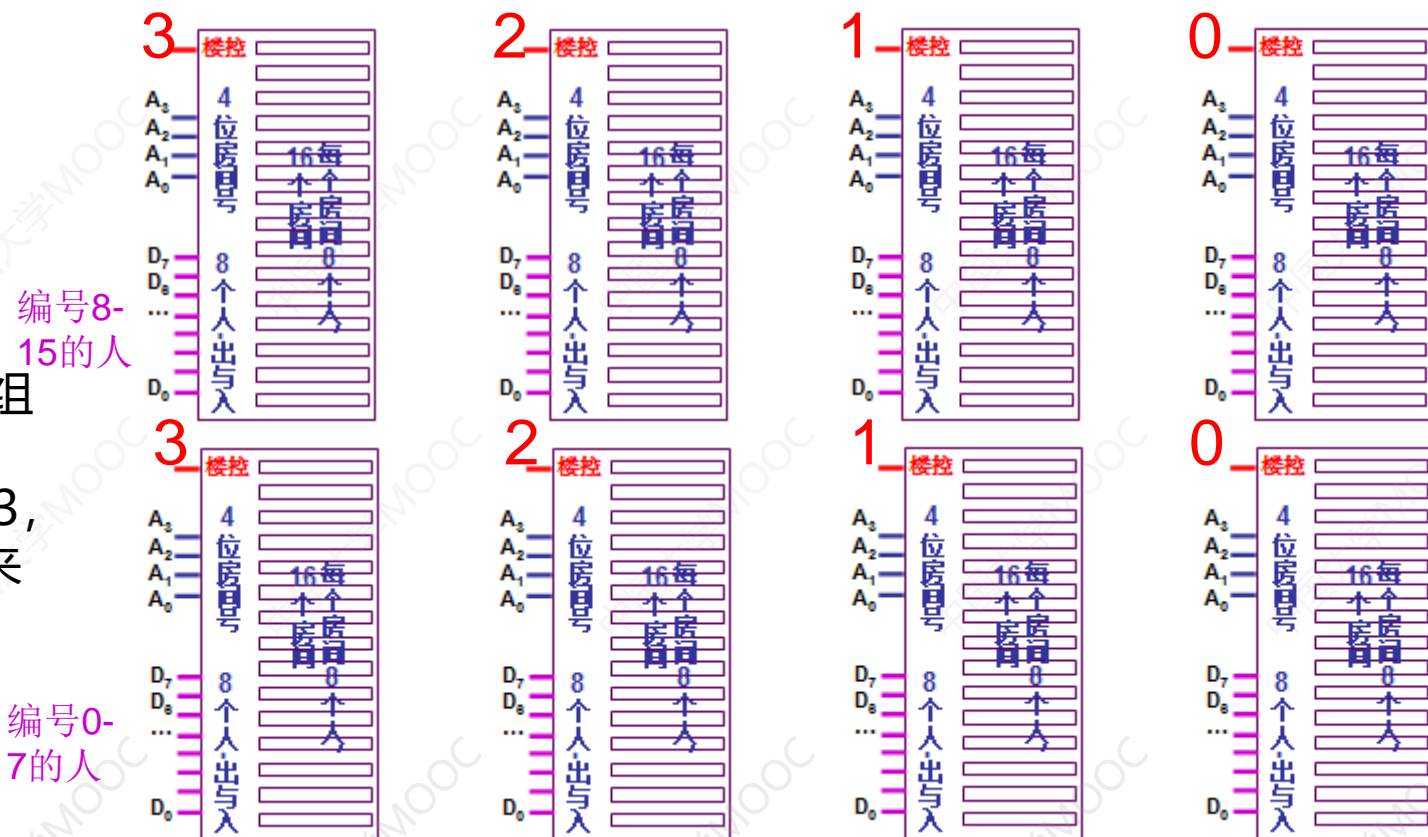
64个大房间，每个大房间住16人？

$$B_5 B_4 B_3 B_2 B_1 B_0$$
$$E_{15} \dots E_8 E_7 \dots E_0$$

### 解决方案第2步:

2个标准宿舍楼为一组，64个房间则需要4组（每组16个房间）。

用楼控信号控制哪一组。楼控信号为0,1,2,3, 使用 $B_5B_4$ 来产生楼控信号, 使用 $B_3B_2B_1B_0$ 来产生楼内的地址。

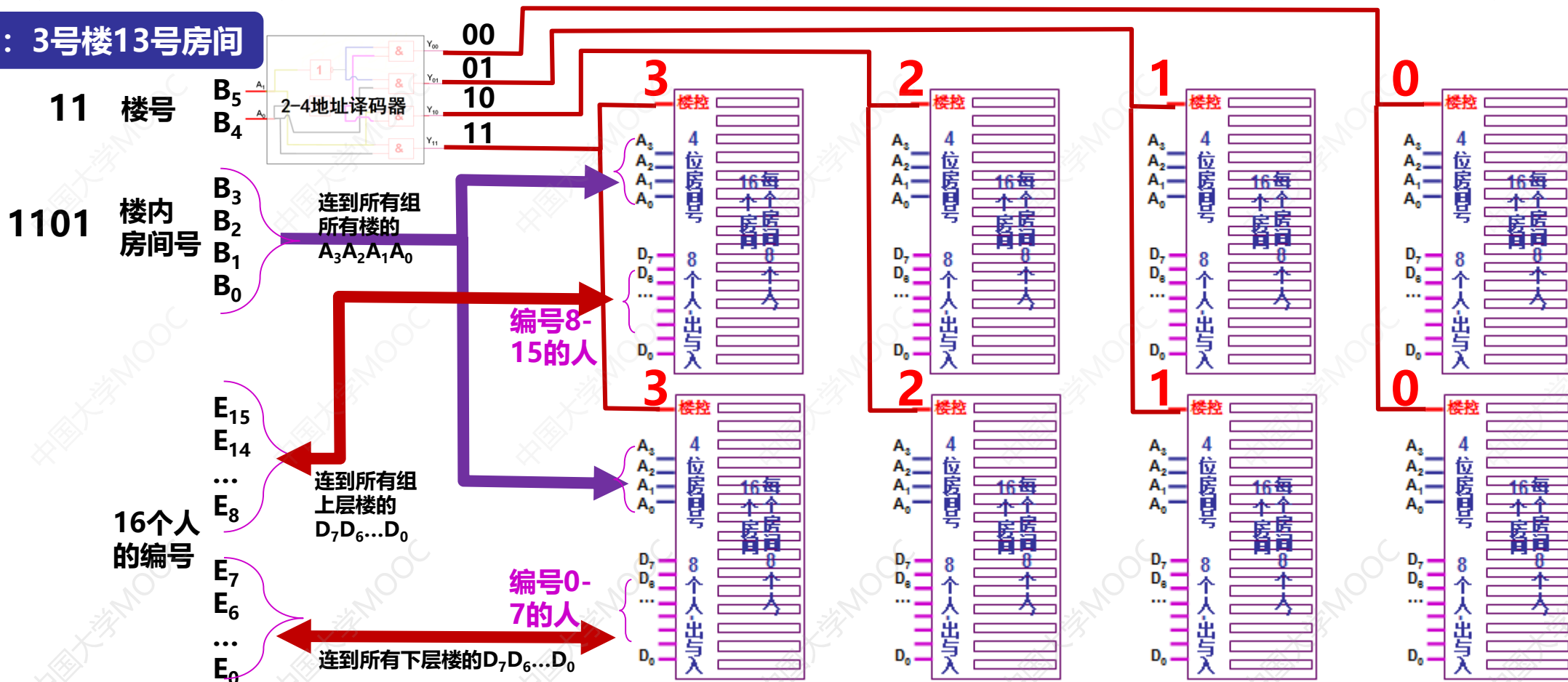


# 存储器容量不够怎么办？

用多个标准宿舍楼可组合出容量更大的宿舍楼

利用 8 个 16x8 的标准楼 扩展出 64x16 的宿舍楼的编码控制图

11-1101: 3号楼13号房间





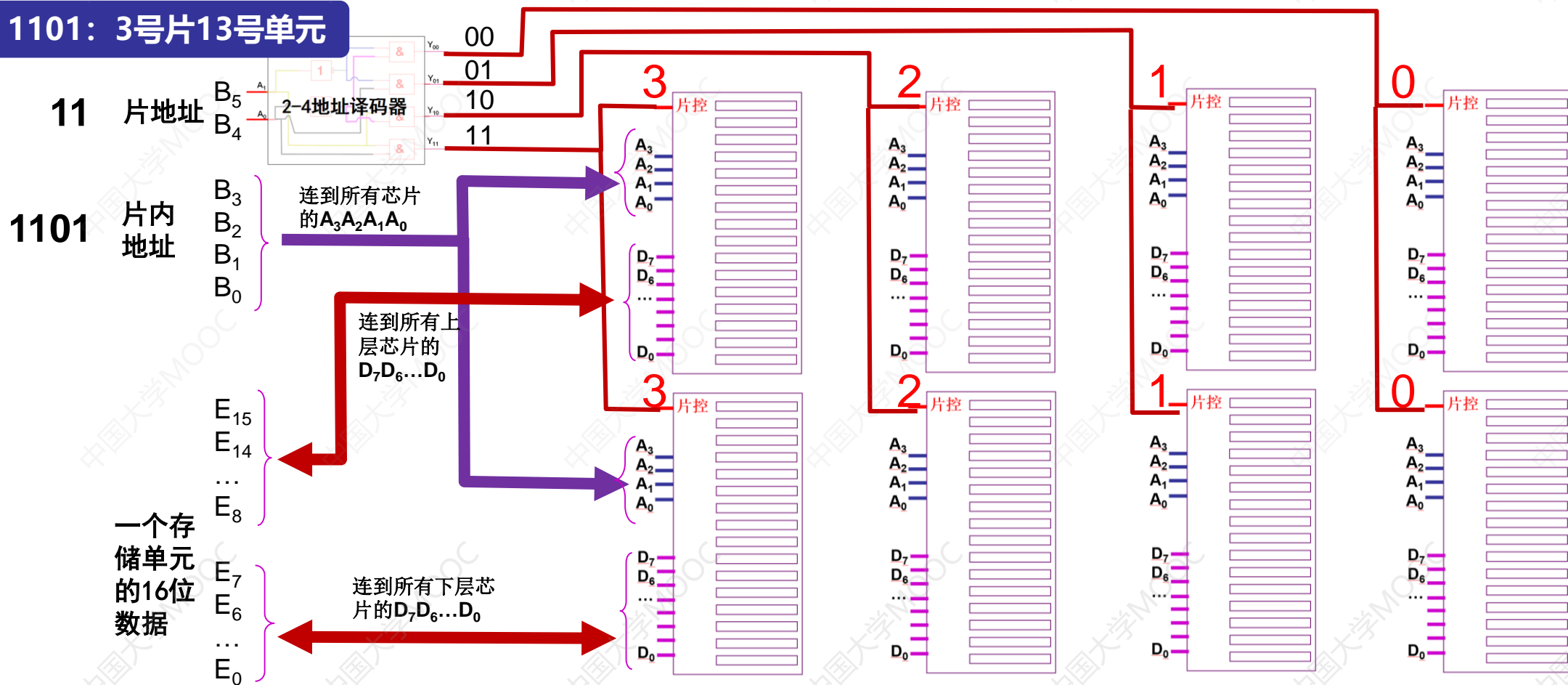
# 存储器容量不够怎么办？

用多个存储器芯片可搭建容量更大的存储器

## 利用8个16x8存储器芯片扩展出64x16位存储器的电路控制图



11 1101: 3号片13号单元



# 从宿舍楼管理到存储器芯片

【看山不是山、看水不是水】

既是学习芯片扩展的思维，也是学习科学管理的思维

