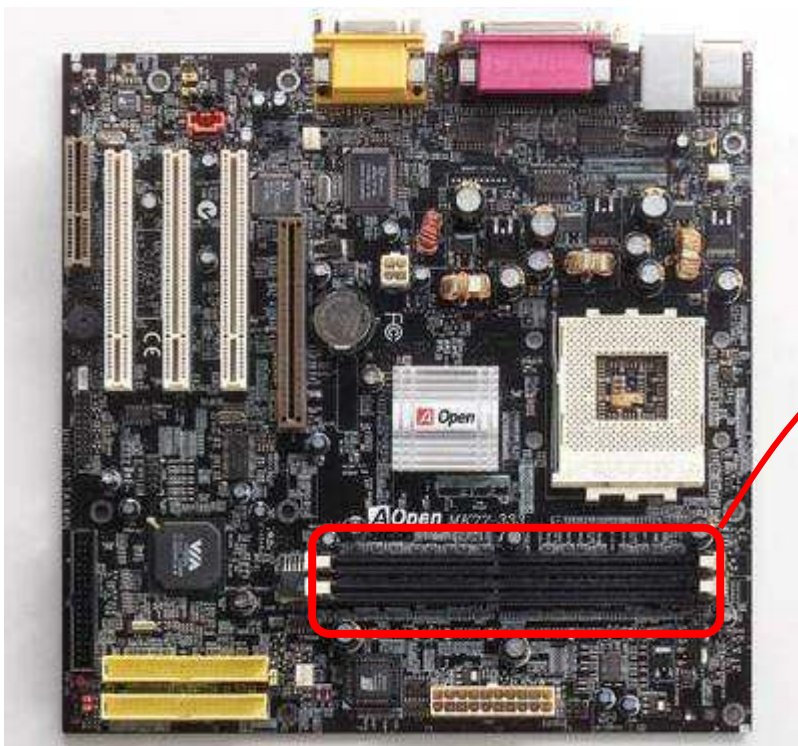


## 概述

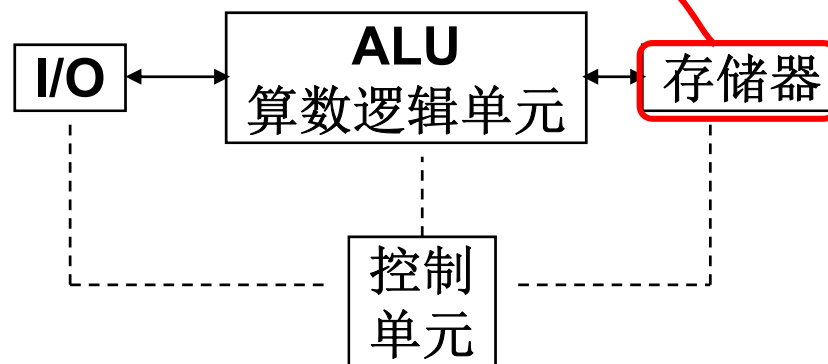
计算机 { 软件系统 指令集  
硬件系统



*Read only memory programmable erasable*

ROM, PROM, EPROM,  
RAM, SRAM, DRAM

*random static dynamic*



取指令，译码，取数，算数逻辑运算，存储

## **第五章 半导体存储器电路**

### **5.5.1 SRAM**

### **5.5.2 DRAM**

### **5.5.3 ROM**

### **5.5.4 存储器容量的扩展**

### **5.5.5 用存储器实现组合逻辑函数**

## 5.5存储器

半导体存储器能存储大量二值信息，是数字系统不可缺少的部分。

1. 衡量指标 { 存储容量  
存取速度

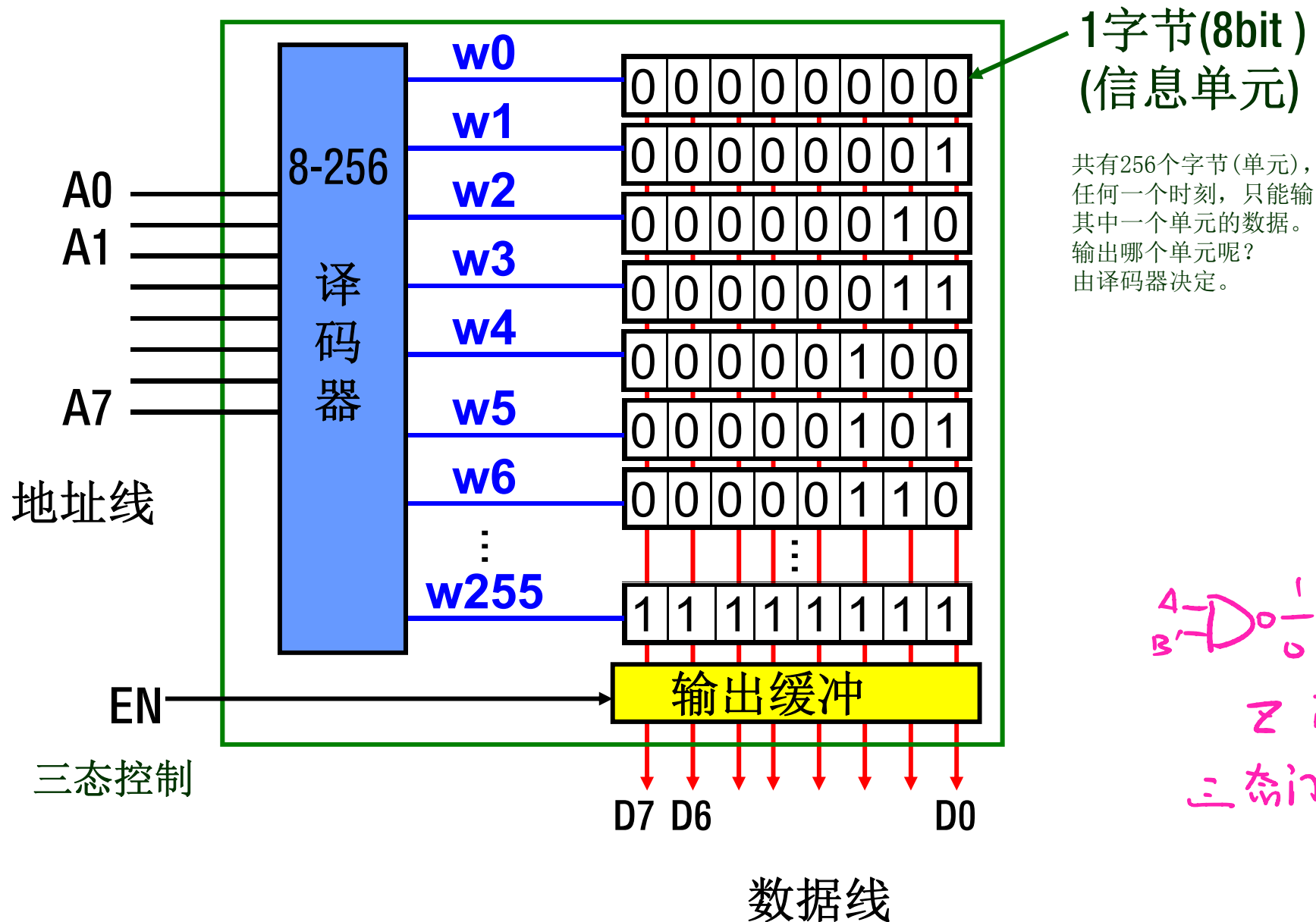
2. 种类

从读写方式分: { 只读存储器(**ROM**)  
(Read-Only Memory)  
随机存储器(**RAM**)  
(Random Access Memory )

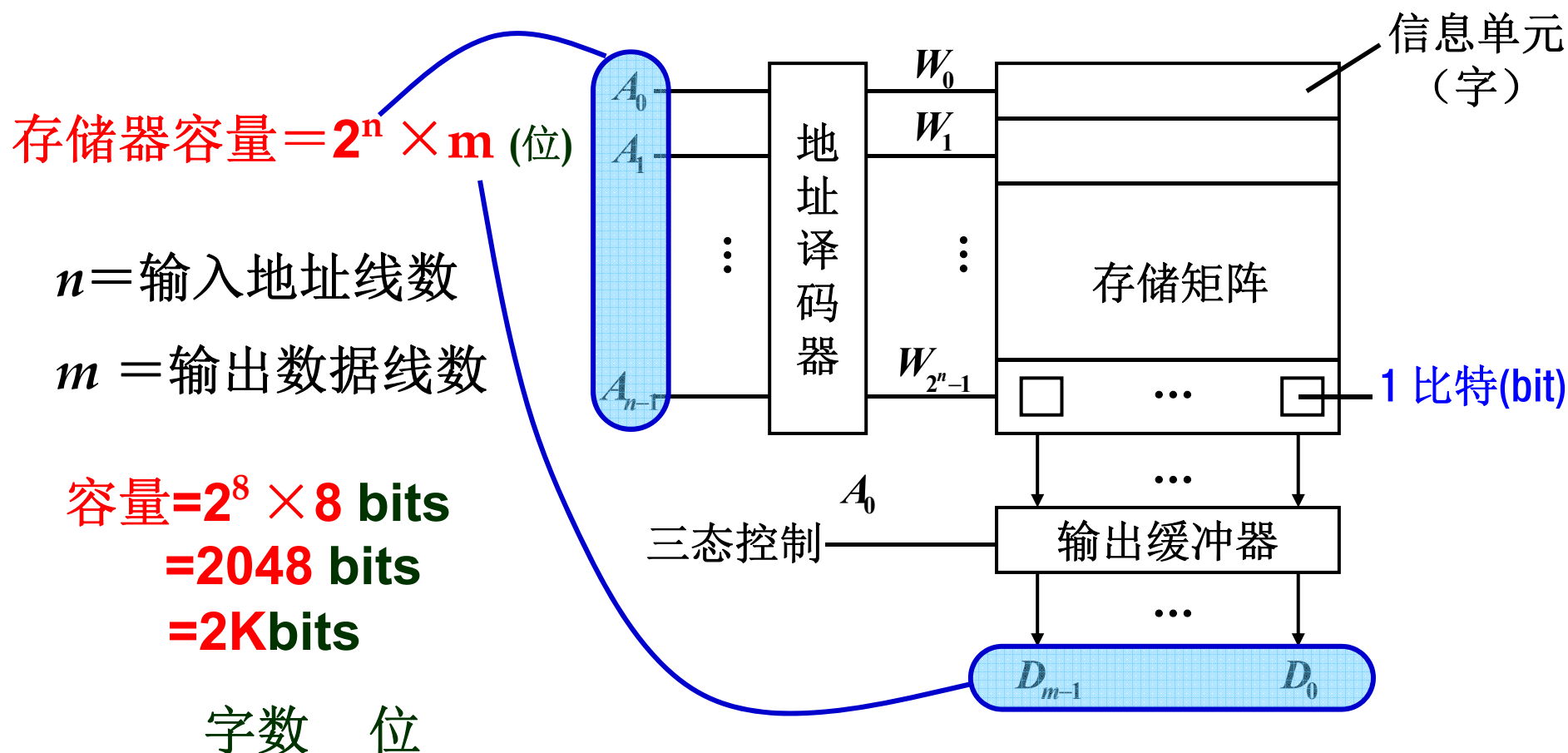
U盘属于哪种?

从工艺上分: { 双极型  
MOS型

### 5.5.3 只读存储器ROM (Read-Only Memory)



### 5.5.3 只读存储器ROM (Read-Only Memory)



例如  $1K \times 4$ 、 $2K \times 8$  和  $64K \times 1$  的存储器，其容量分别是  $1024 \times 4$  位、 $2048 \times 8$  位 和  $65536 \times 1$  位。

练习1:

如果ROM存储器有32位地址线，16个数据线，  
问存储器容量是多少bit?

$$2^{32} \times 16 = 4G \times 16 = 64Gbit$$

$$\begin{array}{ll} 2^{10} & 1k \\ 2^{20} & 1M \\ 2^{30} & 1G \end{array}$$

练习2:

已知ROM容量如下，问有多少地址线？多少数据线？

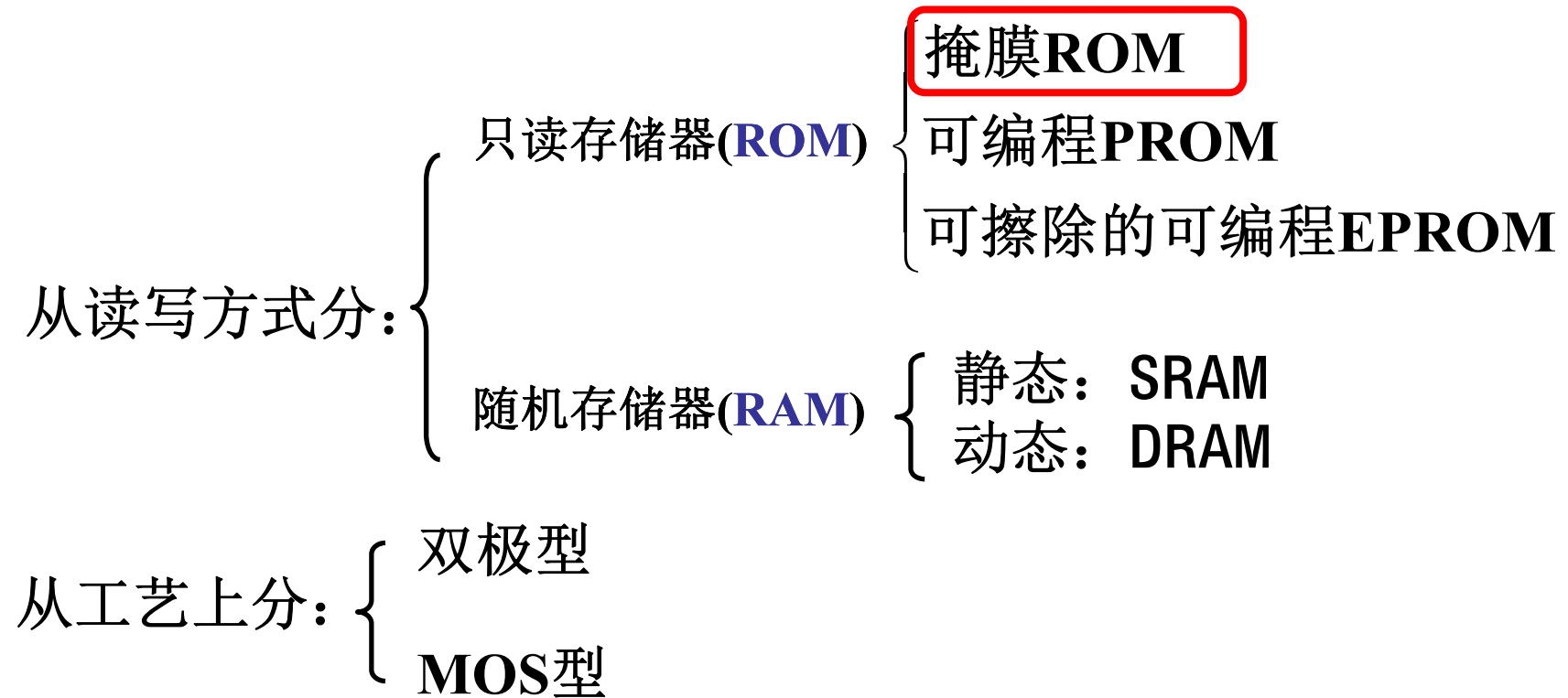
$$1K = 1024$$

$$64K \times 1$$

$$256K \times 4$$

$$1M \times 8$$

### 5.5.3 只读存储器ROM (Read-Only Memory)



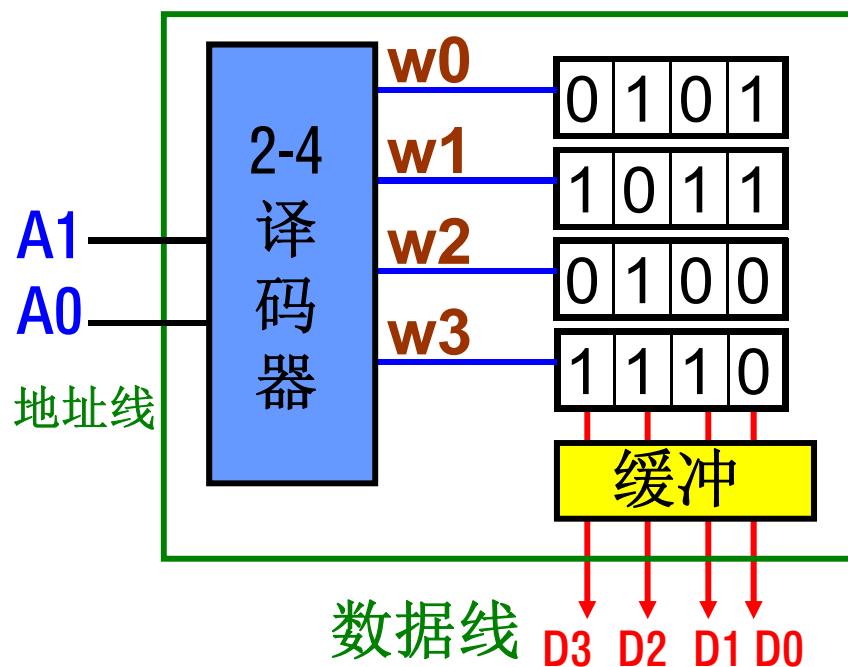
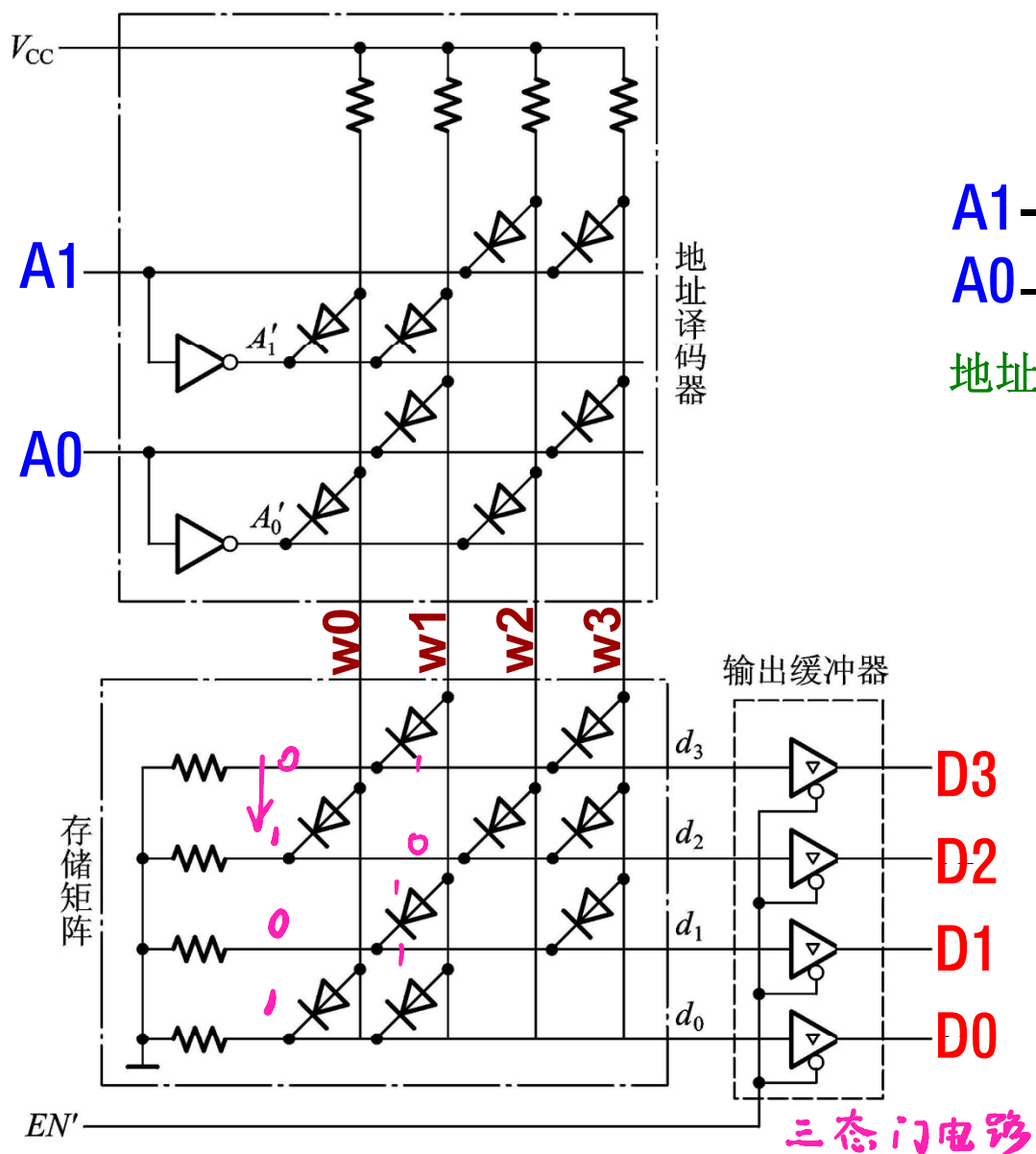
## 一、掩模只读存储器

掩模ROM在出厂时，其内部存储的信息就已经“固化”在里边了，所以也称固定ROM。它在使用时只能读出，不能写入，因此通常只用来存放固定数据、固定程序和函数表等。具有适合大量生产，简单，便宜，非易失性等特点。

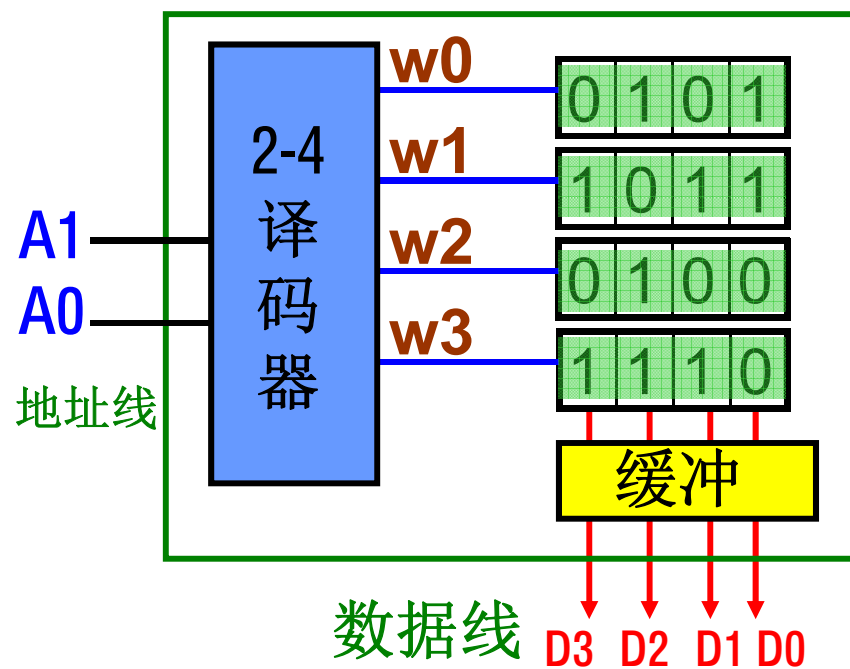
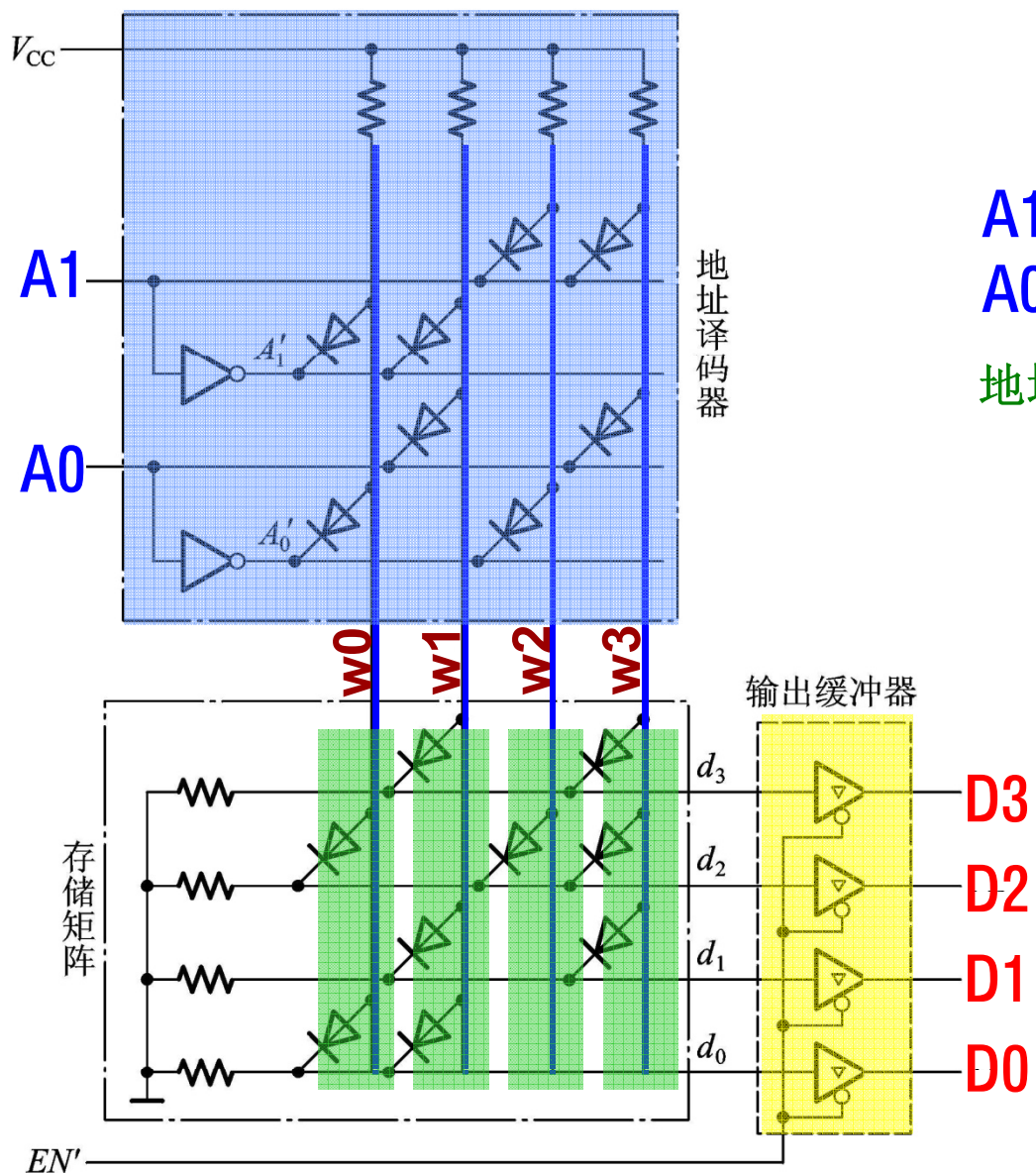
- ROM {
1. 二极管构成的ROM PN结
  2. MOS管构成的ROM 金属氧化物二极管



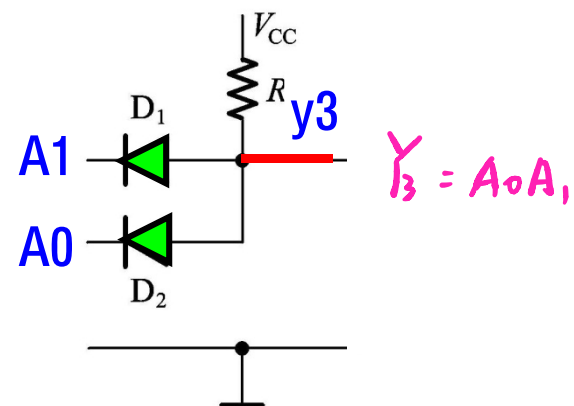
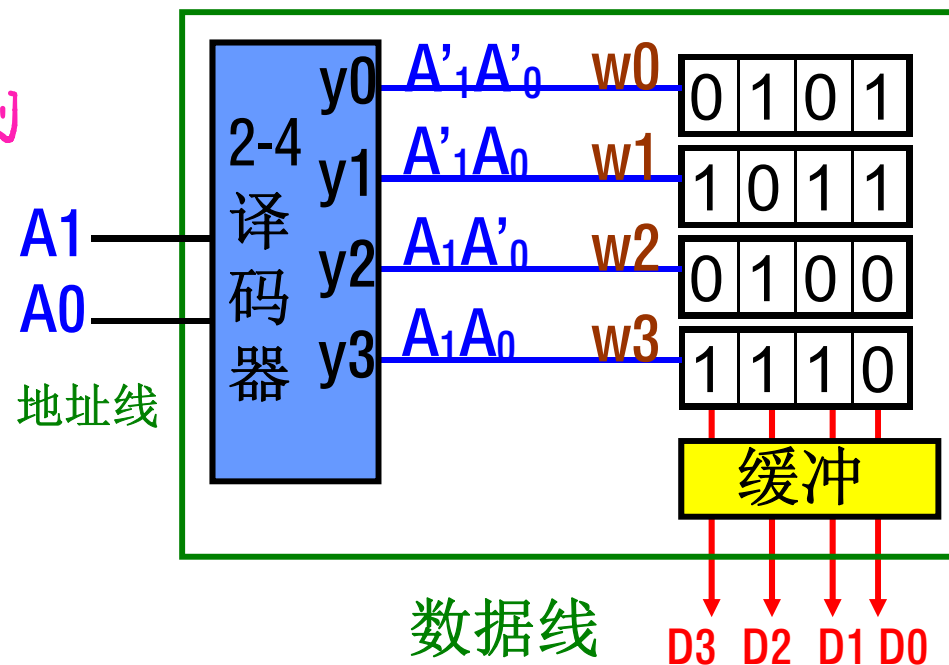
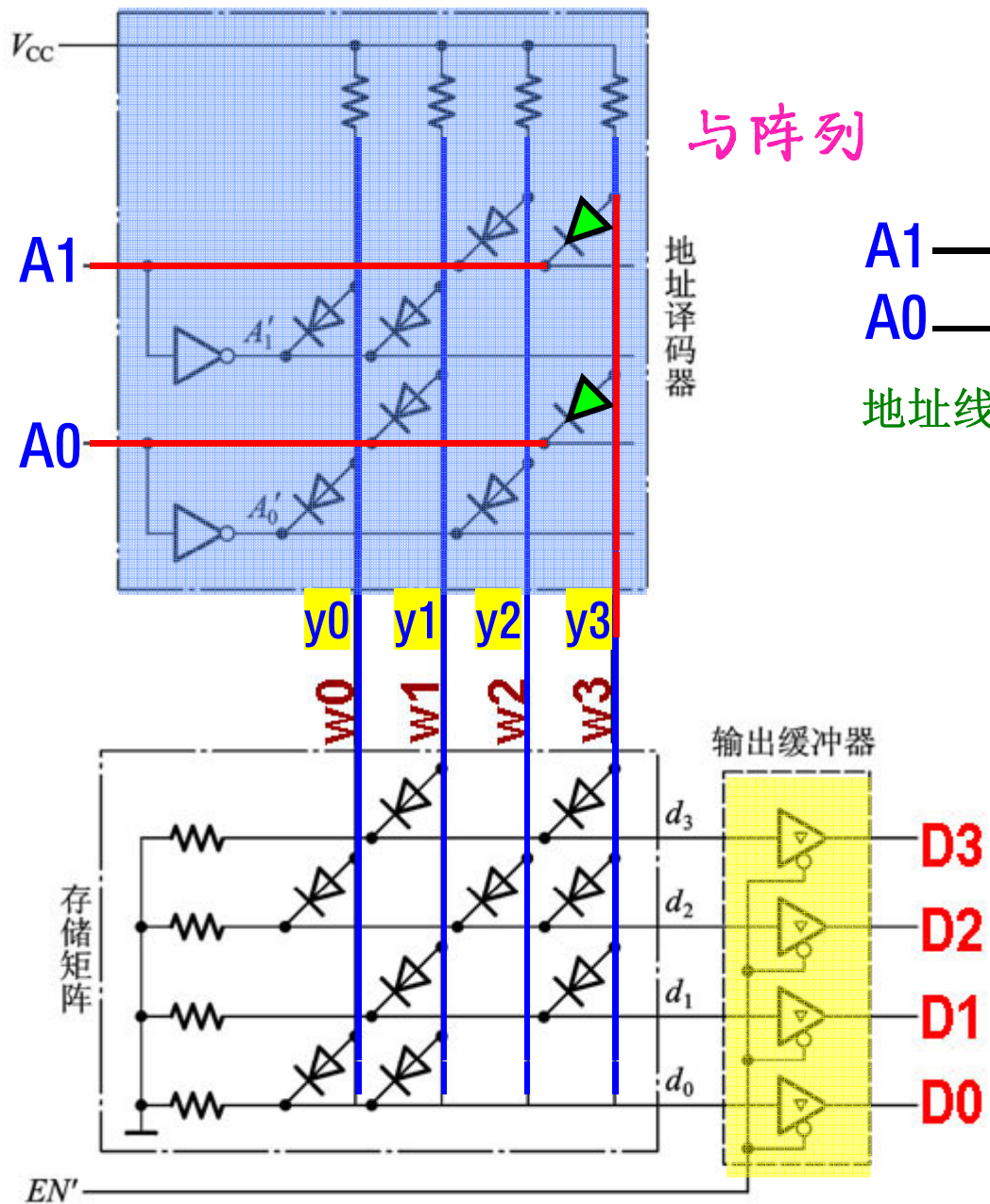
# 1. 二极管ROM电路结构



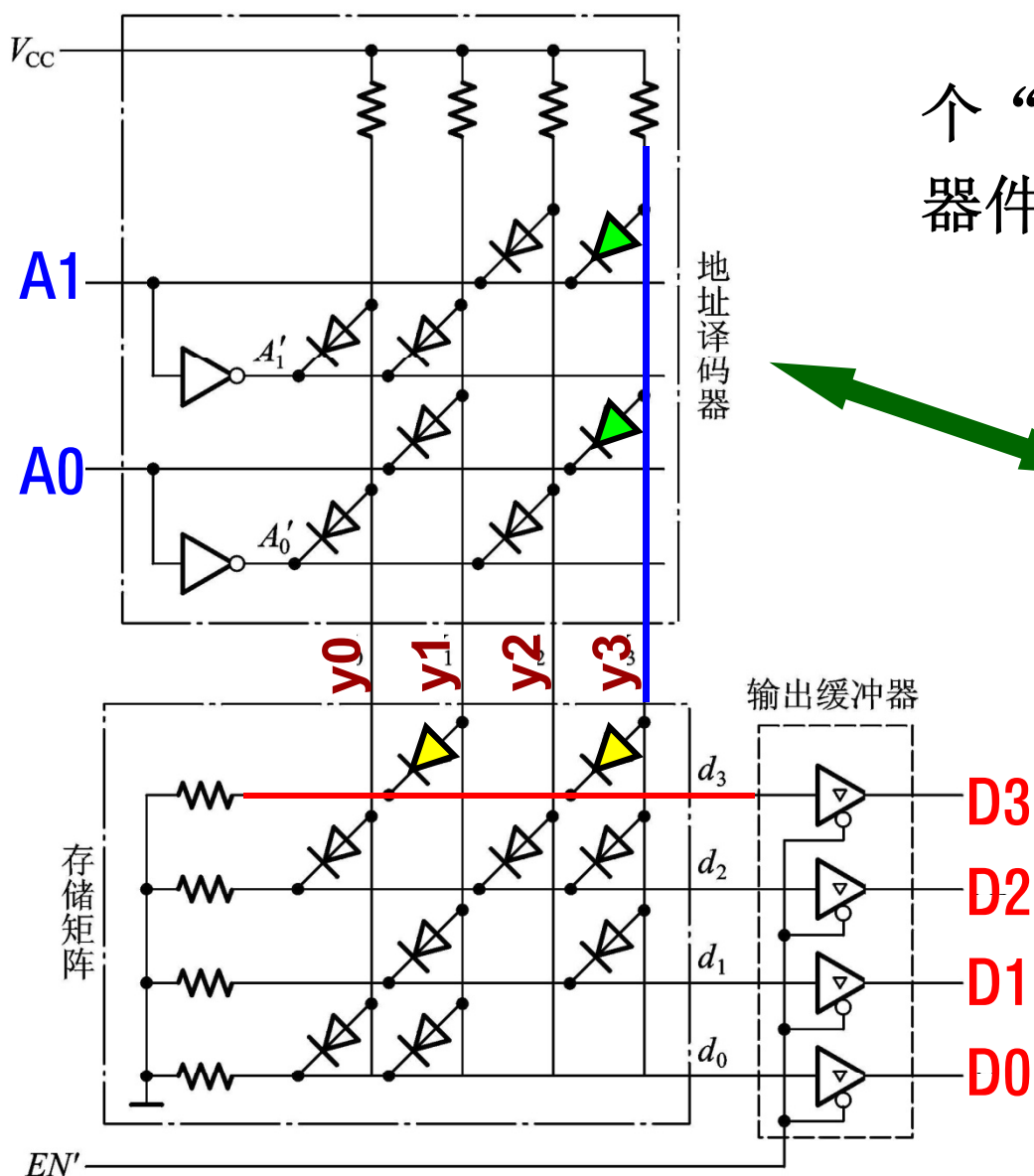
# 1. 二极管ROM电路结构



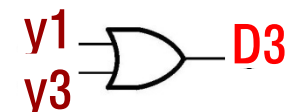
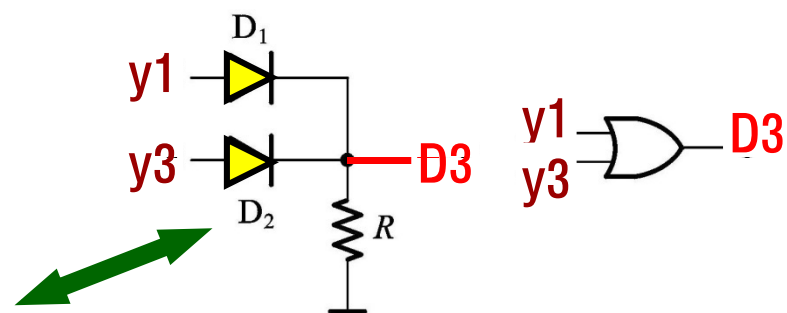
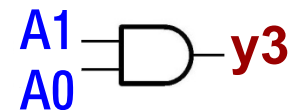
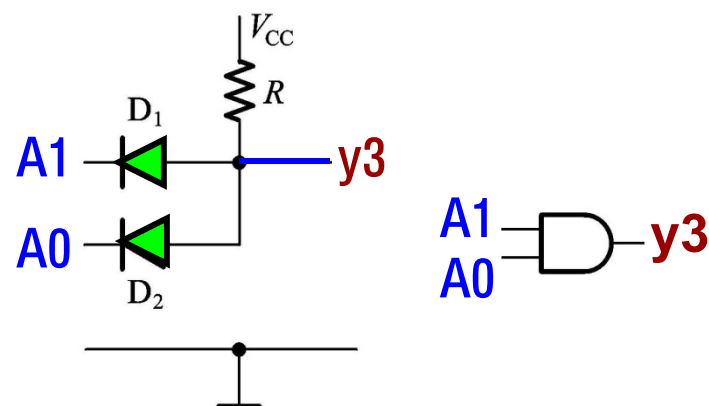
# 1. 二极管ROM电路结构

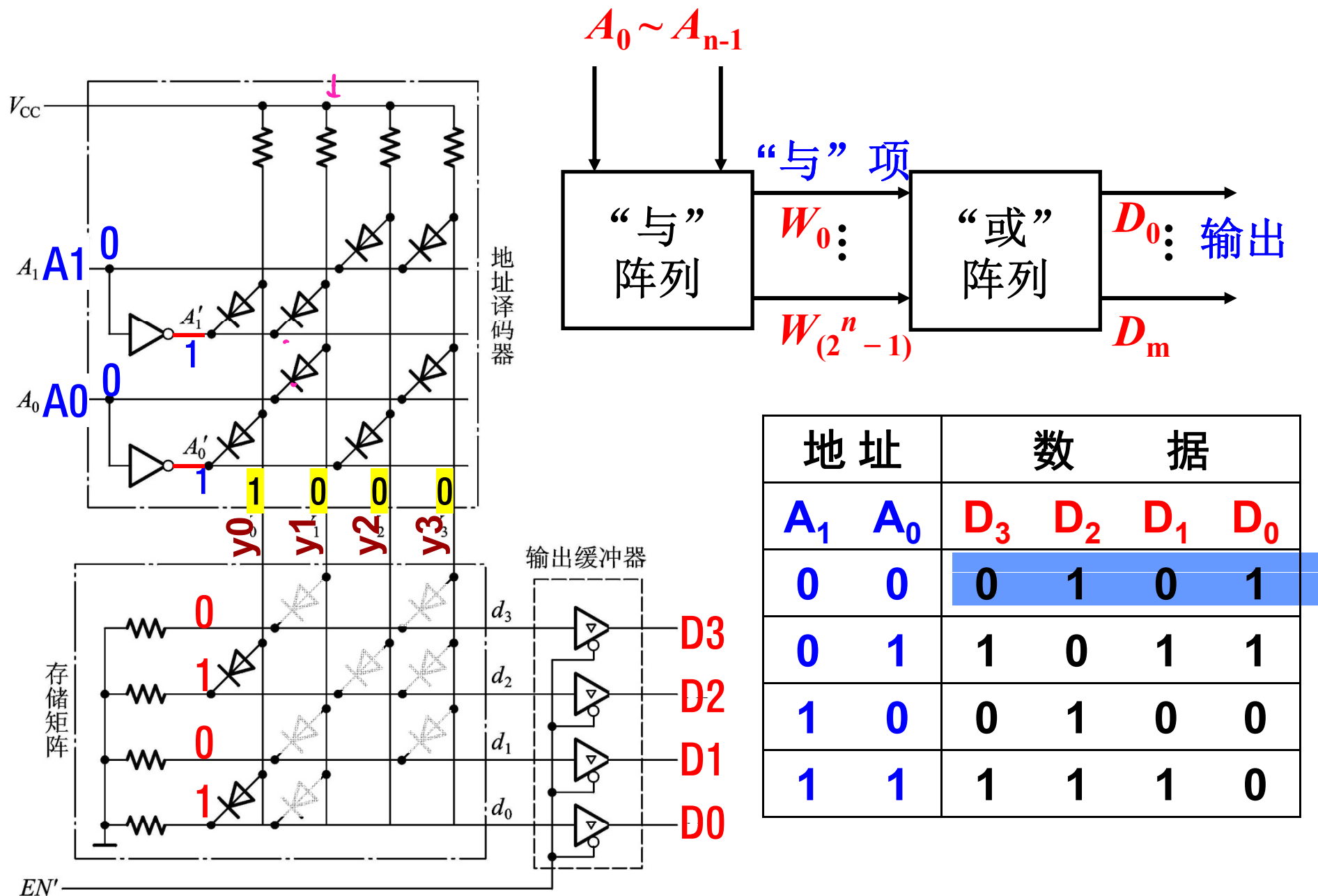


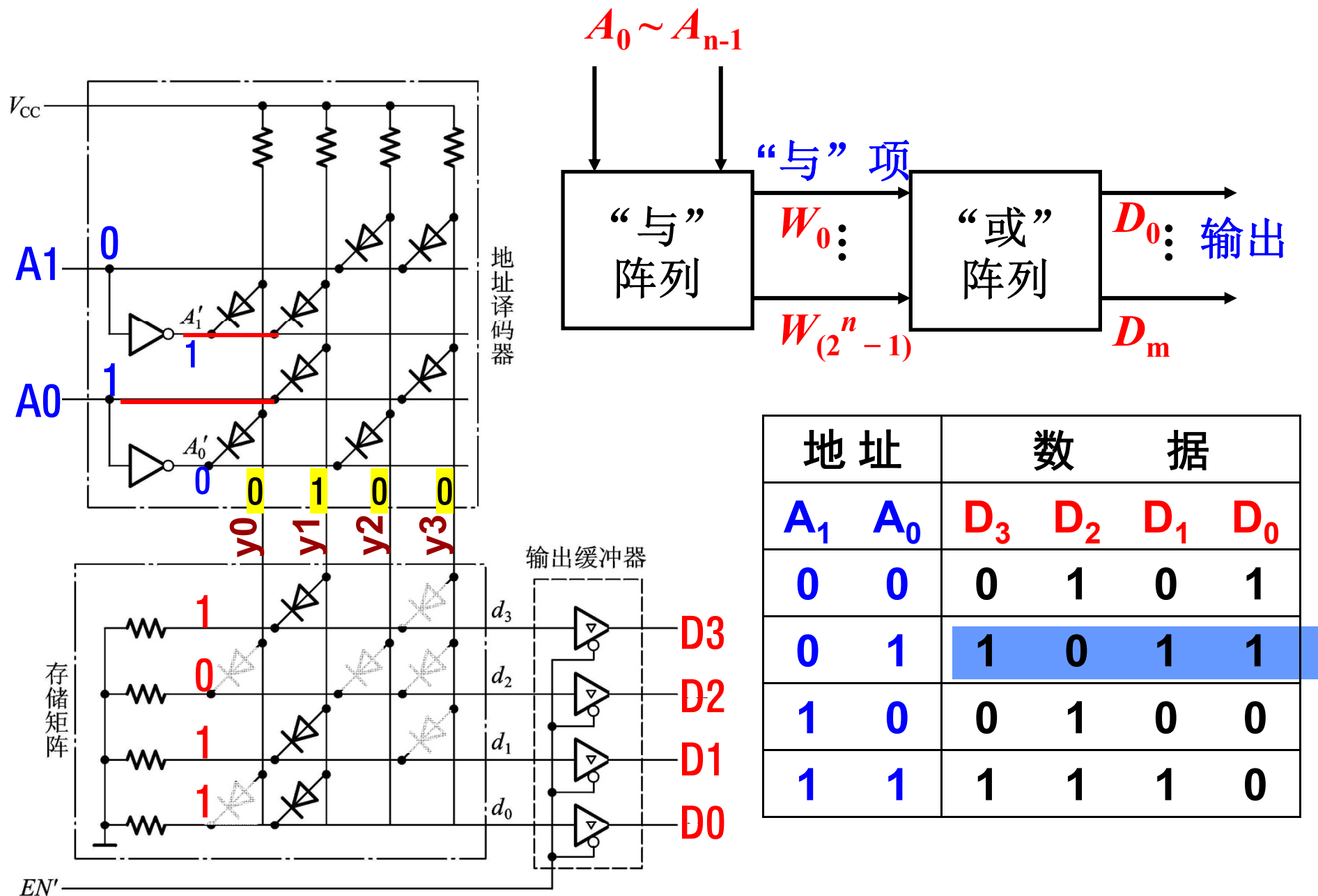
# 1. 二极管ROM电路结构

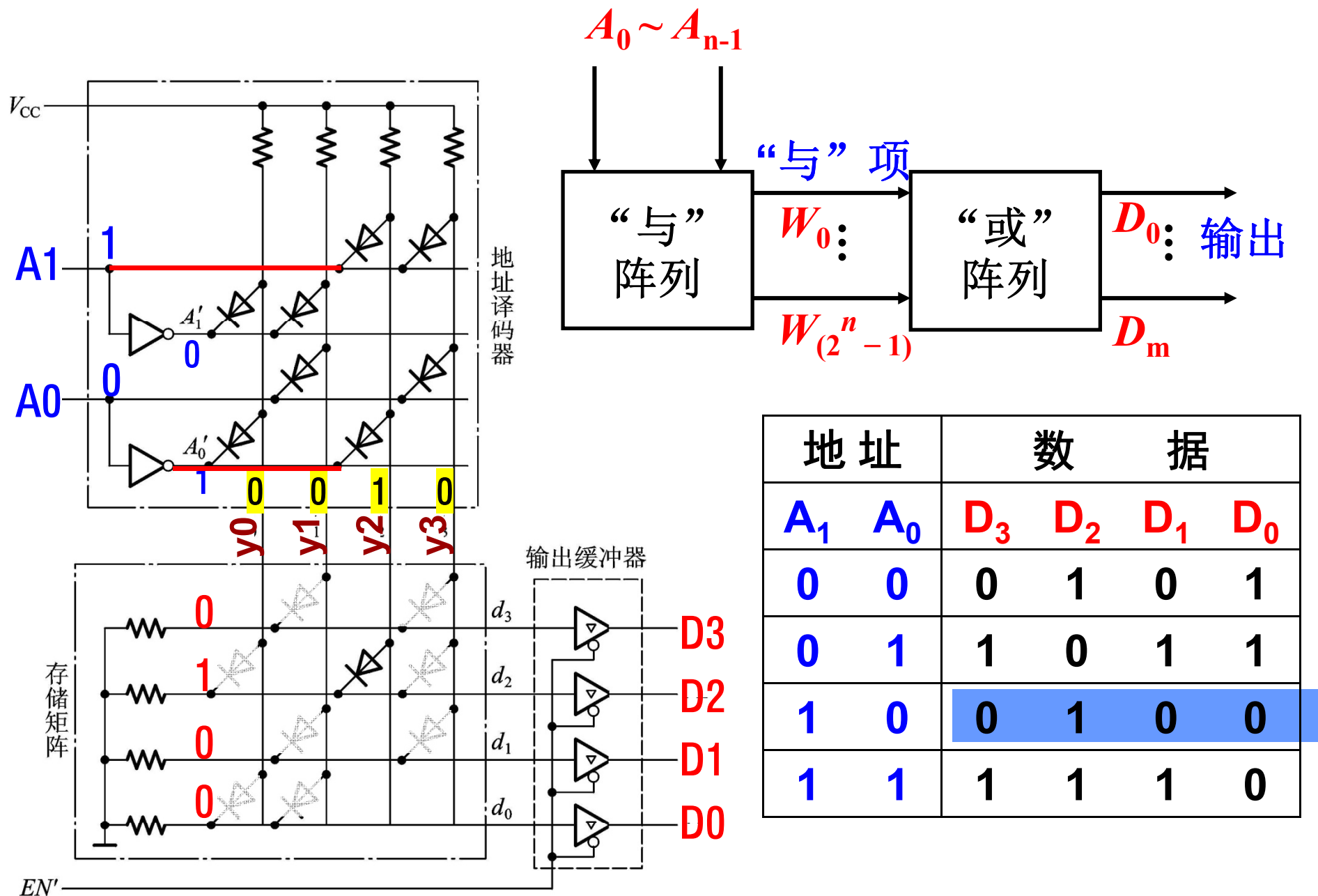


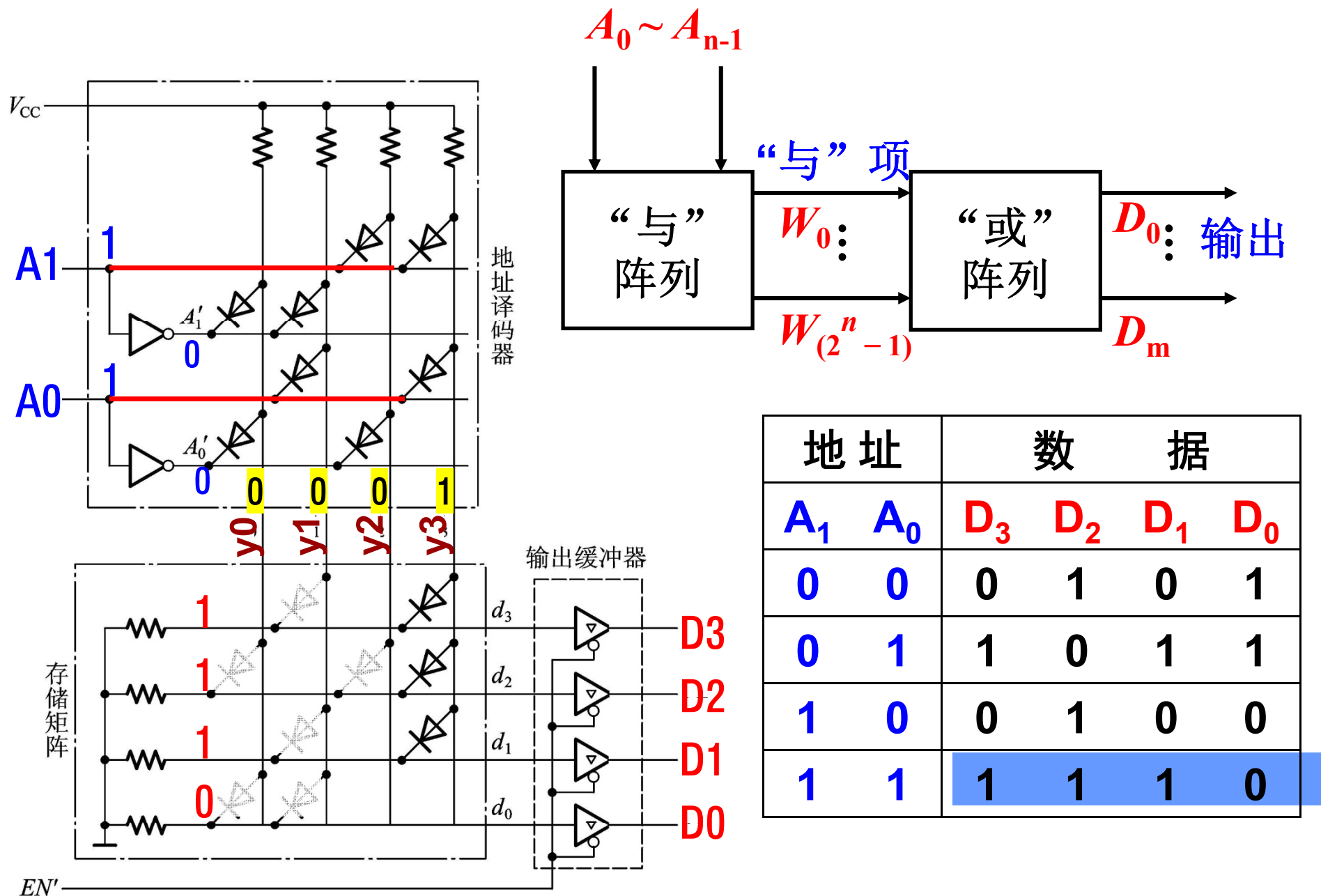
存储矩阵的每个交叉点是一个“存储单元”，存储单元中有器件存入“1”，无器件存入“0”







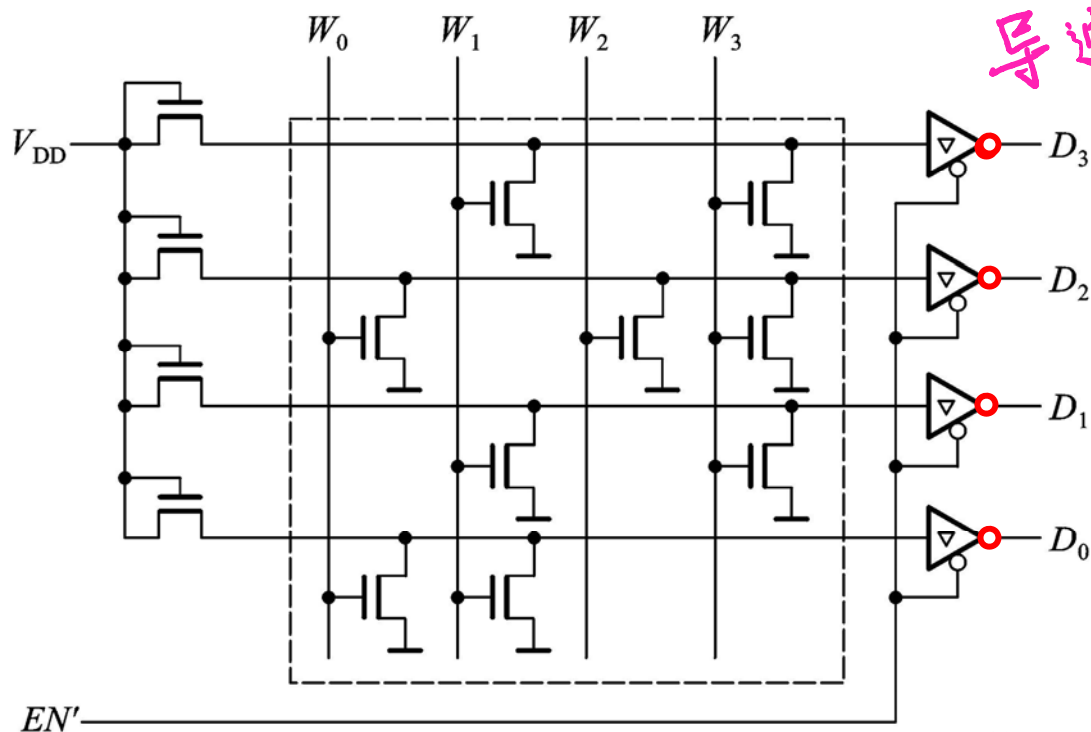




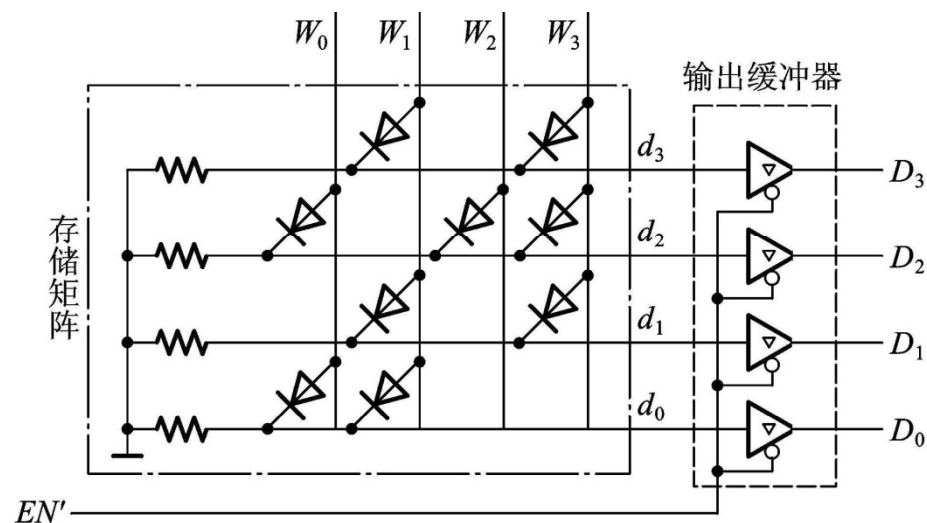


## 2. 用MOS管构成的存储矩阵

导通后是低电平



存储矩阵的每个交叉点是一个“存储单元”，存储单元中有器件存入“1”，无器件存入“0”

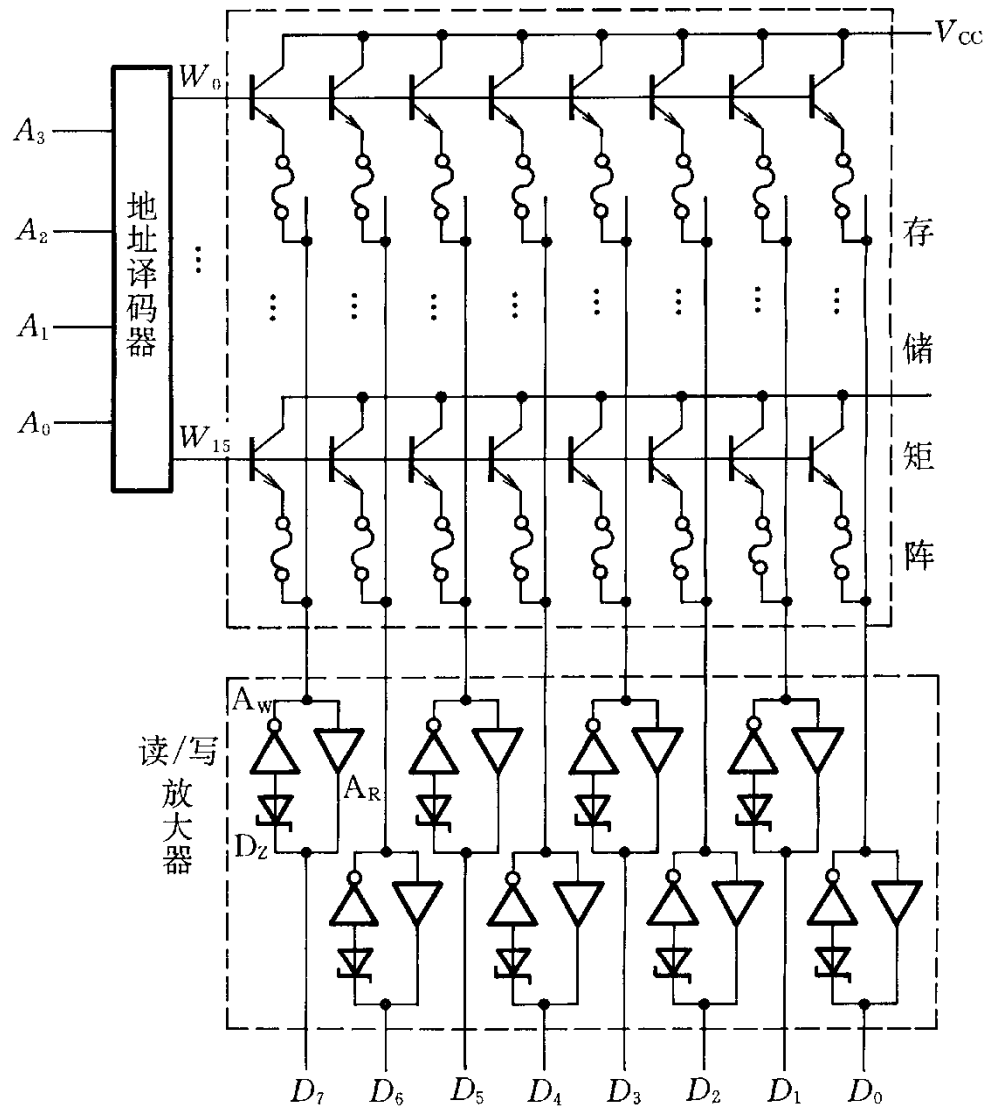


## 二、 可编程ROM ( PROM ) (Programmable ROM)

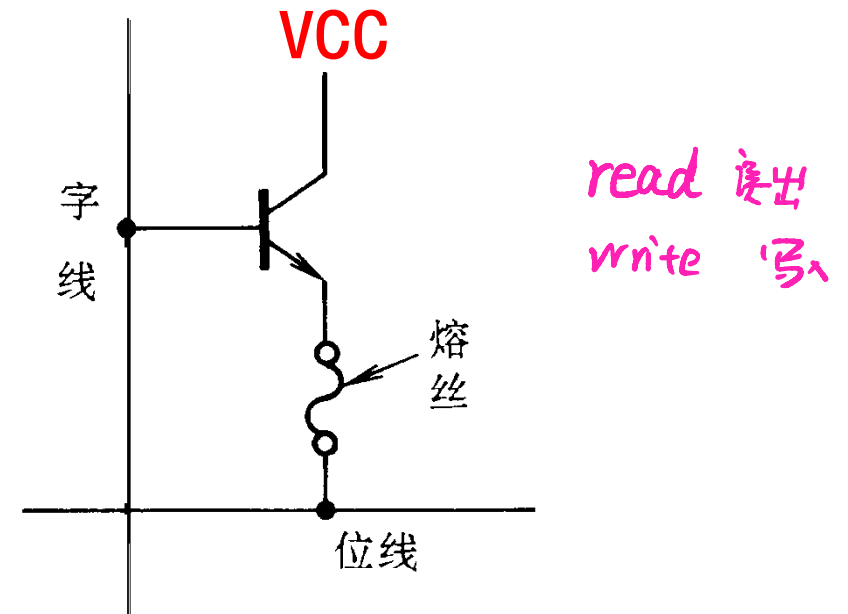
PROM在出厂时，存储的内容为全 0(或全 1)，用户根据需要，可将某些单元改写为 1(或 0)。这种ROM采用熔丝或PN结击穿的方法编程，由于熔丝烧断或PN结击穿后不能再恢复， **因此PROM只能**改写一次。

# PROM总体结构与掩膜ROM一样，但存储单元不同

## PROM的结构图



## 熔丝型PROM的存储单元

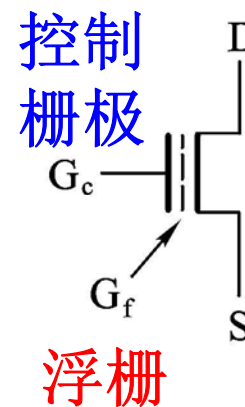
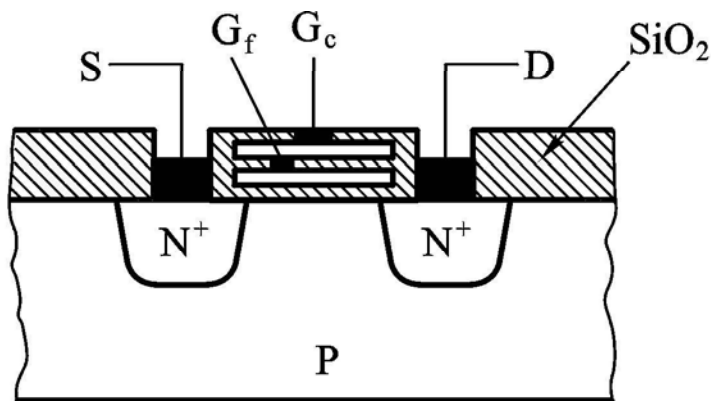


- \* 熔丝由易熔合金制成
  - \* 出厂时，每个结点上都有
  - \* 编程时将不用的熔断
- !! 是一次性编程，不能改写

### 三、可擦除可编程ROM(EPROM) (Erasable Programmable ROM)

EPROM中存储的数据可以擦除重写，因而用在经常需要改写ROM中内容的场合。

#### 1. 用紫外线擦除的EPROM (UVEPROM)

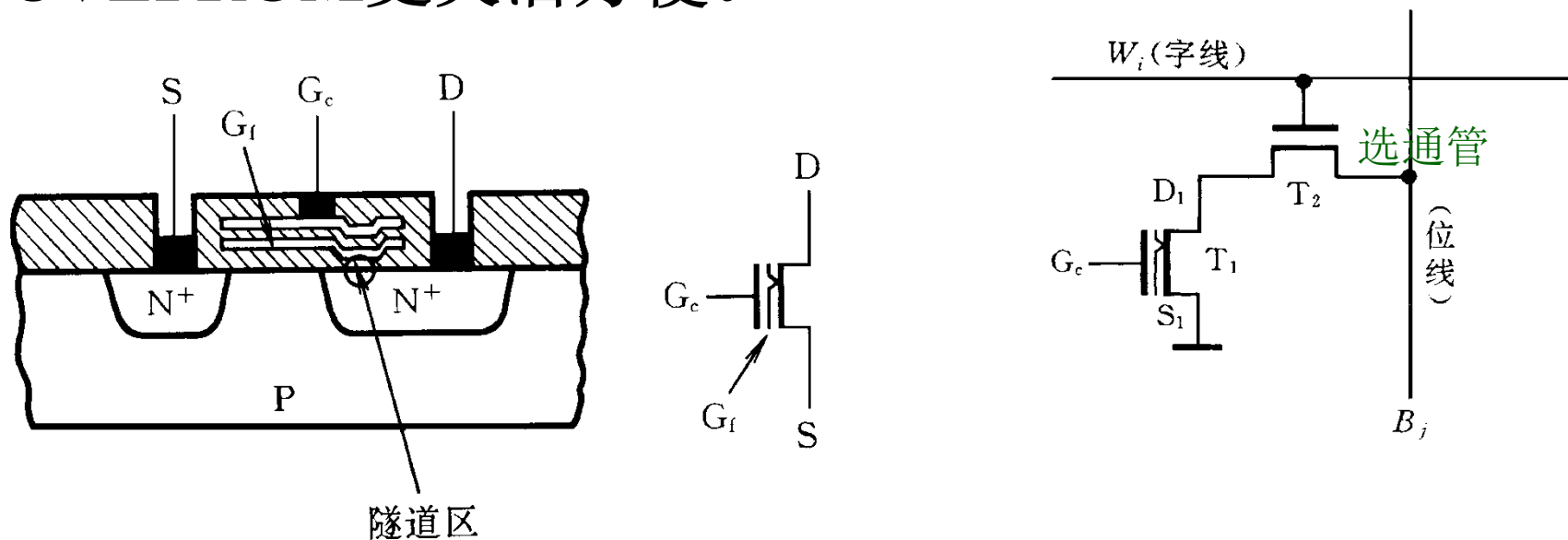


DS之间,Gc端加高压(约25V), 给浮栅注入电荷, 相当于写入了1,  
去掉高压后, 浮栅电荷没有放电回路, 电荷可保存10年以上

## 2. 电可擦除的可编程ROM (E<sup>2</sup>PROM)

(Electrically Erasable Programmable ROM)

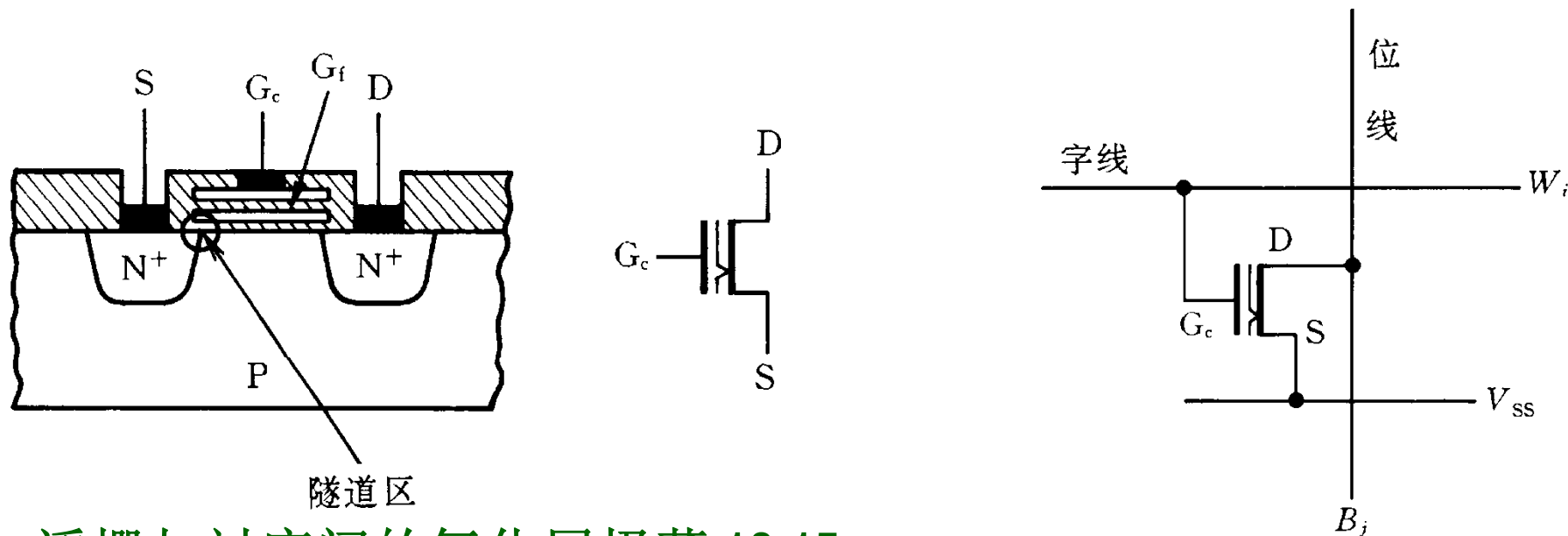
E<sup>2</sup>PROM 只需在高电压脉冲或在工作电压下就可以进行擦除，而不要借助紫外线照射，所以比 UVEPROM 更灵活方便。



浮栅隧道氧化层MOS管，浮栅和漏区之间极薄氧化层，隧道区强电场下，浮栅和漏区间出现隧道，电子可双向通过。擦除和写入都需高电压，需要编程器。

## 四、快闪存储器（Flash Memory）

快闪存储器即吸收了**EPROM**结构简单、编程可靠的优点，又保留了**E<sup>2</sup>PROM**用隧道效应擦除的快捷特性，而且集成度可以作得很高。



浮栅与衬底间的氧化层极薄 10-15nm

擦除和写入都不需要高电压，不需要编程器。

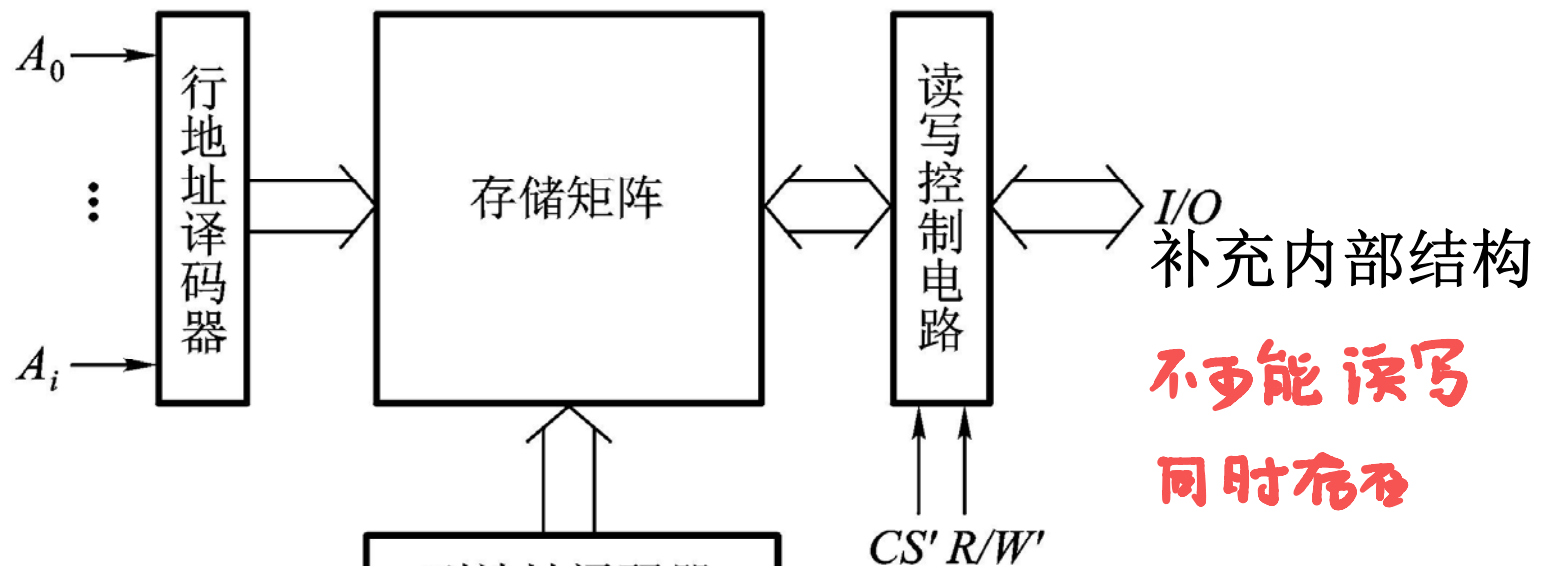
编程次数有限，10000~100000次



SRAM	DRAM
Fast 速度快	Relatively slow speed, RC delay 相对慢
less dense 密度低	Very dense 密度高
more expensive 价格贵	Low cost per bit 价格低
Simple memory cell structure 结构简单	More complex (read and refresh periodically) 需要刷新电路
用途: CPU芯片内一级缓存	内存条

## 5.5.1 随机存储器RAM (Random Access Memory) CPU 内存条

### 静态随机存储器 (SRAM) 的结构与工作原理



不可能读写  
同时存在

地址译码器:

行地址译码选出一行

列地址译码选出一列(1位或几位)

读写控制  
 $RW'=1$ , 允许读  
 $RW'=0$ , 允许写

片选控制  
 $CS'=0$ , 允许读写



# 1024×4位 SRAM的结构图

存储矩阵

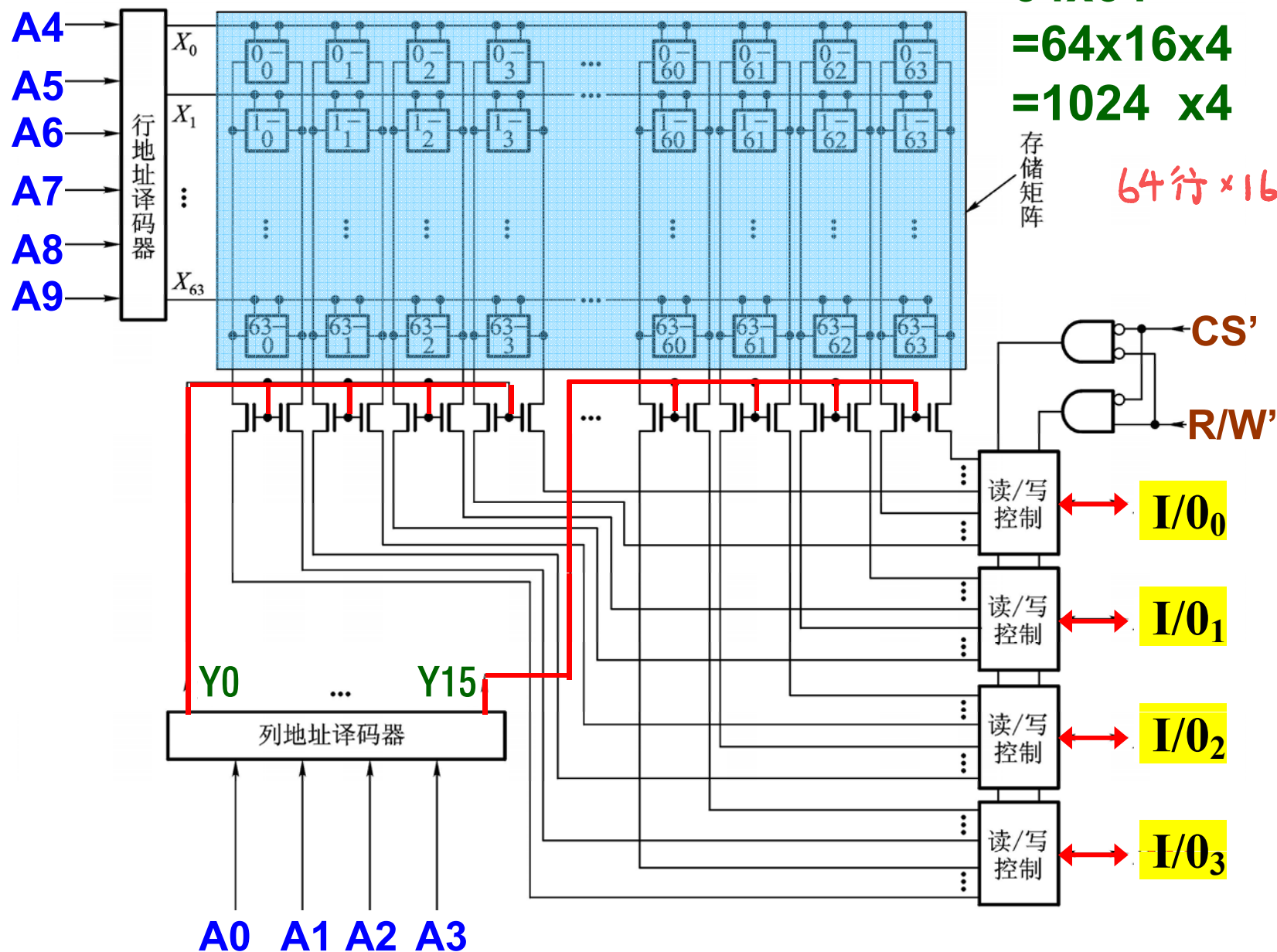
64x64

=64x16x4

=1024 x4

64行×16列

存储矩阵



2<sup>n</sup>.m

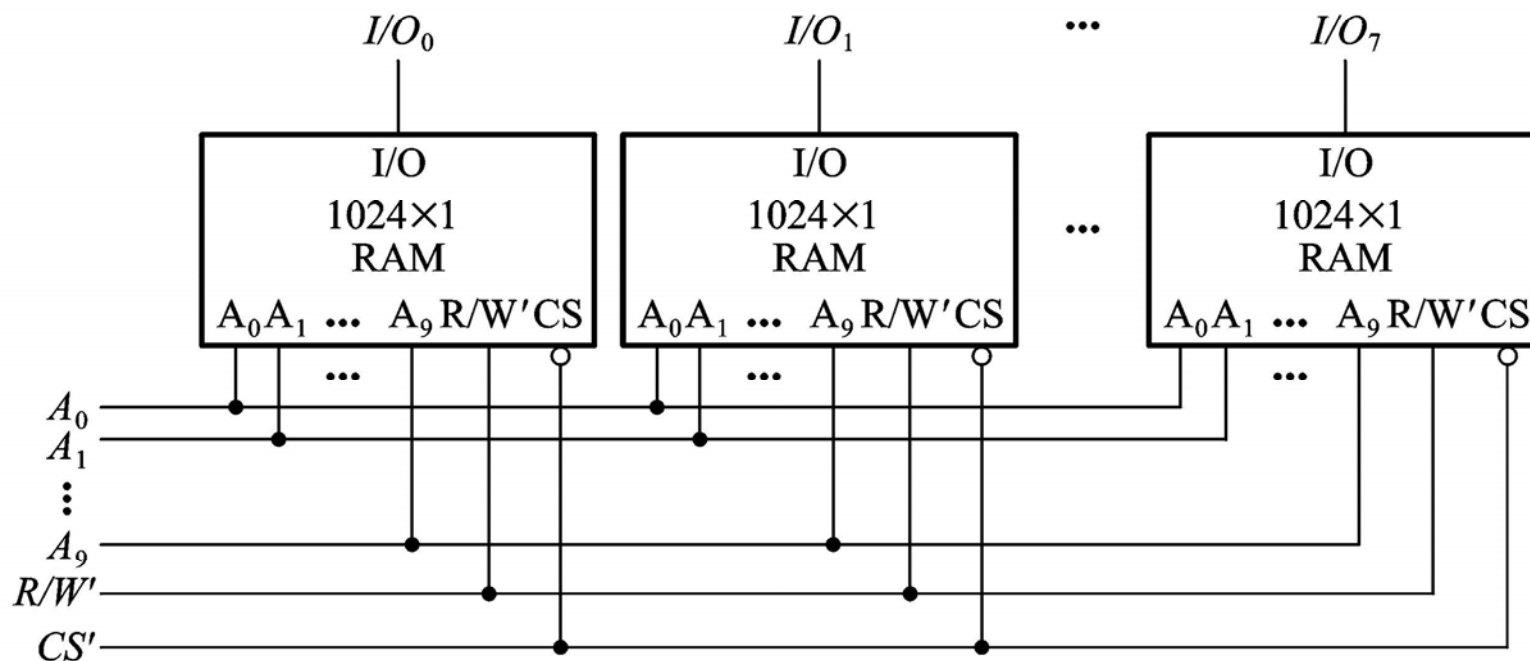
## 5.5.4 存储器容量的扩展

位数  $m$   
字数  $n$

### 一、位扩展方式

适用于每片RAM、ROM字数够用而位数不够时  
接法：将各片的地址线、读写线、片选线并联即可

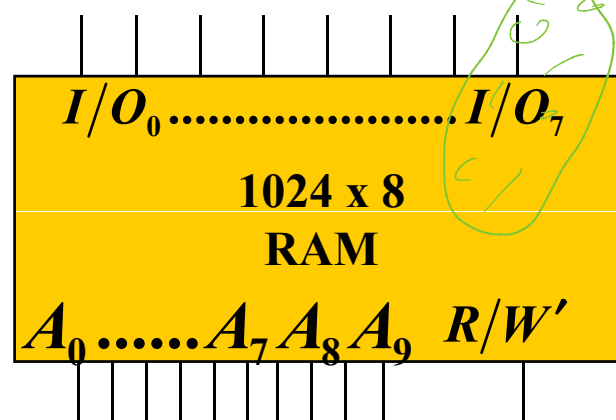
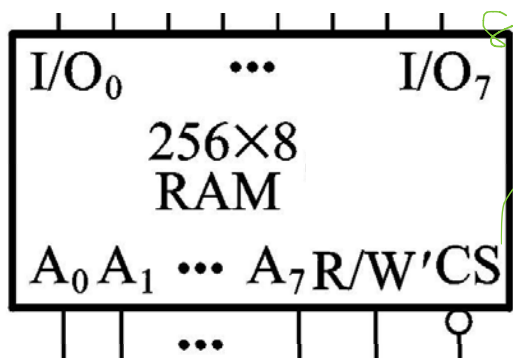
例1：用八片1024 x 1位 → 1024 x 8位的RAM



## 二、字扩展方式

适用于每片RAM、ROM位数够用而字数不够时

例2: 用四片  $256 \times 8$  位  $\rightarrow$   $1024 \times 8$  位 RAM



数据线:  $I/O_0 \sim I/O_7$

地址线:  $A_0 \sim A_7$

读/写信号:  $R/W'$

片选信号:  $CS'$

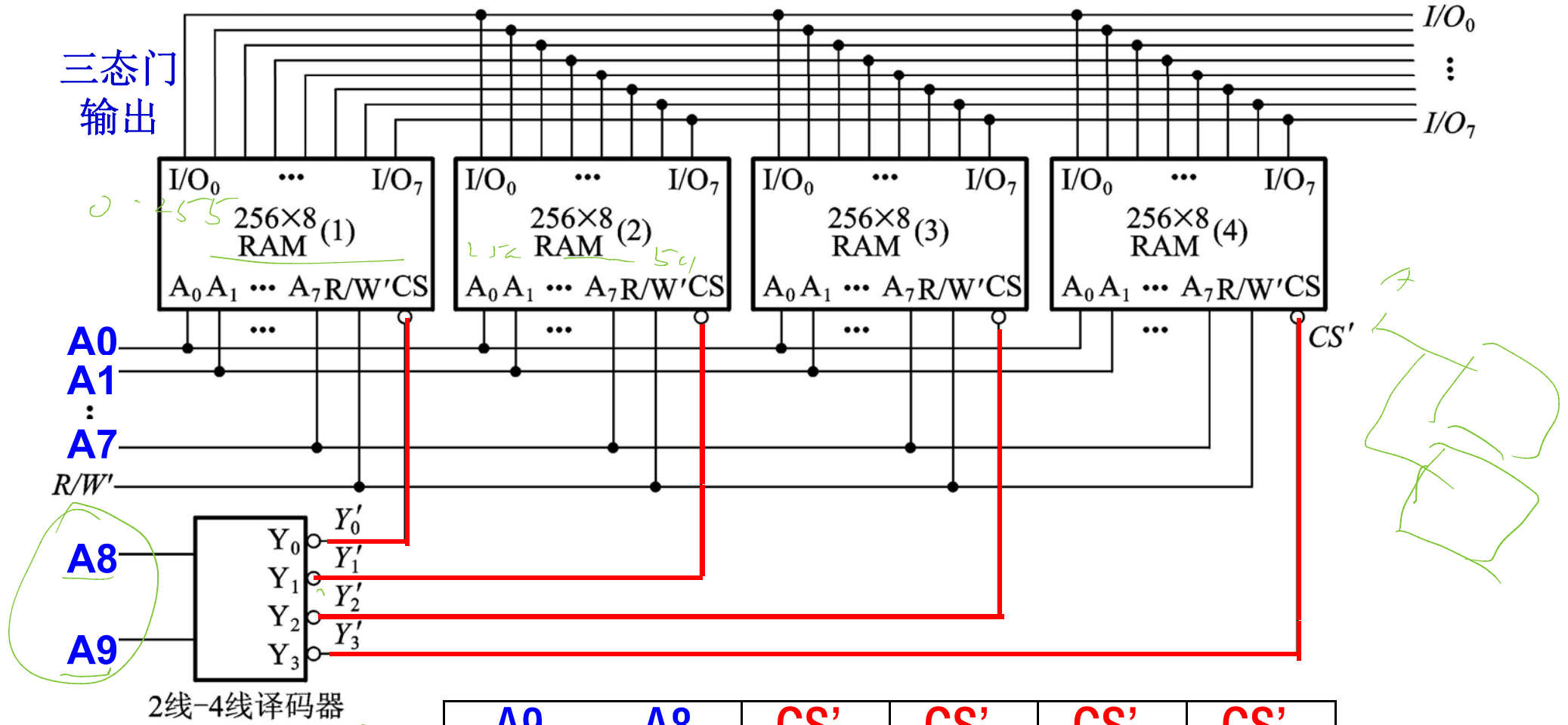
数据线:  $I/O_0 \sim I/O_7$

地址线:  $A_0 \sim A_7, A_8, A_9$

读/写信号:  $R/W'$

片选信号:  $CS'$

例2：用四片 $256 \times 8$ 位 $\rightarrow 1024 \times 8$ 位 RAM



$A_9$	$A_8$	$CS'_1$	$CS'_2$	$CS'_3$	$CS'_4$
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

000 - 0FF

100 - 1FF

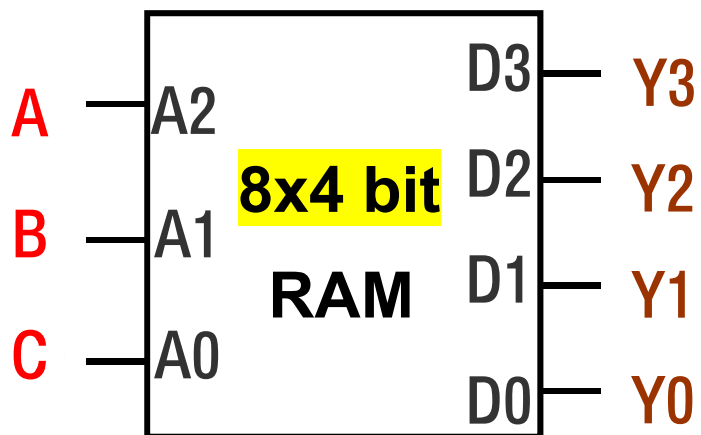
200 - 2FF

300 - 3FF

## 5.5.5 用存储器实现组合逻辑函数



一、基本原理：地址线 $\leftrightarrow$ 输入变量，数据线 $\leftrightarrow$ 组合逻辑输出



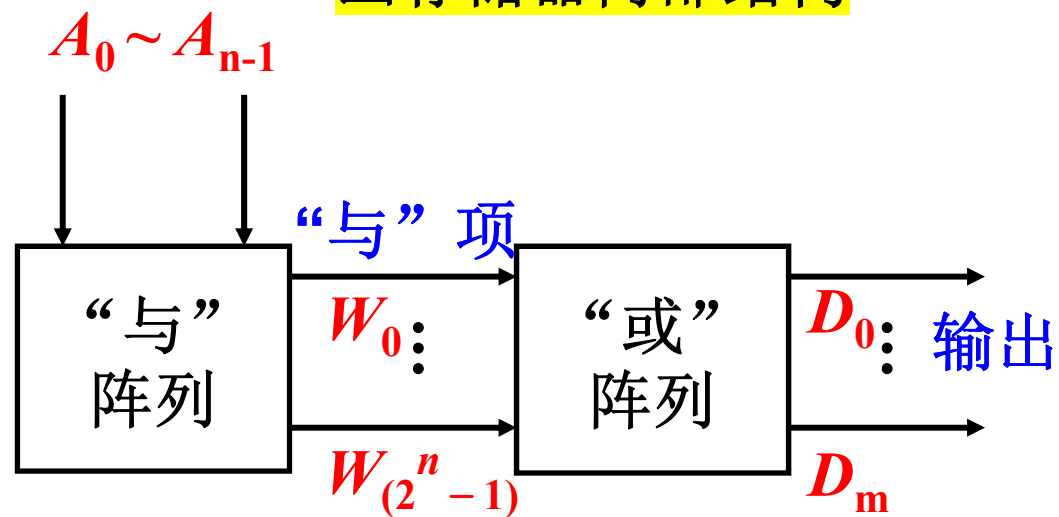
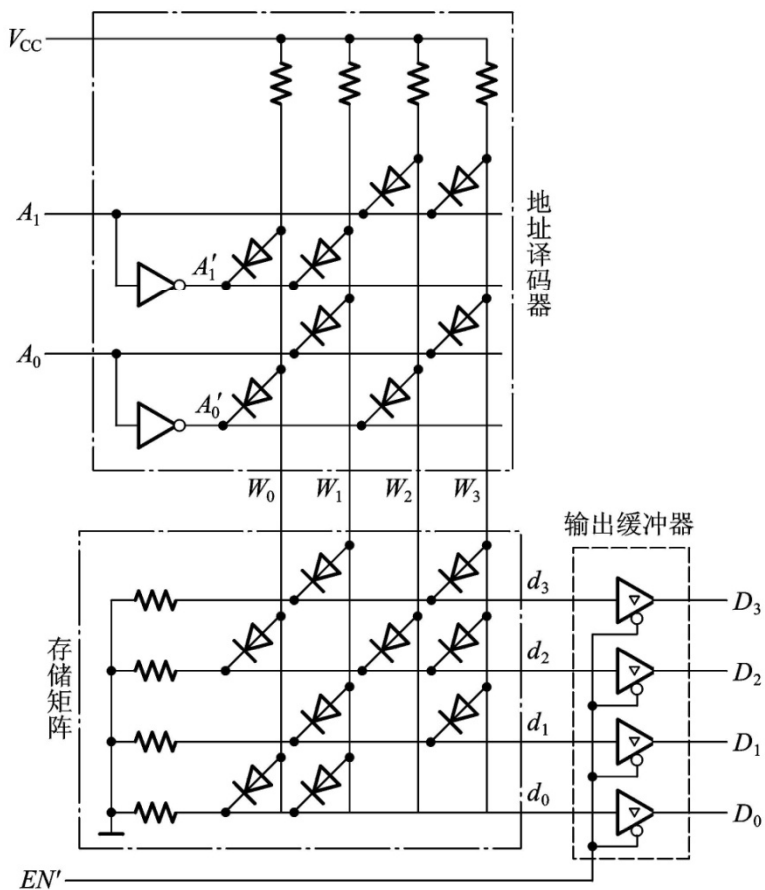
A2	A1	A0	D3	D2	D1	D0
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				



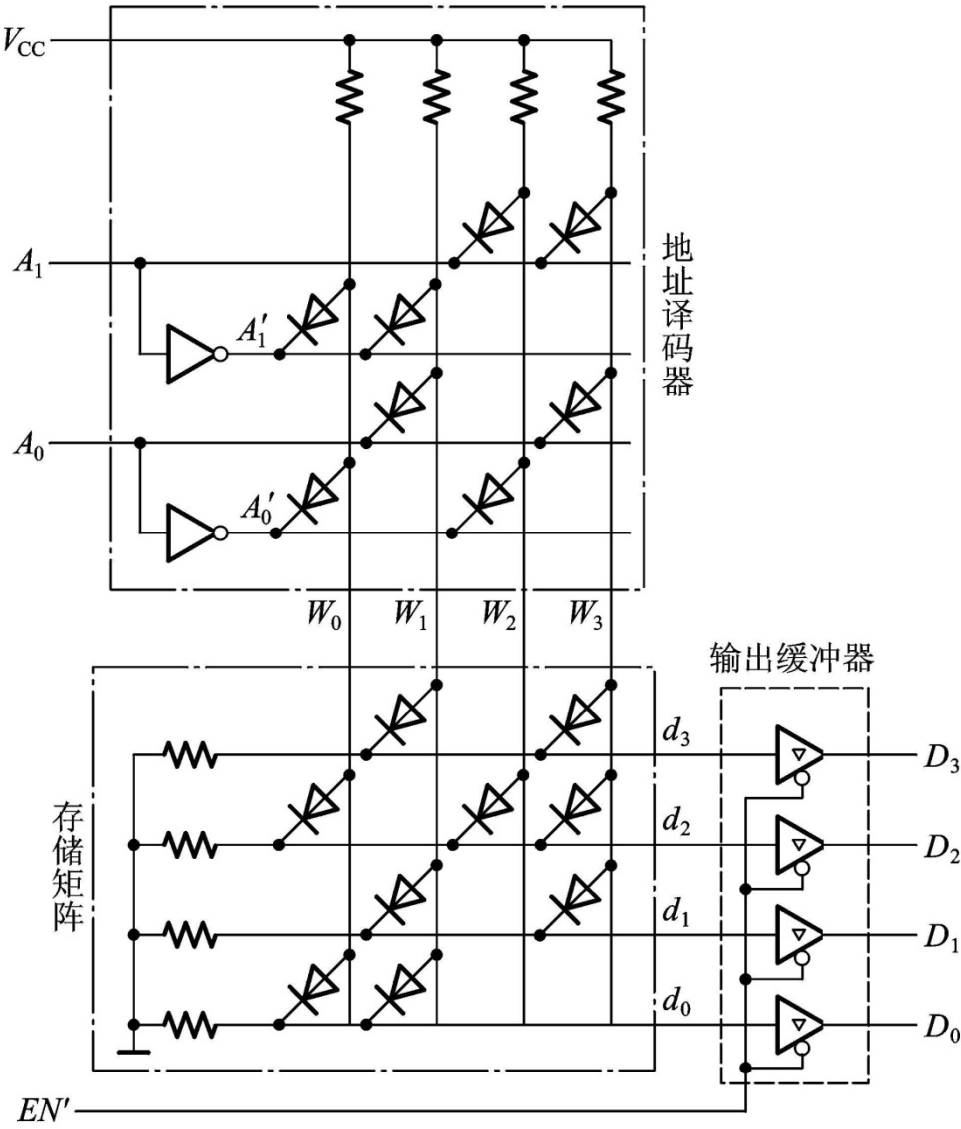
A	B	C	Y3	Y2	Y1	Y0
0	0	0	1	0	0	0
0	0	1	1	1	0	0
0	1	0	1	1	1	0
0	1	1	1	1	1	1
1	0	0	0	1	1	1
1	0	1	0	0	1	1
1	1	0	0	0	0	1
1	1	1	0	0	0	0

## 5.5.5 用存储器实现组合逻辑函数

### 画存储器内部结构



# 5.5.5 用存储器实现组合逻辑函数

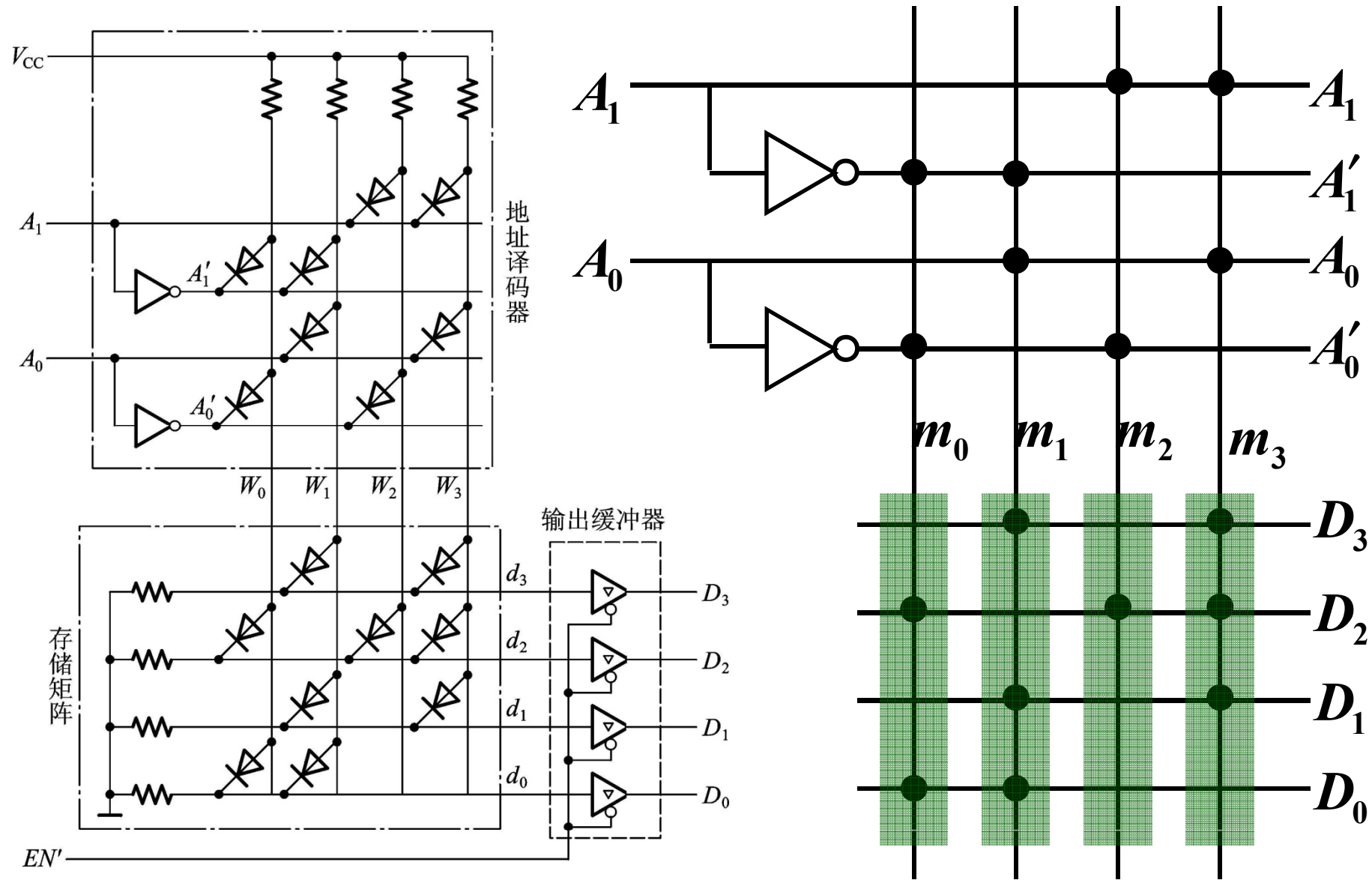


简化画存储器内部结构

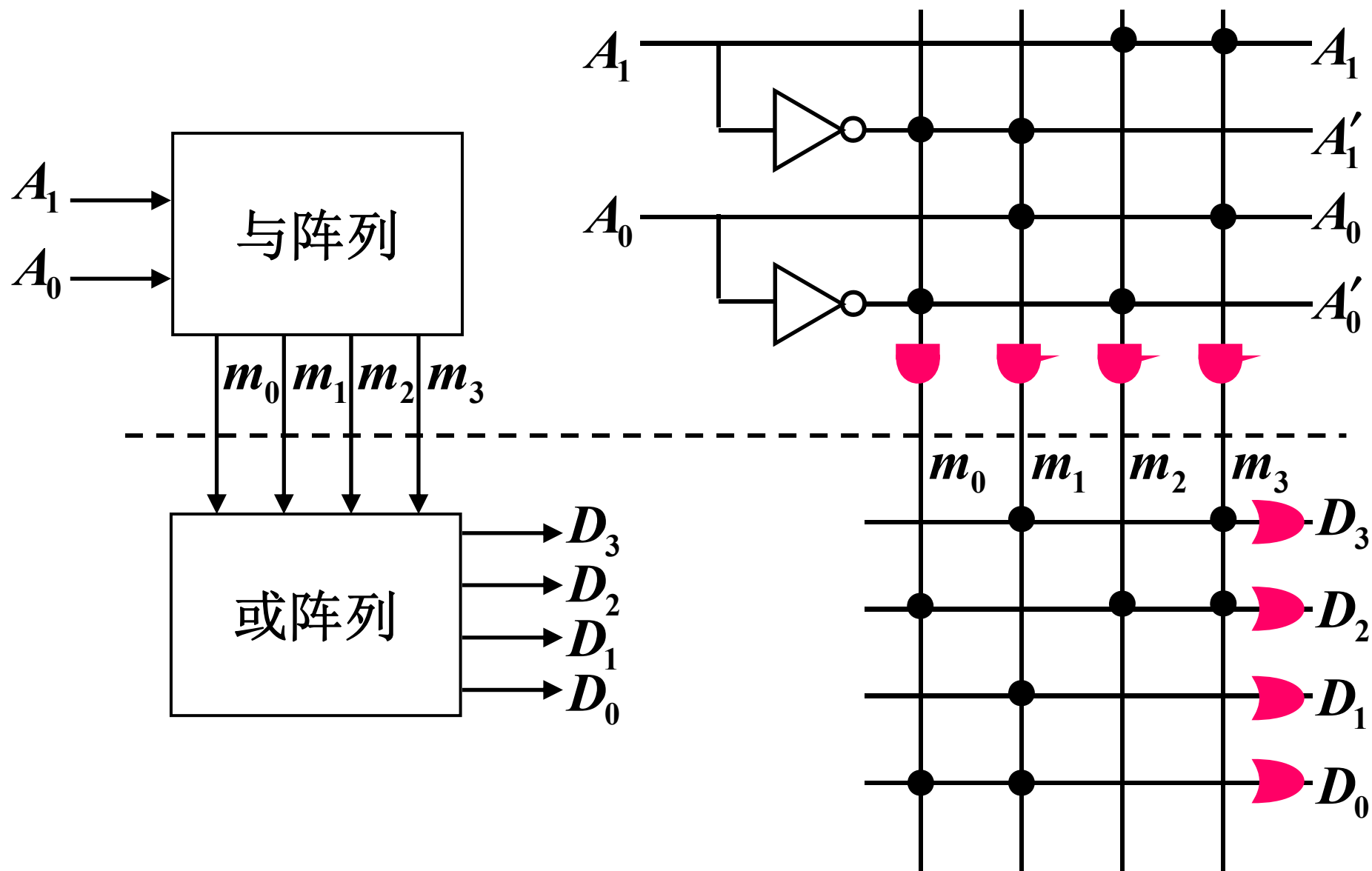
地 址		数 据			
$A_1$	$A_0$	$D_3$	$D_2$	$D_1$	$D_0$
0	0	0	1	0	1
0	1	1	0	1	1
1	0	0	1	0	1
1	1	1	1	0	0

5.5.5 用存储器实现组合逻辑函数

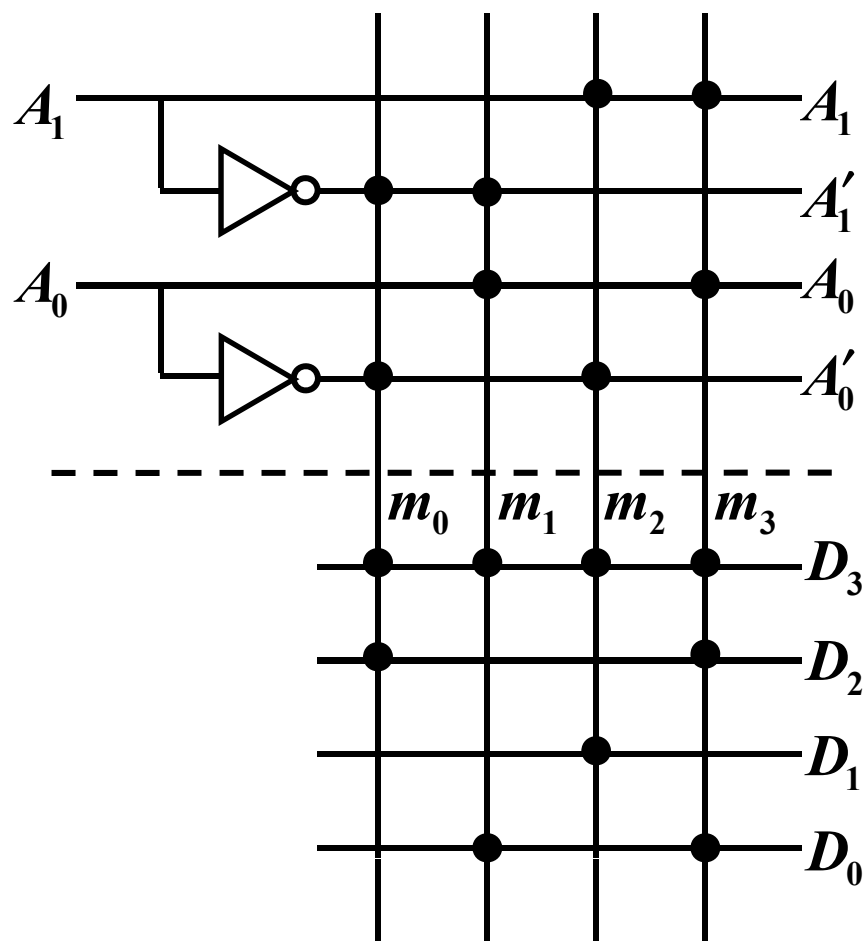
简化画存储器内部结构







ROM的与阵列和或阵列图



已知RAM结构如左图

若将输入地址 $A_1$ 、 $A_0$ 视为输入变量，而将 $D_3$ 、 $D_2$ 、 $D_1$ 、 $D_0$ 视为一组输出变量，则 $D_3$ 、 $D_2$ 、 $D_1$ 、 $D_0$ 就是 $A_1$ 、 $A_0$ 的一组逻辑函数。

$$D_3 = A_1' A_0' + A_1' A_0 + A_1 A_1' + A_1 A_0 = m_0 + m_1 + m_2 + m_3$$

$$D_2 = A_1' A_0' + A_1 A_0 = m_0 + m_3$$

$$D_1 = A_1' A_0 = m_2$$

$$D_0 = A_1' A_0 + A_1 A_0 = m_1 + m_3$$

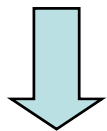
## **用ROM实现逻辑函数一般按以下步骤进行：**

- (1) 根据逻辑函数的输入、输出变量数，确定ROM容量，选择合适的ROM。**
- (2) 写出逻辑函数的最小项表达式，画出ROM阵列图。**
- (3) 根据阵列图对ROM进行编程。**

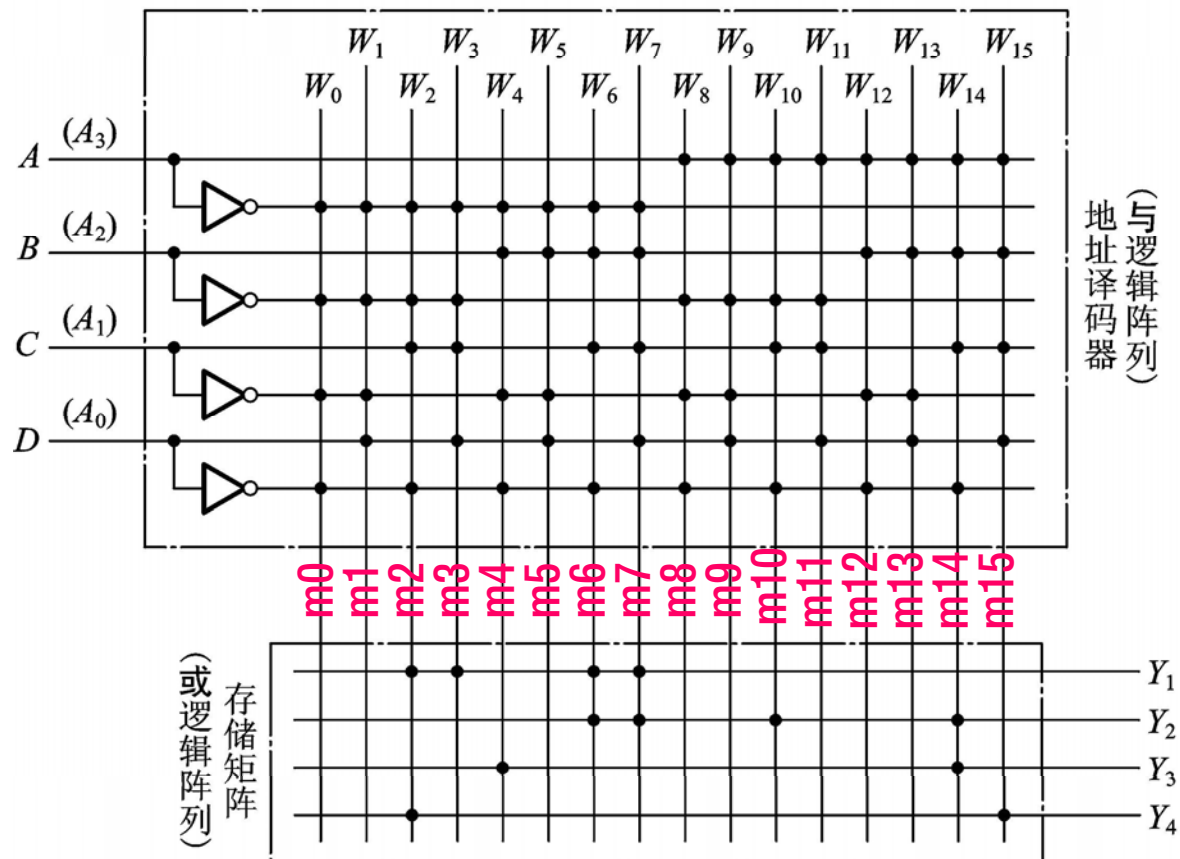
## 例2 用ROM产生以下组合逻辑函数

$$\begin{cases}
 Y_1 = A'BC + A'B'C = A'BC(\underline{D+D'}) + A'B'C(\underline{D+D'}) = \underline{A'BCD} + \underline{A'BCD'} + \underline{A'B'CD} + \underline{A'B'CD'} \\
 Y_2 = AB'CD' + BCD' + A'BCD = AB'CD' + \underline{ABCD'} + \underline{A'BCD'} + A'BCD \\
 Y_3 = ABCD' + A'BC'D' \\
 Y_4 = A'B'CD' + ABCD
 \end{cases}$$

m7    m6    m3    m2  
m10    m14    m6    m7

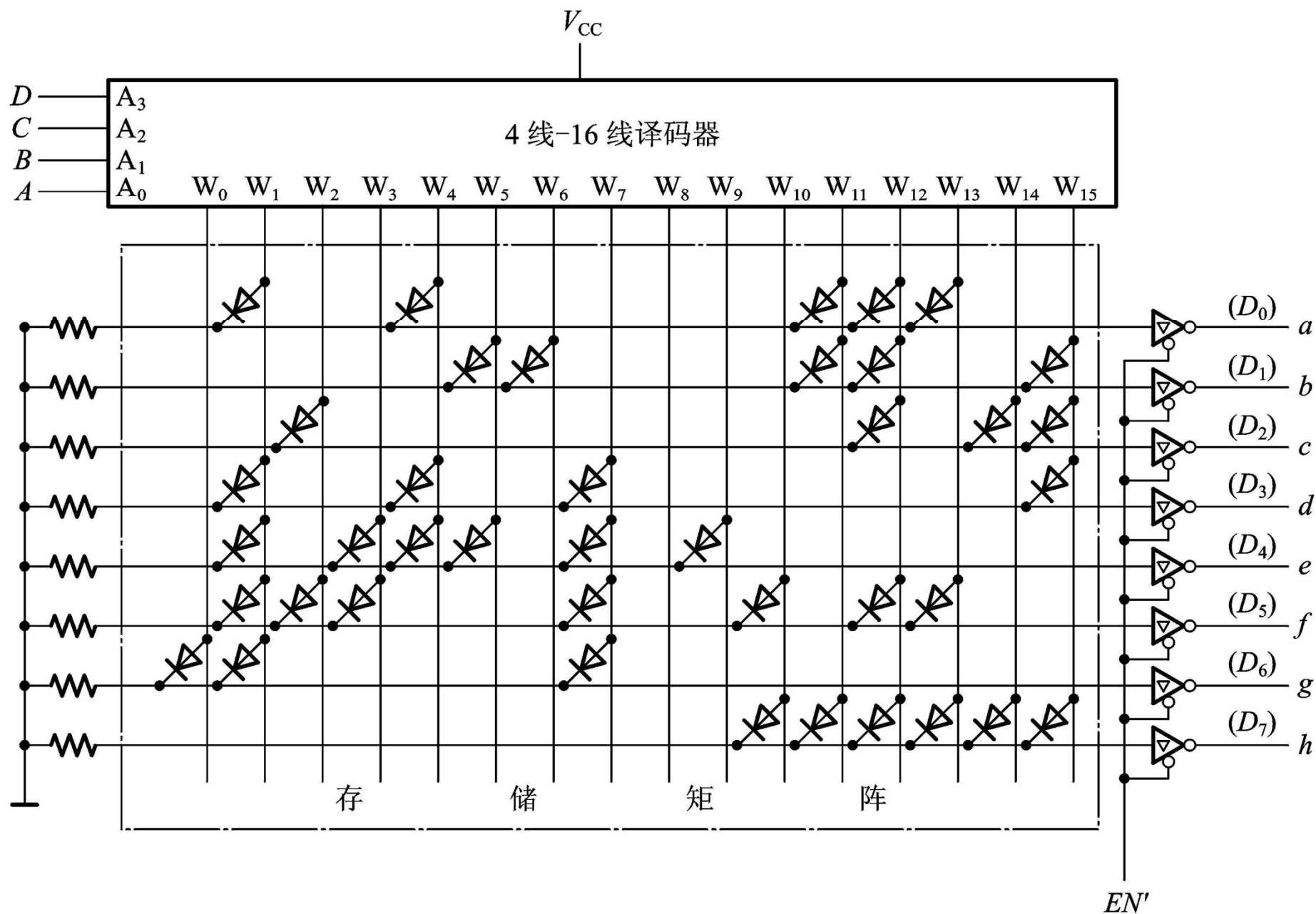


$$\begin{cases}
 Y_1 = \sum m(2, 3, 6, 7) \\
 Y_2 = \sum m(6, 7, 10, 14) \\
 Y_3 = \sum m(4, 14) \\
 Y_4 = \sum m(2, 15)
 \end{cases}$$



【例】用ROM设计一个八段字符显示的译码器

输入				输出								字形
<i>D</i>	<i>C</i>	<i>B</i>	<i>A</i>	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>	
0	0	0	0	1	1	1	1	1	1	0	1	0 . - . - . - . - . - . - . - . - . - .
0	0	0	1	0	1	1	0	0	0	0	1	
0	0	1	0	1	1	0	1	1	0	1	1	
0	0	1	1	1	1	1	1	0	0	1	1	
0	1	0	0	0	1	1	0	0	1	1	1	
0	1	0	1	1	0	1	1	0	1	1	1	
0	1	1	0	0	0	1	1	1	1	1	1	
0	1	1	1	1	1	1	0	0	0	0	1	
1	0	0	0	1	1	1	1	1	1	1	1	
1	0	0	1	1	1	1	0	0	1	1	1	
1	0	1	0	1	1	1	1	1	0	1	0	1 0 - 0 - 0 - 0 - 0 - 0
1	0	1	1	0	0	1	1	1	1	1	0	
1	1	0	0	0	0	0	1	1	0	1	0	
1	1	0	1	0	1	1	1	1	0	1	0	
1	1	1	0	1	1	0	1	1	1	1	0	
1	1	1	1	1	0	0	0	1	1	1	0	

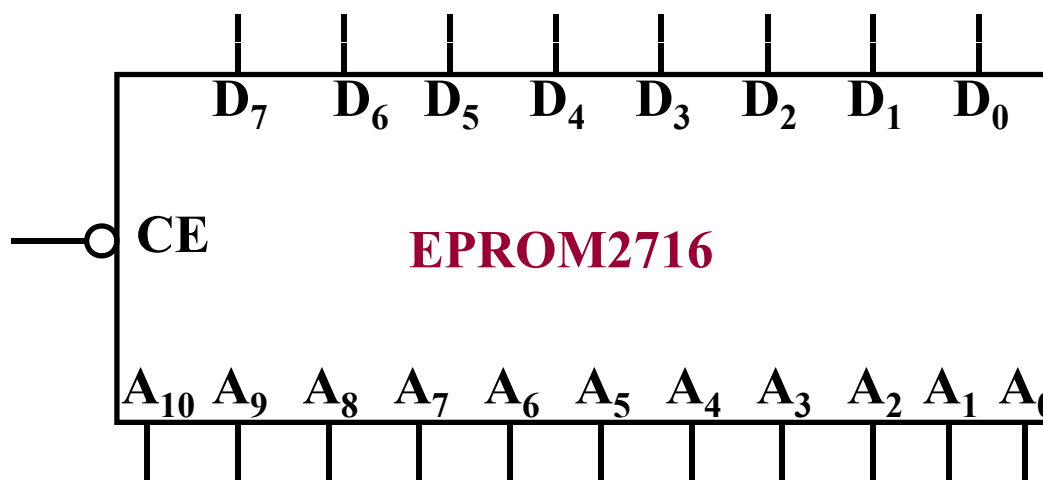


## 练习：

1、指出下列存储系统各具有多少个存储单元，至少需要几根地址线 and 数据线。

(1)  $64K \times 1$       (2)  $256K \times 4$       (3)  $1M \times 1$

2、下图为EPROM2716,其中 $A_{10} \sim A_0$ 为地址线,  $D_7 \sim D_0$ 为数据线, CE为使能端, 现在需要容量为 $16K \times 32$ 的存储器, 问: 需要EPROM2716多少片? 说明理由。



$$2^{11} \times 8$$

$$\frac{32}{8} \times \frac{2^{14}}{2^{11}} = 32$$

## 作 业

**5.27、 5.28、 5.32、 5.35、 5.37、 5.40**