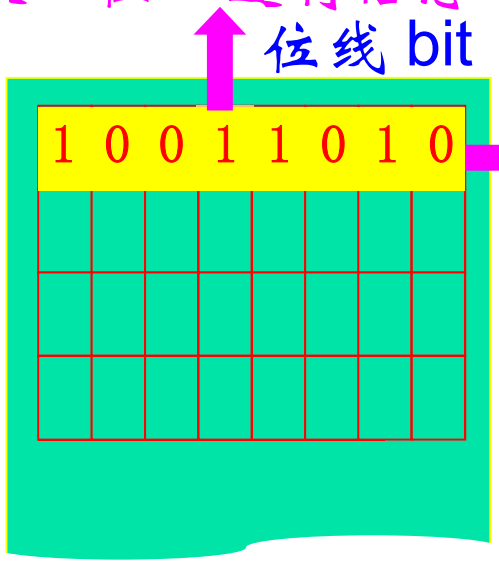
第七章 半导体存储器

半导体存储器是存放大量的二值信息的部件。用以存储不同程序的操作指令，或待计算处理的数据等，是数字系统不可缺少的组成部分。

半导体存储器的结构:

基本存储单元cell：
存放一位二进制信息



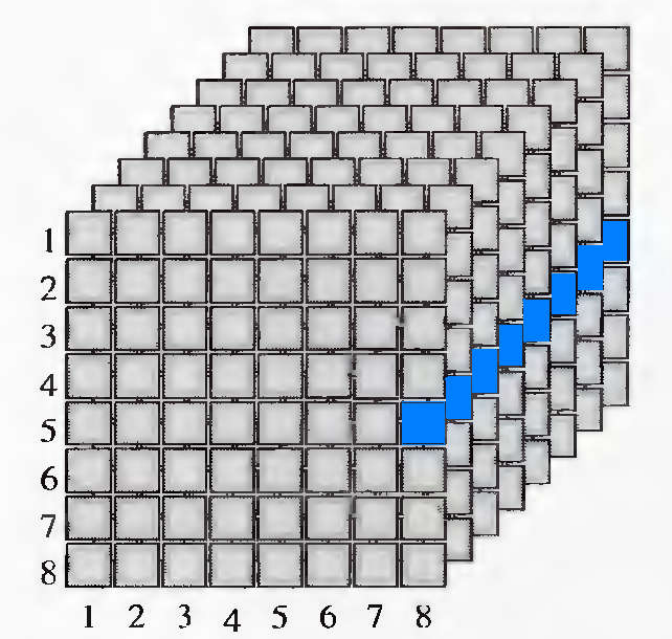
4-bit = nibble
8-bit = byte

★ 存储体
存储容量

capacity = rows X columns

2D memory array

3D memory array



The address of blue byte is row 5, column 8

capacity＝字数 (words) X 位数 (bits)
＝字数 X 字长

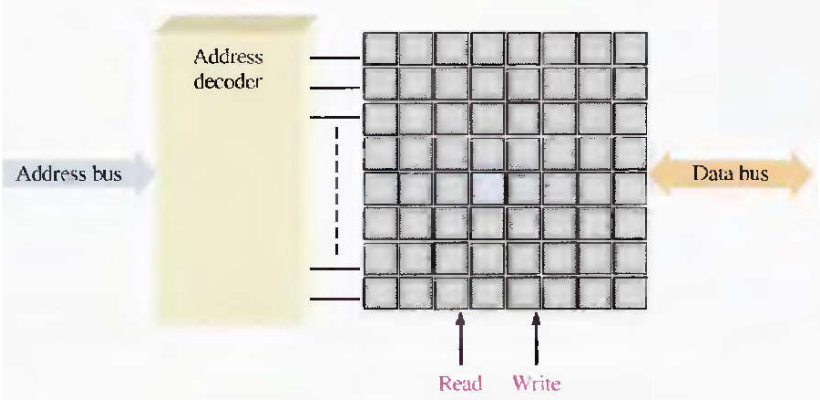
1K×4 ＝ 1024×4　　1K×8
1M ＝1024K　　1G ＝1024M

★ 存储体

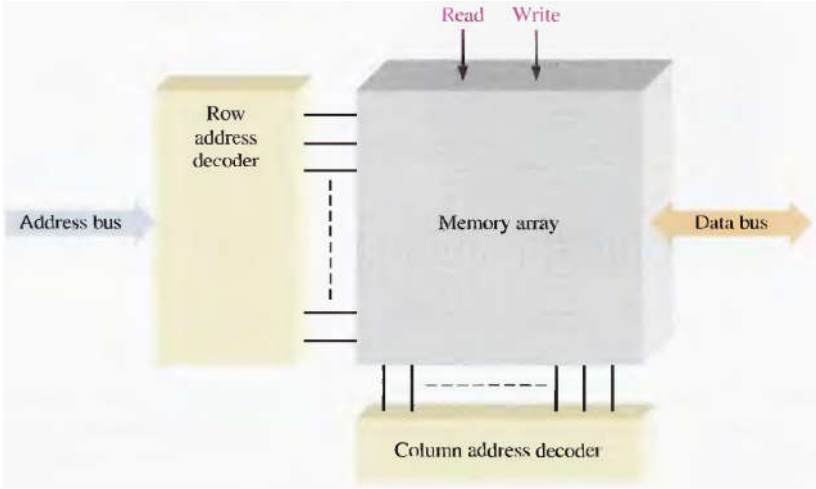
★ 地址译码器

Addressing operation

2 dimensional memory array



3 dimensional memory array



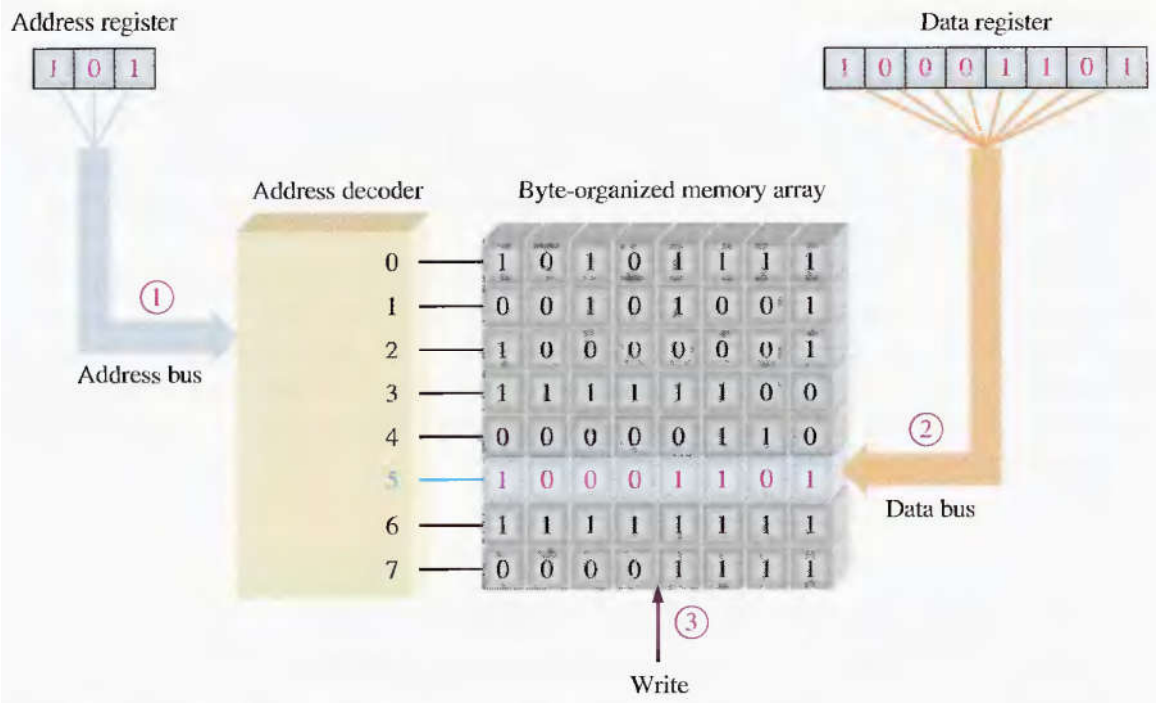
★ 存储体 **memory array**

★ 地址译码器 **address decoder**

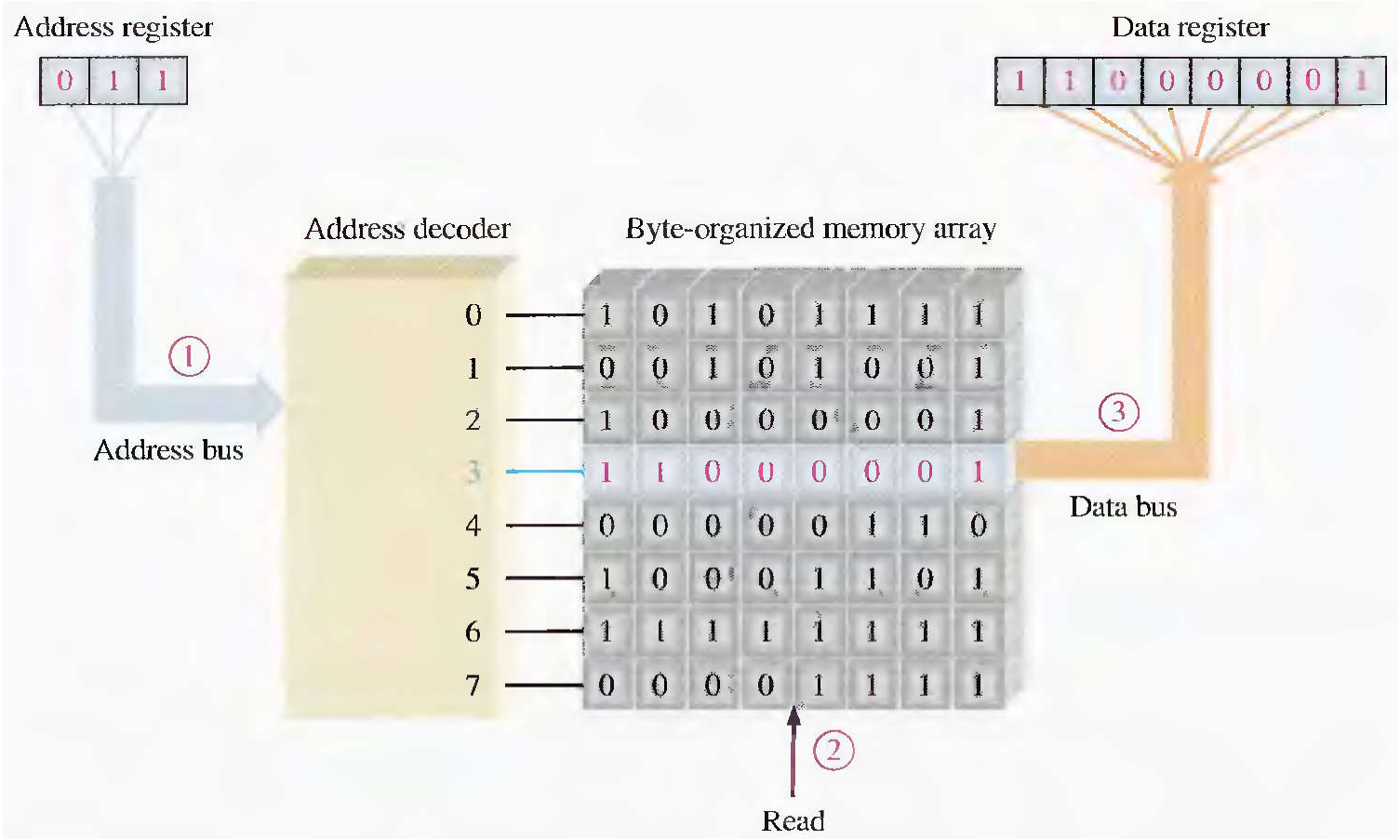
★ 输入、输出电路 **input and output circuit**

Operations: Read & Write

Write



Read


Address register
0 1 1
Data register
1 1 0 0 0 0 0 1
Address decoder
Byte-organized memory array
0 1 0 1 0 1 1 1 1
1 0 0 1 0 1 0 0 1
2 1 0 0 0 0 0 0 1
3 1 1 0 0 0 0 0 1
4 0 0 0 0 0 1 1 0
5 1 0 0 0 1 1 0 1
6 1 1 1 1 1 1 1 1
7 0 0 0 0 1 1 1 1
1
Address bus
2
Read
3
Data bus

半导体存储器的分类

- 从存取信息方式上分：

★ 只读存储器（**Read Only Memory, ROM**）

只读存储器在正常工作时只能读出信息，而不能写入信息，ROM中的信息是在制造时写入的，可以长期保存，即断电后器件中的信息不会消失，也称为非易失性存储器。

★ 随机存取存储器（**Random Access Memory，RAM)**

RAM在正常工作时可以随时写入或读出信息，但断电后器件中的信息也随之消失，因此称为易失性存储器。

- 从制造工艺上分：

★ 双极型存储器

★ 单极型存储器

第一节 只读存储器（ROM）

ROM:是存储固定信息的存储器，使用时只能读出所存的信息而不能写入数据。

一、ROM分类：

1、固定ROM/掩膜ROM (Mask ROM):

专用ROM,用户将程序代码交给IC生产商，生产商在芯片制造过程中将用户程序代码固化在IC的ROM中，用户在使用过程只能读出不能写入。

适合于大批量生产使用，性价比高。

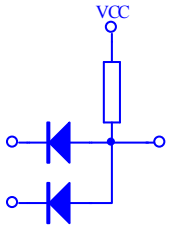
例：如图所示为一存储容量为4×4的固定ROM

- ☆ 存储矩阵　4条字线，4条位线
- ☆ 地址译码器　2位地址码
- ☆ 输出电路

☆ 地址译码器

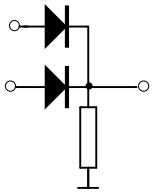
$$W_0 = \overline{A}_1 \overline{A}_0 \quad W_1 = \overline{A}_1 A_0$$
$$W_2 = A_1 \overline{A}_0 \quad W_3 = A_1 A_0$$

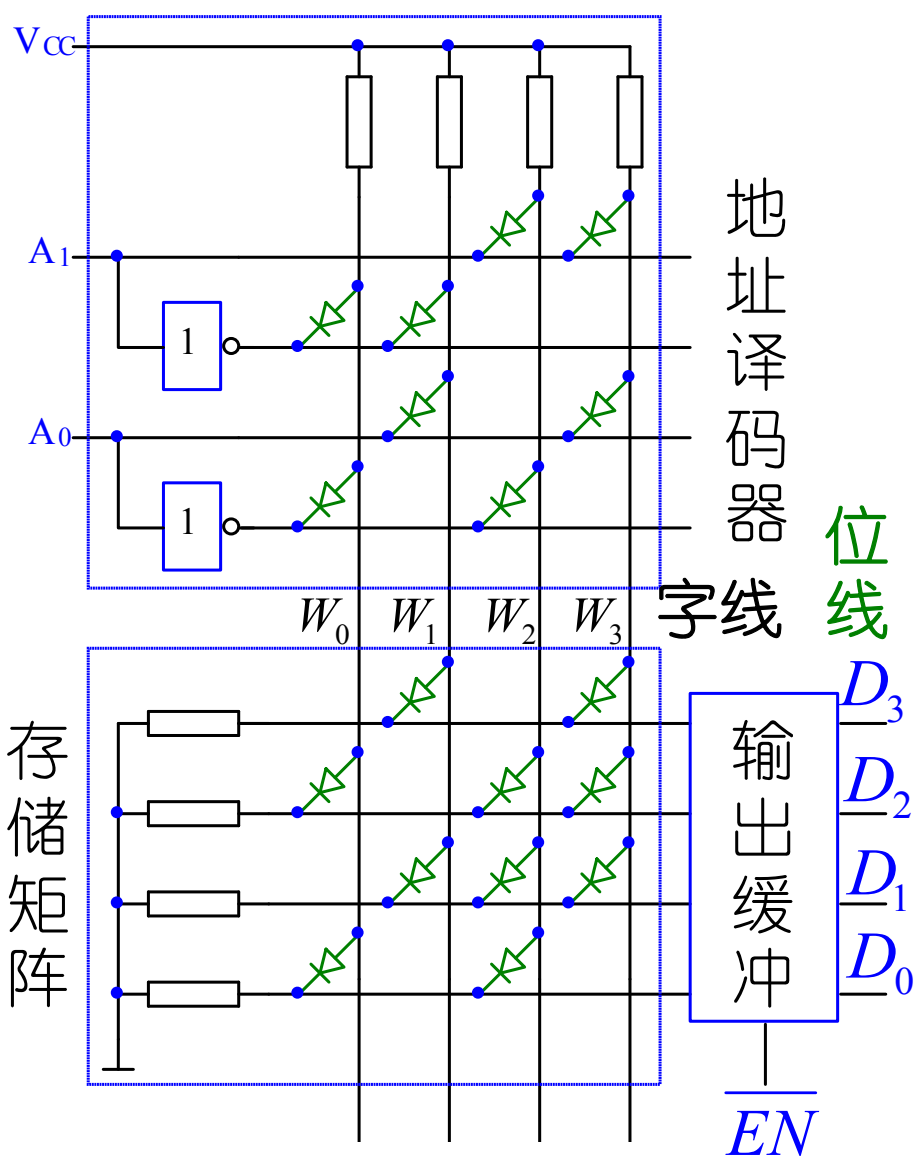
地址译码器是一个**与门阵列**，每一个字线对应一个最小项，且**是全部最小项。**



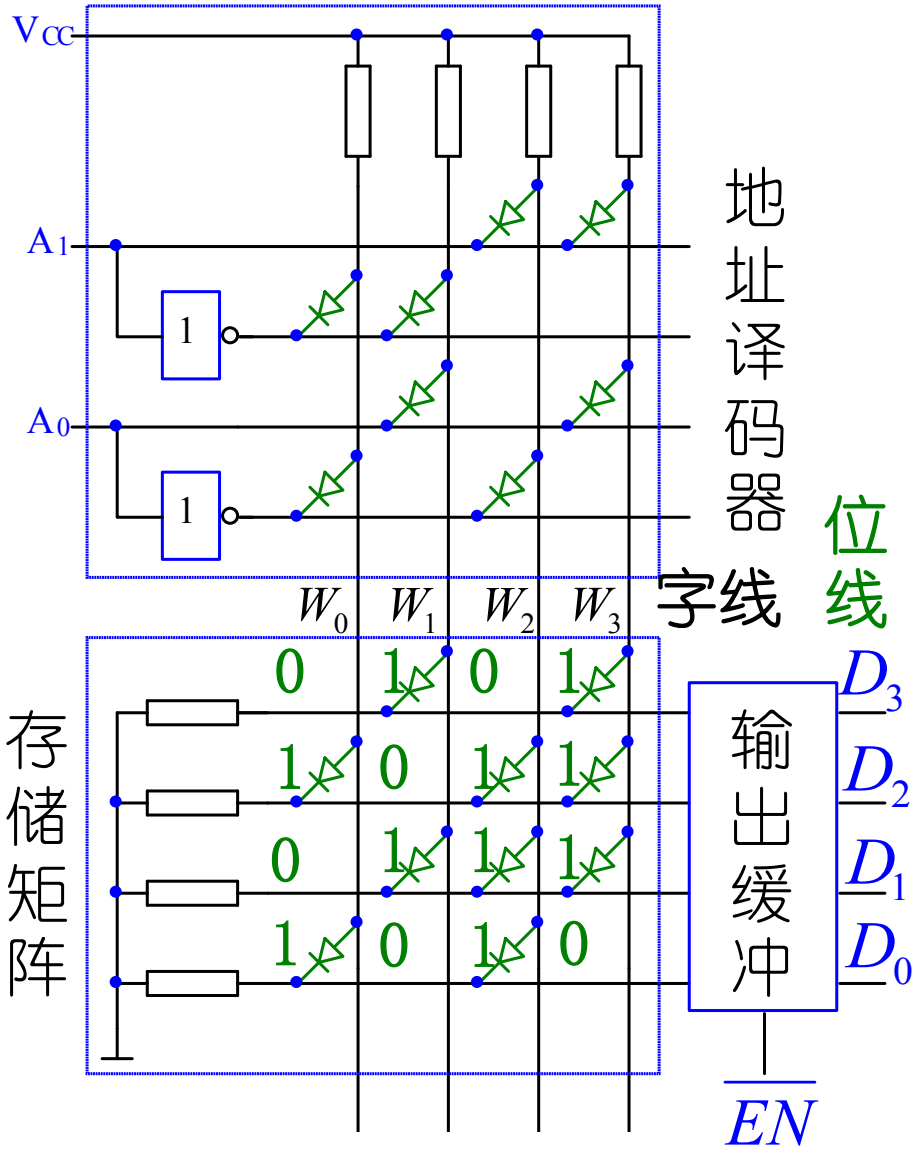
☆ 存储矩阵

$$D_3 = W_3 + W_1 = A_1 A_0 + \overline{A}_1 A_0$$
$$D_2 = W_3 + W_2 + \overline{W}_0 = A_1 A_0 + A_1 \overline{A}_0 + \overline{A}_1 \overline{A}_0$$
$$D_1 = W_3 + W_2 + \overline{W}_1 = A_1 A_0 + A_1 \overline{A}_0 + \overline{A}_1 A_0$$
$$D_0 = W_2 + W_0 = A_1 \overline{A}_0 + \overline{A}_1 \overline{A}_0$$



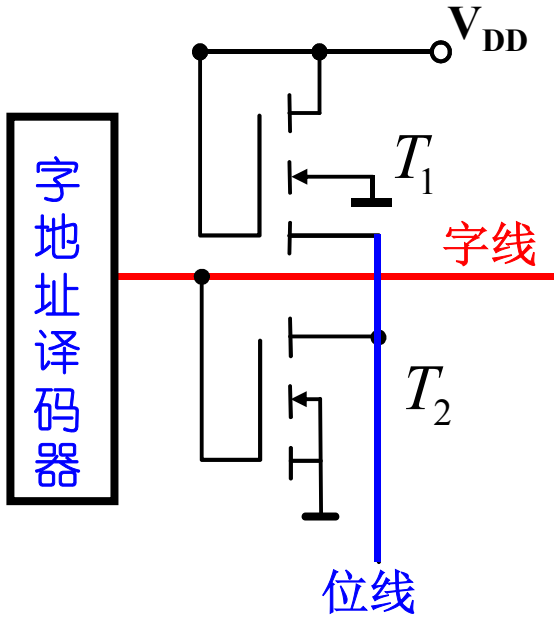


存储矩阵是一个**或门阵列**，每一个位线是将所对应的与项相加，**是最小项之和。**



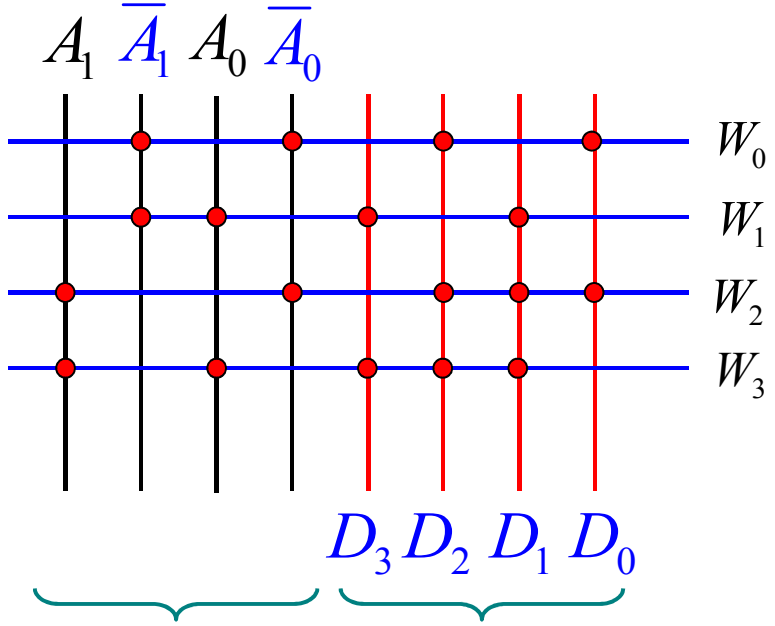
字线**W**和位线**D**的每个交叉点都是一个基本存储单元。交叉点接二极管时相当于存**1**，没有接二极管相当于存**0**。交叉点的数目就是基本存储单元数。

交叉点还可以接三极管、**MOS**管等。



4×4固定ROM的结点图

结点图 Node Diagram



与阵列　　或阵列

译码器　　存储内容

ROM与或逻辑阵列

A_1	A_0	W_0	W_1	W_2	W_3	D_3	D_2	D_1	D_0
0	0	1	0	0	0	0	1	0	1
0	1	0	1	0	0	1	0	1	0
1	0	0	0	1	0	0	1	1	1
1	1	0	0	0	1	1	1	1	0

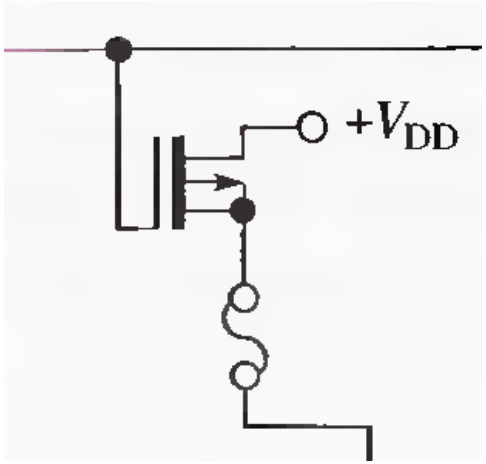
$$W_0 = \overline{A}_1 \overline{A}_0 \quad W_1 = \overline{A}_1 A_0$$
$$W_2 = A_1 \overline{A}_0 \quad W_3 = A_1 A_0$$

$$D_3 = W_3 + W_1 = A_1 A_0 + \overline{A}_1 A_0$$
$$D_2 = W_3 + W_2 + W_0 = A_1 A_0 + A_1 \overline{A}_0 + \overline{A}_1 \overline{A}_0$$
$$D_1 = W_3 + W_2 + W_1 = A_1 A_0 + A_1 \overline{A}_0 + \overline{A}_1 A_0$$
$$D_0 = W_2 + W_0 = A_1 \overline{A}_0 + \overline{A}_1 \overline{A}_0$$

逻辑函数发生器

2、可编程ROM (Programmable ROM, PROM)

PROM所存的数据，由用户自己根据要求写入。但是只能写一次，不允许第二次改写。



适合于小批量试产使用，有保密位，可以加密。

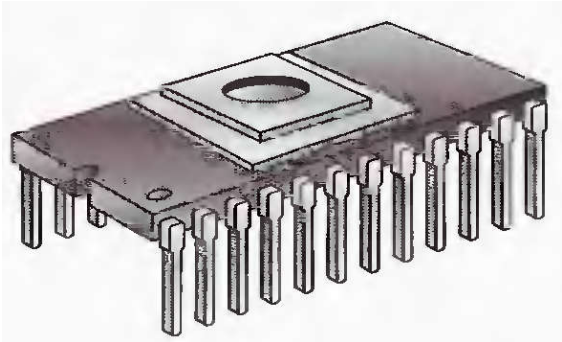
3、可擦除可编程ROM (Erasable EPROM, EPROM)

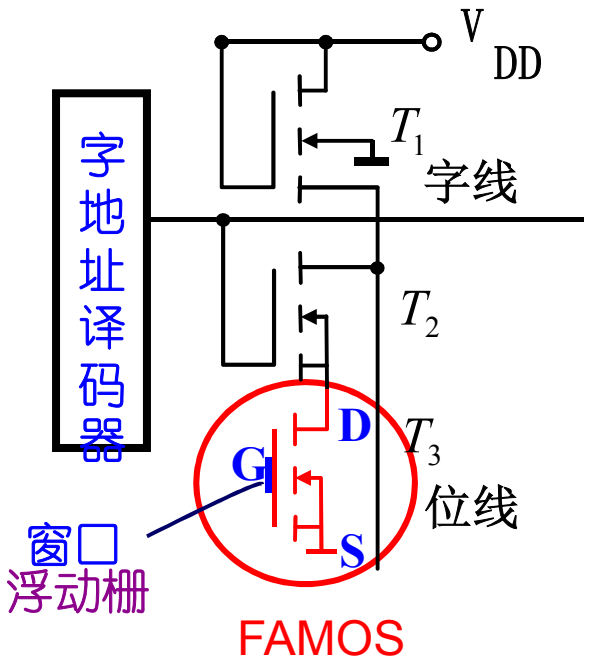
EPROM:用紫外光可以擦除ROM中全部信息。擦除时间几分钟，然后用专用编程器进行编程写入。

EEPROM:电可擦除ROM，直接在编程器上用电压信号进行擦除。重新写入和擦除同步进行。擦

EAPROM:直接在系统中擦除和改写，容，也可以只擦除部分字节。正常使用只

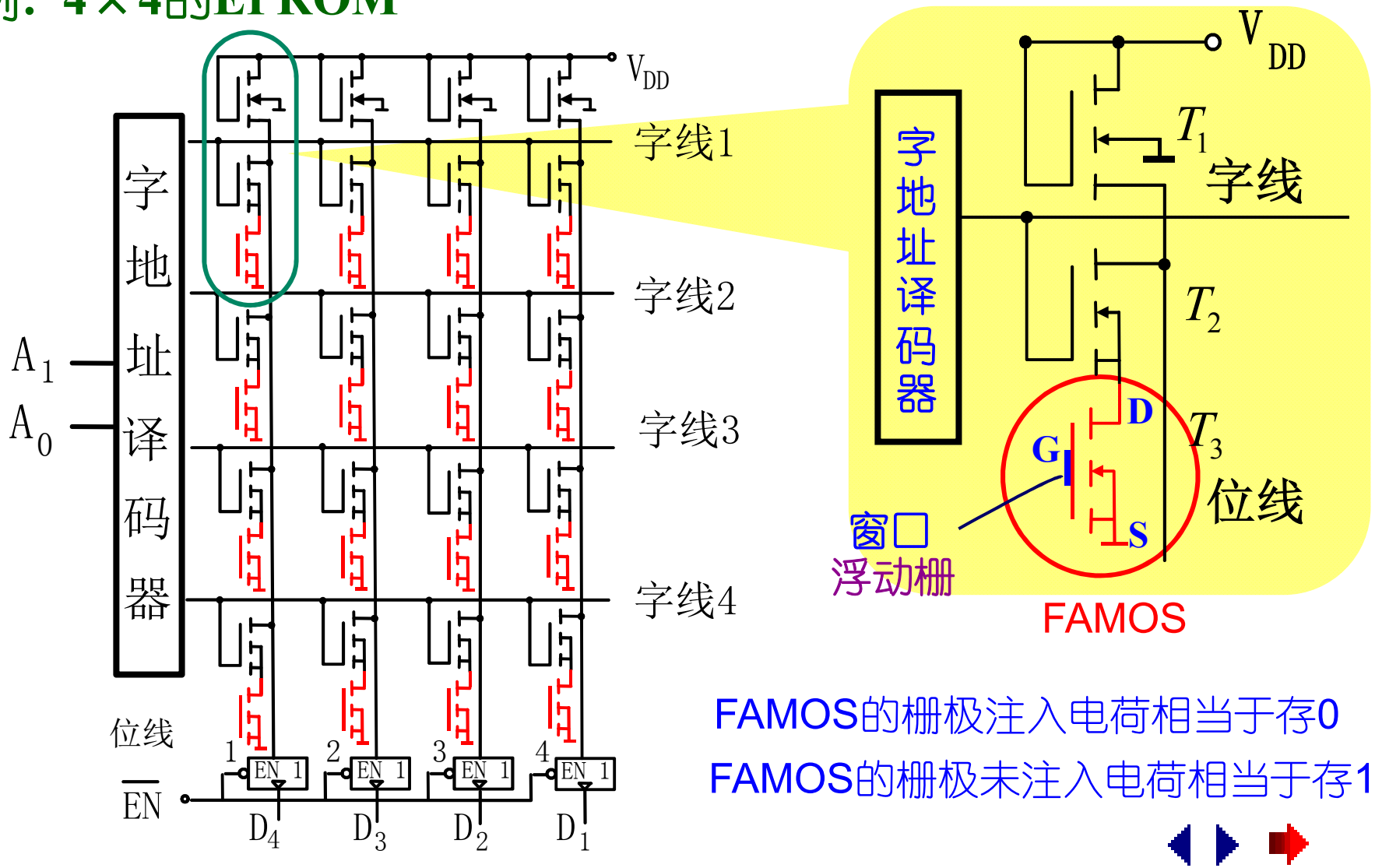
程序调试期间使用





3、可擦除可编程ROM(EPROM)

例：4×4的EPROM



FAMOS的栅极注入电荷相当于存0

FAMOS的栅极未注入电荷相当于存1

	与阵列 (地址译码)	或阵列 (存储矩阵)
ROM	固定	固定
PROM	固定	可编程(一次)
EPROM	固定	可编程(多次)
PLA	可编程	可编程

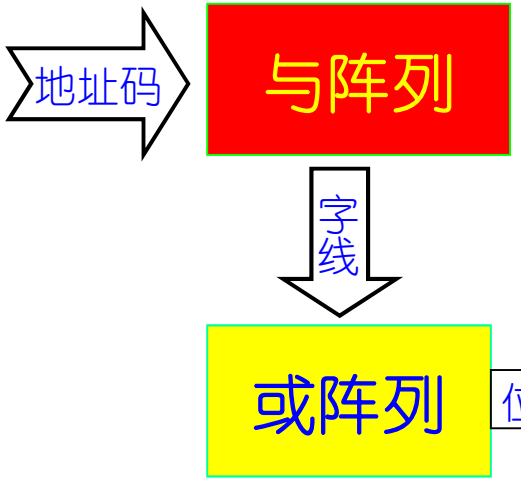
PLA, Programmable Logic Array

ROM:与阵列是固定的，是不可编程的，叫做完全译码器，如果有n位地址输入，与阵列就必须存储2^n个最小项。或阵列根据需要是可编程的。缺点：不使用的最小项占用存储容量。

PLA特点:与阵列、或阵列都是可编程的，不使用的最小项不占用存储容量。

第二节 可编程逻辑阵列（PLA）

一、PLA的结构与工作原理



与阵列存放的不是最小项，而是**与项**。

逻辑函数不用最小项之和表达式，而是用最简与－或表达式。

与项相加，可编程

存储容量＝输入端数×与项数×输出端数

二、PROM应用举例

例**1**：用**ROM**实现**4**位二进制到格雷码的转换。

代码转换表

A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

① 列真值表：

② 由真值表写出最小项之和表达式。

$W=\sum m(8,9,10,11,12,13,14,15)$

$X=\sum m(4,5,6,7,8,9,10,11)$

$Y=\sum m(2,3,4,5,10,11,12,13)$

$Z=\sum m(1,2,5,6,9,10,13,14)$

③ 根据最小项画出与或逻辑阵列图

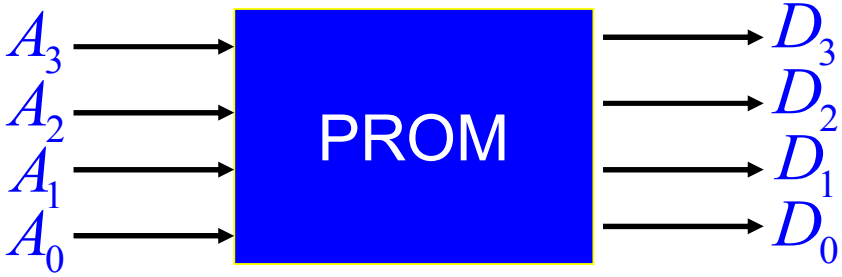
☆ 先画地址译码器，四变量，八输入，十六个最小项，8×16阵列。

☆ 再画或阵列，有四输出，每个输出按最小项加表示。共4×16阵列。

★ 选用存储容量＝16×4的PROM实现，令：

PROM地址码A_3~A_0=ABCD

则PROM D_3~D_0=WXYZ

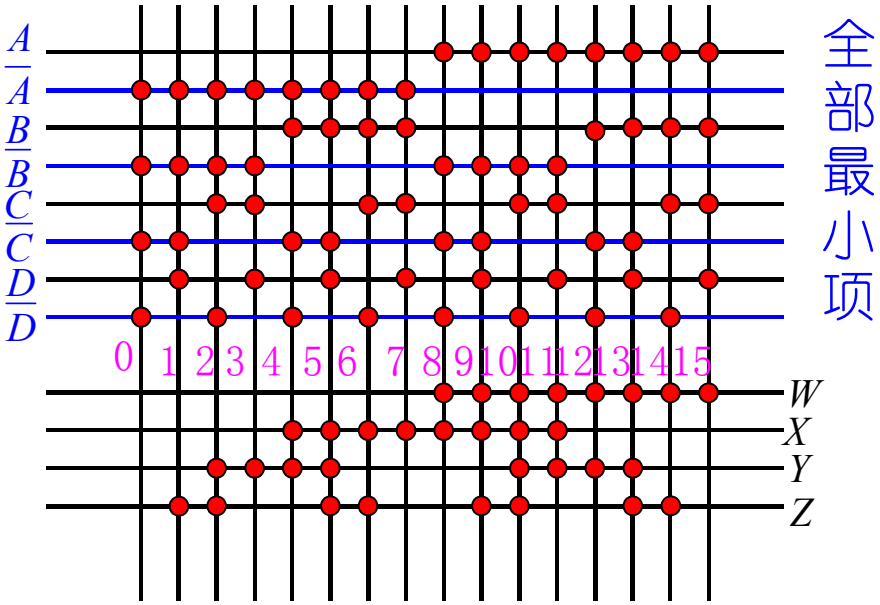


$W=\sum_m(8,9,10,11,12,13,14,15)$

$X=\sum_m(4,5,6,7,8,9,10,11)$

$Y=\sum_m(2,,3,4,5,10,11,12,13)$

$Z=\sum_m(1,2,5,6,9,10,13,14)$



点阵图相当于将真值表存入PROM。

与阵列：不可编程，所有最小项都必须全部画出。

或阵列：可编程，根据要求选用。

2、用ROM实现组合逻辑函数

例2：用ROM实现一位全加器

全加器真值表：

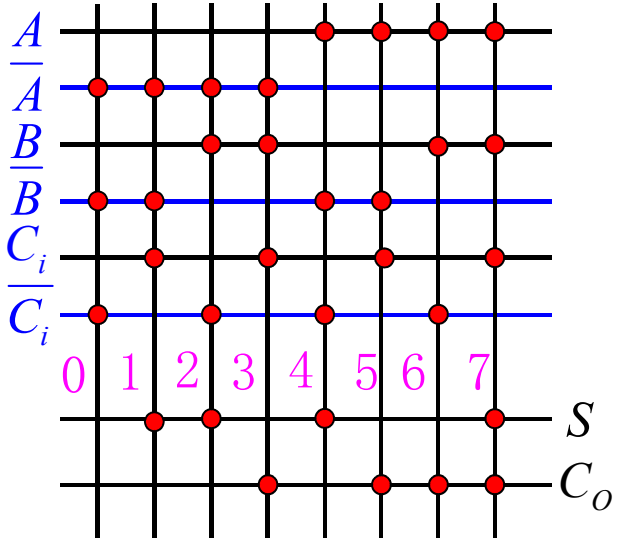
A	B	C_i	S	C_O
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

最小项之和表达式

$S = \sum_m(1、2、4、7)$

$C_0 = \sum_m(3、5、6、7)$

画点阵图：



3、用PLA实现组合逻辑函数

例3：用PLA实现4位二进制到格雷码的转换。

A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

解：1、列真值表

2、写出逻辑函数最简与－或式。

★卡诺图法。★直接观擦法。

W=A

X=A⊕B $= \overline{A}B + A\overline{B}$

Y=B⊕C $= \overline{B}C + B\overline{C}$

Z=C⊕D $= \overline{C}D + C\overline{D}$

地址译码器输出字线是7个与项而不是最小项。

令字线：$P_0 = \overline{A}$ $P_3 = \overline{B}C$ $P_6 = C\overline{D}$

$P_1 = A\overline{B}$ $P_4 = B\overline{C}$

$P_2 = A\overline{B}$ $P_5 = \overline{C}D$

格雷码 $W=P_0$ $Y=P_3+P_4$

输出： $X=P_1+P_2$ $Z=P_5+P_6$

3、画PLA阵列图

★4个变量，八条输入线，对应每个输入变量的原变量和反变量。

★有几个与项画几条字线。

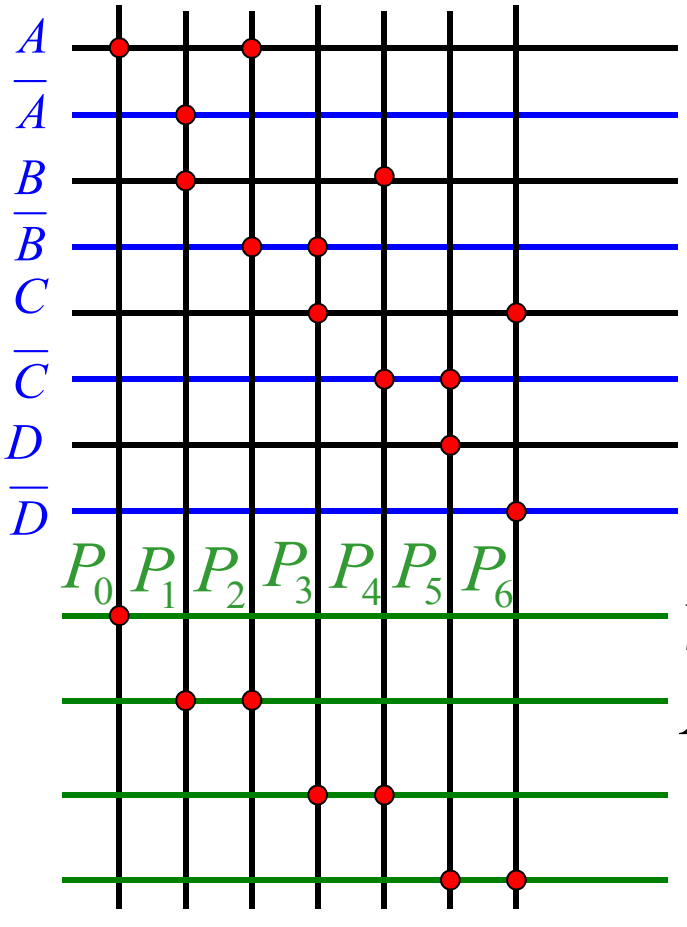
$P_0 = A$　$P_1 = \overline{A}B$　$P_2 = A\overline{B}$

$P_3 = \overline{B}C$　$P_4 = B\overline{C}$　$P_5 = \overline{C}D$

$P_6 = C\overline{D}$

★或阵列是与项相加

存储容量为：**8×7×4**



W　W=P_0

X　X=P_1+P_2

Y　Y=P_3+P_4

Z　Z=P_5+P_6

同样一个码制变换电路，ROM占用存储单元个数比PLA占用个数要多很多。

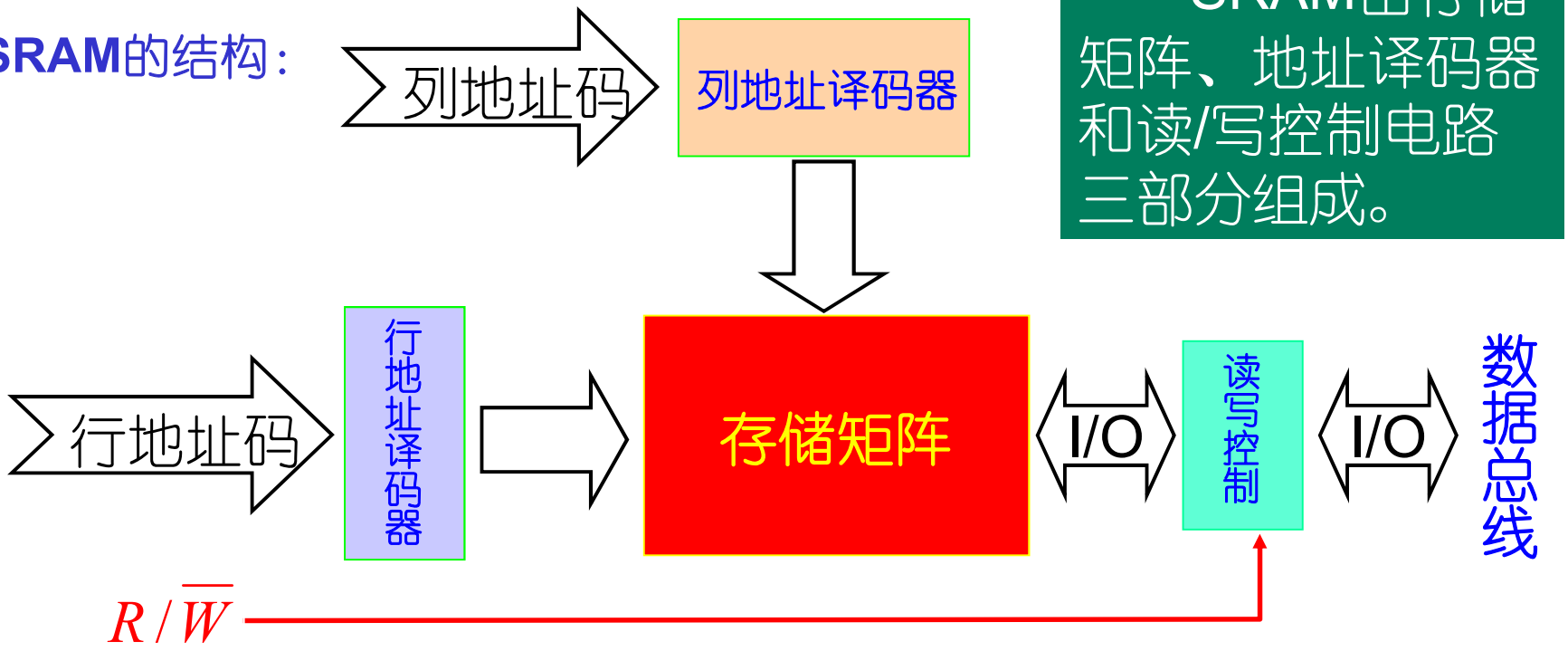
用同样的硅片面积PLA可以实现更多逻辑功能。

第三节 随机存储器（RAM）

RAM又可以分为静态存储器SRAM和动态存储器DRAM两类，DRAM存储单元结构非常简单，它所能达到的集成度远高于SRAM,但DRAM的工作速度没有SRAM快。

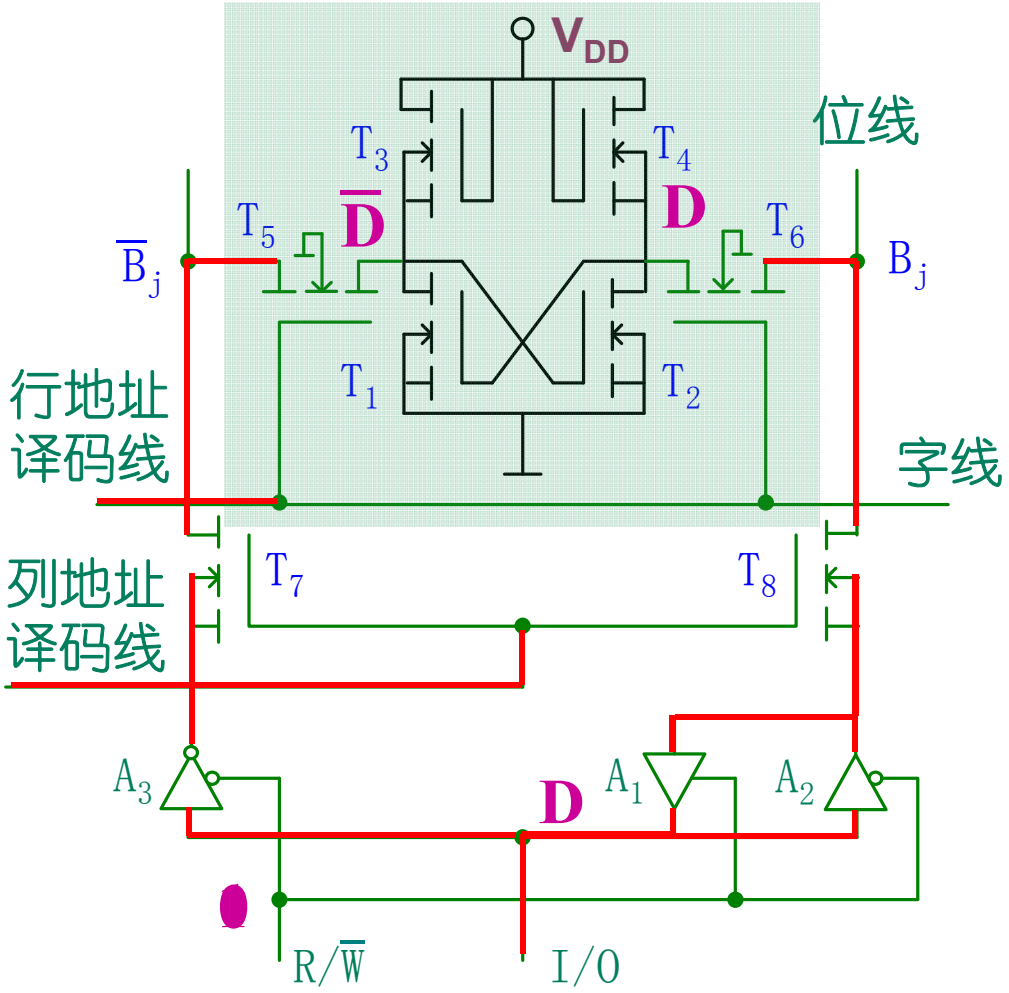
- 静态随机存储器（SRAM）

1. SRAM的结构：



SRAM由存储矩阵、地址译码器和读/写控制电路三部分组成。

2. SRAM的基本存储单元：



工作原理：

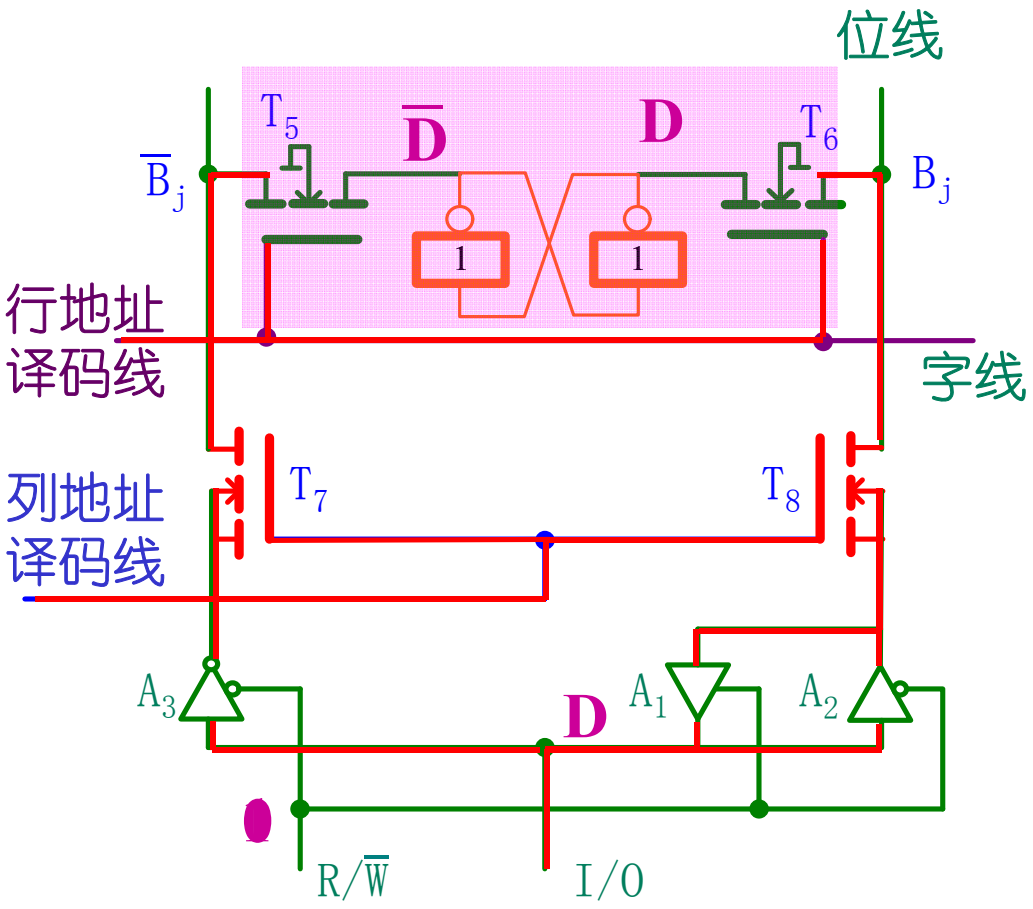
- 写入：行、列地址选中

  T_5、T_6、T_7、T_8导通，
  $R/\overline{W}=0$

- 读出：行、列地址选中

  T_5、T_6、T_7、T_8导通，
  $R/\overline{W}=1$

2. SRAM的基本存储单元：



工作原理：

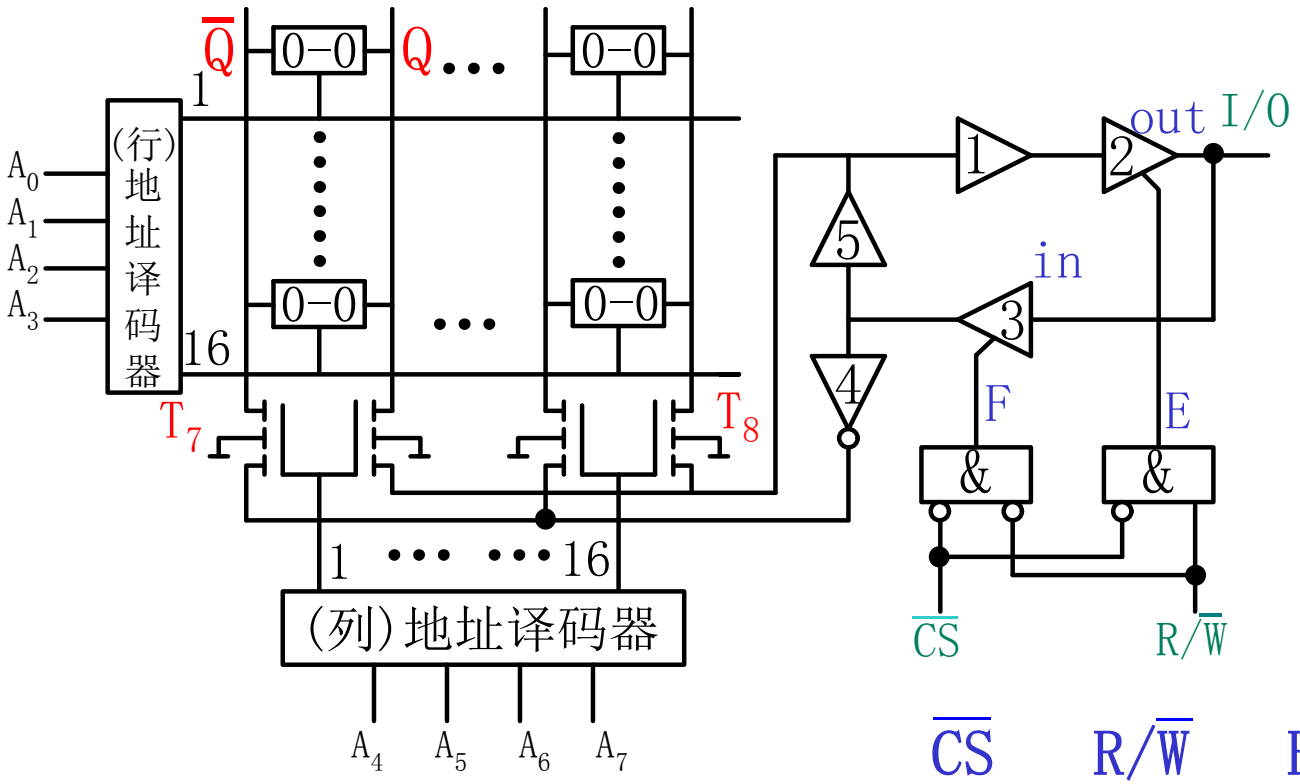
- 写入：行、列地址选中

  T_5、T_6、T_7、T_8导通，
  $R/\overline{W}=0$

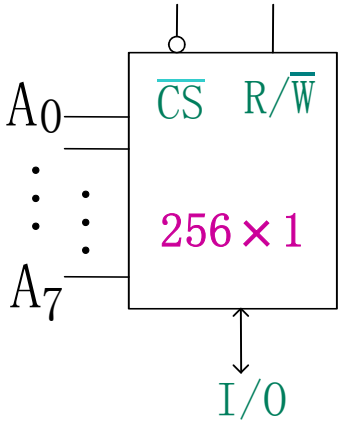
- 读出：行、列地址选中

  T_5、T_6、T_7、T_8导通，
  $R/\overline{W}=1$

例. 256×1位RAM



框图：



$\overline{CS}$	$R/\overline{W}$	E	F	
0	1	1	0	读出
0	0	0	1	写入
1	×	0	0	禁止

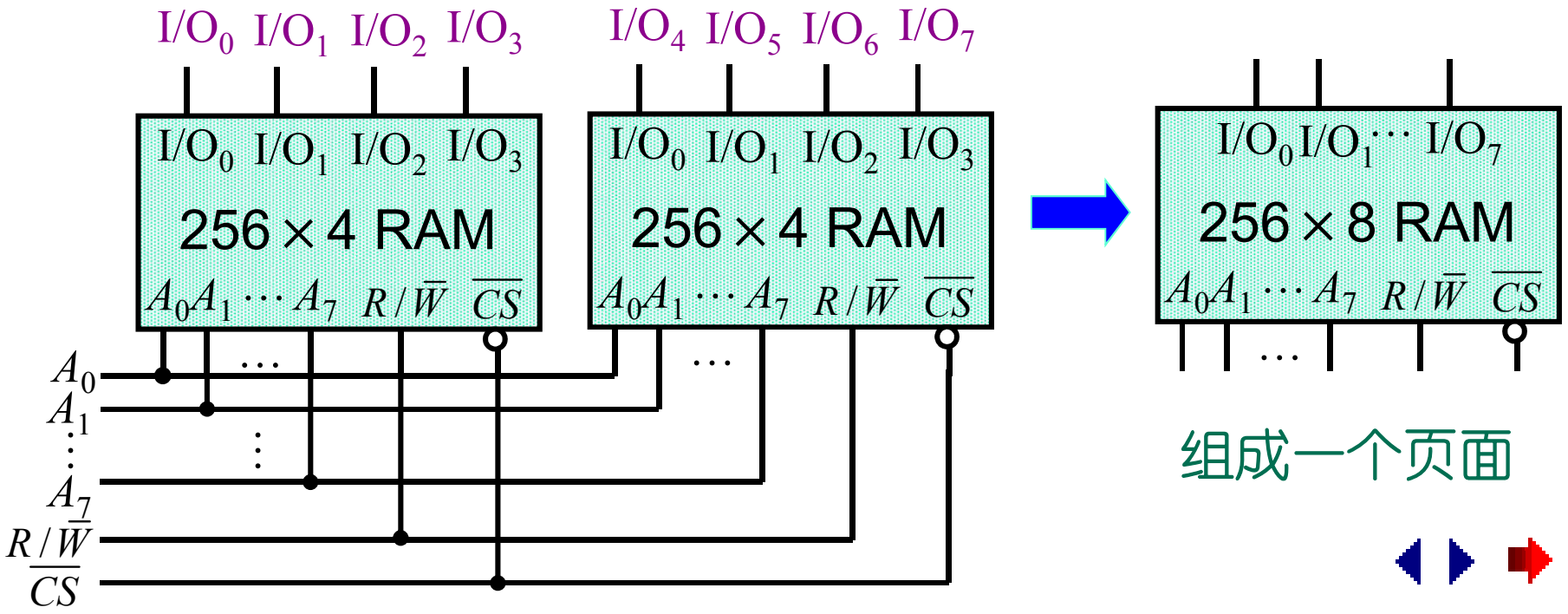
存储容量的扩展

例：如何将256×4的RAM扩展成64K×8的RAM？

☆ 首先确定需要多少片进行扩展

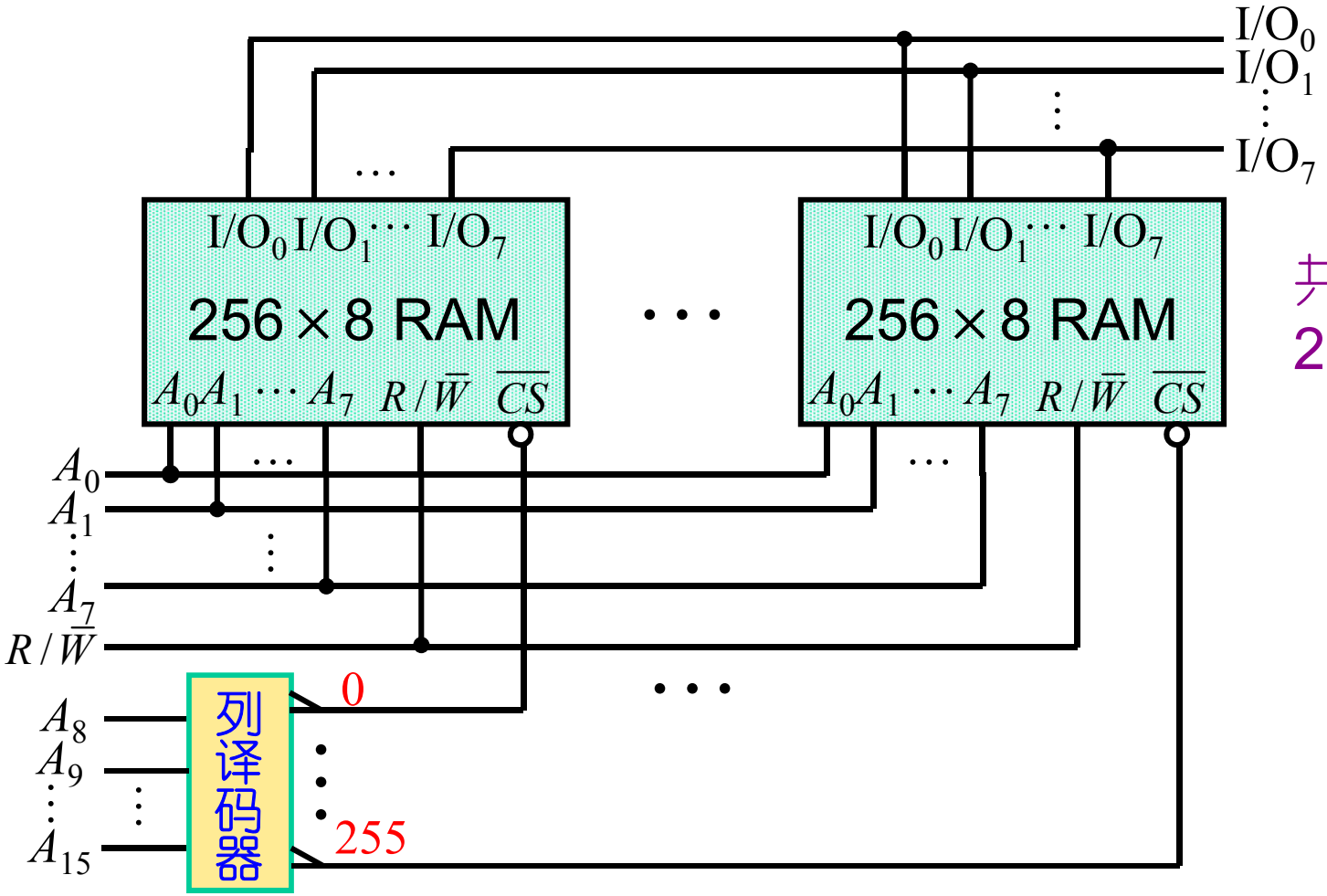
$$\frac{64\text{K}\times 8}{256\times 4}=512\text{ 片}$$

☆ 位扩展



组成一个页面

☆ 字扩展　　64K×8RAM需16位地址码A_{15}~A_0，其中高8位作列地址码，低8位作行地址码。



共256片
256×8 RAM