第14章

存储器与可编程逻辑器件

1　只读存储器

存储器的分类

RAM：在工作时既能从中读出（取出）信息，又能随时写入（存入）信息，但断电后所存信息消失。

ROM：在工作时只能从中读出信息，不能写入信息，且断电后其所存信息在仍能保持。

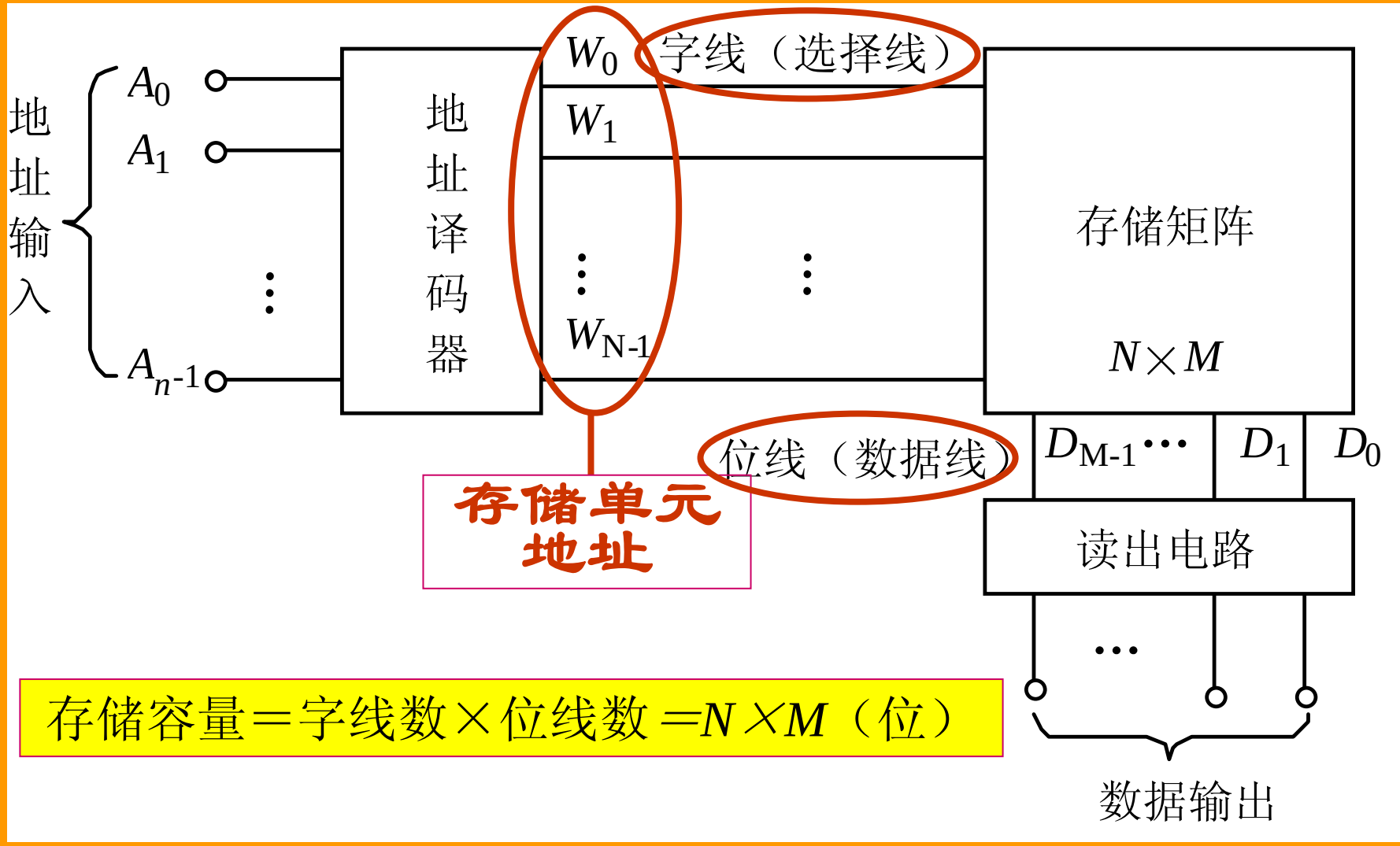
ROM的分类

掩膜ROM：不能改写。

PROM：只能改写一次。

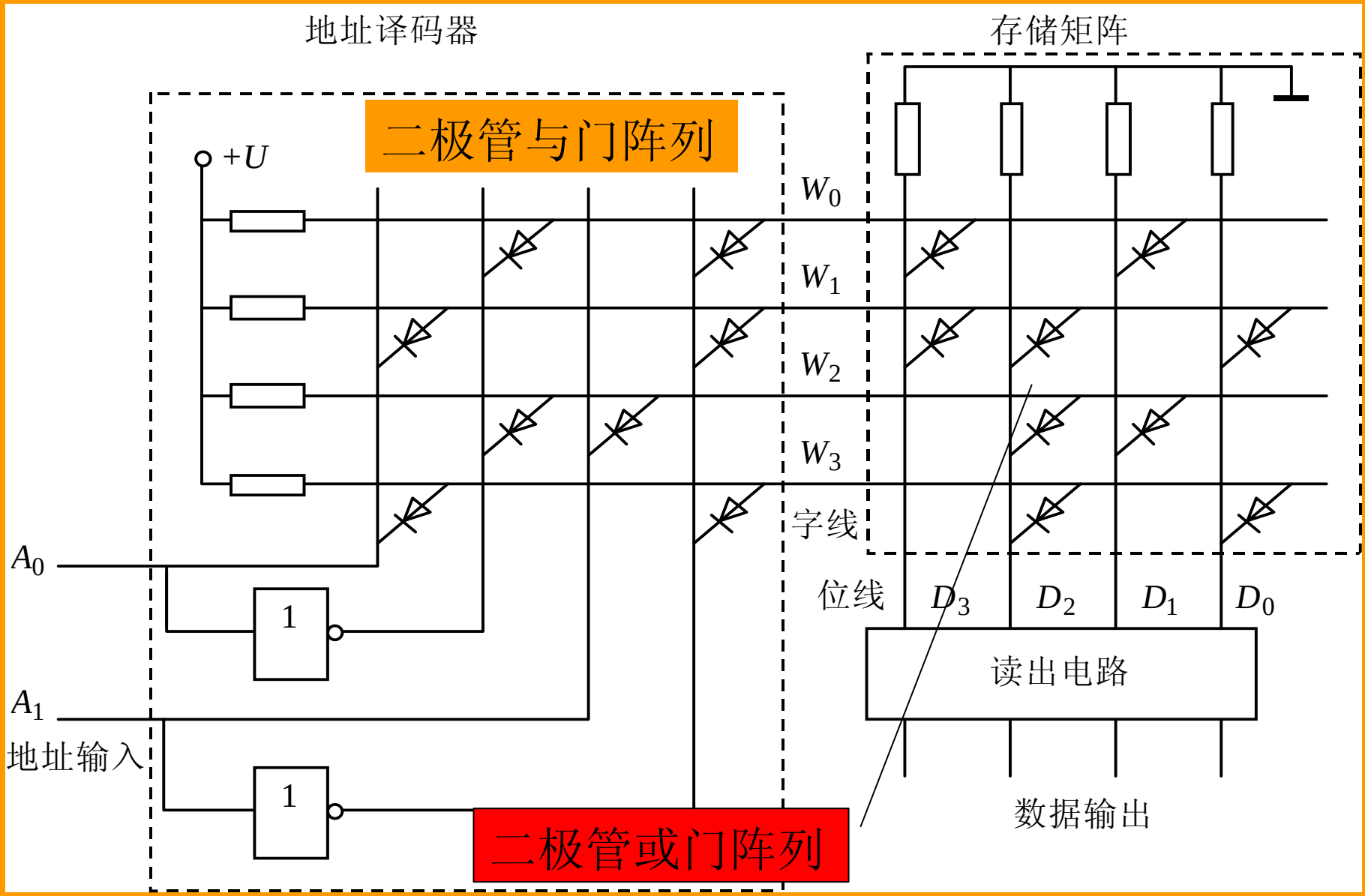
EPROM：可以改写多次。

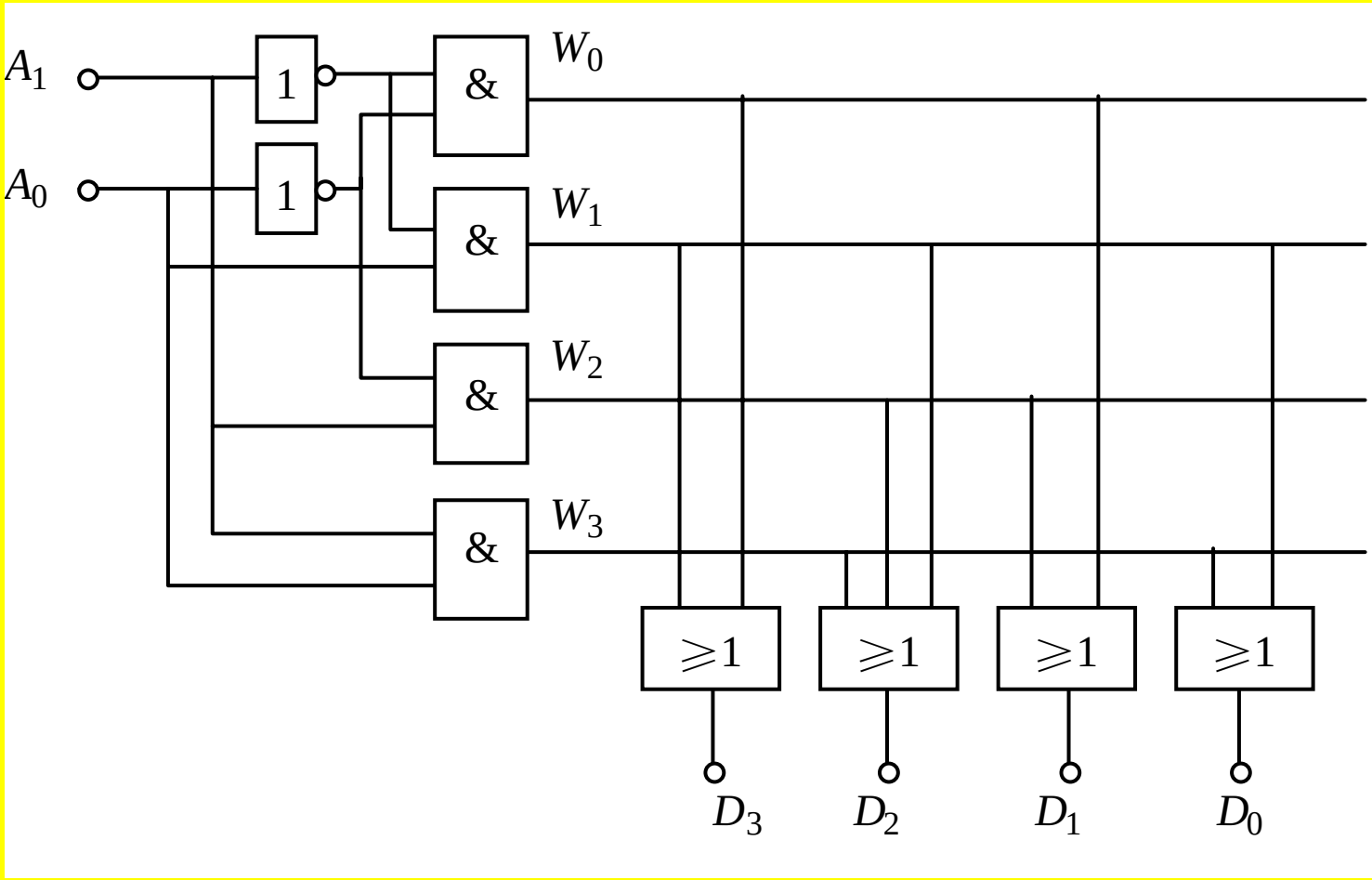
ROM的结构



存储容量＝字线数×位线数 $=N \times M$（位）

ROM的工作原理


地址译码器
存储矩阵
二极管与门阵列
+U
W_0
W_1
W_2
W_3
字线
A_0
A_1
地址输入
1
1
位线
D_3
D_2
D_1
D_0
读出电路
数据输出
二极管或门阵列



$$W_0 = \overline{A}_1 \overline{A}_0$$

$$W_1 = \overline{A}_1 A_0$$

$$W_2 = A_1 \overline{A}_0$$

$$W_3 = A_1 A_0$$

$$D_3 = W_0 + W_1 = \overline{A}_1 \overline{A}_0 + \overline{A}_1 A_0$$

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

$$D_1 = W_0 + W_2 = \overline{A}_1 \overline{A}_0 + A_1 \overline{A}_0$$

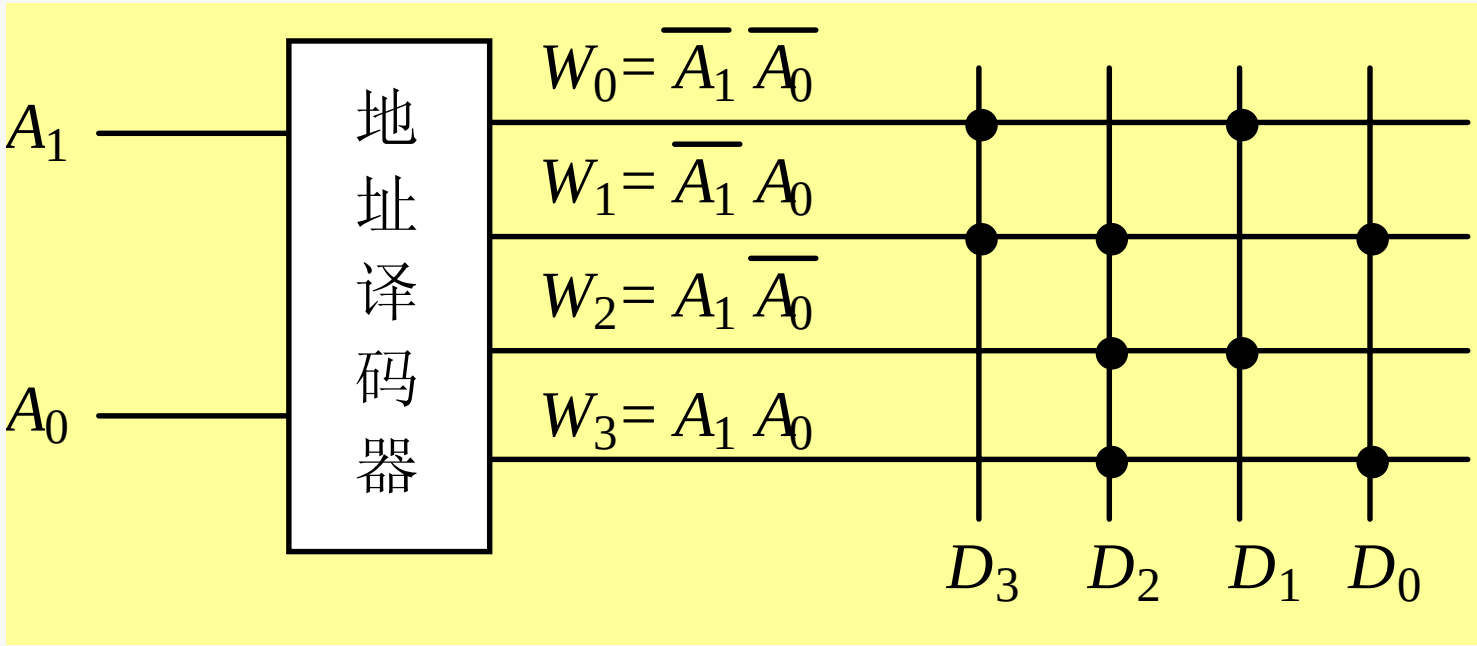
$$D_0 = W_1 + W_3 = \overline{A}_1 A_0 + A_1 A_0$$

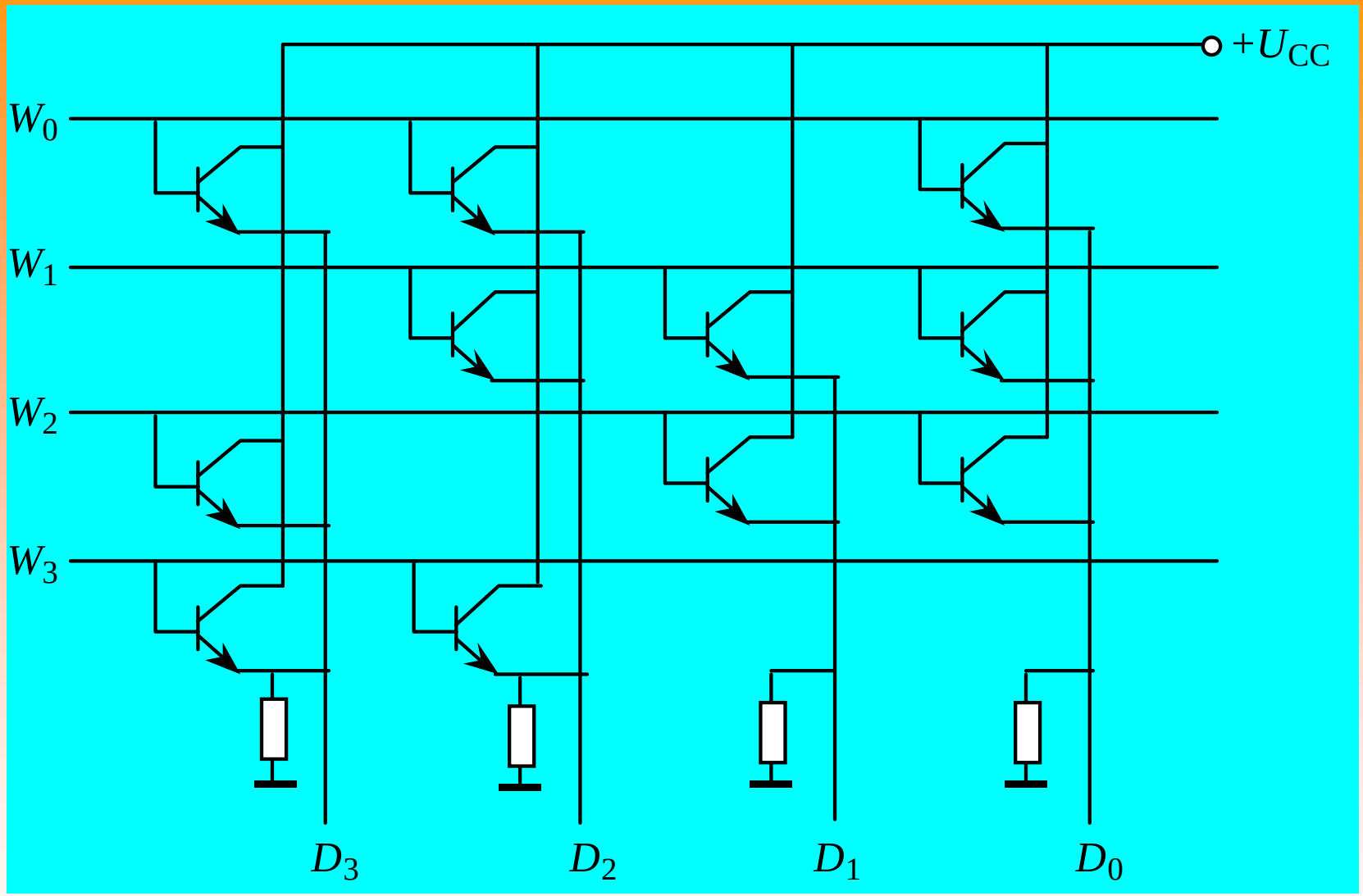
存储内容

地址代码		字线译码结果				存储内容			
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	1	0	1	0
0	1	0	1	0	0	1	1	0	1
1	0	0	0	1	0	0	1	1	0
1	1	0	0	0	1	0	1	0	1

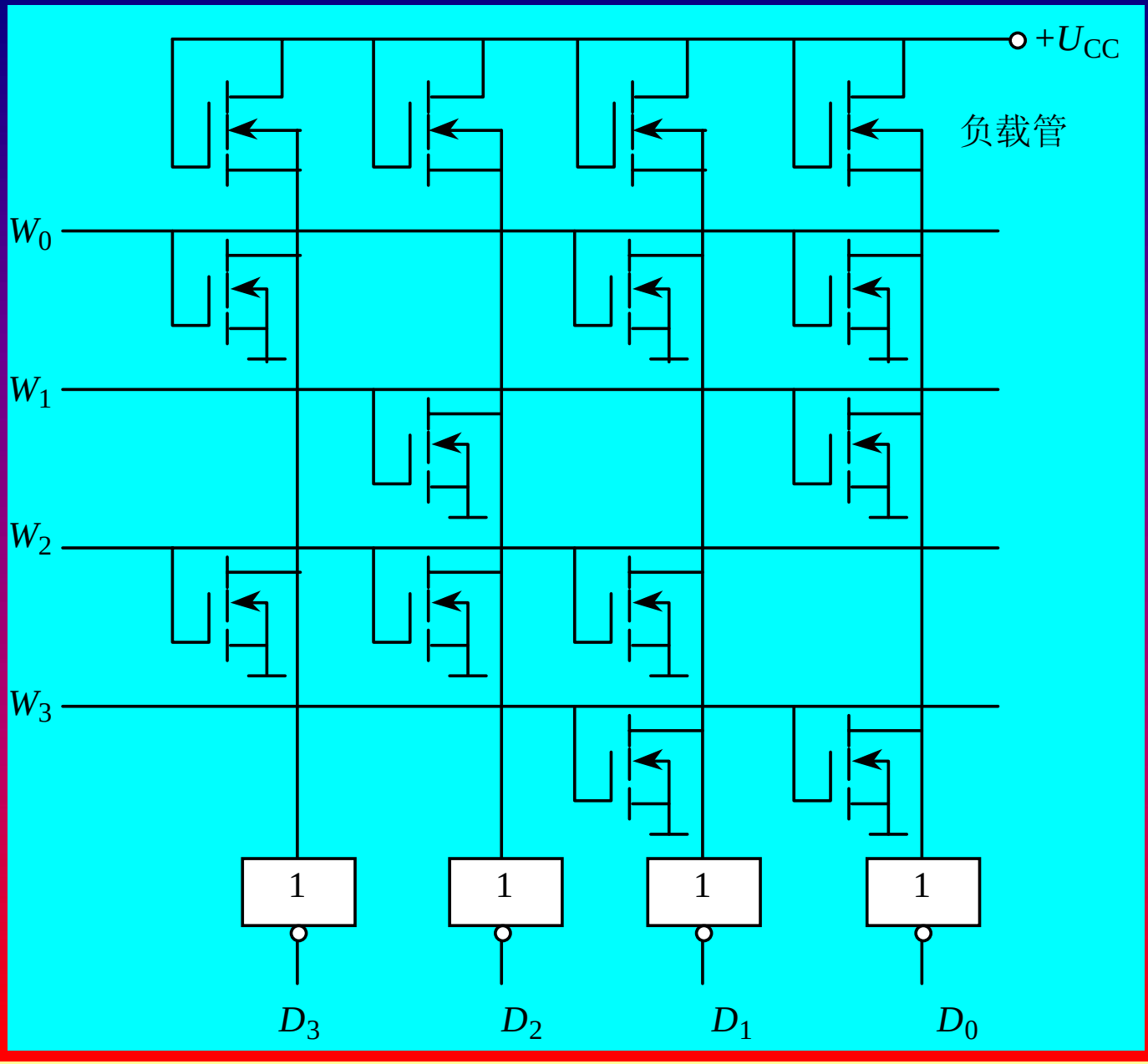
结合电路图及上表可以看出，接有二极管的交叉点存1，未接二极管的交叉点存0。存储单元是存1还是存0，完全取决于只读存储器的存储需要，设计和制造时已完全确定，不能改变；而且信息存入后，即使断开电源，所存信息也不会消失。所以，只读存储器又称为固定存储器。

ROM的阵列图


A_1
A_0
地址译码器
$W_0 = \overline{A_1}\,\overline{A_0}$
$W_1 = \overline{A_1}\,A_0$
$W_2 = A_1\,\overline{A_0}$
$W_3 = A_1\,A_0$
D_3 D_2 D_1 D_0

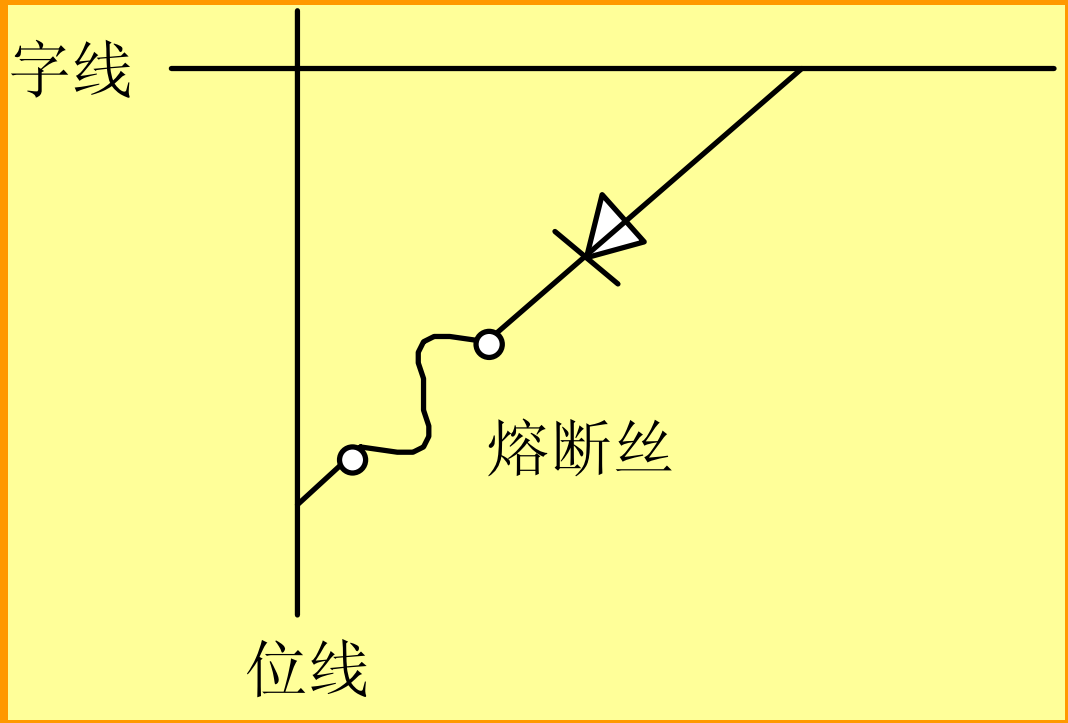


接有三极管的交叉点存1，未接三极管的交叉点存0。



接有场效应管的交叉点存1，未接场效应管的交叉点存0。

EPROM的存储单元



ROM的应用

1、用ROM实现组合逻辑函数

例 用ROM实现下列一组逻辑函数。

$$Y_1 = A \oplus B$$

$$Y_2 = AB + AC + BC$$

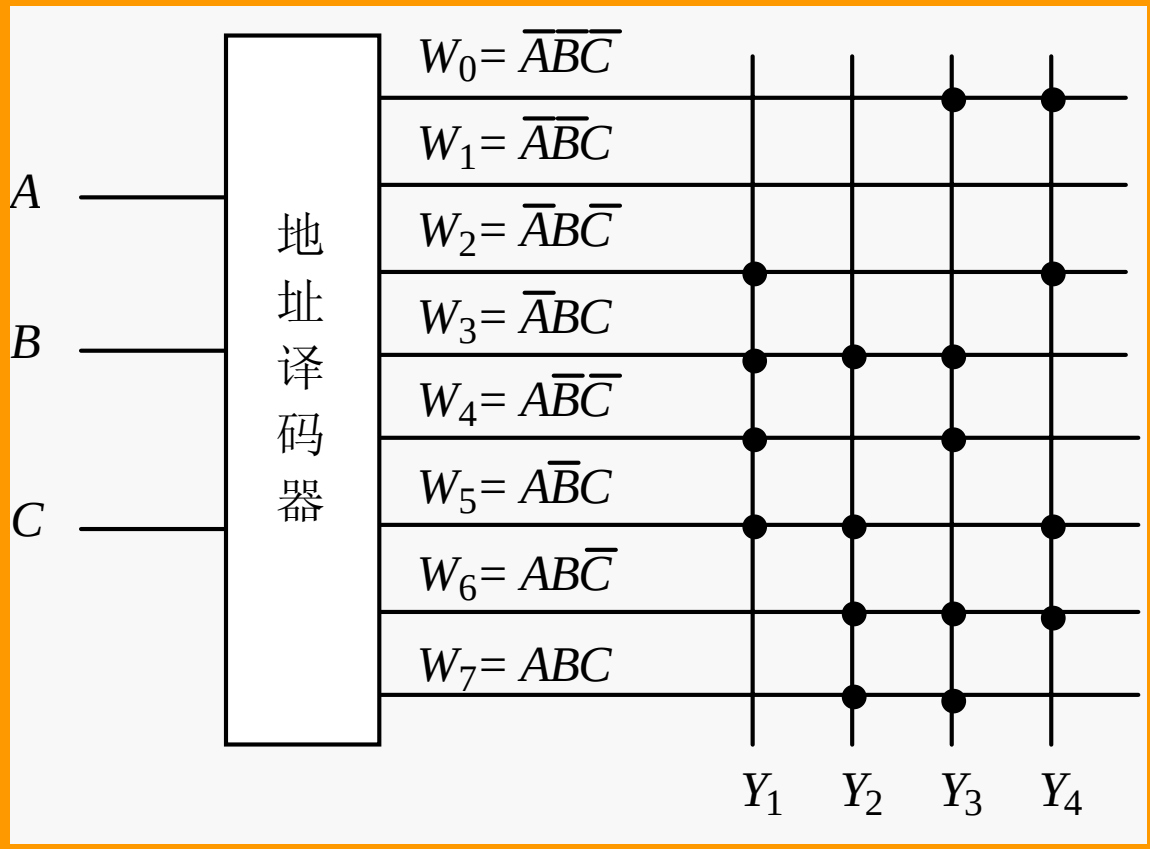
$$Y_3 = AB + BC + \overline{B}\,\overline{C}$$

$$Y_4 = \overline{A}\,\overline{C} + B\overline{C} + A\overline{B}C$$

解 （1）列真值表

A	B	C	被选中的字线	Y_1	Y_2	Y_3	Y_4
0	0	0	$W_0 = \overline{A}\,\overline{B}\,\overline{C} = 1$	0	0	1	1
0	0	1	$W_1 = AB\,\overline{C} = 1$	0	0	0	0
0	1	0	$W_2 = \overline{A}\,B\,\overline{C} = 1$	1	0	0	1
0	1	1	$W_3 = \overline{A}\,BC = 1$	1	1	1	0
1	0	0	$W_4 = A\,\overline{B}\,\overline{C} = 1$	1	0	1	0
1	0	1	$W_5 = A\,\overline{B}\,C = 1$	1	1	0	1
1	1	0	$W_6 = AB\,\overline{C} = 1$	0	1	1	1
1	1	1	$W_7 = ABC = 1$	0	1	1	0

（2）选择合适的ROM，对照真值表画出逻辑函数的阵列图。



2、用ROM作函数运算表

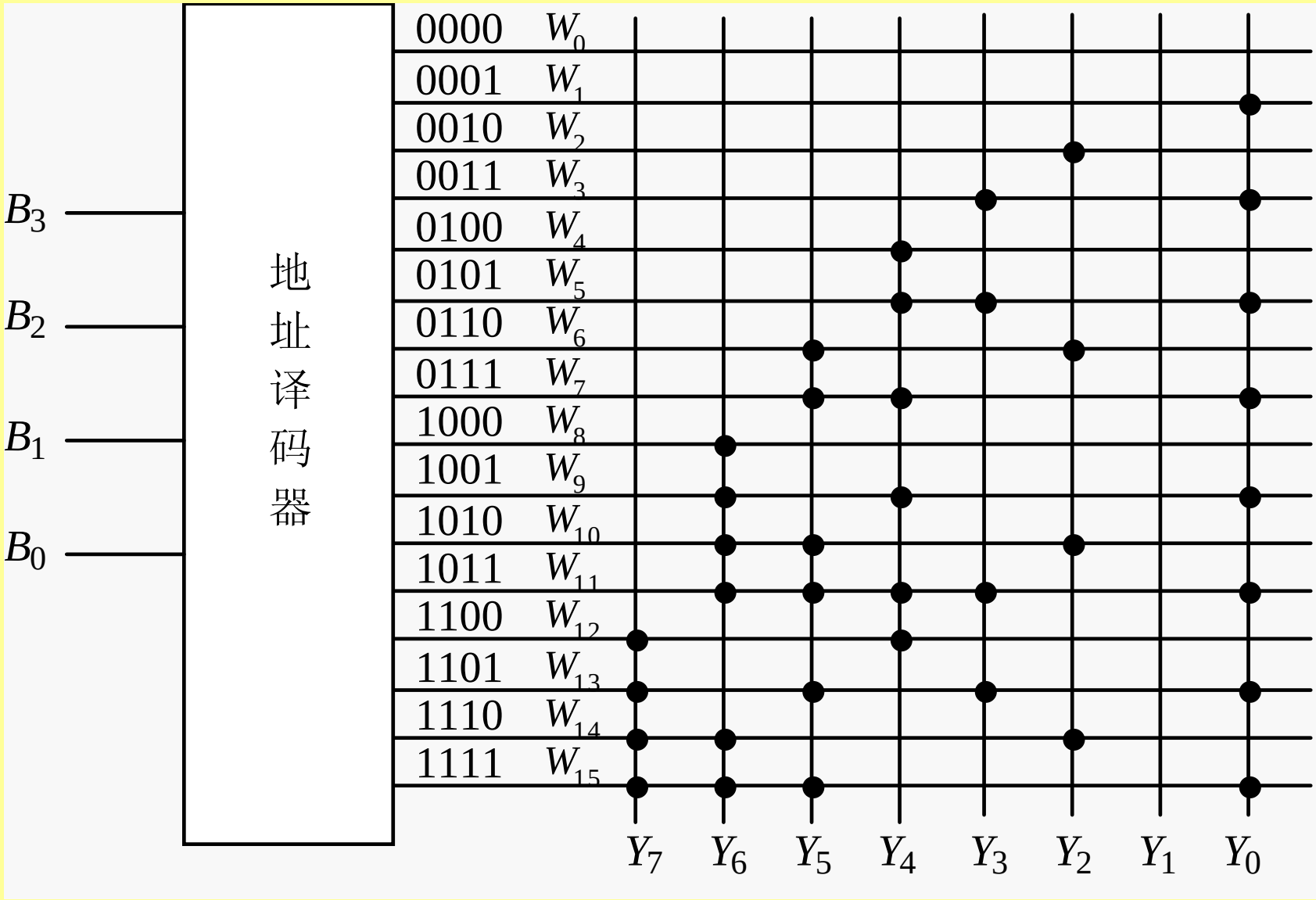
例 用ROM构成能实现函数$y=x^2$的运算表电路。

设x的取值范围为0～15的正整数，则对应的是4位二进制正整数，用$B=B_3B_2B_1B_0$表示。根据$y=x^2$可算出y的最大值是$15^2=225$，可以用8位二进制数$Y=Y_7Y_6Y_5Y_4Y_3Y_2Y_1Y_0$表示。由此可列出$Y=B^2$即$y=x^2$的真值表。

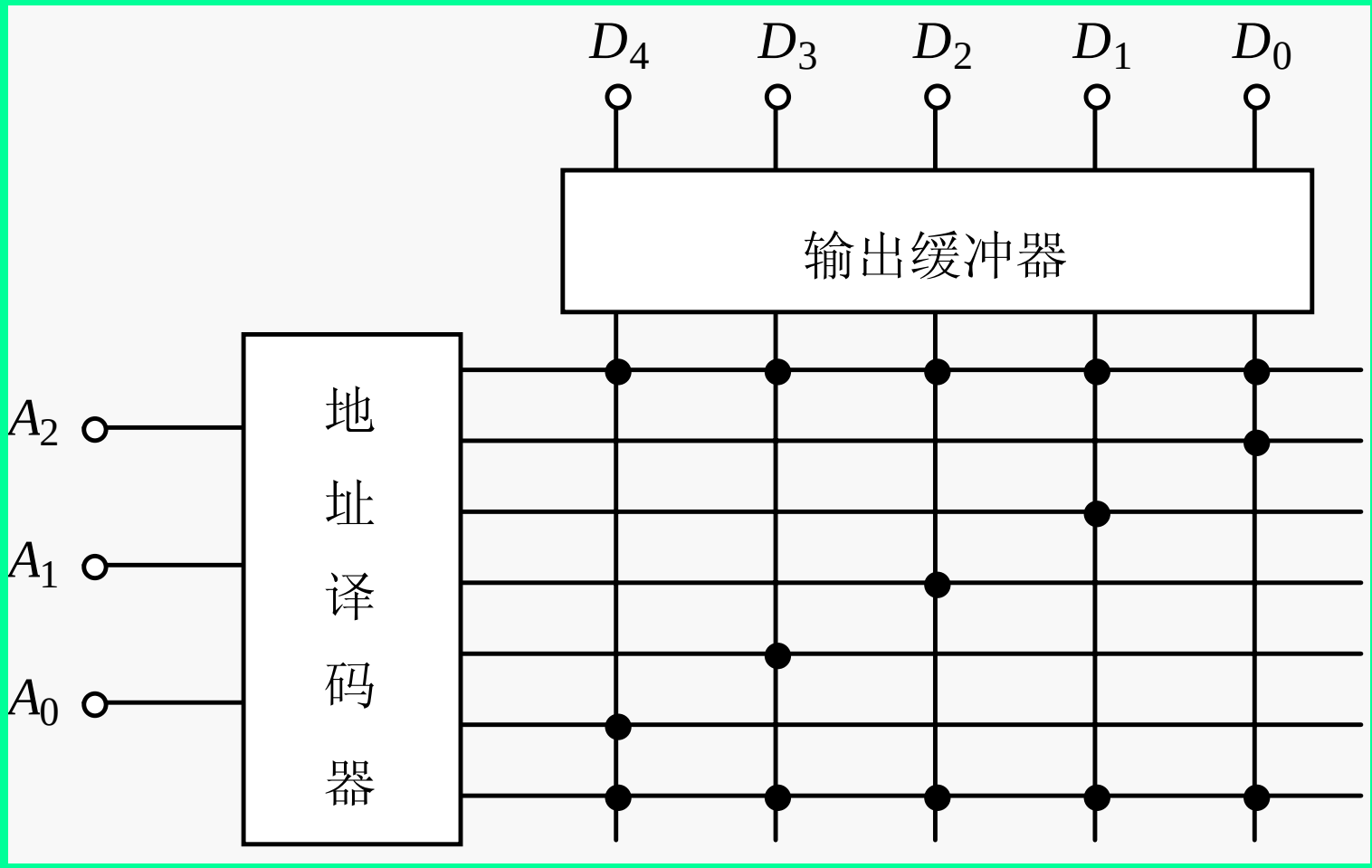
真值表

输入				输出								注
B_3	B_2	B_1	B_0	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0	十进制数
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	1	1
0	0	1	0	0	0	0	0	0	1	0	0	4
0	0	1	1	0	0	0	0	1	0	0	1	9
0	1	0	0	0	0	0	1	0	0	0	0	16
0	1	0	1	0	0	0	1	1	0	0	1	25
0	1	1	0	0	0	1	0	0	1	0	1	36
0	1	1	1	0	0	1	1	0	0	0	1	49
1	0	0	0	0	1	0	0	0	0	0	0	64
1	0	0	1	0	1	0	1	0	0	0	1	81
1	0	1	0	0	1	1	0	0	1	0	0	100
1	0	1	1	0	1	1	1	1	0	0	1	121
1	1	0	0	1	0	0	1	0	0	0	0	144
1	1	0	1	1	0	1	0	1	0	0	1	169
1	1	1	0	1	1	0	0	0	1	0	0	196
1	1	1	1	1	1	1	0	0	0	0	1	225

阵列图


B_3
B_2
B_1
B_0
地址译码器
0000 W_0
0001 W_1
0010 W_2
0011 W_3
0100 W_4
0101 W_5
0110 W_6
0111 W_7
1000 W_8
1001 W_9
1010 W_{10}
1011 W_{11}
1100 W_{12}
1101 W_{13}
1110 W_{14}
1111 W_{15}
Y_7 Y_6 Y_5 Y_4 Y_3 Y_2 Y_1 Y_0

3、用ROM作字符发生器电路

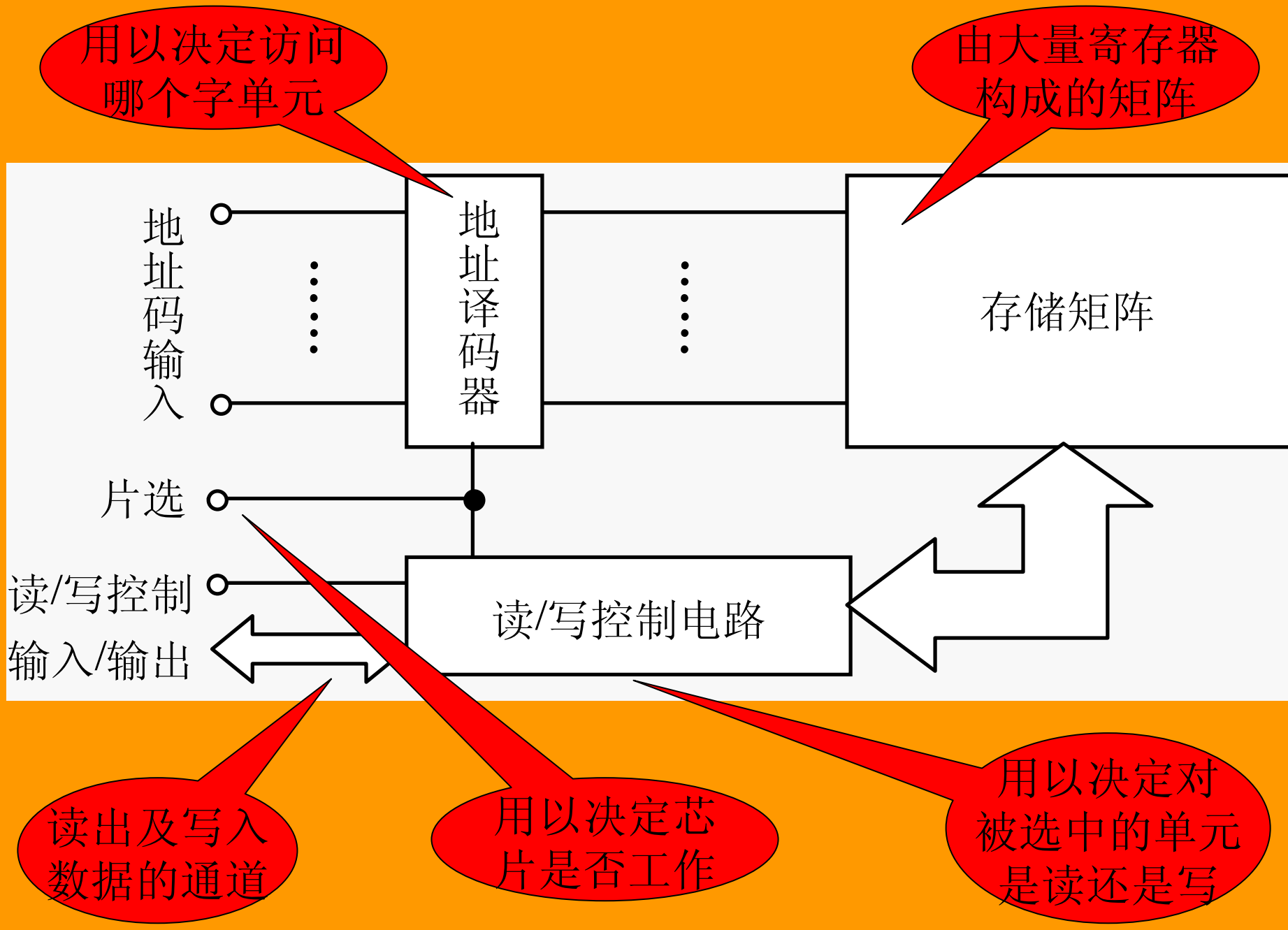


用ROM存储字符Z

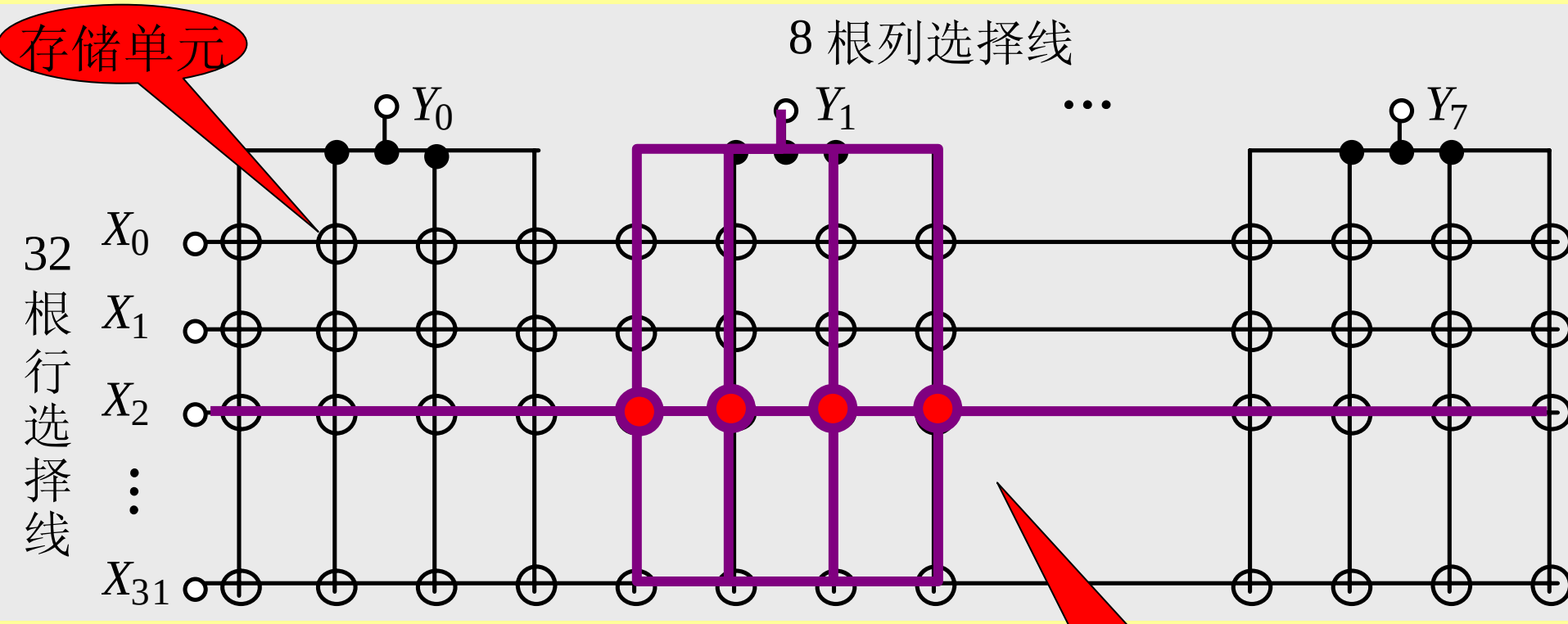
2　随机存取存储器

RAM的结构

RAM是由许许多多的基本寄存器组合起来构成的大规模集成电路。RAM中的每个寄存器称为一个字，寄存器中的每一位称为一个存储单元。**寄存器的个数（字数）与寄存器中存储单元个数（位数）的乘积，叫做RAM的容量**。按照RAM中寄存器位数的不同，RAM有多字1位和多字多位两种结构形式。在多字1位结构中，每个寄存器都只有1位，例如一个容量为1024×1位的RAM，就是一个有1024个1位寄存器的RAM。多字多位结构中，每个寄存器都有多位，例如一个容量为256×4位的RAM，就是一个有256个4位寄存器的RAM。


用以决定访问
哪个字单元
由大量寄存器
构成的矩阵
地址码输入
地址译码器
存储矩阵
片选
读/写控制
读/写控制电路
输入/输出
读出及写入
数据的通道
用以决定芯
片是否工作
用以决定对
被选中的单元
是读还是写

容量为256×4 RAM的存储矩阵

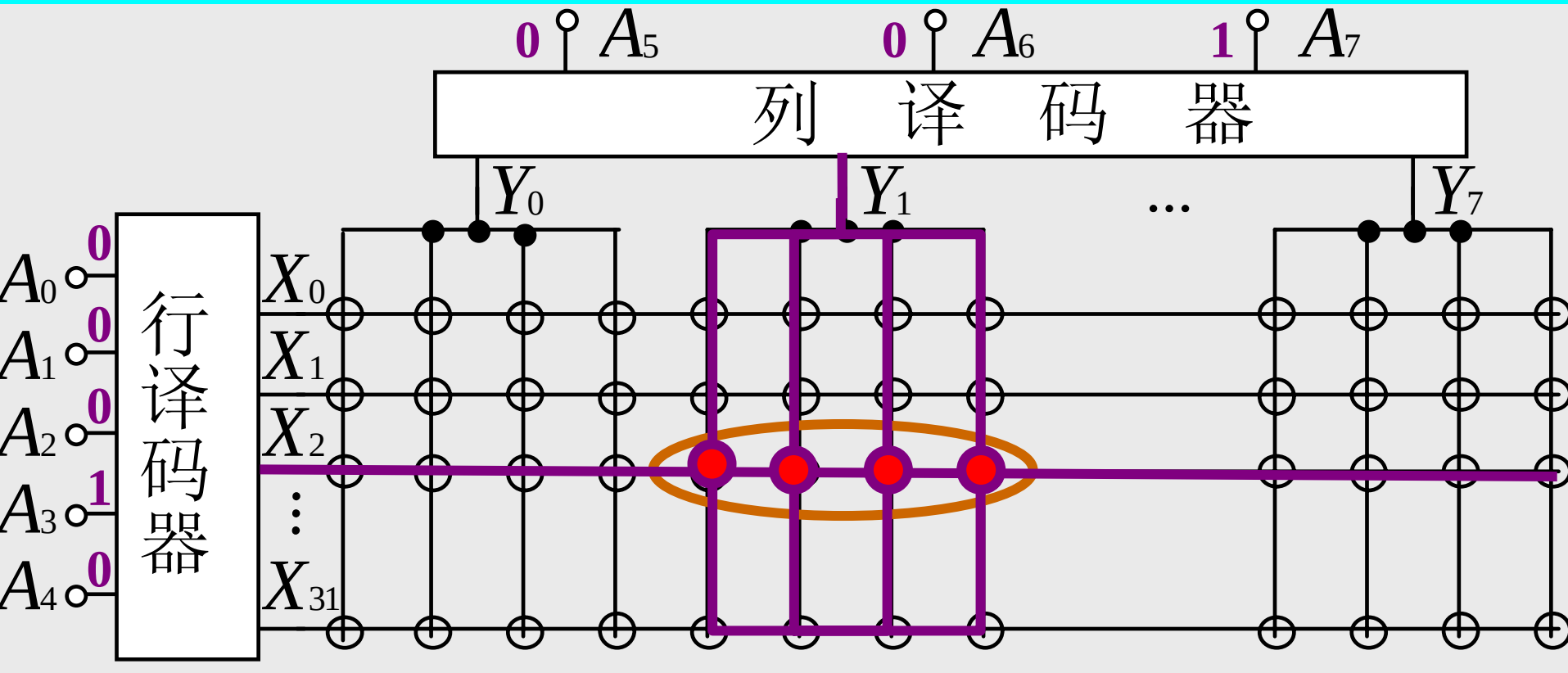


每根行选择线选择一行

每根列选择线选择一个字列

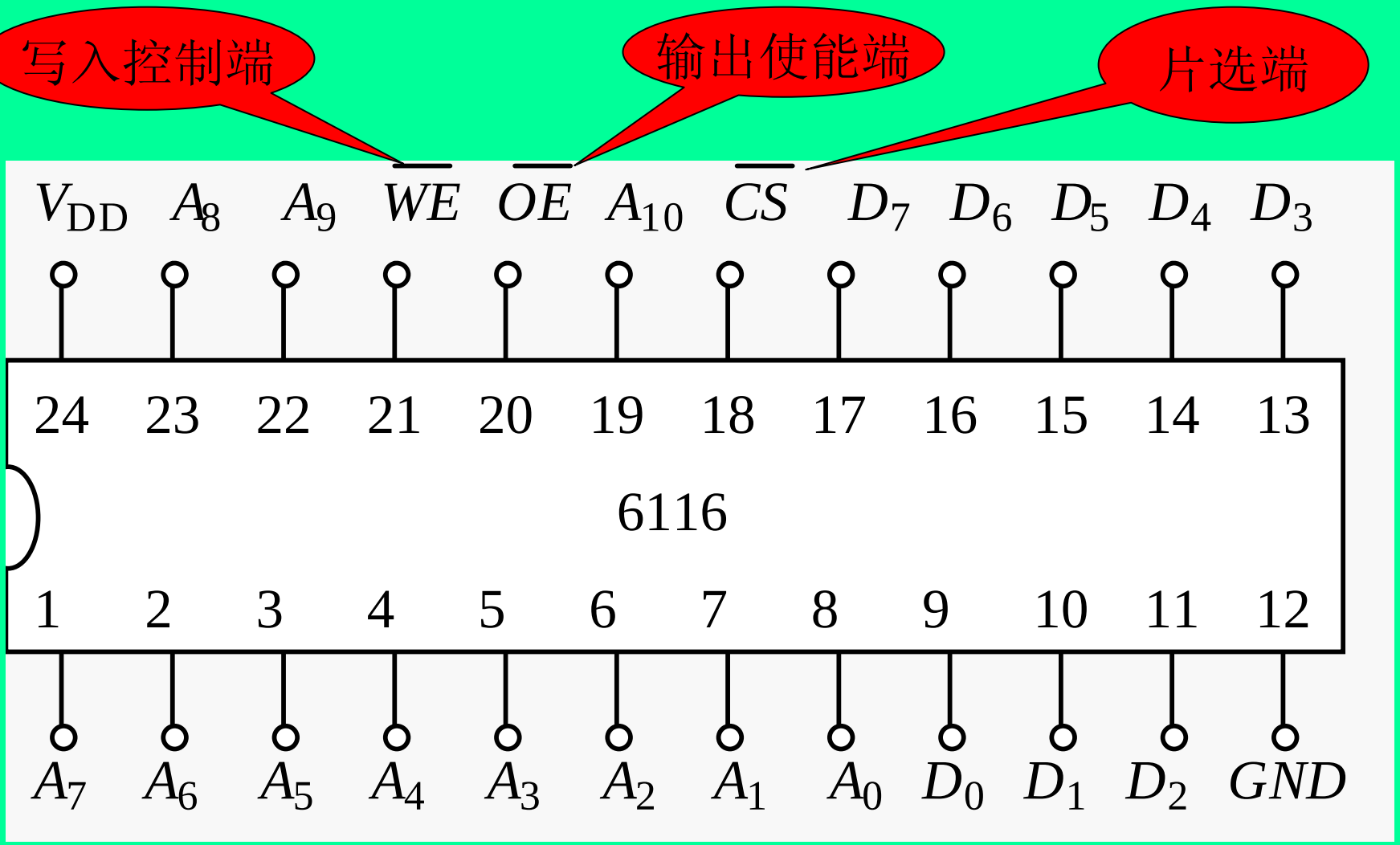
$Y_1=1$，$X_2=1$，位于X_2和Y_1交叉处的字单元可以进行读出或写入操作，而其余任何字单元都不会被选中。

地址的选择通过地址译码器来实现。地址译码器由行译码器和列译码器组成。行、列译码器的输出即为行、列选择线，由它们共同确定欲选择的地址单元。



256×4 RAM存储矩阵中，256个字需要8位地址码A_7～A_0。其中高3位A_7～A_5用于列译码输入，低5位A_4～A_0用于行译码输入。A_7～A_0=00100010时，Y_1=1、X_2=1，选中X_2和Y_1交叉的字单元。

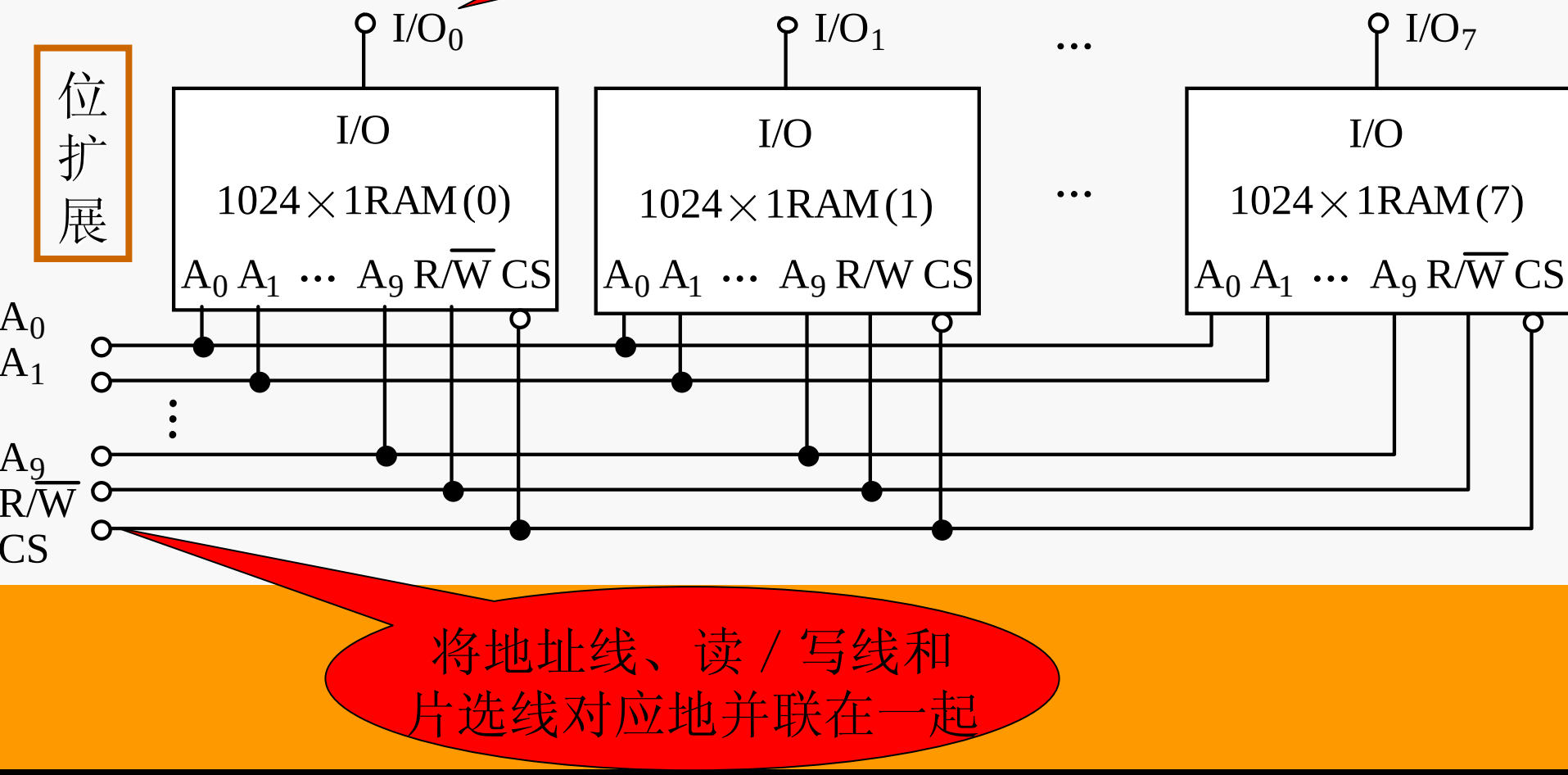
集成2kB×8位RAM6116



A_0～A_{10}：地址码输入端，D_0～D_7：数码输出端。

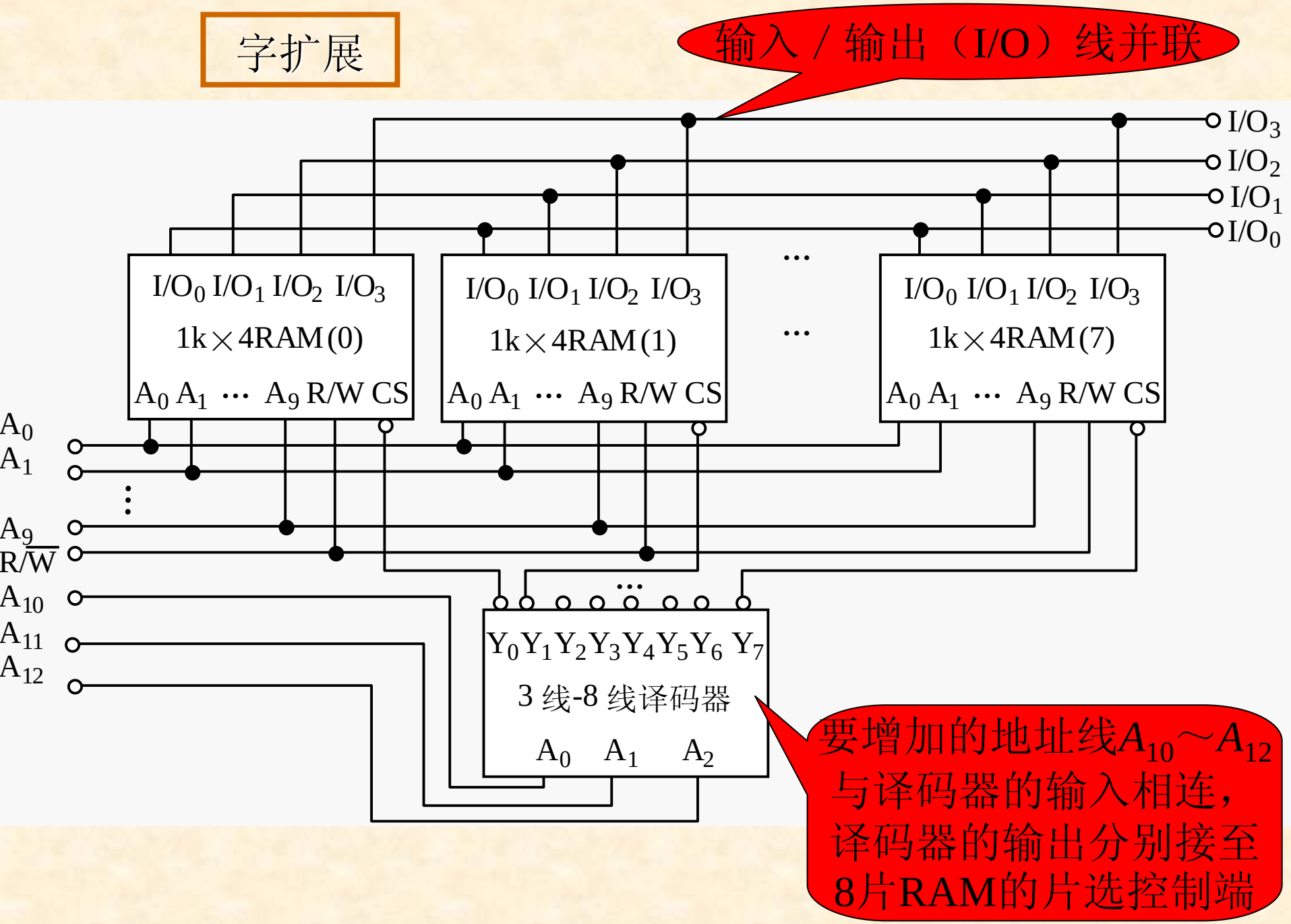
RAM容量的扩展

输入／输出（I/O）分开
使用作为字的各个位线



将地址线、读／写线和
片选线对应地并联在一起

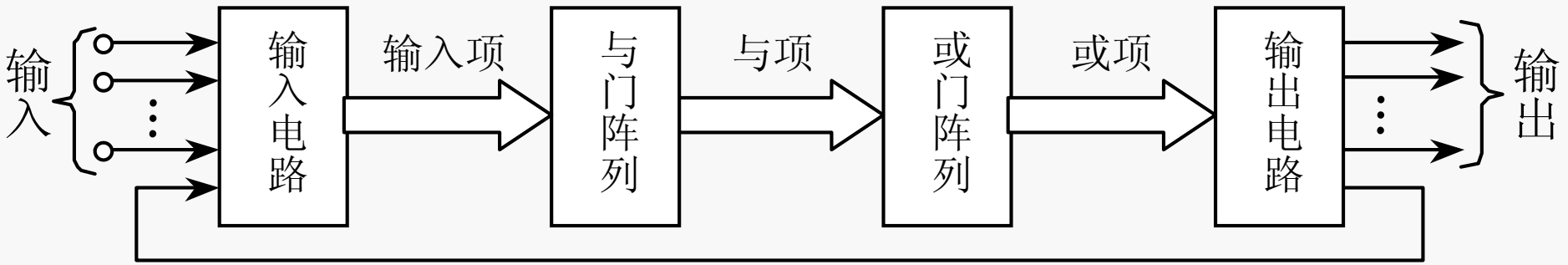
字扩展


输入／输出（I/O）线并联
I/O3
I/O2
I/O1
I/O0
I/O0 I/O1 I/O2 I/O3
1k×4RAM(0)
A0 A1 … A9 R/W CS
I/O0 I/O1 I/O2 I/O3
1k×4RAM(1)
A0 A1 … A9 R/W CS
I/O0 I/O1 I/O2 I/O3
1k×4RAM(7)
A0 A1 … A9 R/W CS
A0
A1
A9
R/W
A10
A11
A12
Y0 Y1 Y2 Y3 Y4 Y5 Y6 Y7
3 线-8 线译码器
A0 A1 A2
要增加的地址线A10～A12与译码器的输入相连，译码器的输出分别接至8片RAM的片选控制端

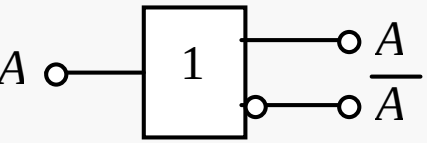
3　可编程逻辑器件

PLD的基本结构

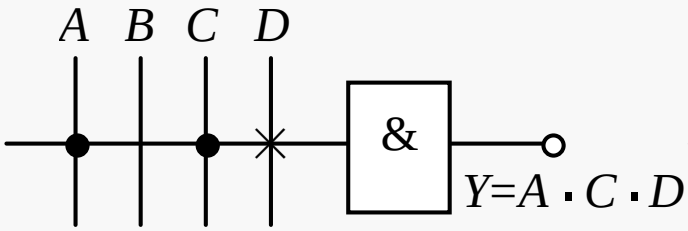
PLD的基本结构



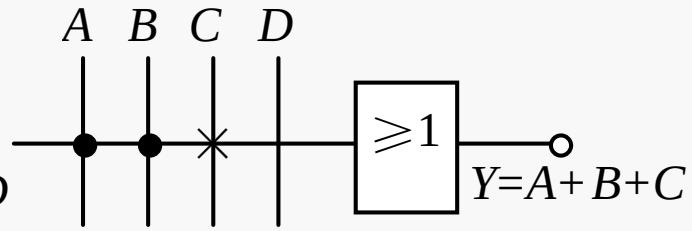
门电路的简化画法



(a)　缓冲器画法



(b)　与门画法



(c)　或门画法

PLD分类

分类	与阵列	或阵列	输出电路
PROM	固定	可编程	固定
PLA	可编程	可编程	固定
PAL	可编程	固定	固定
GAL	可编程	固定	可组态

PLD应用

用PROM实现组合逻辑函数

例 用PROM实现下列一组函数

$$Y_1 = A\overline{B} + AB + ABC\overline{D} + ABCD$$

$$Y_2 = \overline{A}B + B\overline{C} + AC$$

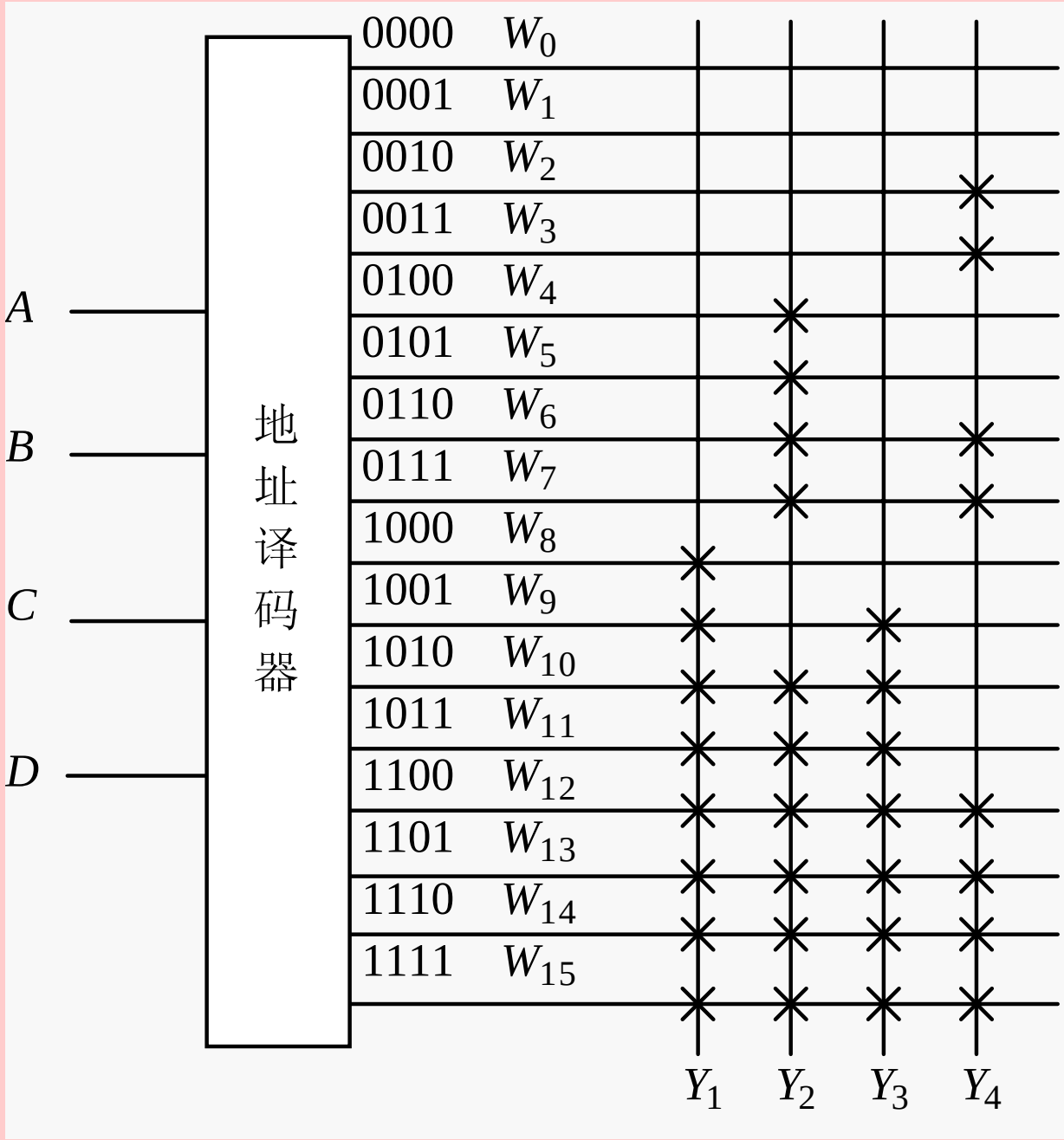
$$Y_3 = AB\overline{D} + A\overline{C}D + AC + AD$$

$$Y_4 = \overline{A}\,\overline{B}C + \overline{A}BC + AB\overline{C} + ABC$$

真值表

A	B	C	D	Y_1	Y_2	Y_3	Y_4
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	1
0	0	1	1	0	0	0	1
0	1	0	0	0	1	0	0
0	1	0	1	0	1	0	0
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	1
1	0	0	0	1	0	0	0
1	0	0	1	1	0	1	0
1	0	1	0	1	1	1	0
1	0	1	1	1	1	1	1
1	1	0	0	1	1	1	1
1	1	0	1	1	1	1	1
1	1	1	0	1	1	1	1
1	1	1	1	1	1	1	1

阵列图


A
B
C
D
地址译码器
0000 W_0
0001 W_1
0010 W_2
0011 W_3
0100 W_4
0101 W_5
0110 W_6
0111 W_7
1000 W_8
1001 W_9
1010 W_{10}
1011 W_{11}
1100 W_{12}
1101 W_{13}
1110 W_{14}
1111 W_{15}
Y_1 Y_2 Y_3 Y_4

用PLA实现组合逻辑函数

用PLA实现逻辑函数的基本原理是基于函数的最简与或表达式

例 用PLA实现下列一组函数

$$Y_1 = A\overline{B} + AB + ABC\overline{D} + ABCD$$

$$Y_2 = \overline{A}B + B\overline{C} + AC$$

$$Y_3 = AB\overline{D} + A\overline{C}D + AC + AD$$

$$Y_4 = \overline{A}\,\overline{B}C + \overline{A}BC + AB\overline{C} + ABC$$

化简

$$Y_1 = A$$

$$Y_2 = B + AC$$

$$Y_3 = AB + AC + AD$$

$$Y_4 = AB + \overline{A}C$$

阵列图