3.6 只读存储器（ROM）

分类
- 掩模 ROM
- 可编程 ROM（PROM — Programmable ROM）
- 可擦除可编程 ROM（EPROM — Erasable PROM）

说明：

掩模 ROM 生产过程中在掩模板控制下写入，内容固定，不能更改

PROM 内容可由用户编好后写入，一经写入不能更改

紫外光擦除（约二十分钟）

EPROM 存储数据可以更改，但改写麻烦，工作时只读

EEPROM 或 E^2PROM 电擦除（几十毫秒）

3.6.1 ROM 的结构和工作原理

一、ROM 的结构示意图

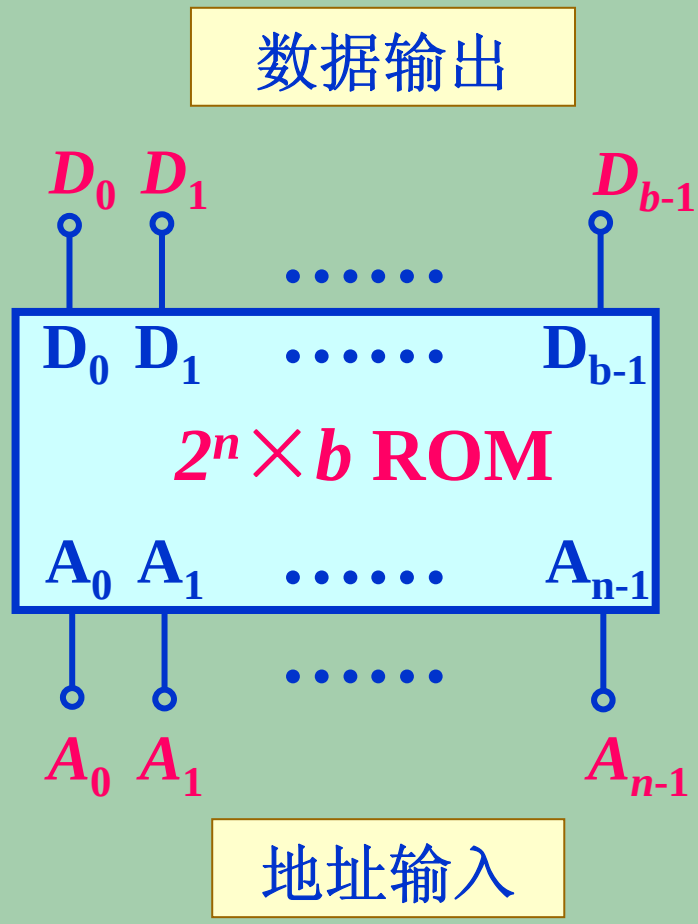
1. 基本结构

$A_{n-1} \sim A_0$ — n 位地址

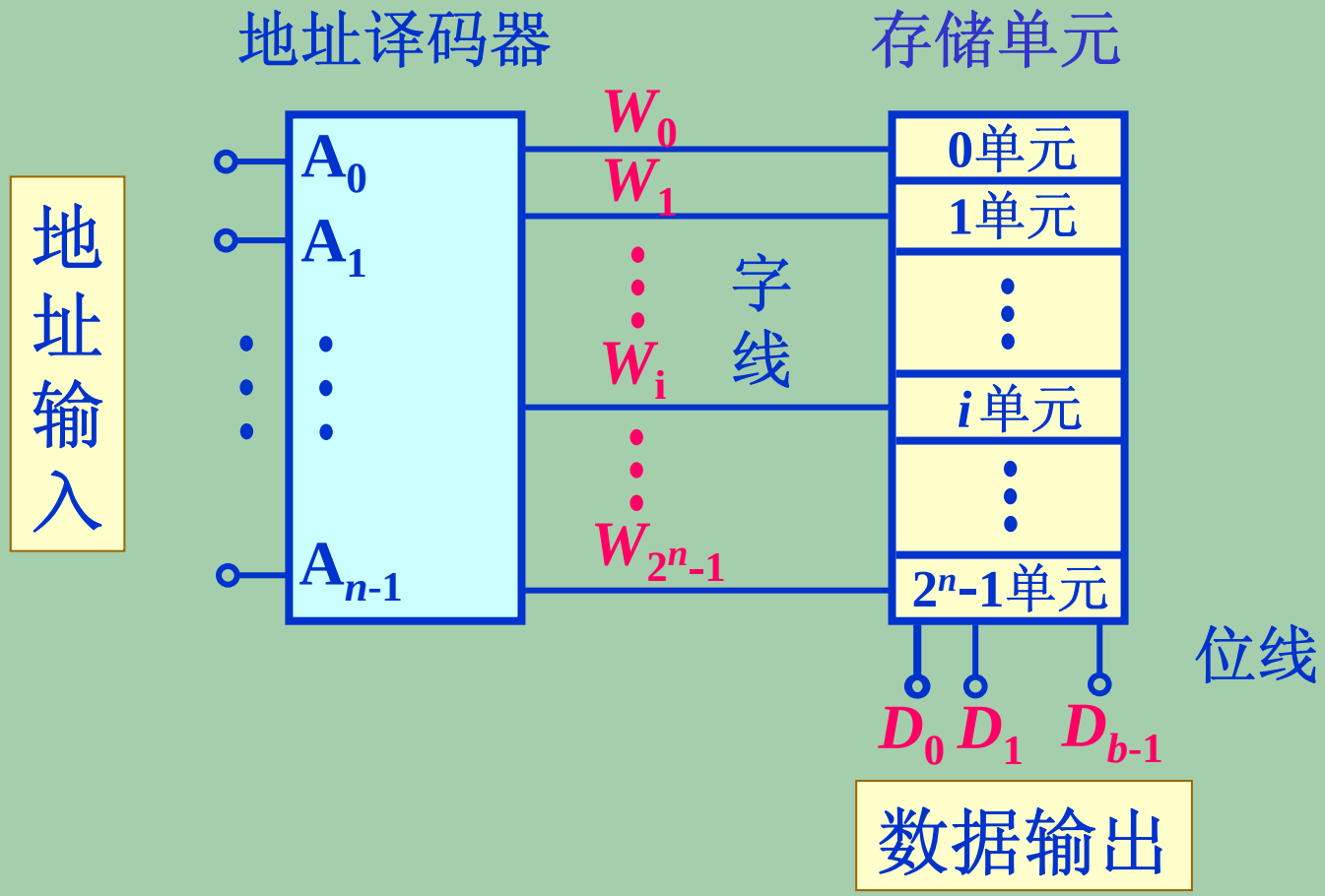
$D_{b-1} \sim D_0$ — b 位数据

最高位

最低位



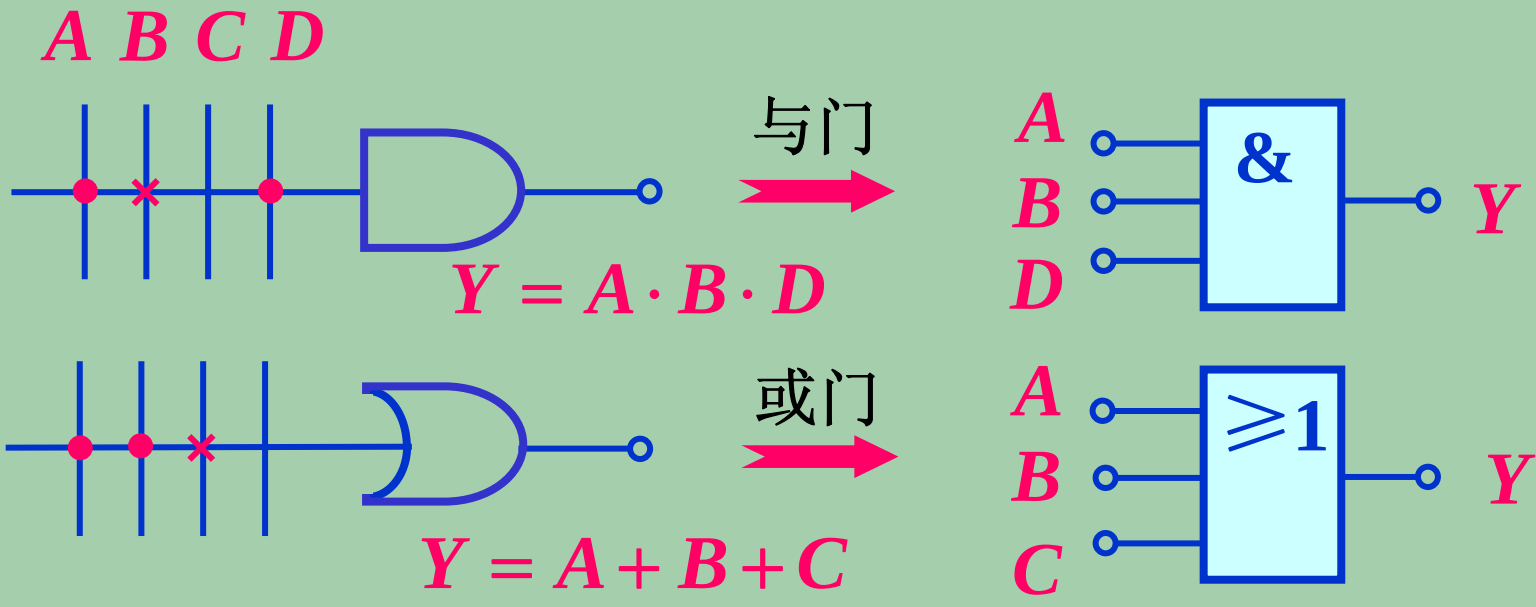
2. 内部结构示意图



ROM 存储容量 = 字线数 × 位线数 = $2^n \times b$（位）

3. 逻辑结构示意图

(1) 中、大规模集成电路中逻辑图简化画法的约定

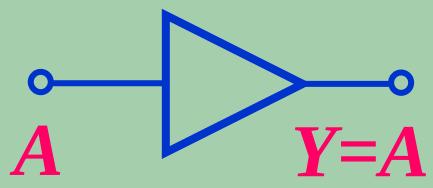


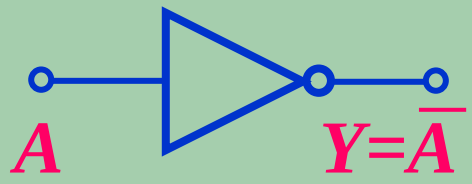
连上且为硬连接，不能通过编程改变

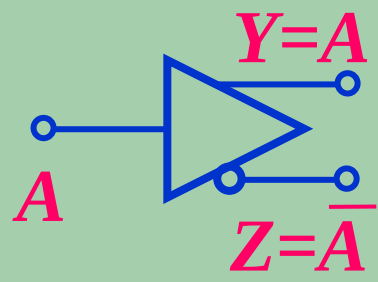
编程连接，可以通过编程将其断开

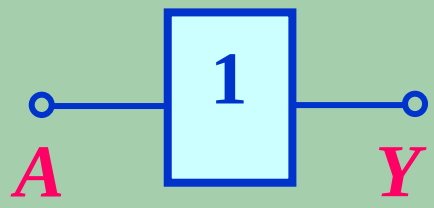
断开

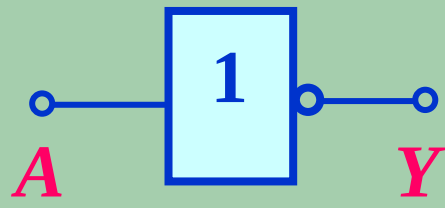
缓冲器

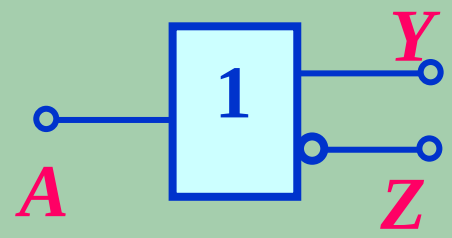










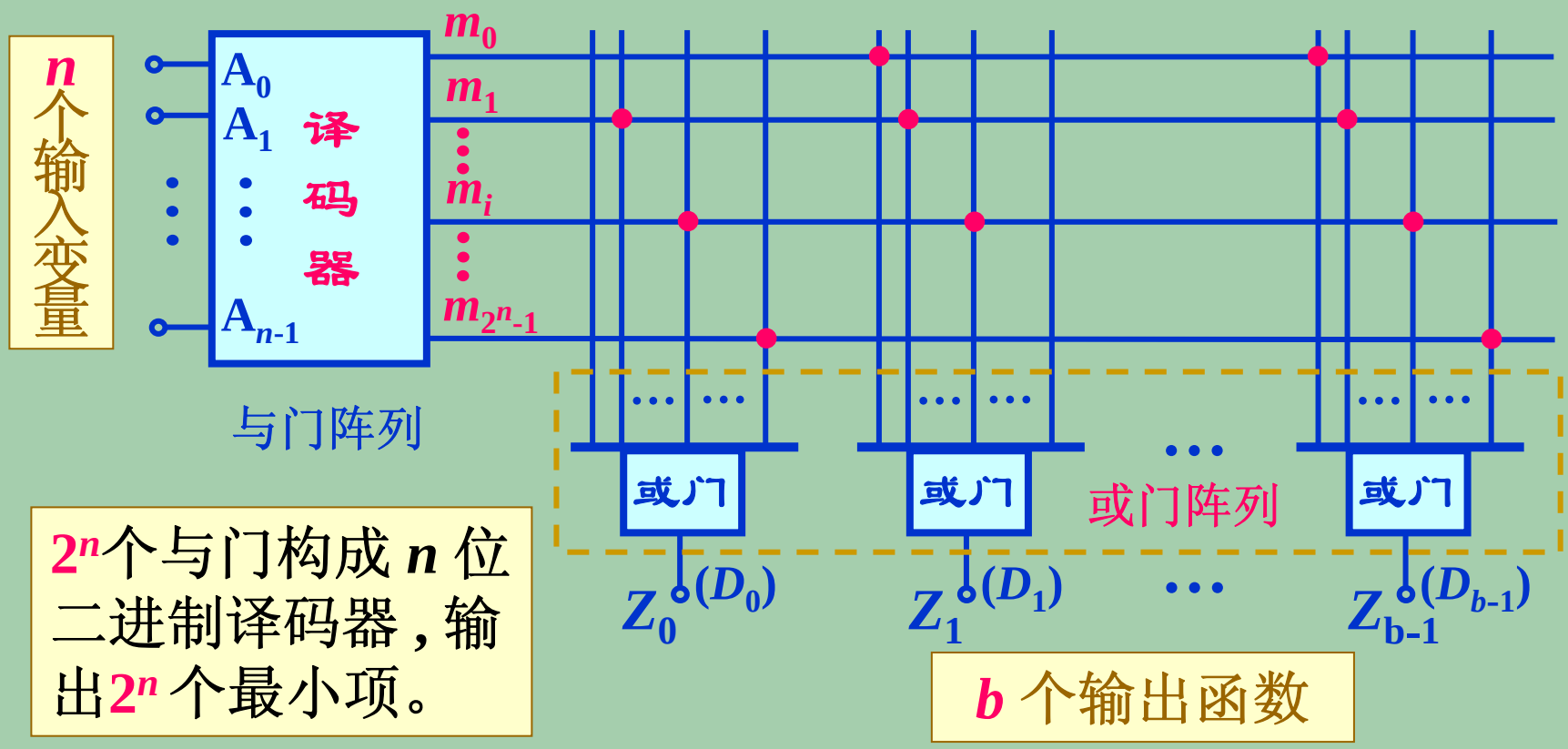


同相输出

反相输出

互补输出

(2) 逻辑结构示意图



2^n个与门构成 n 位二进制译码器，输出2^n 个最小项。

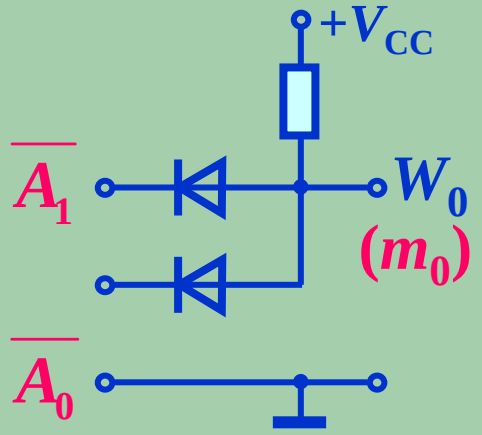
$$Z_0 = m_1 + m_i + m_{2^n-1} = D_0$$

$$Z_1 = m_0 + m_1 + m_i = D_1$$

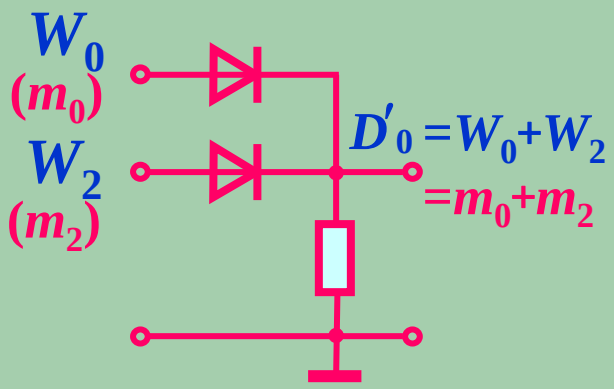
$$\vdots \quad Z_{b\text{-}1} = m_0 + m_1 + m_i + m_{2^n-1} = D_{b\text{-}1}$$

二、ROM 的基本工作原理

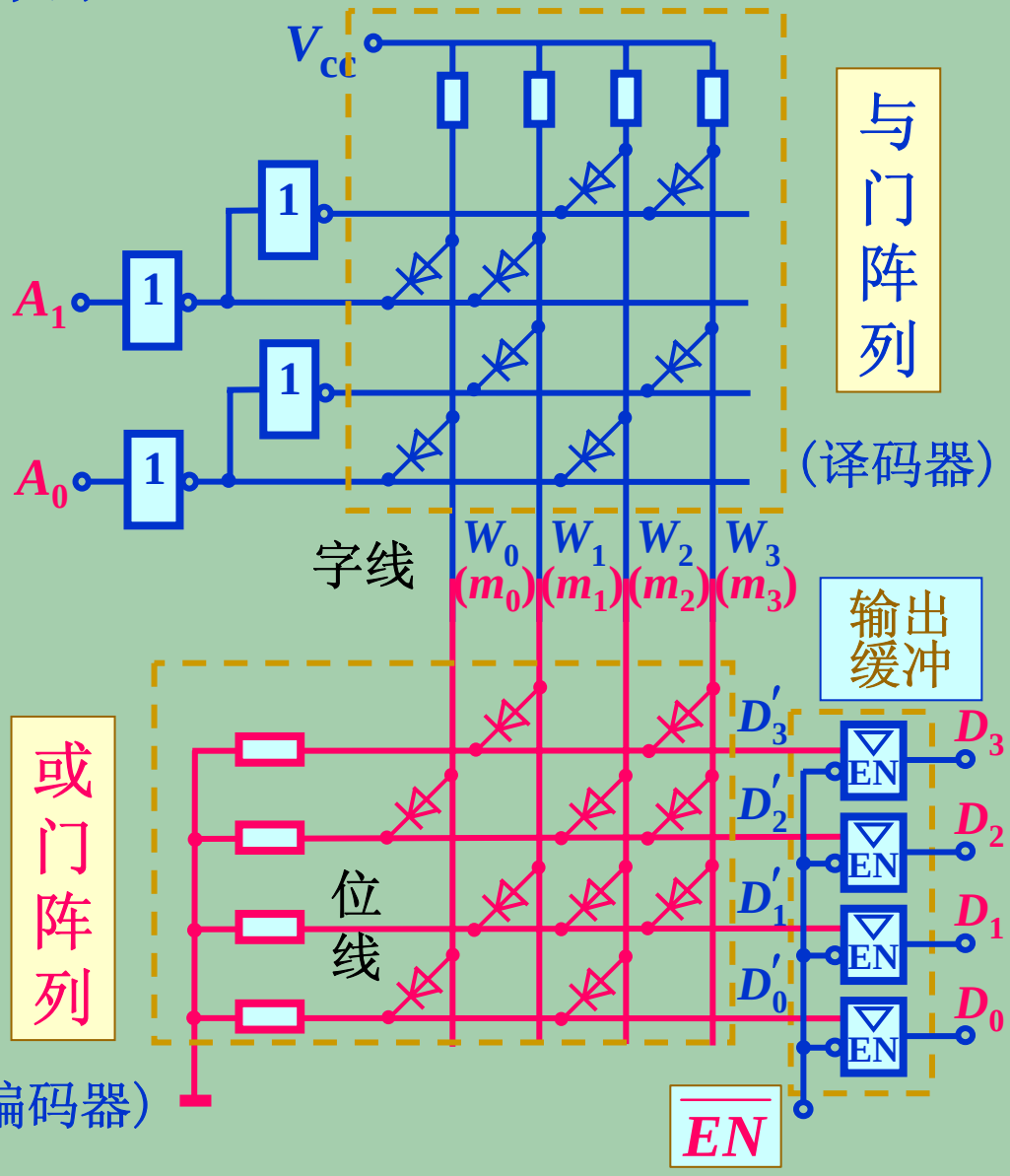
1. 电路组成



二极管与门



二极管或门



2. 工作原理

输出信号的逻辑表达式

字线：

$$W_0 = m_0 = \overline{A_1}\,\overline{A_0}$$

$$W_1 = m_1 = \overline{A_1} A_0$$

$$W_2 = m_2 = A_1 \overline{A_0}$$

$$W_3 = m_3 = A_1 A_0$$

位线：

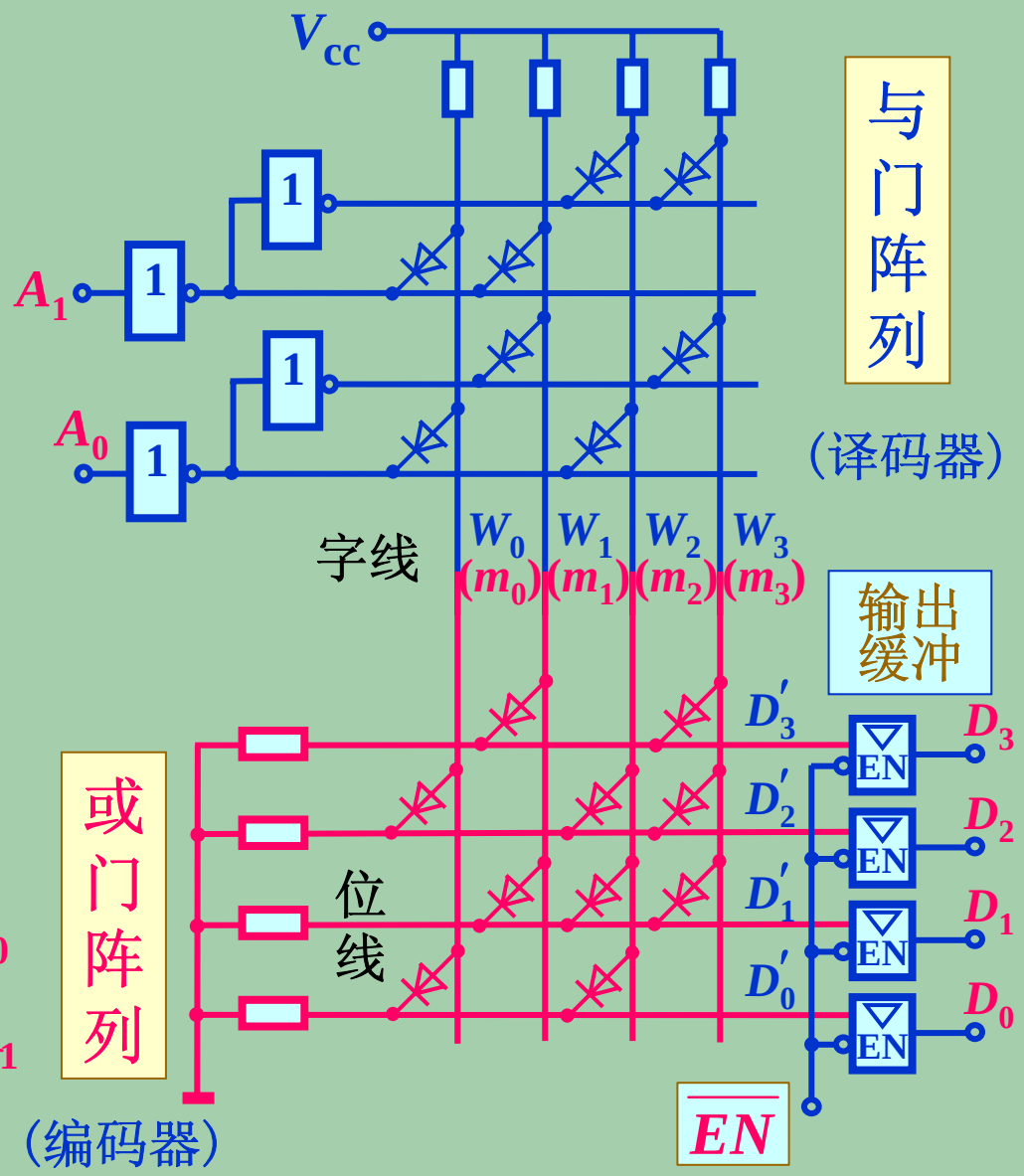
$$D_0 = W_0 + W_2 = m_0 + m_2$$

$$= \overline{A_1}\,\overline{A_0} + A_1 \overline{A_0} = \overline{A_0}$$

$$D_1 = W_1 + W_2 + W_3 = A_1 + A_0$$

$$D_2 = W_0 + W_2 + W_3 = \overline{A_0} + A_1$$

$$D_3 = W_1 + W_3 = A_0$$



3. 功能说明

(1) 存储器

输出信号的真值表

A_1	A_0	D_3	D_2	D_1	D_0
0	0	0	1	0	1
0	1	1	0	1	0
1	0	0	1	1	1
1	1	1	1	1	0

输入变量

输出函数

(2) 函数发生器

输入变量 A_1 A_0

输出函数 D_3 D_2 D_1 D_0

(3) 译码编码

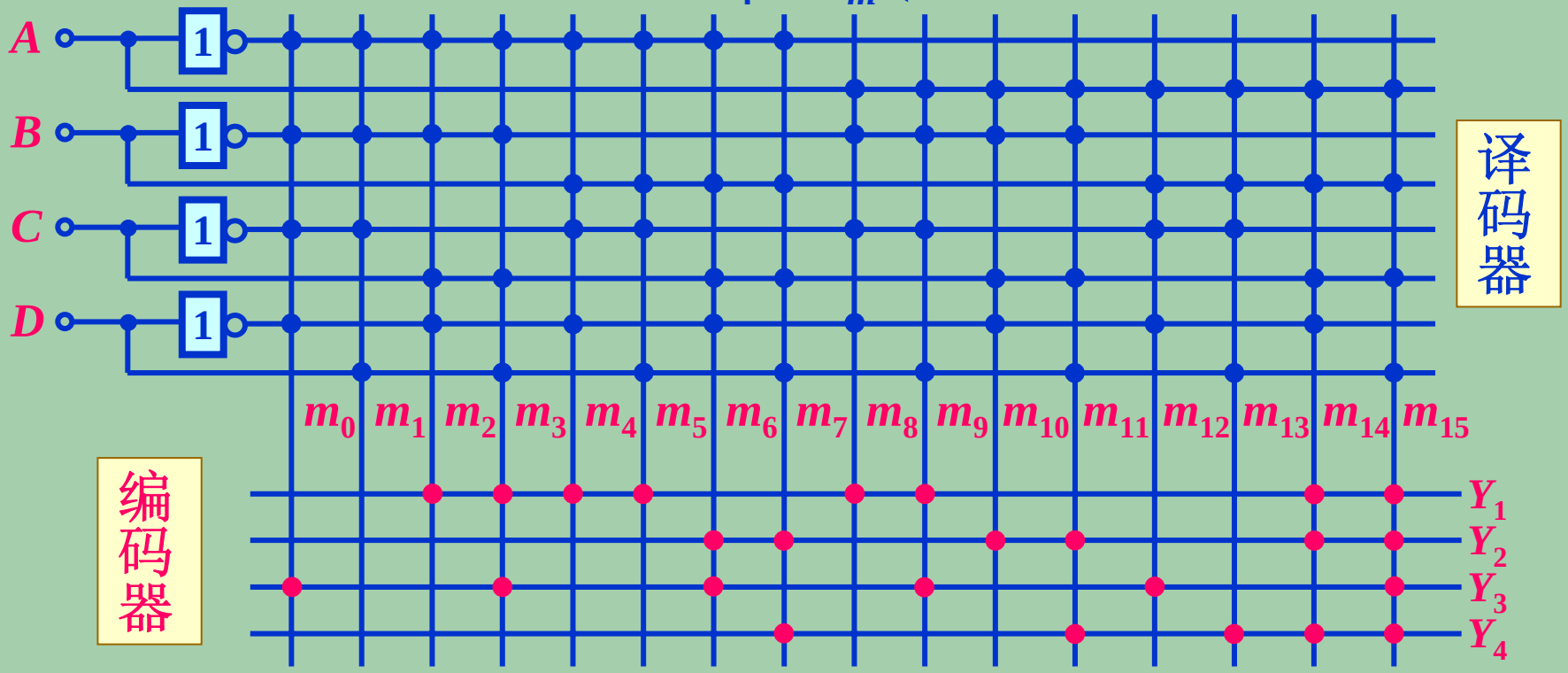
A_1	A_0	字线	编码
0	0	W_0	0 1 0 1
0	1	W_1	1 0 1 0
1	0	W_2	0 1 1 1
1	1	W_3	1 1 1 0

3.6.2 ROM 应用举例及容量扩展

一、ROM 应用举例

[例 3.6.2] 用 ROM 实现以下逻辑函数

$$
\begin{cases}
Y_1 = \Sigma_m(2,3,4,5,8,9,14,15) \\
Y_2 = \Sigma_m(6,7,10,11,14,15) \\
Y_3 = \Sigma_m(0,3,6,9,12,15) \\
Y_4 = \Sigma_m(7,11,13,14,15)
\end{cases}
$$



二、ROM 容量扩展

1. 存储容量 存储器存储数据的能力，为存储器含存储单元的总位数。

存储容量 = 字数× 位数　　字 — word　位 — bit

1k × 1 : 1024 个字　每个字 1 位　存储容量 1 k

1k × 4 : 1024 个字　每个字 4 位　存储容量 4 k

256 × 8 : 256 个字　每个字 8 位　存储容量 2 k

64 k × 16: 64 k 个字　每个字 16 位　存储容量 1024（1M）

2. 存储容量与地址位数的关系

存储容量 256 ×4　　$256 = 2^8$　　8 位地址　4 位数据输出

存储容量 8k×8　　$8k=8\times2^{10}=2^{13}$　13 位地址　8 位数据输出

3. 常用 EPROM

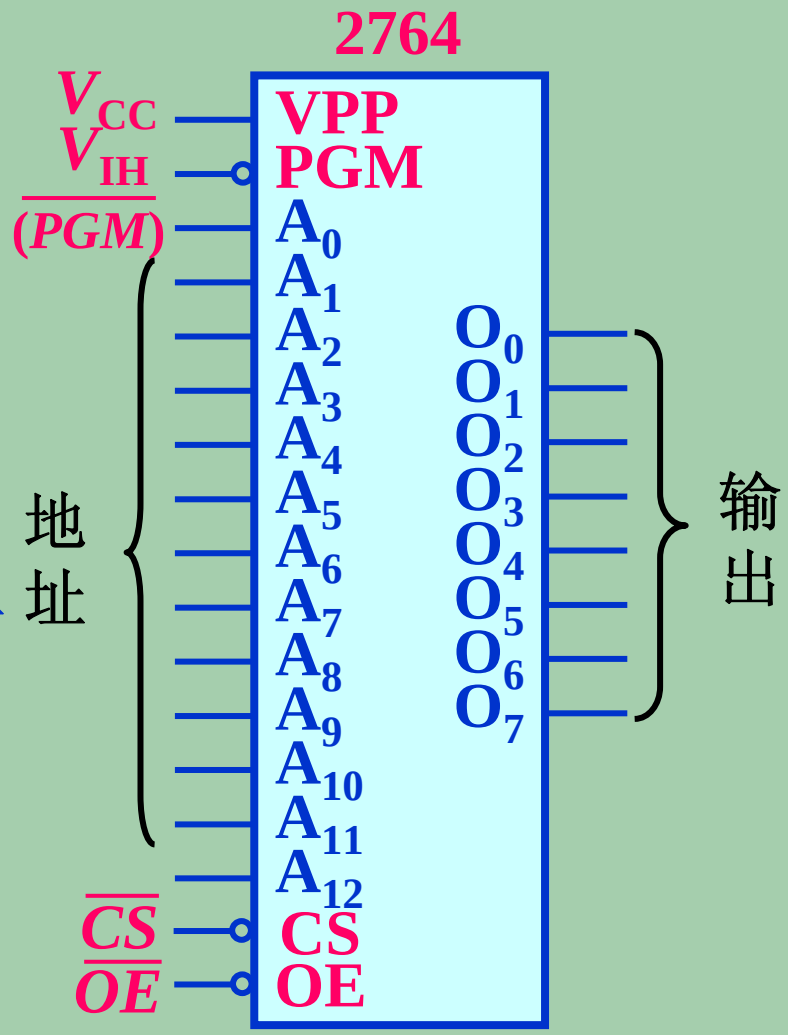
2764 : 8k×8 (64k)

13 位地址输入： $A_0 \sim A_{12}$

8 位数据输出： $O_0 \sim O_7$

输出使能端 $\overline{OE}$ { 0 使能；1 输出呈高阻

片选端 $\overline{CS}$ { 0 ROM 工作；1（$\overline{OE}$ 任意）ROM 不工作输出呈高阻



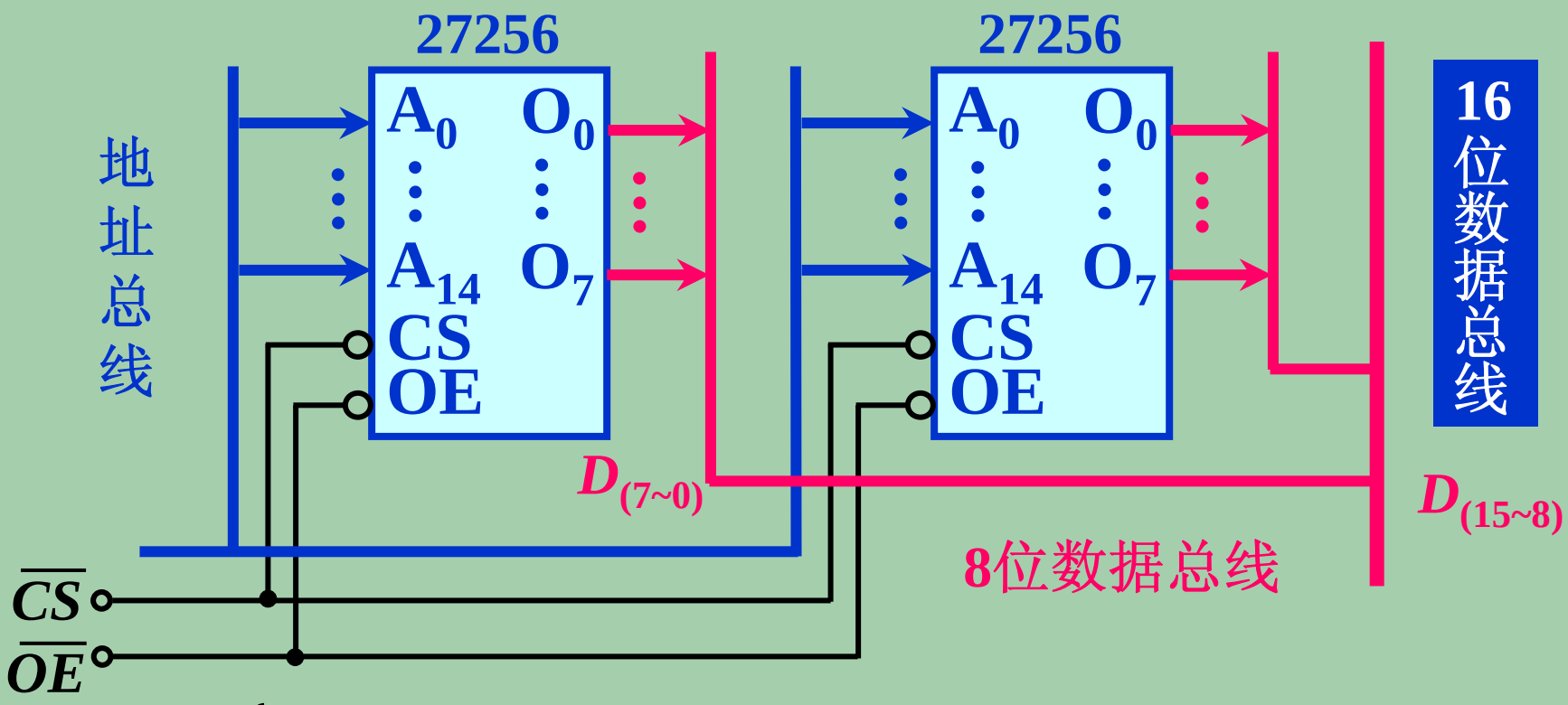
其他常用的 EPROM

27128 : 16k×8 (128k) 16k = 16×2^{10} = 2^{14}

27256 : 32k×8 (256k) 32k = 32×2^{10} = 2^{15}

4. ROM 容量的扩展

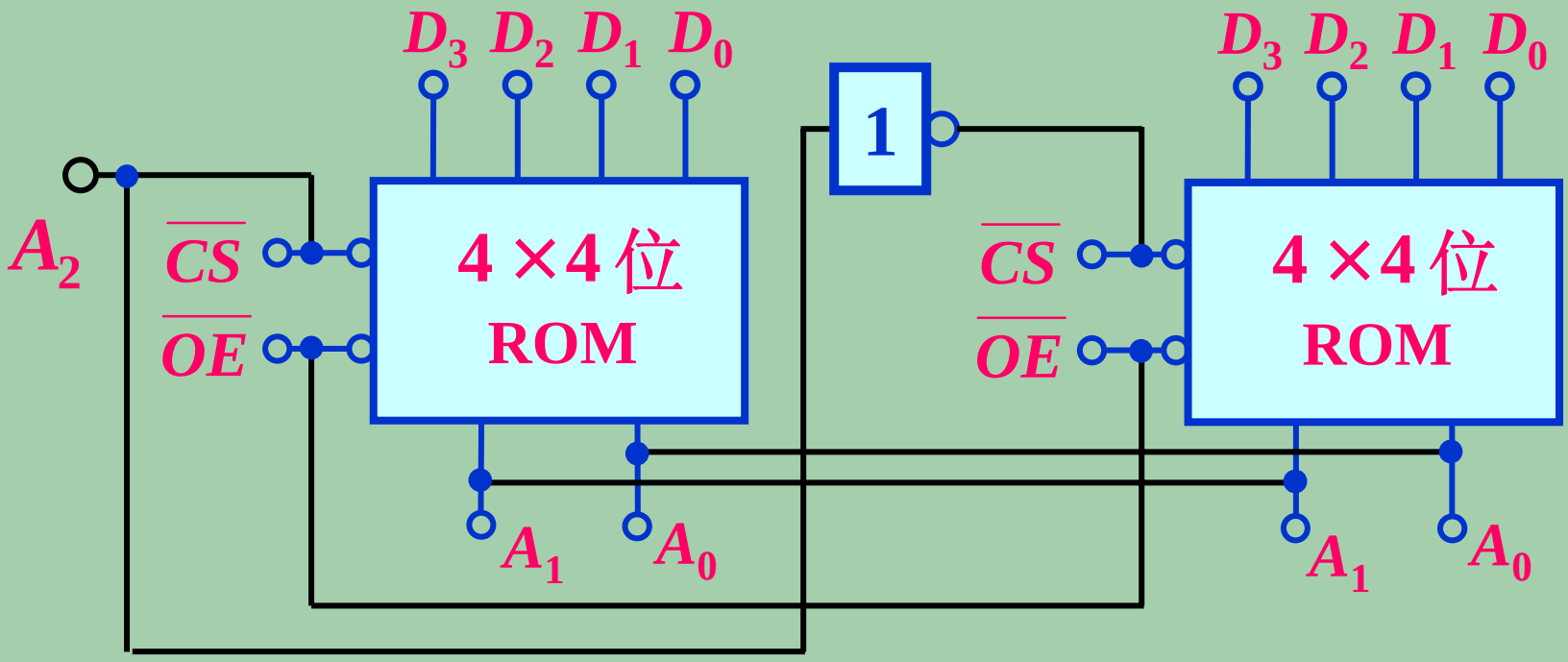
(1) 字长的扩展（位扩展）：　8 位 → 16 位



方法
- 地址线合并（共用）
- 输出使能端、片选端合并（共用）
- 数据输出端分为高 8 位和低 8 位

(2) 字线的扩展（地址码的扩展 — 字扩展）

两片 $4\times4\rightarrow8\times4$：增加一位地址 A_2



四片 $32\text{ k}\times8\rightarrow 4\times32\text{ k}\times8$：

15 位地址输入 $A_0 \sim A_{14}$　　增加两位地址 A_{15} A_{16}

经过 2 线- 4 线译码控制四个芯片的 $\overline{CS}$（电路略）