

第 10 章

半导体存储器及可编程逻辑器件

Memory and Programmable Logic Device

§ 10.1 半导体存储器概述

半导体存储器是存储信息的器件，由许多存储单元组成。

每个存储单元都有唯一的地址代码，能存储一位（或一组）二进制信息，可以由触发器或电容构成。

10.1.1 半导体存储器分类

按存取方式 { **SAM** (Sequential Access Memory, 顺序存储器)
RAM (Random Access Memory, 随机存储器)
ROM (Read Only Memory, 只读存储器)

按基本单元电路 { **Bipolar** { 工作速度快、功耗大、价格较高
主要用于对速度要求较高的场合，
如数字电子计算机中的高速缓存。
MOS { 集成度高、功耗小、工艺简单、价格低
主要用于大容量存储系统中，如数字
电子计算机中的主存储器（内存）。

10.1.2 存储器的技术参数

{	存储周期	存储器的性能基本上取决于从存储器读出信息和把信息写入存储器的速率。
		把连续两次读（写）操作间隔的最短时间称为存取周期。存取周期越短越好。
{	存储容量	

(1) 存储单元:

存储一位信息的单元

(2) 二进制数据单位

Bits

Bytes

Words

Bit（位）是二进制最小单元.

Byte (字节) 缩写 **B** , **1 Byte = 8 bits**

Word (存储字)

一组存储单元，表示某种类型的数据或信息

$2^{10} = 1024 \approx 1k$; $2^{20} \approx 1M$; $2^{30} \approx 1G$.

字长 n 位 8位/字, 16 位/字, 32 位/字

(3) 容量: 存储单元总数 字数 \times 位数

容量 $2k \times 16$ $\left\{ \begin{array}{l} 2k \text{ 字 } (2^{11} \text{ 个字}) \\ 16 \text{ 位 } (2 \text{ bytes}) \text{ 字长 } 16 \text{ 位} \end{array} \right.$

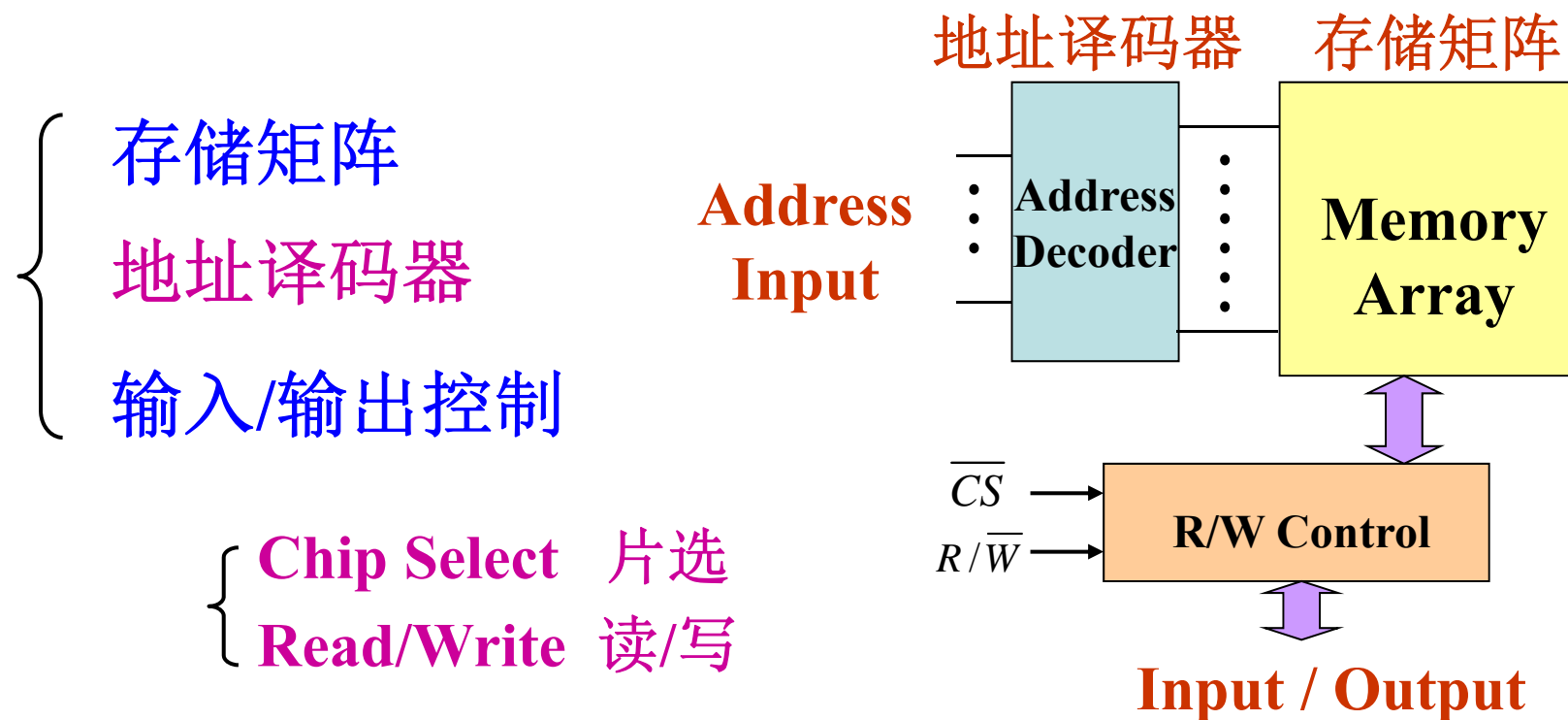
常用多少字节表示容量: **32k bits** 或 **4k bytes**

§ 10.2 随机存储器 RAM

优点：所有 RAM 具有读、写功能。

缺点：RAM 属于挥发性元件，断电将丢失全部数据

10.2.1 RAM 基本结构



1. 存储矩阵

RAM的主要部分

1024 个字排成
32×32 的矩阵

编号:

32行编号为

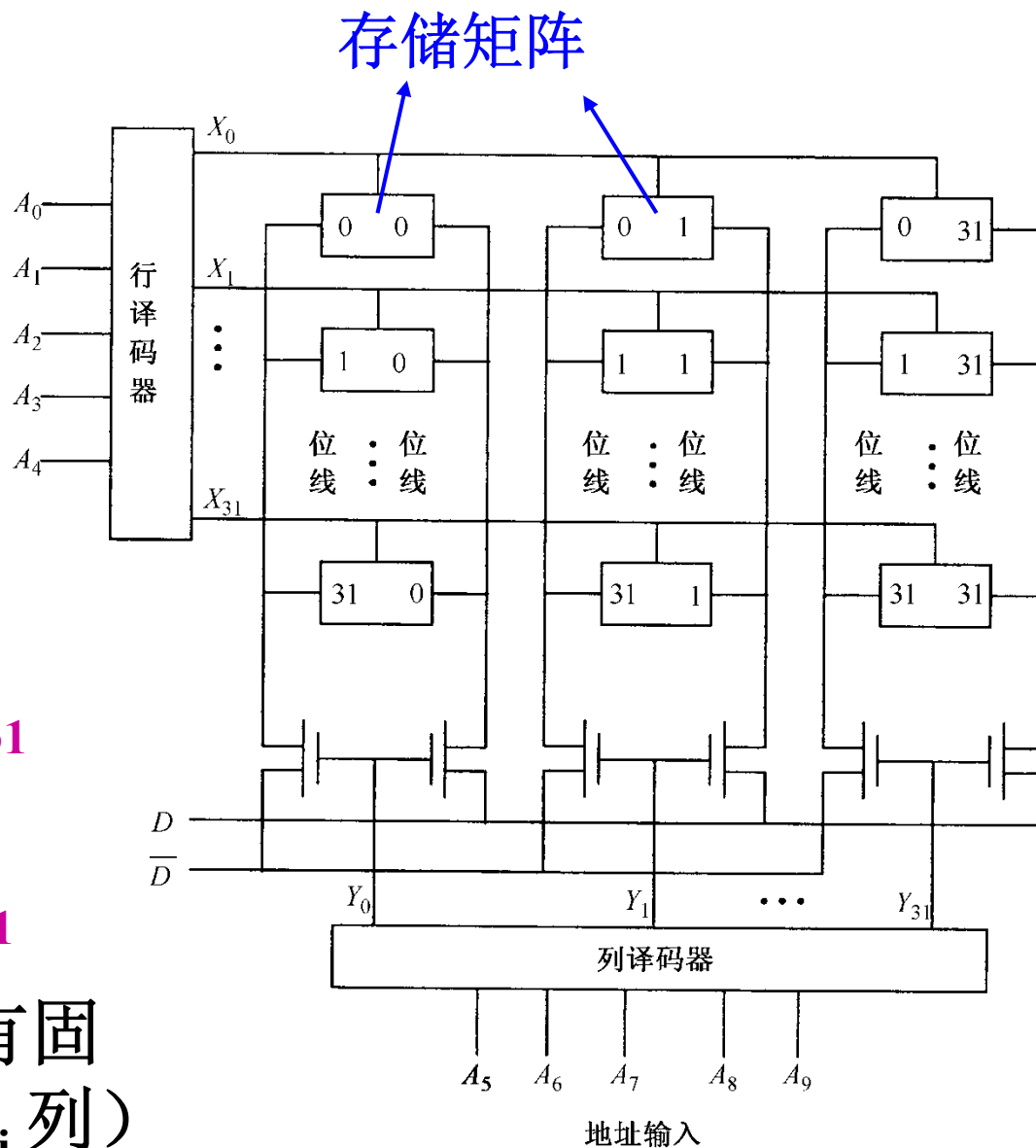
X_0, X_1, \dots, X_{31}

32列编号为

Y_0, Y_1, \dots, Y_{31}

每个存储单元都有固
定的编号 (X_i 行、 Y_j 列)

即地址 Address

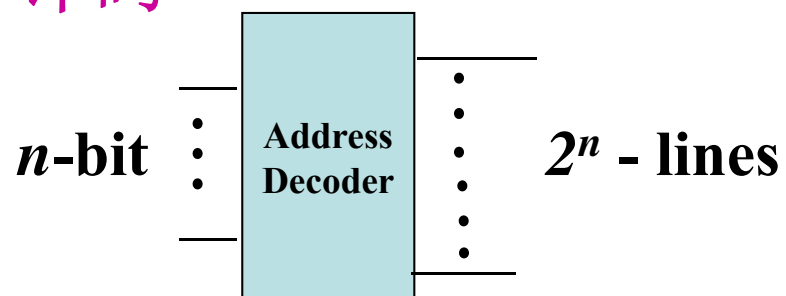


1k×1位存储矩阵及地址译码器

2. 地址译码器

{ 单译码
双译码

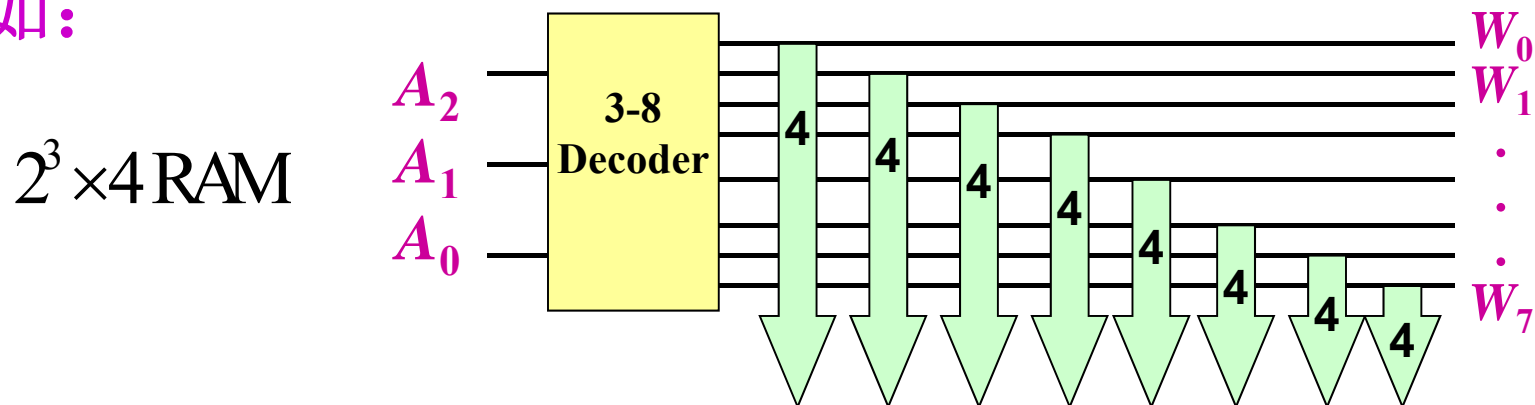
单译码



双译码

将地址所对应的
二进制数译成：行
选信号和列选信号，
从而选中该存储单
元。

如：



1024 字: 2^{10}

字线 (地址线): 10条

地址译码器
双译码结构:

行地址译码器:
5-32 译码器

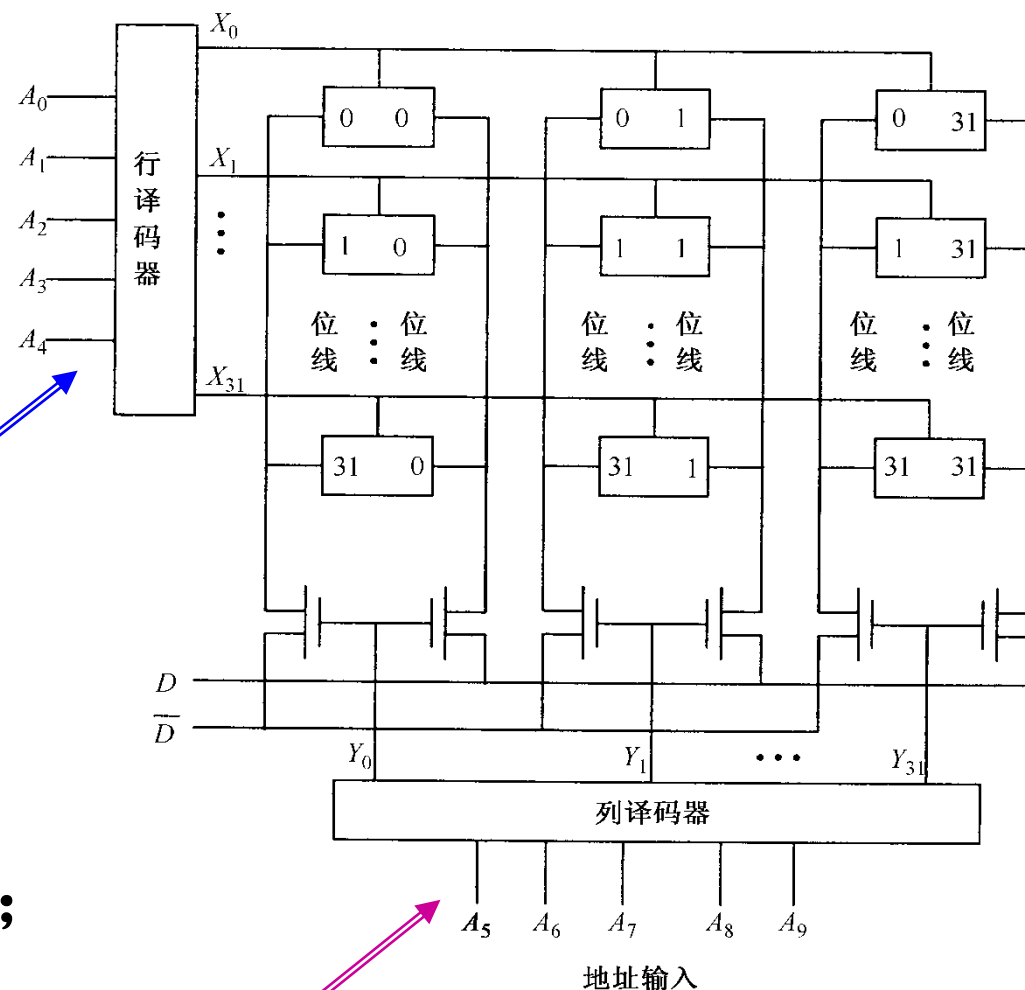
输入: A_0 、 A_1 、...、 A_4

输出: X_0 、 X_1 、...、 X_{31} ;

列地址译码器: 5-32 译码器

输入: A_5 、 A_6 、...、 A_9

输出: Y_0 、 Y_1 、...、 Y_{31} ;



地址:

$A_9A_8A_7A_6A_5 A_4A_3A_2A_1A_0$
 $= 00000\ 00001$

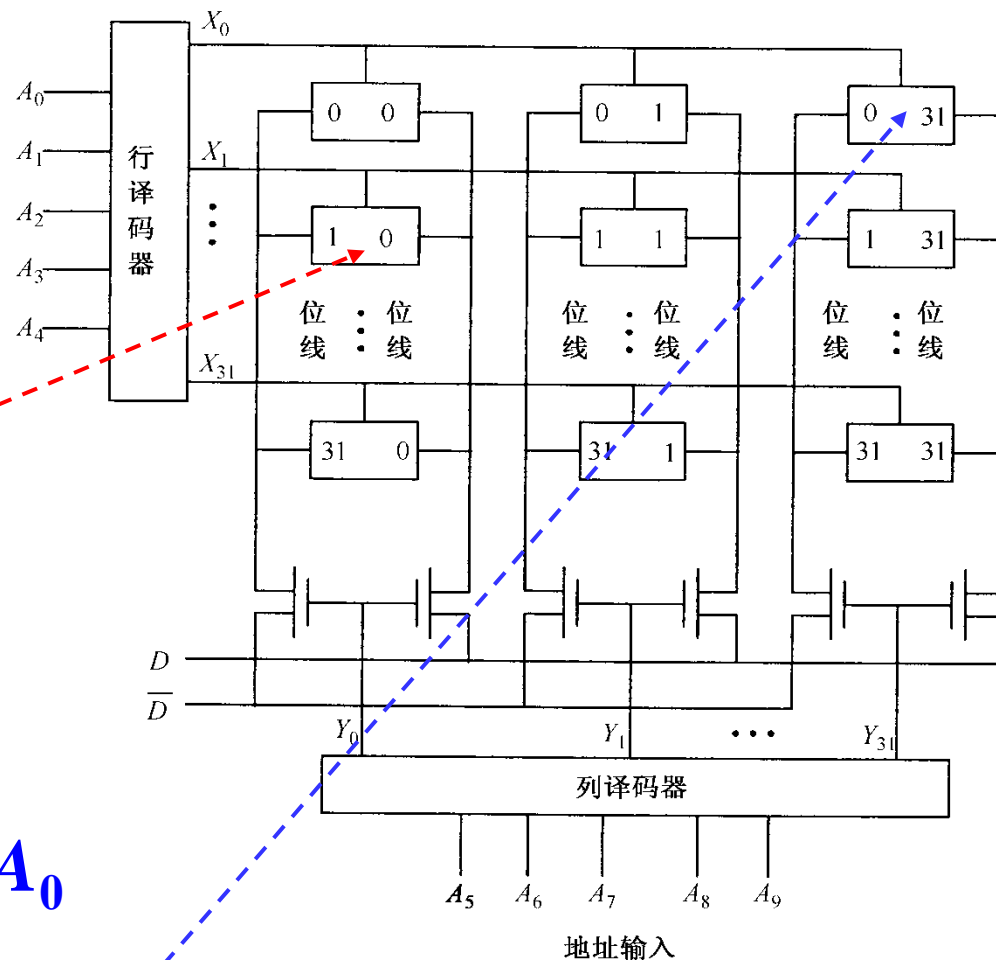
行线 $X_1 = 1$

列线 $Y_0 = 1$

地址:

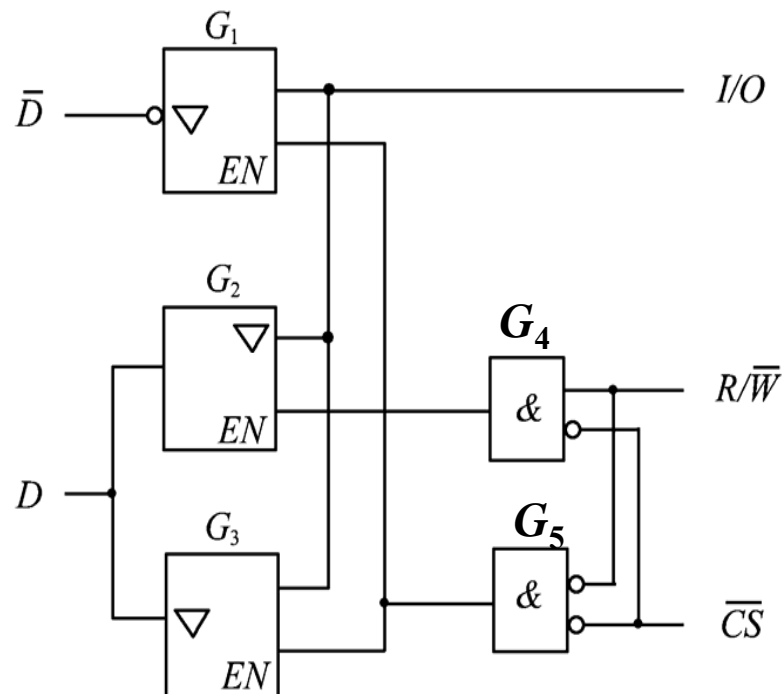
$A_9A_8A_7A_6A_5 A_4A_3A_2A_1A_0$
 $= 11111\ 00000$

行线 $X_0 = 1$ 、列线 $Y_{31} = 1$



3. 输入/输出控制电路 (Input/Output)

1位数据输入/输出控制电路



$\left\{ \begin{array}{l} \text{Read/Write } R/\overline{W} \\ \text{片选 } \overline{CS} \end{array} \right. \left\{ \begin{array}{l} R/\overline{W} = 1 \text{ 读} \\ R/\overline{W} = 0 \text{ 写} \end{array} \right.$
低有效

输入/输出 (I/O) 端数据线的条数=寄存器位数

容量 $2k \times 16$ RAM

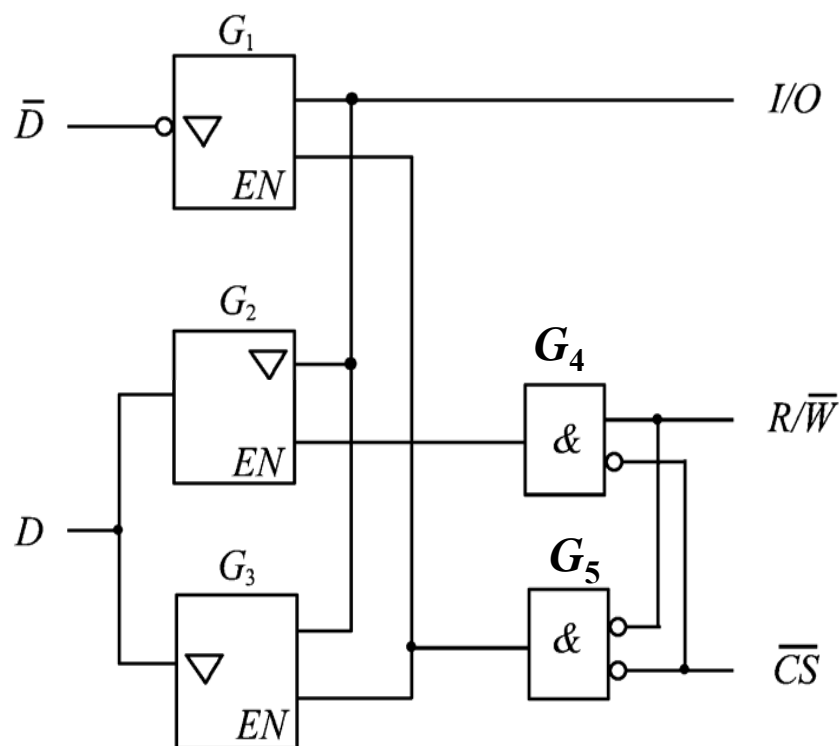
数据线 16 条 (位线)

D 和 \overline{D} 分别与存储矩阵的两条位线相连。

G_1 、 G_2 、 G_3 为三态门

当 $\overline{CS} = 1$ 时, G_4 、 G_5 输出为 0, RAM 芯片呈高阻态, 输入/输出 (I/O) 端与存储器内部完全隔离, 不进行任何操作;

1位数据输入/输出控制电路



当 $\bar{CS} = 0$ 时，芯片被选通

若 $R/\bar{W} = 1$, $G_4=1$, G_2 导通, $G_5=0$, G_1 和 G_3 呈高阻态截止;

被选中的字的存储单元数据出现在输入/输出 I/O 端, 存储器执行读操作;

若 $R/\bar{W} = 0$, $G_5=1$, G_1 和 G_3 导通, $G_4=0$, G_2 呈高阻态截止;

此时 I/O 端数据以互补形式(D, \bar{D})出现在内部数据线上, 并被存入所选中的字的存储单元中, 存储器执行写操作。

10.2.2 存储单元

工作原理 { 静态RAM (static) SRAM
动态RAM (dynamic) DRAM

器件分类 { Bipolar
MOS

{ 6管 NMOS 静态存储单元
双极晶体管存储单元
4管 动态MOS 存储单元
1管 动态MOS存储单元

10.2.3 芯片简介

例 1. 16×2 RAM

容量: 16×2 单元

字: $16 = 2^4$ 地址线: 4 ($A_3A_2A_1A_0$)

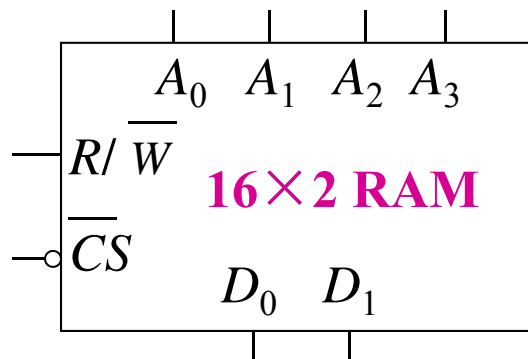
字长: 2-bit 数据线: 2 (D_1D_0)

地址

A_3, D_1 高位

读/写 控制

片选
低有效



数据 输入 / 输出

例 2. RAM2114

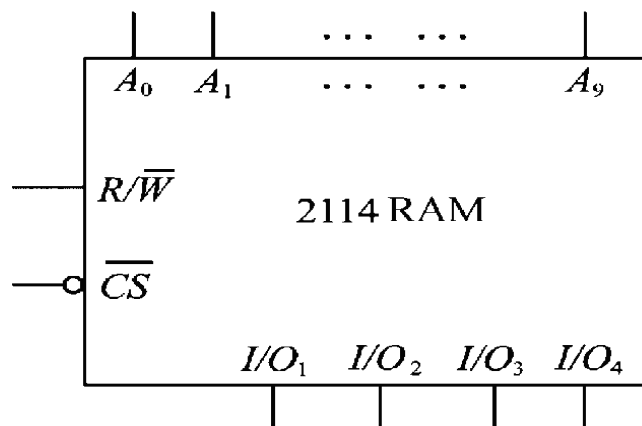
容量:

1024 字 \times 4 位

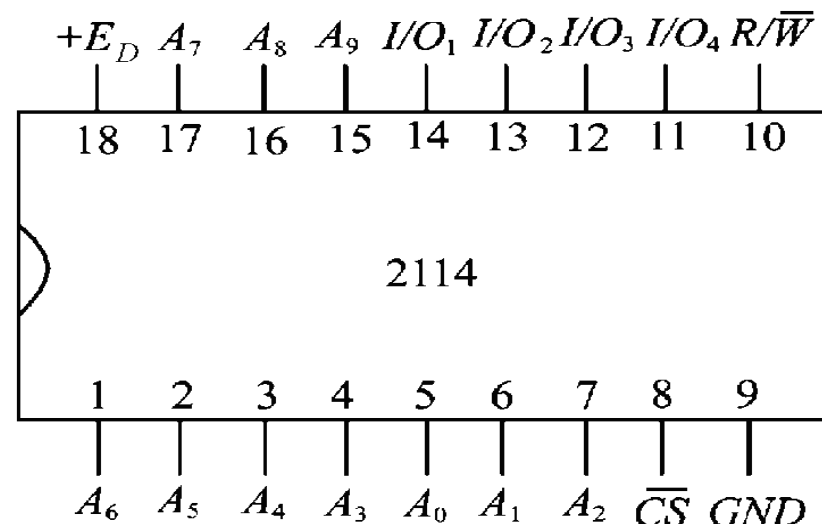
地址线: 10, $A_0 \sim A_9$,

数据线 (输入/输出): 4, $I/O_1 \sim I/O_4$

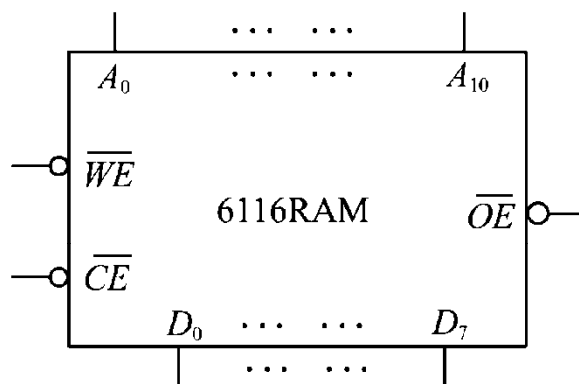
符号



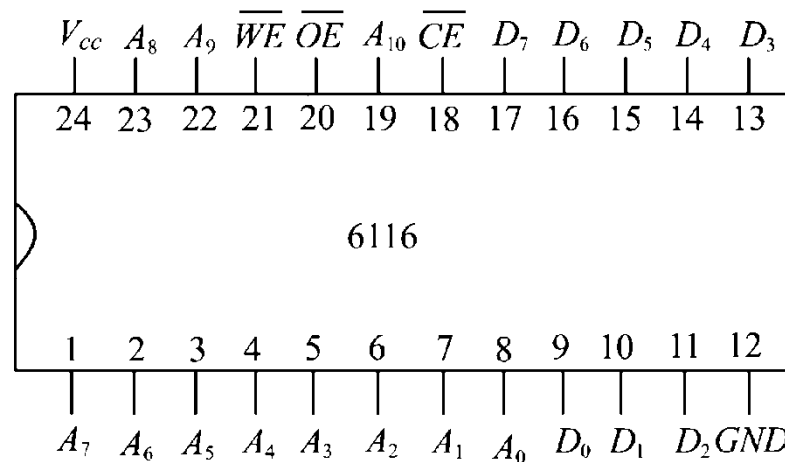
管脚图



例 3. RAM 6116



(a) 符号图



(b) 管脚图

字: 地址线: 11 2^{11} 字 (2k)

位: 8 位

容量: $2^{11} \times 8 = 2k \times 8 = 16k$ 单元 = 2k 字节

\overline{CE} : 片选, 低有效; \overline{OE} : 使能, 低有效;

\overline{WE} : 读/写 控制

10.2.4 RAM 扩展

当芯片容量不足时，需要扩展。

{ 位扩展
字扩展

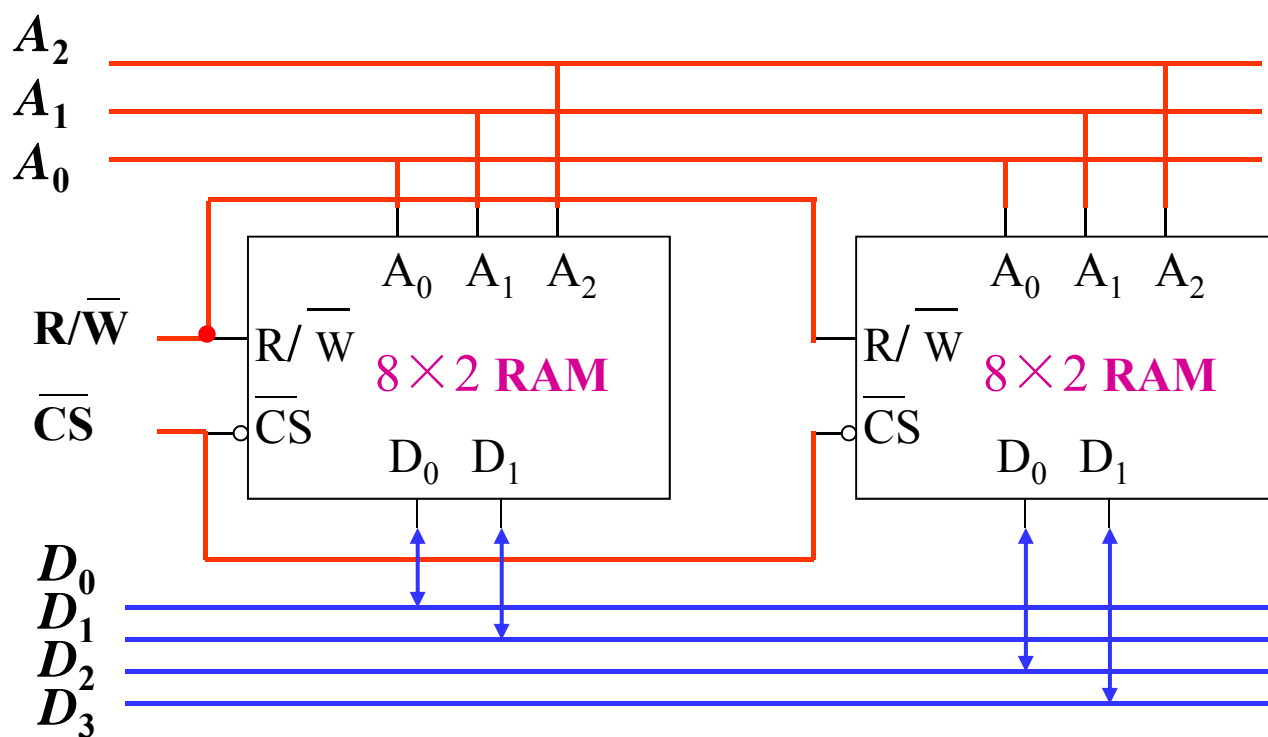
1. 位扩展

方法：同样的 RAM 芯片 并联

共用：{ 地址线
 R/\overline{W}
 \overline{CS}

例：将 8×2 RAM 扩展成 8×4 RAM

$$\frac{8 \times 4}{8 \times 2} = 2 \quad (8 \times 2 \text{ RAM})$$



两片RAM 同时工作, 地址范围不变, 位（字长）扩展.

2-bit \rightarrow 4-bit

2. 字扩展

增加地址线

用 \overline{CS} 扩展字

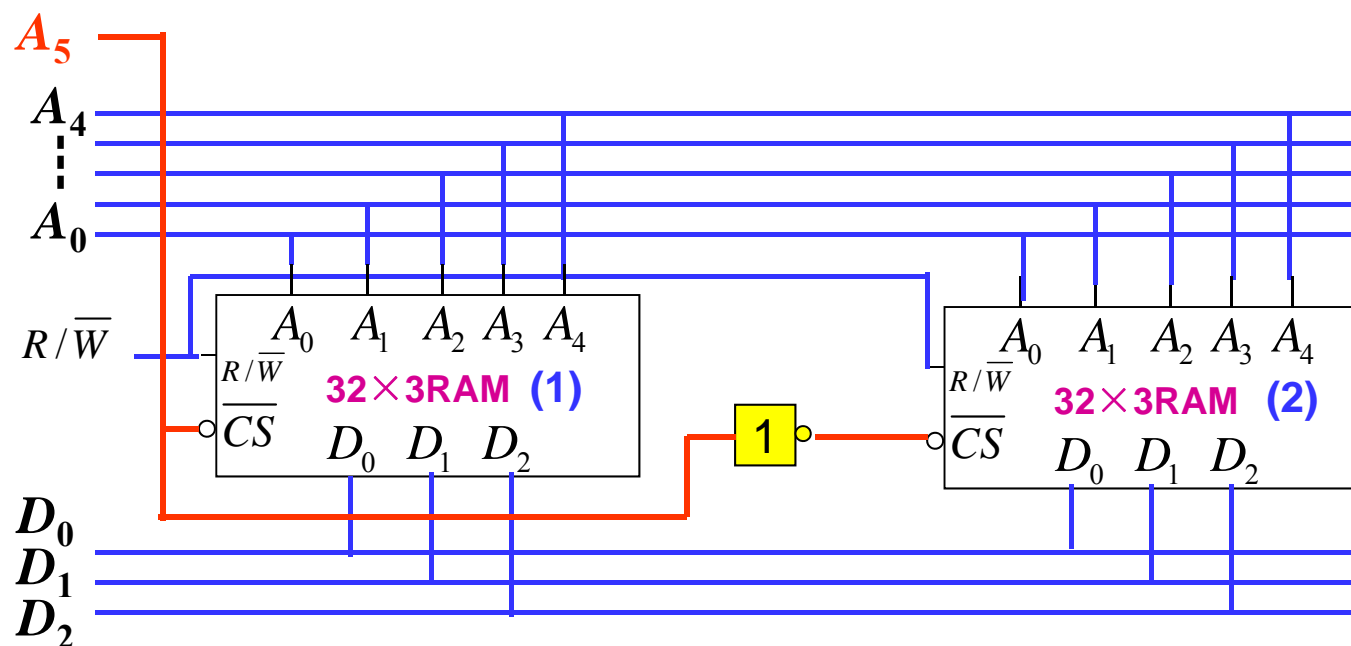
共用:

{ 原地址线
 R/\overline{W}
数据线

例 1. 将 32×3 RAM 扩展成 64×3 RAM.

$$\frac{64 \times 3}{32 \times 3} = 2 \text{ (} 32 \times 3 \text{ RAM)}$$

地址线 $5 \rightarrow 6$



$A_5=0$ (1) 工作, (2) 停止; 地址 000000-011111

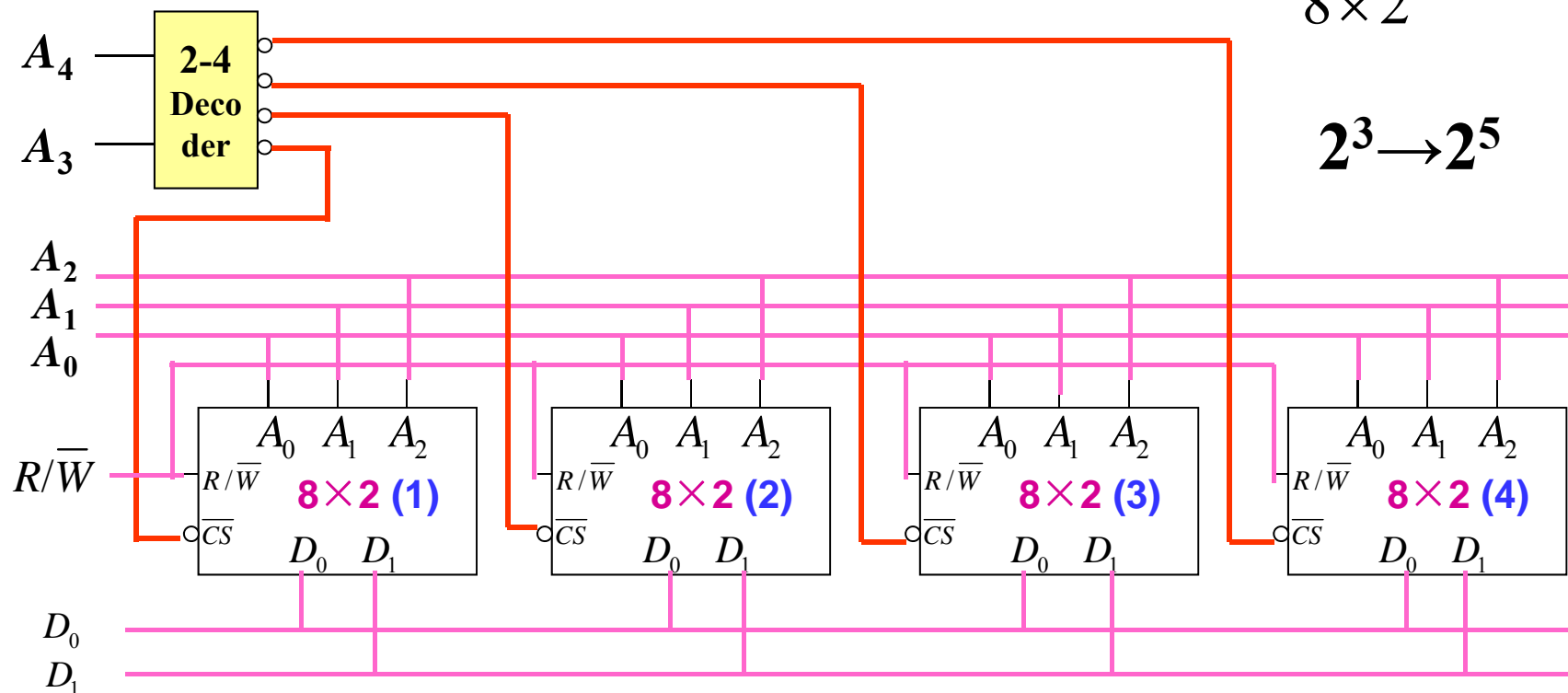
$A_5=1$ (1) 停止, (2) 工作; 地址 100000-111111

分时
工作

例 2. 将 8×2 RAM 扩展成 32×2 RAM.

$$\frac{32 \times 2}{8 \times 2} = 4$$

$$2^3 \rightarrow 2^5$$



地址范围:

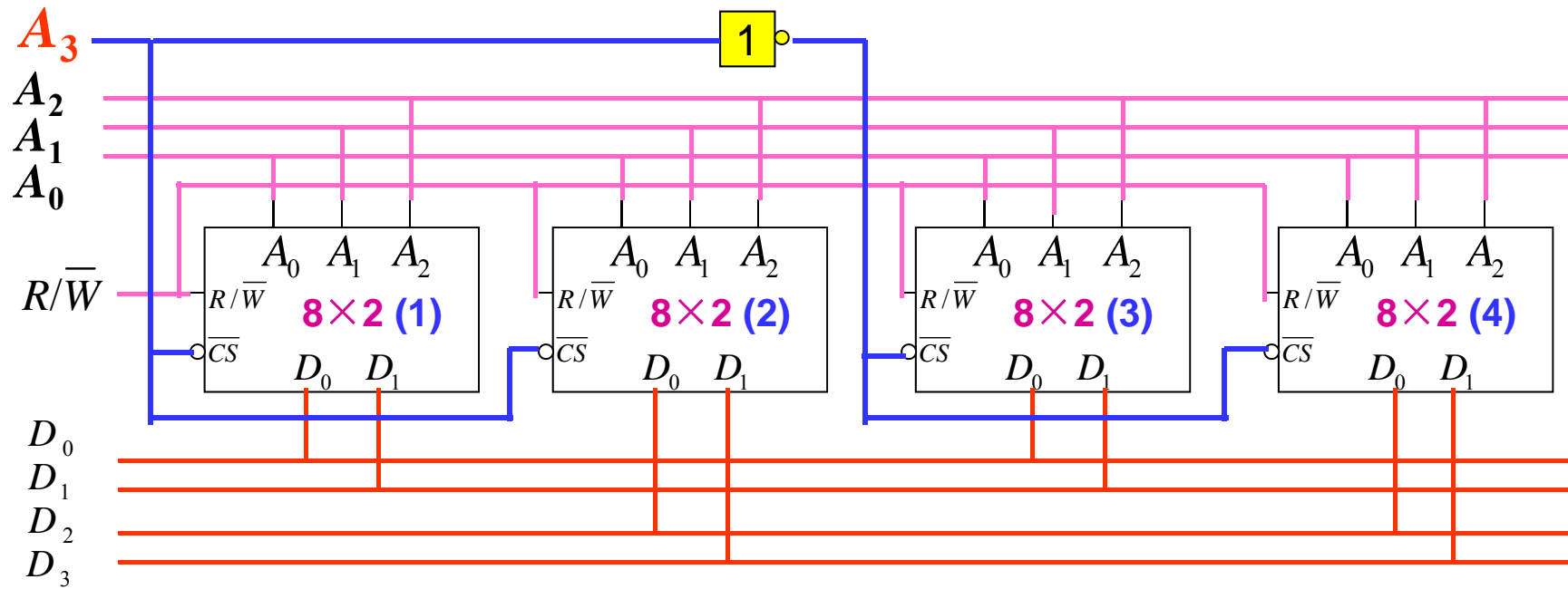
$A_4 A_3 A_2 A_1 A_0$	{	(1) 00000-00111
		(2) 01000-01111
		(3) 10000-10111
		(4) 11000-11111

3. 字位同时扩展

例 1 . 将 8×2 RAM 扩展成 16×4 RAM.

最好是先扩位，后扩字

$$\frac{16 \times 4}{8 \times 2} = 4 \text{ (} 8 \times 2 \text{ RAM)}$$



地址范围 $\left\{ \begin{array}{l} (1)(2): 0000 - 0111 \\ (3)(4): 1000 - 1111 \end{array} \right.$

十六进制表示法

16 进制: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

基数为 16.

存储器地址:

每4位二进制码表示一位十六进制码 后缀 **H**

例: 0000 1011 1111 \longrightarrow **0BFH**

地址: 0000 0000 ~ 0011 1111 \longrightarrow 00H ~ 3FH

地址: B00H ~ BFFH \longrightarrow

$A_{11}A_{10}A_9A_8A_7A_6A_5A_4A_3A_2A_1A_0$

1 0 1 1 0 0 0 0 0 0 0 0

1 0 1 1 1 1 1 1 1 1 1 1

§ 10.3 ROM (Read Only Memory)

ROM 是半导体存储器的一类。数据永久或半永久地存储在其中。可以从 **ROM** 中“读出”数据, 但没有“写”功能.

ROM 一般由专用装置写入数据, 数据一旦写入便不能随意改写, 断电后, 数据也不会丢失。

ROM : 组合逻辑电路

地址 → 变量	}	真值表
数据输出 → 函数		

10.3.1 ROM 分类

ROM { 固定ROM (Mask ROM)
可编程ROM (Programmable ROM)

PROM { **PROM** (一次可编程ROM)
EPROM (Erasable PROM 可擦除可编程)
EEPROM (Electrically Erasable PROM
电可擦除可编程)
Flash Memory (快闪存储器)
类似于E²ROM, 容量大

10.3.2 ROM 结构

1. ROM (Mask ROM):

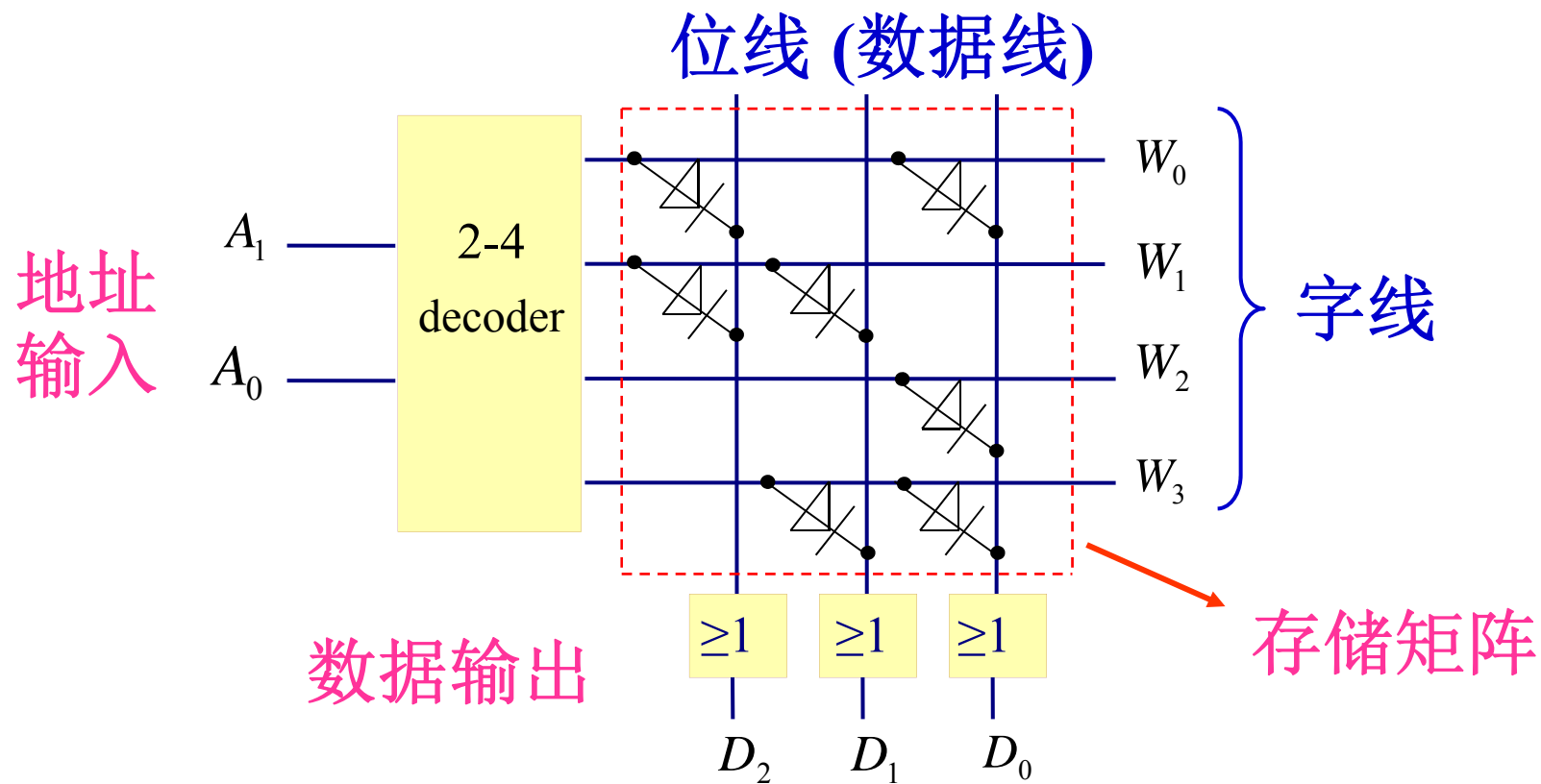
在生产过程中，数据被永久地储存在存储器内。
广泛用于标准函数

三个主要部分 { 地址译码器
存储矩阵
输出缓冲器

ROM 的结构与 **RAM**类似.

地址译码器：与门阵列

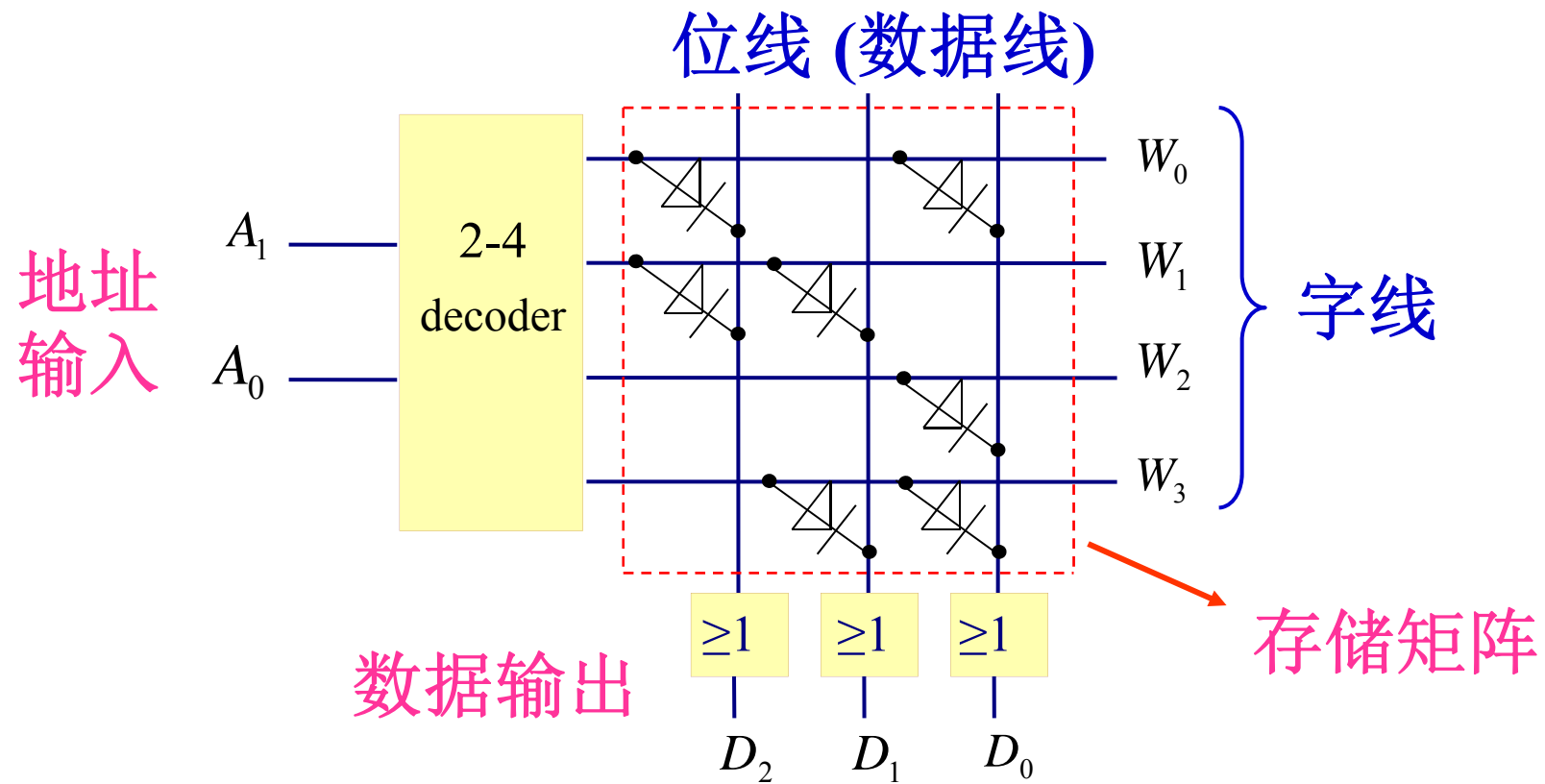
n -bit 地址码输入， 2^n 字线输出



地址译码器：把 n 条地址线译成 2^n 条字线 $W_0 \sim W_3$ 。

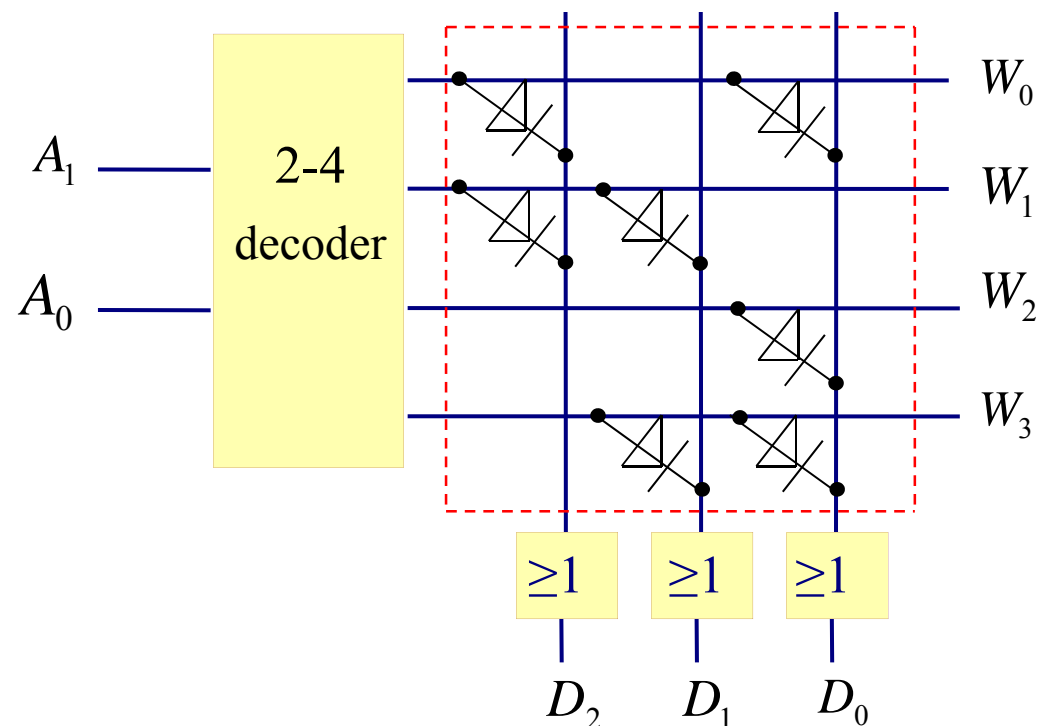
译码器 高电平有效

一组地址只能使一条字线高电平，使二极管导通。



存储矩阵 二极管, 晶体管, MOSFET
连接字线和数据线.

输出缓冲 或门, 3 位 ($D_2 D_1 D_0$)



当 $A_1A_0=00$, $W_0=1$, 二极管导通, 输出 $D_2D_1D_0=101$

按上图

ROM 实现:

$$D_2 = \overline{A_1} \overline{A_0} + \overline{A_1} A_0 = \overline{A_1}$$

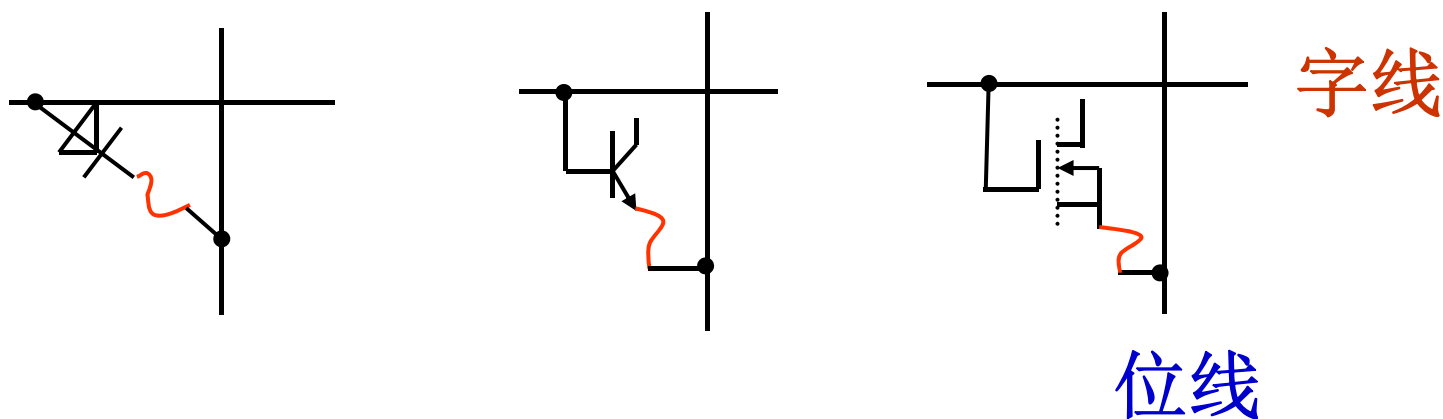
$$D_1 = \overline{A_1} A_0 + A_1 A_0 = A_0$$

$$D_0 = \overline{A_1} \overline{A_0} + A_1 \overline{A_0} + A_1 A_0 = \overline{A_0} + A_1$$

ROM实现的是标准与或式

2. PROM (一次可编程ROM):

每个存储单元 { Diode or transistor or MOSFET
熔丝



连接1，熔断0：出厂时全连(1)，用户根据需要
将需要存入0的单元上的熔丝熔断

PROM一旦进行了编程，就不能再修改了。

10.3.3 ROM 应用

存储器 (ROM 主要功能),
也可以实现标准与或式逻辑功能

1. ROM 实现组合连接功能

例：用ROM实现全加器

解：

全加器真值表

ROM 与译码器相同，
只能实现与或标准式

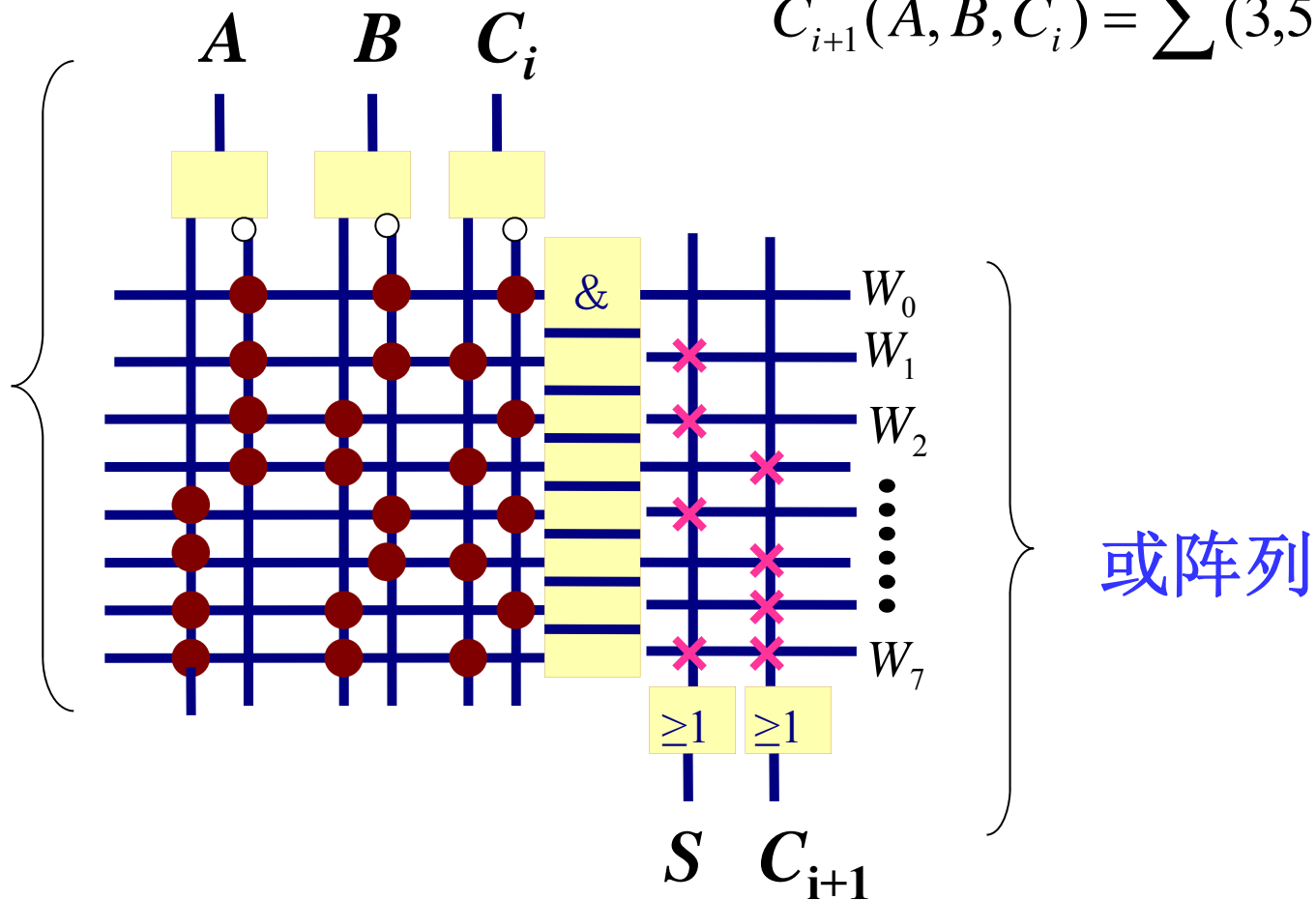
A	B	C_i	S	C_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$S(A, B, C_i) = \sum (1, 2, 4, 7)$$

$$C_{i+1}(A, B, C_i) = \sum (3, 5, 6, 7)$$

全译码
与阵列

3-8
译码器



或阵列

2. 用于函数运算表电路

例：用**ROM**构成能实现函数 $y = x^2$ 的运算表电路，
 x 的取值范围为 $0 \sim 15$ 的正整数。

解：（1）分析

变量 x : $0 \sim 15$ 正整数，

→ 4 位二进制数 $x = X_3X_2X_1X_0$

函数 $y = x^2$,

y 的最大值 : $15^2 = 225$,

→ 8 位二进制数 $y = Y_7Y_6Y_5Y_4Y_3Y_2Y_1Y_0$

(2) 真值表

x_3	x_2	x_1	x_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	0	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

(3) 写标准与或表达式

$$Y_7 = m_{12} + m_{13} + m_{14} + m_{15}$$

$$Y_6 = m_8 + m_9 + m_{10} + m_{11} + m_{14} + m_{15}$$

$$Y_5 = m_6 + m_7 + m_{10} + m_{11} + m_{13} + m_{15}$$

$$Y_4 = m_4 + m_5 + m_7 + m_9 + m_{11} + m_{12}$$

$$Y_3 = m_3 + m_5 + m_{11} + m_{13}$$

$$Y_2 = m_2 + m_6 + m_{10} + m_{14}$$

$$Y_1 = 0$$

$$Y_0 = m_1 + m_3 + m_5 + m_7 + m_9 + m_{11} + m_{13} + m_{15}$$

(4) 画ROM存储矩阵节点连接图

