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

Introduction of Semiconductor Memory

半导体存储器几乎是当今数字系统中不可缺少的组成部分，它是存储信息的器件，由许多存储单元组成。按照集成度划分，半导体存储器属于大规模集成电路。

每个存储单元能存储一位二进制信息，可以由触发器或电容构成。

10.1.1 半导体存储器分类 Memory Family

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

Technical Parameters of Memory

存储周期 **Access period**

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 **Capacity**

(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×位数

容量 2K ×16 { 2K 字 (2^{11}个字)
16位 (2 bytes) 字长16位

常用多少字节表示容量： 32K bits 或 4K bytes

§10.2 随机存储器 RAM

Random Access Memory

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

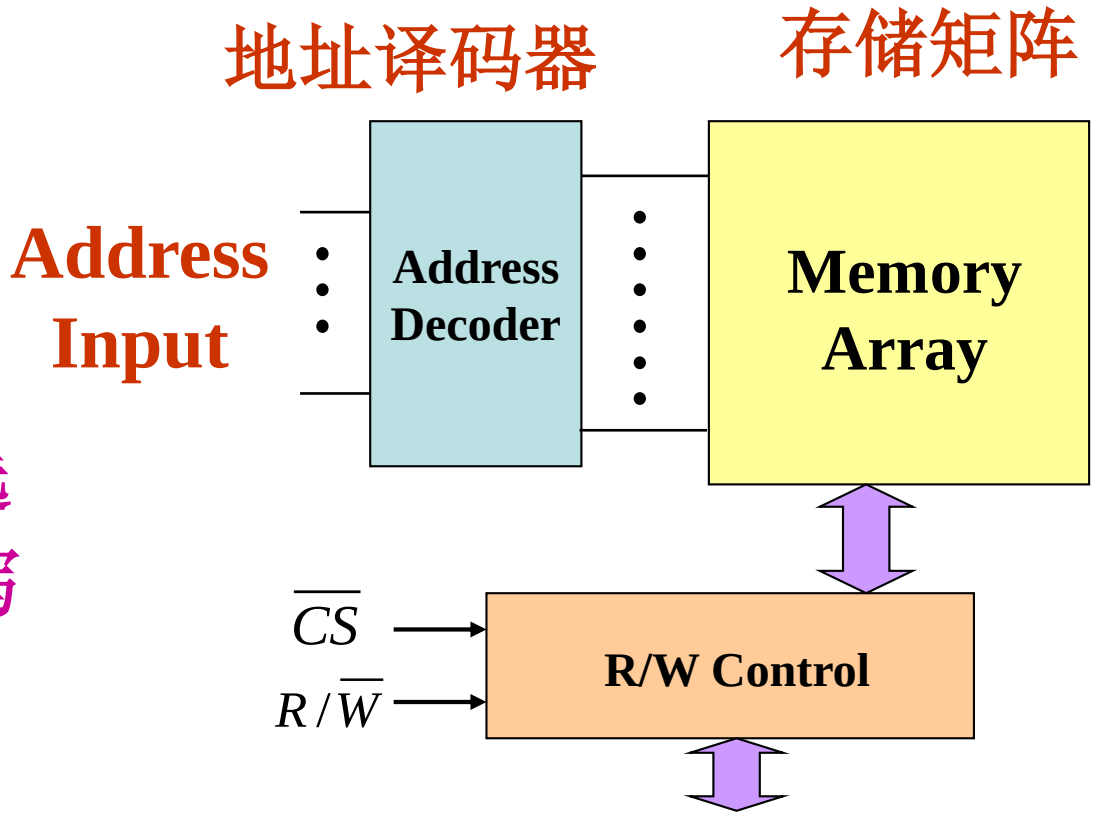
RAM分成
- 静态 SRAM（Static ）
- 动态 DRAM（Dynamic ）

10.2.1 RAM 基本结构 RAM Structure

RAM属于时序电路。

RAM主要包括
- 存储矩阵
- 地址译码器
- 输入/输出控制

存储矩阵
地址译码器
输入/输出控制



Chip Select 片选
Read/Write 读/写

1. 存储矩阵

是RAM的主要部分

SRAM的存储单元由触发器构成。

DRAM的存储单元由MOS管和电容器构成。

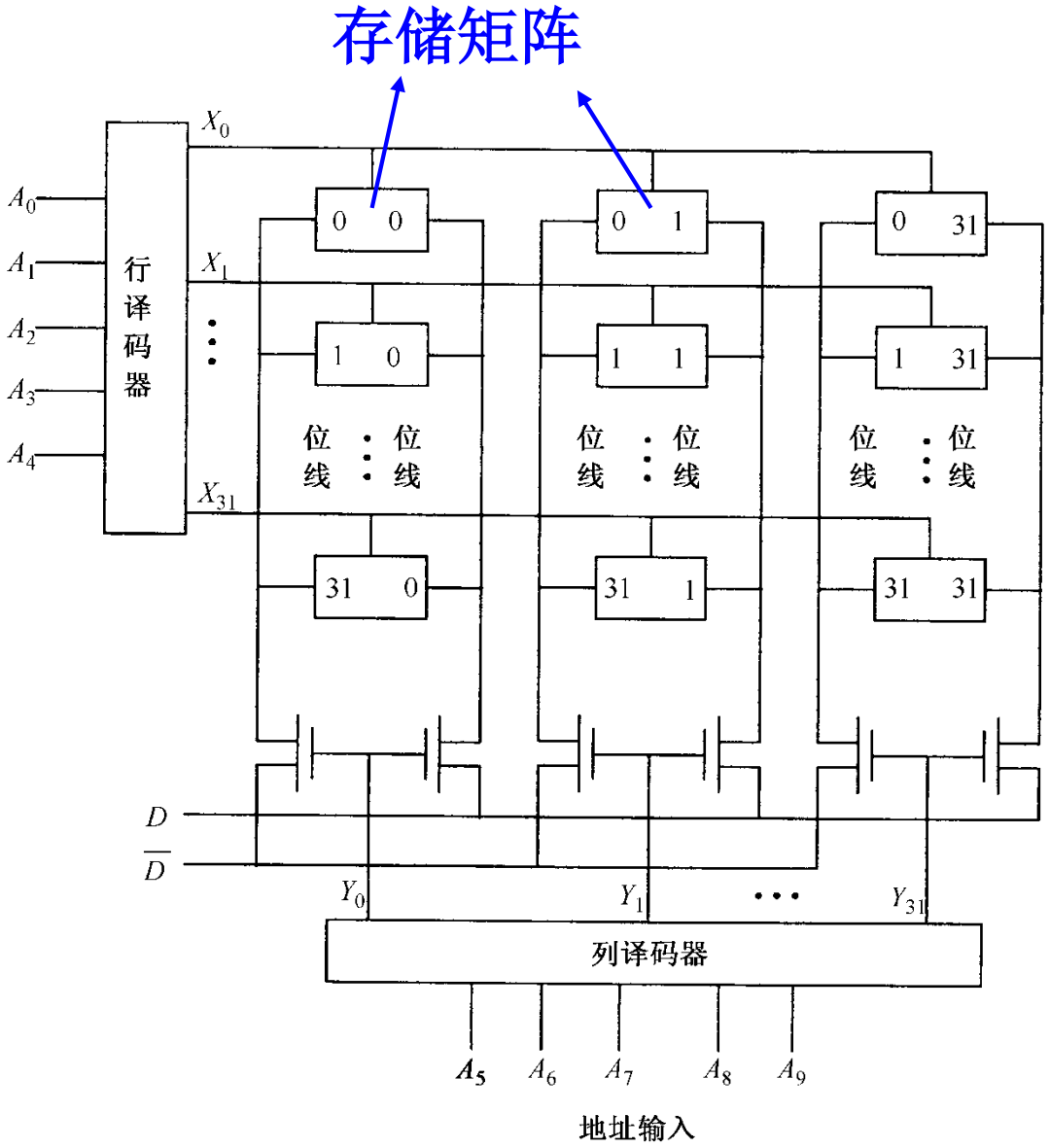
存取时间、时钟频率、工作电压等参数不同。

1K×1位存储矩阵及地址译码器

此图中每个字是一位。

1024 个字排成 32×32 的矩阵

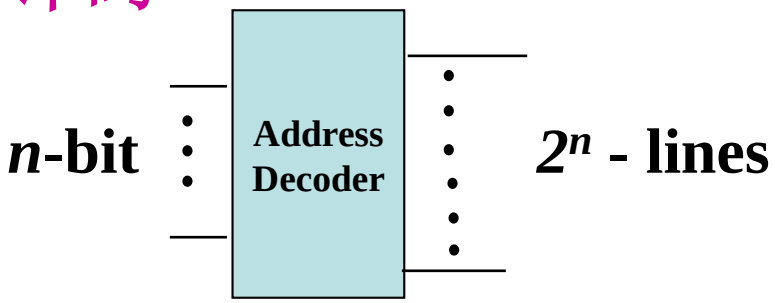
每个存储单元都有固定的编号（X_i 行、Y_j 列）即**地址 Address**



2. 地址译码器

- 单译码
- 双译码

单译码

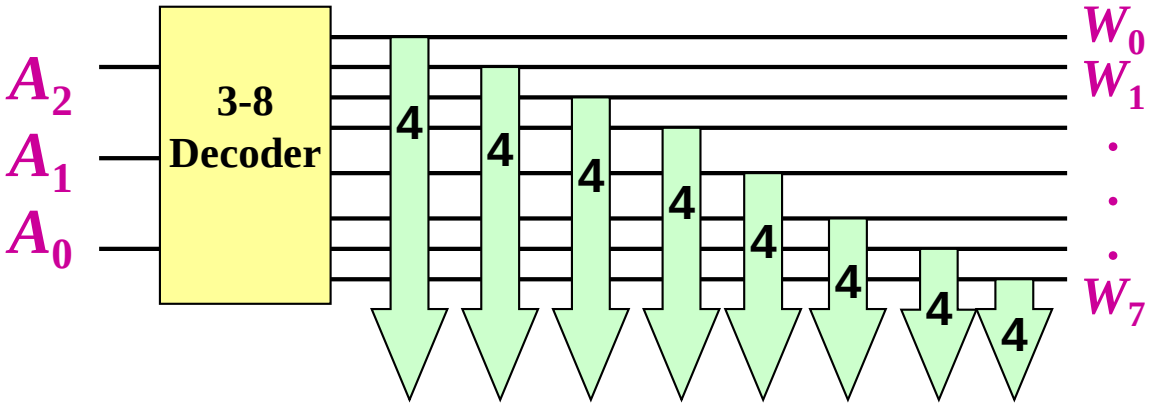


双译码

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：

$2^3 \times 4$ RAM



1024 字：2^{10}

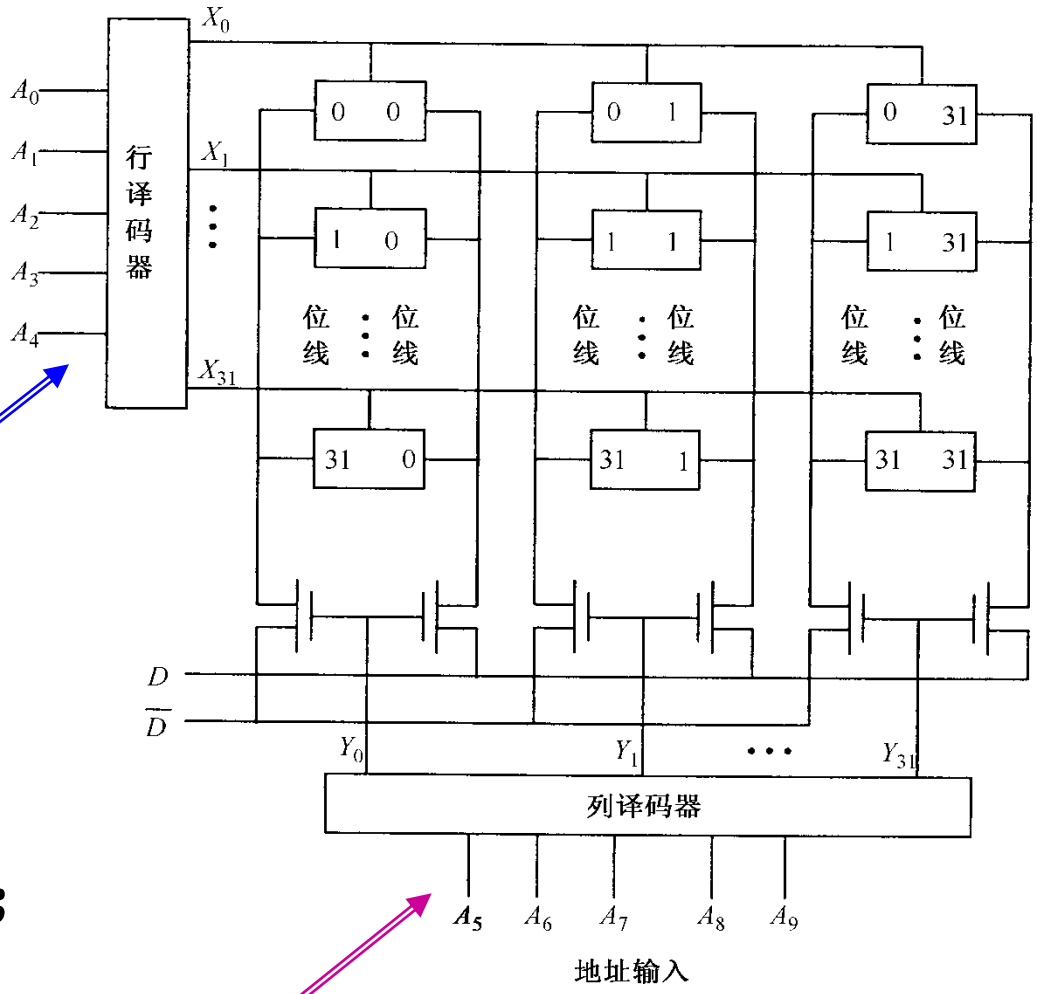
字线 (地址线): 10条

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}；**

高电平有效译码器

地址:

$A_9A_8A_7A_6A_5\ A_4A_3A_2A_1A_0$

= 00000 00001

行线 $X_1 = 1$

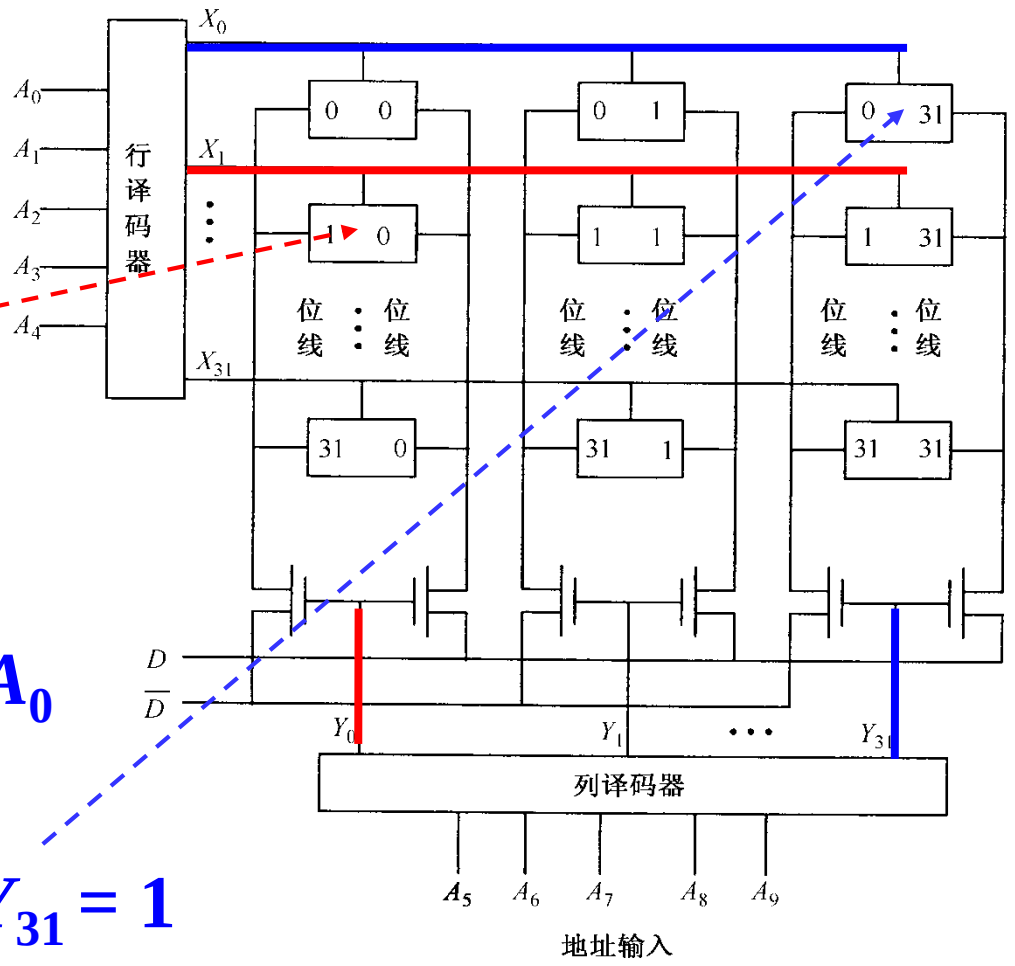
列线 $Y_0 = 1$

地址:

$A_9A_8A_7A_6A_5\ A_4A_3A_2A_1A_0$

= 11111 00000

行线 $X_0 = 1$、 列线 $Y_{31} = 1$



每个字有一个自己的地址。

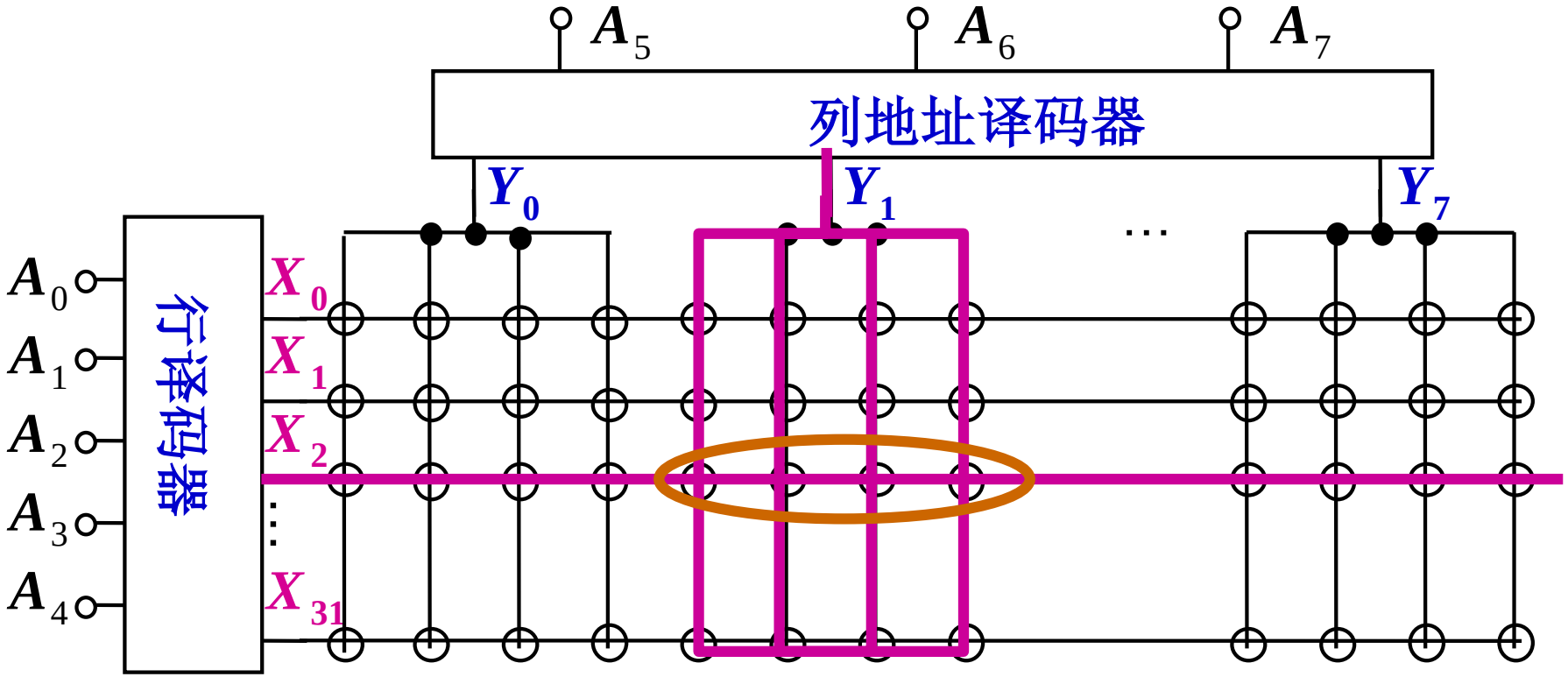
双译码结构如何表示位数：

例: 容量 $2^8 \times 4$ RAM

地址线8条

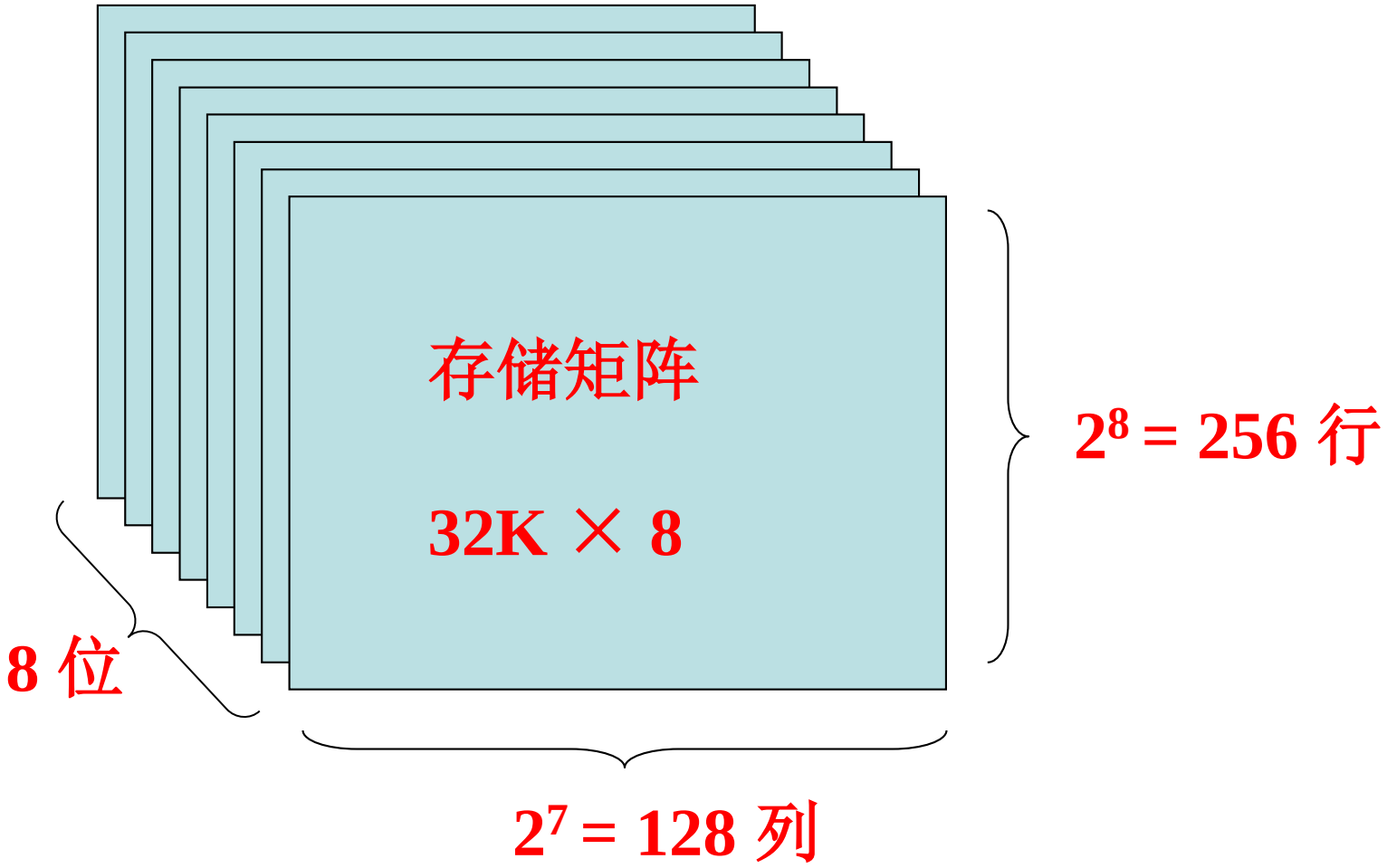
- 5 – 行线 → 32 (X)
- 3 – 列线 → 8 (Y)

若 $A_7A_6A_5A_4A_3A_2A_1A_0 = 00100010$, 有 $Y_1 = 1$, $X_2 = 1$.



选中 4 个存储单元， 即 4 位数据

立体图 $32K \times 8 = 2^{15} \times 8 = 2^8 \times 2^7 \times 8$



该芯片有多少个地址？ 2^{15} 个地址

10.2.2 RAM芯片简介

例 1. 16×2 RAM

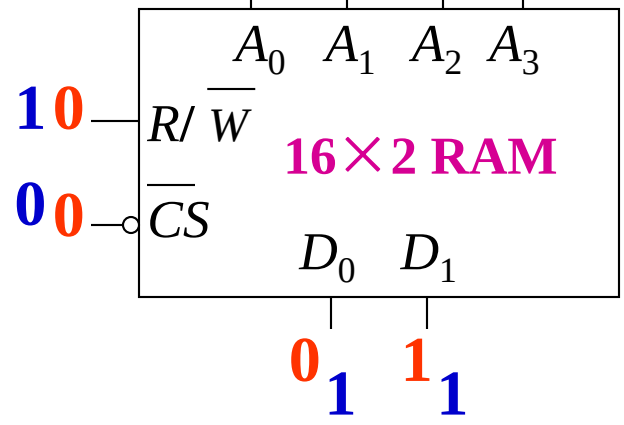
容量: 16 ×2 个存储单元

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 0 1 0 0

1 0 1 1



读/写 控制

1 0

片选
低有效

0 0

0 1 1 1

数据 输入 / 输出

A_3, D_1 高位

练习

例 2. RAM2114

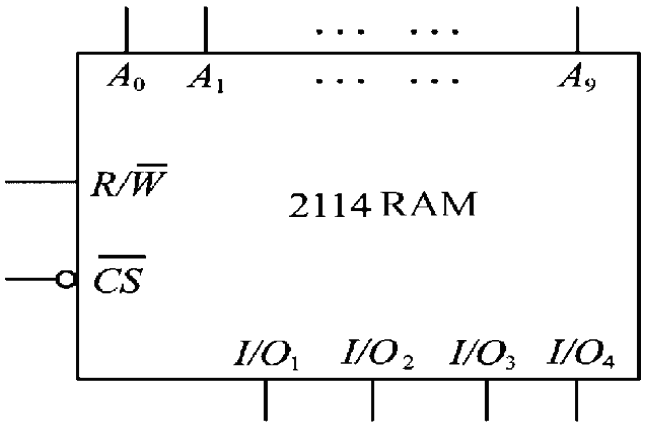
容量:

1024 字 × 4 位

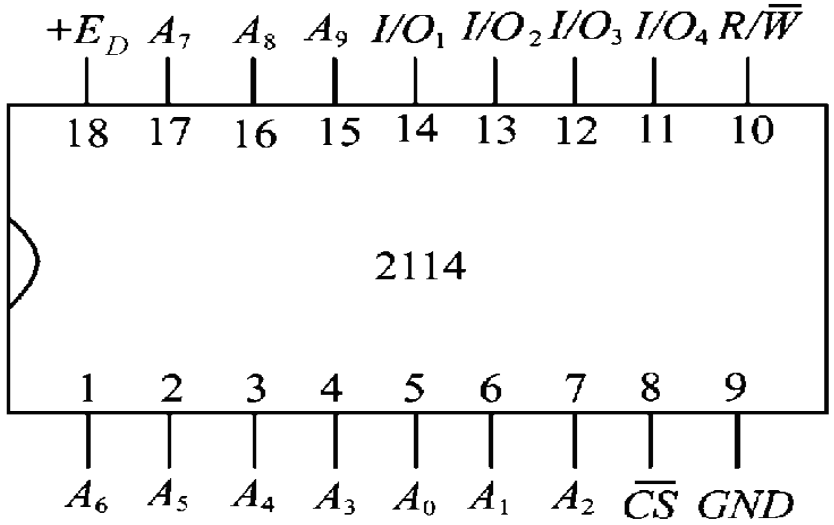
地址线: 10, $A_0 \sim A_9$,

数据线 (输入/输出): 4, $I/O_1 \sim I/O_4$

符号



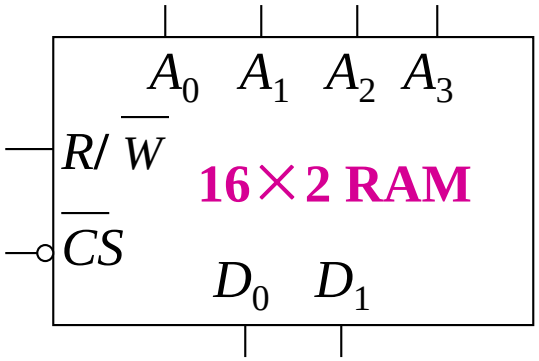
管脚图



10.2.3 RAM 扩展 Memory Expansion

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

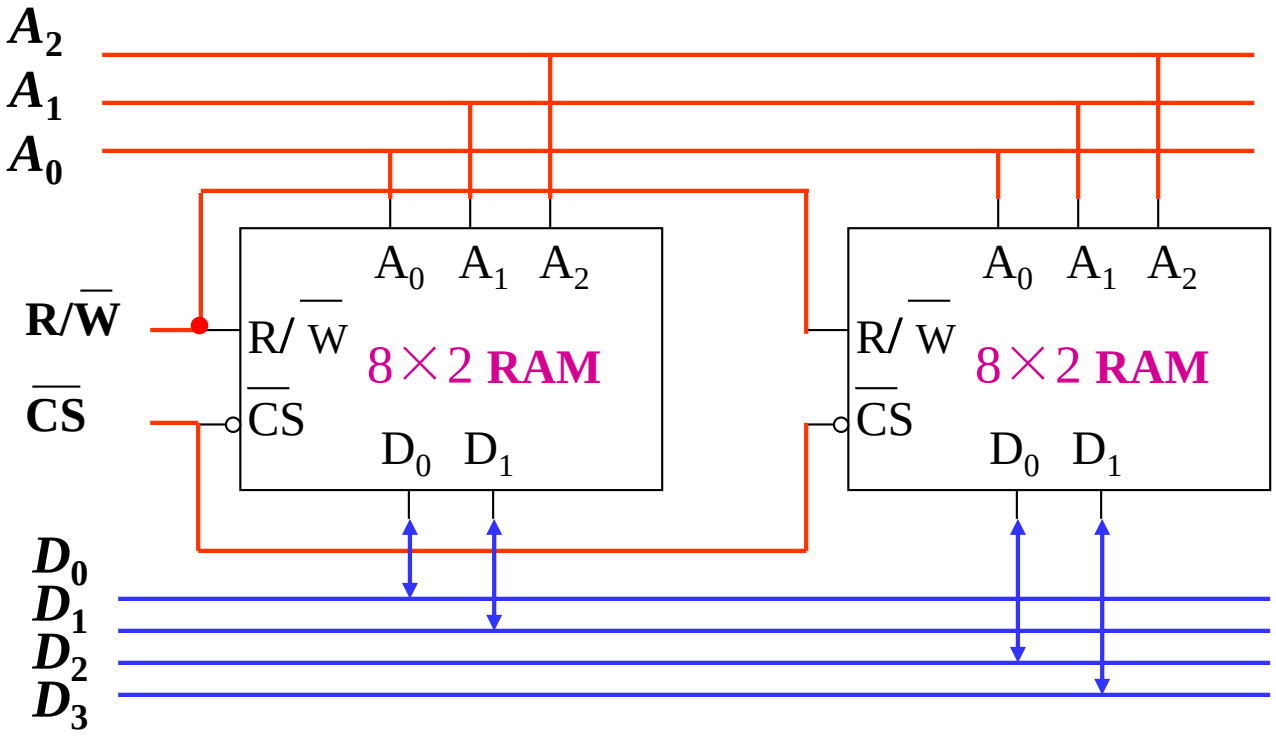
共用:
- 地址线
- $R/\overline{W}$
- $\overline{CS}$

增加（扩展）：
数据线（位线）

例：将 8×2 RAM 扩展成 8×4 RAM

$$\frac{8\times4}{8\times2}=2\text{片}\quad\textbf{(8 ×2 RAM)}$$

2-bit → 4-bit



地址线共用

$R/\overline{W}$ 共用

$\overline{CS}$ 共用

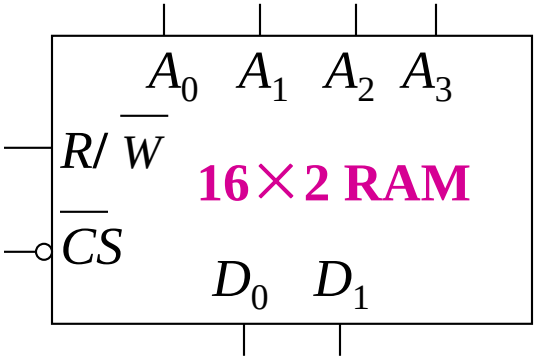
两片RAM 同时工作, 地址范围不变, 位（字长）扩展.

增加：数据线（位线）

2. 字扩展

增加地址线

用 $\overline{CS}$ 扩展字



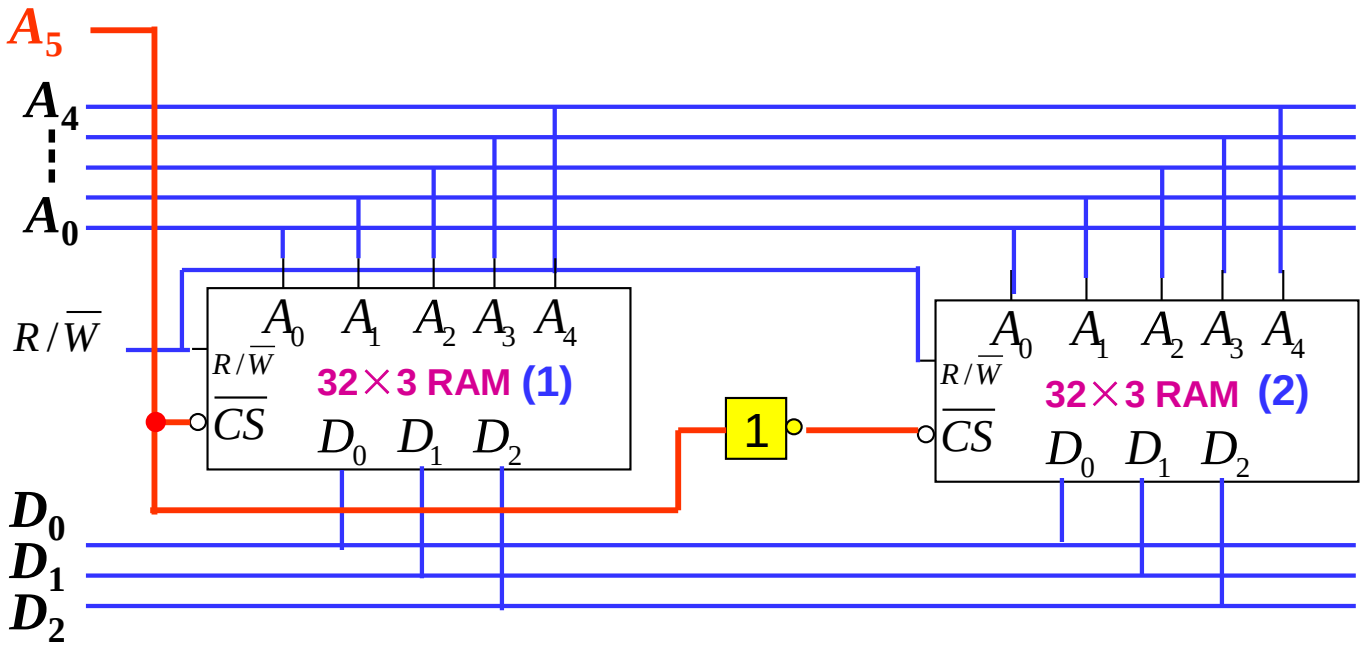
共用：

- 原地址线
- $R/\overline{W}$
- 数据线

例 1. 将 32×3 RAM 扩展成 64×3 RAM.

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

地址线 5 → 6



原地址线共用

$R/\overline{W}$ 共用

数据线共用

A_5 =0 (1) 工作，(2) 停止；地址 000000-011111

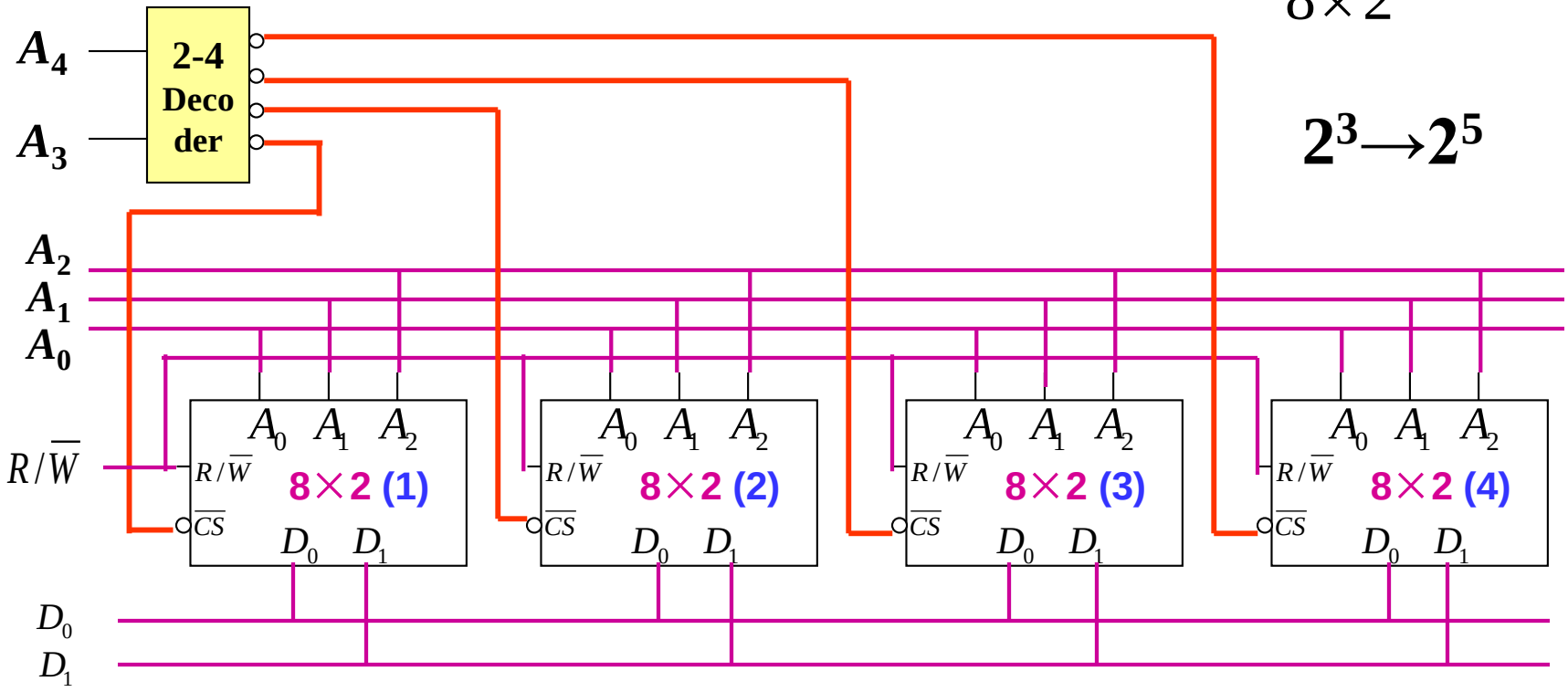
A_5 =1 (1) 停止，(2) 工作；地址 100000-111111

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：

$A_4A_3A_2A_1A_0$

(1) 00000-00111

(2) 01000-01111

(3) 10000-10111

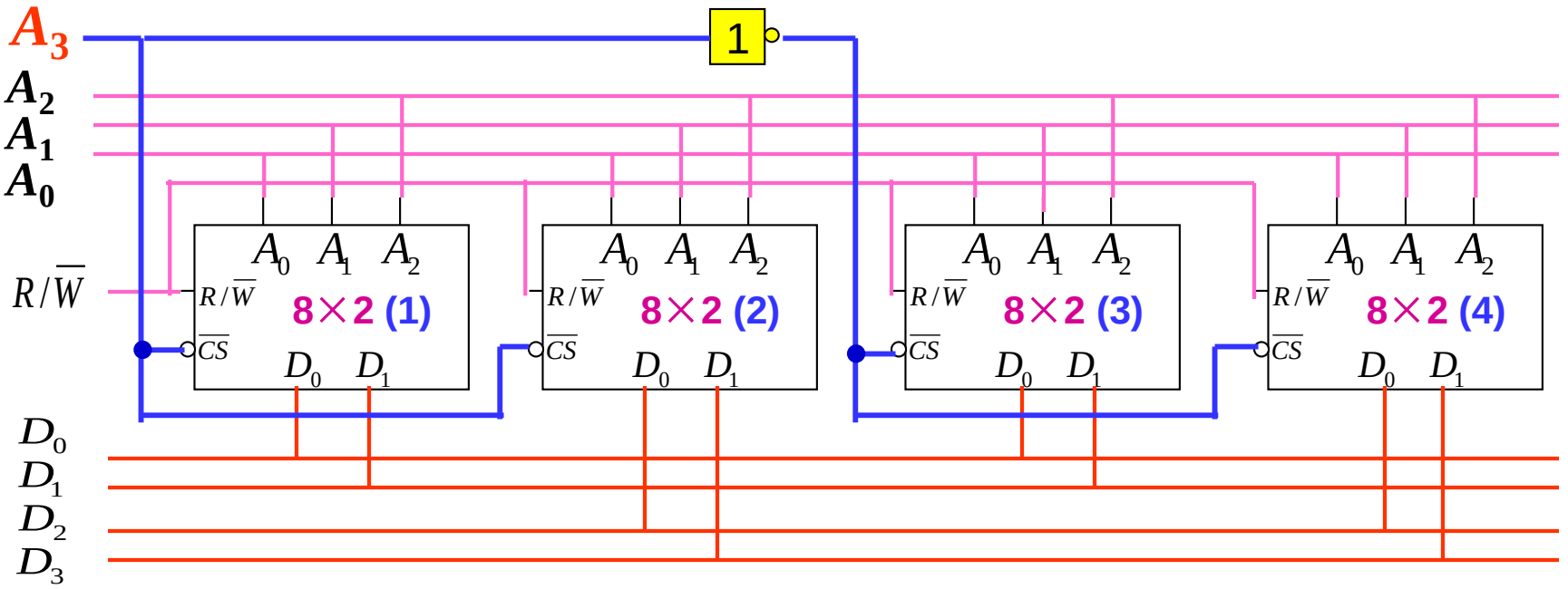
(4) 11000-11111

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

$$\frac{16\times4}{8\times2}=4\ (8\times2\ \text{RAM})$$



地址范围

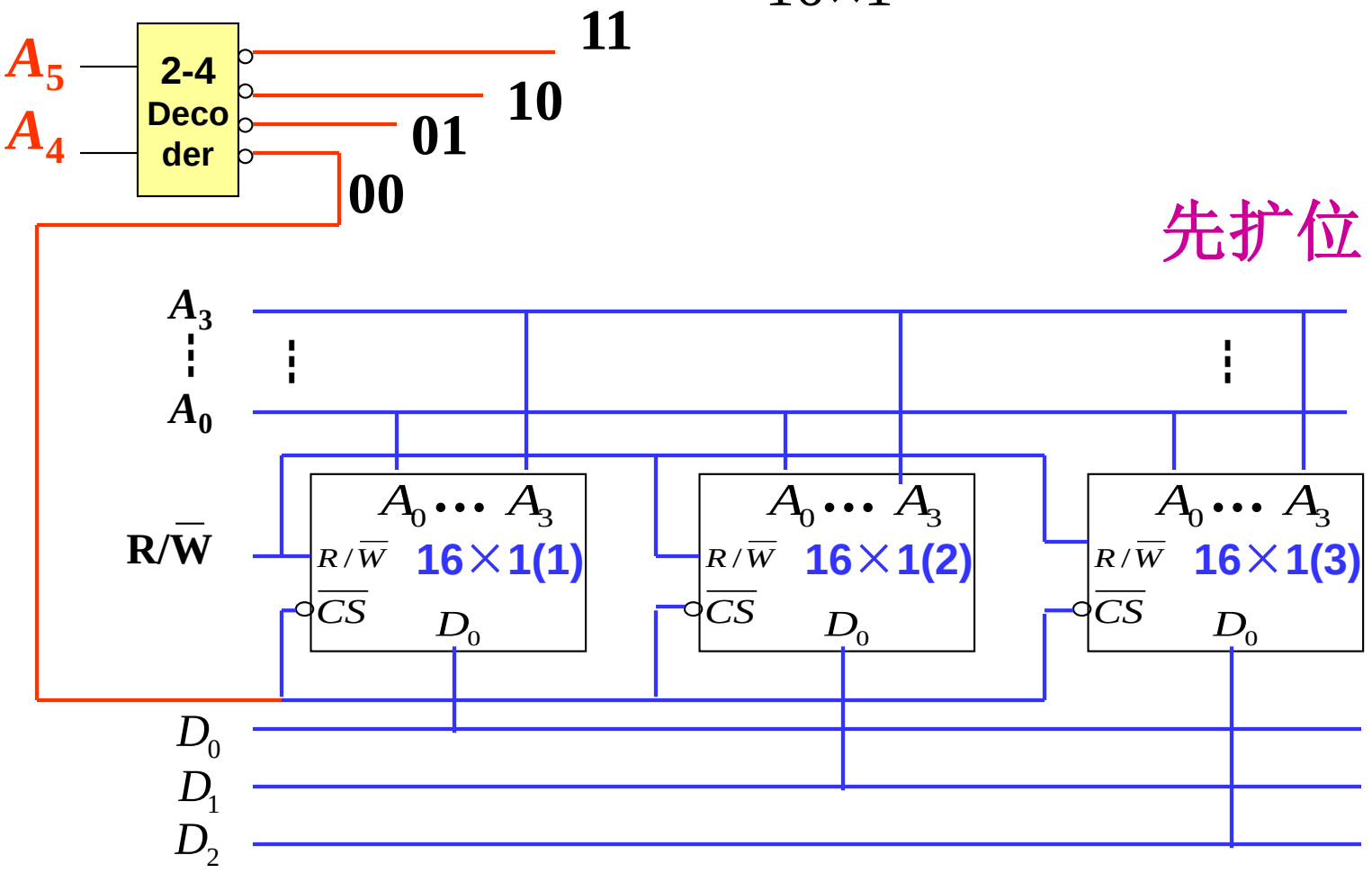
(1)(2): 0000 - 0111

(3)(4): 1000 - 1111

例2 . 将 16×1 RAM 扩展成 64×3 RAM.

扩字：地址线 4→6

$$\frac{64\times 3}{16\times 1}=12 \quad (16\times 1\ \text{RAM})$$



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** ⇒ **0BFH**

地址：**0000 0000 ~ 0011 1111** ⇒ **00H ～ 3FH**

地址：**B00H ~ BFFH** ⇒

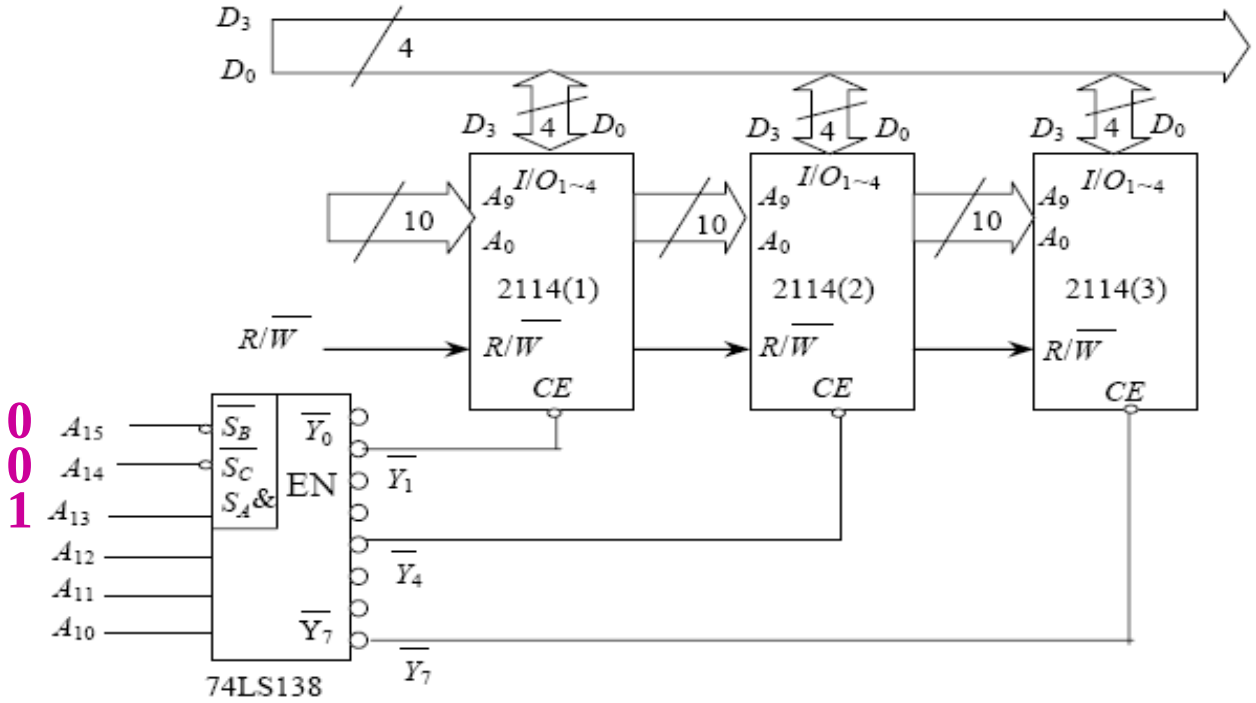
$A_{11}A_{10}A_9A_8A_7A_6A_5A_4A_3A_2A_1A_0$

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

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

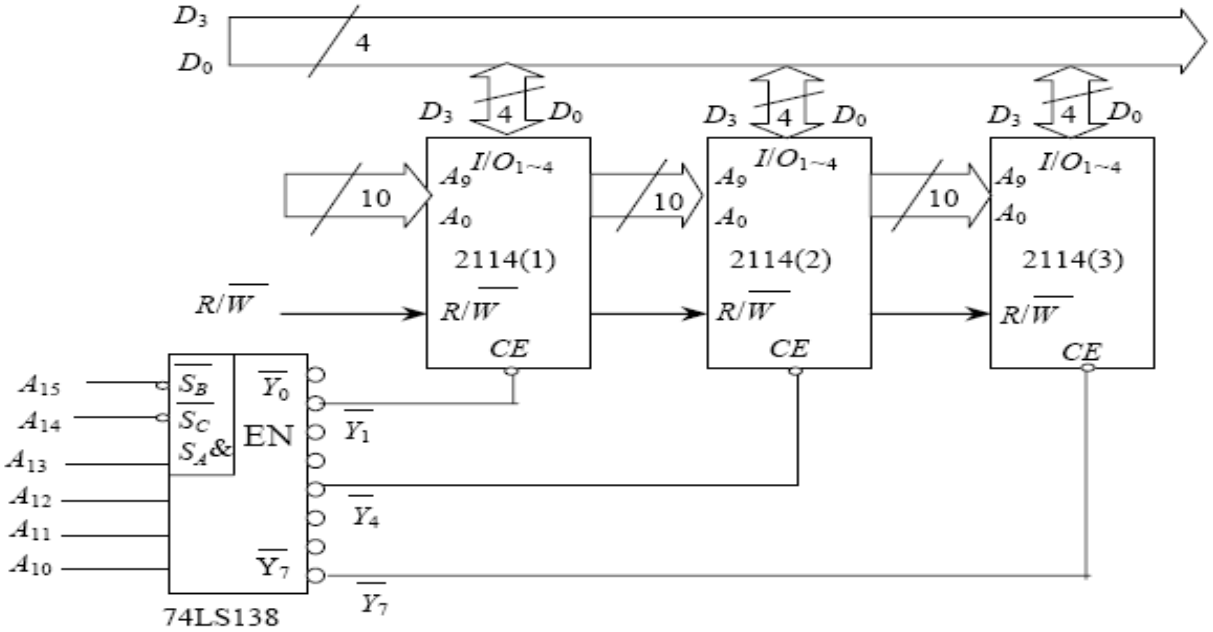
例题：（10.6） RAM2114（1k×4）组成如题图10.6所示电路。（1）确定图示电路内存单元的容量是多少？若要实现2k×8的内存，需要多少片2114芯片？

（2）写出2114(1)至2114(3)的地址范围（用十六进制表示）。



解：（1）容量是(1K×4) ×3 = 3K×4

$$\frac{2K\times 8}{1K\times 4}=4$$ 片 2114 可以实现 2K×8 内存



（2）写出2114(1)至2114(3)的地址范围（用十六进制表示）

(2)

$A_{15}A_{14}A_{13}A_{12}A_{11}A_{10}\ A_9A_8\ A_7A_6A_5A_4A_3A_2A_1A_0$

2114(1) $\overline{Y}_1$

0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 **2400H ~**

0 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 **27FFH;**

2114(2) $\overline{Y}_4$

0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 **3000H ~**

0 0 1 1 0 0 1 1 1 1 1 1 1 1 1 1 **33FFH;**

2114(3) $\overline{Y}_7$

0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 **3C00H ~**

0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 **3FFFH**

§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
数据输出 → 函数
} 真值表

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） E^2ROM
- Flash Memory (快闪存储器）类似于E^2ROM，容量大它能在字节水平上进行删除和重写而不是整个芯片擦写，这样闪存就比E^2PROM的更新速度快。

施敏博士发明（台湾，中国工程院外籍院士）

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。

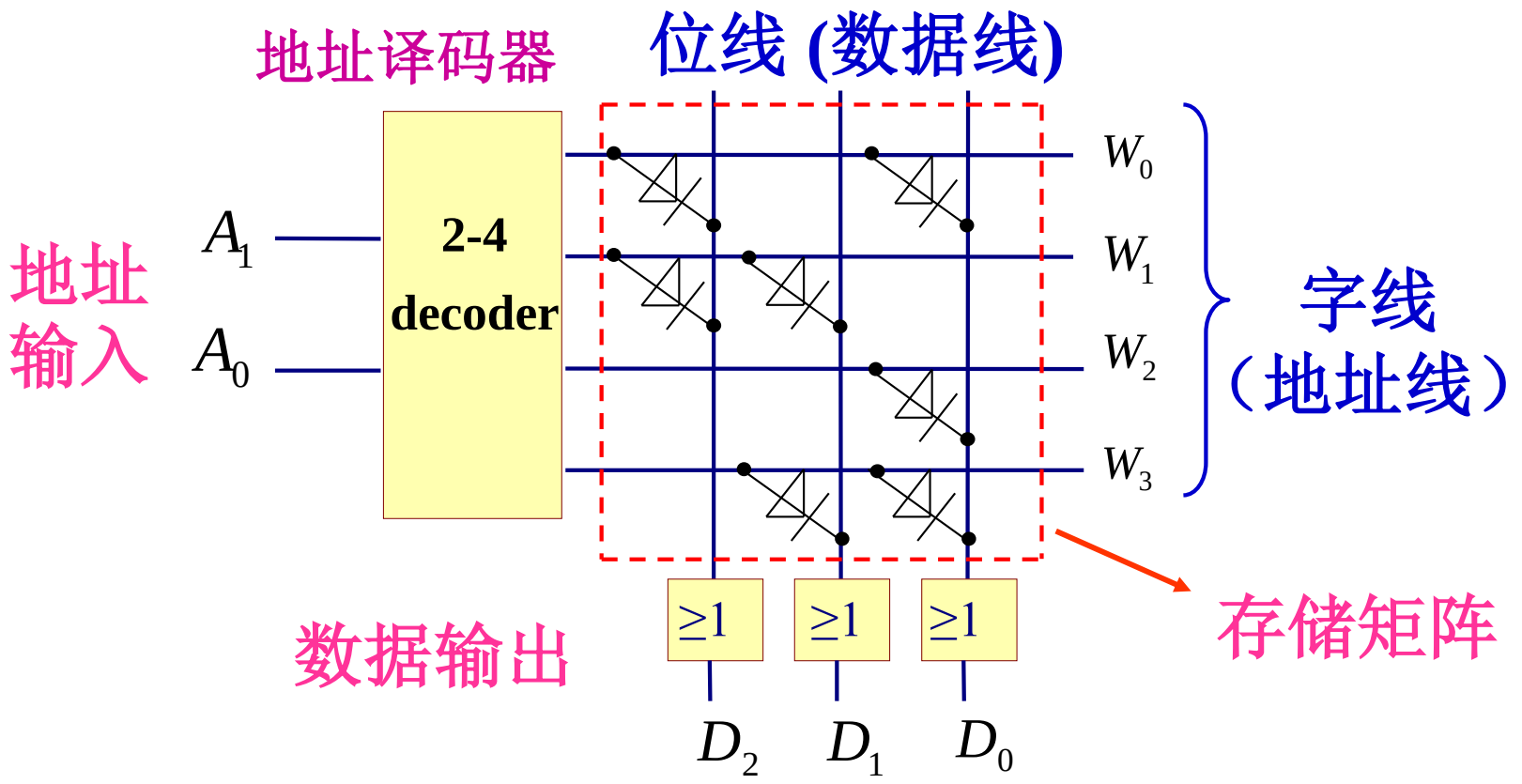
广泛用于标准函数

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

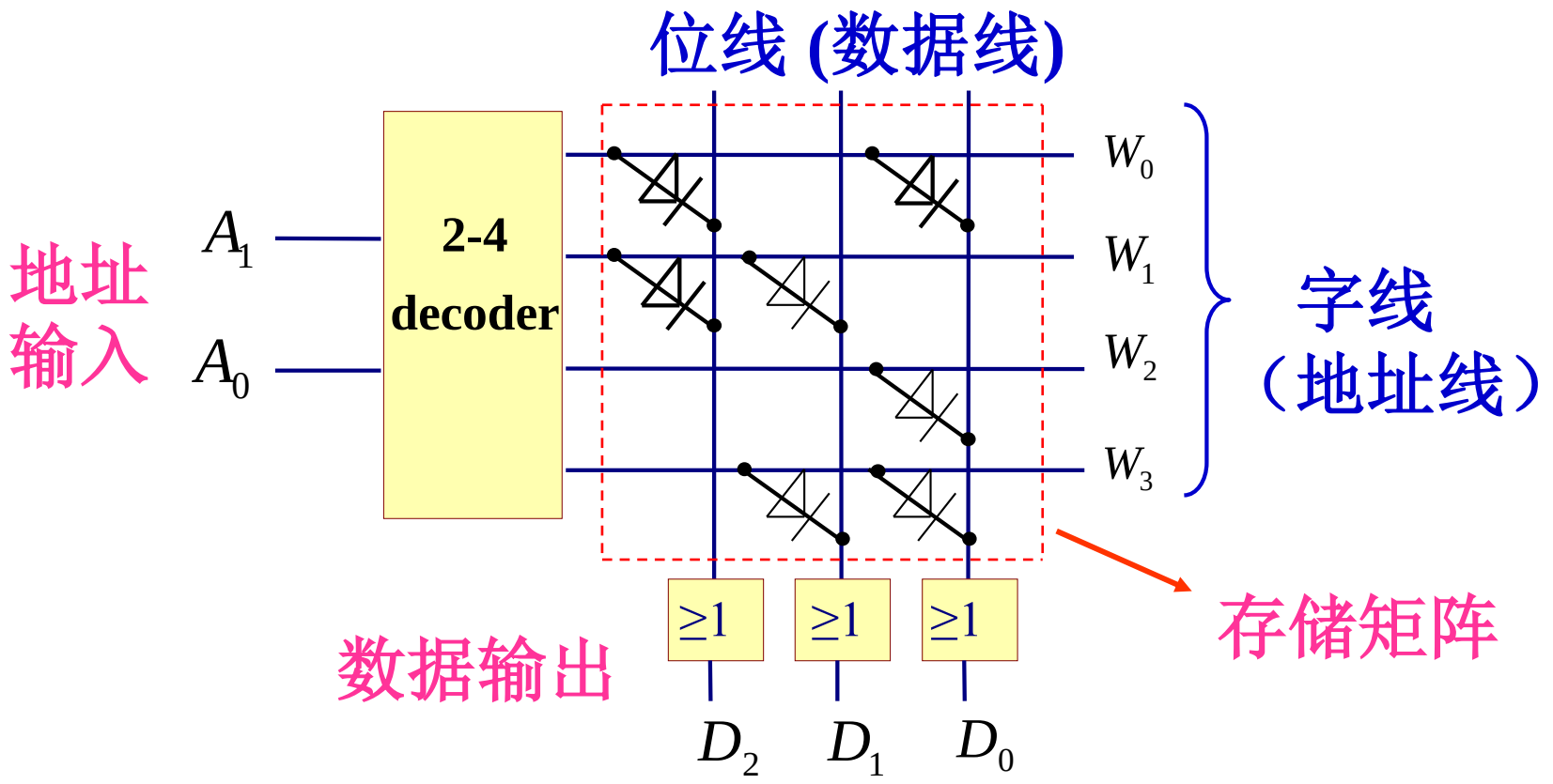
***n* –bit 地址码输入，2^n 字线输出**



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

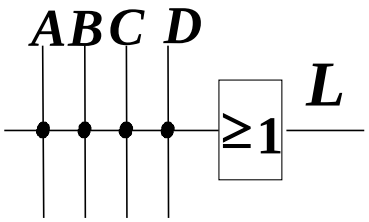
译码器 高电平有效

一组地址只能使一条字线高电平，使二极管导通。

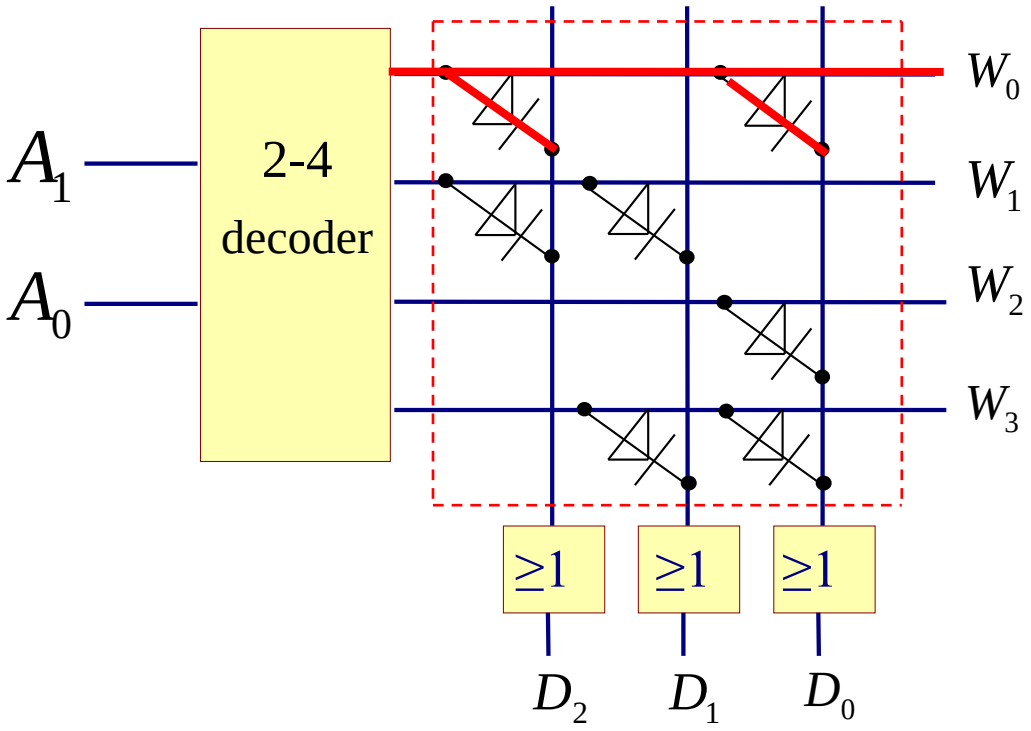


存储矩阵 二极管, 晶体管, MOSFET
连接字线和数据线.

输出缓冲 或门, 3 位 (D_2 D_1 D_0)



$$L = A + B + C + D$$



当 A_1A_0 =00,
W_0 =1,
相应的二极管导通,
输出 $D_2D_1D_0$ =101

按上图 ROM 实现:

$$D_2 = W_0 + W_1 = \overline{A}_1\overline{A}_0 + \overline{A}_1A_0 = \overline{A}_1$$

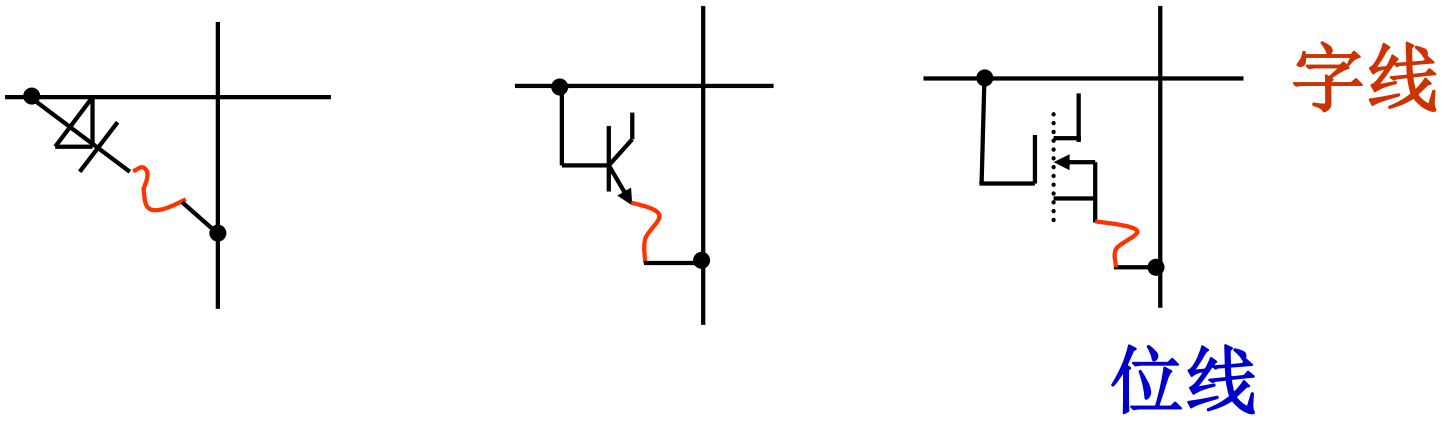
$$D_1 = W_1 + W_3 = \overline{A}_1A_0 + A_1A_0 = A_0$$

$$D_0 = W_0 + W_2 + W_3 = \overline{A}_1\overline{A}_0 + A_1\overline{A}_0 + A_1A_0 = \overline{A}_0 + A_1$$

ROM实现的是标准与或式

2. PROM (一次可编程ROM):

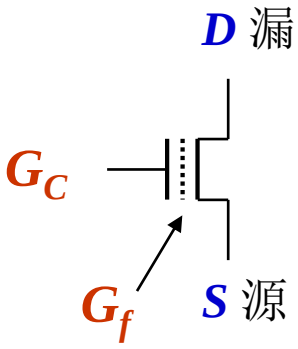
每个存储单元 { Diode or transistor or MOSFET
熔丝



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。

3. EPROM



SIMOS管

EPROM的存储单元多采用叠栅注入MOS管(SIMOS: Stacked-gate Injection MOS)

叠层栅MOS管是一个N沟道增强型MOS管

两个栅极 { 控制栅G_C：控制读出和写入
浮置栅G_f：长期保存注入电荷（信息）

施敏博士发明

浮置栅上注入了电荷的SIMOS管相对于写入了1，未注入电荷相当于存入了0。

在浮置栅上注入电荷及使电荷消失的方法包括：

- UVEPROM （通常只写EPROM）
  （Ultra – Violet EPROM 紫外线擦除PROM)
- EEPROM (Electrically EPROM 电擦除PROM，E^2ROM)

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 实现组合逻辑功能

例 : 用ROM实现全加器

解:

全加器真值表

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

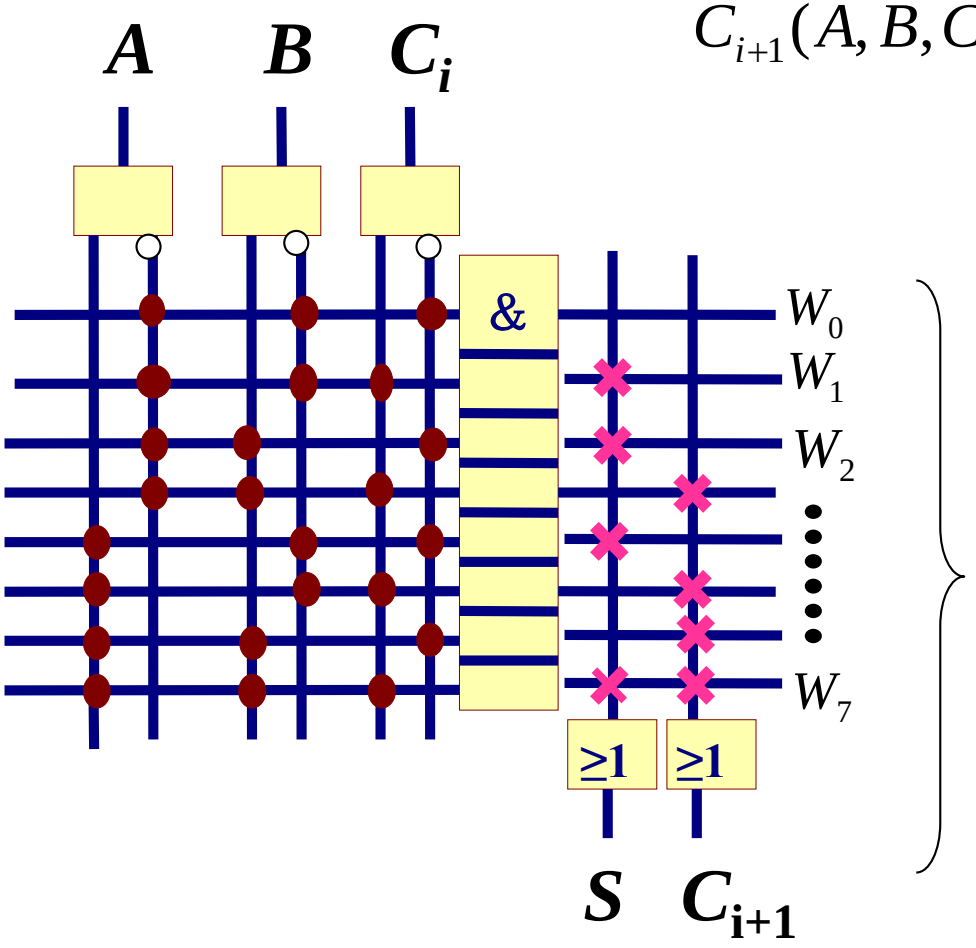
ROM 与译码器相同,
只能实现与或标准式

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

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

全译码与阵列

3-8译码器



或阵列

计算机编程，自动生成熔丝图，写入电路。

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 ~ 15 的正整数。

解：（1）分析

变量 x : 0~15 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

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

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

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

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

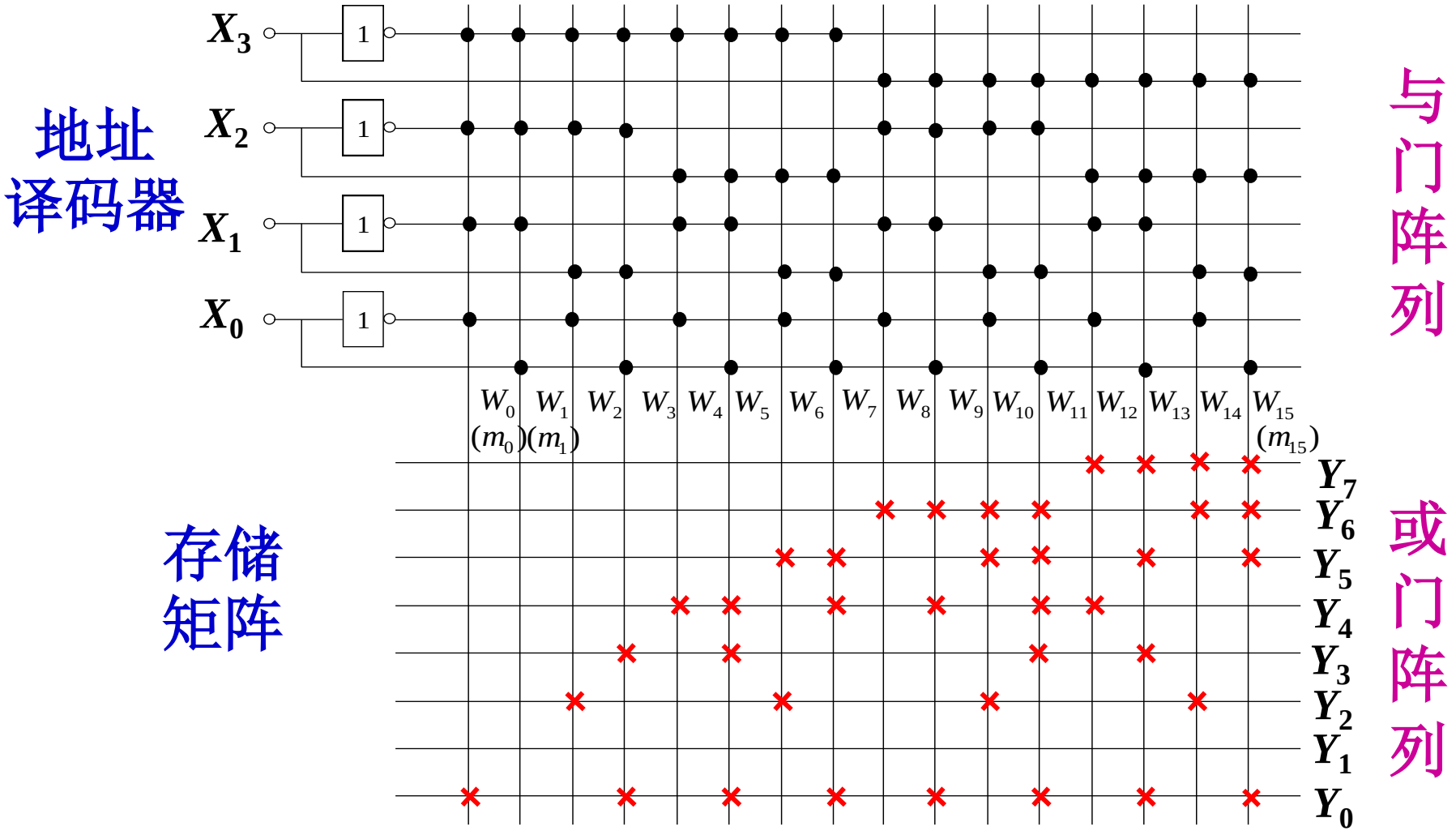
$$Y_3 = m_3 + m_5 + m_{11} + m_{13}$$

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

$$Y_1 = 0$$

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

（4）画ROM存储矩阵节点连接图



作业：

10.1 现有容量为256 × 8 RAM一片，试回答：

（1）该片RAM共有多少个存储单元？

（2）该片RAM共有多少个字？字长多少位？

（3）该片RAM有多少条地址线？

（4）访问该片RAM时，每次会选中几个存储单元？

解：（1）共有256 × 8 ＝ 2048 个存储单元

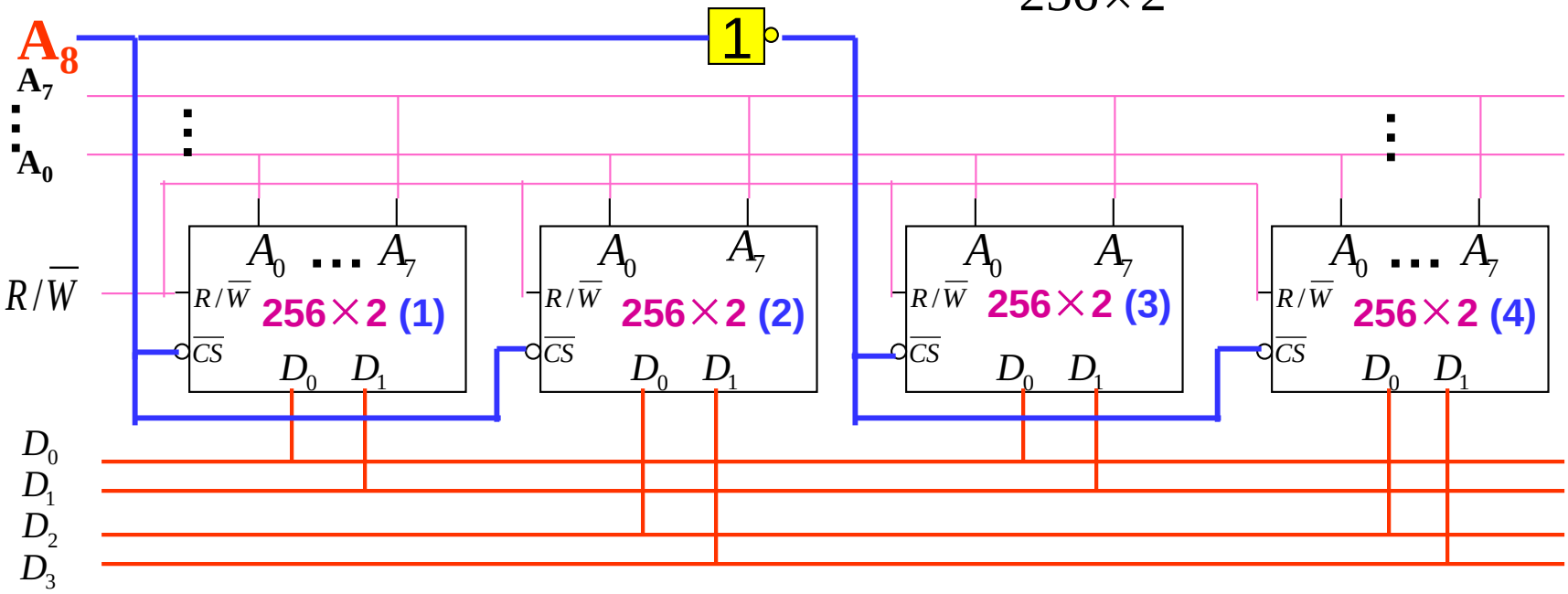
（2）该RAM共有256个字，字长8位

（3）地址线有 $\log_2 256$ **=8条**

（4）访问该片RAM时，每次选中一个字即8位，即8个存储单元

10.4. 画出把 256 × 2 RAM扩展成 512 × 4 RAM的连接图，并说明各片RAM的地址范围 。

$$\frac{512\times4}{256\times2}=4\ (256\times2\ \text{RAM})$$



$A_8A_7A_6A_5A_4A_3A_2A_1A_0$

(1)(2) 0 0 0 0 0 0 0 0 0

0 1 1 1 1 1 1 1 1

000H～0FFH

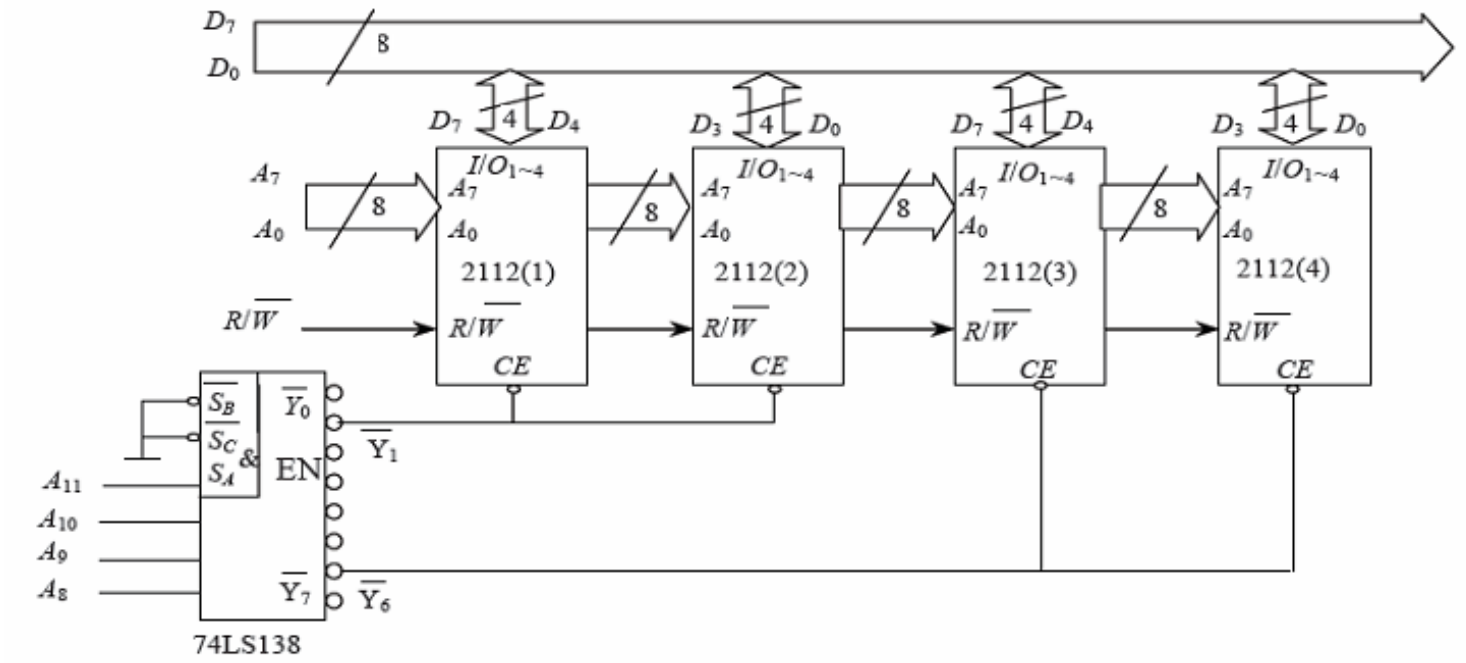
(3)(4) 1 0 0 0 0 0 0 0 0

1 1 1 1 1 1 1 1 1

100H～1FFH

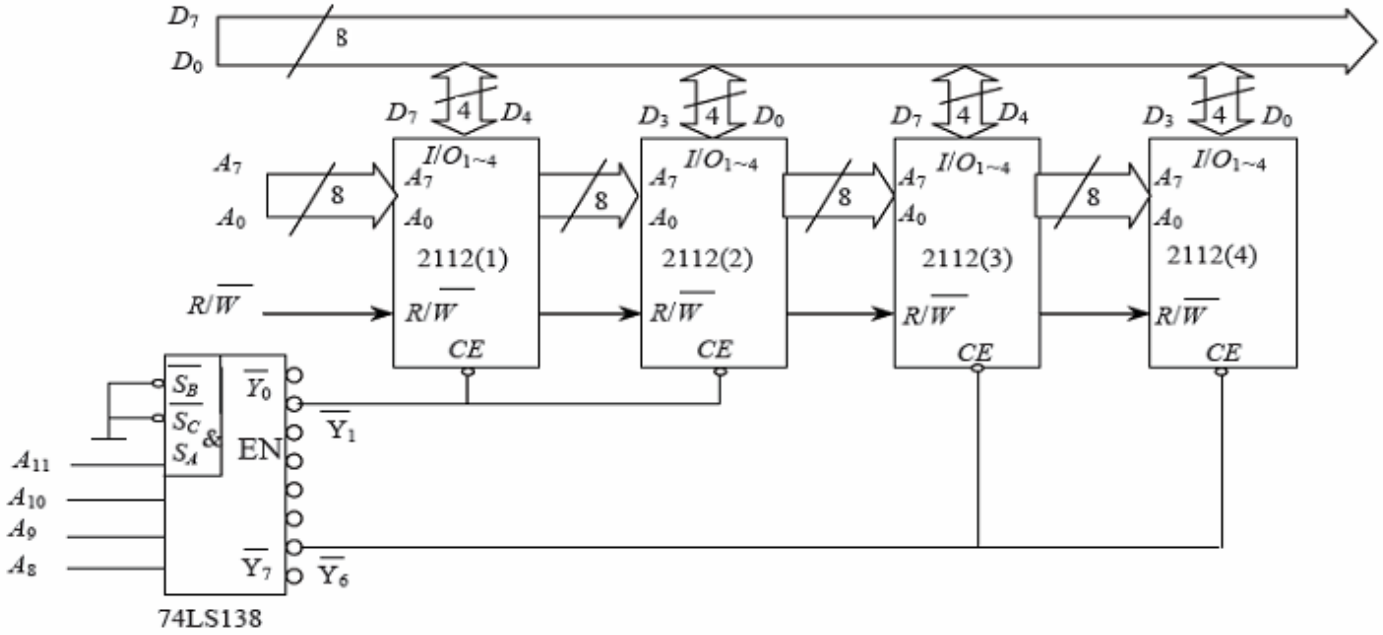
10.5 RAM2112（256×4）组成如题图10.5所示电路。

（1）按图示接法，写出2112(1)至2112(4)的地址范围（用十六进制表示）。（2）按图示接法，内存单元的容量是多少？若要实现2k×8的内存，需要多少片2112芯片？（3）若要将RAM的寻址范围改为B00H~BFFH和C00H~CFFH，电路应做何改动？



两组地址不是连续的

（1）按图示接法，写出2112(1)至2112(4)的地址范围（用十六进制表示）。



解：

(1) RAM2112 (1) (2)片：900H ~ 9FFH

RAM2112 (3) (4)片：E00H ~ EFFH

$A_{11}A_{10}A_9A_8\ A_7A_6A_5A_4\ A_3A_2A_1A_0$

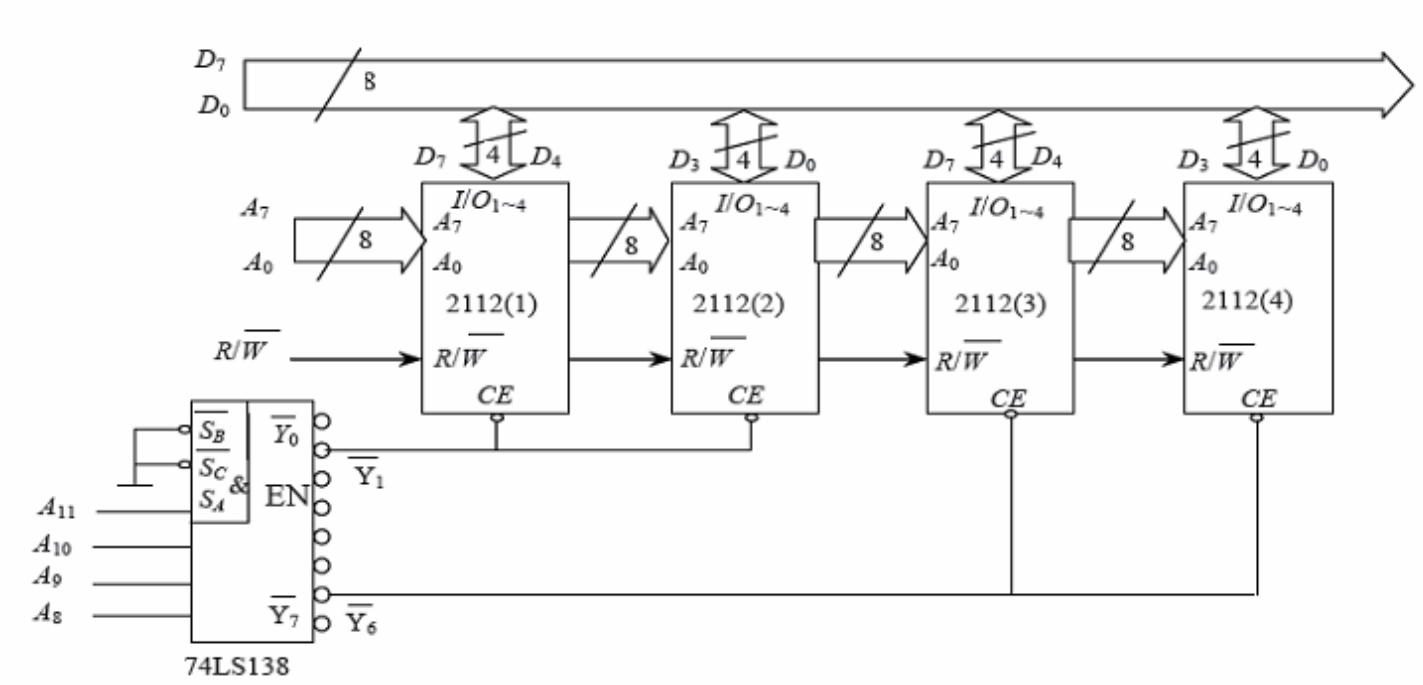
(1) (2): 1 0 0 1 0 0 0 0 0 0 0 0 **900H**

1 0 0 1 1 1 1 1 1 1 1 1 **9FFH**

(3) (4): 1 1 1 0 0 0 0 0 0 0 0 0 **E00H**

1 1 1 0 1 1 1 1 1 1 1 1 **EFFH**

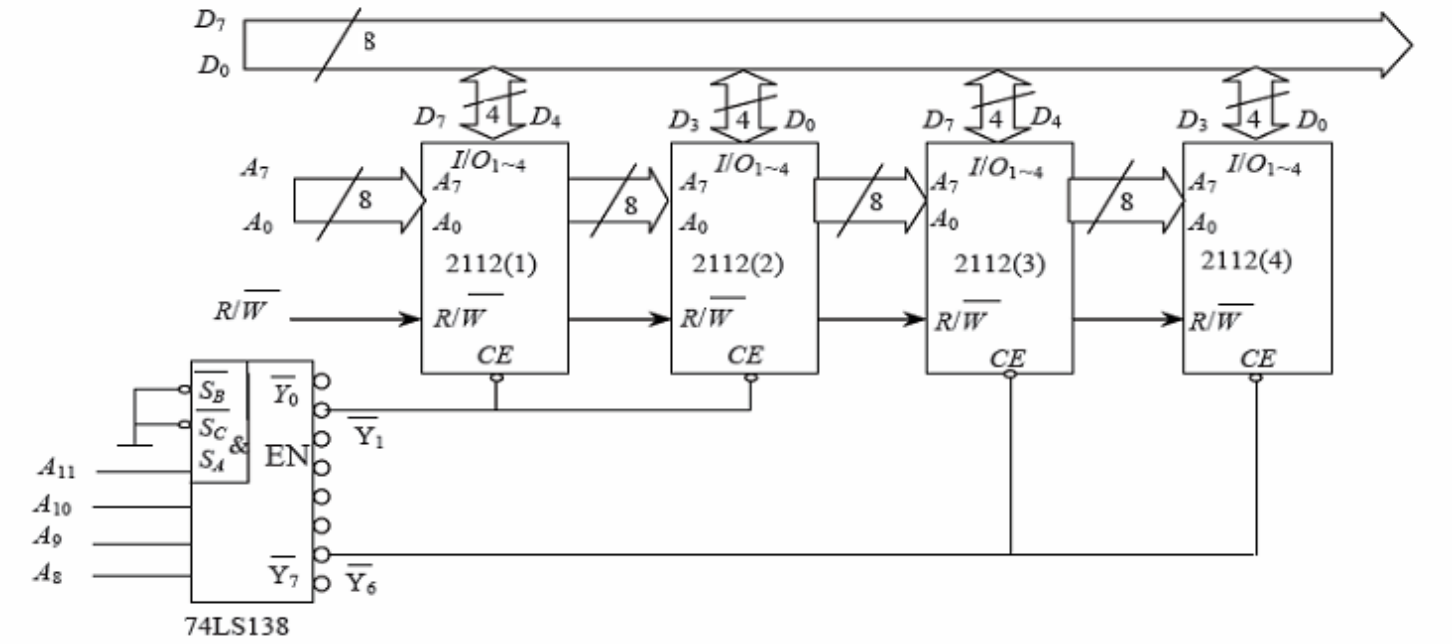
（2）按图示接法，内存单元的容量是多少？若要实现2k×8的内存，需要多少片2112芯片？



(2) 图中接法内存容量：

$$2^8\times4\times2\times2=2^9\times8=512\times8$$

若实现2K×8需要 $\dfrac{2\times2^{10}\times8}{2^8\times4}=16$片 2112RAM



(3) 若寻址范围改为 B00H~BFFH 和 C00H~CFFH 电路改为：

$\overline{Y_1} \rightarrow \overline{Y_3}$

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

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

$11)_{10} \rightarrow$ B

$\overline{Y_6} \rightarrow \overline{Y_4}$

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

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

$12)_{10} \rightarrow$ C

§10.4 可编程逻辑器件

Programmable Logic Device

10.4.1 可编程逻辑器件概述

Programmable Logic Device，简称PLD

按逻辑功能的特点，将数字集成电路分类

- 通用型
- 专用型

通用型：

中、小规模数字集成电路（如74系列、CC400系列等）

逻辑功能较简单，而且固定不变。有很强的通用性。

专用型：为某种专门用途而设计的集成电路 ASIC
（Application Specific Integrated Circuit）

所设计的数字系统做成一片大规模集成电路，不仅减小电路的体积、重量、功耗，而且使电路的可靠性大为提高。

然而，在用量不大的情况下，设计和制造这样的专用集成电路不仅成本很高，而且设计、制造的周期也嫌太长。

可编程逻辑器件PLD成功地解决这个矛盾。PLD虽然是作为一种通用器件生产的，但它的逻辑功能是由用户通过对器件编程来设定的。可以由设计人员自行编程把一个数字系统“集成”在一片PLD上，而不必去请芯片制造厂商设计和制作专用集成电路芯片。

PLD包括：

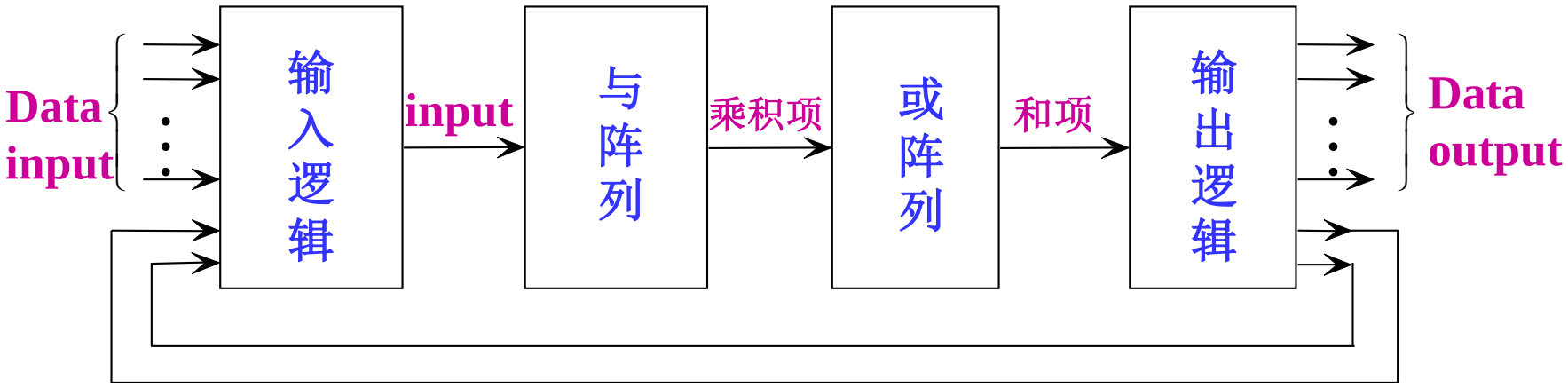
- Filed Programmable Logic Array 现场可编程逻辑阵列 FPLA
- Programmable Array Logic 可编程阵列逻辑 PAL
- Generic Array Logic 通用阵列逻辑 GAL
- Complex Programmable Logic Device 复杂可编程逻辑器件 CPLD
- Field Programmable Gate Array 现场可编程门阵列 FPGA

集成度较高，称为高密度PLD。

10.4.2 可编程逻辑器件的基本结构

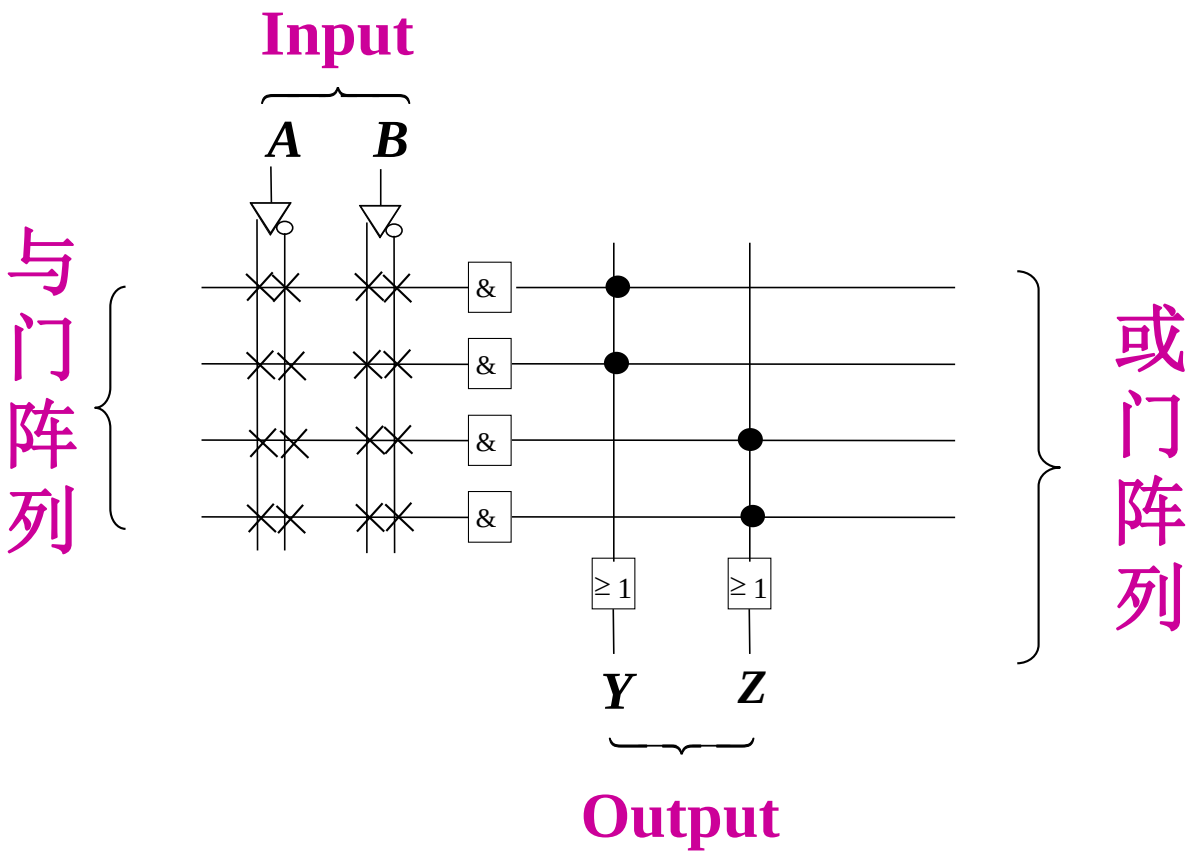
Structure of PLD

1. PLD 基本结构



PLD
- 与阵列可编程
- 或阵列可编程
- 与、或阵列都可以编程

2. PLD 电路描述



硬件连接

编程连接

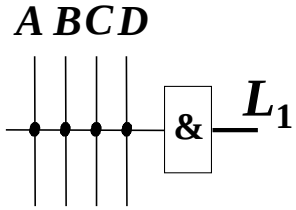
编程断开

逻辑器件（导通/截至）

熔丝连接（烧断/保留）

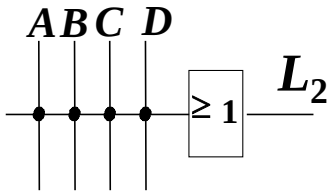
基本门的 PLD 表示

4-input AND gate



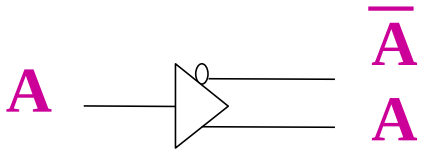
$L_1 = ABCD$ 积项

4-input OR gate



$L_2 = A + B + C + D$ 和项

输入端互补输出缓冲器



例：用PAL实现逻辑函数 $F_1 = A \oplus B \oplus C$ 及其反函数 F_2。

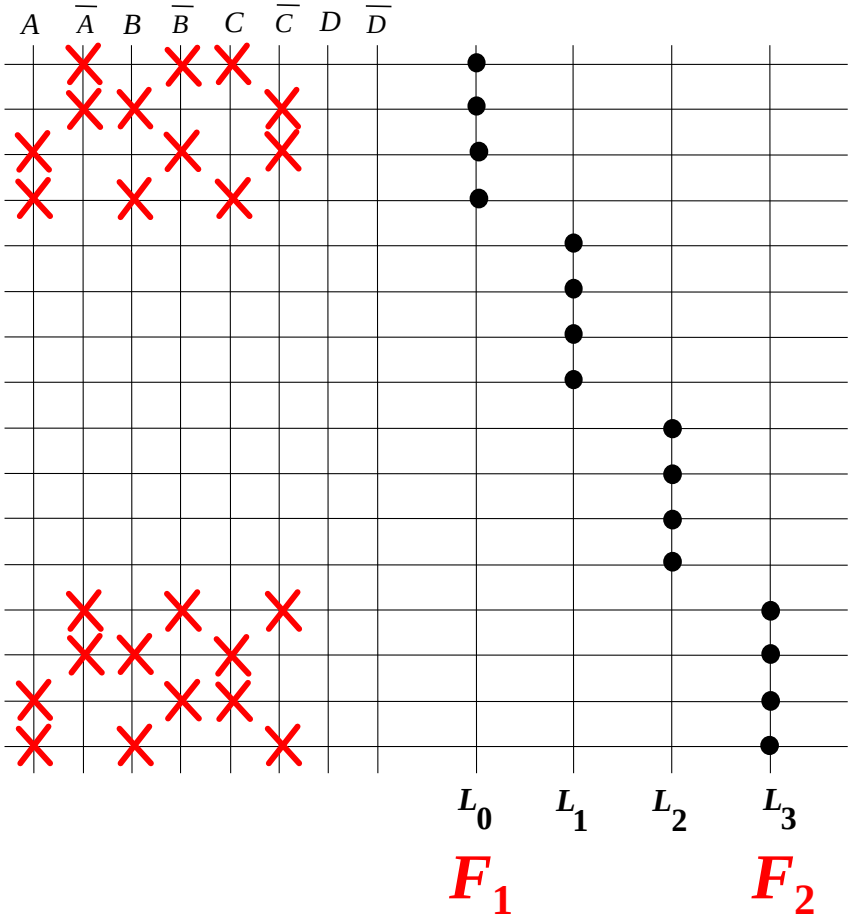
用“×”表示编程连接，画出简化内部结构。

PAL是一种与阵列可编程的PLD器件，按逻辑函数与或表达式中乘积项可确定与阵列的编程连接。实际应用中，用编程程序，输入函数表达式，编程程序会自动对PAL进行连接。

$$F_1(A,B,C) = \sum m(1,2,4,7)$$

$$F_2(A,B,C) = \sum m(0,3,5,6)$$

设L_0输出F_1，L_3输出F_2，编程后内部结构如右图。



2. 通用阵列逻辑电路 GAL （Generic Array Logic）

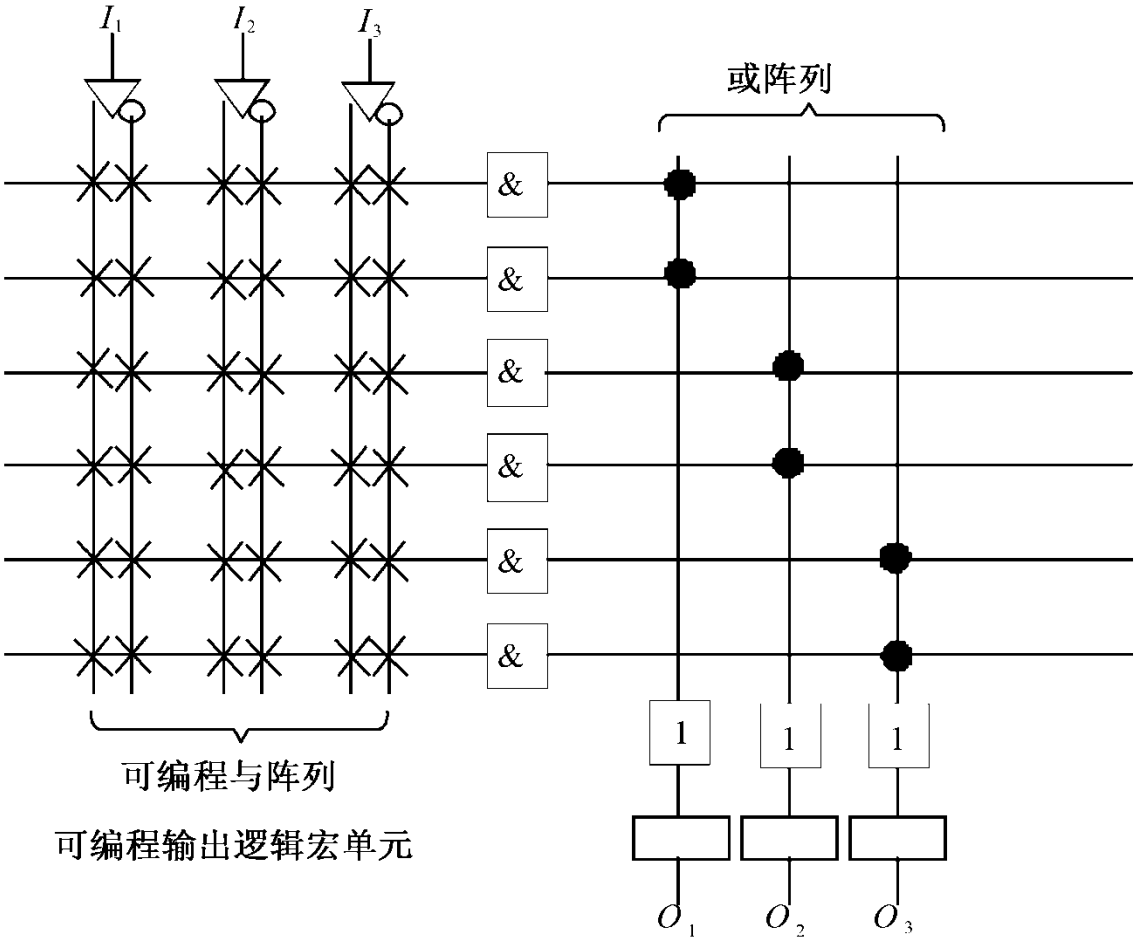
1985年由LATTICE公司推出，采用电可擦除的CMOS（E^2CMOS）制作。

组成

可编程与阵列

不可编程或阵列

可编程输出逻辑宏单元OLMC



PAL和GAL逻辑资源少，结构过于简单不能满足实际需要

作业:

10. 4

10. 5

10. 8

10. 18

10. 19

10.20 F1

10.21 F1