

《数字逻辑》 Digital Logic

存储器和可编程 逻辑器件

北京工业大学软件学院
王晓懿

教学要求

1.了解二极管、晶体管ROM的基本结构和存储单元结构；会用ROM实现组合逻辑函数。

2.熟悉RAM的结构和操作过程；了解RAM的扩展方式。

概述

半导体存储器是一种能存储大量二值信息的半导体器件。

按存储功能分

只读存储器（ROM）

随机存储器（RAM）

按制造工艺分

双极性

MOS型

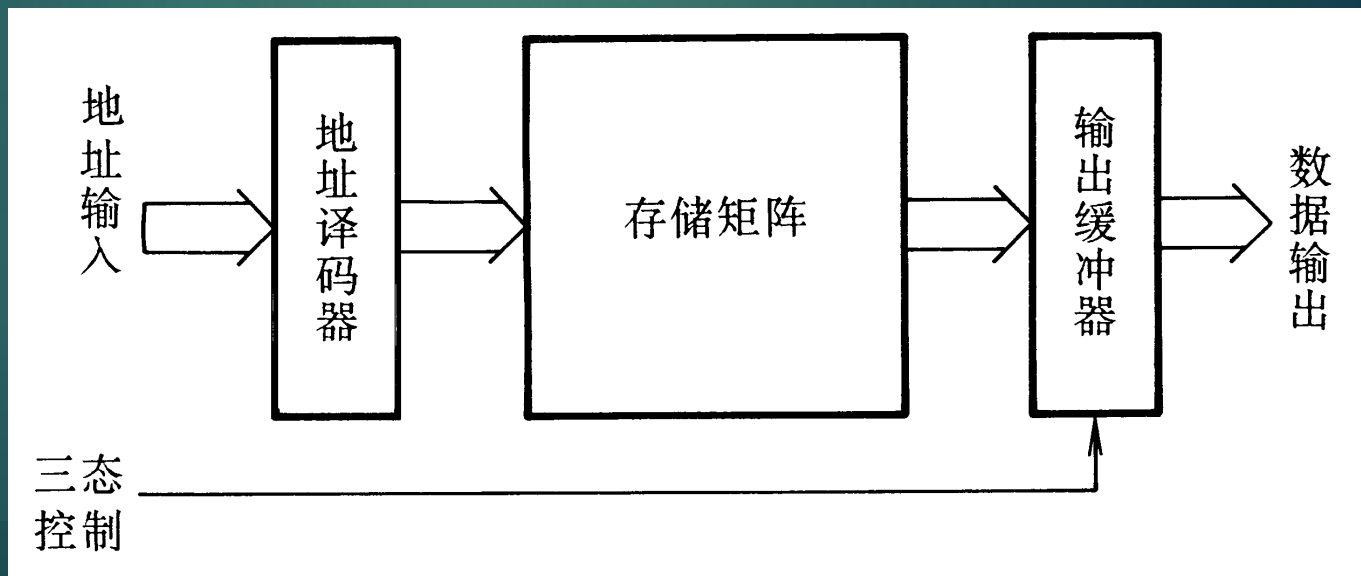
只读存储器 (ROM)

Read Only Memory

优点： 电路结构简单，断电后数据不丢失，具有非易失性。

缺点： 只适用于存储固定数据的场合。

电路结构



只读存储器分类:

掩膜ROM: 出厂后内部存储的数据不能改动, 只能读出。

PROM: 可编程, 只能写一次。

EPROM: 用紫外线擦除, 擦除和编程时间较慢, 次数也不宜多。

E²PROM: 电信号擦除, 擦除和写入时需要加高电压脉冲, 擦、写时间仍较长。

快闪存储器(Flash Memory): 吸收了EPROM结构简单, 编程可靠的优点, 又保留了E²PROM用隧道效应擦除的快捷特性, 集成度可作得很高。

随机存储器 (RAM)

Random Access Memory

优点： 读、写方便,使用灵活。

缺点： 一旦停电所存储的数据将随之丢失（易失性）。

基本结构： 地址译码器、存储矩阵和读\写控制电路构成。

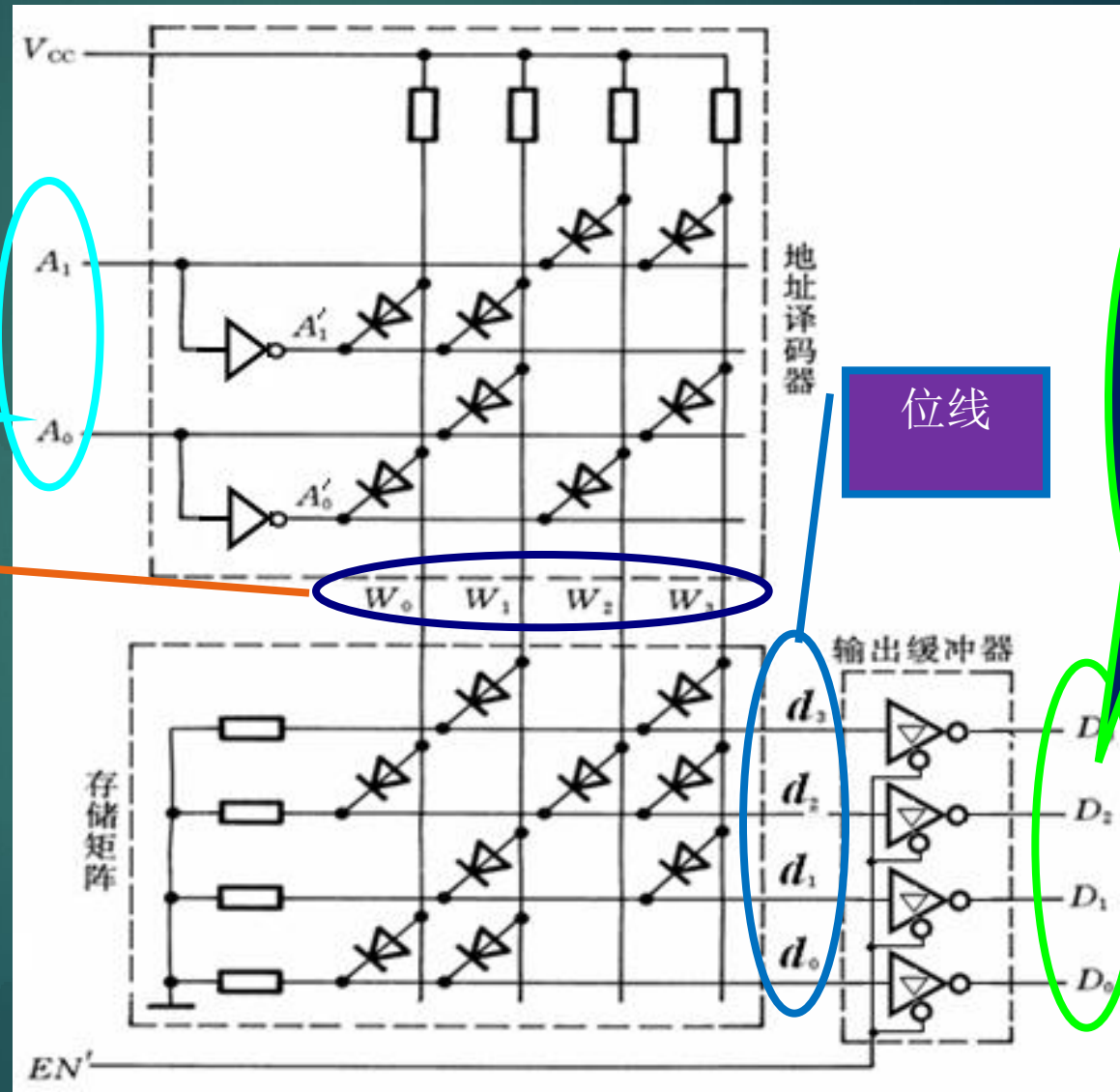
存储容量的扩展

存储容量 = 字数 × 位数

地址输入端

字线

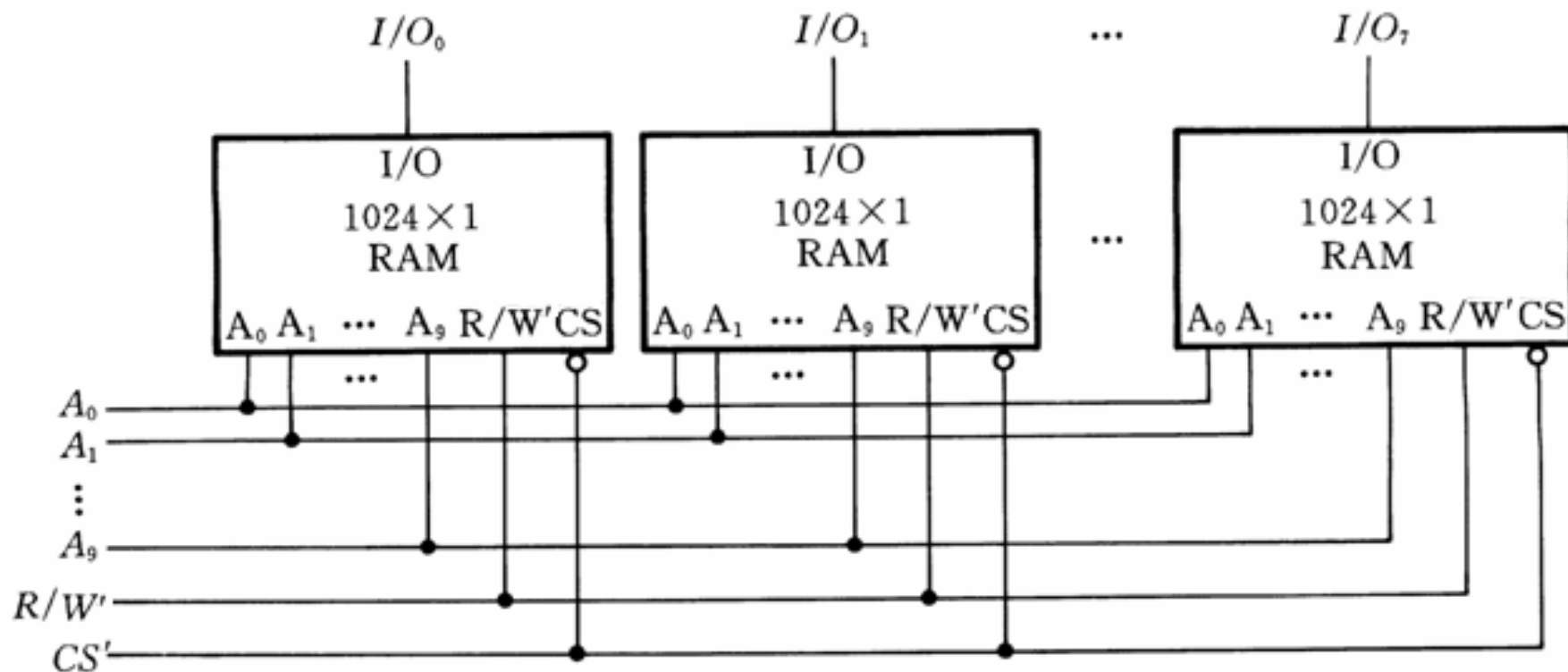
存储容量 = $2^2 \times 4 = 4 \times 4$



位线

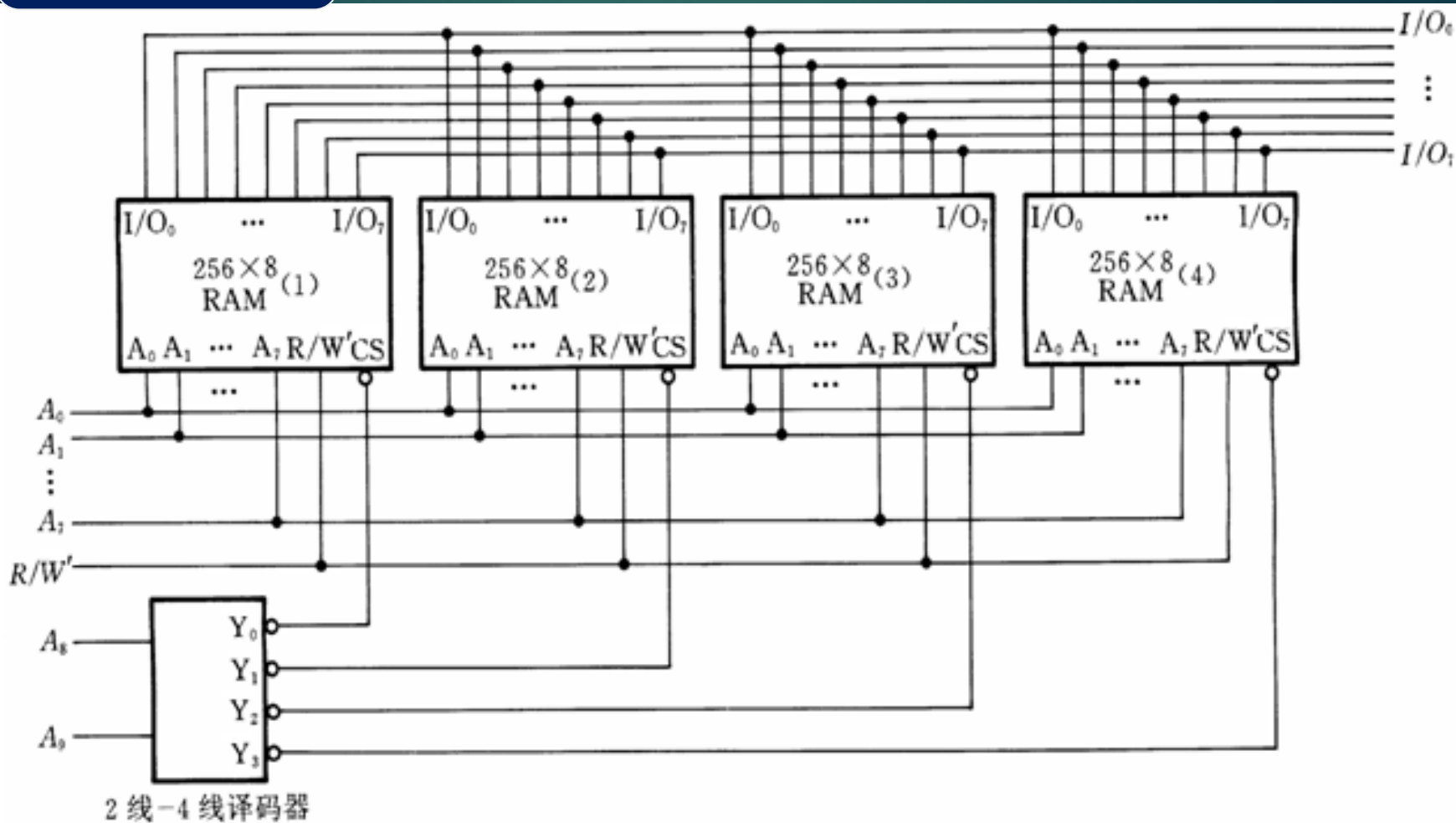
数据输出端

位扩展



8片 1024×1 位RAM接成 1024×8 位的RAM。

字扩展



4片 256×8 位的RAM接成 1024×8 位的RAM。

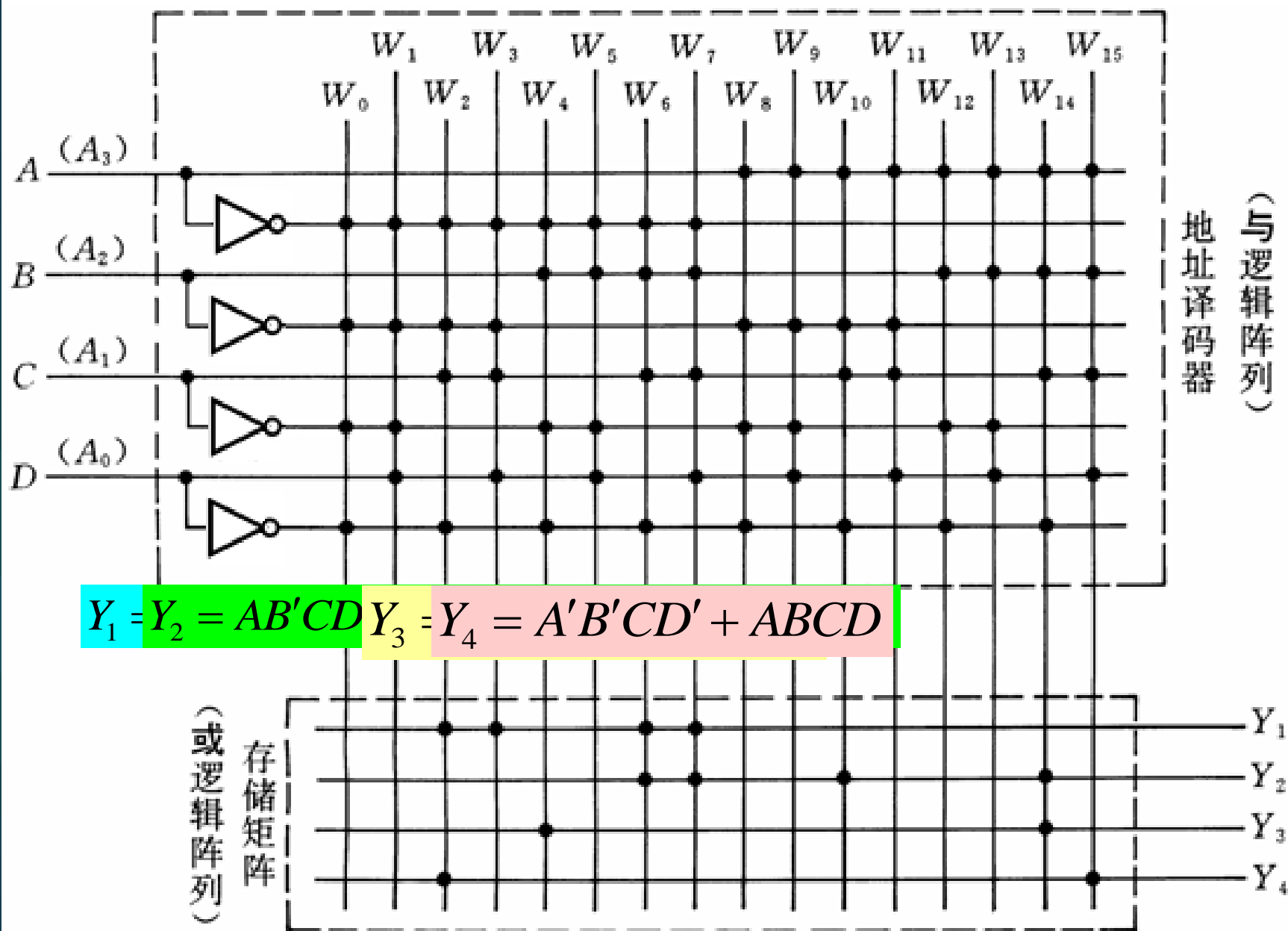
用存储器实现组合逻辑函数

例 试用ROM产生如下一组多输出逻辑函数

$$\begin{cases} Y_1 = A'BC + A'B'C \\ Y_2 = AB'CD' + BCD' + A'BCD \\ Y_3 = ABCD' + A'BC'D' \\ Y_4 = A'B'CD' + ABCD \end{cases}$$

解：化为最小项之和的形式：

$$\begin{cases} Y_1 = A'BCD' + A'BCD + A'B'CD' + A'B'CD \\ Y_2 = AB'CD' + A'BCD' + ABCD' + A'BCD \\ Y_3 = ABCD' + A'BC'D' \\ Y_4 = A'B'CD' + ABCD \end{cases}$$

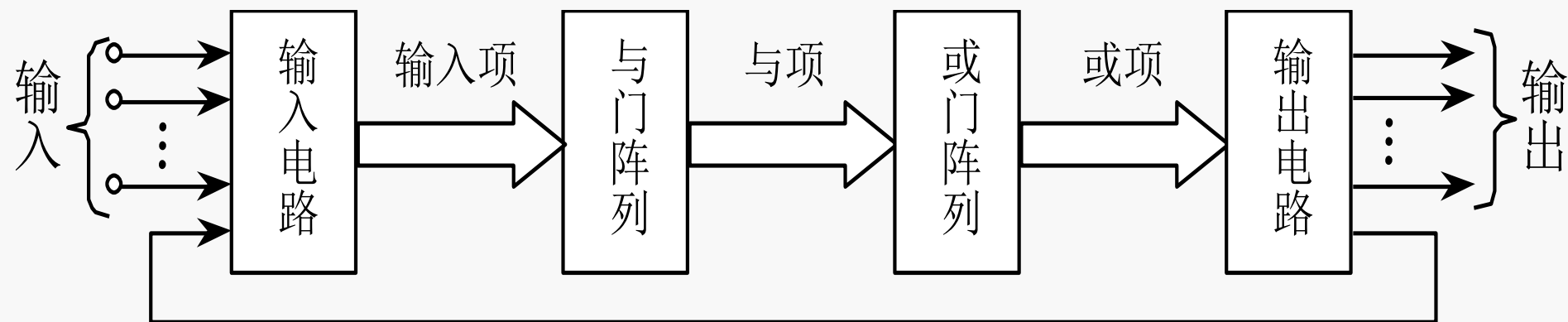




可编程逻辑器件 (PLD)

(Programmable Logic Device)

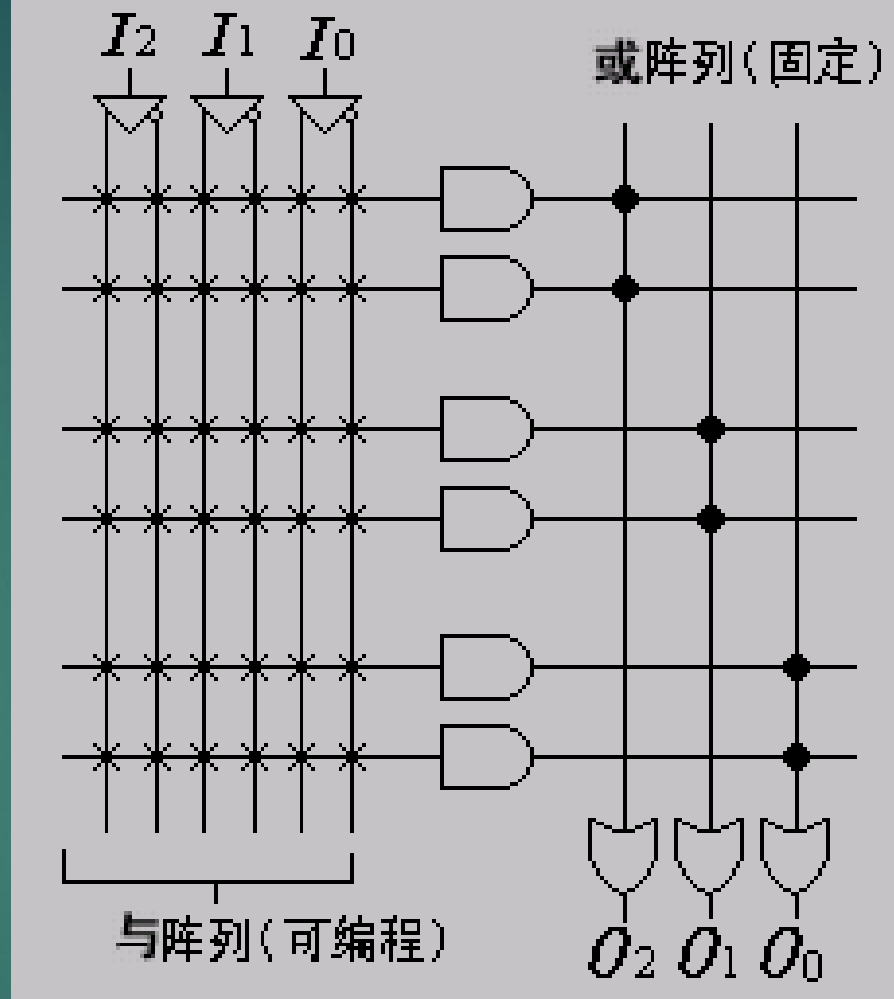
1. PLD的基本结构



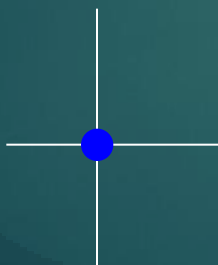
2. 连接方式

- 可编程“接通”单元：
它依靠用户编程来实现“接通”连接。

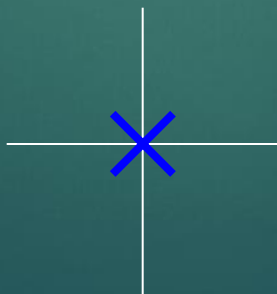
- 可编程“断开”单元：
编程实现断开状态。这种单元又称为被编程擦除单元。



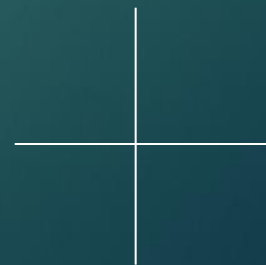
固定
连接:



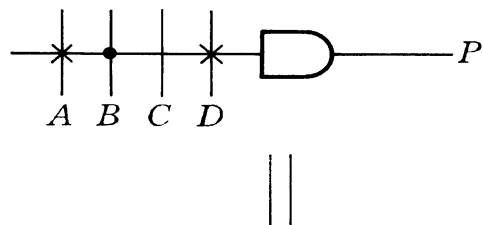
可
编程
连接:



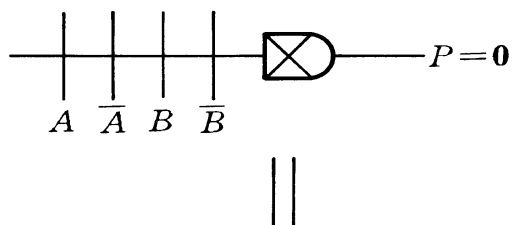
不
连
接:



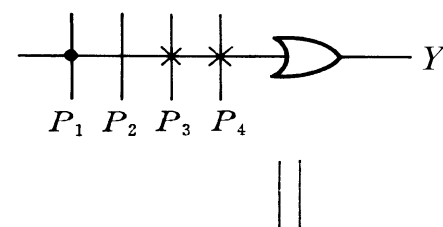
3. PLD的画法



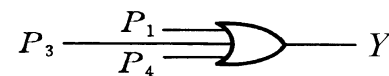
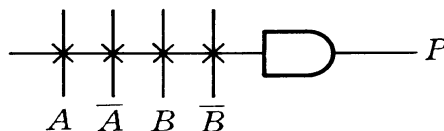
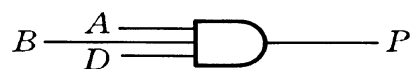
$$P=ABD$$



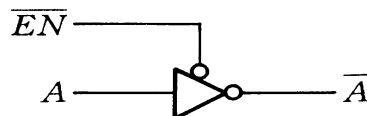
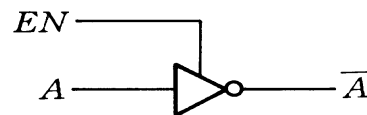
$$P=0$$



$$Y=P_1+P_3+P_4$$



互补输出的缓冲器



三态输出的缓冲器

4. PLD分类

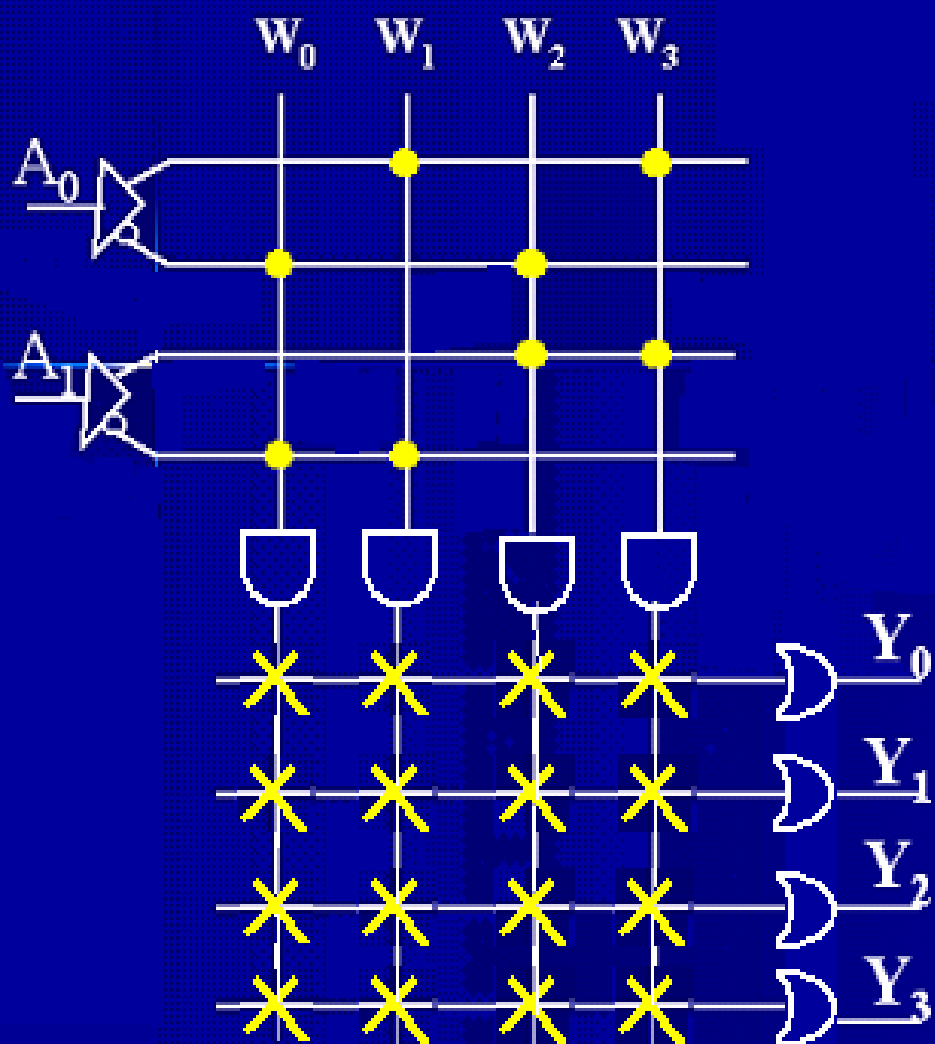
可编程只读存储器

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

可编程逻辑阵列

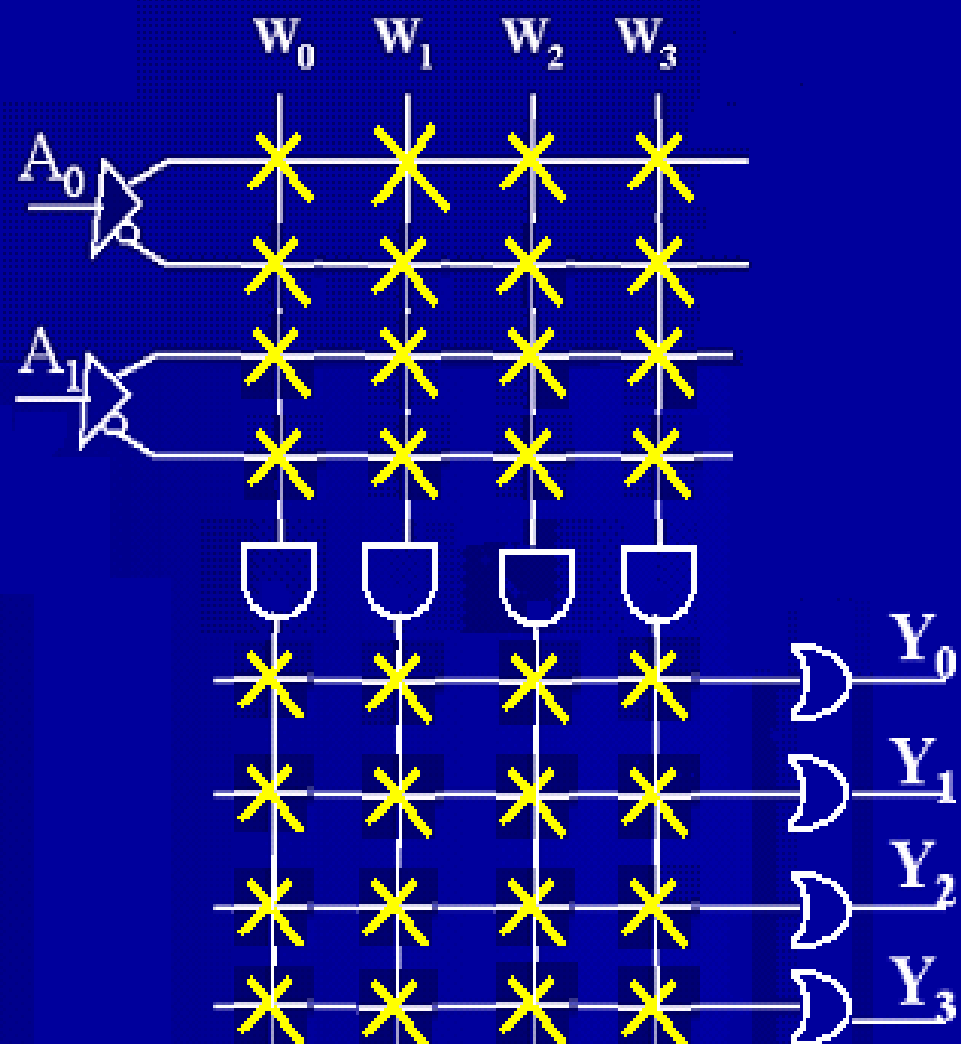
可编程阵列逻辑

通用阵列逻辑



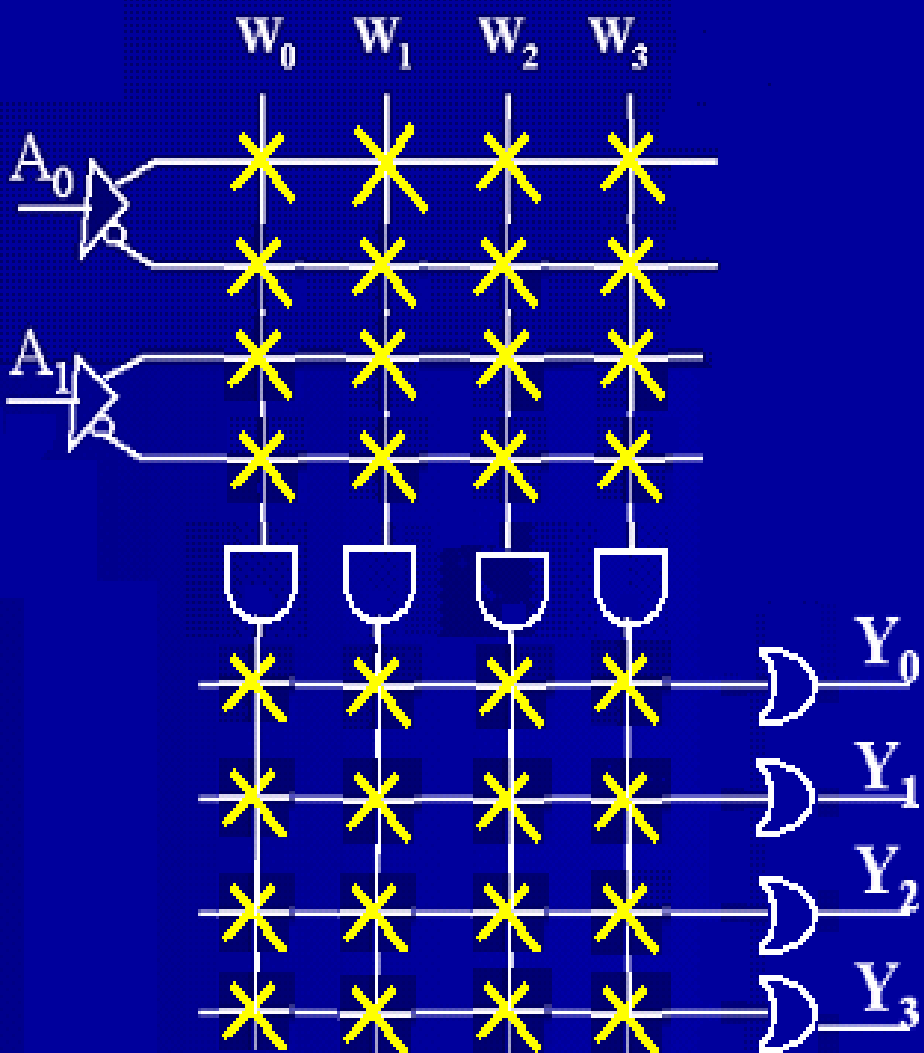
PROM

与阵列固定、或阵列可编程



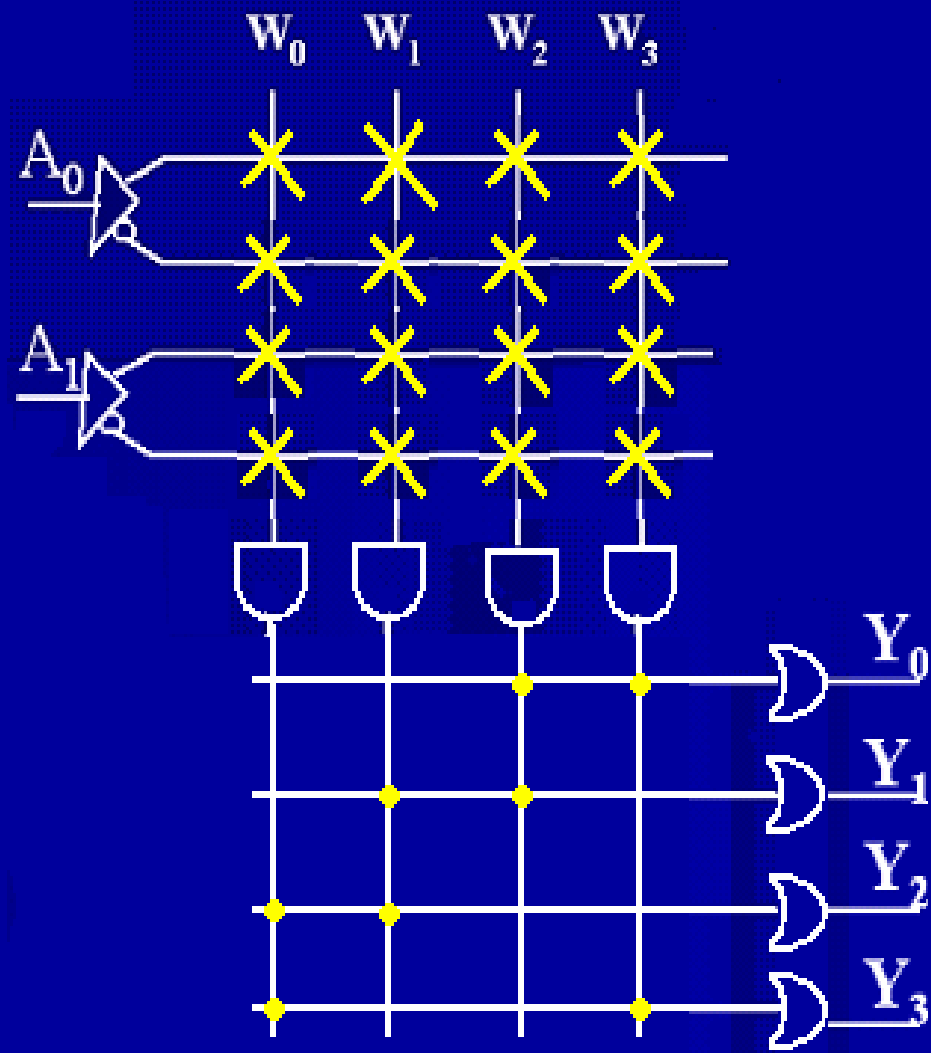
PLA

与、或阵列均可编程



PLA

与、或阵列均可编程



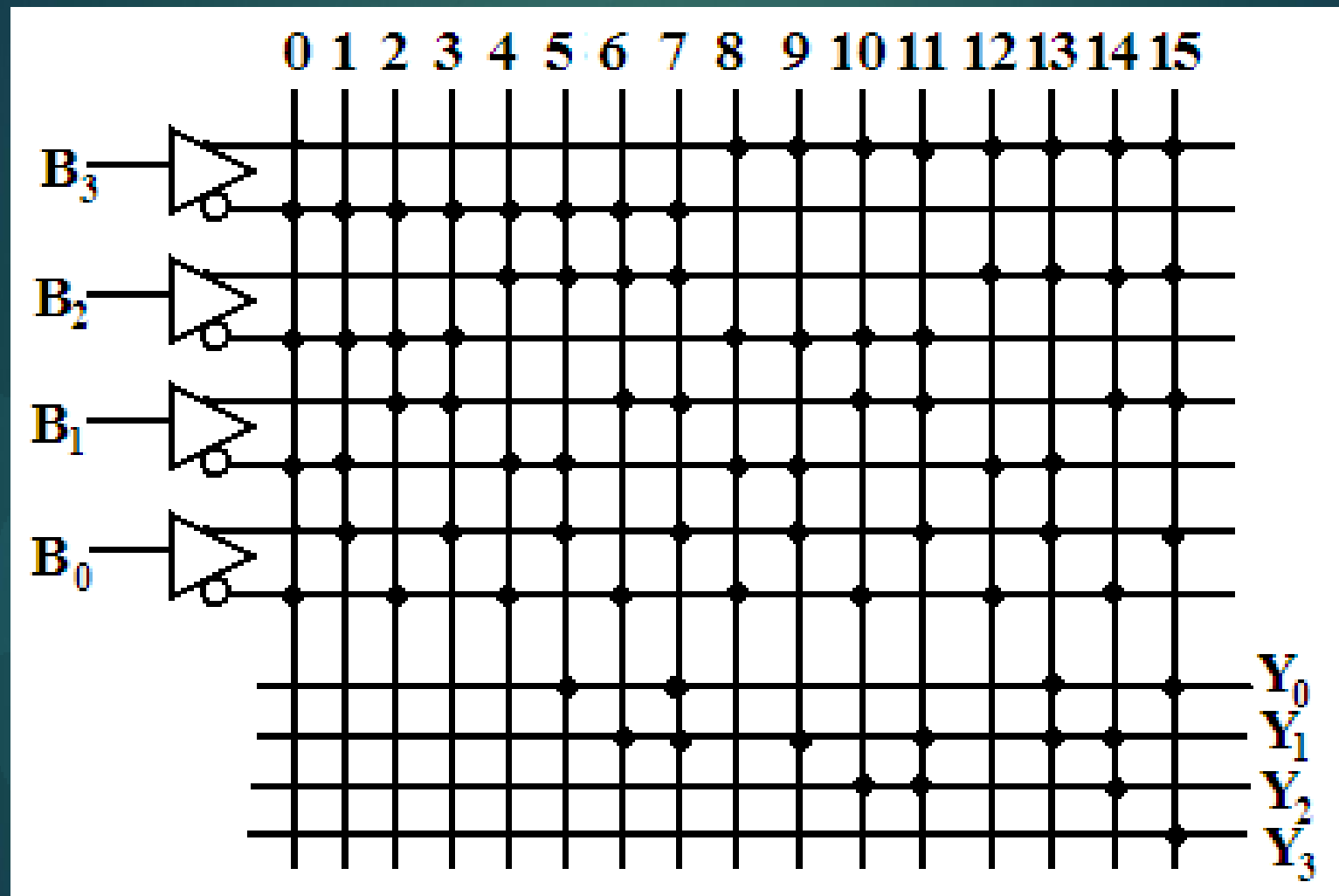
PAL

与阵列可编程、或阵列固定

用PLA实现逻辑函数：

用**ROM**实现逻辑函数是基于该逻辑函数的**最小项表达式** $F=\sum m_i$ ，而用**PLA**实现逻辑函数是基于该逻辑函数的最简与或表达式 $F=\sum P_i$ ，所以用**PLA**来实现逻辑函数比用**ROM**实现逻辑函数更简单、灵活。

例：下图为用ROM构成的逻辑函数的与或阵列图。试设计出与其对应的逻辑函数的PLA与或阵列图。



其真

[illegible]

Y_3 B_1B_0

B_3B_2 00 01 11 10

00				
01				
11			1	
10				

Y_2 B_1B_0

B_3B_2 00 01 11 10

00				
01				
11				1
10			1	1

Y_1 B_1B_0

B_3B_2 00 01 11 10

00				
01			1	1
11		1		1
10		1	1	

Y_0 B_1B_0

B_3B_2 00 01 11 10

00				
01		1	1	
11		1	1	
10				

PLA 与或阵列图

$$\begin{cases} Y_3 = B_3 B_2 B_1 B_0 \\ Y_2 = B_3 B'_2 B_1 + B_3 B_1 B'_0 \\ Y_1 = B_3 B'_1 B_0 + B_3 B'_2 B_0 + B'_3 B_2 B_1 + B_2 B_1 B'_0 \\ Y_0 = B_2 B_0 \end{cases}$$

