

第6章 可编程逻辑器件

PLD: Programmable Logic Device

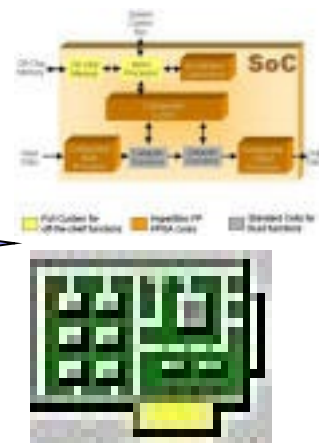
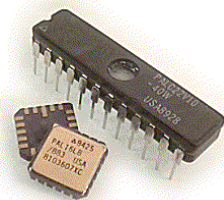
6.1 PLD概述

PLD是厂家作为一种通用型器件生产的半定制电路，用户可以利用软、硬件开发工具对器件进行设计和编程，使之实现所需要的逻辑功能。

一、PLD的基本特点

1. 数字集成电路从功能上有分为通用型、专用型两大类

数字
系统



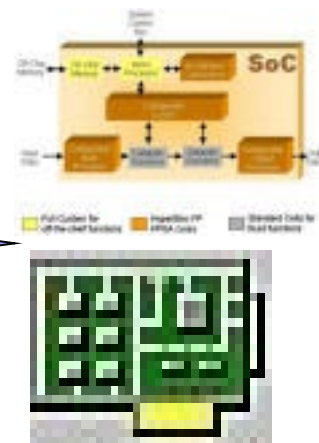
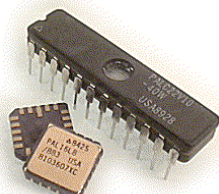
6.1 PLD概述

PLD是厂家作为一种通用型器件生产的半定制电路，用户可以利用软、硬件开发工具对器件进行设计和编程，使之实现所需要的逻辑功能。

一、PLD的基本特点

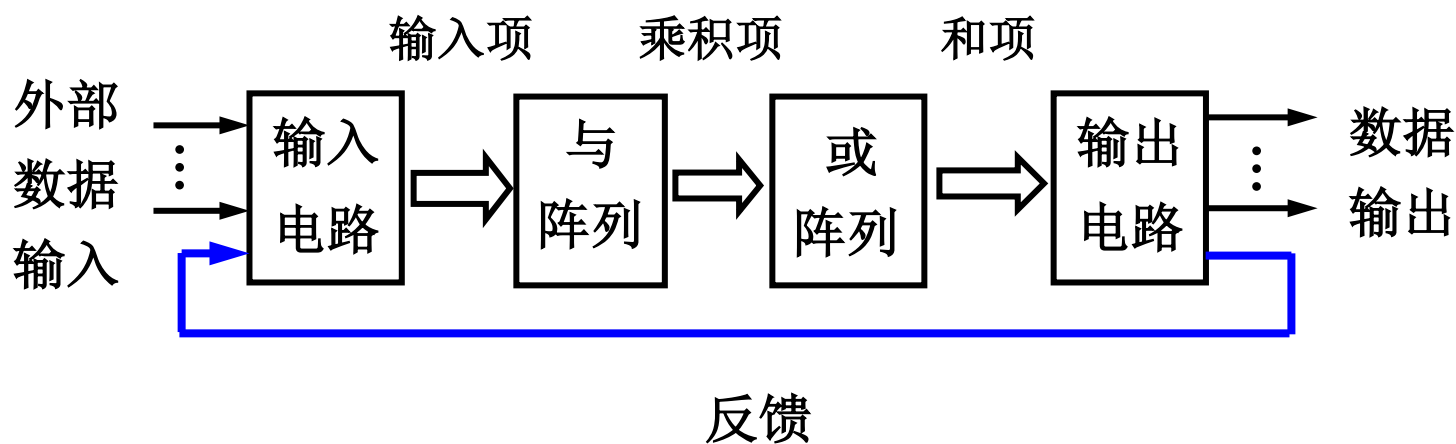
2. PLD的特点：是一种按通用器件来生产，但逻辑功能是由用户通过对器件编程来设定的

数字
系统



二、PLD的结构

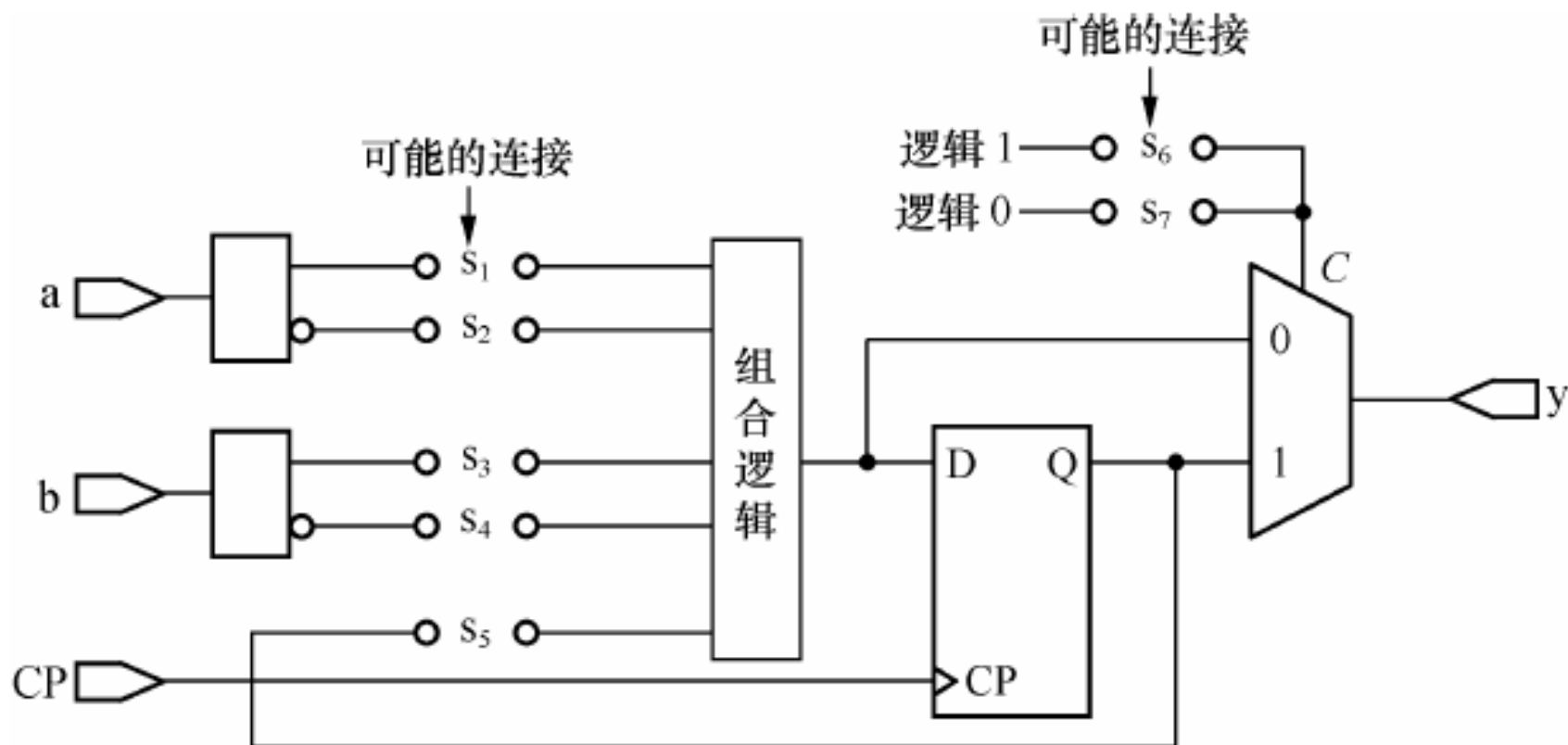
1、传统PLD的总体结构



传统PLD的总体结构

二、PLD的结构

2、可编程功能的实现



可编程功能的实现

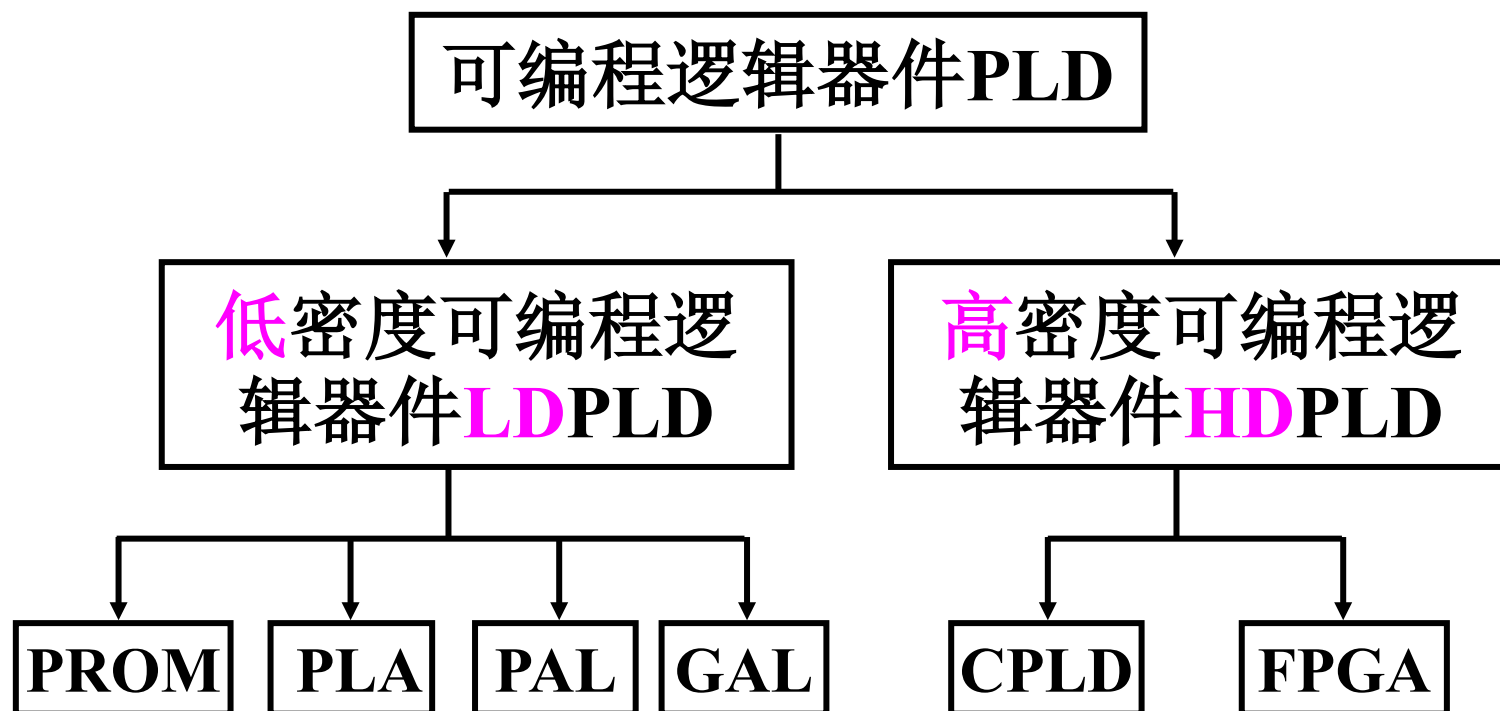
三、PLD的发展和分类

PROM是最早的PLD

1. PAL 可编程逻辑阵列
2. FPLA 现场可编程阵列逻辑
3. GAL 通用阵列逻辑
4. EPLD 可擦除的可编程逻辑器件
5. FPGA 现场可编程门阵列
6. ISP-PLD 在系统可编程的PLD

三、PLD的发展和分类

按集成度分类



三、PLD的发展和分类

按集成度分类

名称	与阵列	或阵列	输出部分
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	可配置

LDPLD的分类与结构特点

四、PLD的开发过程

PLD均需离线进行编程操作，使用开发系统

开发系统

1. 硬件：计算机+编程器
2. 软件：开发环境（软件平台）

VHDL, Verilog

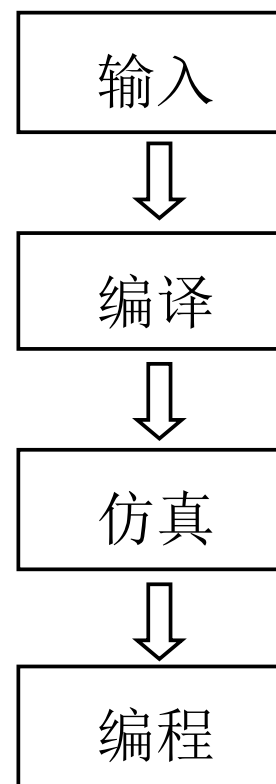
真值表，方程式，电路逻辑图（Schematic）

状态转换图（FSM）

四、PLD的开发过程

步骤

- 抽象（系统设计采用Top-Down的设计方法）
- 选定PLD
- 选定开发系统
- 编写源程序（或输入文件）
- 调试，运行仿真，产生下载文件
- 下载
- 测试



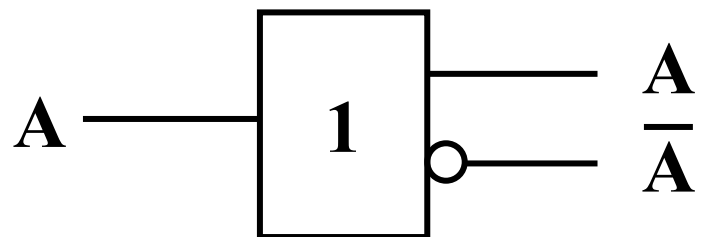
PLD的开发流程示意图

五、PLD的表示方法

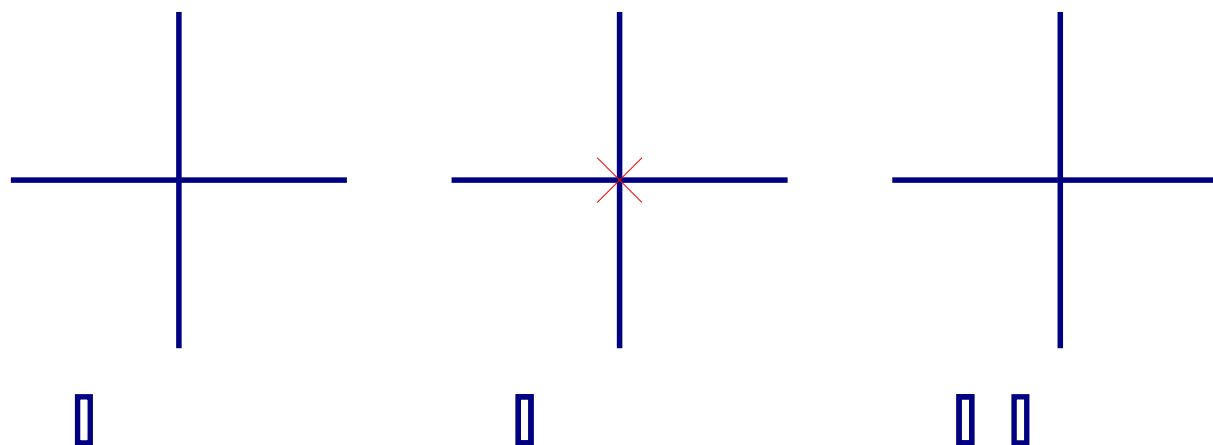
一、输入缓冲电路

二、门电路及导线连接表示

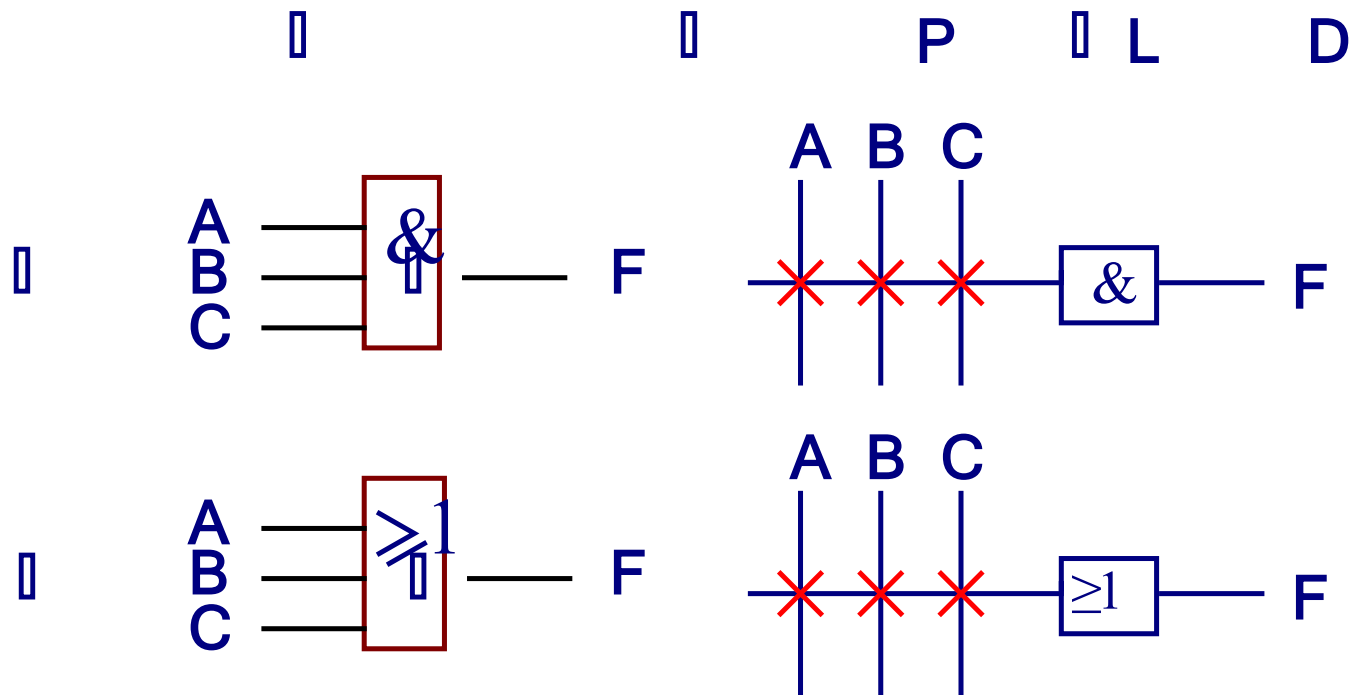
三、多路选择器



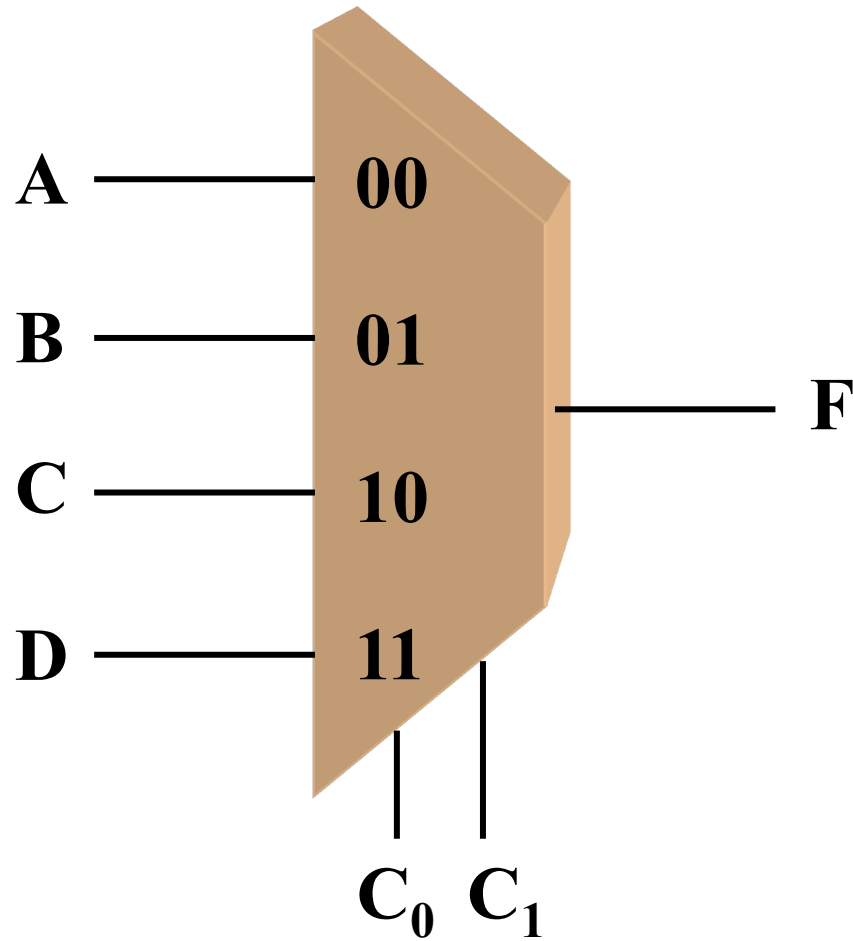
PLD输入缓冲电路



连接方式表示



PLD中的与门、或门表示



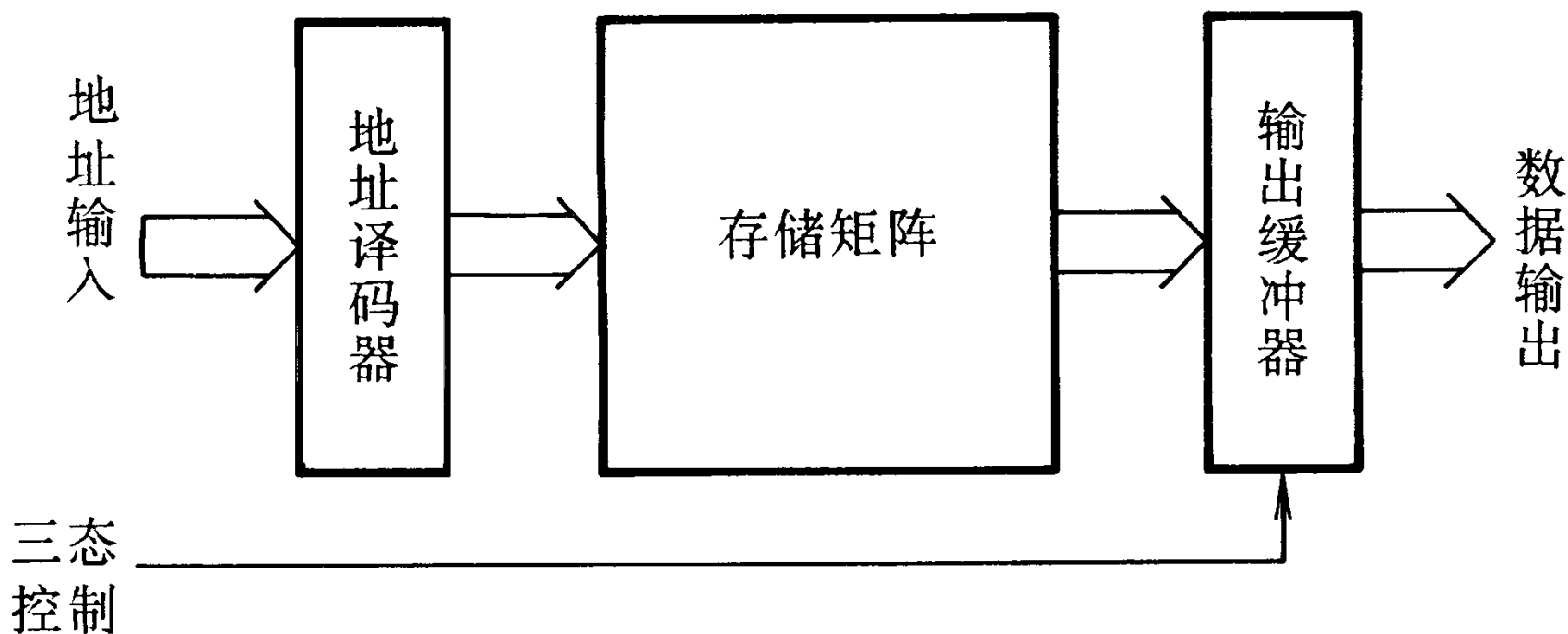
PLD中的多路选择器

6.2 可编程只读存储器（PROM）

6.2.1 PROM的结构和功能

一、基本结构：

地址译码器、存储单元矩阵、输出电路



6.2 可编程只读存储器（PROM）

6.2.1 PROM的结构和功能

一、基本结构：

二、工作原理：

逻辑关系：

译码器部分的输出变量和输入变量（包括原变量和反变量）构成“与”的关系。

存储矩阵和输出电路部分的输出变量和存储矩阵的输入变量构成“或”的关系。

6.2 可编程只读存储器 (PROM)

6.2.2 ROM的应用

2.实现组合逻辑函数

例6.2.1 试用ROM实现如下组合逻辑函数。

$$F_1 = AB + \overline{A}C$$

$$F_2 = AB + \overline{B}C$$

解：首先应将以上两个逻辑函数化成由最小项组成的标准“与-或”式,即

$$F_1 = ABC + AB\overline{C} + \overline{A}BC + \overline{A}\overline{B}C$$

$$F_2 = ABC + AB\overline{C} + \overline{A}BC + \overline{A}\overline{B}C$$

6.2 可编程只读存储器 (PROM)

6.2.2 ROM的应用

2.实现组合逻辑函数

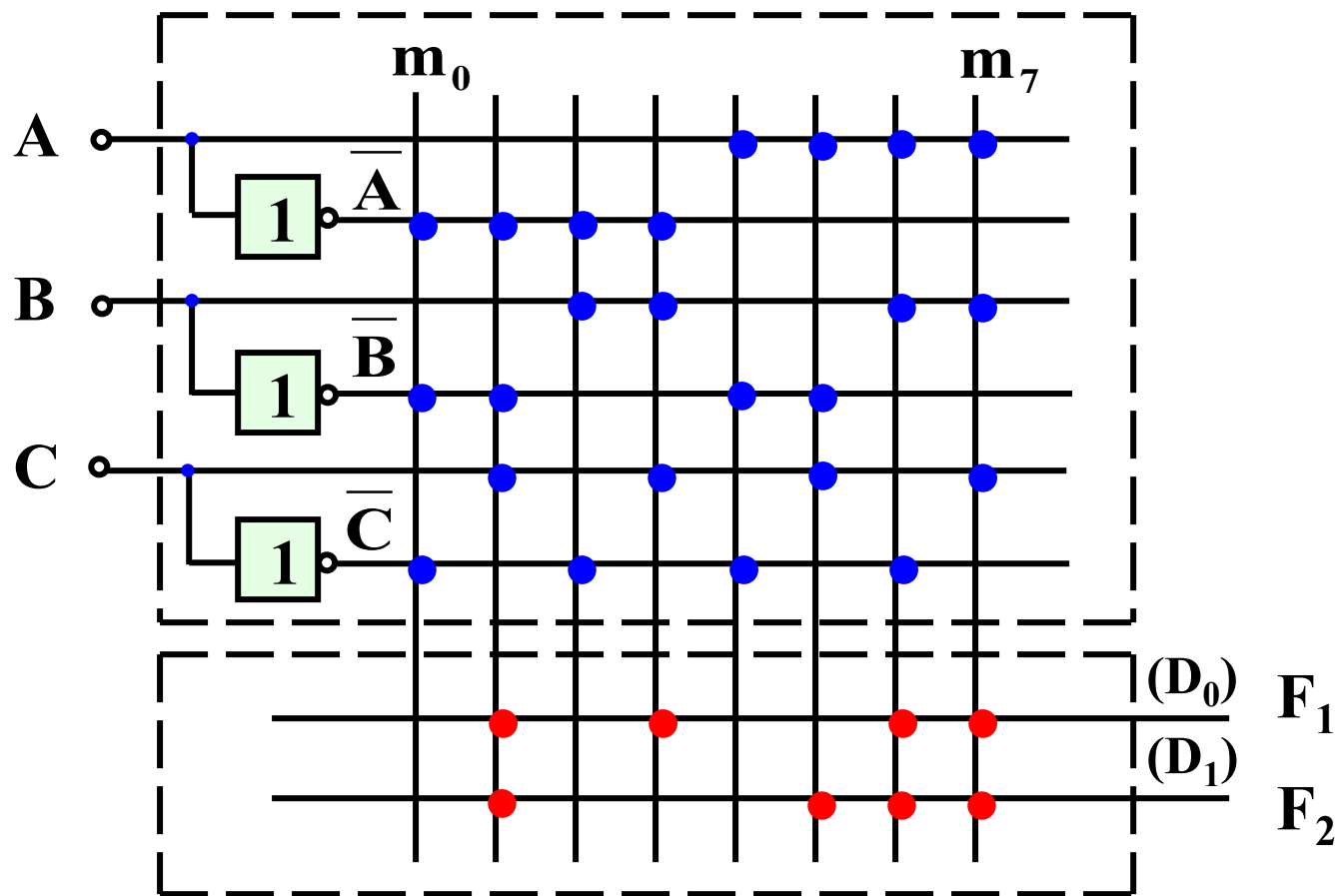
例6.2.1 试用ROM实现如下组合逻辑函数。

$$F_1 = AB + \overline{A}C$$

$$F_2 = AB + \overline{B}C$$

解：采用有3位地址码、2位数据输出的8字节×2位ROM。将A、B、C3个变量分别接至地址输入端A₂A₁A₀。按逻辑函数要求存入相应数据,即可在数据输出端D₀、D₁得到F₁和F₂,其

ROM 阵列如图所示



用***ROM***产生:


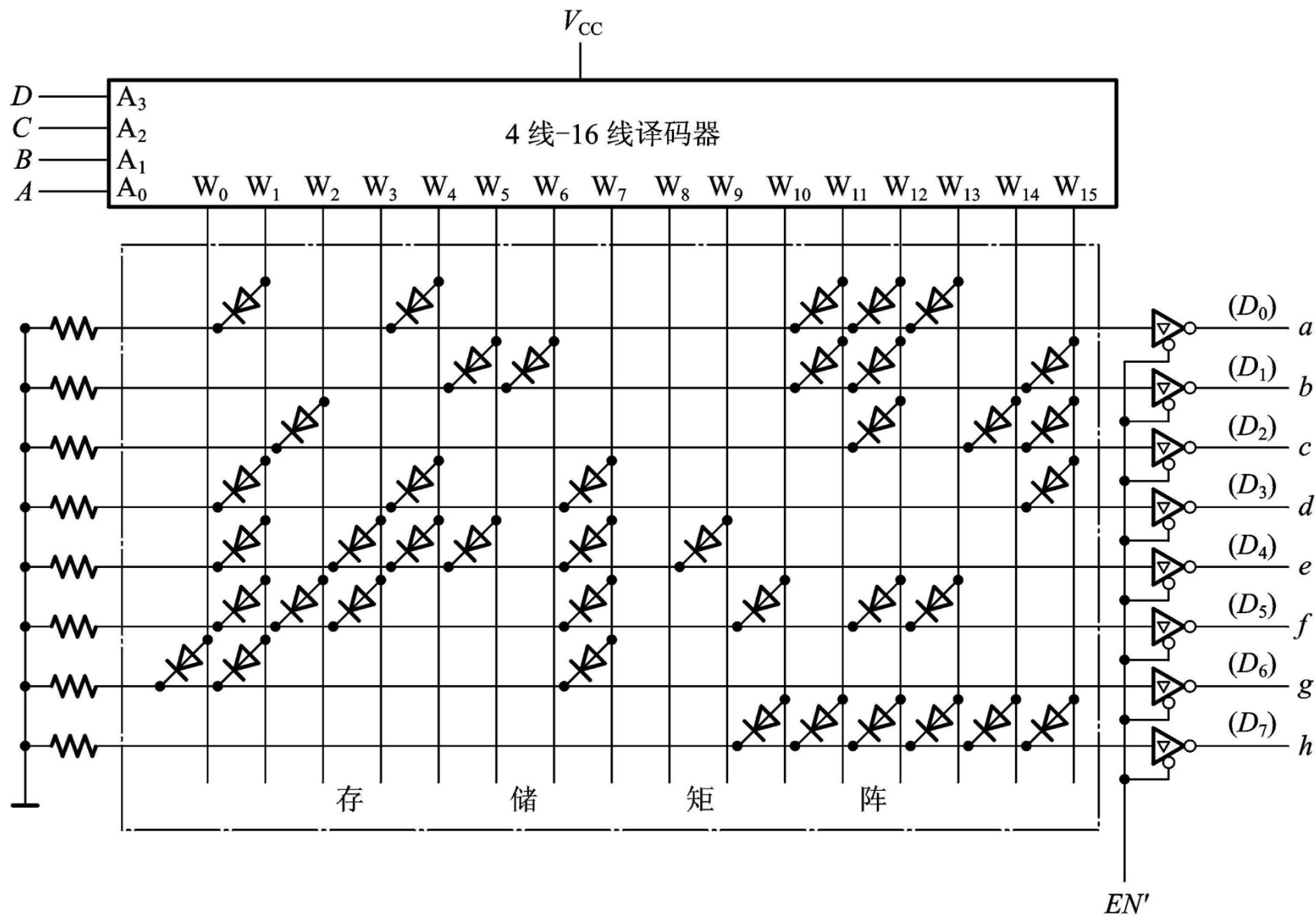


Figure 1 illustrates a logic diagram for a 4x16 address decoder. The diagram is divided into two main sections: an address decoder (top) and a storage matrix (bottom).

The address decoder section consists of a 4x16 grid of AND gates. The columns are labeled W_0 through W_{15} . The rows are labeled W_0 , W_2 , W_4 , and W_6 . Each AND gate has two inputs: one from the address lines and one from the inverted address lines. The output of the AND gates is connected to the output lines Y_0 through Y_3 .

The storage matrix section consists of a 4x16 grid of OR gates. The columns are labeled W_0 through W_{15} . The rows are labeled Y_0 , Y_2 , Y_3 , and Y_4 . Each OR gate has two inputs: one from the address lines and one from the inverted address lines. The output of the OR gates is connected to the output lines Y_0 through Y_3 .



6.3 可编程逻辑阵列 (PLA)

6.3.1 PLA的结构与应用

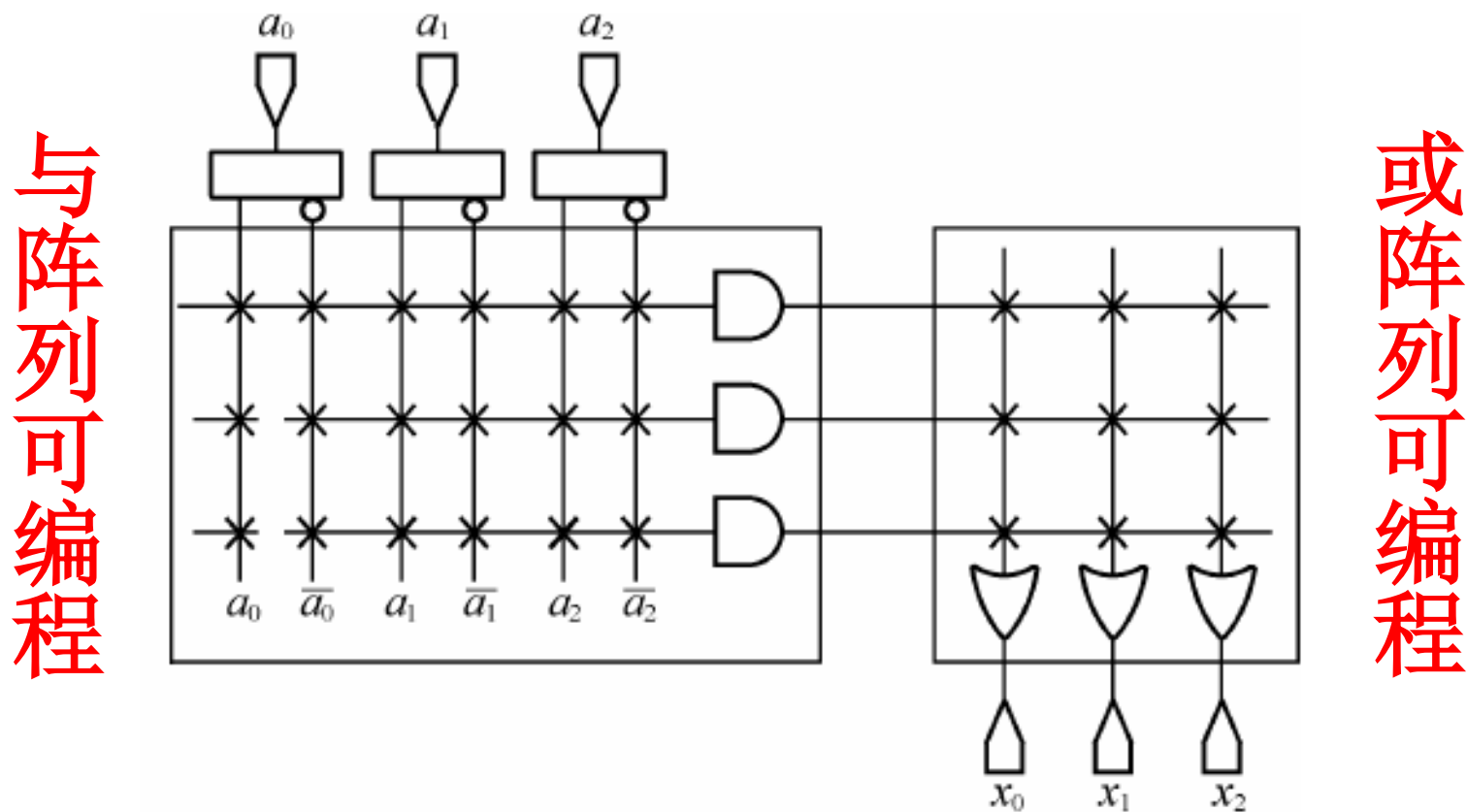


图6.3.1

PLA应用举例

例 用PLA器件实现函数

$$F_1(A_2, A_1, A_0) = \sum m(3, 4, 6, 7),$$

$$F_2(A_2, A_1, A_0) = \sum m(0, 2, 3, 4, 7)。$$

解： 用PLA器件实现，需3个输入端，2个输出端。

用卡诺图法化简，得出 F_1 、 F_2 的最简与或式：

$$F_1(A_2, A_1, A_0) = A_2 \overline{A_0} + A_1 A_0$$

$$F_2(A_2, A_1, A_0) = A_1 A_0 + \overline{A_2} A_1 + \overline{A_1} \overline{A_0}$$

相应的实现电路如图所示。

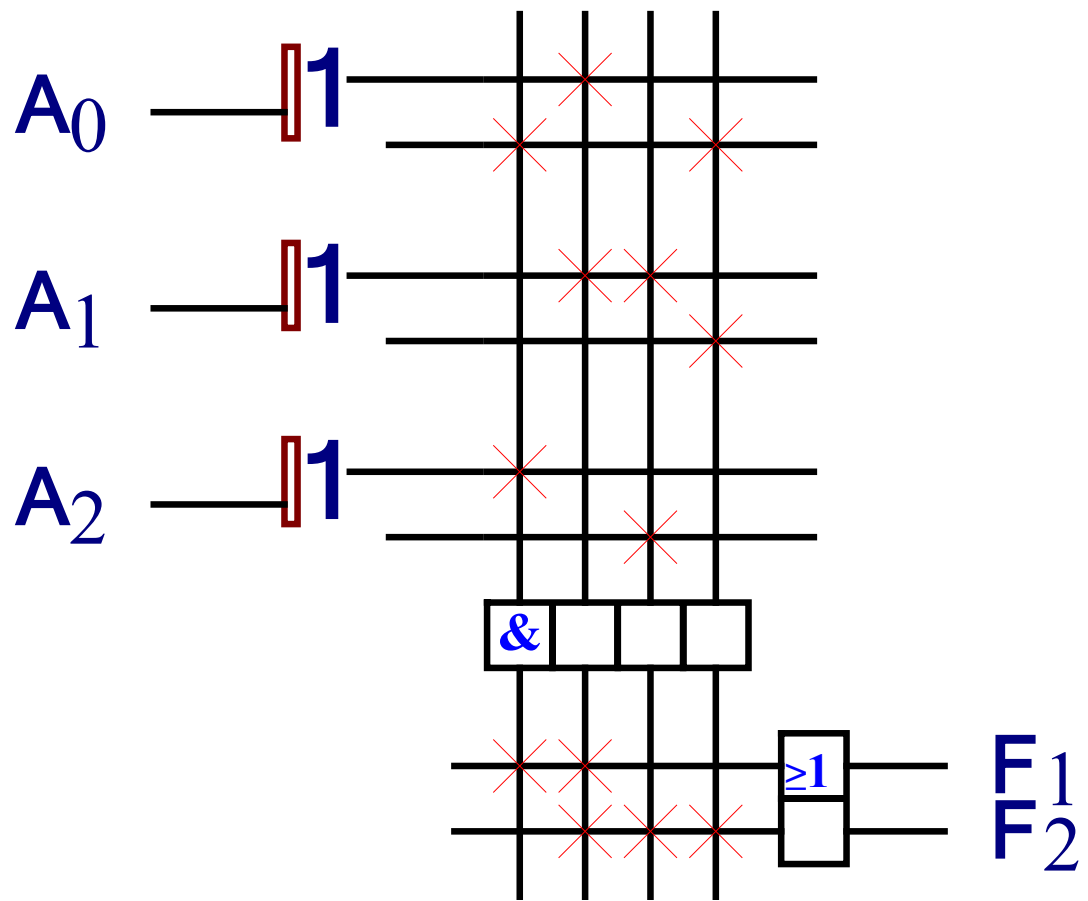


图 用PLA实现组合函数的设计

6.3 可编程逻辑阵列（PLA）

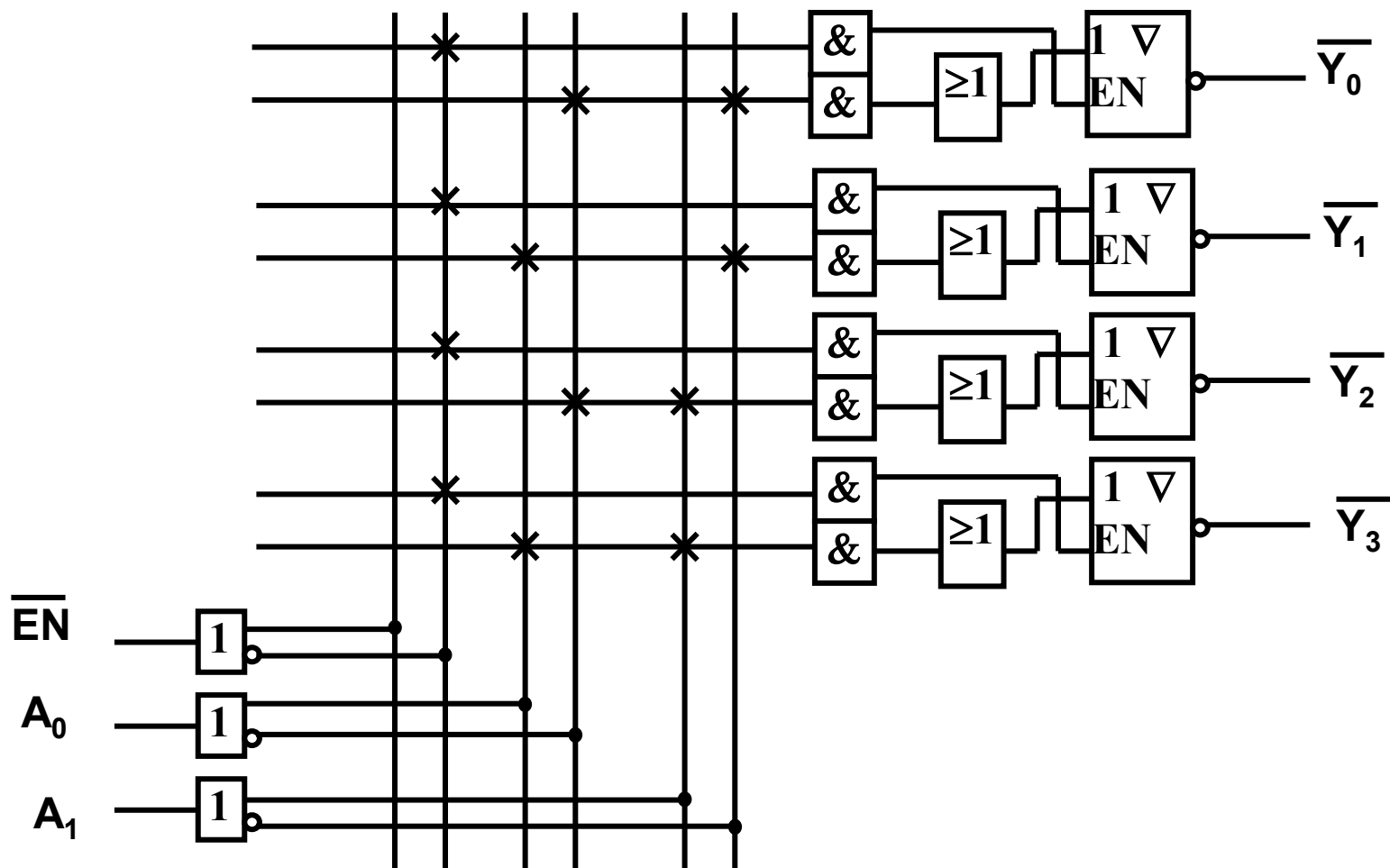
6.3.2 PAL的结构与应用

除了具有与阵列（可编程）和或阵列（固定）以外，还有输出和反馈电路：

- 专用输出结构
- 可编程输入/输出结构
- 寄存器输出结构
- 异或输出结构



例1：用**PAL**设计一个带使能端（低电平有效）的**2/4**线译码器，输出低电平有效。



例1实现电路图



PAL的优点:

① 提高了功能密度, 节省了空间。通常一片PAL可以代替4~12片SSI或2~4片MSI。同时, 虽然PAL只有20多种型号, 但可以代替90%的通用器件, 因而进行系统设计时, 可以大大减少器件的种类。

② 提高了设计的灵活性, 且编程和使用都比较方便。

③ 有上电复位功能和加密功能, 可以防止非法复制。

6.4 通用阵列逻辑（GAL）

- GAL的结构特点：输出端有一个组态可编程的输出逻辑宏单元OLMC，通过编程可以将GAL设置成不同的输出方式。这样，具有相同输入单元的GAL可以实现PAL器件所有的输出电路工作模式，故而称之为通用可编程逻辑器件。
- GAL与PAL的区别：① PAL是PROM熔丝工艺，为一次编程器件，而GAL是E² PROM工艺，可重复编程；② **PAL**的输出是固定的，而**GAL**用一个可编程的输出逻辑宏单元（**OLMC**）做为输出电路。**GAL**比**PAL**更灵活，功能更强，应用更方便，几乎能替代所有的**PAL**器件。

6.4 通用阵列逻辑（**GAL**）

GAL是低密度**PLD**，因此，它的结构和前述的低密度**PLD**类似。如**GAL16V8**，**GAL20V8**等。但是**GAL**的输出电路更为复杂，灵活性大，不需要用户来选择芯片。

一、GAL16V8总体结构

20个引脚的器件；

8个输入缓冲器（引脚2~9）；

8个输出缓冲反相器（引脚12~19）；

8个输出反馈/输入缓冲器（既可做输入也可做输出），因此为16V8；

1个时钟输入缓冲器；

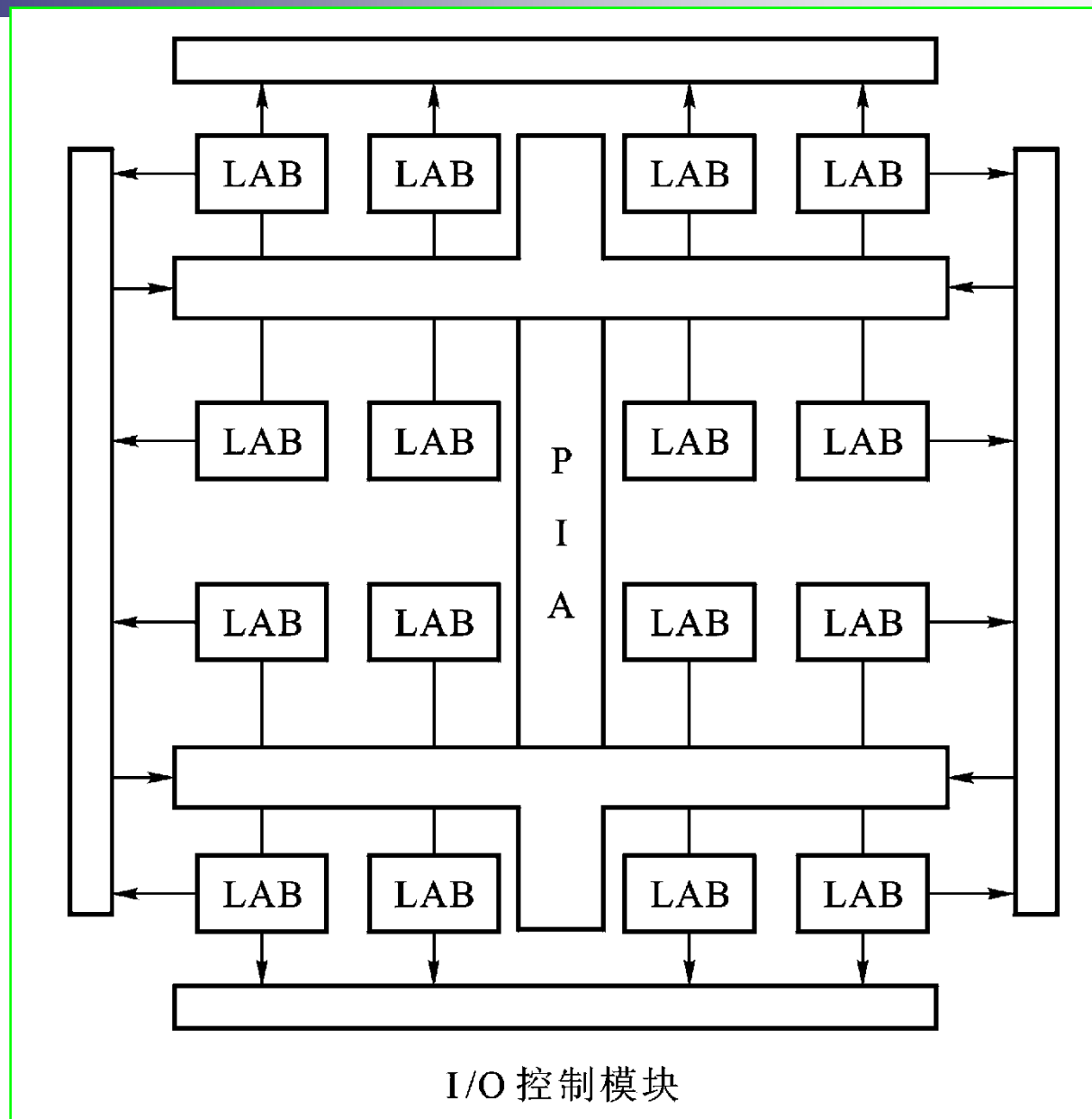
1个选通信号输入反相器；

6.5 复杂可编程逻辑器件（CPLD）

CPLD是阵列型高密度可编程控制器，其基本结构形式和PAL、GAL相似，都由可编程的与阵列、固定的或阵列和逻辑宏单元组成，但集成规模都比PAL和GAL大得多。

基本包含三种结构：

- 逻辑阵列块（LAB）
- 可编程I/O单元
- 可编程连线阵列（PIA）



CPLD的结构图

(1) 逻辑阵列块 (LAB)

一个LAB由十多个宏单元的阵列组成。

每个宏单元由三个功能块组成：

{ 逻辑阵列
乘积项选择矩阵
可编程寄存器

它们可以被单独的配置为时序逻辑或组合逻辑工作方式。

如果每个宏单元中的乘积项不够用时，还可以利用其结构中的共享和并联扩展乘积项。

(2) 可编程I/O单元

I/O端常作为一个独立单元处理。通过对I/O端口编程，可以使每个引脚单独的配置为输入输出和双向工作、寄存器输入等各种不同的工作方式。

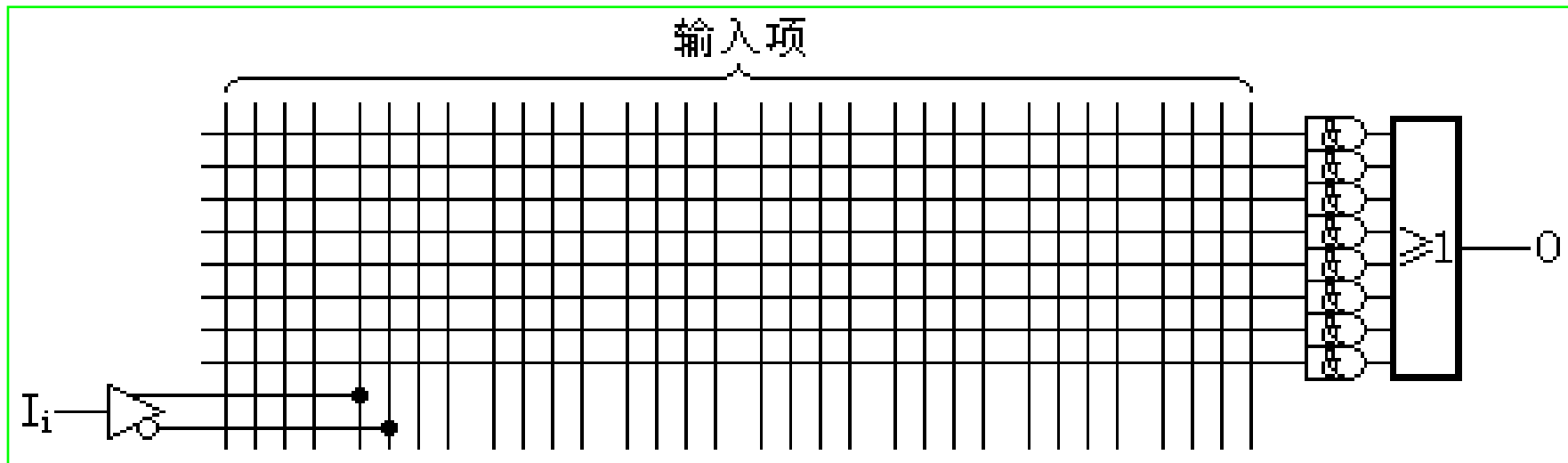
(3) 可编程连线阵列

在各LAB之间以及各LAB和I/O单元之间提供互连网络。这种互连机制有很大的灵活性，它允许在不影响引脚分配的情况下改变内部的设计。

作业

PAL的输出结构

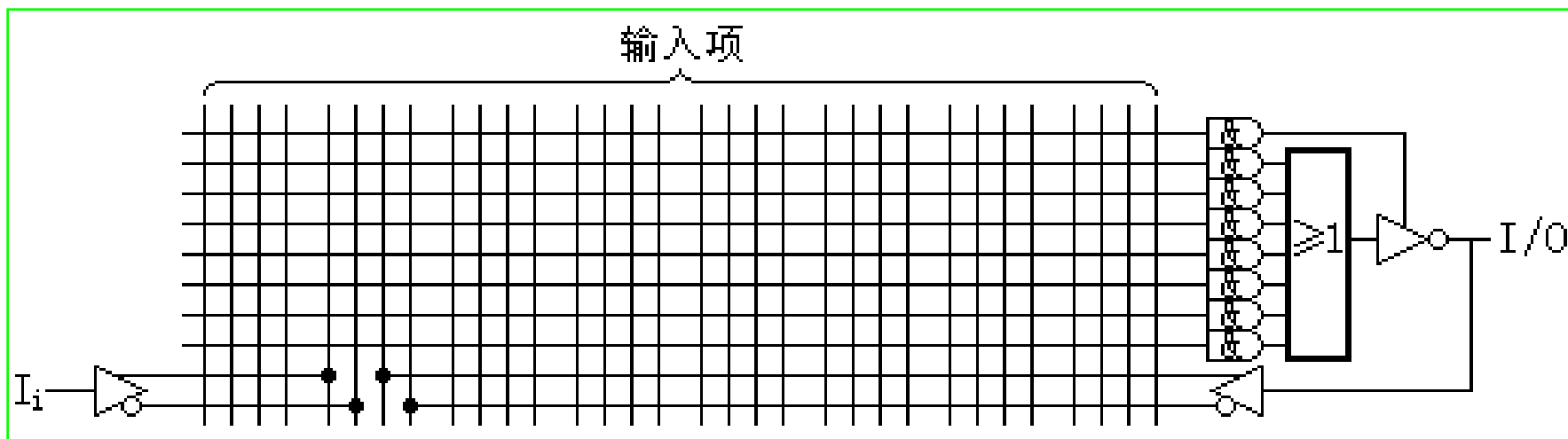
① 专用输出结构。输出端只能输出信号，不能兼作输入。只能实现组合逻辑函数。目前常用的产品有PAL10H8、PAL10L8等。



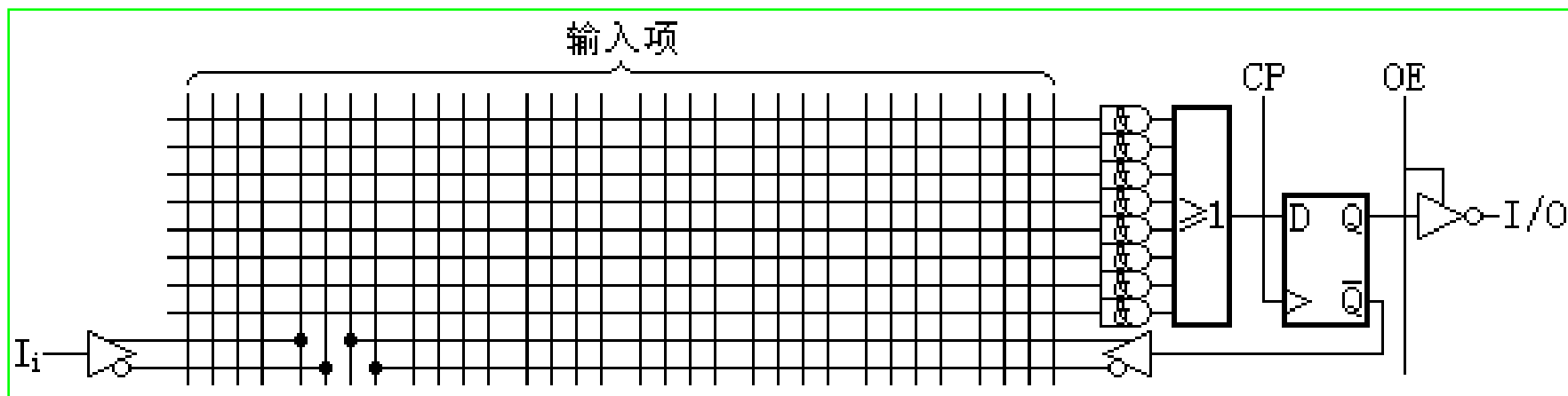
② 可编程I/O结构。输出端有一个三态缓冲器，三态门受一个乘积项的控制。

当三态门禁止，输出呈高阻状态时，I/O引脚作输入用；

当三态门被选通时，I/O引脚作输出用。



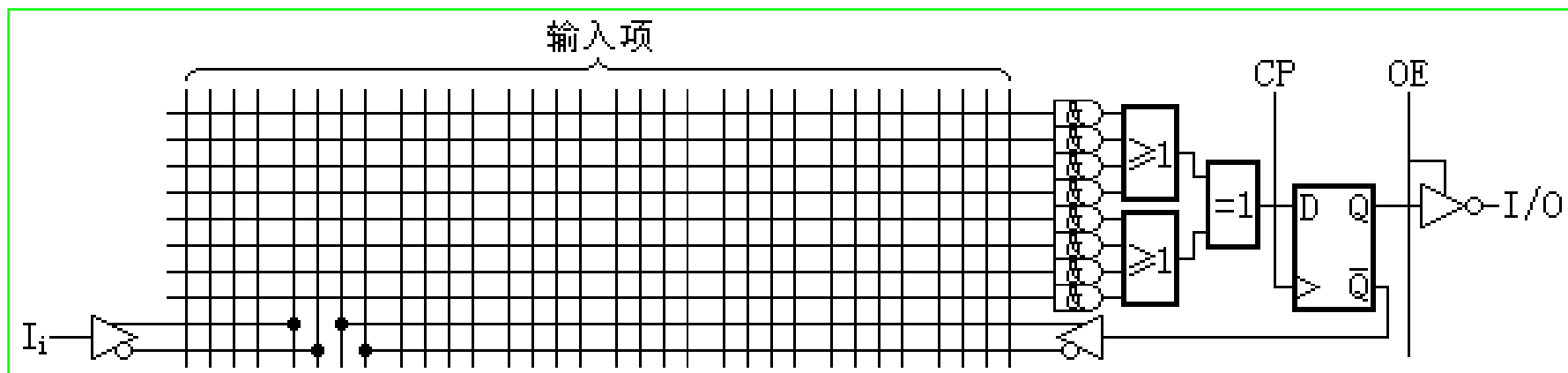
③ 寄存器输出结构。输出端有一个D触发器，在使能端的作用下，触发器的输出信号经三态门缓冲输出。能记忆原来的状态，从而实现时序逻辑功能。



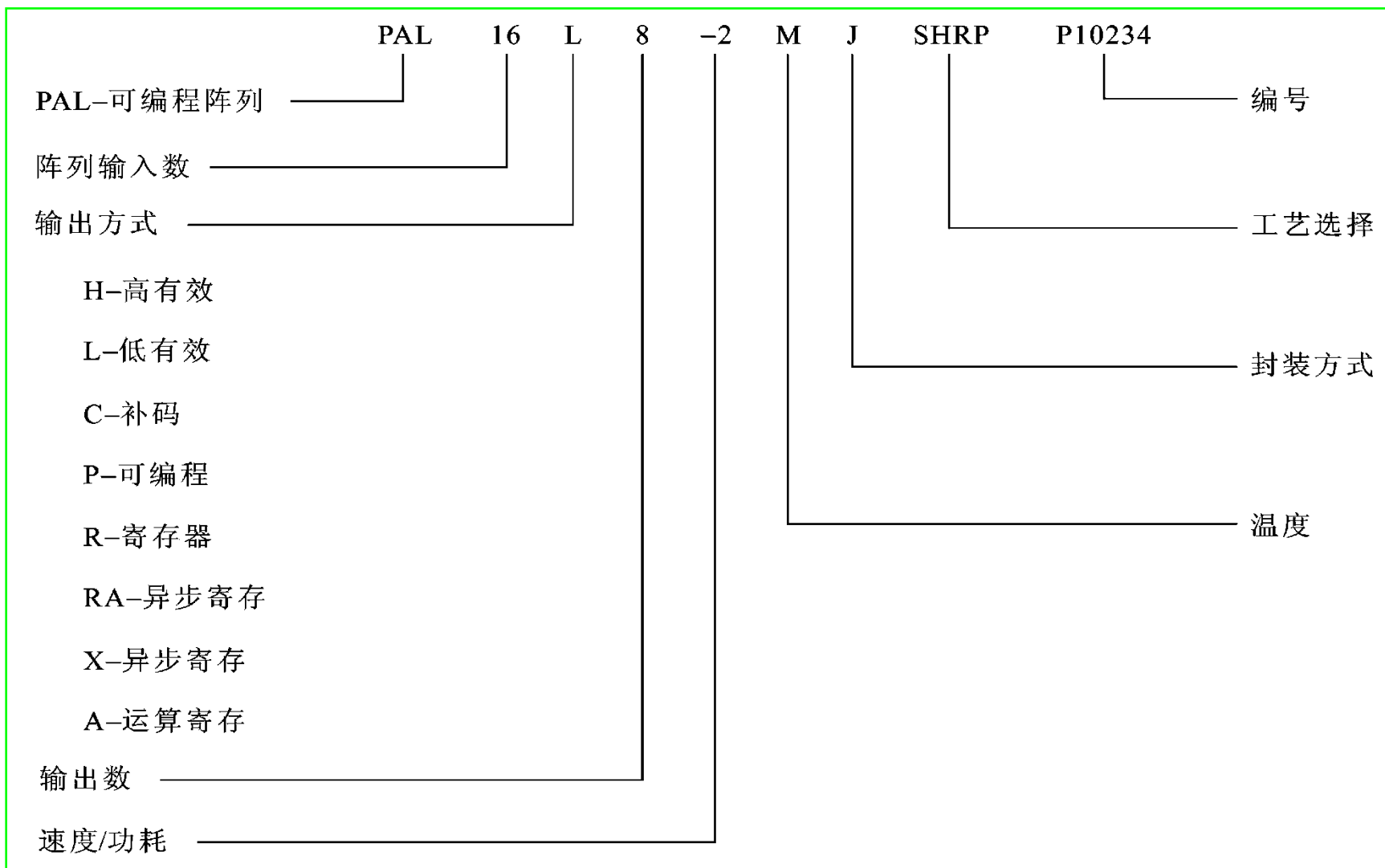
④ 异或—寄存器型输出结构。

输出部分有两个或门，它们的输出经异或门后再经D触发器和三态缓冲器输出，这种结构便于对与或逻辑阵列输出的函数求反，还可以实现对寄存器状态进行维持操作，适用于实现计数器及状态。

$$(A \oplus 0 = A, A \oplus 1 = \bar{A})$$



PAL共有21种，通过不同的命名可以区别。



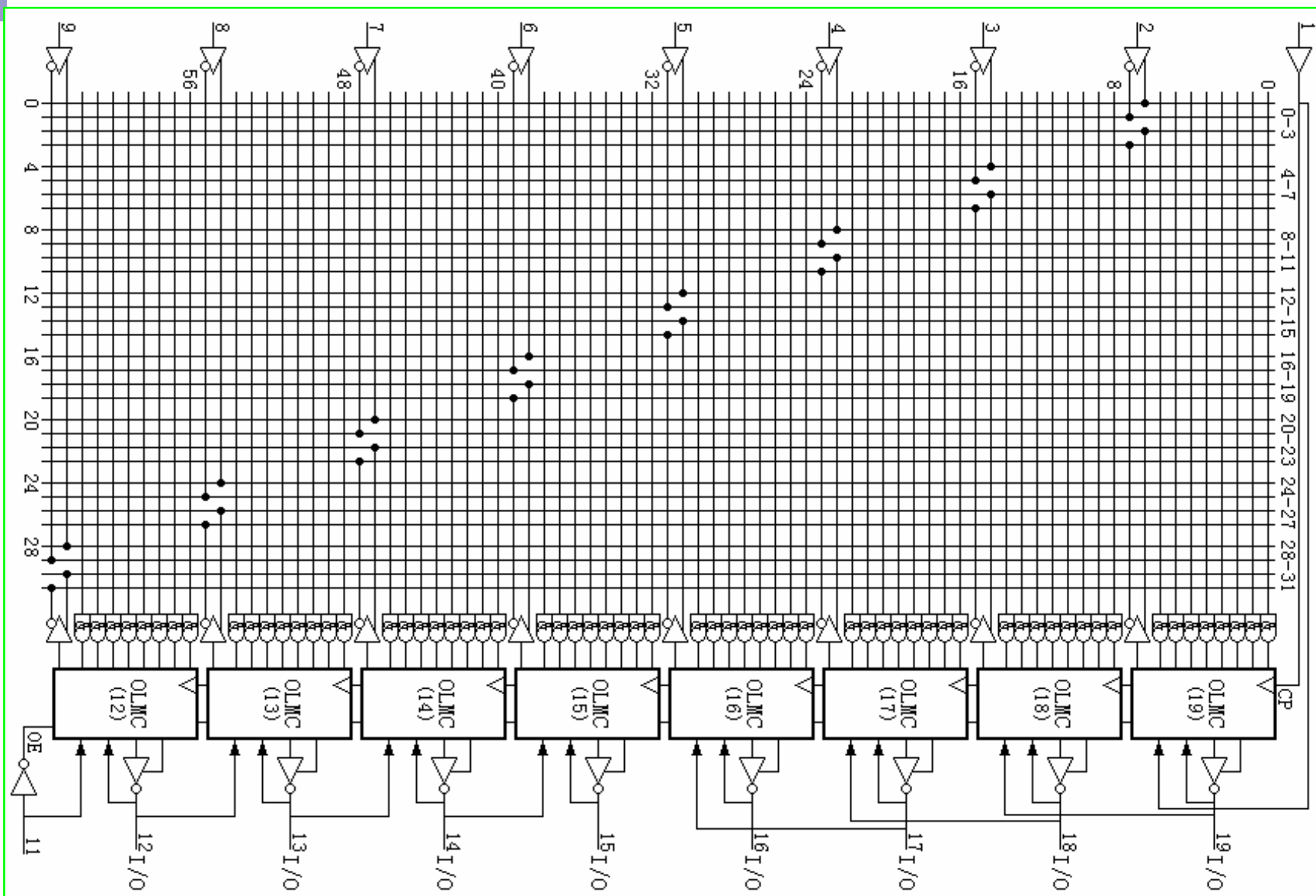


普通型GAL16V8的基本特点。

(1) GAL的基本结构。

① 8个输入缓冲器和8个输出反馈/输入缓冲器。

② 8个输出逻辑宏单元OLMC和8个三态缓冲器，
每个OLMC对应一个I/O引脚。

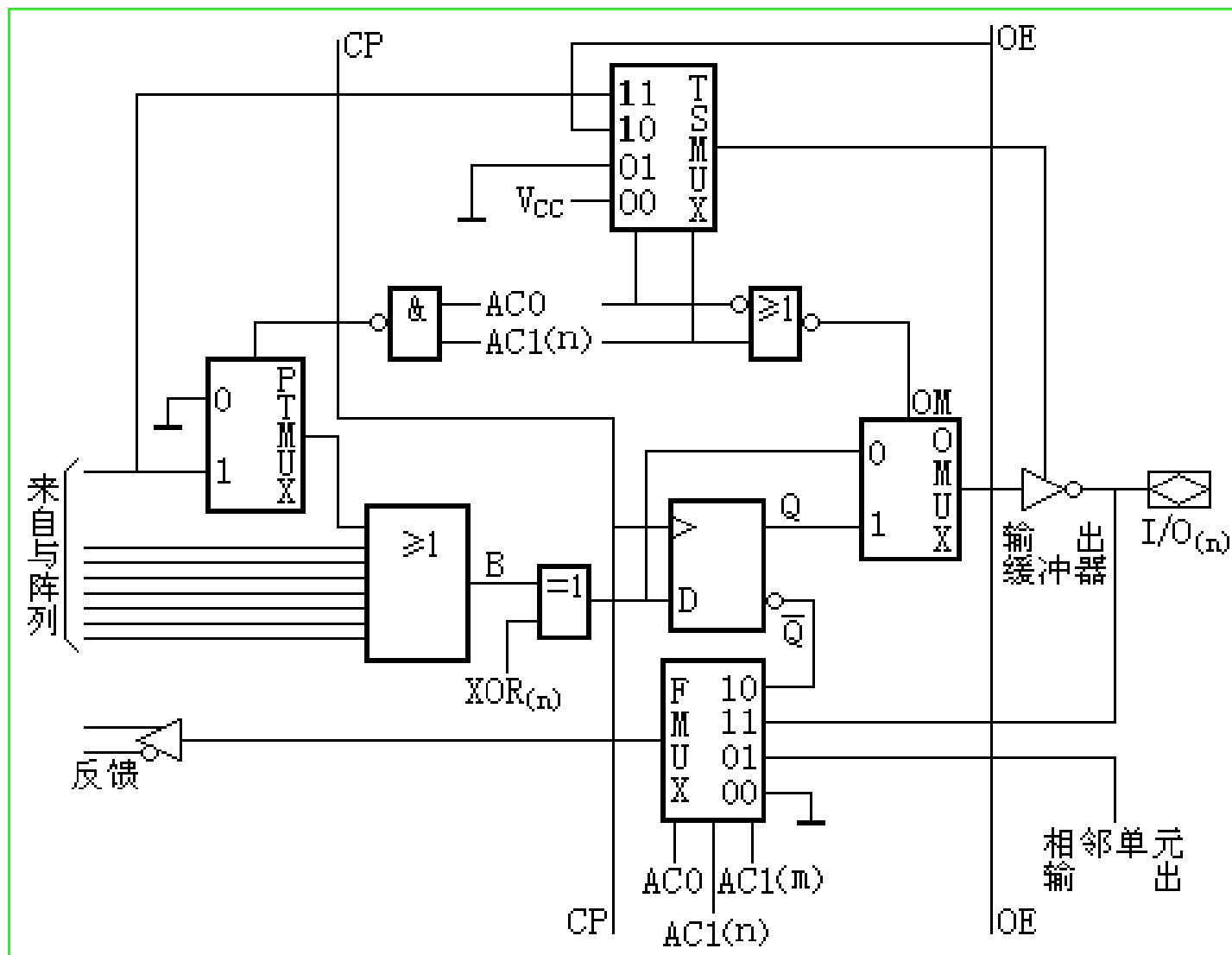


GAL16V8的逻辑图


③ 由 8×8 个与门构成的与阵列，共形成64个乘积项，每个与门有32个输入项，由8个输入的原变量、反变量（16）和8个反馈信号的原变量、反变量（16）组成，故可编程与阵列共有 $32 \times 8 \times 8 = 2048$ 个可编程单元。

④ 系统时钟CK 和三态输出选通信号OE的输入缓冲器。

GAL器件没有独立的或阵列结构，各个或门放在各自的输出逻辑宏单元（OLMC）中。



OLMC的逻辑图

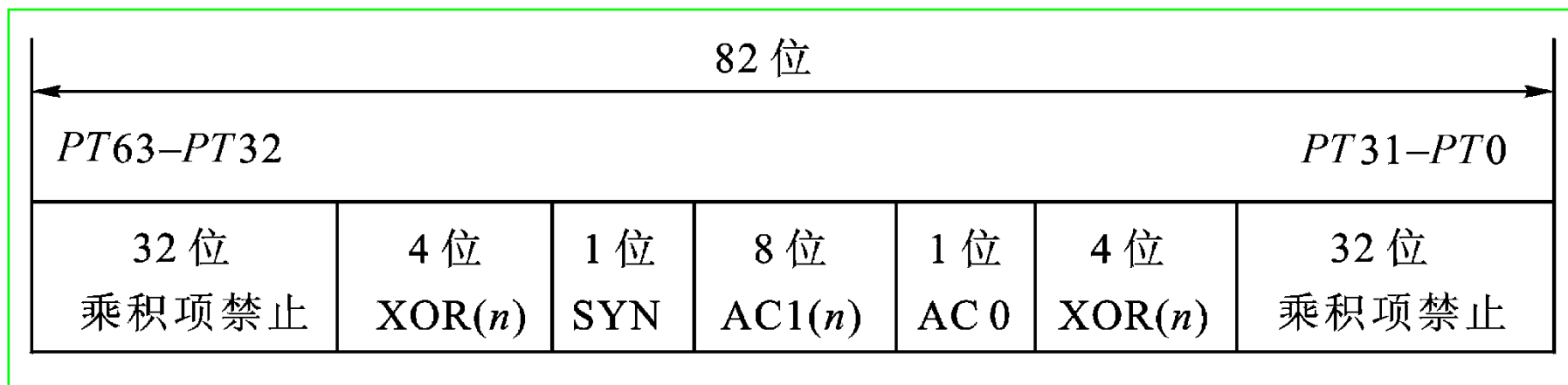


或门：有8个输入端，和来自与阵列的8个乘积项（PT）相对应。

异或门：用于选择输出信号的极性。

D触发器：使GAL适用于时序逻辑电路。

4个多路开关（MUX）：在结构控制字段作用下设定输出逻辑宏单元的状态。



GAL的结构控制字

① XOR (n) : 输出极性选择位。共有8位，分别控制8个OLMC的输出极性。异或门的输出D与它的输入信号B和XOR (n) 之间的关系为：

$$D = B \oplus \text{XOR}$$

当XOR=0时，即D = B；

当XOR=1时，即D = \neg B

择位。

当 $\text{SYN}=0$ 时，D触发器处于工作状态，OLMC可为时序逻辑电路；

当 $\text{SYN}=1$ 时，D触发器处于非工作状态，OLMC只能是组合逻辑电路。

注意：当 $\text{SYN}=0$ 时，可以通过其它控制字，使D触发器不被使用，这样便可以构成组合逻辑输出。但只要有一个OLMC需要构成时序逻辑电路时，就必须使 $\text{SYN}=0$ 。

③ AC0 、 AC1 (n) ：与 SYN 相配合，用来控制输出逻辑宏单元的输出组态。

SYN	AC0	AC1	XOR	功 能	输出极性
1	0	1	/	组合逻辑专用输入三态门禁止	/
1	0	0	0 1	组合逻辑专用输出	低有效 高有效
1	1	1	0 1	组合逻辑带反馈双向I/O输出	低有效 高有效
0	1	1	0 1	时序逻辑组合I/O输出	低有效 高有效
0	1	0	0 1	时序逻辑寄存器输出	低有效 高有效

只要写入不同的结构控制字，就可以得到不同类型的输出电路结构。

FPGA的基本结构:

- 可编程逻辑模块CLB
- 输入 / 输出模块IOB
- 互连资源IR

(1) 可编程逻辑模块CLB

结构形式:

- ① 查找表结构
- ② 多路开关结构
- ③ 多级与非门结构。

电路组成:

- 逻辑函数发生器
- 触发器
- 数据选择器
- 信号变换

(2) 可编程输入 / 输出模块 (IOB)

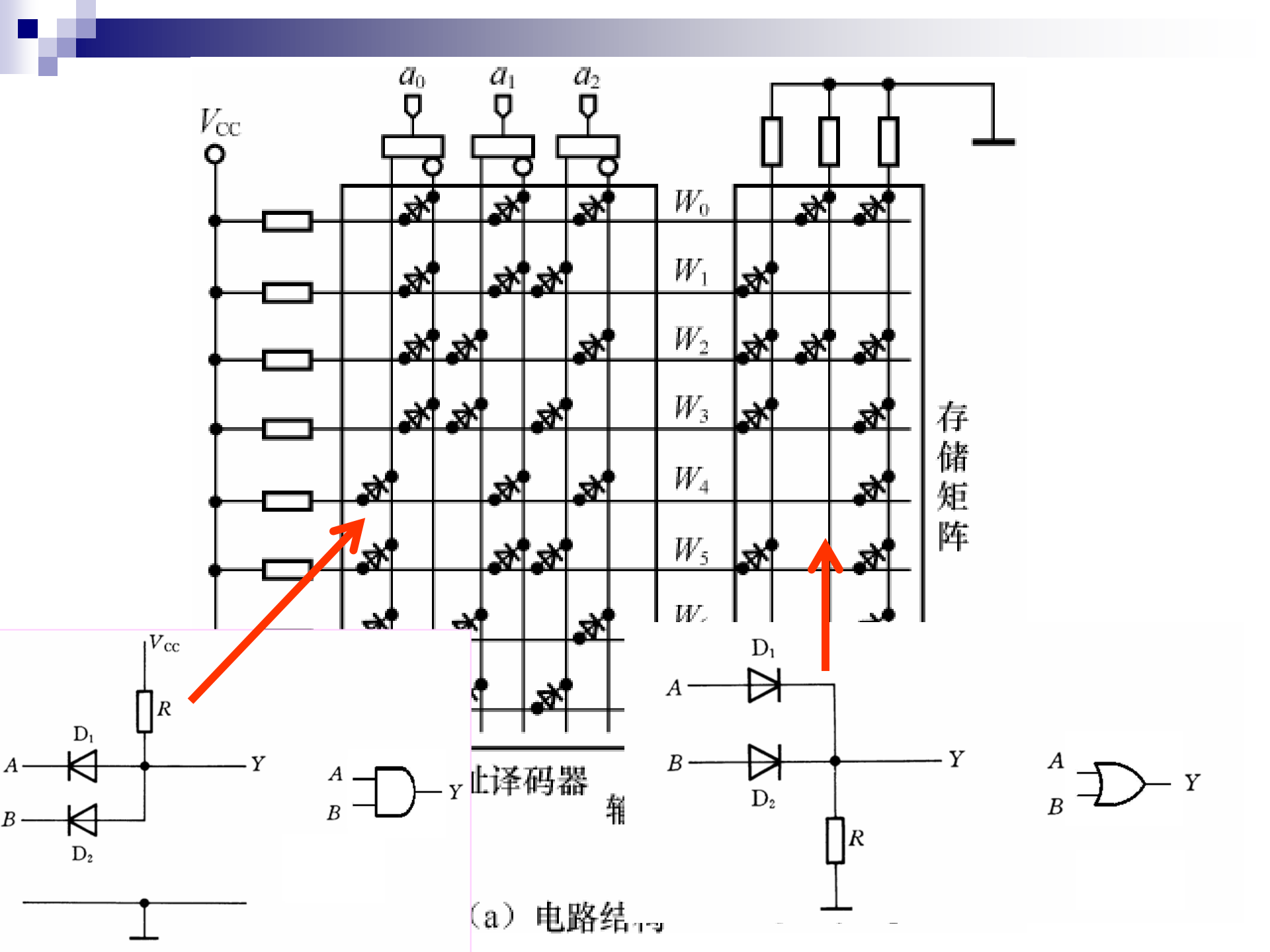
IOB主要完成芯片内部逻辑与外部封装脚的接口，它通常排列在芯片的四周；提供了器件引脚和内部逻辑阵列的接口电路。每一个IOB控制一个引脚(除电源线和地线引脚外)，将它们可定义为输入、输出或者双向传输信号端。

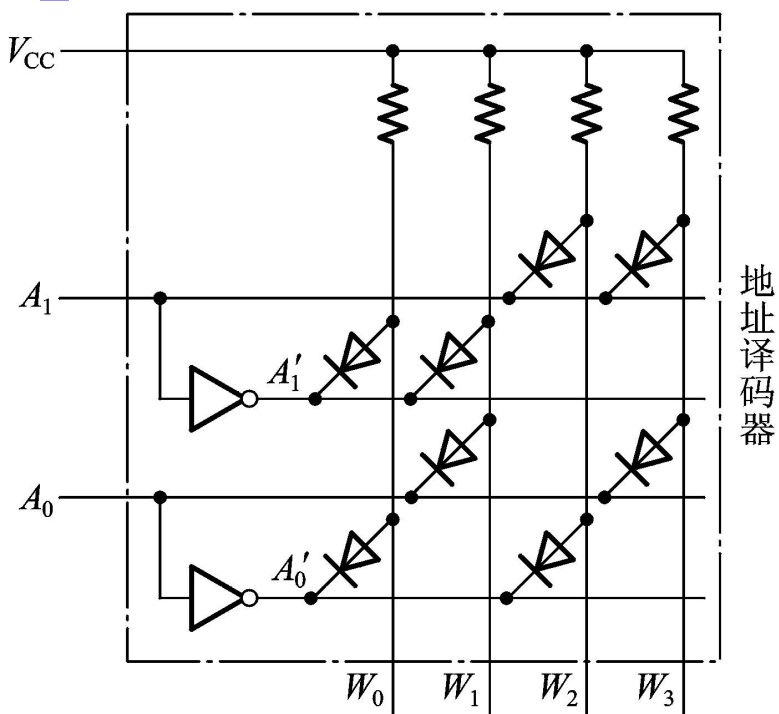
(3) 可编程互连资源(IR)

包括各种长度的连线线段和一些可编程连接开关。

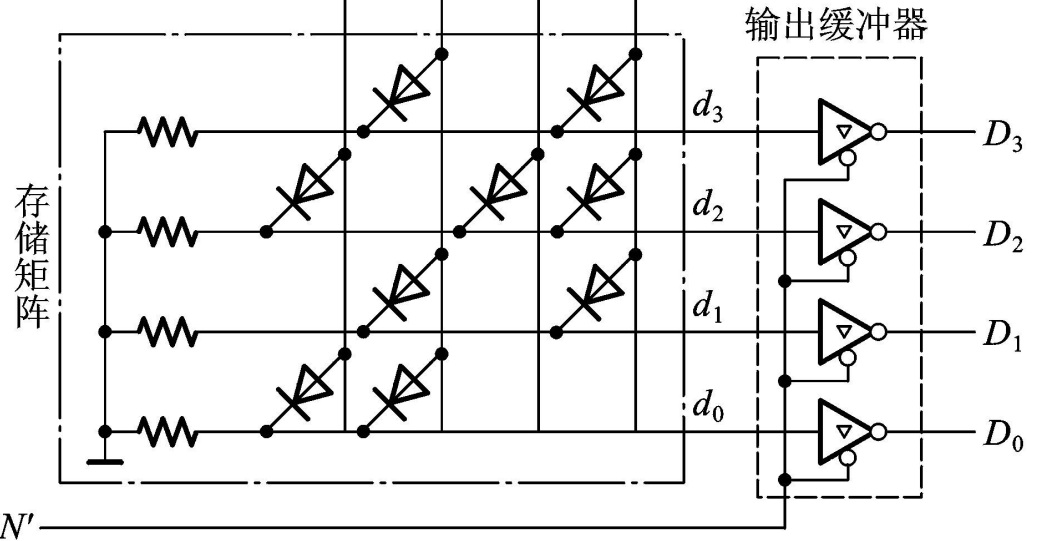
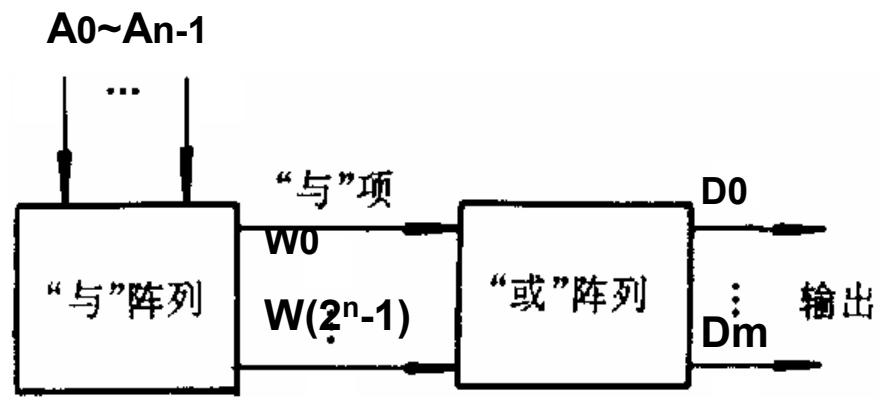
连线通路的数量与器件内部阵列的规模有关，阵列规模越大，连线数量越多。

互连线按相对长度分为单线、双线和长线三种。





地址译码器

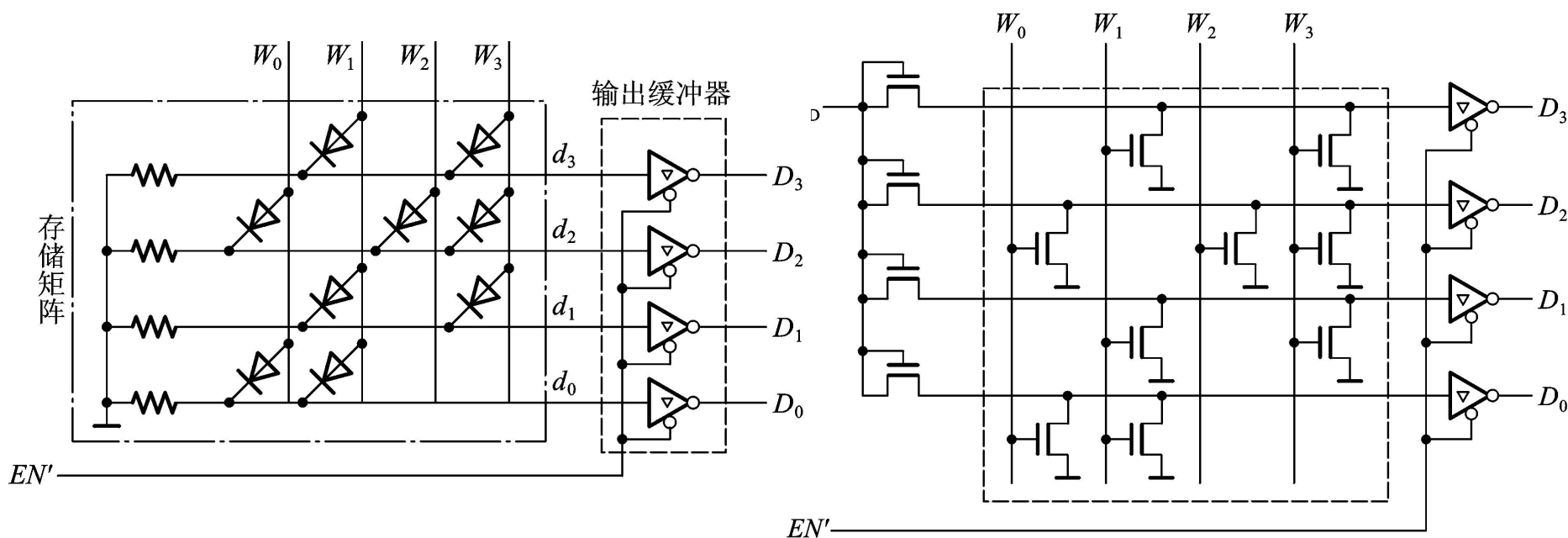


输出缓冲器

地 址		数 据			
A_1	A_0	D_3	D_2	D_1	D_0
0	0	0	1	0	1
0	1	1	0	1	1
1	0	0	1	0	0
1	1	1	1	1	0

两个概念：

- 存储矩阵的每个交叉点是一个“存储单元”，
存储单元中有器件存入“1”，无器件存入“0”



- 存储器的容量：“字数 x 位数”