

《数字电子技术基础》（第五版）教学课件

清华大学
阎石 王红

联系地址：清华大学 自动化系
邮政编码：100084

电子信箱：wang_hong@tsinghua.edu.cn
联系电话：(010)62792973

第八章 可编程逻辑器件

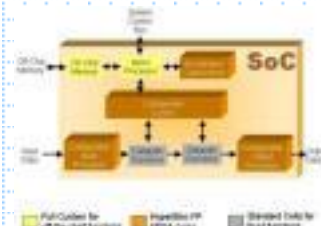
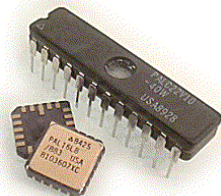
第八章 可编程逻辑器件 (PLD, Programmable Logic Device)

8.1 概述

一、PLD的基本特点

1. 数字集成电路从功能上分为通用型、专用型两大类

数字
系统



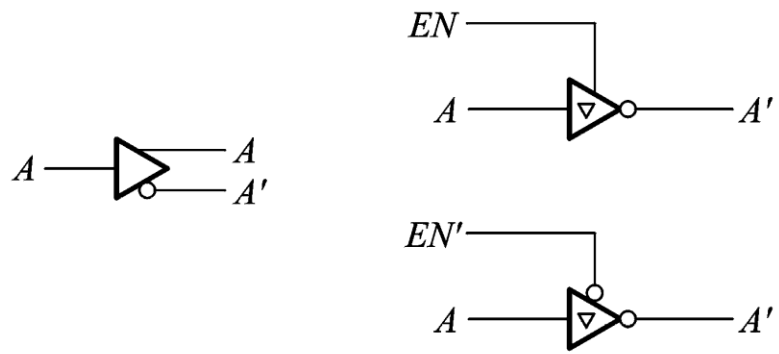
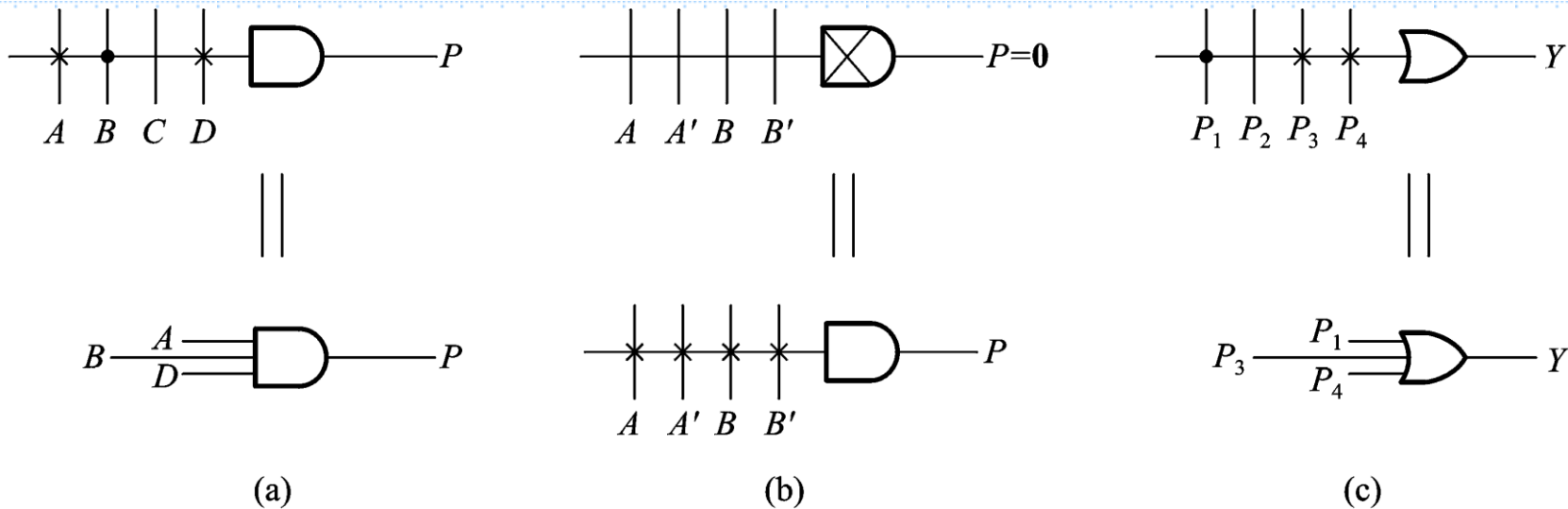
2. PLD的特点：是一种按通用器件来生产，但逻辑功能是由用户通过对器件编程来设定的

二、PLD的发展和分类

PROM是最早的PLD

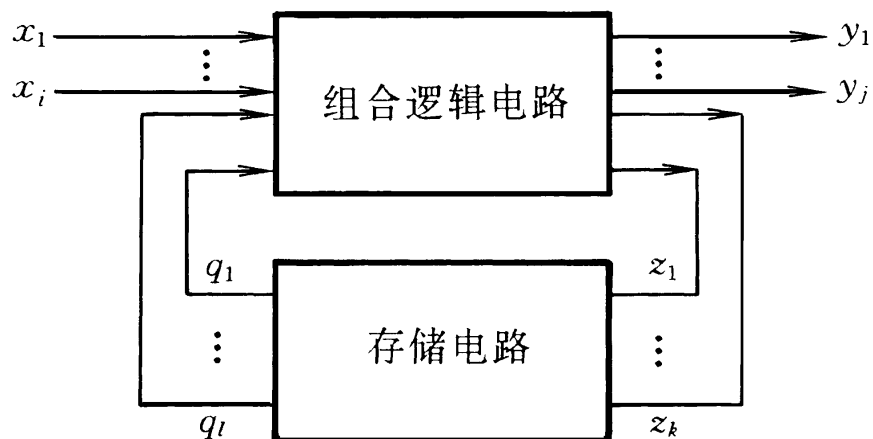
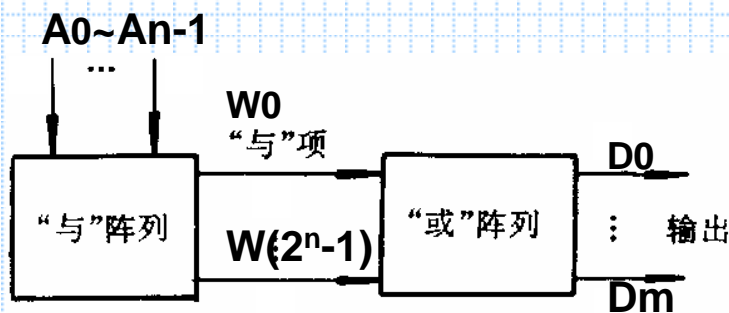
1. PAL 可编程逻辑阵列
2. FPLA 现场可编程阵列逻辑
3. GAL 通用阵列逻辑
4. EPLD 可擦除的可编程逻辑器件
5. FPGA 现场可编程门阵列
6. ISP-PLD 在系统可编程的PLD

三、LSI中用的逻辑图符号



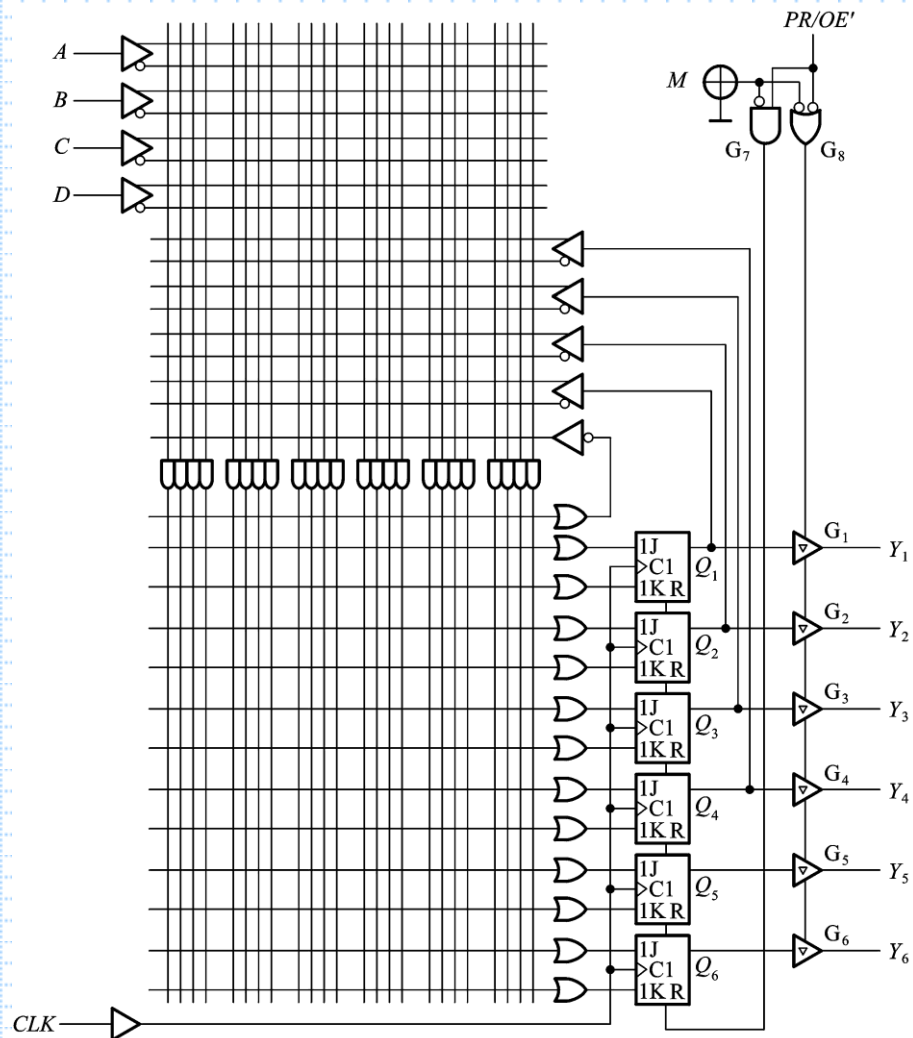
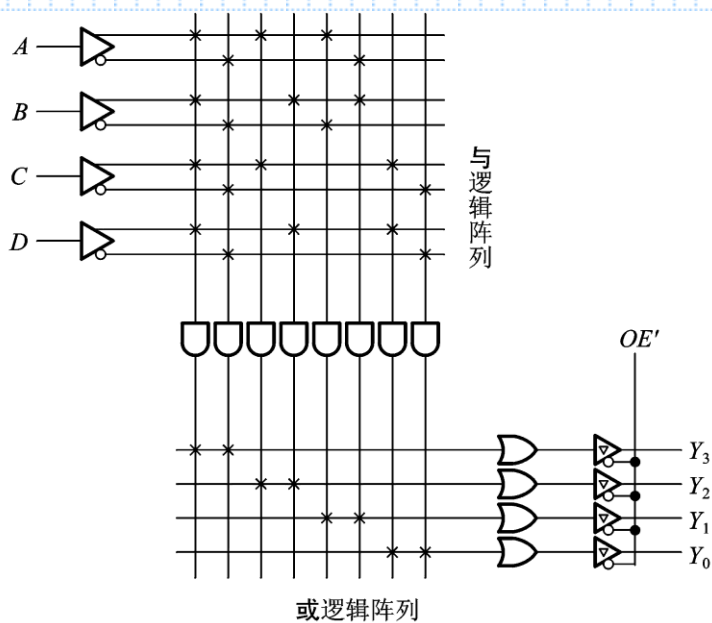
8.2 现场可编程逻辑阵列 FPLA

组合电路和时序电路结构的通用形式



8.2 FPLA

组合电路和时序电路结构的通用形式



可编程的“与”阵列
+ 可编程的“或”阵列

8.3 PAL (Programmable Array Logic)

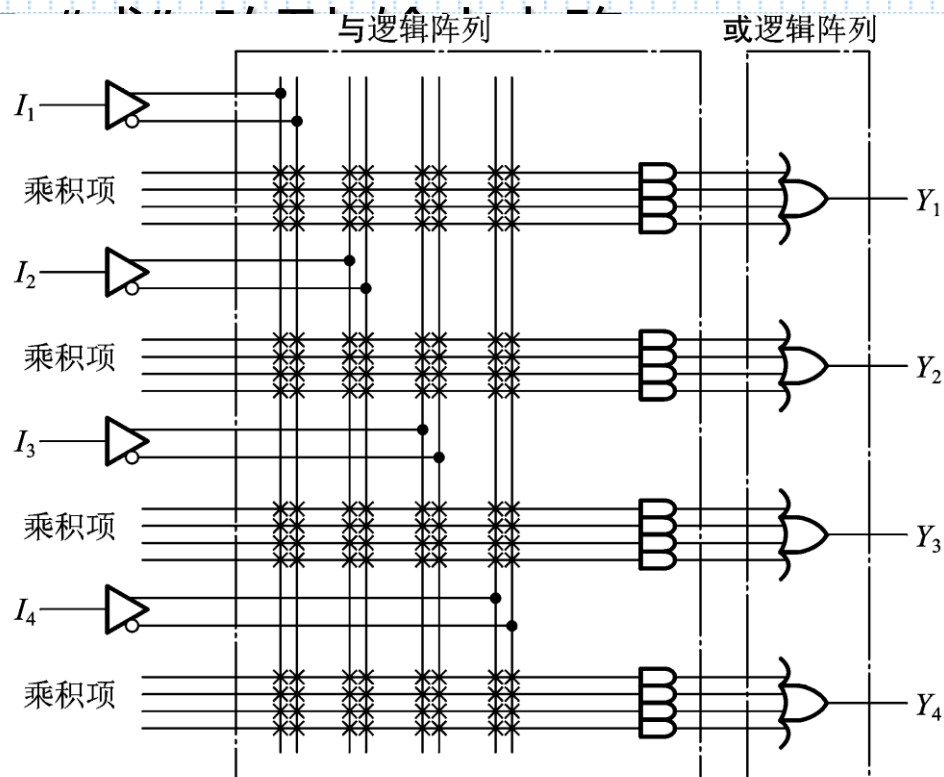
8.3.1 PAL的基本电路结构

一、基本结构形式

可编程“与”阵列+固定
最简单的形式为：

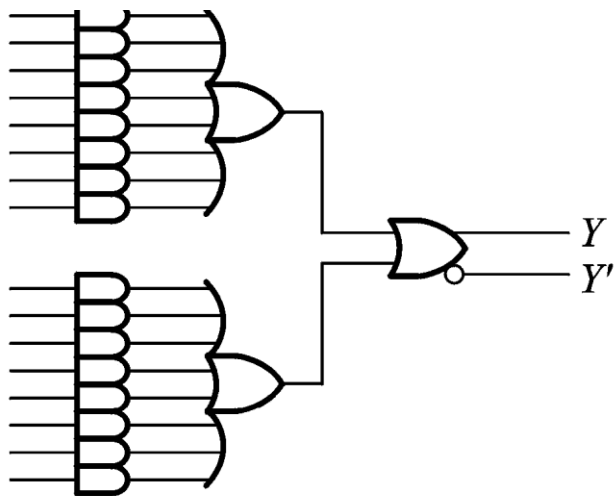
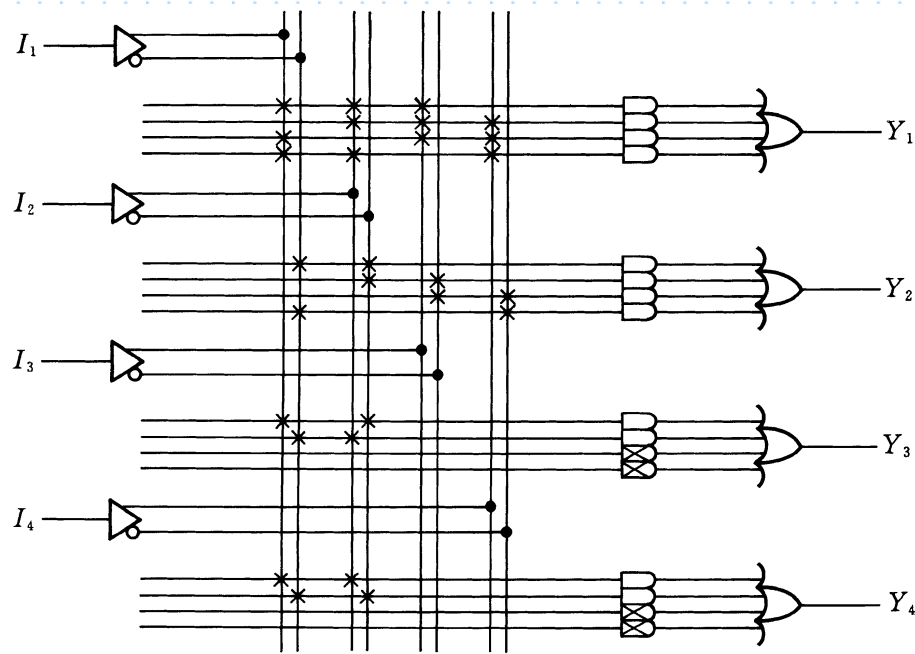
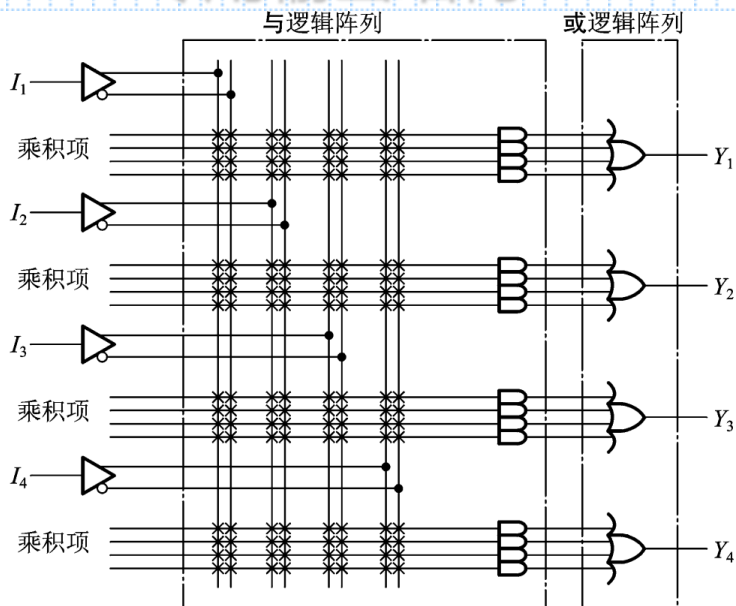
二、编程单元

出厂时，
所有的交叉点均有熔丝



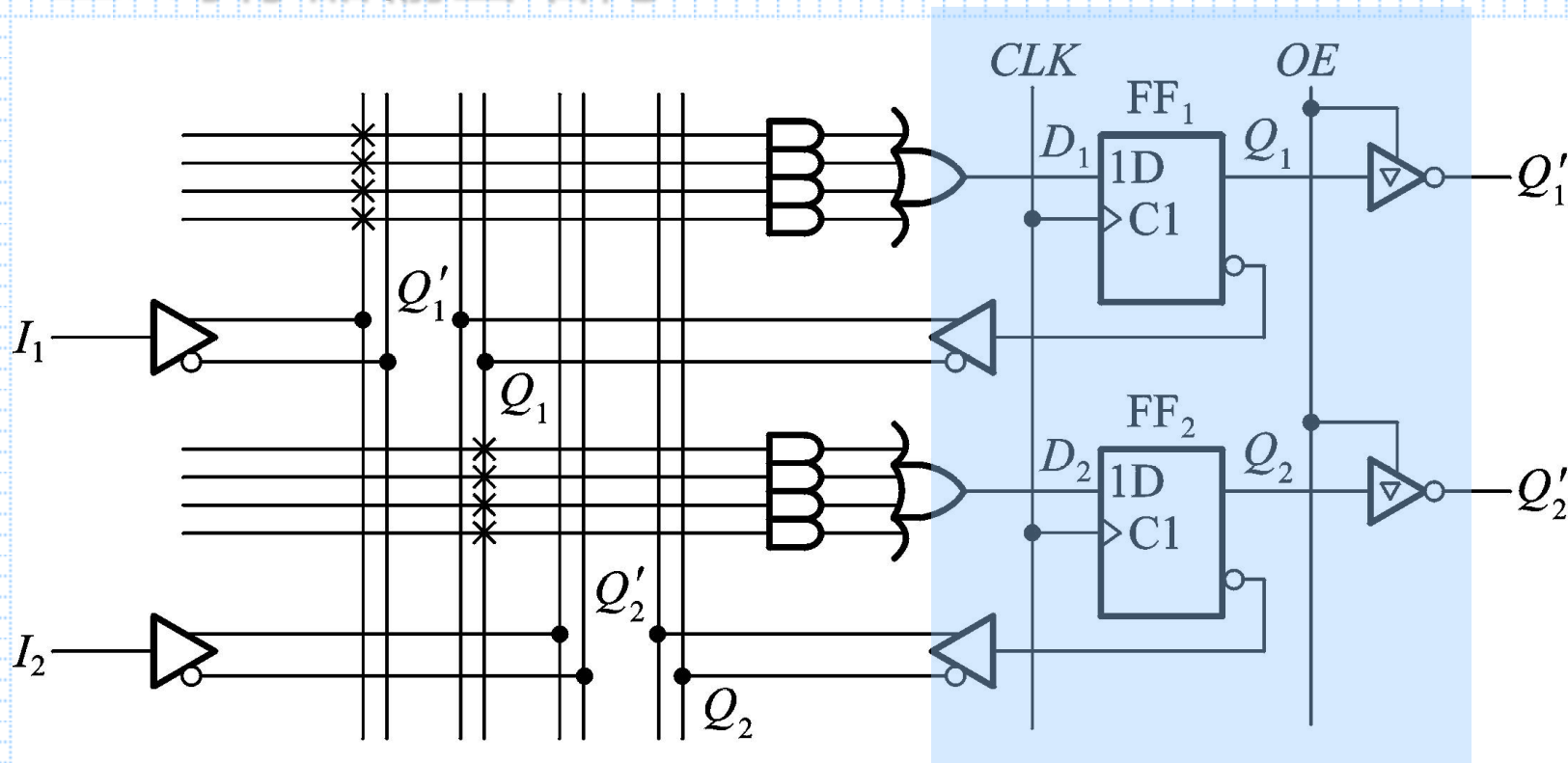
8.3.2 PAL的输出电路结构和反馈形式

一. 专用输出结构



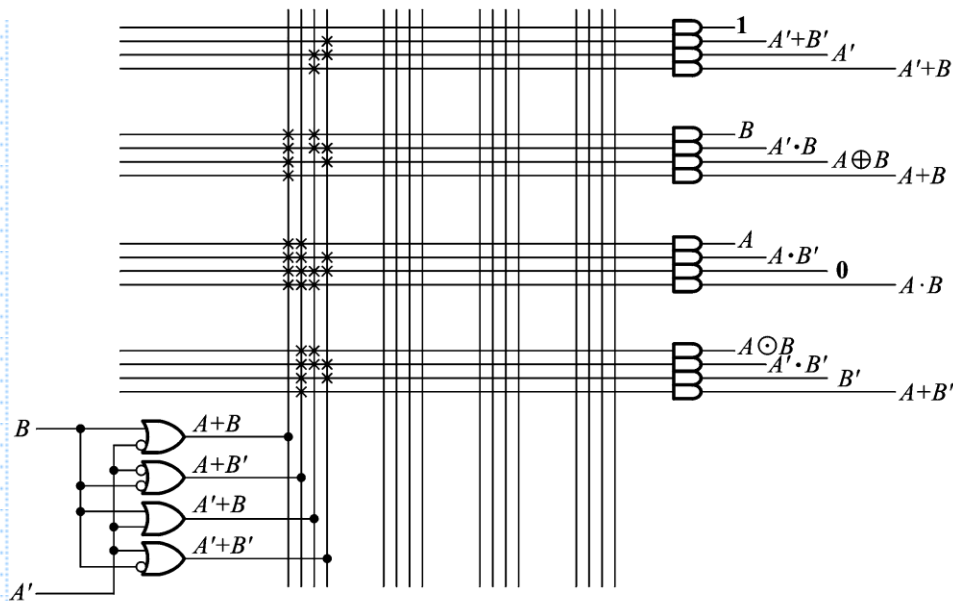
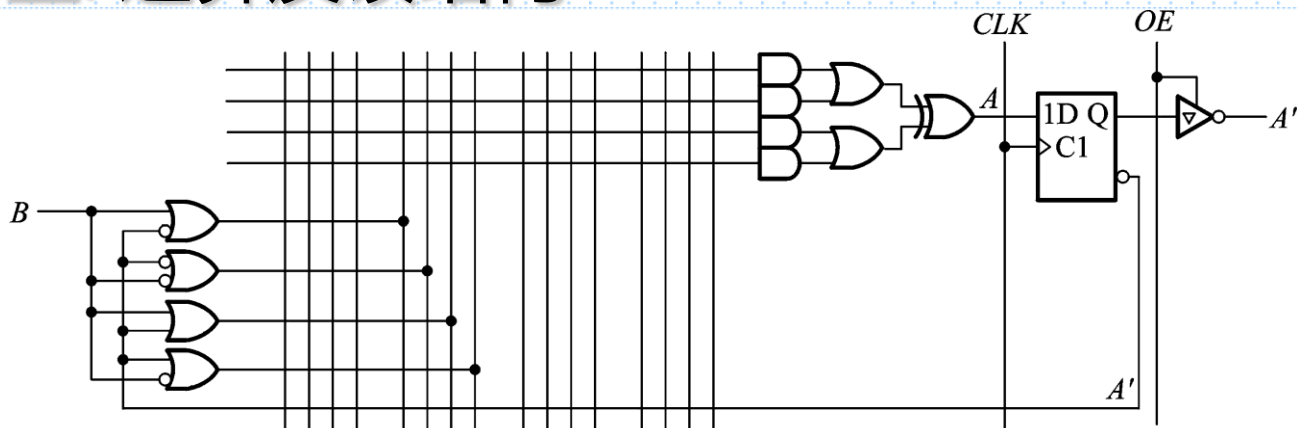
用途：产生组合逻辑电路

三. 寄存器输出结构



用途：产生时序逻辑电路

五. 运算反馈结构



时序逻辑电路

可产生A、B的十六种算术、逻辑运算

8.3.3 PAL的应用举例

8.4 通用逻辑阵列 GAL

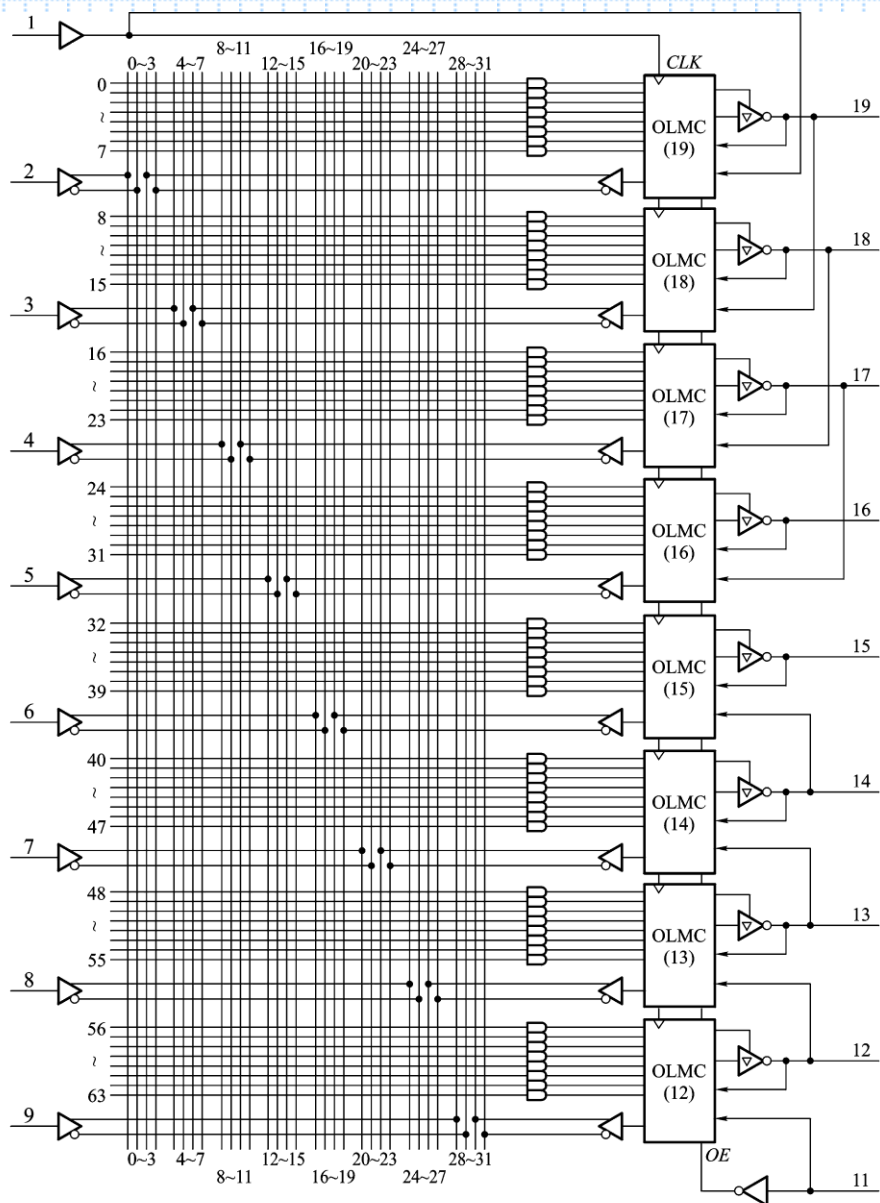
8.4.1 电路结构形式

可编程“与”阵列 + 固定“或”阵列 + 可编程输出
电路

编程单元

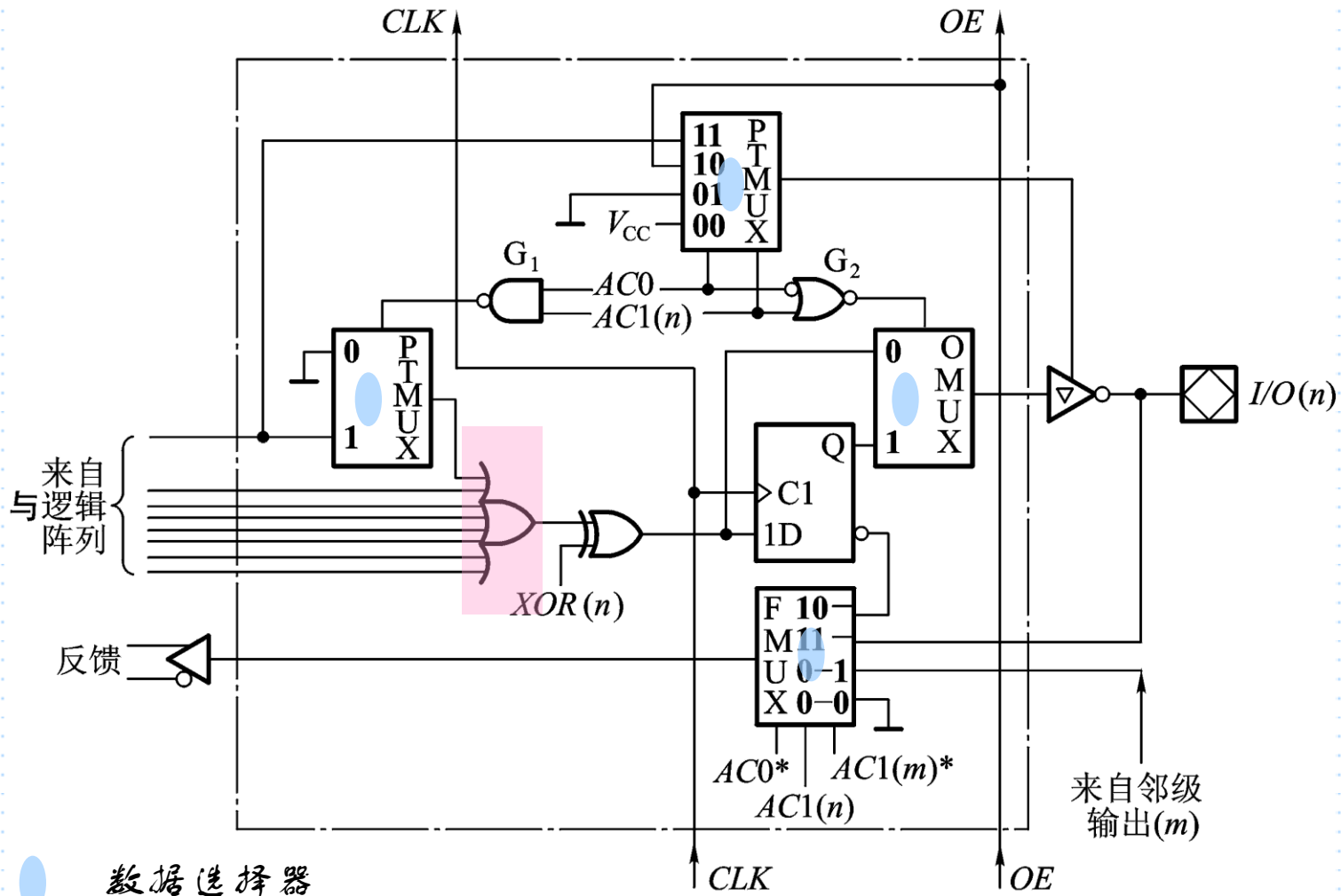
OLMC

采用E²CMOS 可改写



GAL16V8

8.4.2 OLMC



● 数据选择器

8.4.3 GAL的输入和输出特性

GAL是一种较为理想的高输入阻抗器件

GAL输出缓冲级

8.5 可擦除的可编程逻辑阵列EPLD

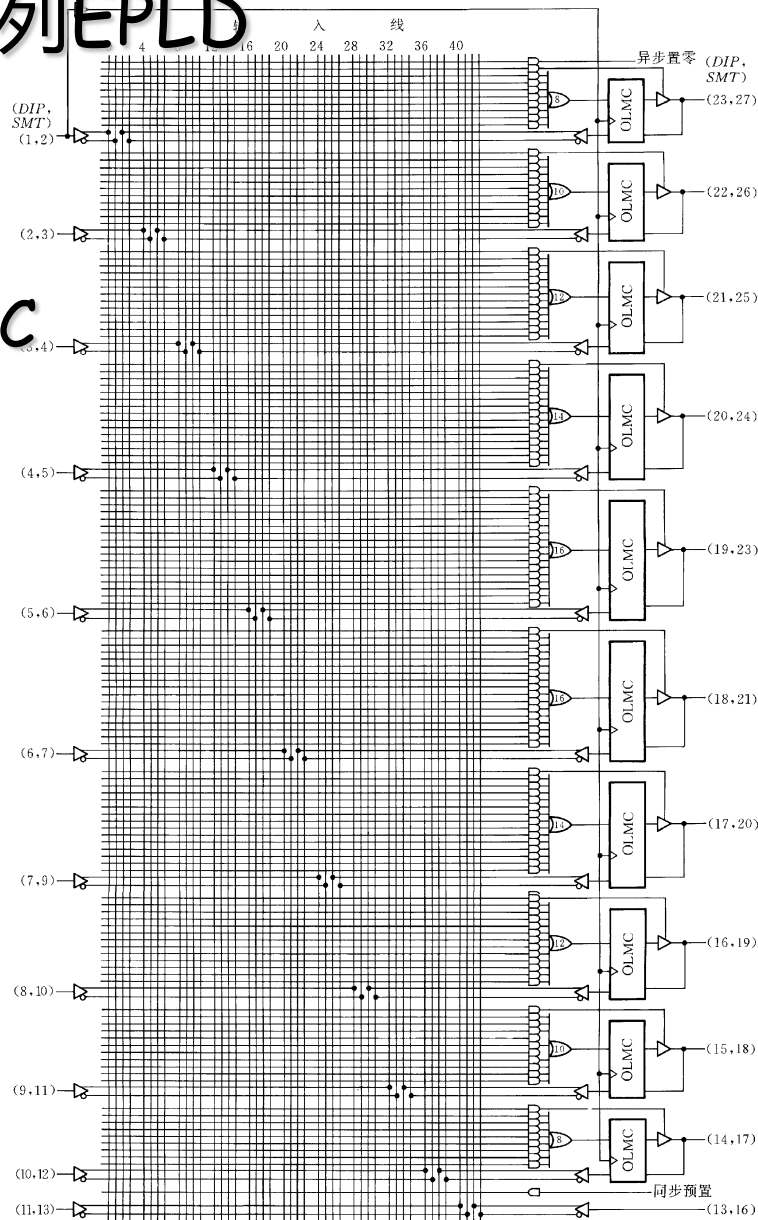
一、结构特点

相当于

“与-或”阵列 (PAL) + OLMC

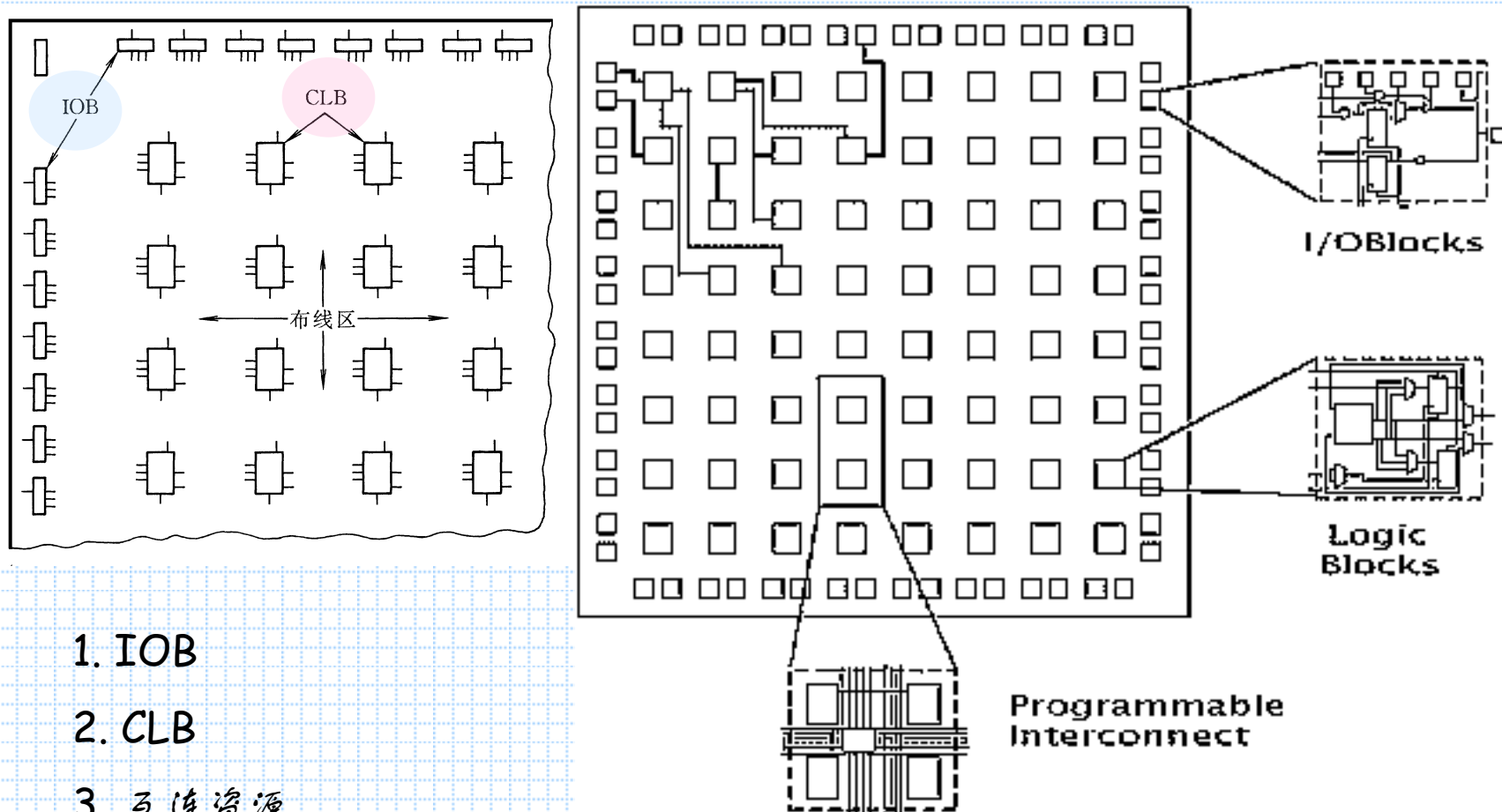
二、采用EPROM工艺

集成度提高



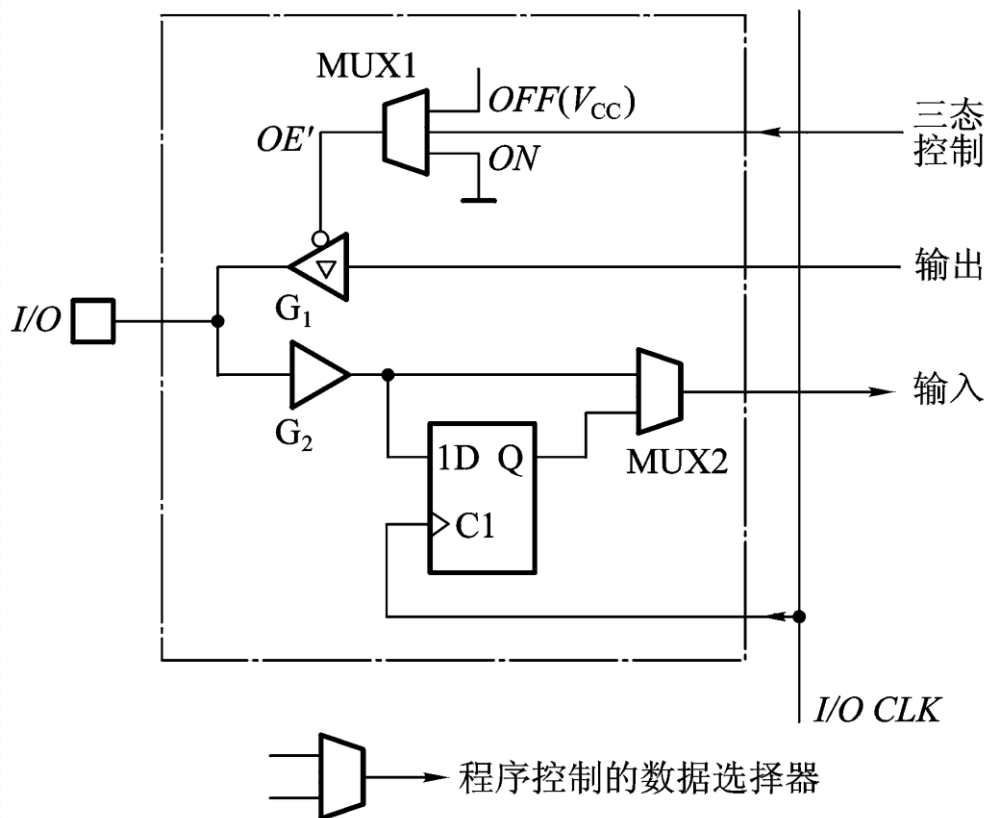
8.7 现场可编程门阵列FPGA

一、基本结构



1. IOB
2. CLB
3. 互连资源
4. SRAM

1. IOB

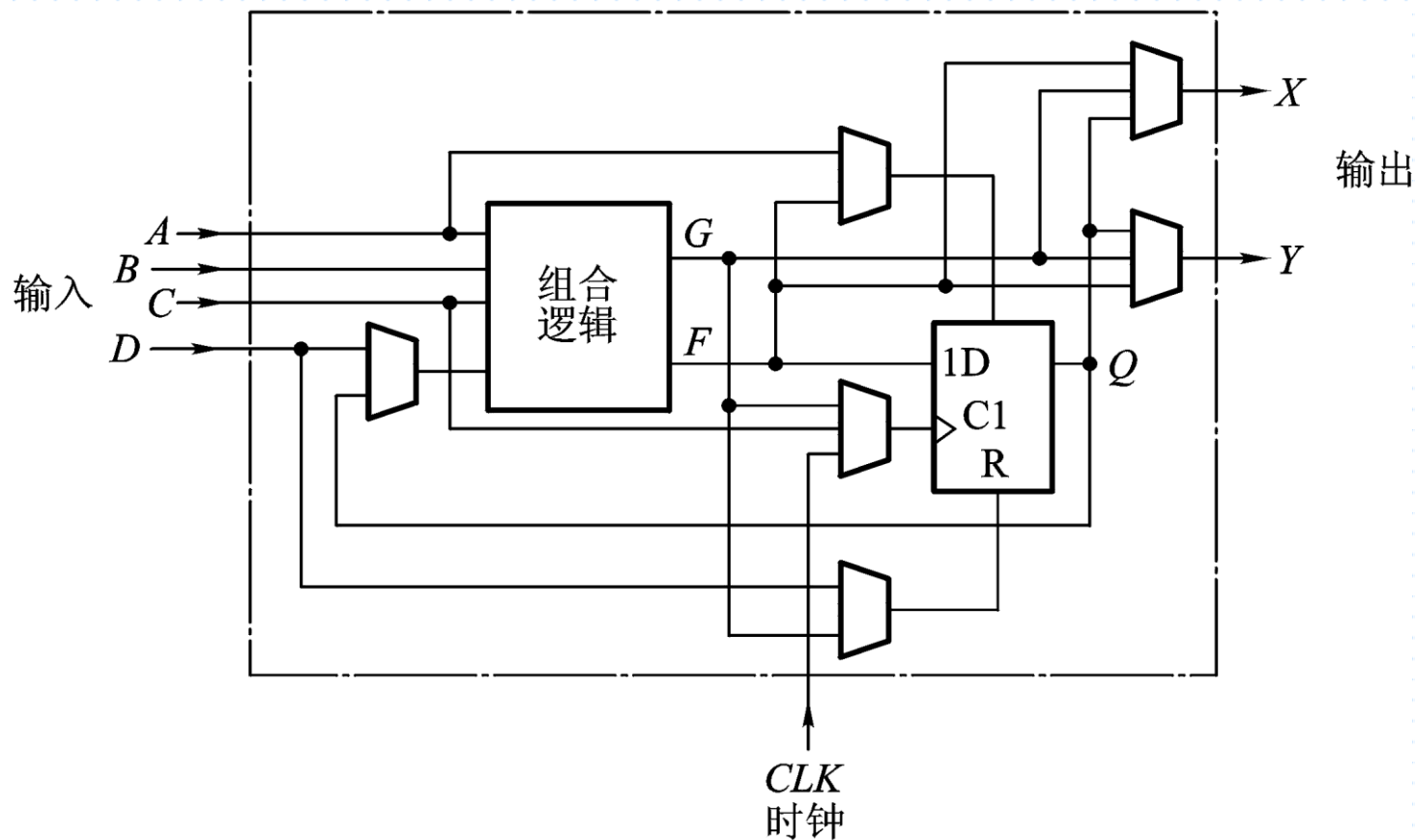


可以设置为输入/输出；

输入时可设置为：同步（经触发器）

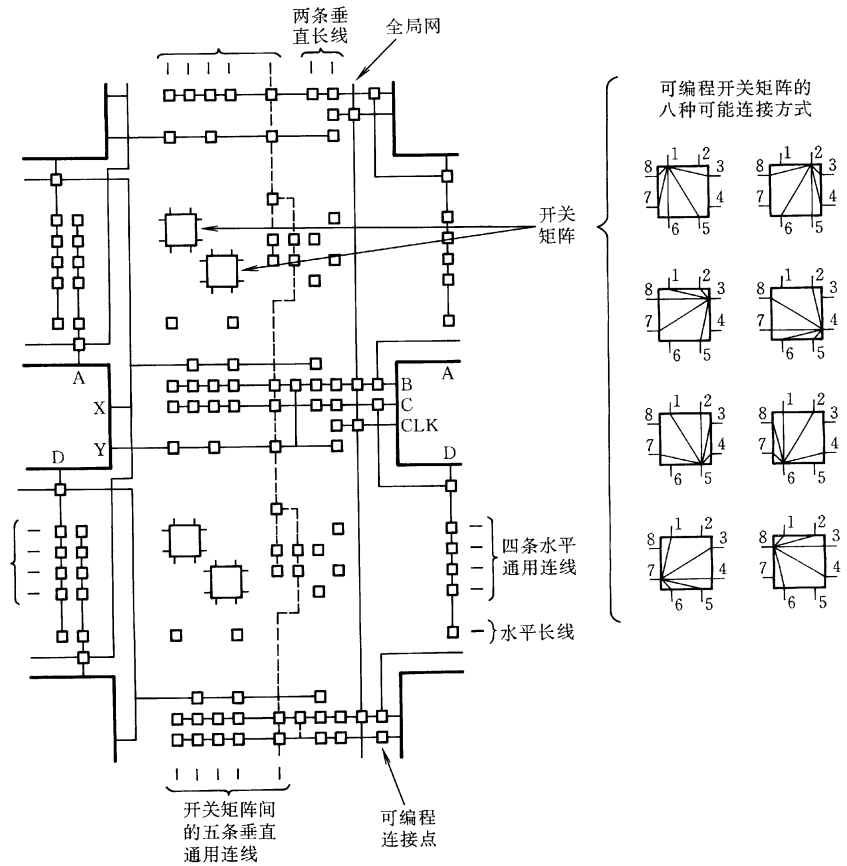
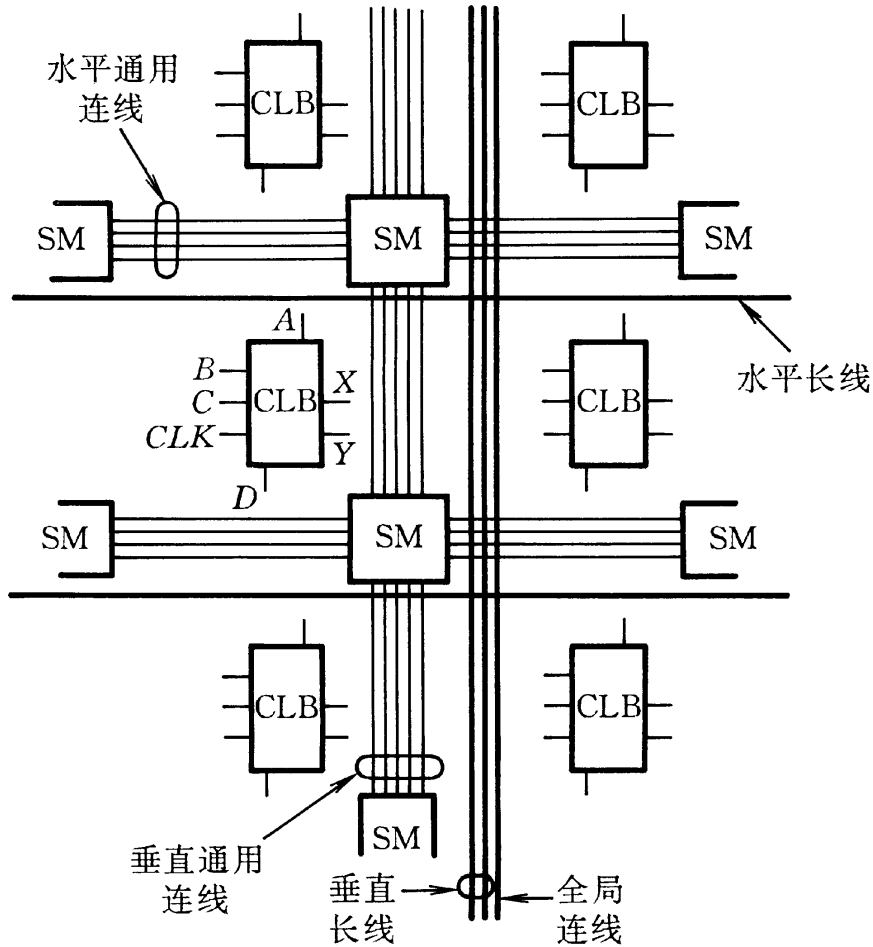
异步（不经触发器）

2. CLB



本身包含了组合电路和触发器，可构成小的时序电路
将许多CLB组合起来，可形成大系统

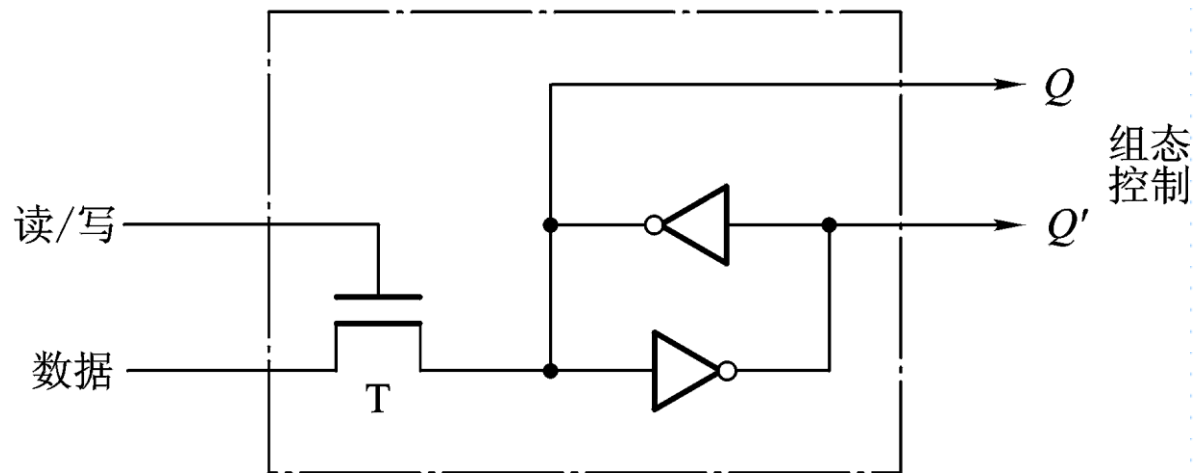
3. 互连资源



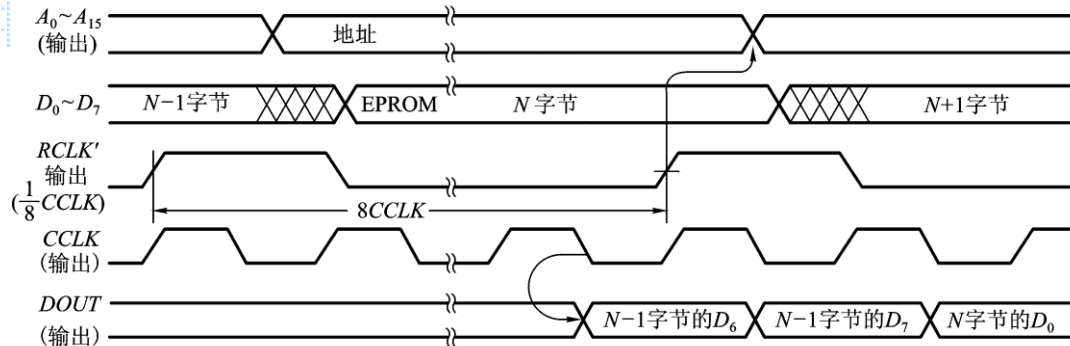
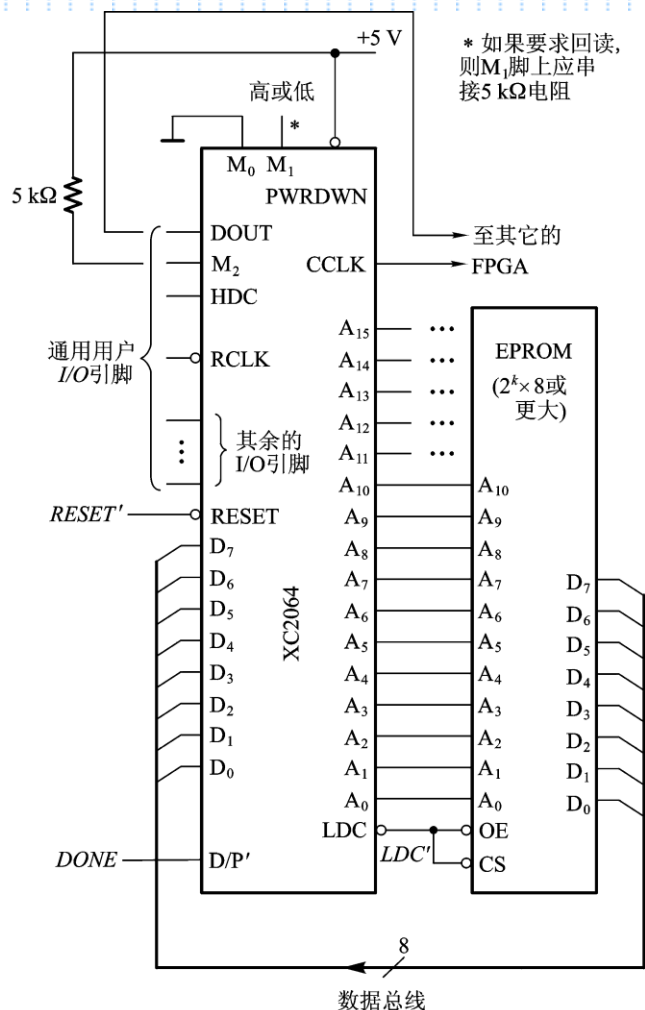
4. SRAM

分布式

每一位触发器控制一个编程点



二、编程数据的装载

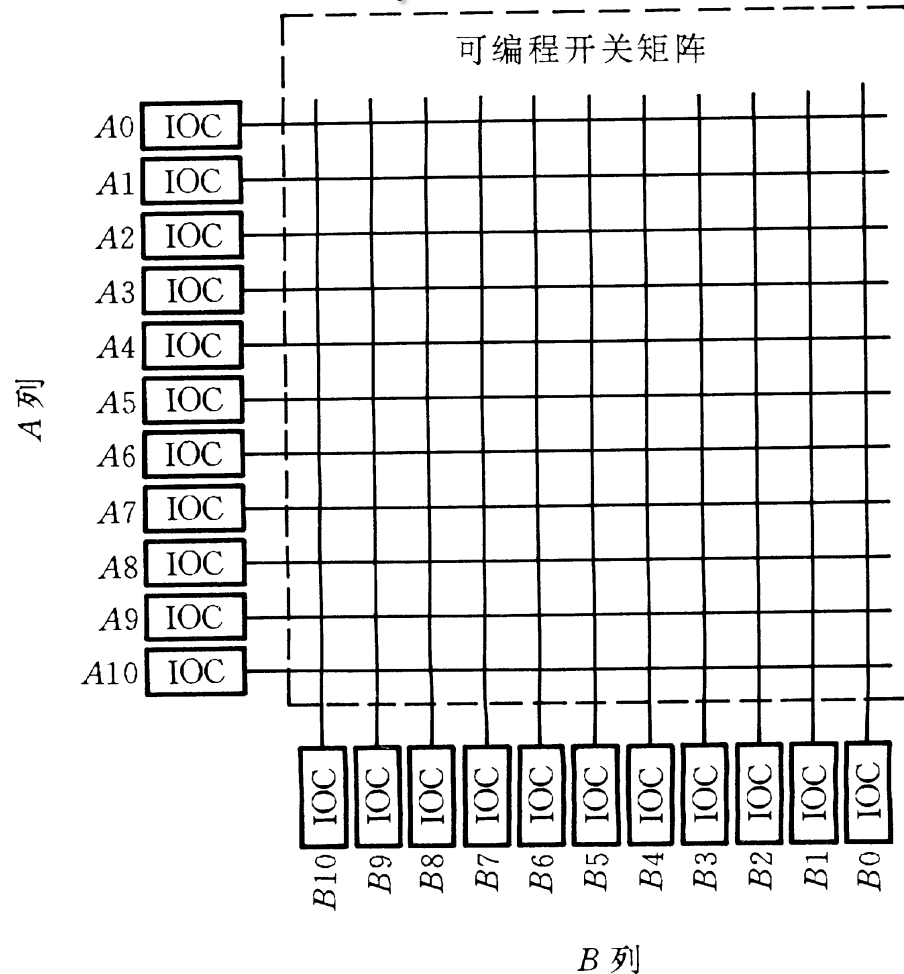
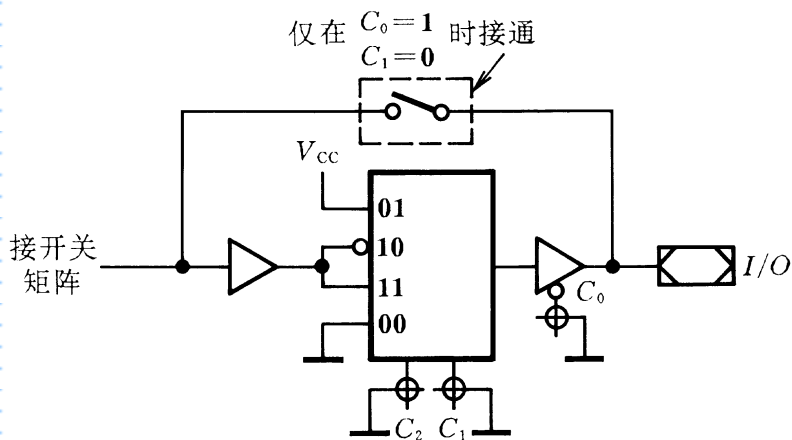


1. 数据可先放在EPROM或PC机中
2. 通电后，自行启动FPGA内部的一个时序控制逻辑电路，将在EPROM中存放的数据读入FPGA的SRAM中
3. “装载”结束后，进入编程设定的工作状态

!! 每次停电后，SRAM中数据消失
下次工作仍需重新装载

8.8 在系统可编程通用数字开关 (ispGDS)

ispGDS22的结构框图



8.9 PLD的编程

以上各种PLD均需离线进行编程操作，使用开发系统

一、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

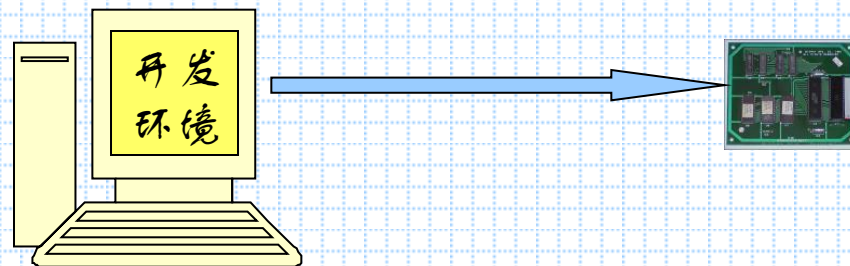
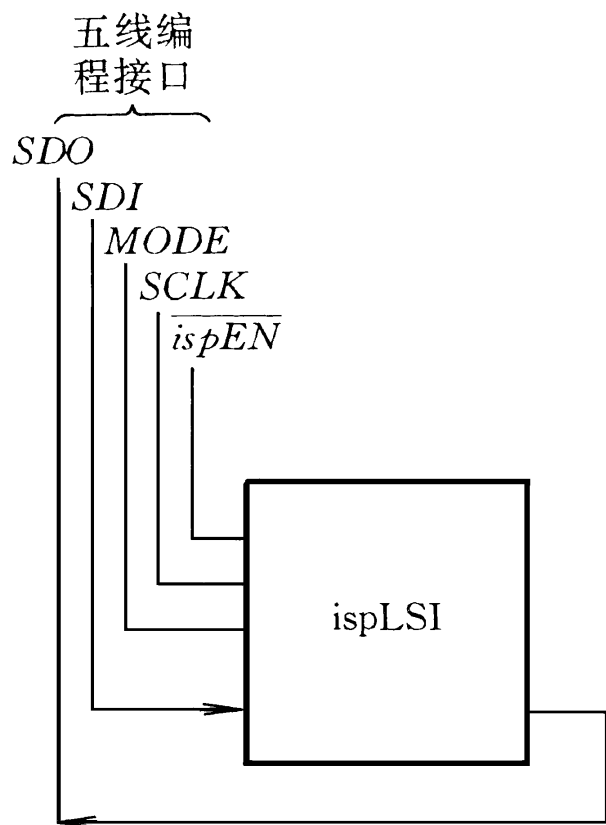
真值表，方程式，电路逻辑图（Schematic）

状态转换图（FSM）

二、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

isp器件的编程接口 (Lattice)



- 使用ispPLD的优点：
 - *不再需要专用编程器
 - *为硬件的软件化提供可能
 - *为实现硬件的远程构建提供可能