

## 可编程逻辑器件(PLD)

集成电路规模的大小，经常用其所包含的晶体管或逻辑门的数量来区分，如 1000 个元件或 100 个门以上的称为大规模集成电路。而我们讲过的编码器、译码器、计数器为中规模电路，反相器、与非门为小规模电路。

随着集成电路制造工艺的飞速发展，集成度不断提高因此划分的标准也不断提高，几十~几百门、几百~几千门、几千~几万门成为小、中、大的划分标准。

大规模集成电路的结构不一定是中小规模电路的简单集成，它通常常常作为一个相对独立的数字系统，完成某一方面的功能。典型的大规模集成电路有：微处理器、存储器、可编程逻辑器件等。

从应用角度来看，大规模集成电路可分为通用型和专用型两类。通用型由生产厂家设计制作，力求满足大多数用户的需要；专用型是根据某一用户的特殊要求而专门设计制作的。

设计专用集成电路有以下几种形式：

专用集成电路 **ASIC**（Application Specific Integrated Circuits）先由用户提出设计要求，或给出设计的软件，再由厂家制作。前期投入较大，适于大批量生产。

可编程逻辑器件 **PLD**（Programmable Logic Device）由用户自己完成设计和编程。单片成本相对较高，但几乎不需要前期投入，综合成本完全可以接受，适于小批量使用。

## 一、可编程逻辑器件

EPROM、E<sup>2</sup>PROM都是可编程器件，可以实现组合逻辑函数，但它们的主要用途是存储数据，所以将它们单列出去。

通常我们所说的可编程逻辑器件主要由以下几种：

可编程逻辑阵列（PLA）——“与阵列”、“或阵列”都可编程，更加灵活，“与阵列”可有选择地提供部分最小项，有利于简化电路。一次性编程。

可编程阵列逻辑（PAL）——“与阵列”可编程，“或阵列”固定，输出形式多种多样可选：三态、反馈、触发器，但不可全部兼得。一次性编程。

**通用阵列逻辑 (GAL)** ——“与阵列”可编程，“或阵列”固定（个别型号可编程），采用E<sup>2</sup>PROM和CMOS静态RAM技术，可以有限次重复使用。

GAL的输出采用逻辑宏单元，包括所有输出形式。不用选择不同类型的芯片。

以上三种属于早期的小规模的PLD器件，内部结构简单，工作速度快，适于完成较简单的逻辑功能。

**现场可编程门阵列 (FPGA: Field Programmable Gates Array)** ——在PAL、GAL的基础上发展起来的，规模较大，外部引脚多，内部资源丰富，有的包含几十万个逻辑门，可以构成一个独立的子系统。内部包含RAM结构，上电下载，掉电丢失。便于修改逻辑设计。

**复杂的可编程逻辑器件 (CPLD: Complex Programmable Logic Device)** ——与FPGA类似

# 1、PLA可编程逻辑阵列（与或阵列均可编程）

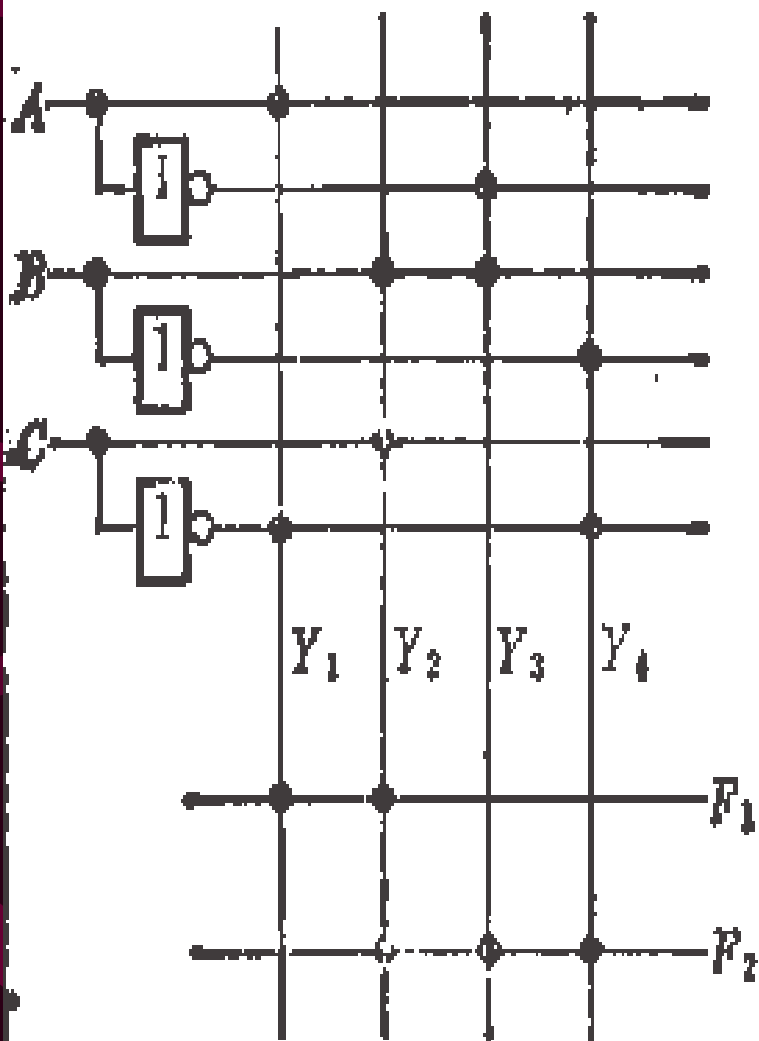


图 9-4-16 例 9-2 PLA 结构

四条与输出线，也可以解决三变量逻辑问题。

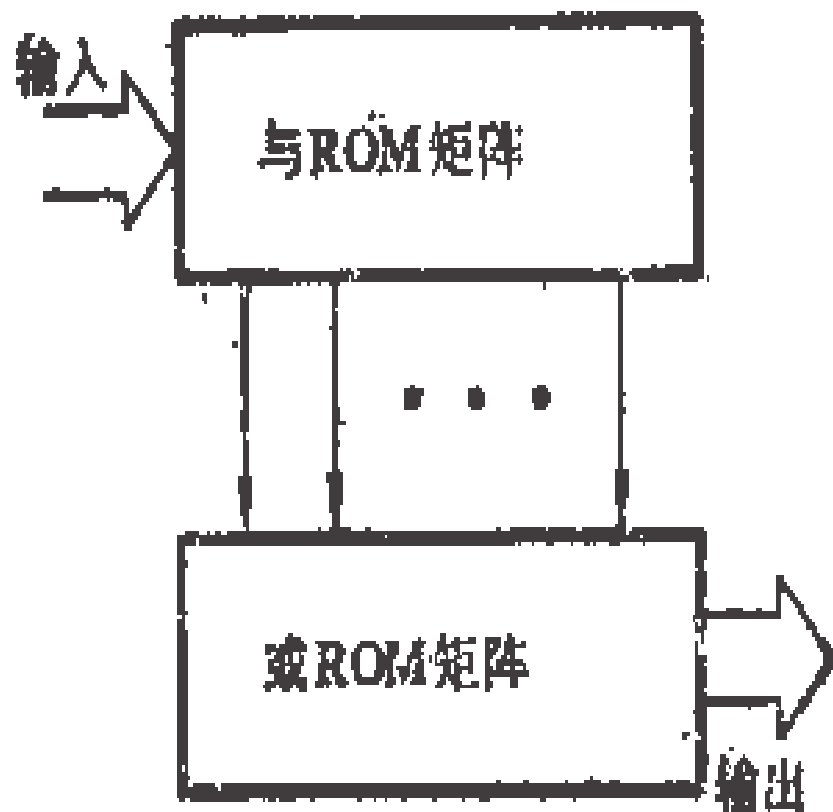


图 9-4-17 PLA 实现组合逻辑的框图

## 2、PAL可编程阵列逻辑

熔丝结构，与阵列可编程，或阵列固定。

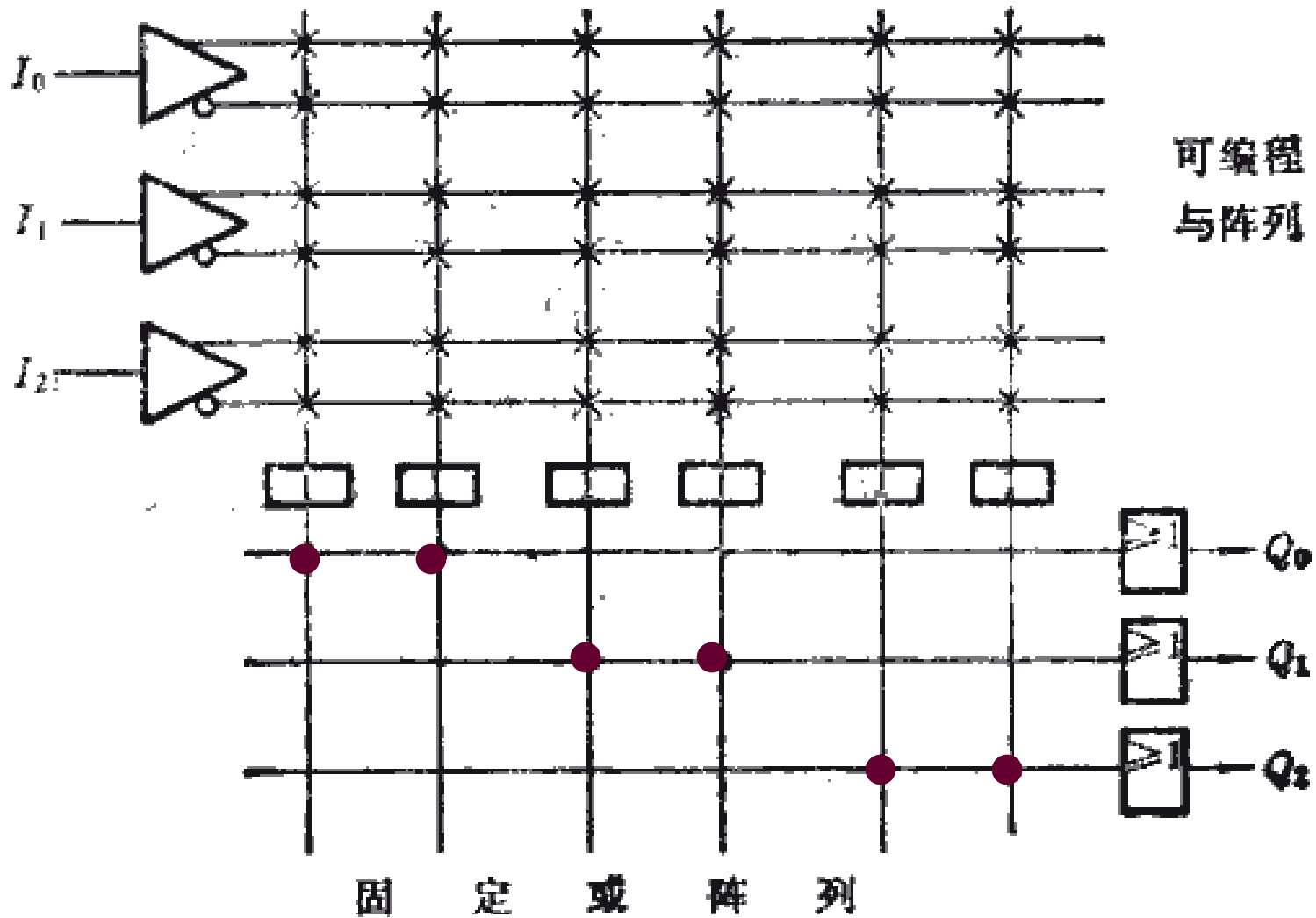


图 9-5-2 PAL 的基本门阵列结构

### 3、GAL通用阵列逻辑

GAL从阵列结构上分有两类：

PAL型---与阵列可编程，或阵列固定，与PAL一致。

PLA型---与、或阵列均可编程，与PLA一致。

GAL的电路特点是：

GAL采用E<sup>2</sup>PROM的浮栅技术，可以电擦除重复使用。

GAL采用逻辑宏单元（LMC），将输出、输入的多种形式集于一身，可通过编程选择，使芯片型号种类减少。

GAL的编程方式有两类：用专用编程器或在线编程。

GAL目前仍然在广泛应用，主要用于综合几种逻辑门电路，不规则译码器，实现专用逻辑功能等。

常用的GAL芯片有GAL16V8、GAL20V8等，下面我们简单介绍一下GAL的内部结构。以GAL16V8为例。



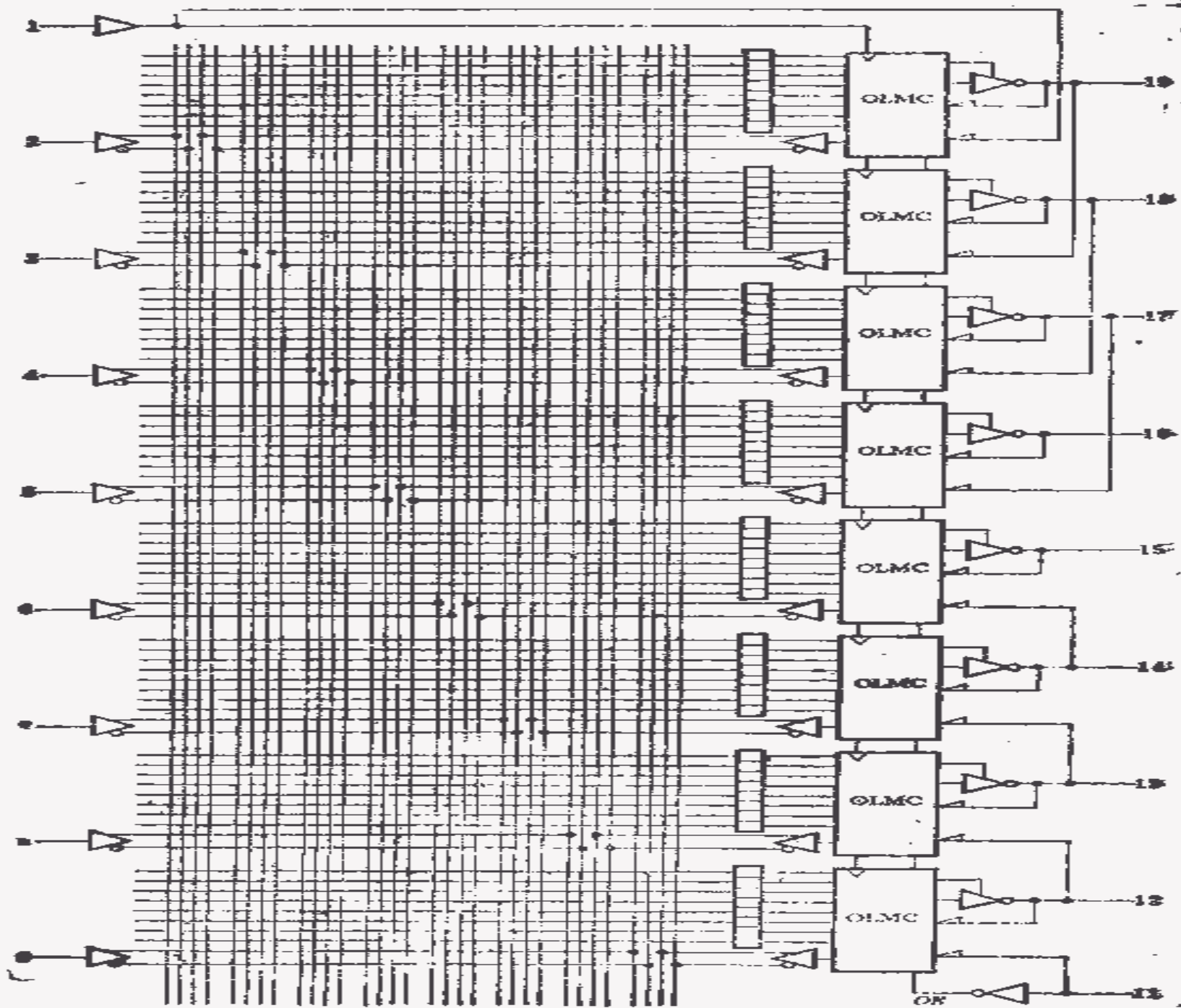


图 9-5-8 GAL 16V8 逻辑图



# GAL内部输出逻辑宏单元的内部结构

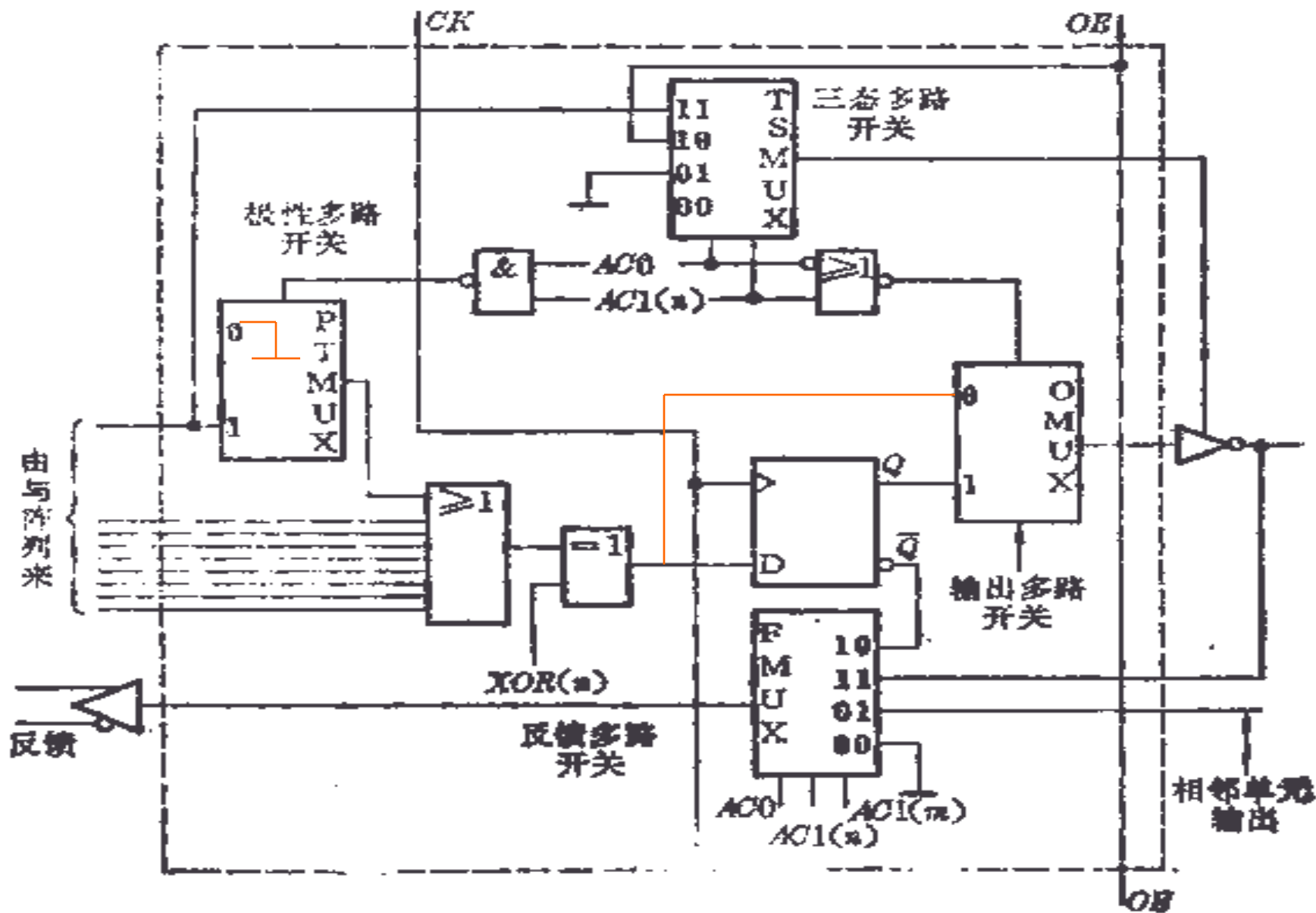


图 9-5-9 GAL 输出逻辑宏单元(OLMC)

**GAL的编程有相应的软件，可以用逻辑图的形式给出，也可以用特定格式的表达式描述。**

**无论用什么方式描述逻辑功能，最终都要编译成对应交叉点通断的编程文件，最后用编程器写入就可以了。**

**GAL设有加密位，加密后不能读出内部的编程状态。但目前有解密的方法，所以不可靠。**

## 4、复杂的可编程逻辑器件（CPLD）

随着集成工艺的发展，PLD的集成规模已经有了很大的改变。CPLD就是一种集成度远远高于PAL和GAL的电路。

表 8-3-1 部分 CPLD 产品

器件名称	集成规模/门	I/O 端数	宏单元数	触发器数	编 程
EPM9560	12 000	216	560	772	EEPROM
EPM5032	600	24	32	32	EPROM
EPF10K10	10 000	134	— <sup>(1)</sup>	720	SRAM
EPX8160	3 200	172	160	160	快闪 SRAM
AT5100	5 100	52	52	128	EPROM
ATV750	750	10	10	20	EPROM
pLSI3320	14 000	160	320	480	EEPROM
pLSI2032	1 000	32	32	32	EEPROM
M5-512	20 000	256	512	512	EEPROM
XC4025	25 000	192	— <sup>(2)</sup>	2 560	SRAM
XC7354	— <sup>(3)</sup>	54	54	108	EPROM

## CPLD有两大类：

**基本结构的CPLD**----仍然保持GAL的特点，与阵列加宏单元结构。主要是规模扩大，其次在相邻乘积项的利用、触发器结构方面也有所改进。

**分区阵列结构的CPLD**----将整个器件划分为若干个区域，每个区域相当于一个GAL，通过全局互连总线将各个区域连接起来。

下面介绍一下分区阵列结构的几种形式。

# 分区阵列结构有以下几种形式：

- 通用互连阵列UIM结构
- 多阵列矩阵MAX结构
- 灵活逻辑单元阵列FLEX结构
- 其他结构形式

# CPLD举例

(以FLEX结构的EPF10K20为例)

LE逻辑单元----最小的逻辑单位

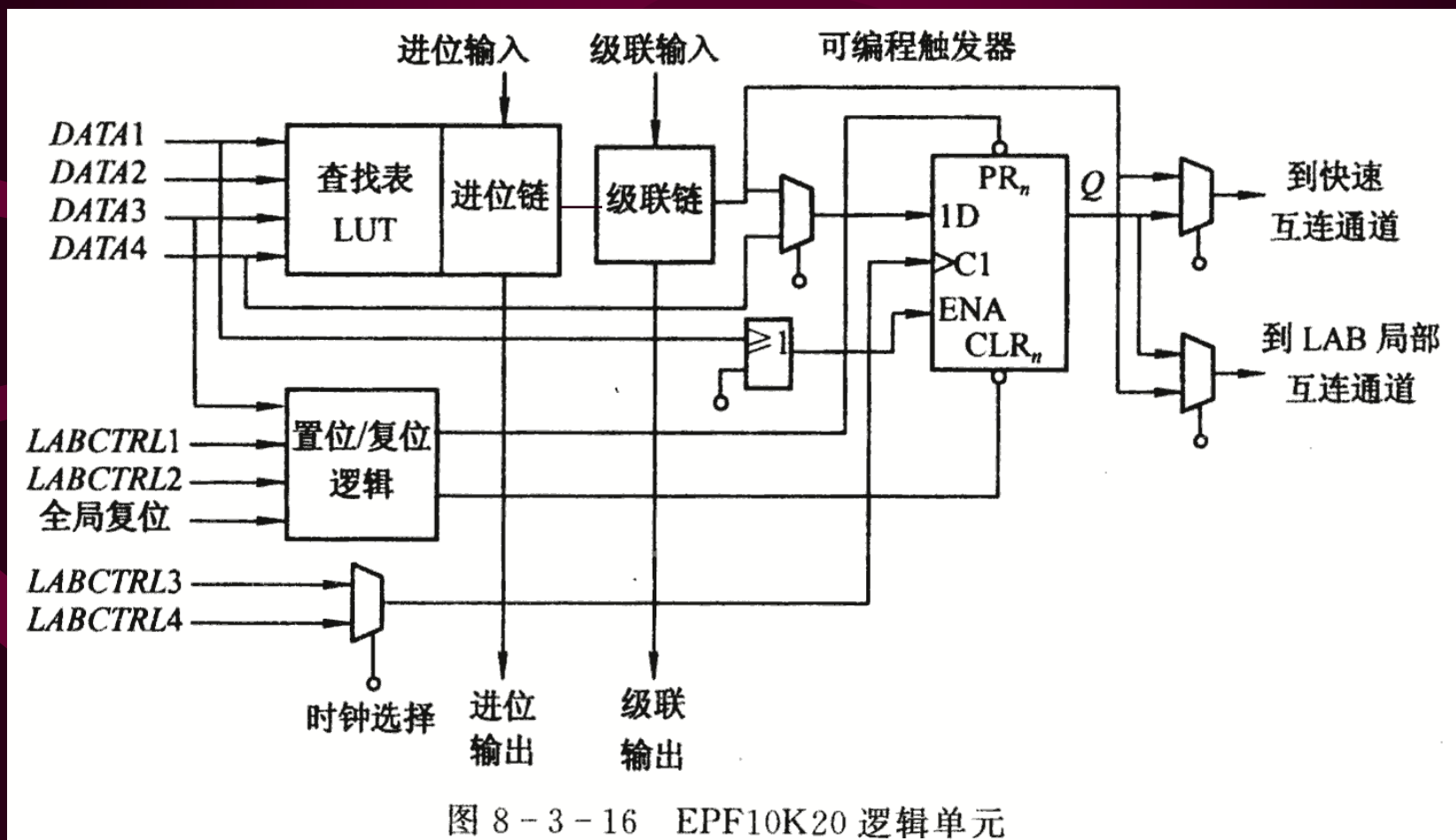


图 8-3-16 EPF10K20 逻辑单元

# 逻辑阵列块LAB

由8个LE及控制、互连、级联进位信号组成。

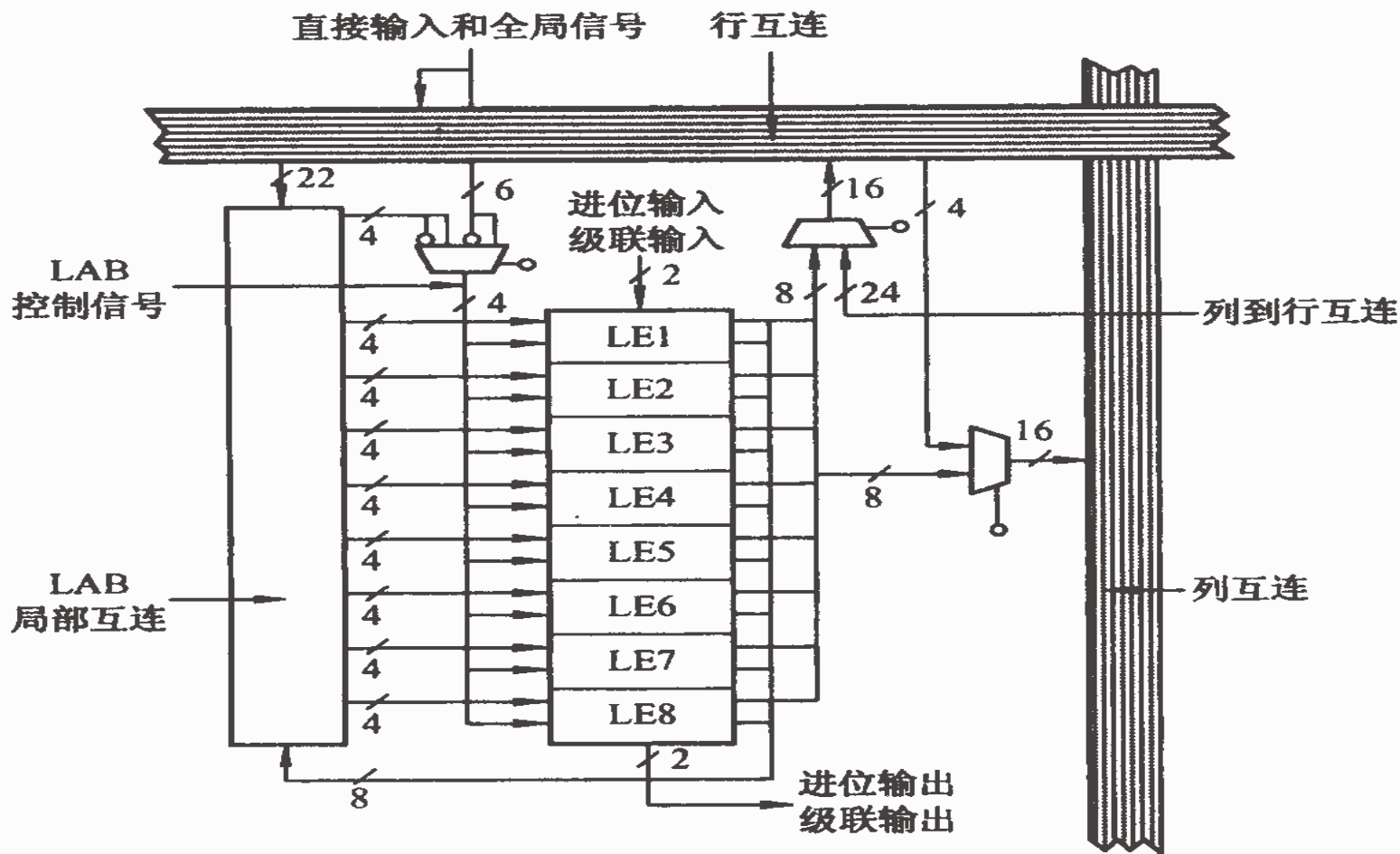


图 8 - 3 - 19 EPF10K20 逻辑阵列块 LAB



# 嵌入阵列块EAB

由输入输出端带有寄存器的 RAM / ROM 组成。

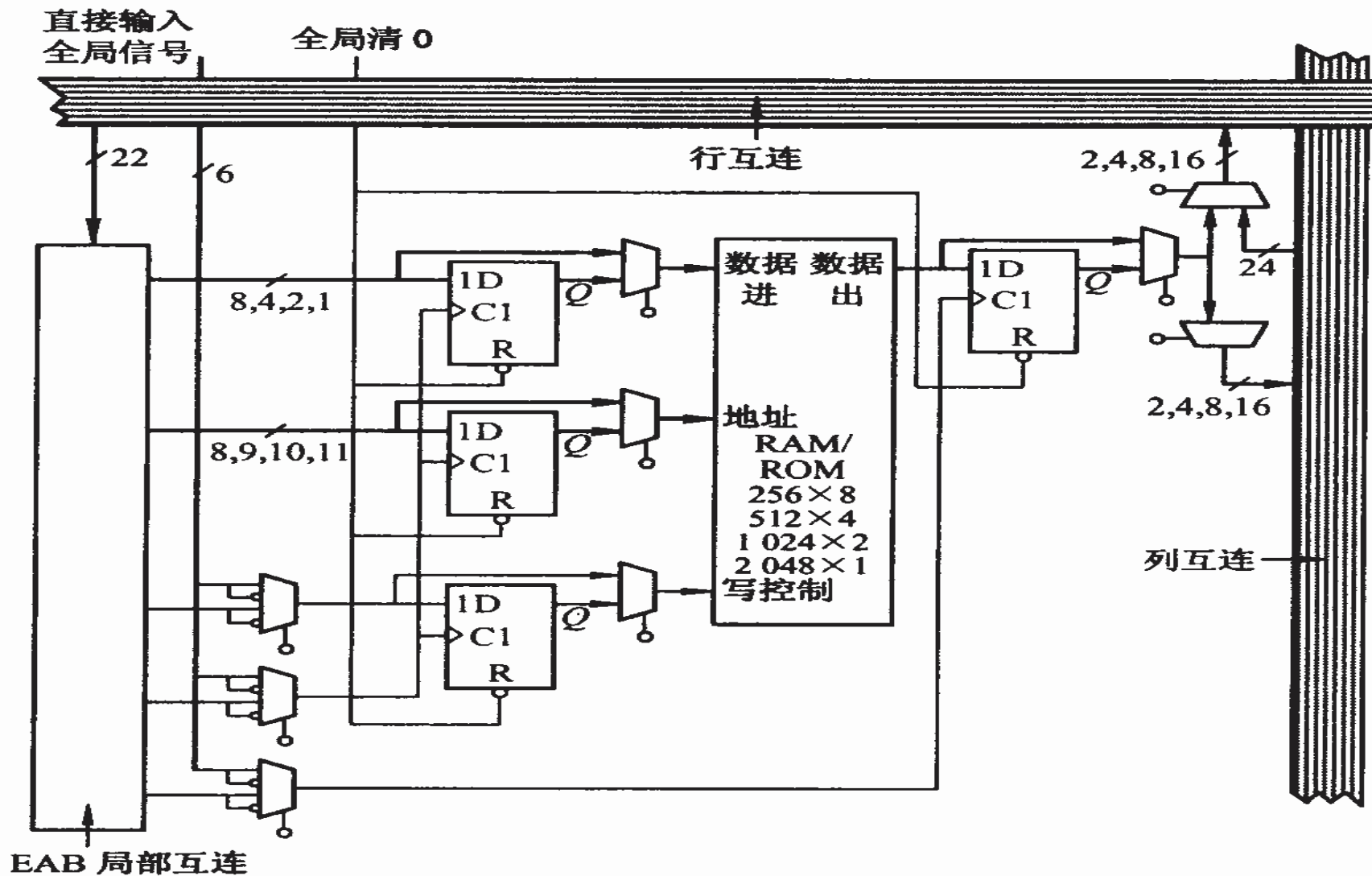


图 8-3-20 EPF10K20 嵌入阵列块 EAB

# CPLD的主要特点:

- 可重复编程、擦除或配置数据。
- 采用多种存储器类型EPROM、E<sup>2</sup>PROM、FLASH和SRAM等，高（密度、速度、可靠性），低功耗。
- 内部时间延迟固定、可预测，易消除冒险竞争。
- 有多级加密位，具有较好的保密性。

## 5、现场可编程门阵列 FPGA

FPGA是另一种高密度PLD芯片。它由三个可编程模块组成，编程的结果存放在一个SRAM中，所以需要上电时下载编程数据。

表 8-4-1 XC4000 系列器件主要特征

器 件	门数	CLB 阵列	CLB 数	触发器数	最大 RAM 位数
XC4003/A	3 000	10×10	100	360	3 200
XC4003H	3 000	10×10	100	200	3 200
XC4003E	3 000	10×10	100	360	3 200
XC4010	10 000	20×20	400	1 120	12 800
XC4025	25 000	32×32	1 024	2 560	32 768
XC4025E	25 000	32×32	1 024	2 560	32 768
XC4044EX	44 000	40×40	1 600	3 840	51 200
XC4062XL	62 000	48×48	2 304	6 376	73 728

# FPGA三个模块之间的关系如下：

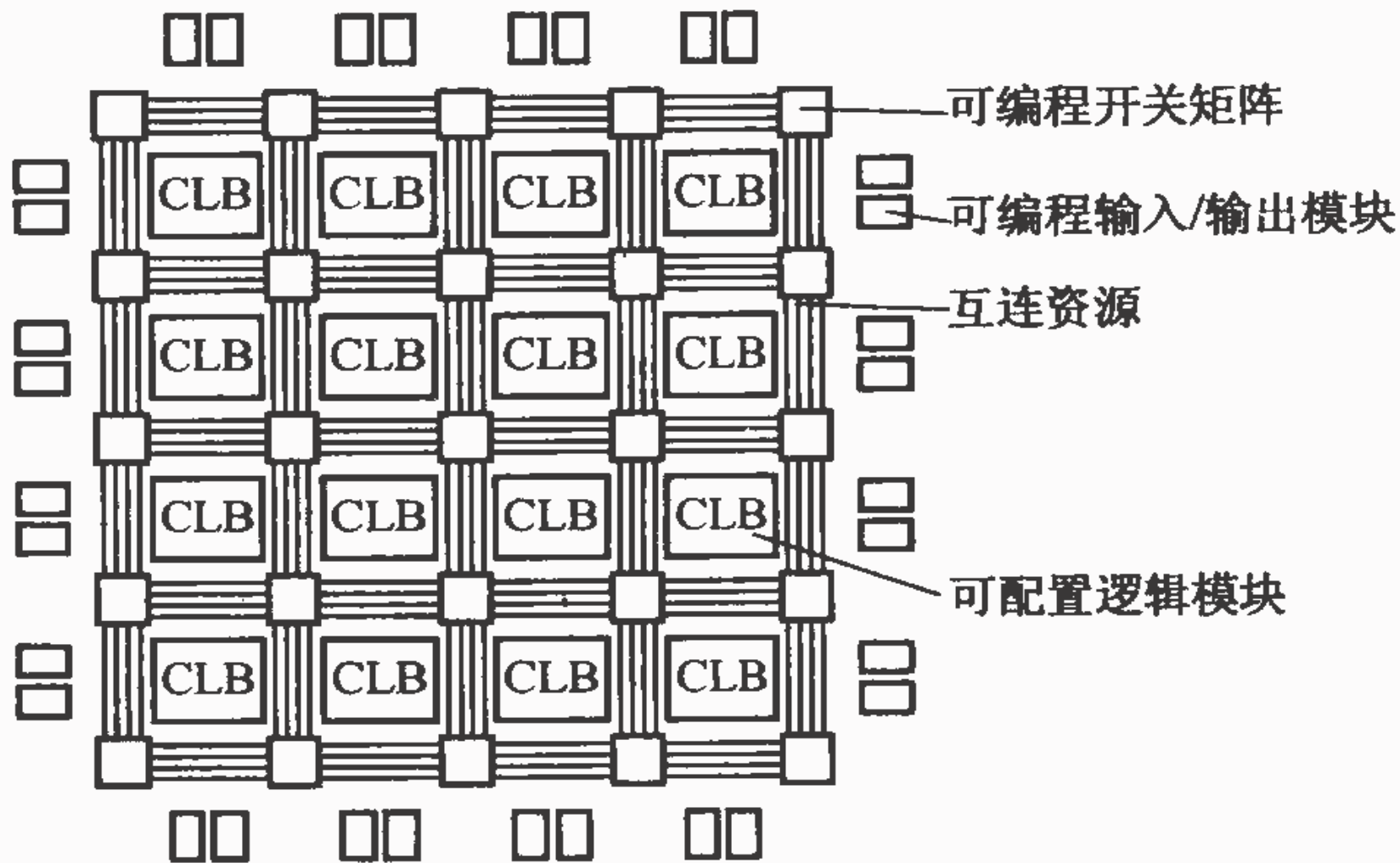


图 8 - 4 - 1 XC4000 系列 FPGA 基本结构

# 下面简单介绍FPGA的三种模块

## 可配置逻辑模块CLB

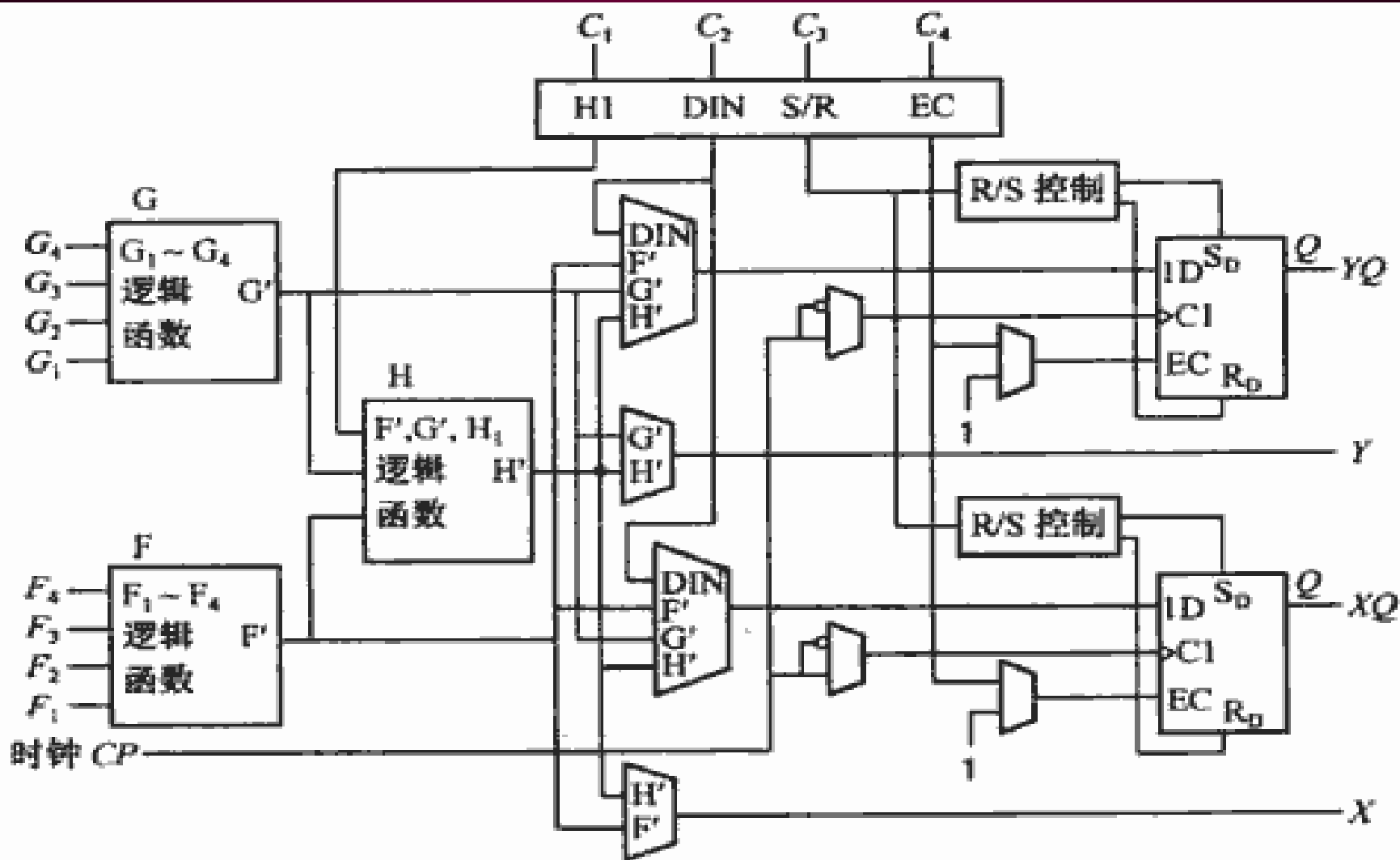


图 8-4-2 XC4000 系列 CLB 基本结构

# 可编程 I/O 模块 (IOB)

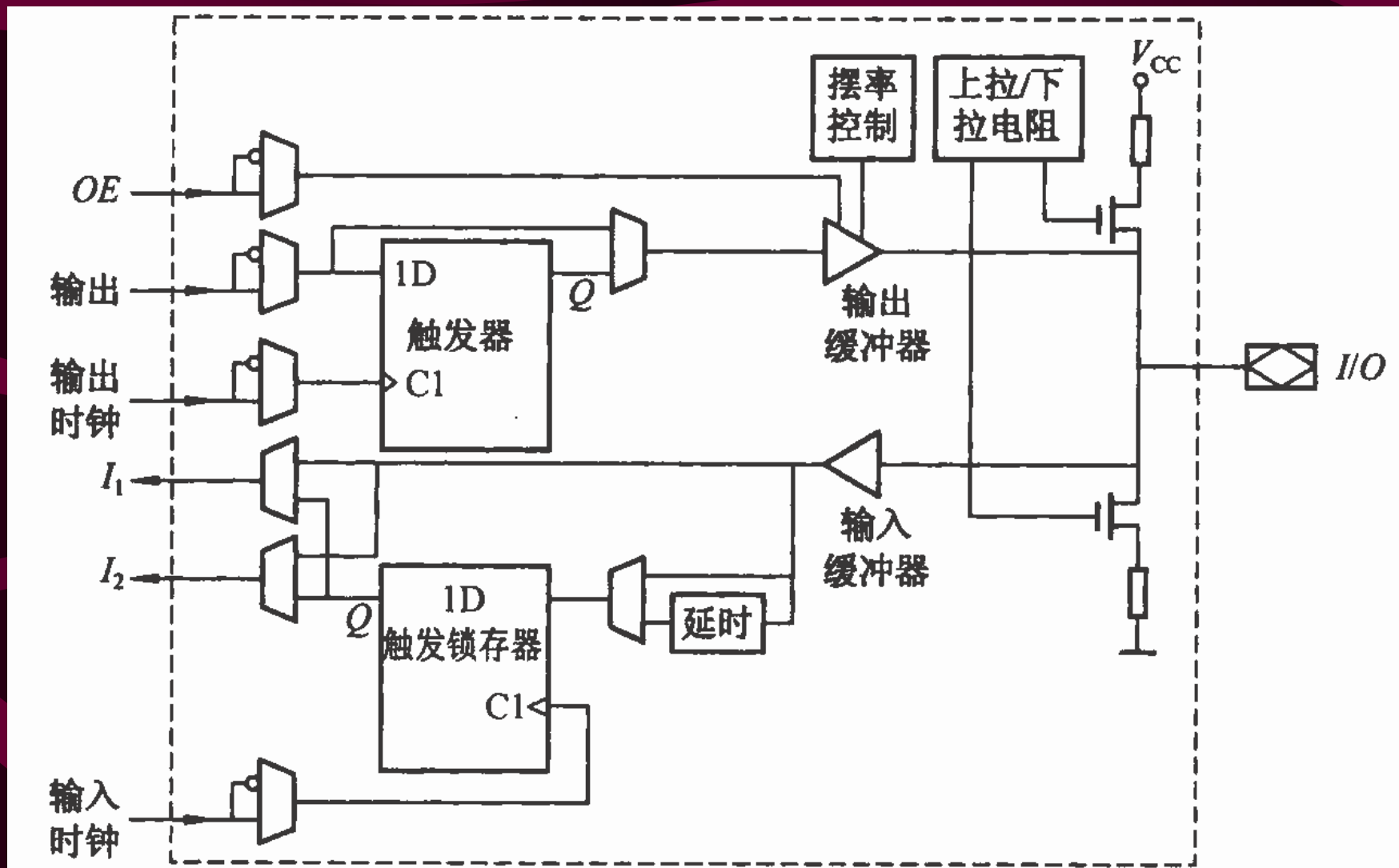
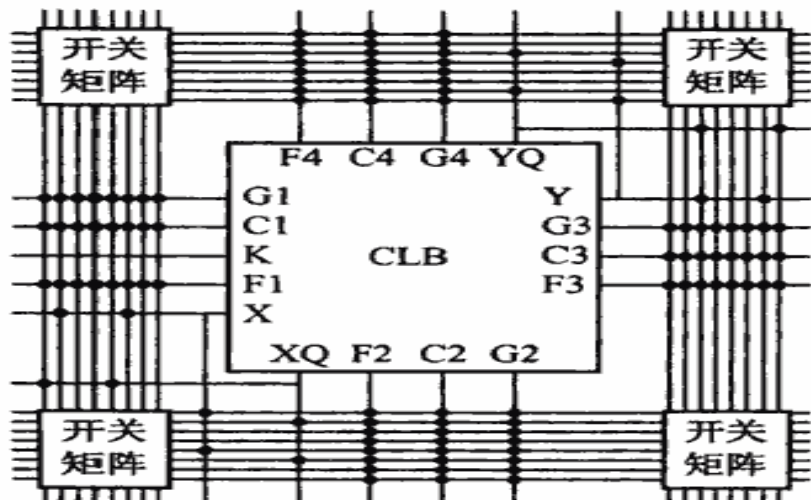


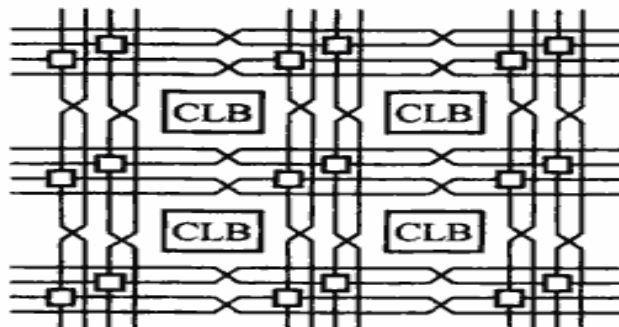
图 8-4-6 XC4000 的 IOB 结构

# 可编程内部互连资源 (ICR)

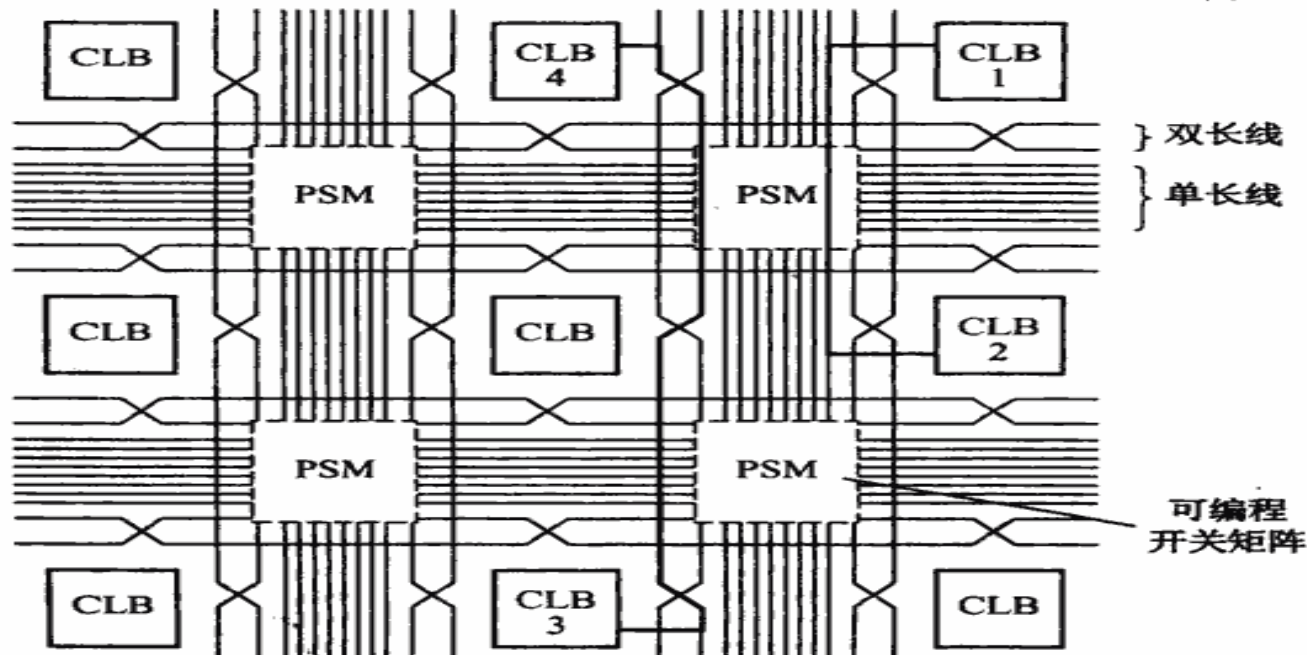


(a)

□ 开关矩阵



(b)



(c)



# FPGA的特点:

- 采用SRAM编程技术，高（密度、速度、可靠性），低功耗。但每次上电后要重新写入编程。
- 结构灵活，CLB、IOB和ICR均可编程，便于实现各种组合和时序逻辑电路。
- 不同延时的内部连线混合使用，使传输延时不可预测。

**FPGA和CPLD的设计编程都有相应的软件平台，可以方便地完成以下几个功能：**

- **设计输入** 用逻辑图或硬件描述语言
- **前仿真** 布局布线前，先验证电路功能是否有效。
- **编译** 将高层行为设计转为底层网表数据。
- **优化** 简化电路规模。
- **布局布线** 针对FPGA或CPLD作出具体规划设计。
- **后仿真** 根据实际布局布线的RC参数再次验证电路功能、时序。
- **编程** 对芯片进行实际连线。

## 附加说明：VHDL硬件描述语言

硬件描述语言（HDL=Hardware Description Language）是一种用来描述硬件电路功能的规范性语言。它和编写计算机程序所用的高级语言十分相似。有源文件，可编译成编程数据，有库资源可供利用。

用HDL设计逻辑电路已成为一种崭新的硬件电路设计方法。

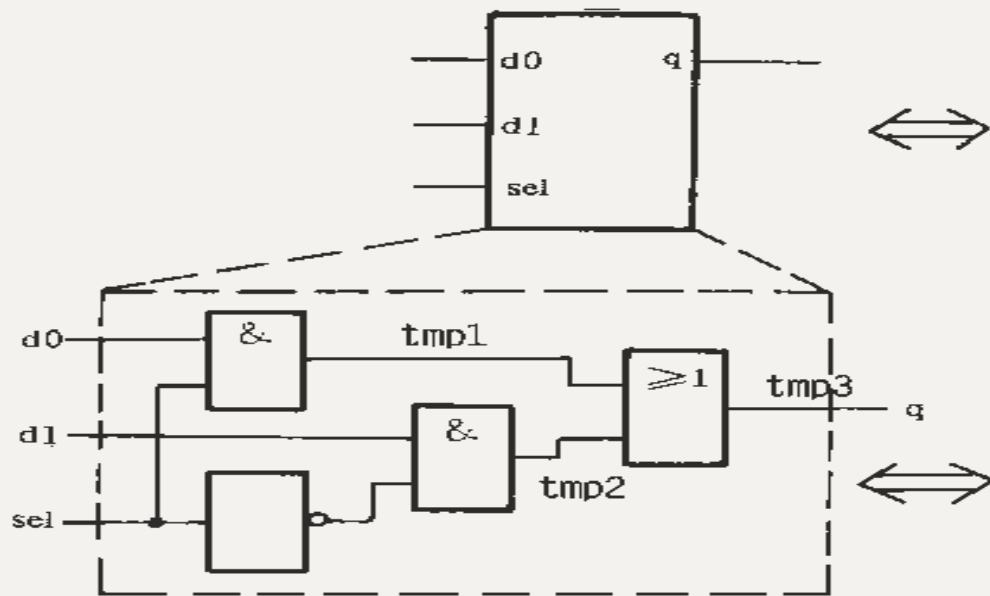
硬件描述语言有多种形式，互不兼容。由美国国防部设计的VHDL语言是最具代表性的，已成为一种标准的硬件描述语言。

用VHDL设计硬件电路有许多优点，如比原理图容易阅读、表达形式简单、易于修改、易于保存，便于借用已有设计（库调用），等等。

学习VHDL语言的语法格式、规则有专门的课程，也可以自己看书学习。希望同学们有时间关心一下有关这方面的内容。

- 下面举一个简单的例子，说明用VHDL描述硬件电路的方法。

例如：用VHDL语言定义一个二选一数据选择器。



(a)

```

ENTITY mux IS
    ...
END mux ;

```

```

ARCHITECTURE struct
    ...
END struct ;

```

```

ENTITY mux IS
    PORT(d0, d1, sel; IN BIT;
         q: OUT BIT);
END mux;
ARCHITECTURE connect OF mux IS
BEGIN
    calc; PROCESS(d0, d1, sel)
        VARIABLE tmp1, tmp2, tmp3; BIT;
    BEGIN
        tmp1 := d0 AND sel;
        tmp2 := d1 AND (NOT sel);
        tmp3 := tmp1 OR tmp2;
        q <= tmp3;
    END PROCESS;
END connect;

```

(b)

实体：说明外部引脚的定义  
三个输入、一个输出。

构造体：说明逻辑电路  
和逻辑关系

图 1 - 3 二选一选择器描述

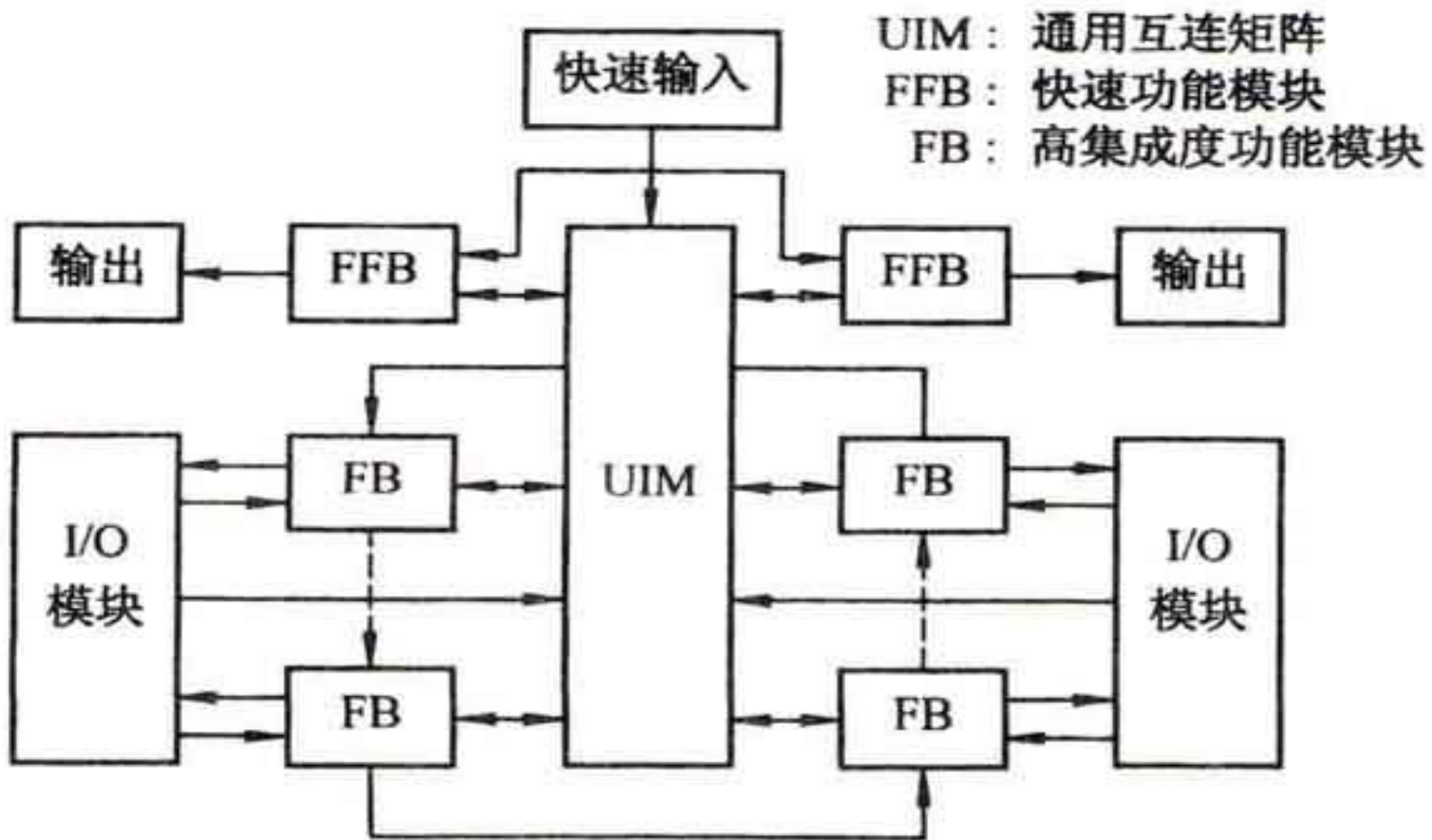


图 8-3-4 通用互连阵列 UIM 结构

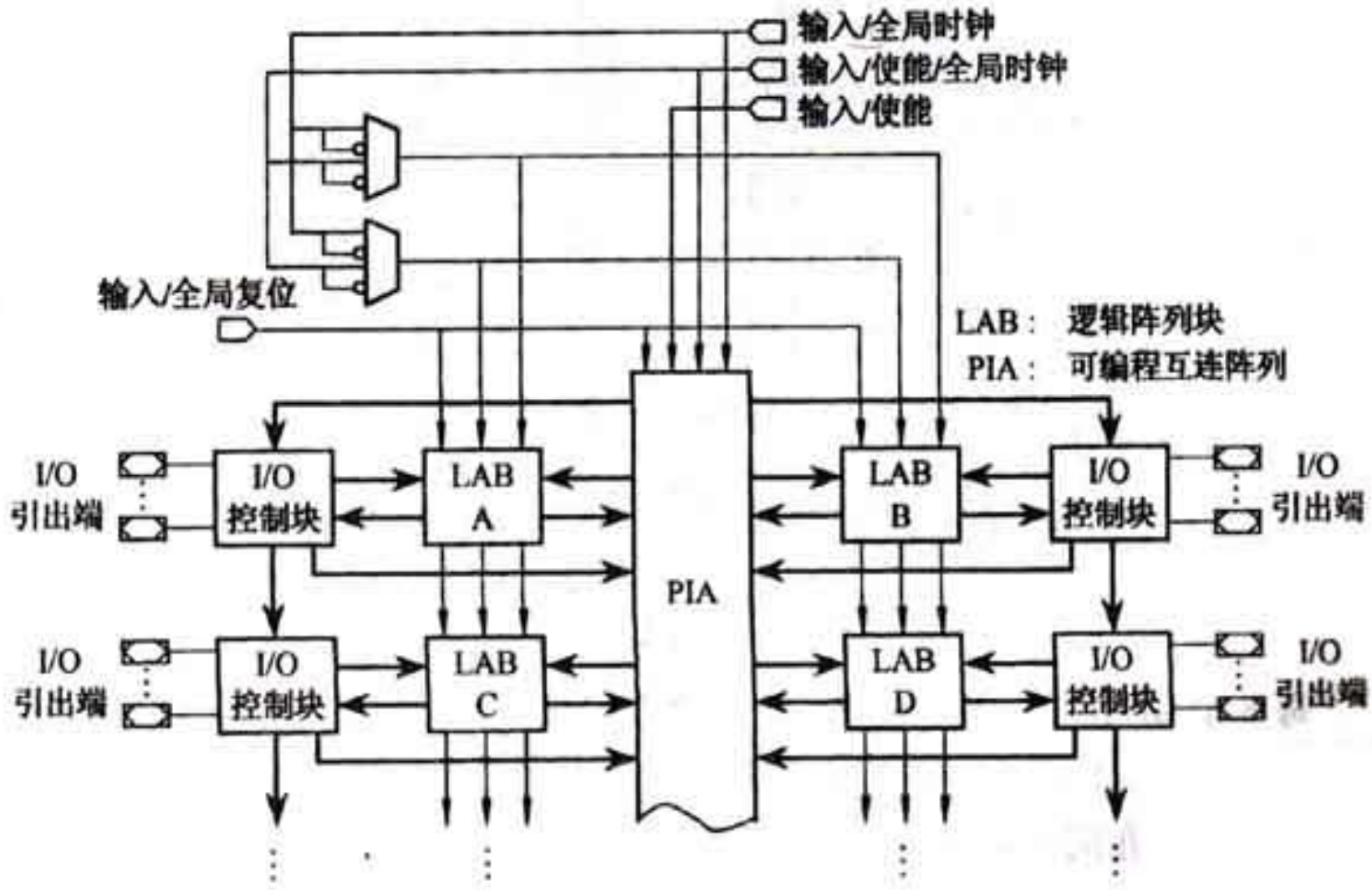


图 8-3-5 多阵列矩阵 MAX 结构



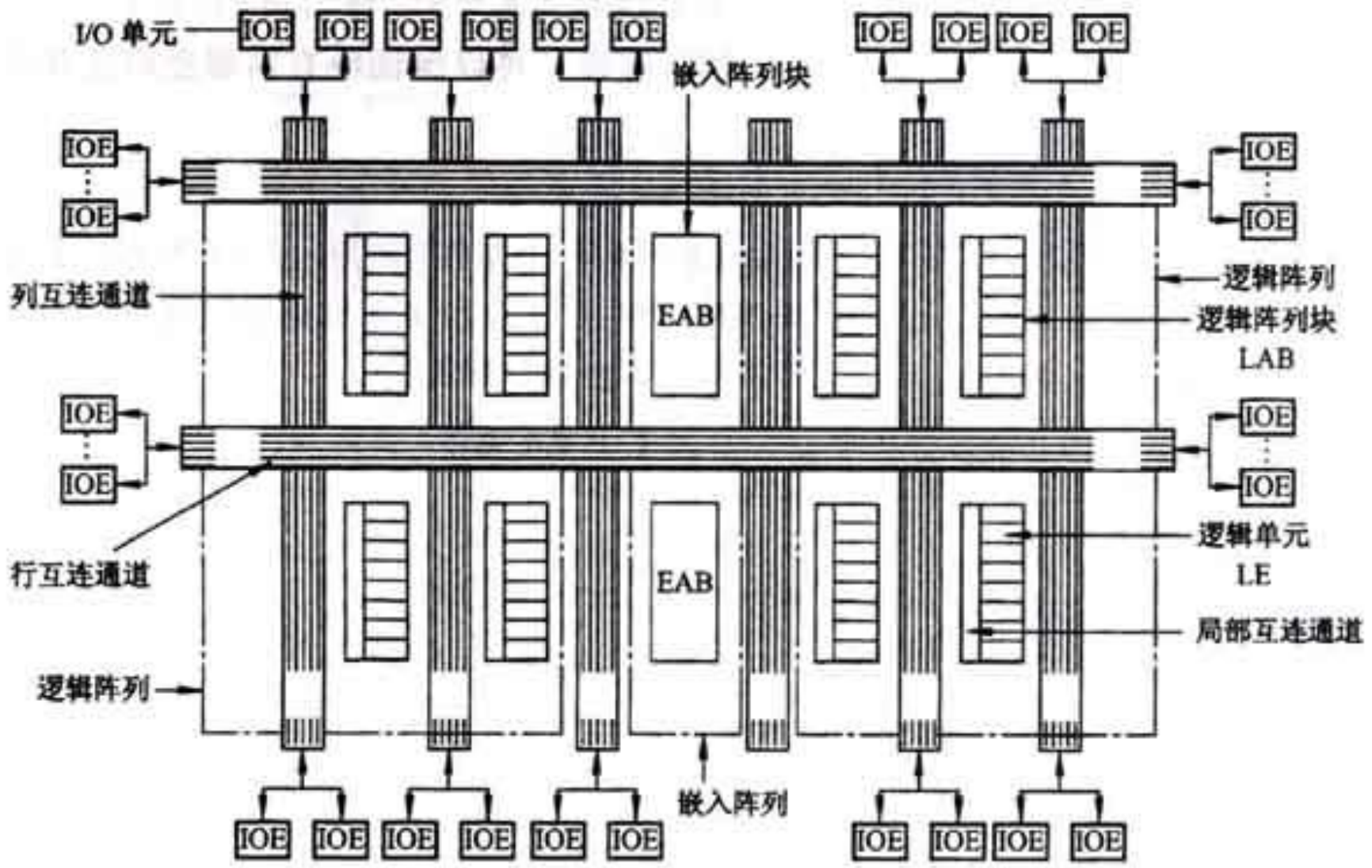


图 8-3-6 灵活逻辑单元阵列 FLEX 结构