
可编程逻辑设计

许芳

2021-2022-2

6 可编程逻辑器件

6.1 概 述

6.2 简单可编程逻辑器件

6.3 复杂可编程逻辑器件

6.4 现场可编程门阵列

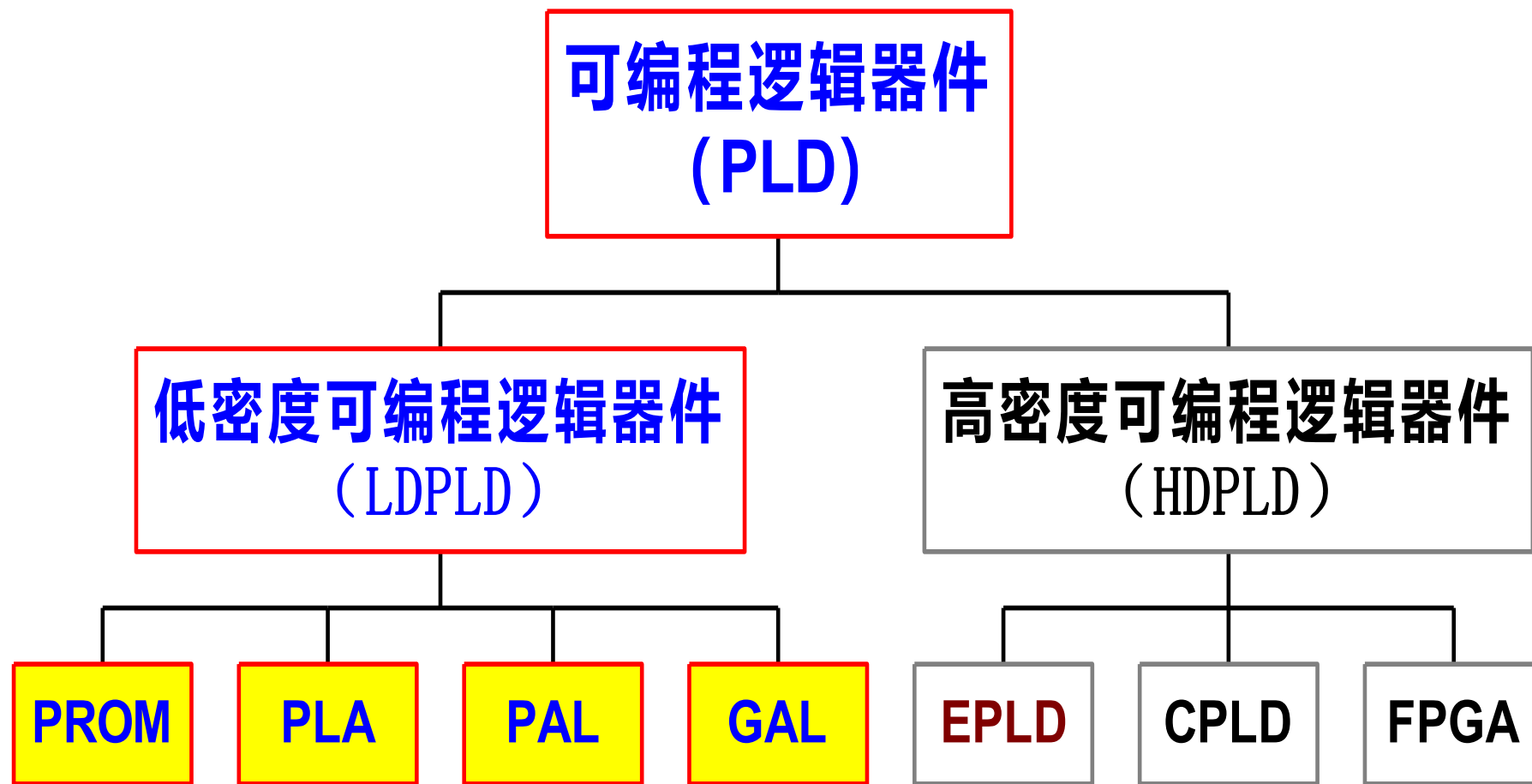
6.1 概述

可编程逻辑器件是一种可以由用户定义和设置逻辑功能的器件。该类器件具有逻辑功能实现灵活、集成度高、处理速度快和可靠性高等特点。

- 70年代，出现熔丝编程结构PROM和PLA
- 70年代末，AMD推出PAL
- 80年代初，Lattice推出GAL
- 80年代中期，Xilinx推出FPGA; Altera推出EPLD
- 80年代末，Lattice提出ISP技术
- 90年代，出现CPLD—EPLD改进型器件
- 2000年以后，出现基于FPGA的SOC

6.1 概述

■ 按集成密度分为



6.1 概述

■ 按结构分为

- 基于与/或阵列结构的器件SPLD（PROM、PLA、PAL、GAL）、CPLD（EPLD）,并称之为PLD。
- 基于门（逻辑单元）阵列结构的器件（FPGA：Field Programmable Gate Array）

表 2-5 4 种 PLD 器件的区别

器件名	“与”矩阵	“或”矩阵	输出
PROM	固定	可编	
PLA	可编	可编	
PAL	可编	固定	I/O 可编
GAL	可编	固定	宏单元

6.1 概述

■ 按编程工艺分为

1. 熔丝和反熔丝编程器件。如：**Actel**的**FPGA**器件。
2. **SRAM** 器件。如：**Xilinx**的**FPGA**器件。
3. **EPROM**器件，即紫外线擦除/电编程器件。
如大多数的**EPLD**器件。
4. **EEPROM**器件。如：**GAL**、**CPLD**器件。

6.1.2 PLD开发流程简介

所谓“在系统可编程”是指未编程的**ISP**器件可以直接焊接在印制电路板上，然后通过计算机的数据传输端口和专用的编程电缆对焊接在电路板上的**ISP**器件直接编程，从而使器件具有所需要的逻辑功能。

- 早期，使用编程器对器件进行编程。
- 后来，使用计算机的并行口编程。
- 现在，通常用计算机的**USB**口进行编程。

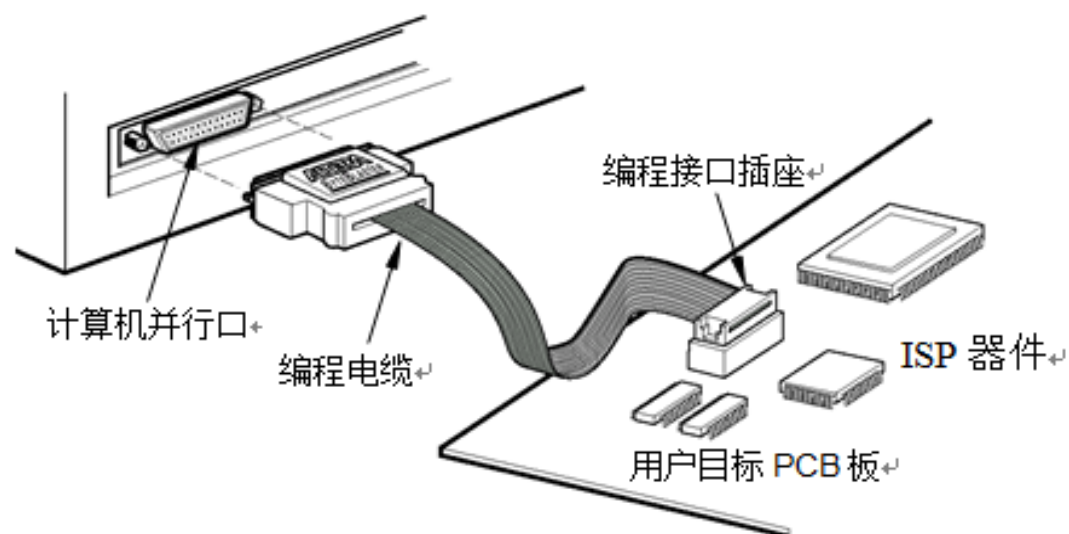


图6.1.1 在系统编程示意图

6.1.2 PLD开发流程简介

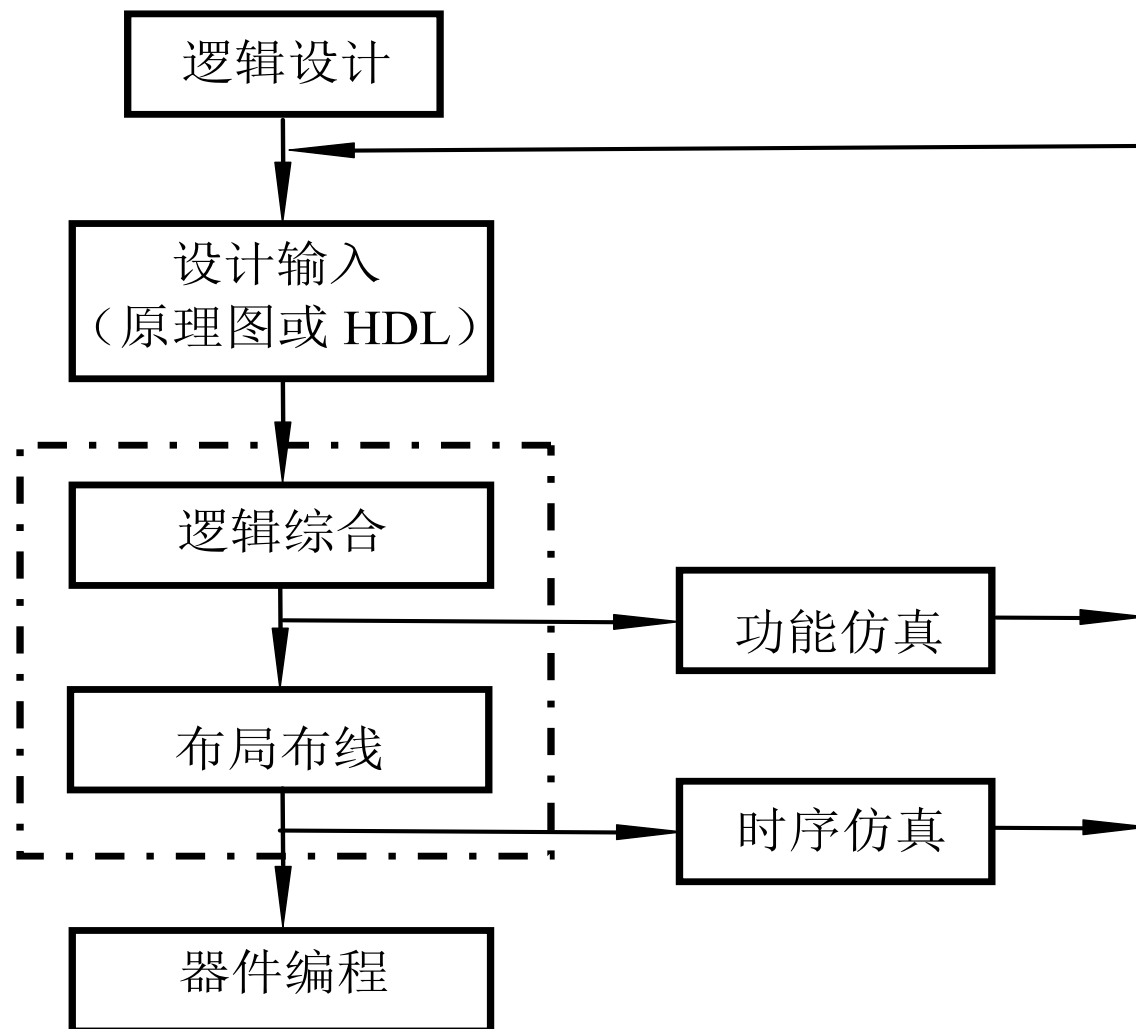
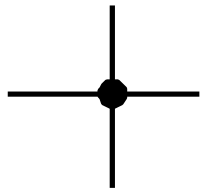


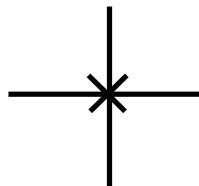
图6.1.2 可编程逻辑器件的开发流程

6.1.3 PLD器件的符号

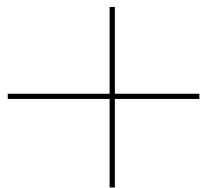
(1) 连接的方式



硬线连接单元

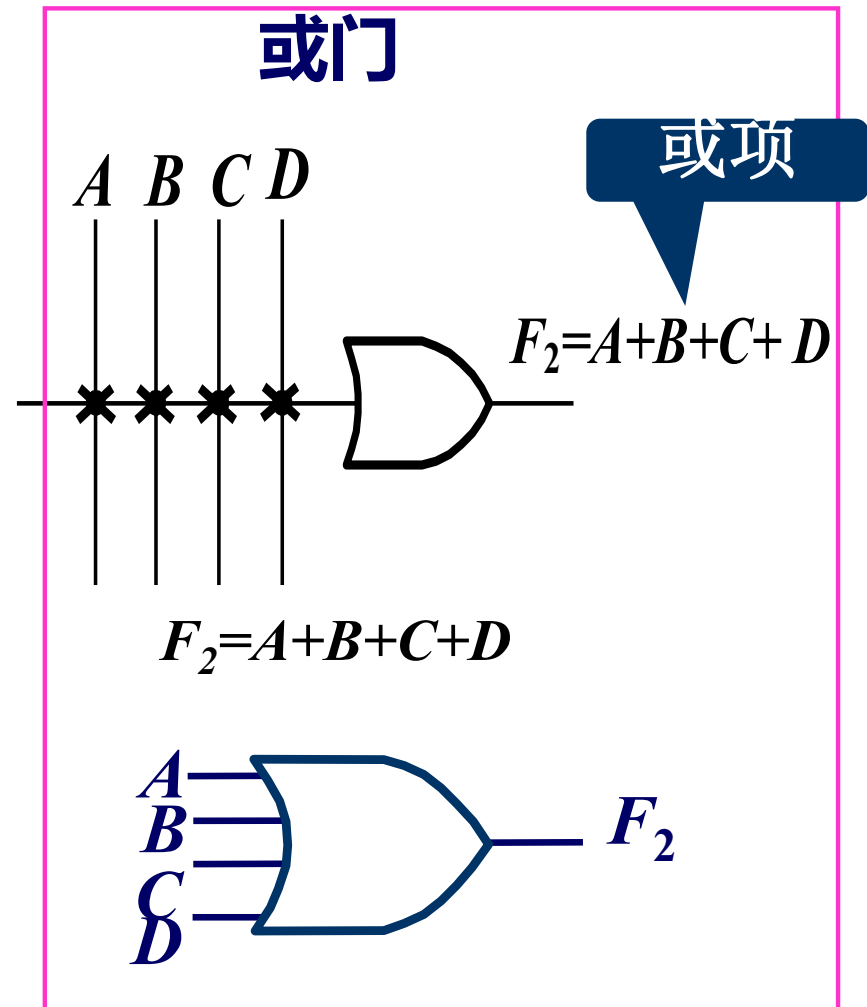
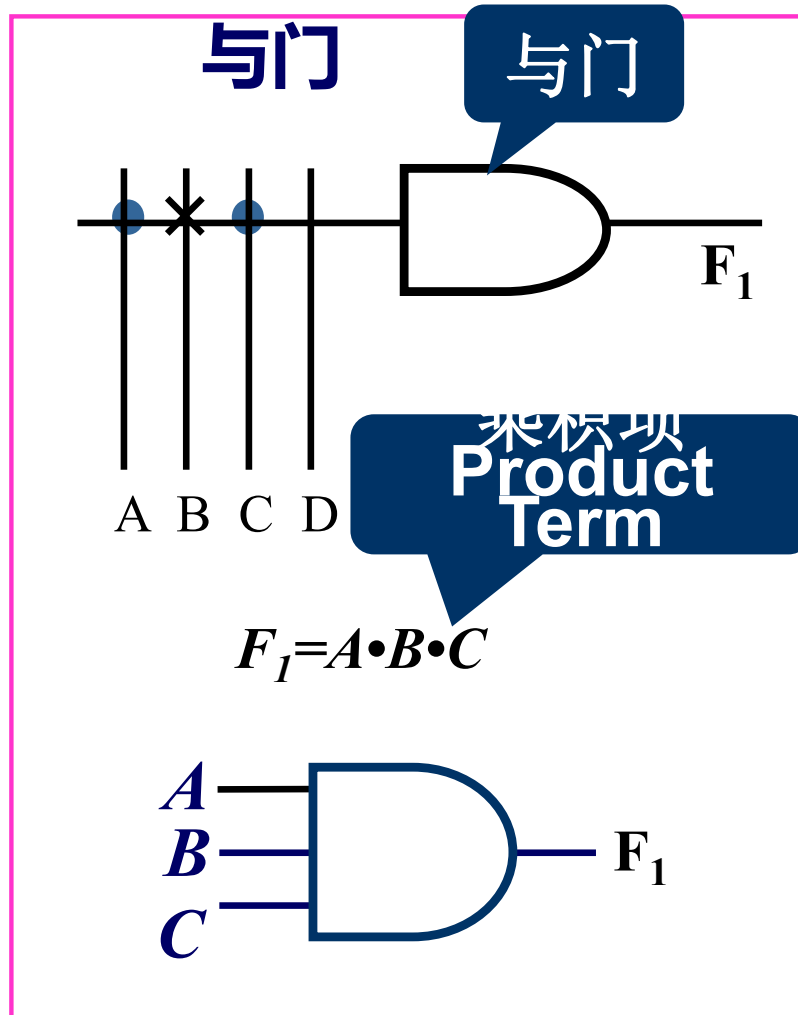


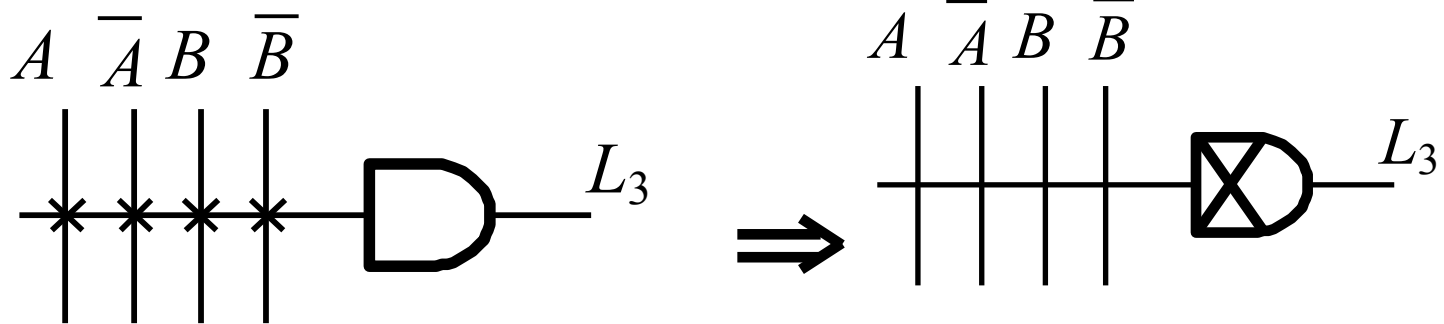
被编程接通单



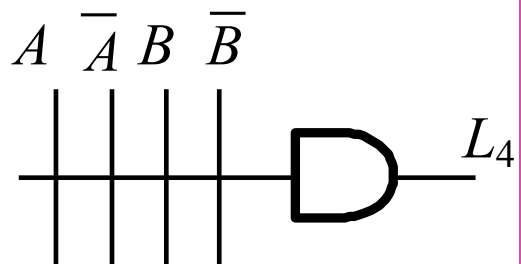
被编程擦除单元

(2)基本门电路的表示方式

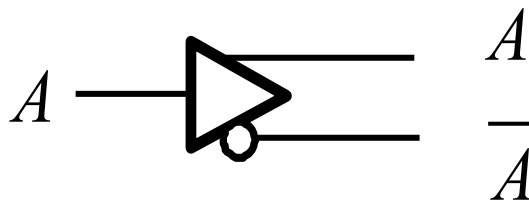




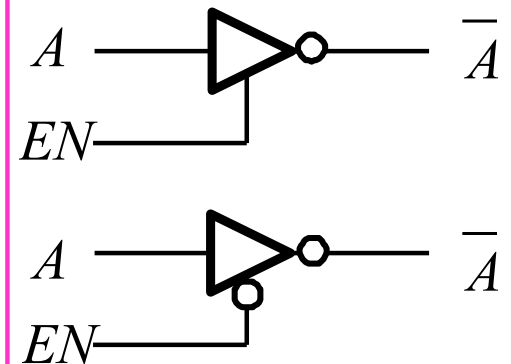
输出恒等于0的与门



输出为1的与门

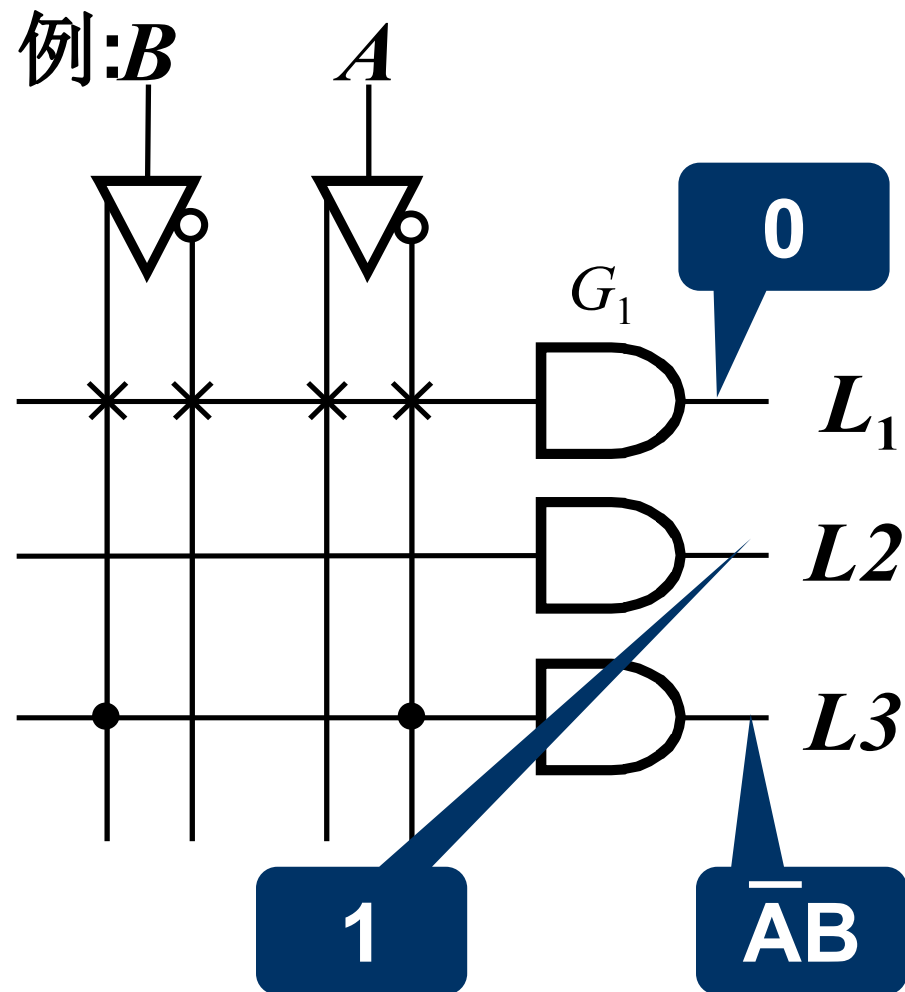
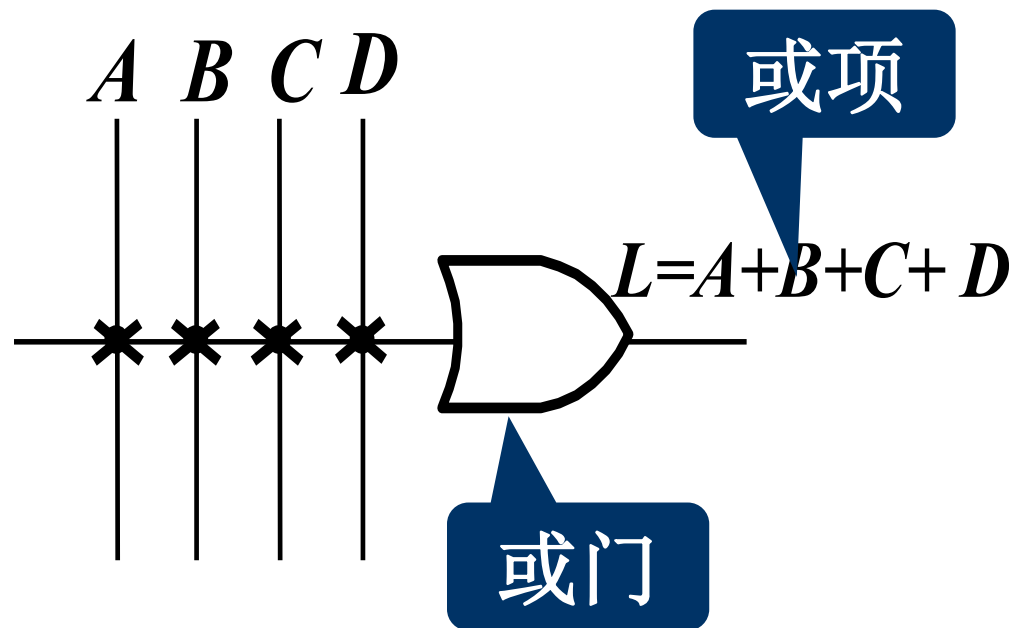


输入缓冲器



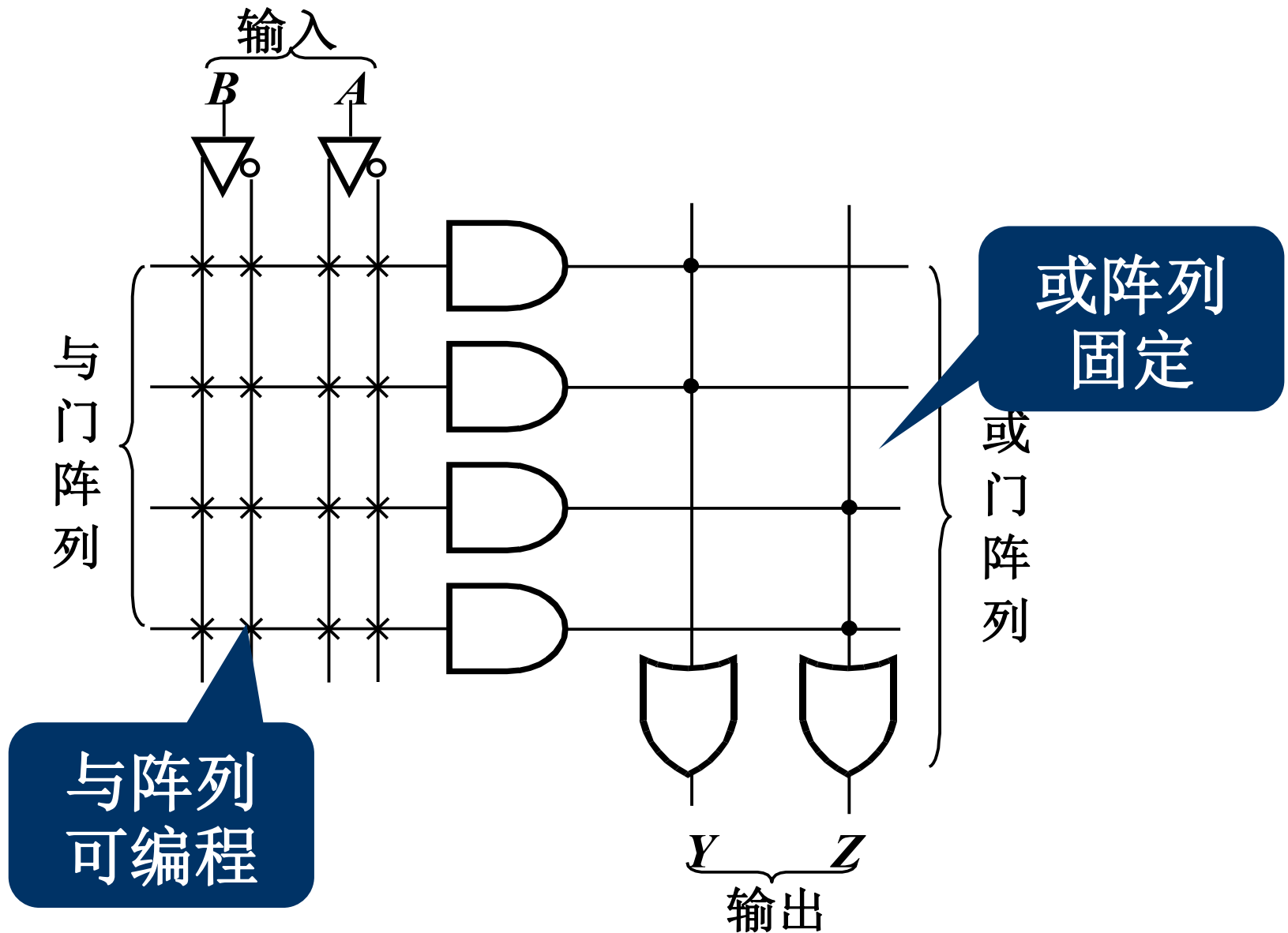
三态输出缓冲器

PLD的电路表示法（续）



PLD的电路表示法（续）

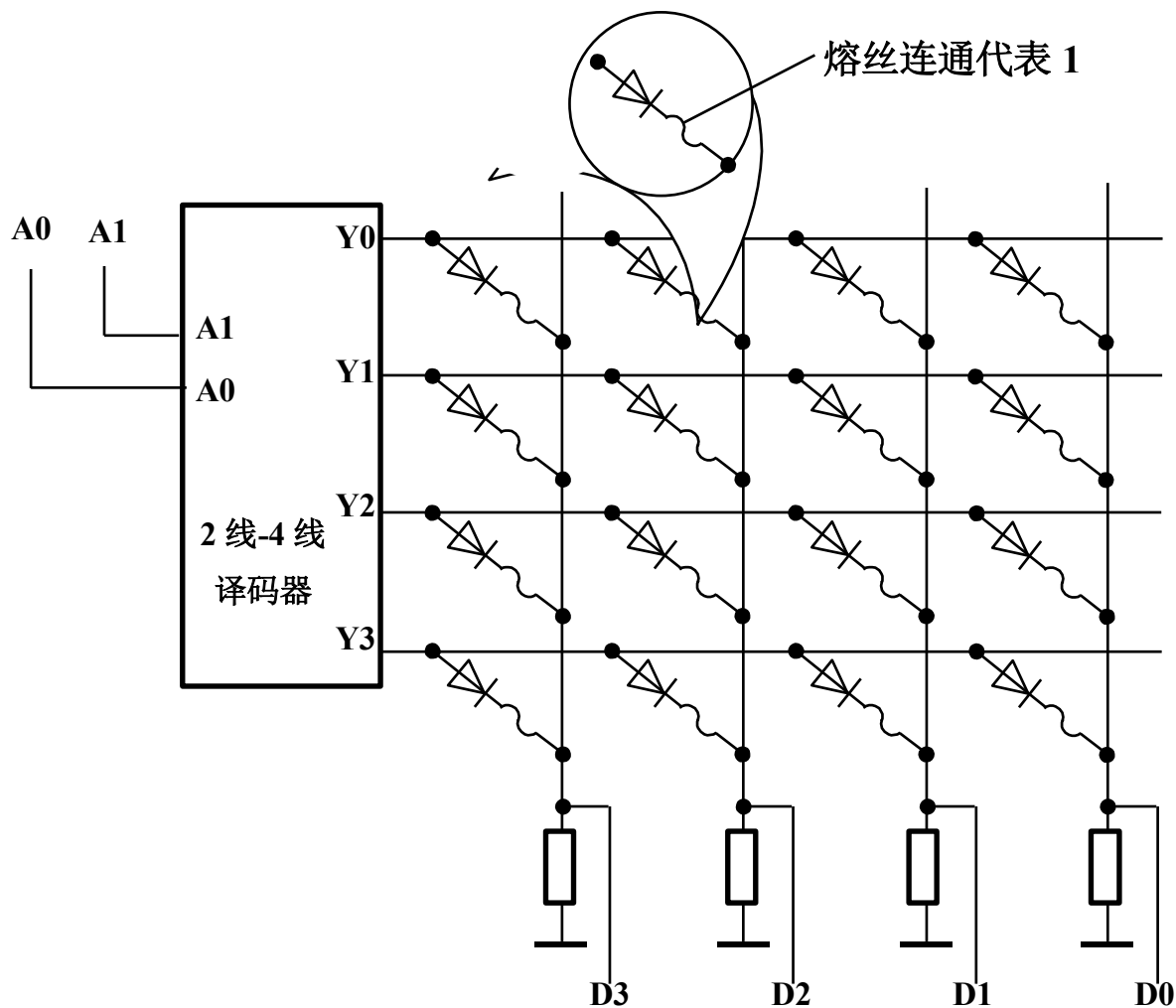
基本的PLD结构



可编程元件

1. 熔丝开关

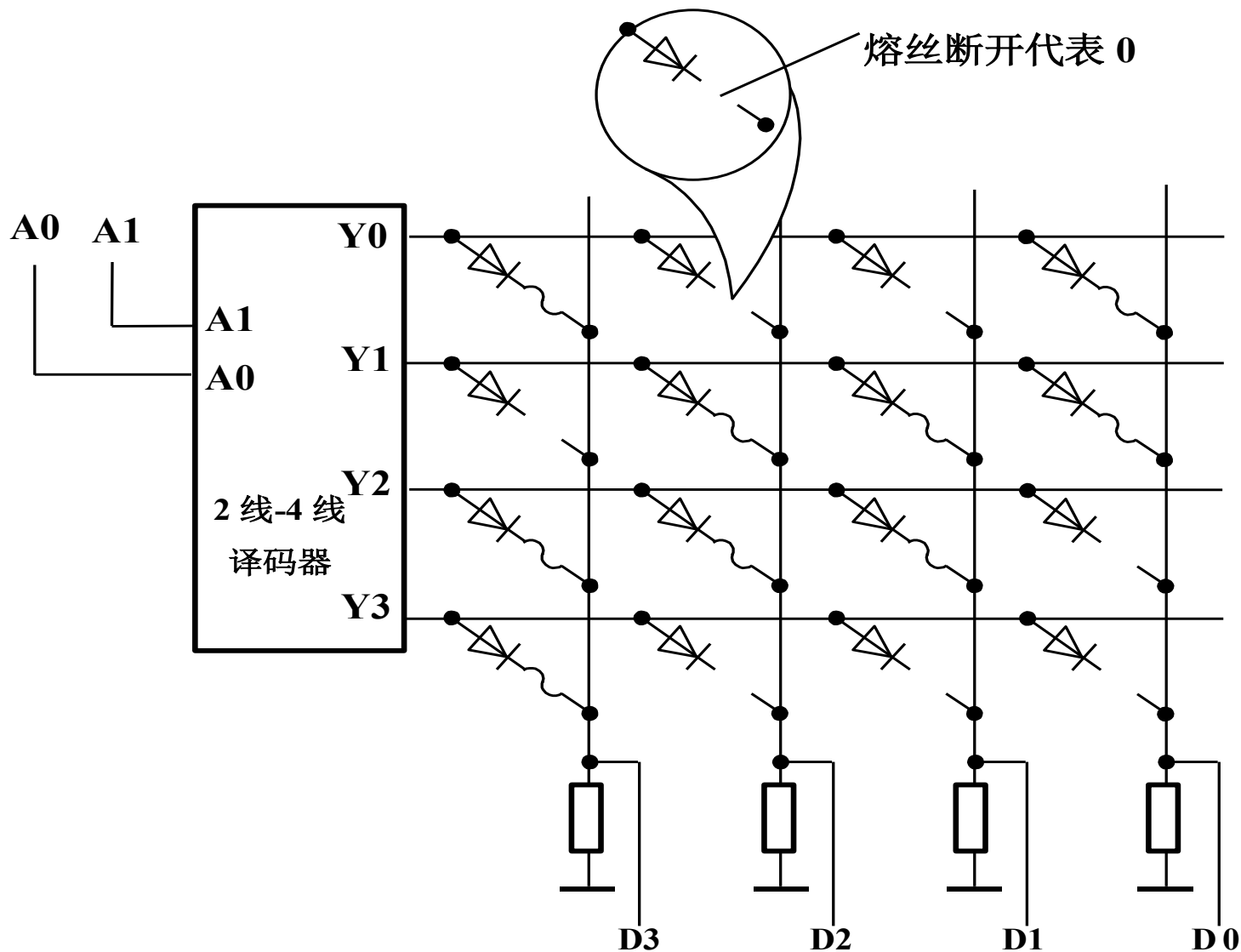
PROM结构示意:



编程前的PROM

熔丝开关 (续)

PROM 结构示意图

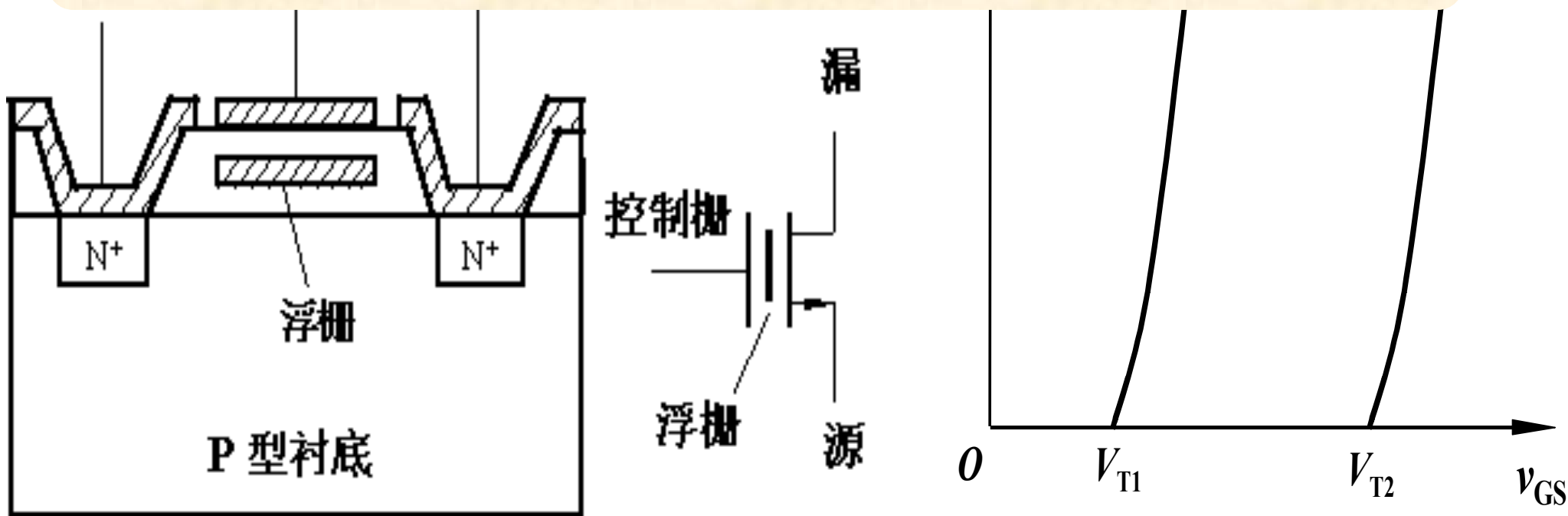


编程后的PROM

当浮栅上没有电荷时，给控制栅加上控制电压，MOS管导通。

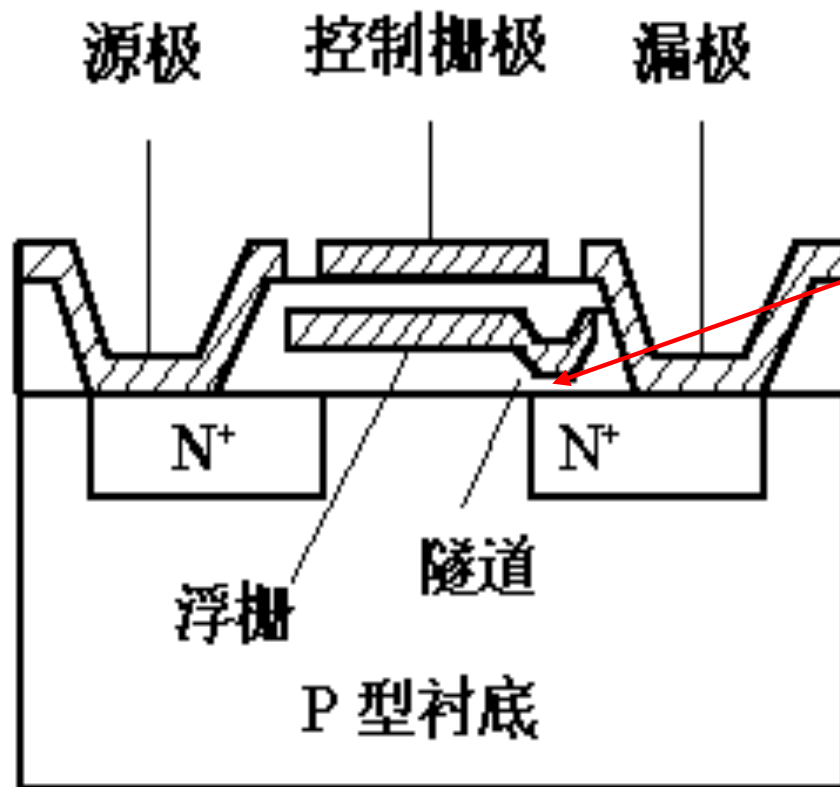
当浮栅上带有负电荷时，则衬底表面感应的是正电荷，这使得MOS管的开启电压变高，如果给控制栅加上同样的控制电压，MOS管仍处于截止状态。

所以，SIMOS管用浮栅是否累积有负电荷来存储二值数据。电子



EPROM的结构示意图

浮栅编程技术（续）

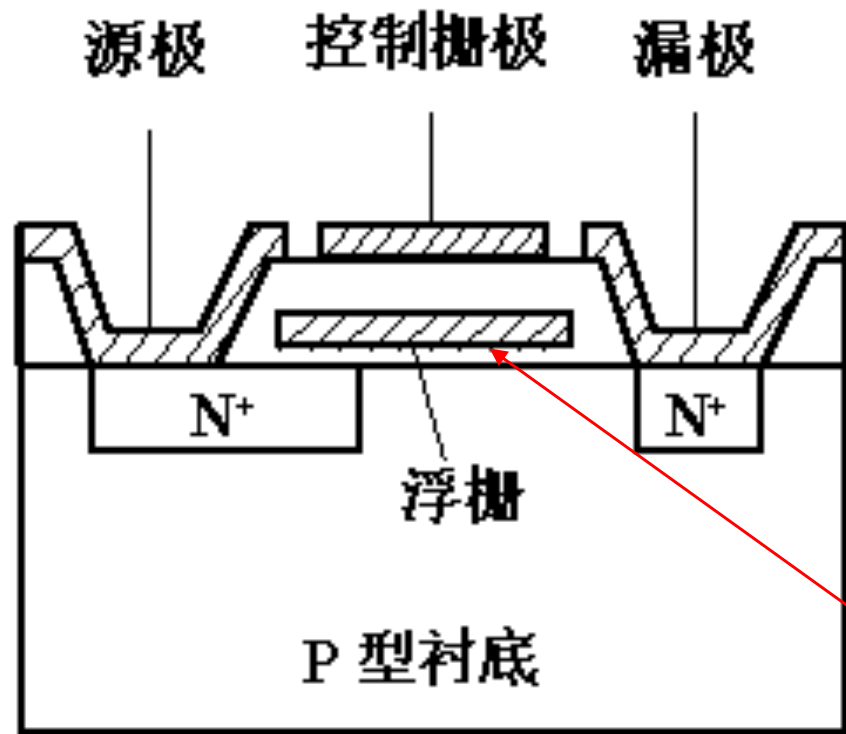


隧道 MOS 管剖面结构示意图

与EPROM管的
区别是：
浮栅延长区与
漏区 N^+ 之间的
交叠处有一个
厚度约为80 (埃)
的薄绝缘层。

E²PROM的结构图

浮栅编程技术（续）



闪速存储器存储单元 MOS 管
剖面结构示意图

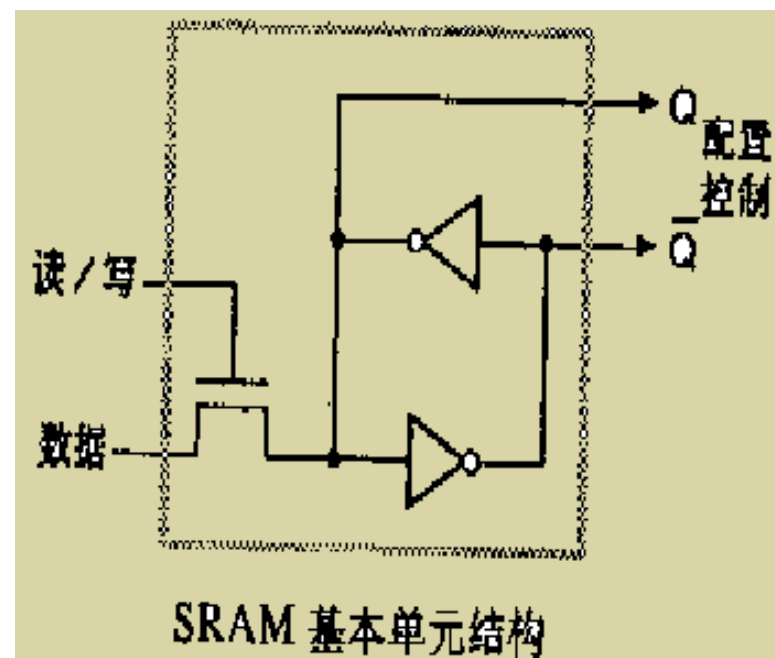
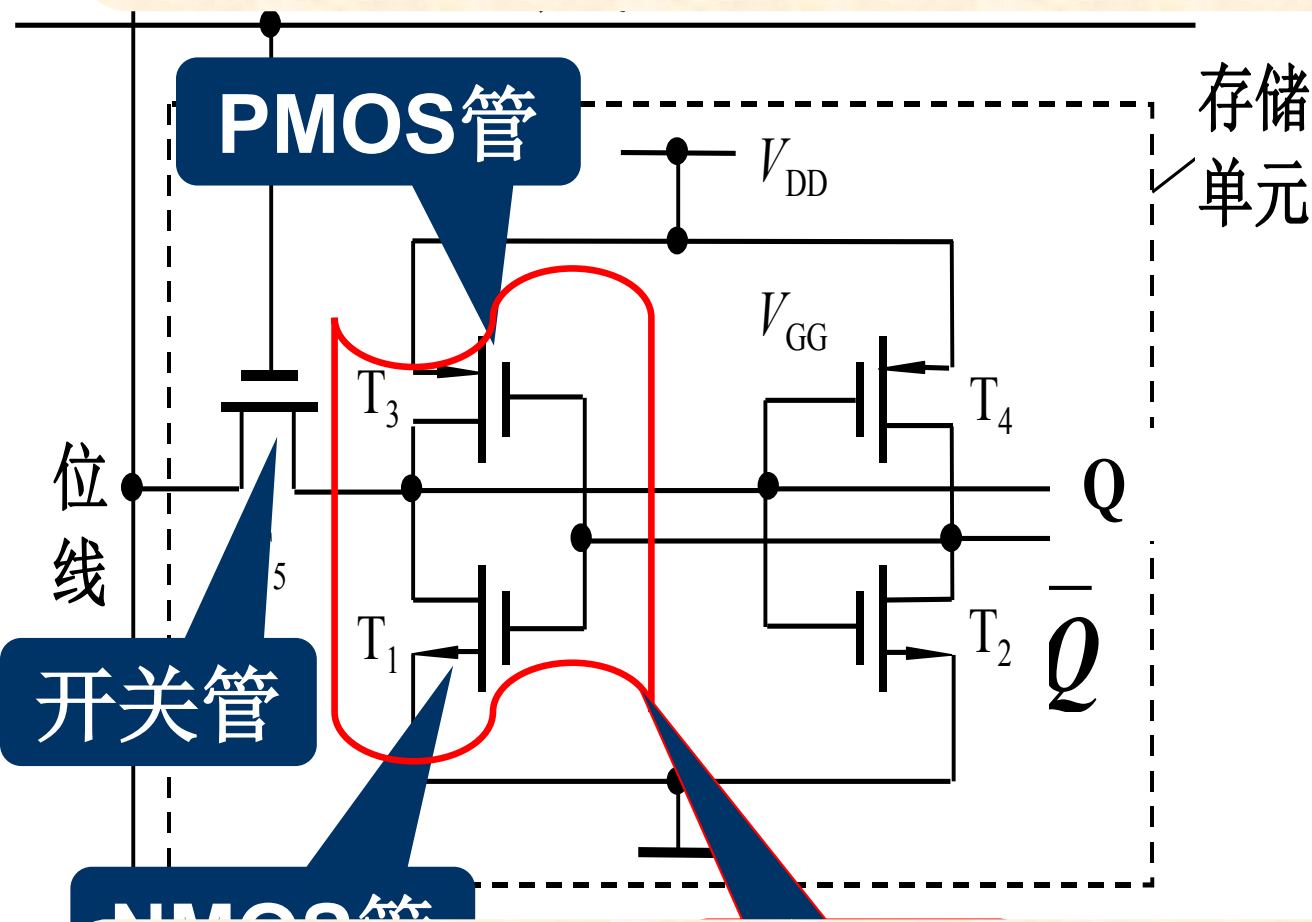
与EPROM的区别是：

1. 闪速存储器存储单元 MOS 管的源极 N+ 区大于漏极 N+ 区，而 SIMOS 管的源极 N+ 区和漏极 N+ 区是对称的；

2. 浮栅到 P 型衬底间的氧化绝缘层比 SIMOS 管的更薄。

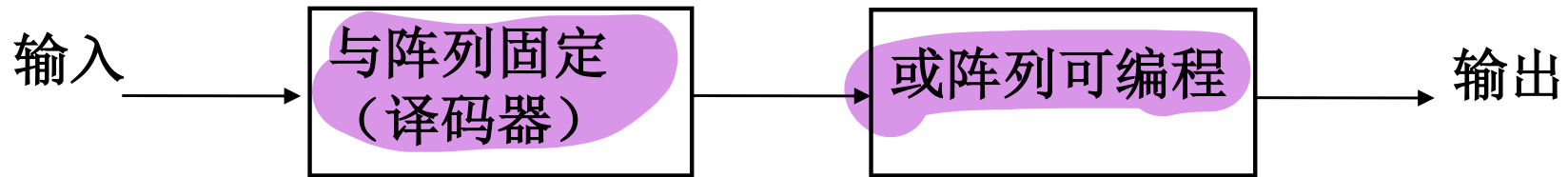
Flash E²PROM 结构图

在写入配置数据时,开关管导通,数据(0或1)写入存储单元。

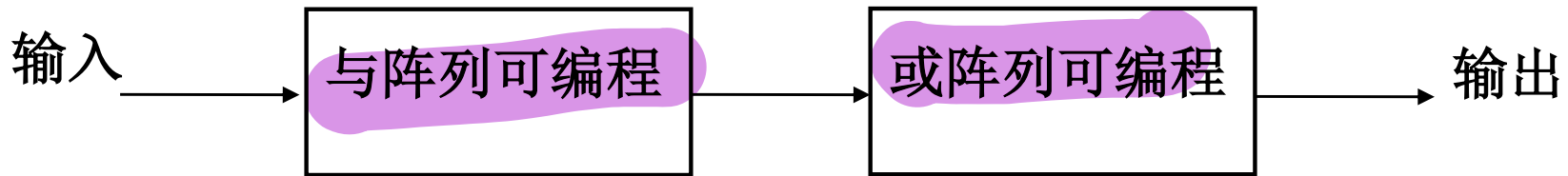


在FPGA工作时,开关管截止,数据从Q端输出。

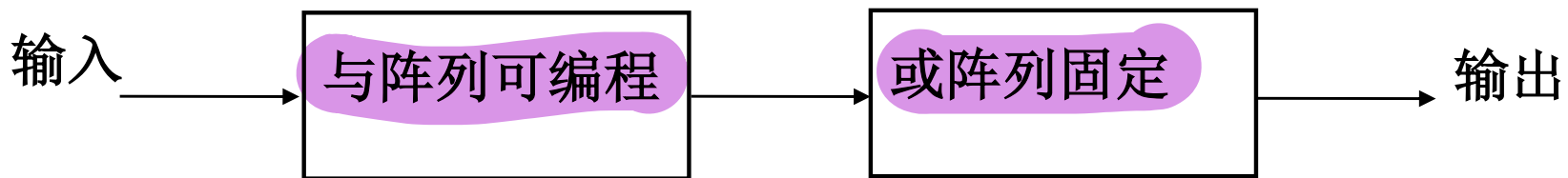
6.2 简单可编程逻辑器件



(a)



(b)



(c)

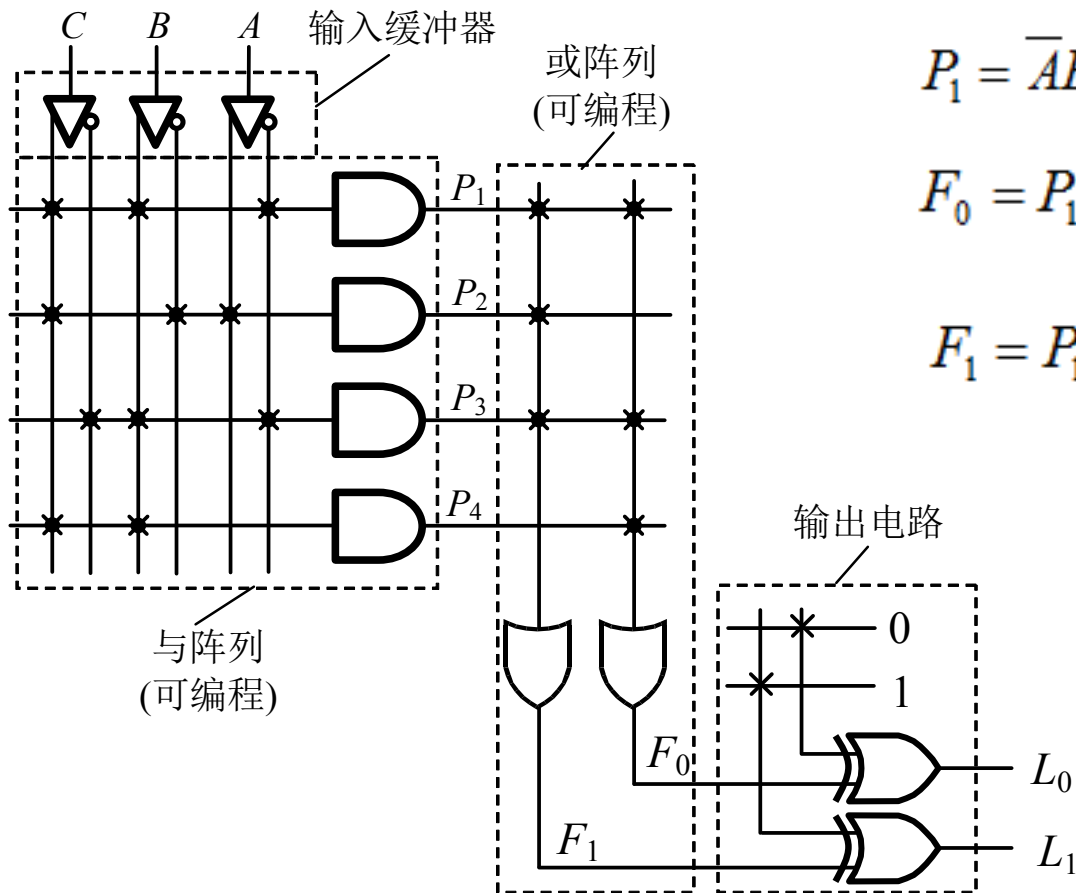
图6.2.1 SPLD的分类

(a) PROM的结构框图

(b) PLA的结构框图

(c) PAL、GAL的结构框图

6.2.1 可编程逻辑阵列PLA



$$P_1 = \overline{A}BC, \quad P_2 = A\overline{B}C, \quad P_3 = \overline{A}\overline{B}C, \quad P_4 = BC$$

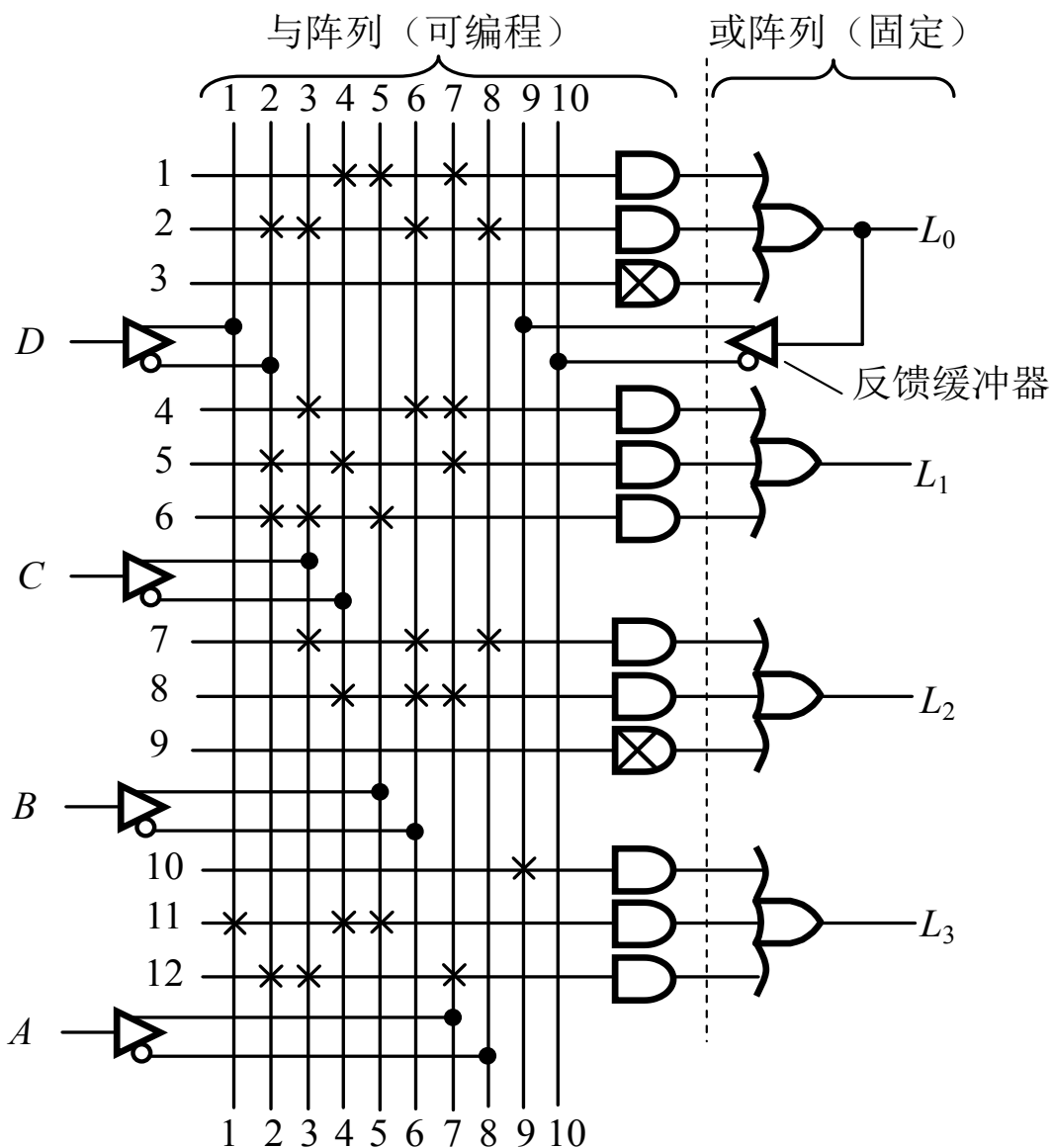
$$F_0 = P_1 + P_3 + P_4 = \overline{A}BC + \overline{A}\overline{B}C + BC$$

$$F_1 = P_1 + P_2 + P_3 = \overline{A}BC + A\overline{B}C + \overline{A}\overline{B}C$$

$$L_0 = F_0 \oplus 0 = F_0 = \overline{A}BC + \overline{A}\overline{B}C + BC$$

$$L_1 = F_1 \oplus 1 = \overline{F_1} = \overline{\overline{A}BC + A\overline{B}C + \overline{A}\overline{B}C}$$

6.2.2 可编程阵列逻辑PAL



$$L_0 = AB\bar{C} + \bar{A}\bar{B}C\bar{D}$$

$$L_1 = \bar{A}\bar{B}C + A\bar{C}\bar{D} + BCD$$

$$L_2 = \bar{A}\bar{B}C + A\bar{B}C$$

$$L_3 = L_0 + B\bar{C}D + AC\bar{D}$$

$$= AB\bar{C} + \bar{A}\bar{B}C\bar{D} + B\bar{C}D + AC\bar{D}$$

•PAL器件的命名

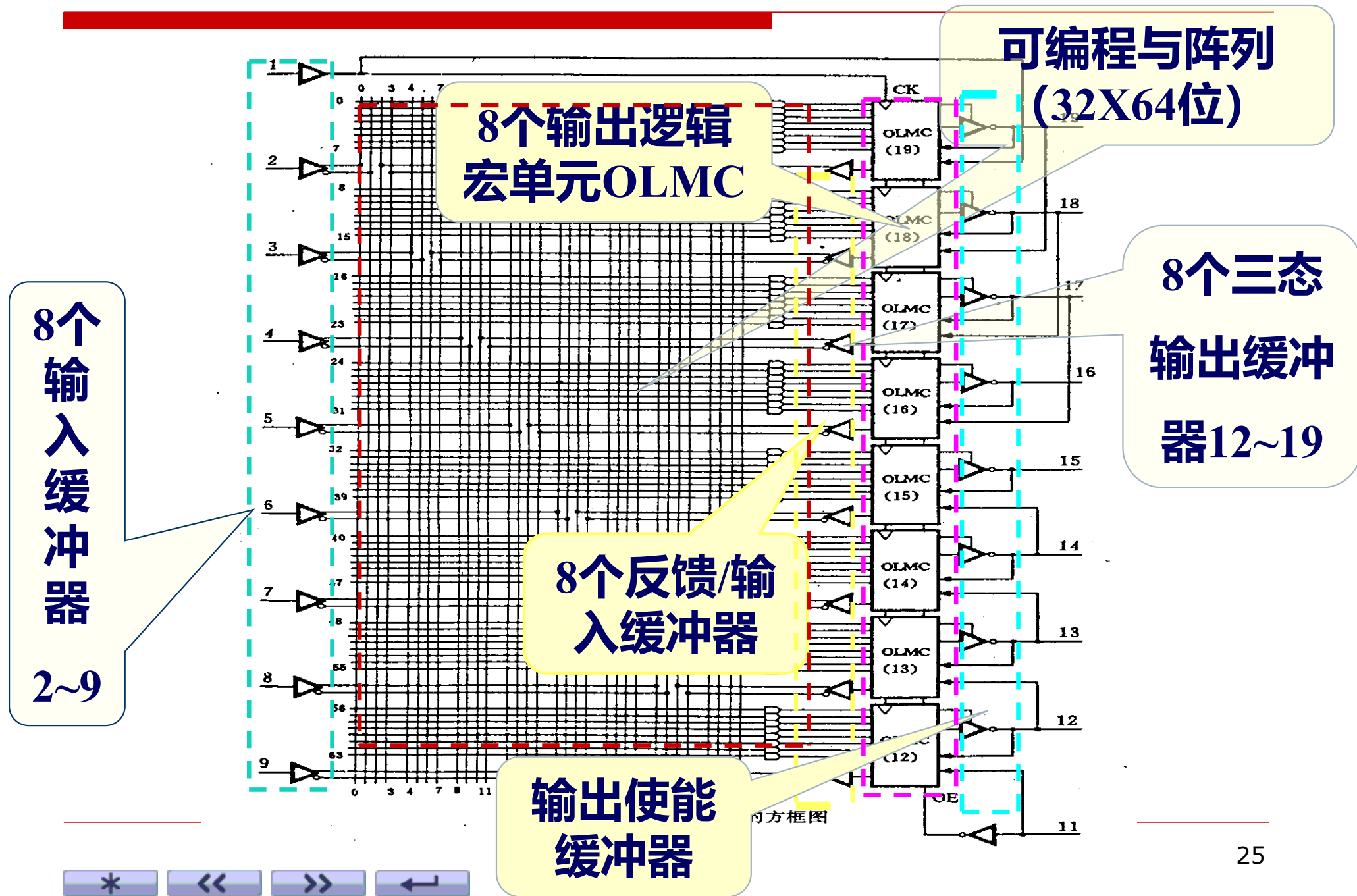
<i>PAL</i>	<i>C</i>	16	<i>R</i>	8	××××
<hr/>					
(1)	(2)	(3)	(4)	(5)	(6)

- (1) 生产厂家对PAL器件命名，前面一般还有厂家的标志；
- (2) 代表制造工艺：空白代表TTL，C代表CMOS；
- (3) 代表PAL器件的最大阵列输入数；
- (4) 代表输出电路类型。
- (5) 代表最大的组合输出端数目或最大的寄存器数目。
- (6) 表示器件功耗级别、速度等级，封装形式等信息。

表 1 PAL 器件的输出电路类型

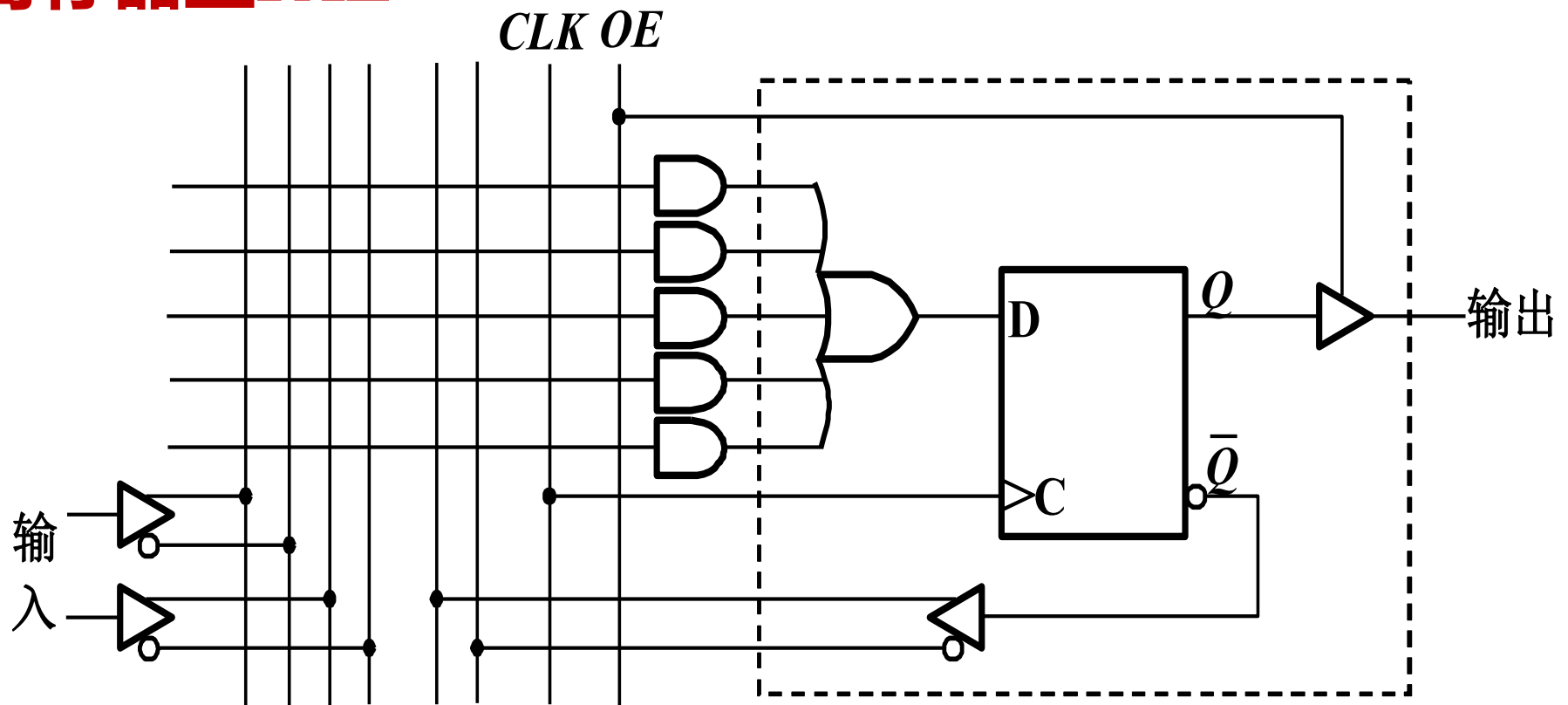
代号	意 义	例 子
H	高电平输出 (Active-High Output)	PAL10H8
L	低电平输出 (Active-Low Output)	PAL16L8
P	输出极性可编程 (Programmable Output Polarity)	PAL16P8
C	互补式输出 (Complementary Output)	PAL16C1
X	带异或门输出 (Exclusive-OR Gate)	PAL20×10
R	带寄存器输出 (Registered Output)	PAL16R8
RP	输出极性可编程且带寄存器的输出 (Registered with Programmable Proarity)	PAL16RP8
RA	带有非同步的寄存器输出 (Registered Asynchronous)	PAL16RA8

6.2.3 GAL的结构——GAL16V8的结构为例



6.2.3 GAL中的输出逻辑宏单元

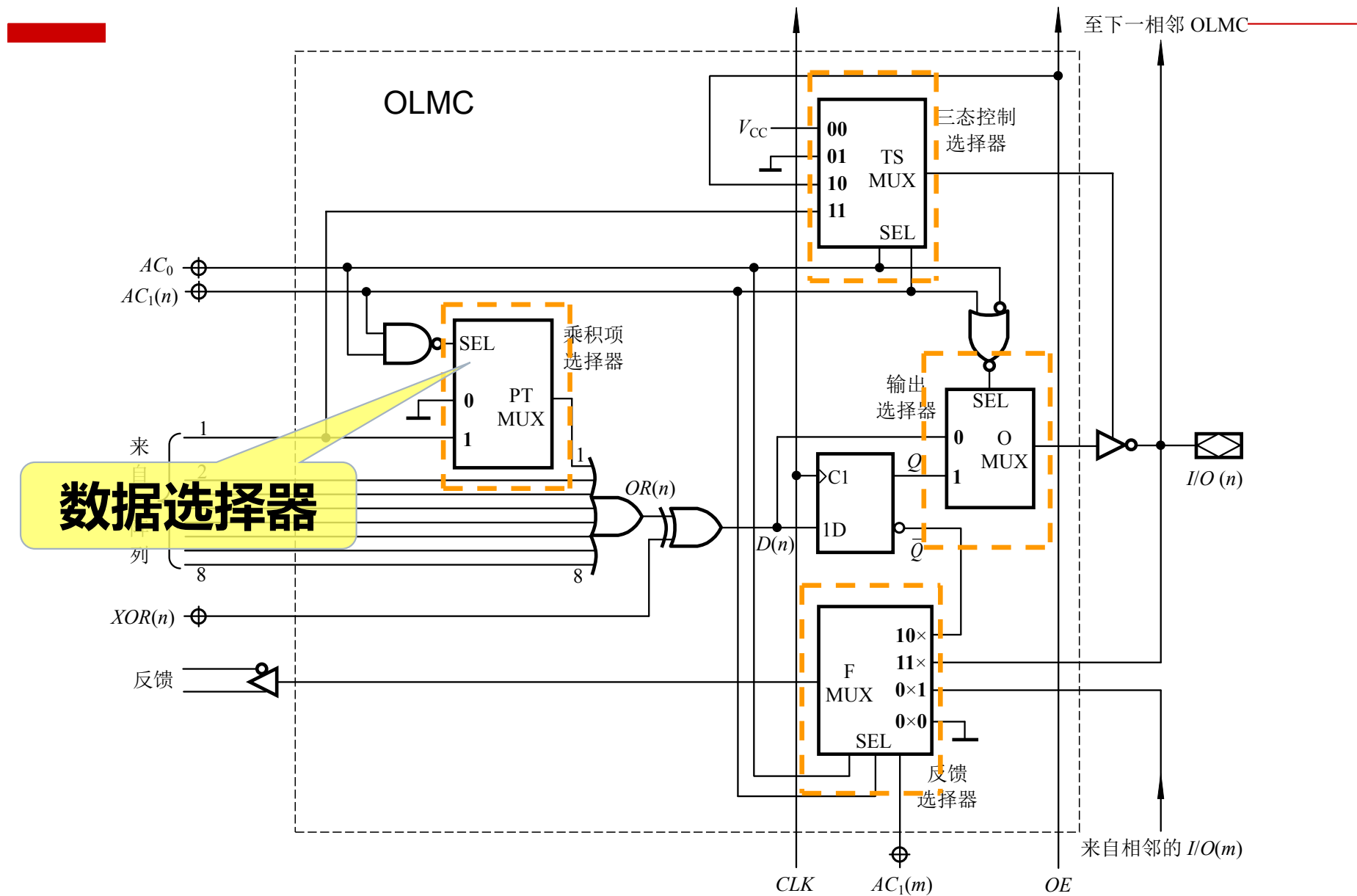
1. 寄存器型PAL



寄存器型PAL如图所示，在组合PLD基础上增加了D触发器，并反馈回到输入与阵列，满足时序电路设计要求。

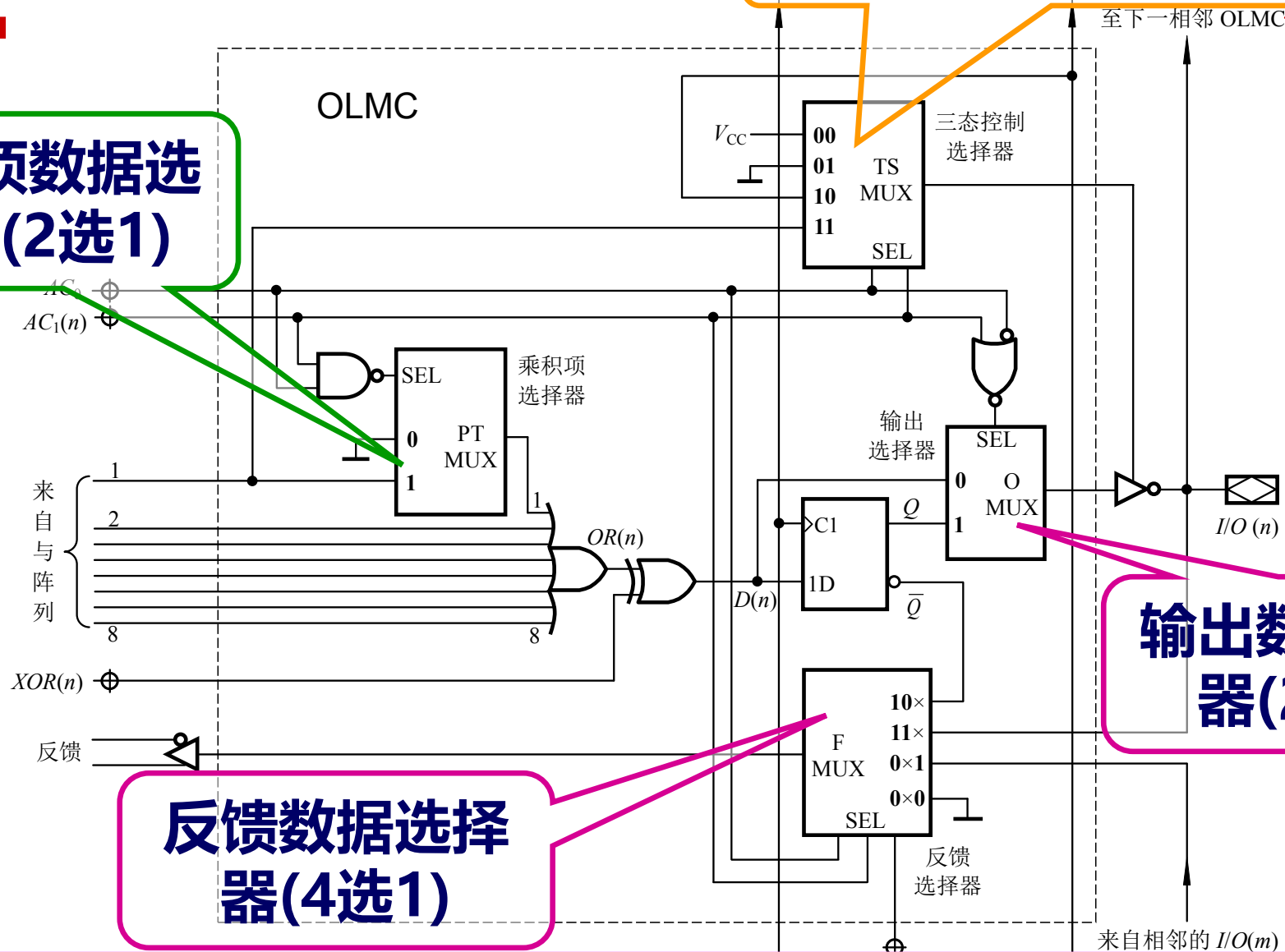
2. GAL中的输出宏单元

GAL的电路结构与PAL类似，由可编程的与逻辑阵列、固定的或逻辑阵列和输出电路组成，但GAL的输出端增设了可编程的输出逻辑宏单元（OLMC）。通过编程可将OLMC设置为不同的工作状态，可实现PAL的所有输出结构，产生组合、时序逻辑电路输出。



三态数据选择器(4选1)

乘积项数据选择器(2选1)

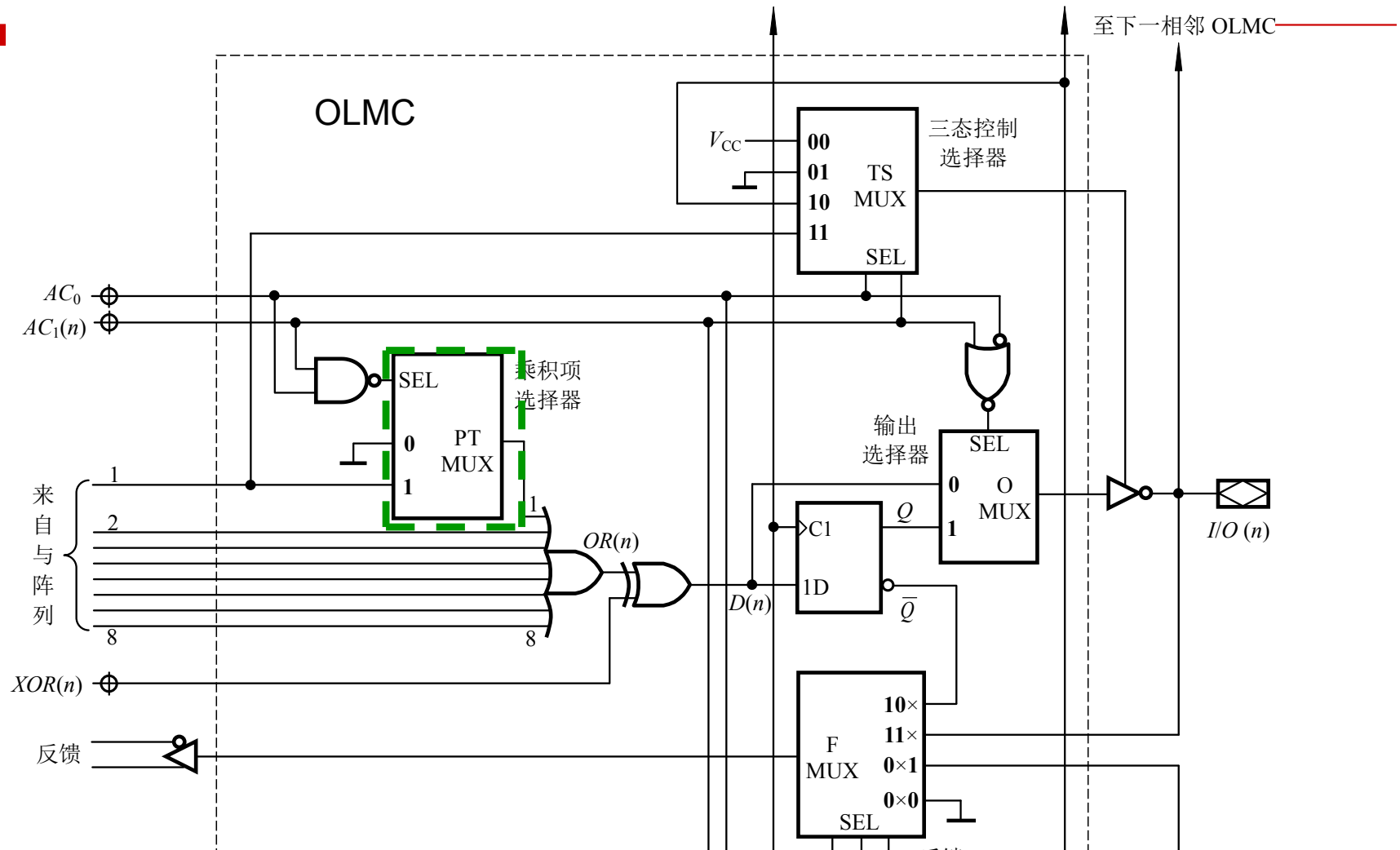


输出数据选择器(2选1)

反馈数据选择器(4选1)

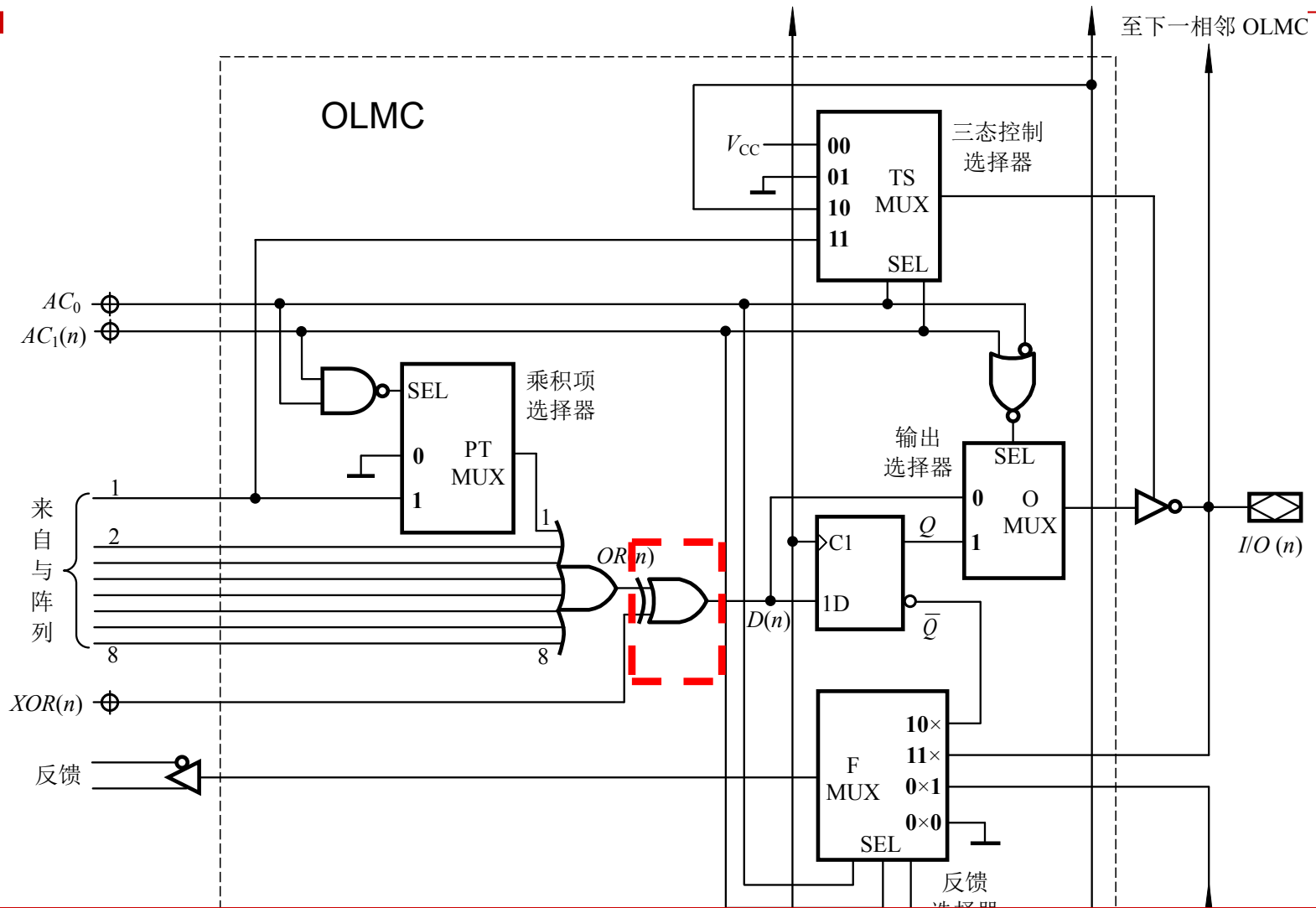
4个数据选择器：用不同的控制字实现不同的输出电路结构形式

(1)输入电路—由乘积项数据选择器(2选1)PTMUX控制



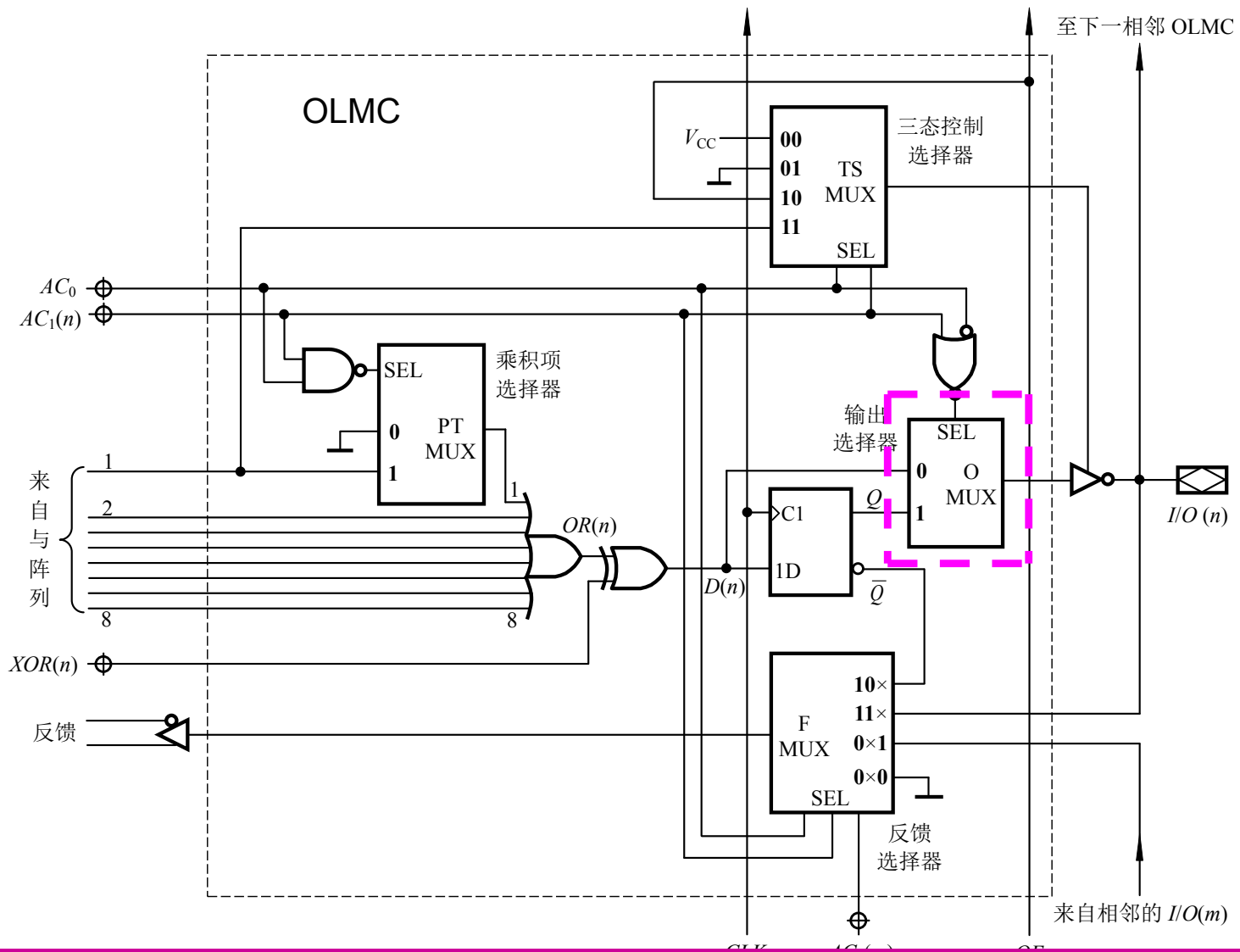
乘积项数据选择器：根据 AC_0 和 $AC_1(n)$ 决定与逻辑阵列的第一乘积项是否作为或门的一个输入端。只有在 G_2 的输出为1时，第一乘积项是或门的一个输入端。

(2)原变量/非变量输出电路—由异或门控制



异或门输出为或门输出 $OR(n)$ 与 $XOR(n)$ 进行异或运算。
 $XOR(n)=0$ ，则 $D(n)=OR(n)$ ，若 $XOR(n)=1$ ，则 $D(n)=\overline{OR(n)}$ 。

(3) 输出电路——由数据选择器(2选1) OMUX控制



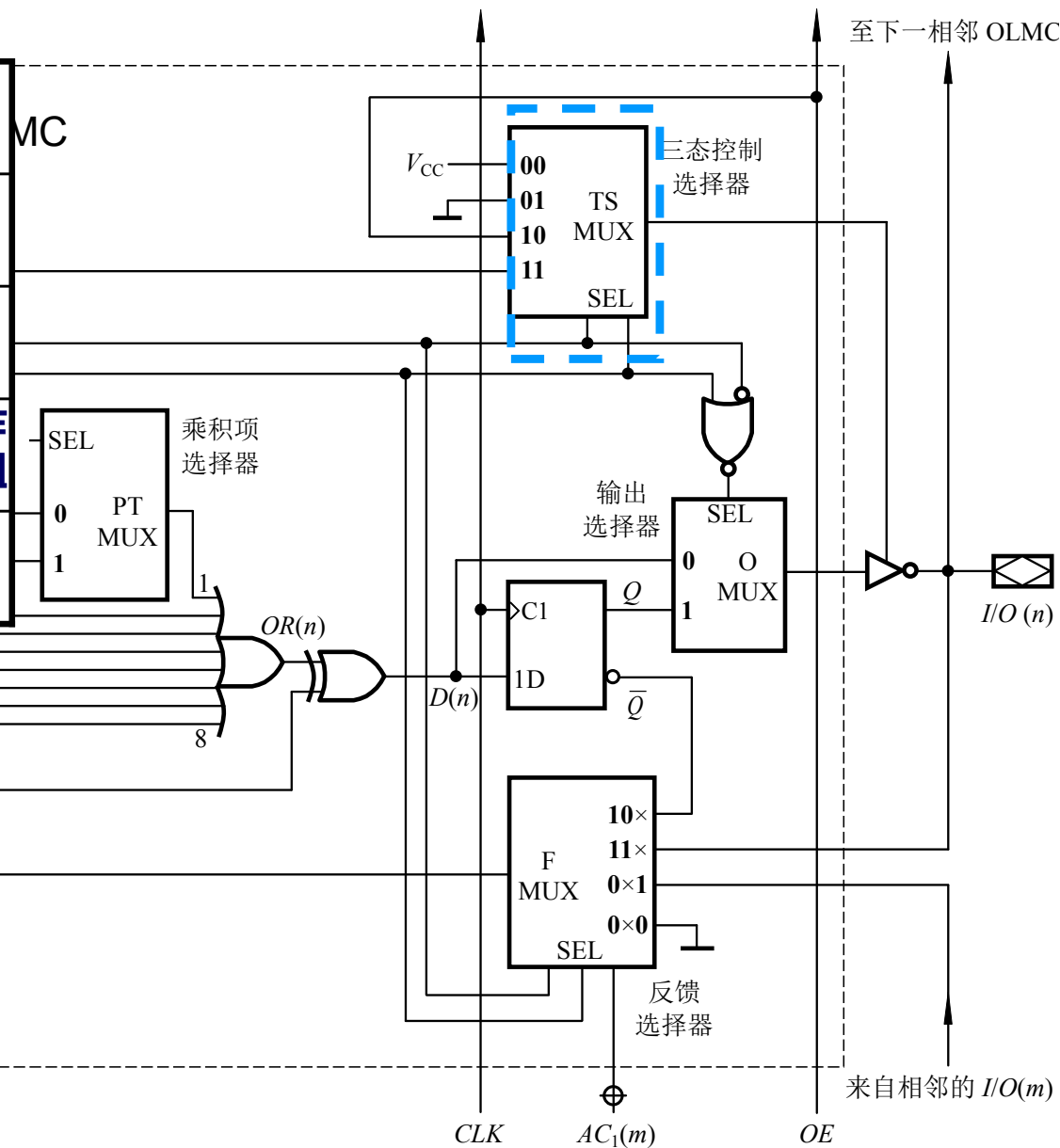
OMUX: 根据 AC_0 和 $AC_1(n)$ 决定OLMC是组合输出还是寄存器输出模式

由三态数据选择器(4选1)控制输出选择器的选通端

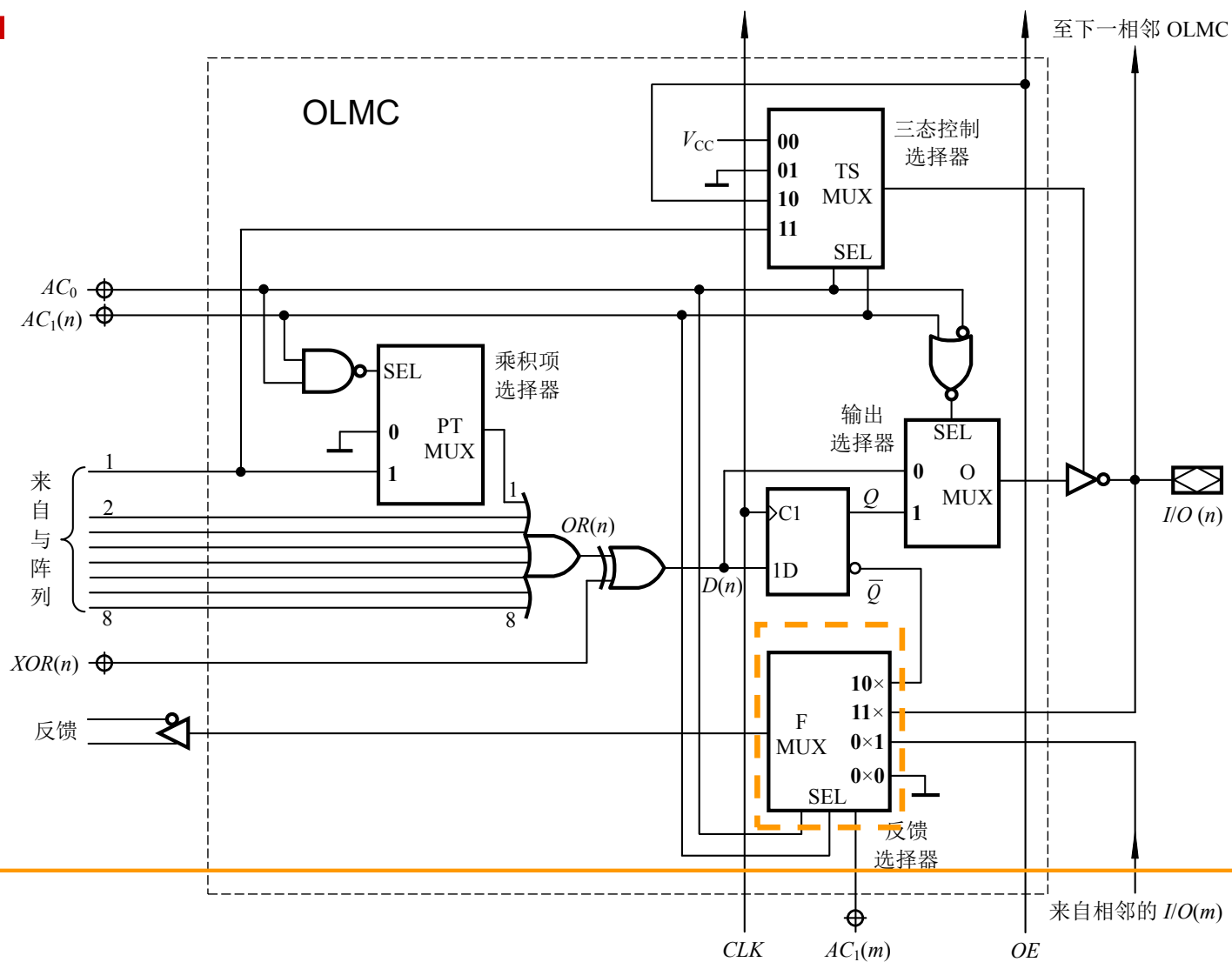
SEL

AC0 AC1(n)	TX (输出)	三态缓冲器的工作状态
0 0	V_{CC}	工作
0 1	地电平	高阻
1 0	OE	OE=1, 工作 OE=0, 高阻
1 1	第一乘积项	1, 工作 0, 高阻

三态数据选择器受AC0和AC1(n)的控制，用于选择输出三态缓冲器的选通信号。可分别选择 V_{CC} 、地、OE和第一乘积项。



反馈数据选择器(4选1)——FMUX

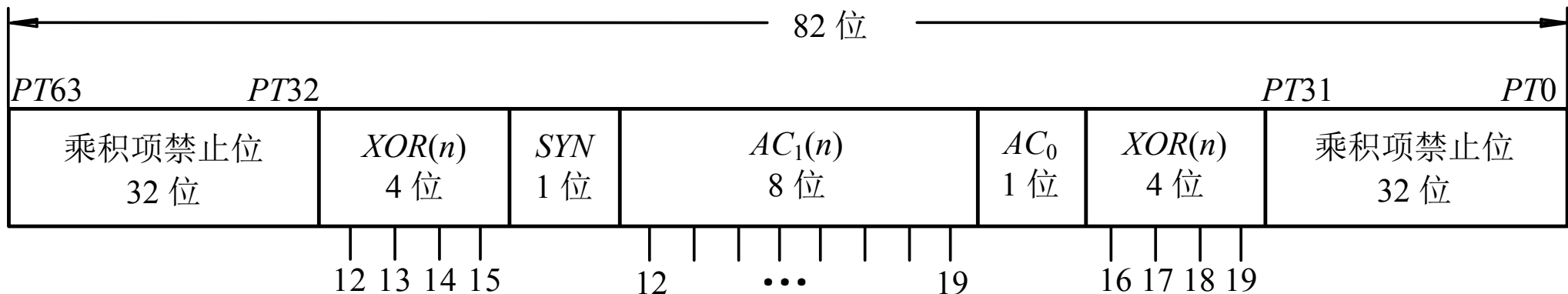


FMUX:

根据AC₀和AC₁(n)的不同编码，使反向传输的电信号也对应不同。

6.6.3 GAL的结构控制字

GAL16V8的结构控制字共有82位，它们的定义如图。
每个OLMC有2个编程单元AC1(n)和XOR(n)，一个全局编程单元AC0，同步控制单元SYN。



6.3 复杂可编程逻辑器件(CPLD)

6.3.1 CPLD的基本结构

6.3.2 逻辑块

6.3.3 I/O块

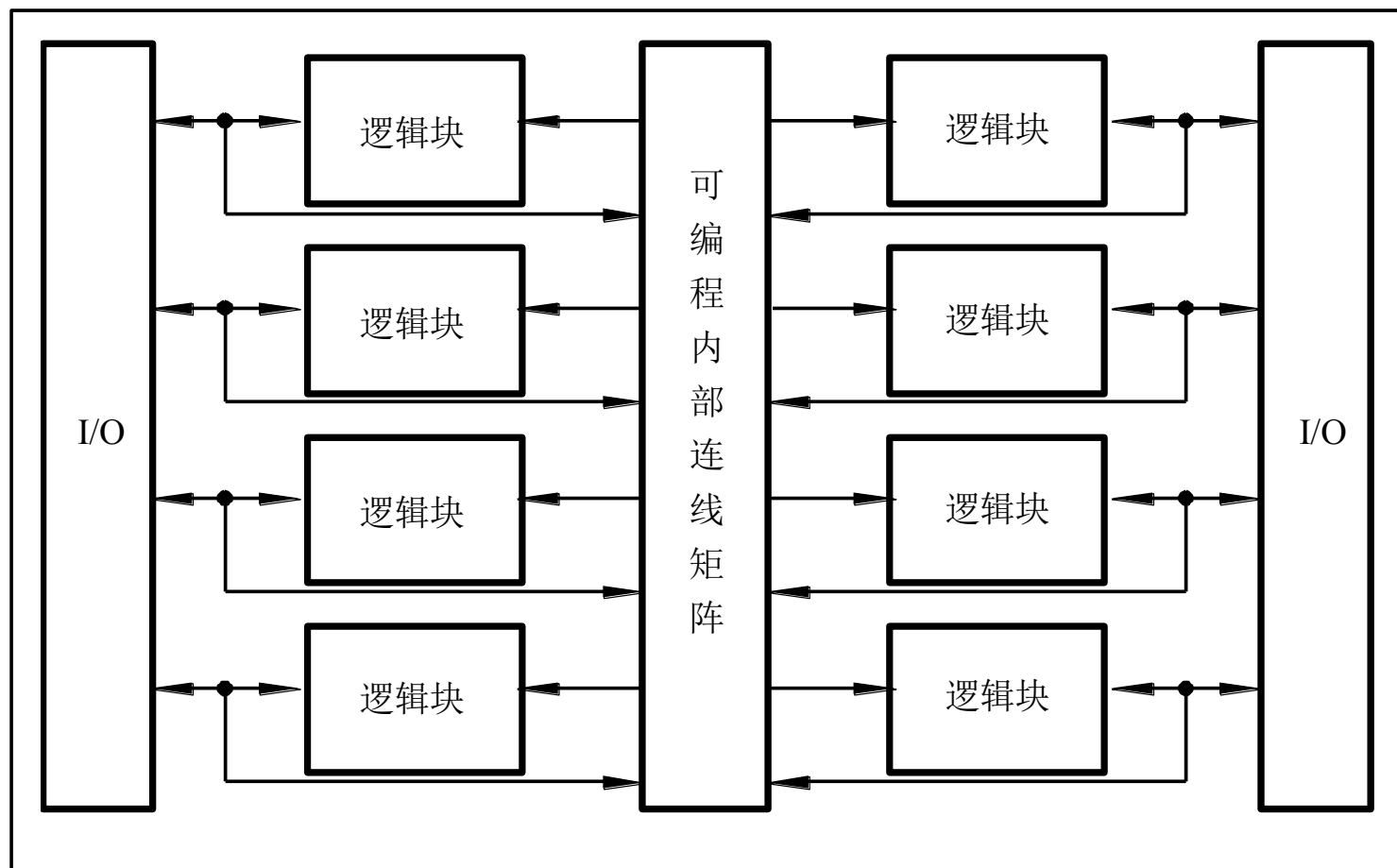
6.3.4 可编程内部互连线资源

6.3 复杂可编程逻辑器件(CPLD)

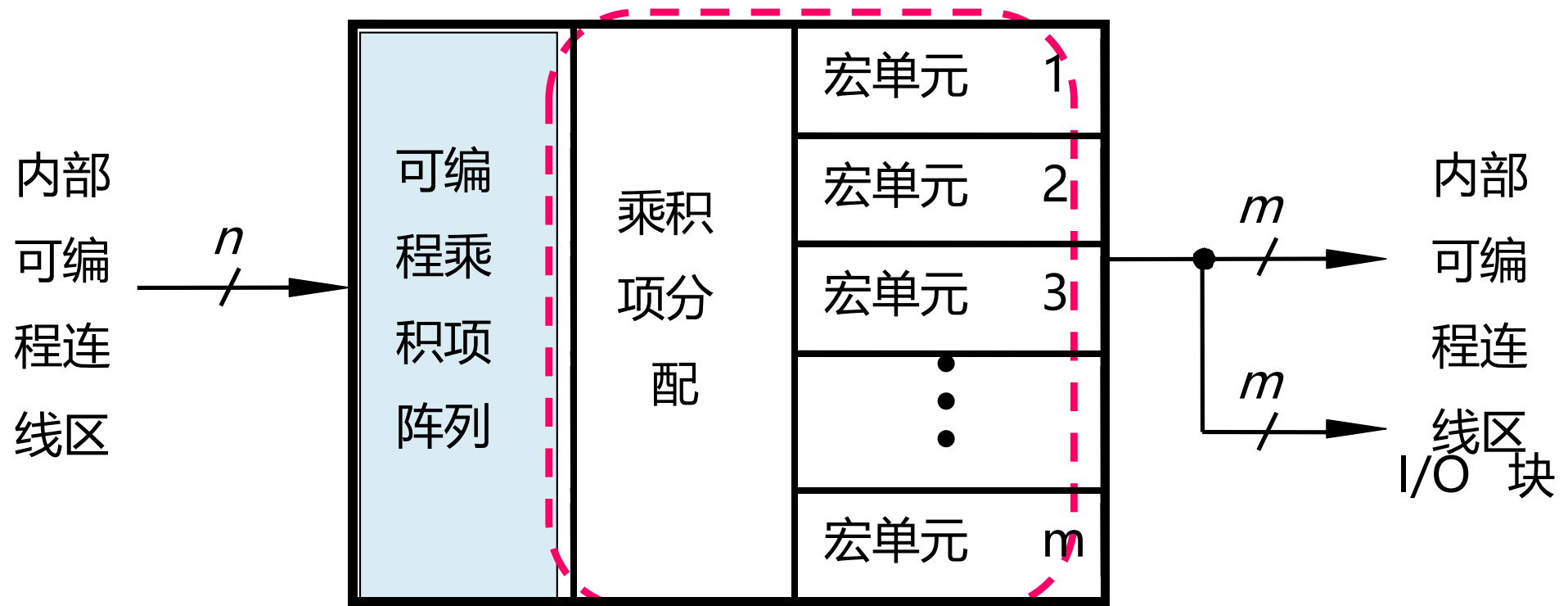
- 与PAL、GAL相比，CPLD的集成度更高，有更多的输入端、乘积项和更多的宏单元；
- CPLD器件内部含有多个**逻辑块**，每个**逻辑块**都相当于一个PAL(或GAL)器件；
- 每个块之间可以使用可编程内部连线(或者称为**可编程的开关矩阵**)实现相互连接。

6.3.1 CPLD的基本结构

更多成积项、更多宏单元、更多的输入信号。



6.3.2 CPLD逻辑块的结构

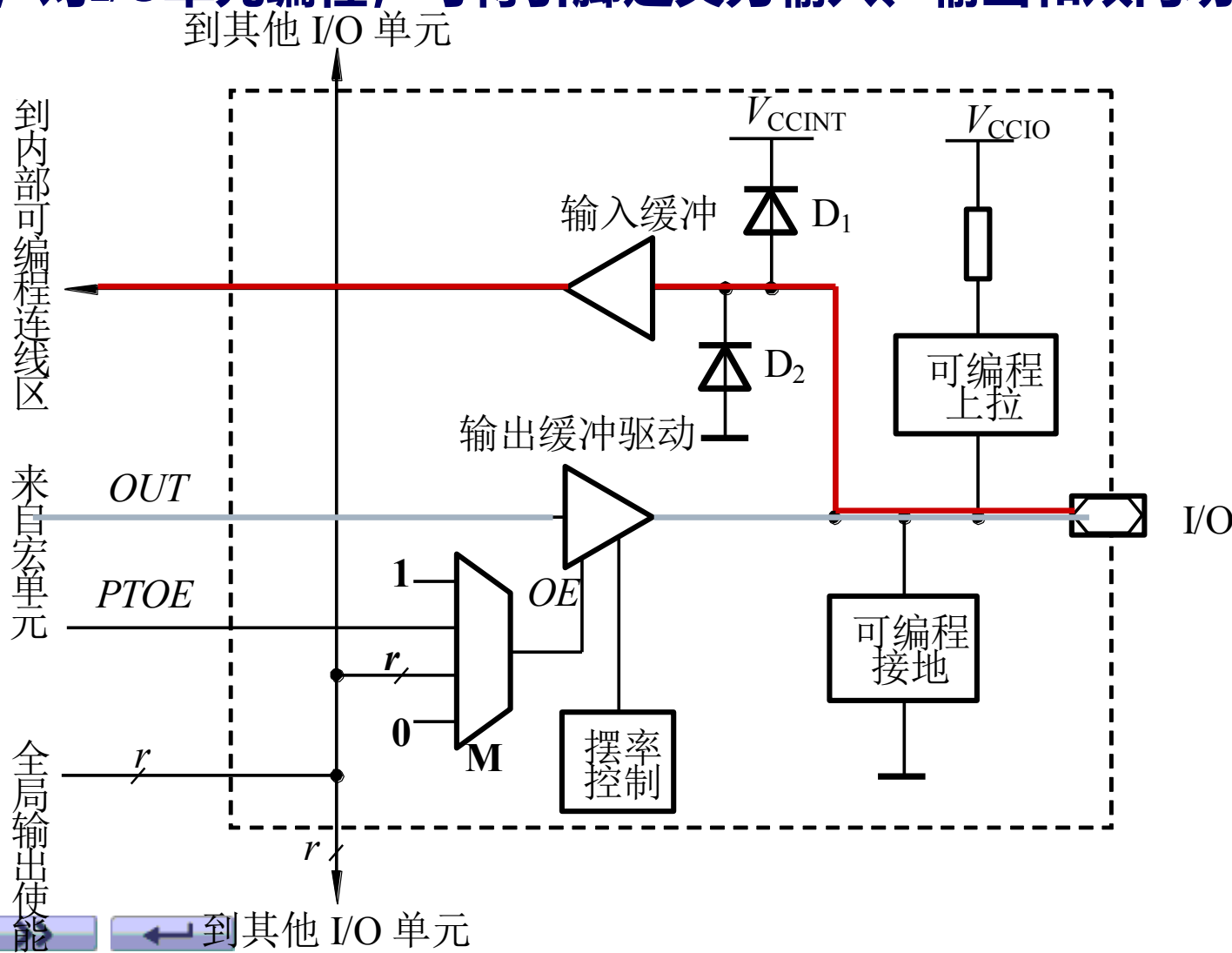


Xilinx XC9500: 90 个 36 变量的乘积项, 宏单元 18 个

Altera MAX7000: 80 个 36 变量的乘积项, 宏单元 16 个

6.3.3 I/O块

I/O块是CPLD外部封装引脚和内部逻辑间的接口。每个I/O单元对应一个封装引脚，对I/O单元编程，可将引脚定义为输入、输出和双向功能。



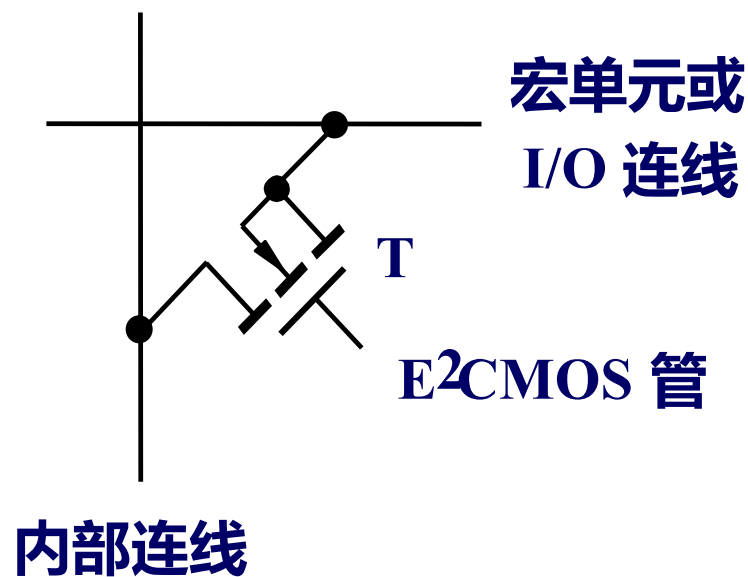
数据选择器提供OE信号。
OE=1, I/O引脚为输出

6.3.4 可编程内部互连线资源

可编程内部连线的作用是实现逻辑块与逻辑块之间、逻辑块与I/O块之间以及全局信号到逻辑块和I/O块之间的连接。

在CPLD内部实现可编程互连线的方法有两种：一是基于存储单元控制的MOS管来实现可编程连接，另一种是基于多路数据选择器实现互连。

- 当E²CMOS管被编程为导通时，纵线和横线连通；未被编程为截止时，两线则不通。



可编程连接原理图

6.3.4 可编程内部互连线资源

- 基于多路数据选择器实现互连:

由图可知, 选择器由输入信号 S 控制

- 当 $S=0$ 时, M_1 选择 X_1 输出, M_2 选择 X_2 输出, 即 $Y_1=X_1$, $Y_2=X_2$;

- 当 $S=1$ 时, M_1 选择 X_2 输出, M_2 选择 X_1 输出, 即 $Y_1=X_2$, $Y_2=X_1$ 。

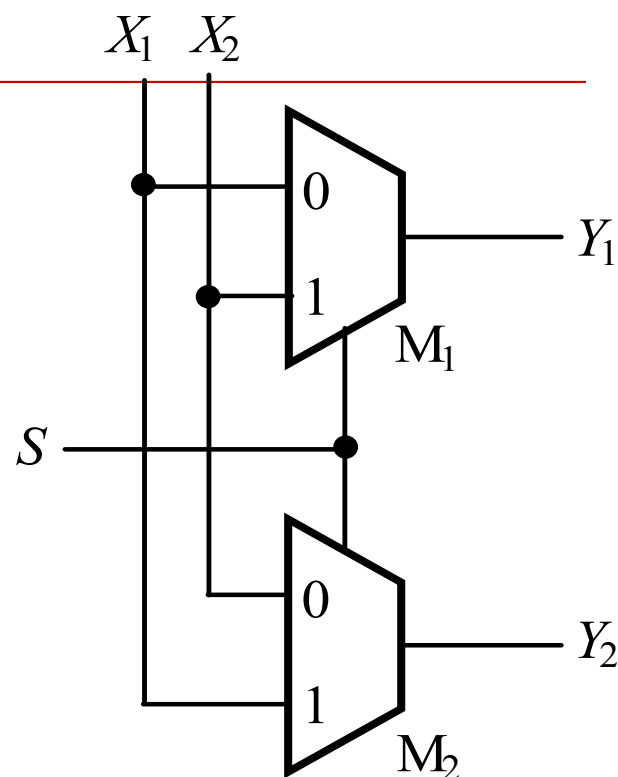


图 9.3.4 2×2 互连线示意图

将该电路扩展成为有 n 个输入、 k 个输出的电路, 使任意一个输入能够与任意一个输出相连, 通常称之为 $n \times k$ 的纵横开关。

S	Y_1	Y_2
0	X_1	X_2
1	X_2	X_1

6.4 现场可编程门阵列(FPGA)

6.4.1 FPGA实现逻辑功能的基本原理

6.4.2 FPGA的一般结构

6.4.3 基于查找表（LUT）的逻辑块

6.4.4 可编程布线资源

6.4.5 I/O块

6.4 现场可编程门阵列(FPGA)

- **CPLD**用可编程“**与-或**”阵列（乘积项技术）实现逻辑函数。**编程**基于**E²PROM**或快闪存储器。
- **FPGA**是用查找表(**Look-Up Table, LUT**)实现逻辑函数。复杂函数使用众多的**LUT**和触发器实现。

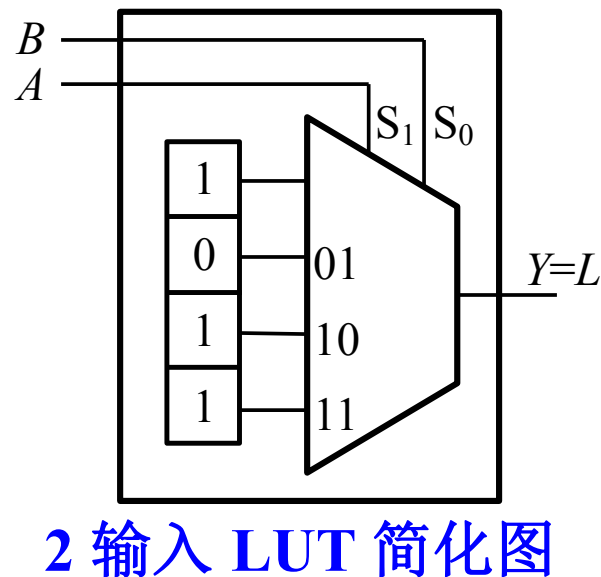
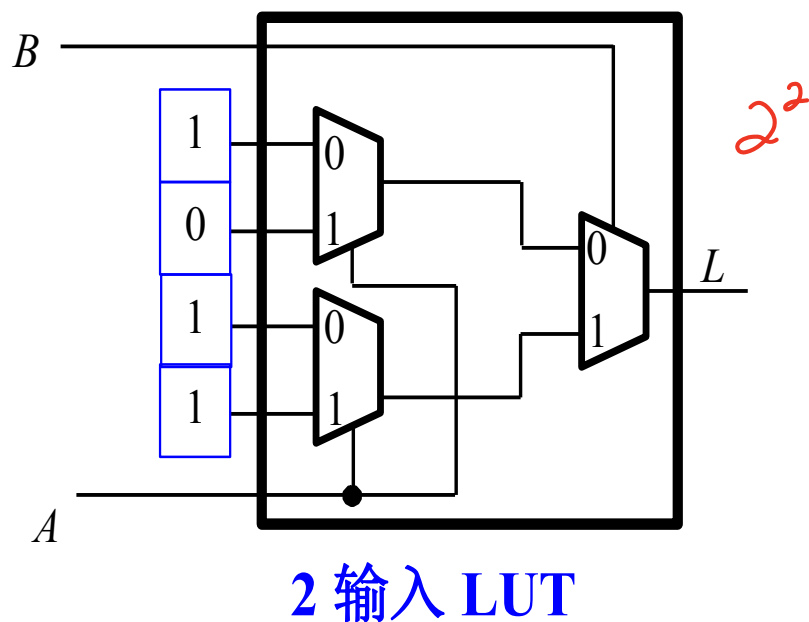
首先自动计算逻辑电路所有可能的结果，并把结果事先写入**RAM**，这样，每输入一个信号进行逻辑运算就等于输入一个**地址**进行**查表**，找出地址对应的内容，然后输出即可。

- **编程**基于**SRAM**。

一般将采用**乘积项技术**的称**CPLD**。采用**LUT**技术的称为**FPGA**。

6.4.1 FPGA实现逻辑功能的基本原理

LUT是FPGA实现逻辑函数的基本单元。2输入LUT可由2选1数据选择器构成，它可以实现任意2变量组合逻辑函数。



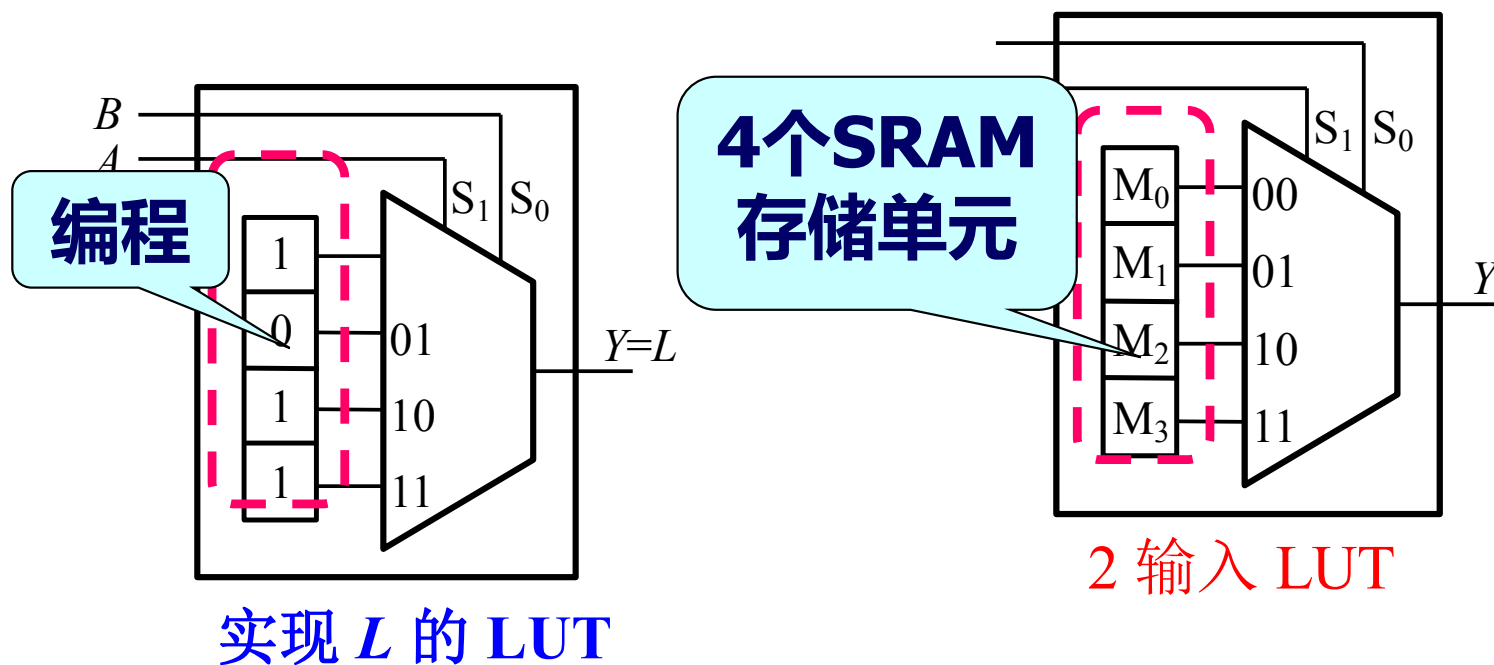
**LUT 由数据选择器和SRAM存储单元构成。数据选择器的选
择端就是LUT的输入，数据选择器的输出端就是LUT的输出。**

6.4.1 FPGA实现逻辑功能的基本原理

2输入LUT可实现任意2变量组合逻辑函数。

某函数 L 的真值表

A	B	L
0	0	1
0	1	0
1	0	1
1	1	1



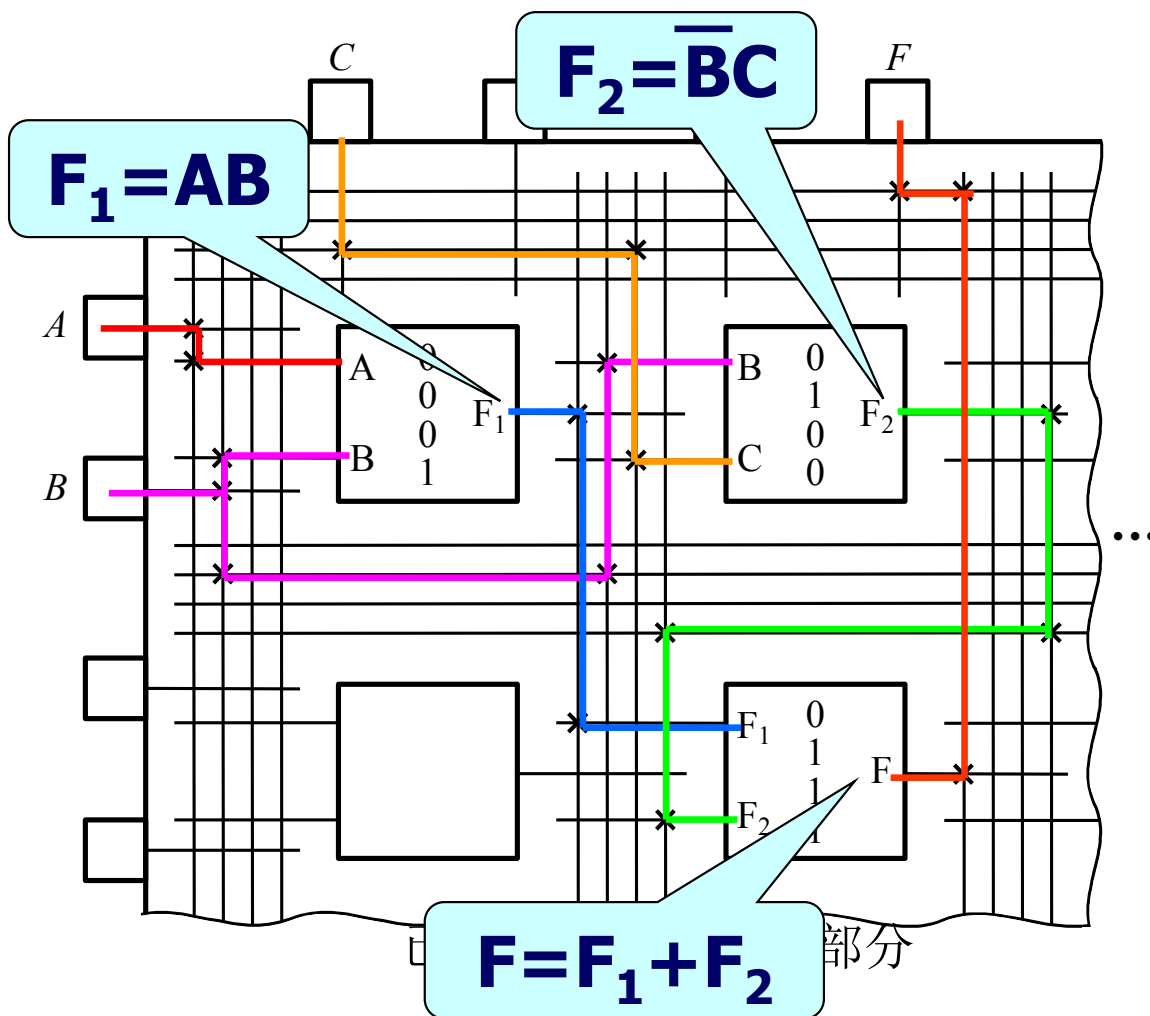
目前FPGA中的LUT大多是4~5个输入，1个输出。当变量数超过一个LUT的输入数时，需要将多个LUT扩展使用。

LUT扩展--用2输入LUT实现函数 $F = AB + \bar{B}C = F_1 + F_2$

函数 F 的真值表

$A B$	F_1	$B C$	F_2
0 0	0	0 0	0
0 1	0	0 1	1
1 0	0	1 0	0
1 1	1	1 1	0

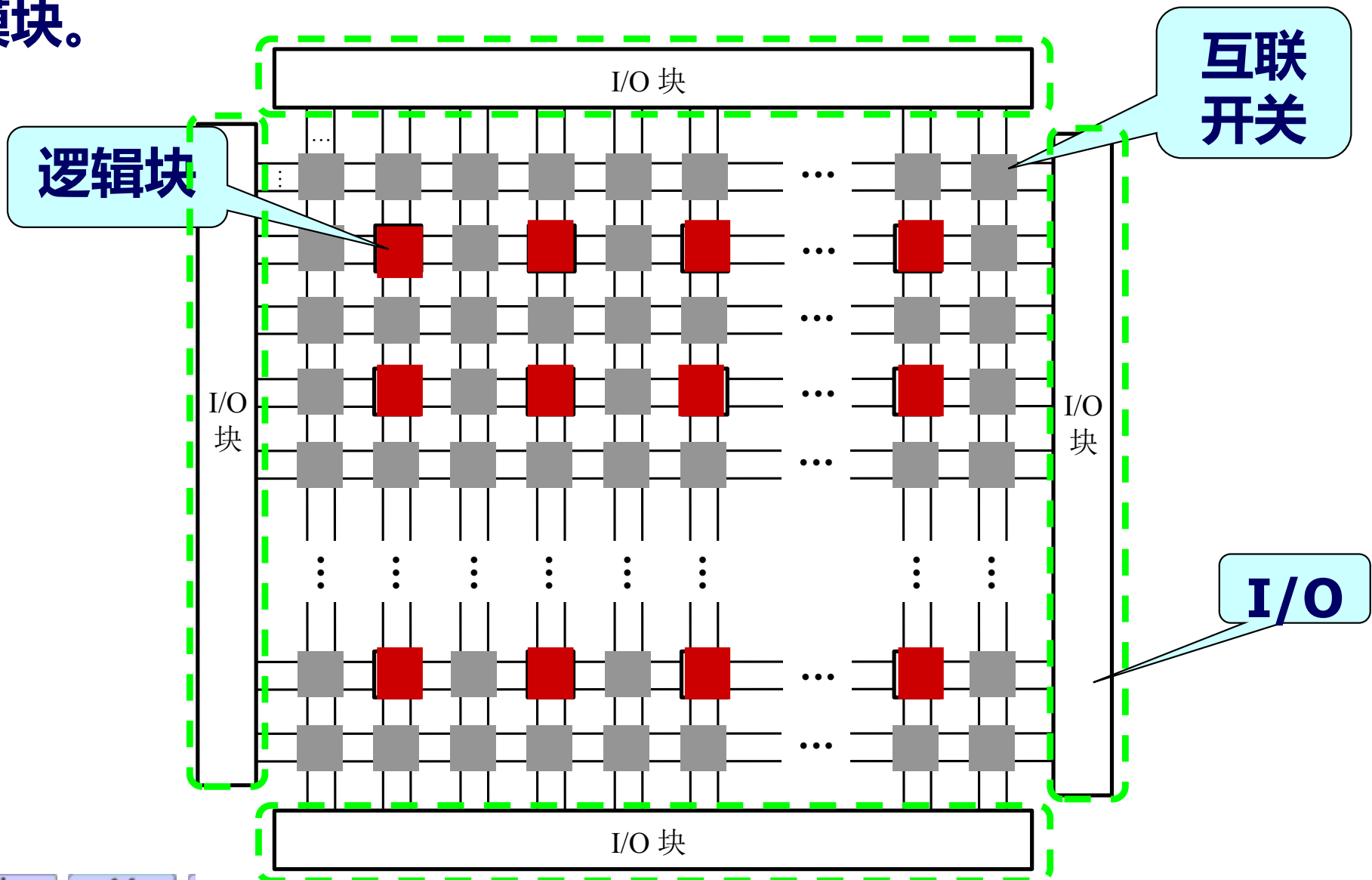
$F_1 F_2$	F
0 0	0
0 1	1
1 0	1
1 1	1



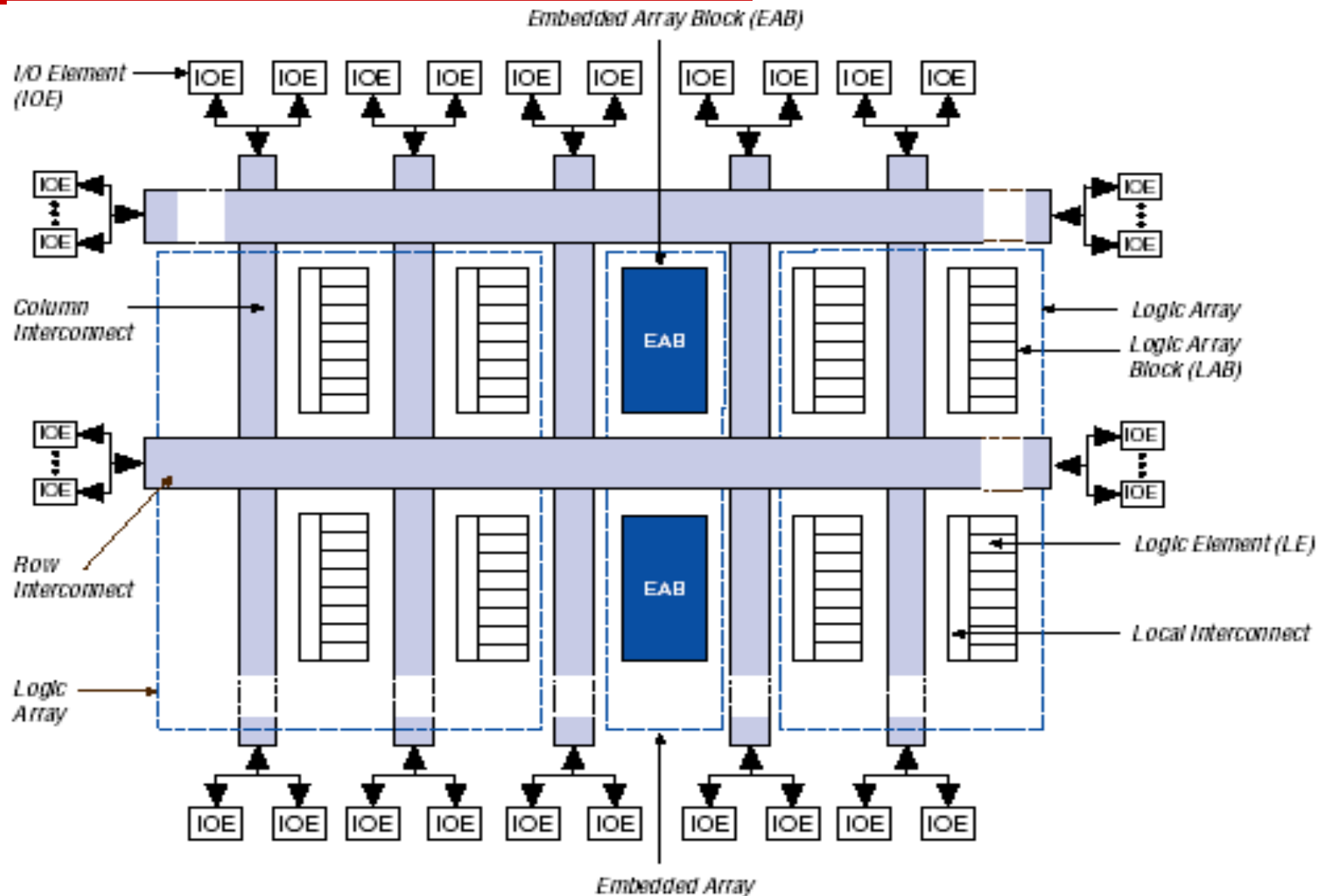
在LUT的基础上增加触发器便可实现时序电路。

6.4.2 FPGA的一般结构

FPGA包括：可编程逻辑块、可编程互联开关、可编程I/O模块。



FLEX10K系列芯片的总结构 (Altera FPGA)



6.4.3 基于查找表 (LUT) 的逻辑块

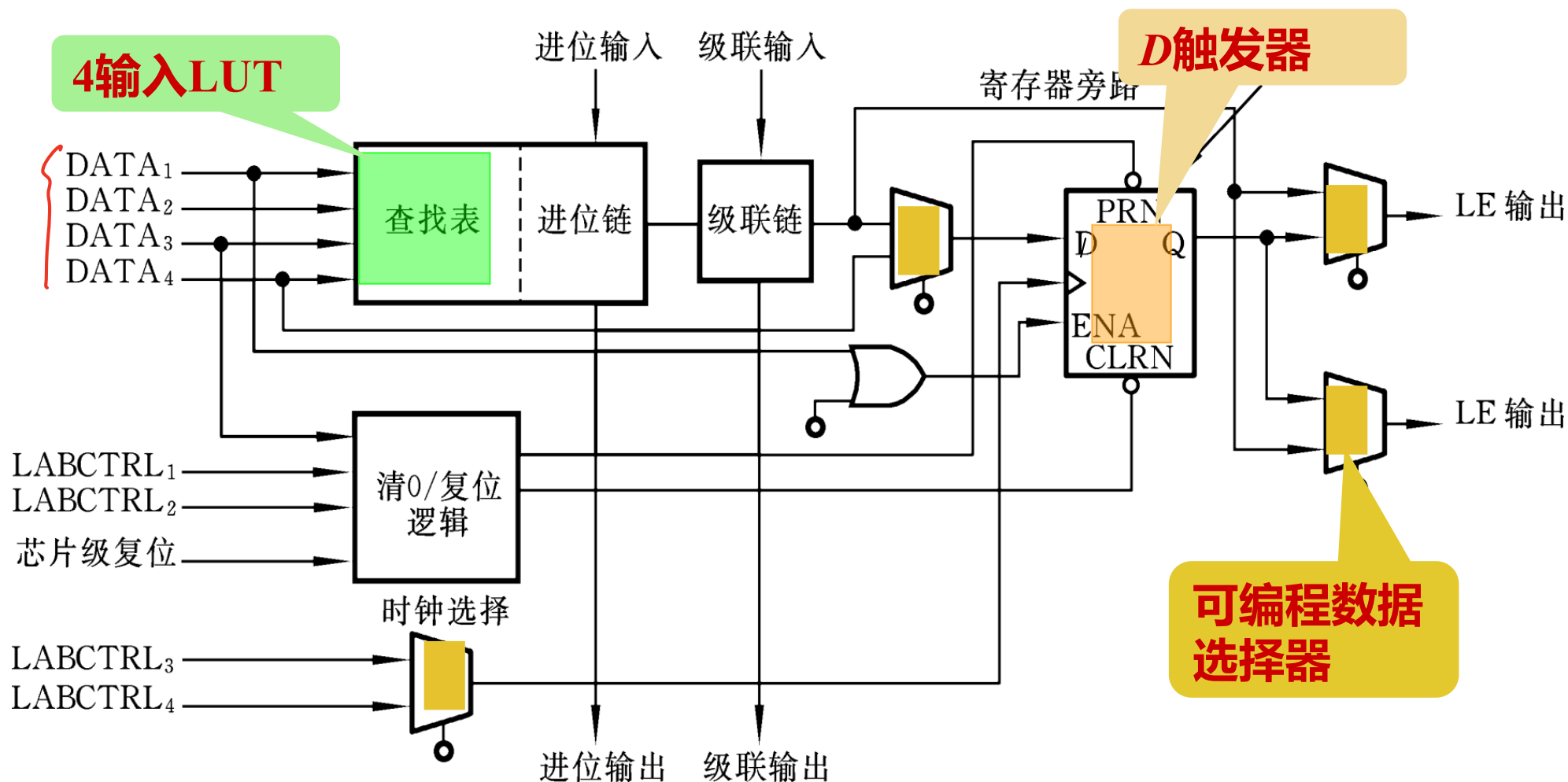
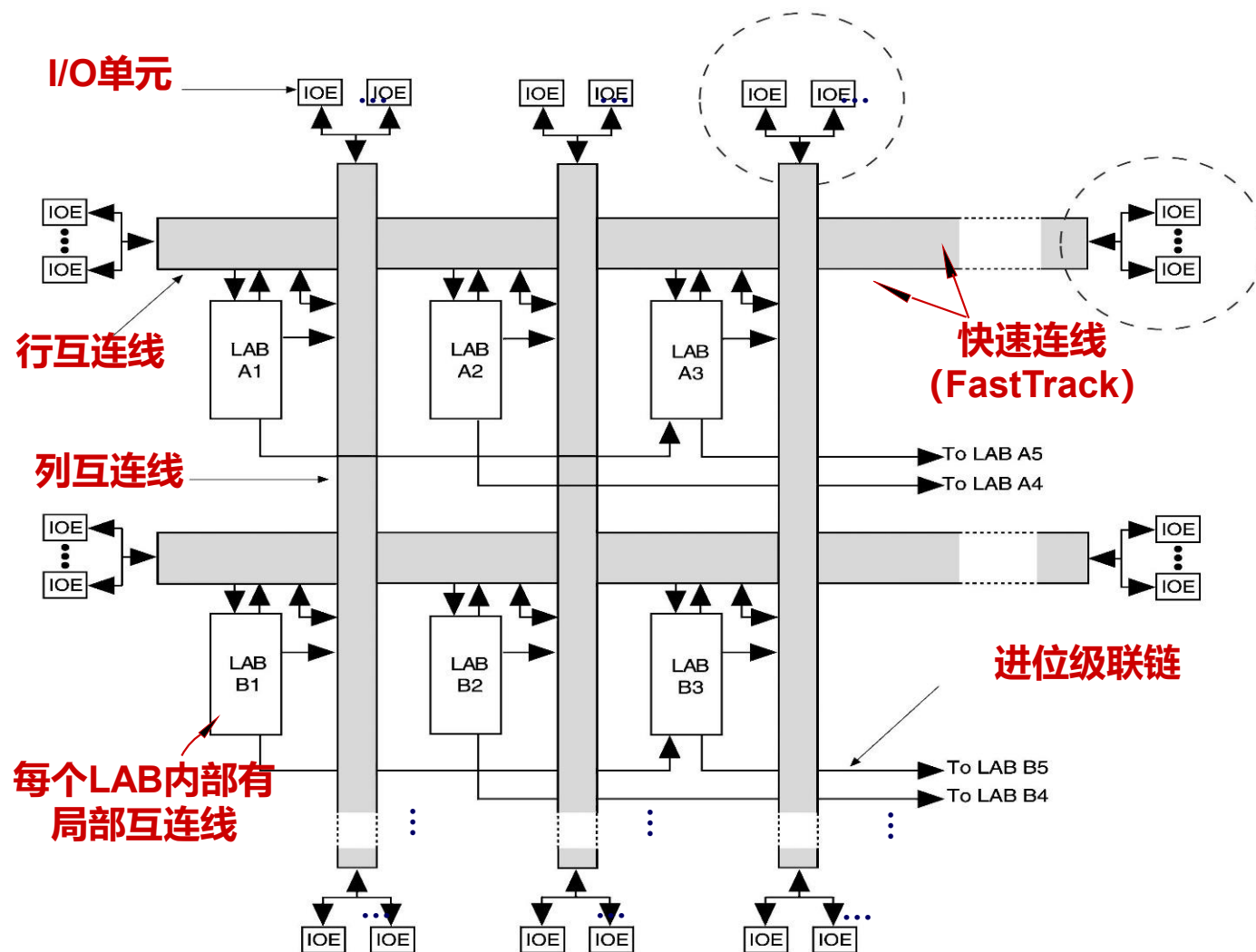


图6.4.8 FLEX 10K的LE结构示意图

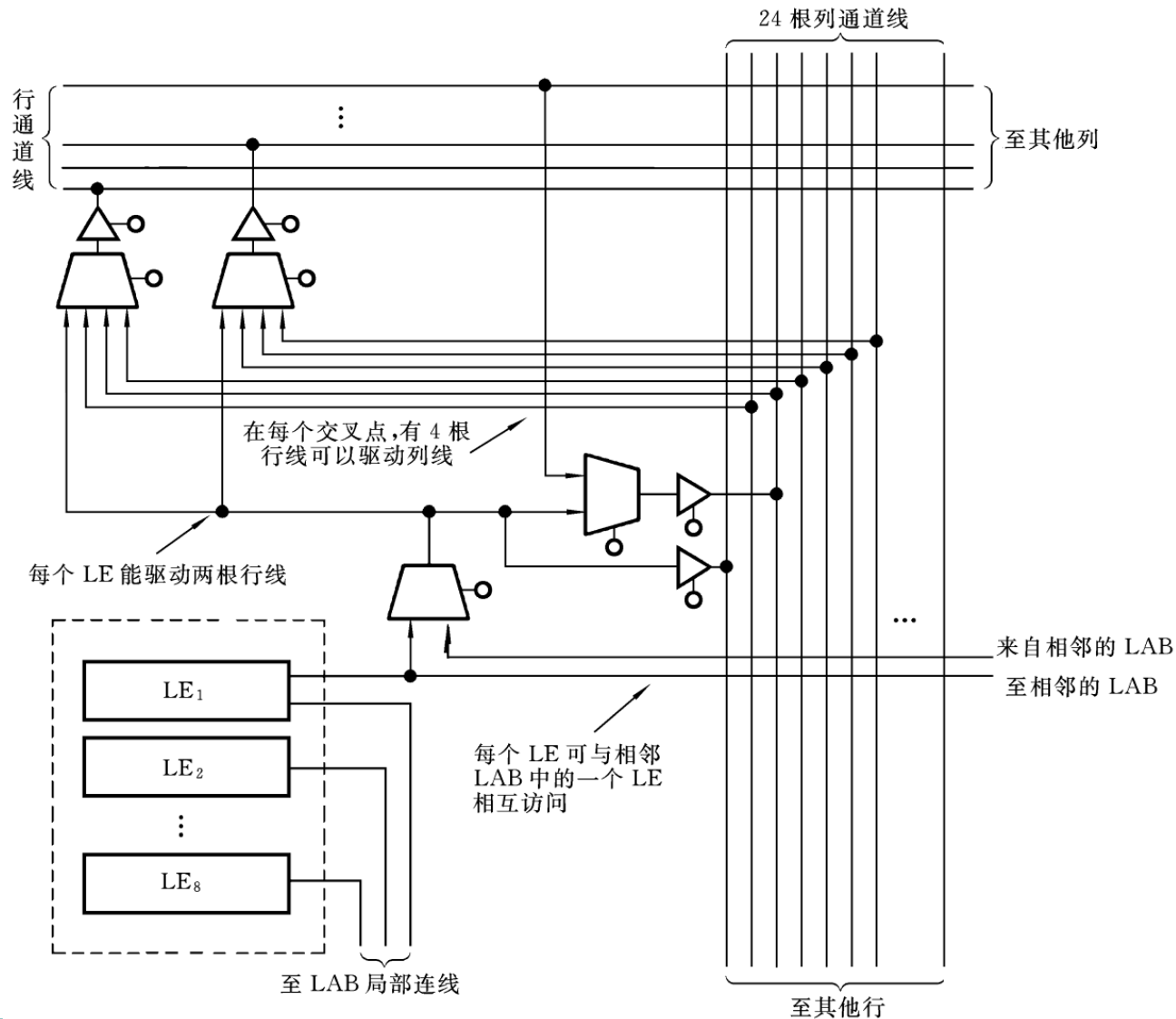
6.4.4 可编程布线资源

FPGA中有多种布线资源，包括局部布线资源、通用布线资源、I/O布线资源、专用布线资源和全局布线资源等，它们分别承担了不同的连线任务。



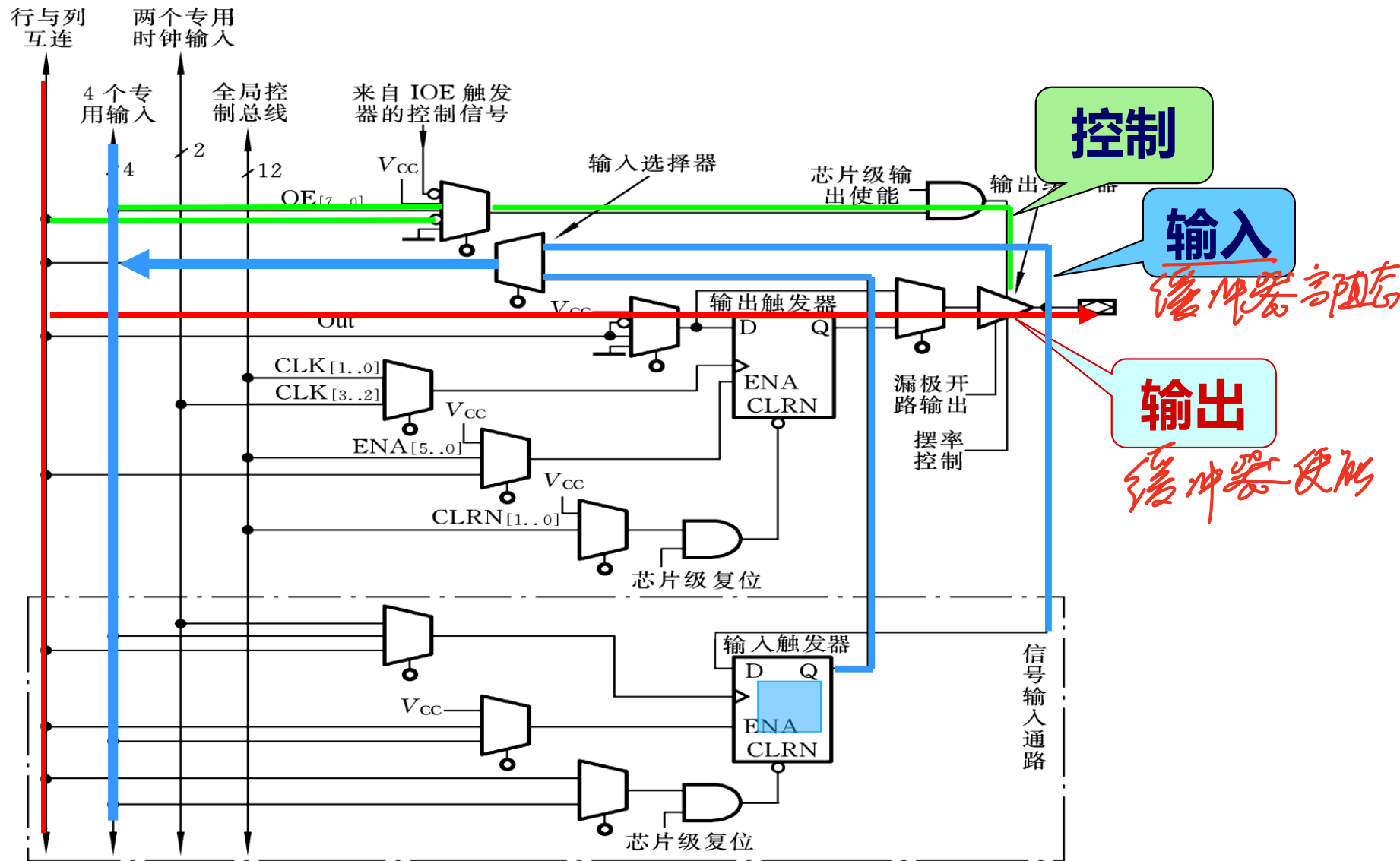
6.4.4 可编程布线资源

Altera公司将贯穿整个器件的一系列水平连线（或称行连线、行通道）和垂直连线（或称列连线、列通道）称为**快速通道**（Fast Track）。



6.4.5 I/O块

将引脚编程为输入、输出和双向功能。



CPLD与FPGA的区别

	CPLD <i>大的与或阵列</i>	FPGA
内部结构	Product - term	Look - up Table
程序存储	内部E ² PROM	<u>SRAM</u> , <u>外挂E²PROM</u>
资源类型	组合电路资源丰富	触发器资源丰富
集成度	低	高
使用场合	完成控制逻辑	能完成比较复杂的算法
速度	快	慢
其他资源	-	<u>EAB</u> , <u>锁相环</u>
保密性	可加密	一般不能保密

为什么FPGA需要编程数据装载？

- CPLD采用CMOS E²PROM工艺制造，编程后，即使切断电源，其逻辑也不会消失，且可以在系统编程（ISP特性）。
- FPGA的LUT由数据选择器和SRAM构成，切断电源后，其逻辑会消失。所以FPGA需要外部的PROM保存编程数据。每次通电，自动将PROM中的编程数据装载到FPGA中。