# 4.5 组合可编程逻辑器件

## 4.5.1 PLD的结构、表示方法及分类

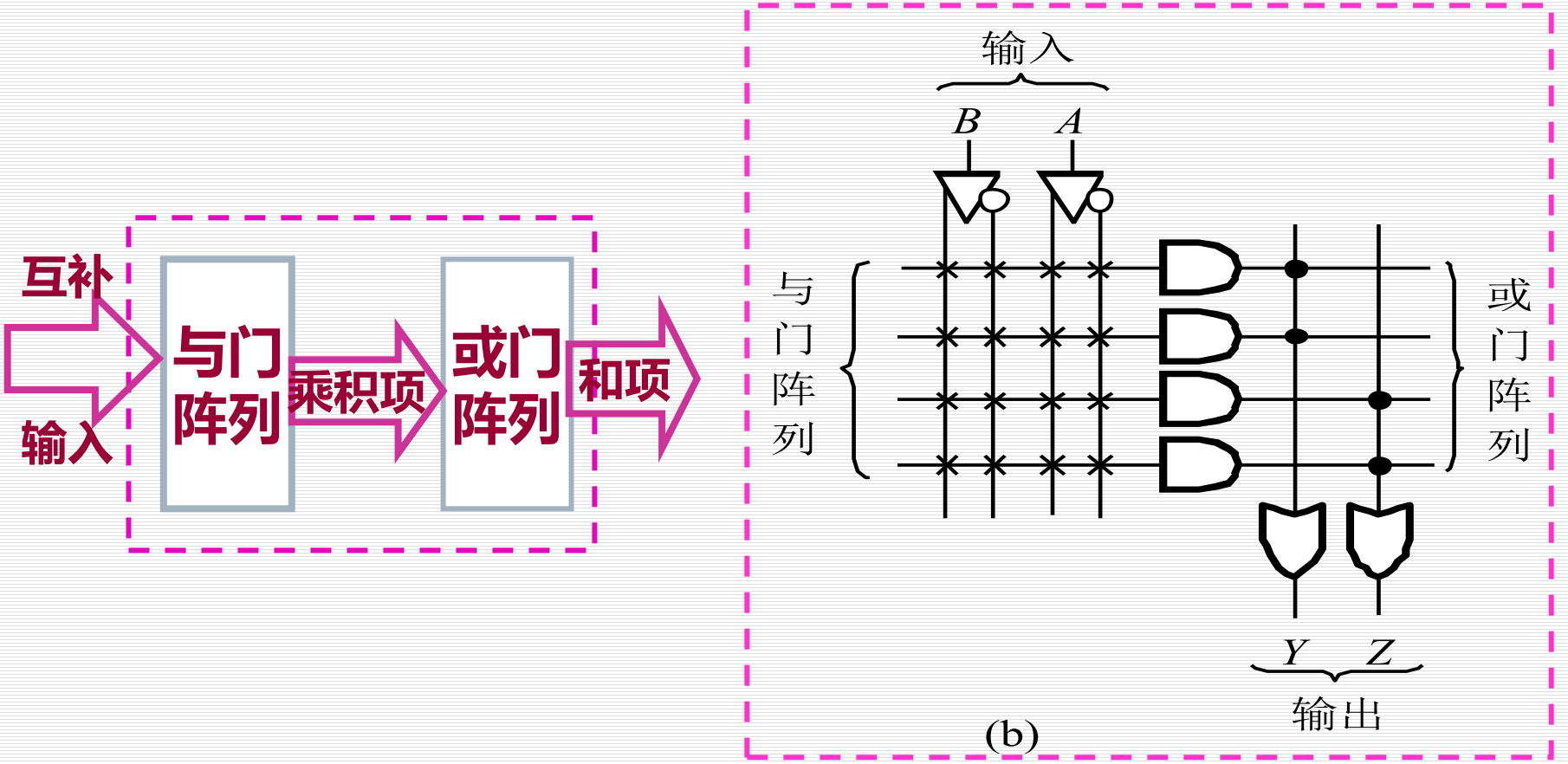
## 4.5.2 组合逻辑电路的PLD实现

# 4.5 组合可编程逻辑器件

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。

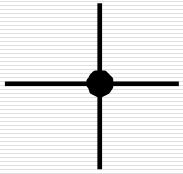
# 4.5.1 PLD的结构、表示方法及分类

## 1、PLD的基本结构

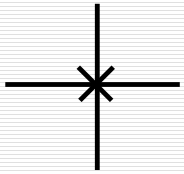

互补
输入
与门阵列
乘积项
或门阵列
和项
输入
B
A
与门阵列
或门阵列
Y
Z
输出
(b)

## 2. PLD的逻辑符号表示方法

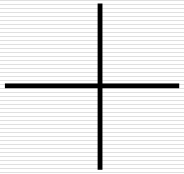
### (1) 连接的方式



硬线连接单元



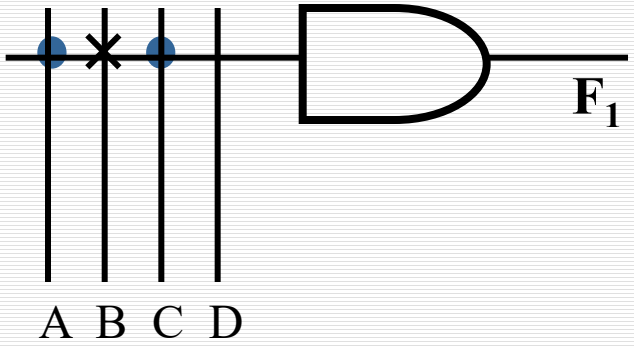
被编程接通单



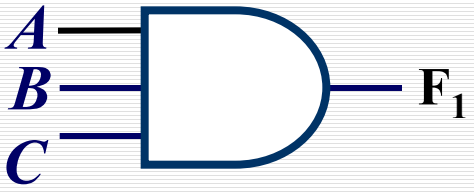
被编程擦除单元

# (2)基本门电路的表示方式

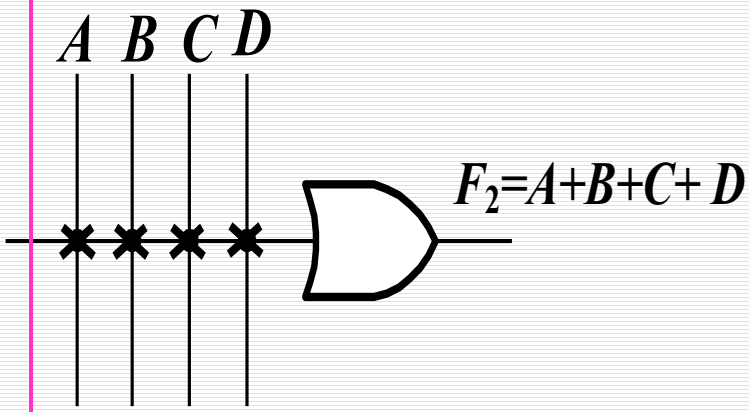
## 与门



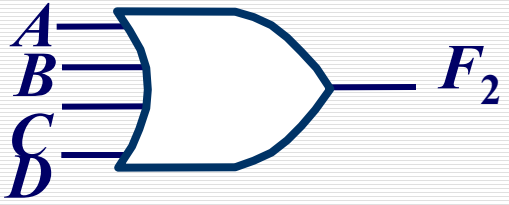

$F_1=A\cdot B\cdot C$

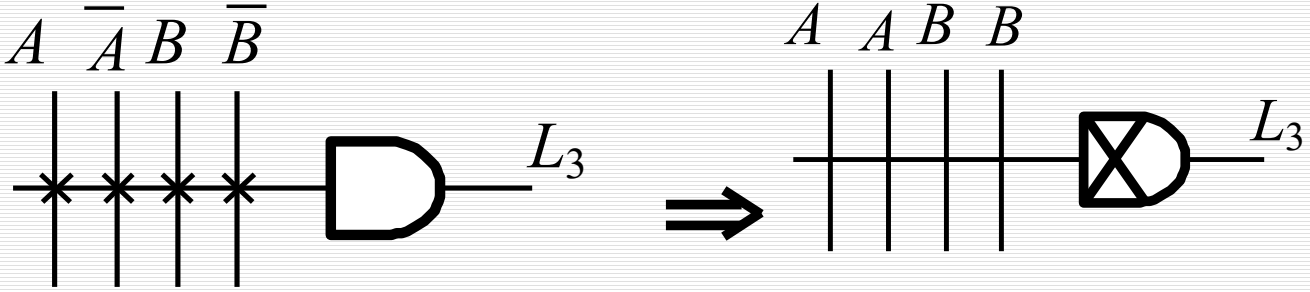



## 或门

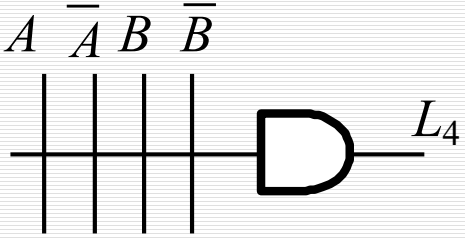



$F_2=A+B+C+D$

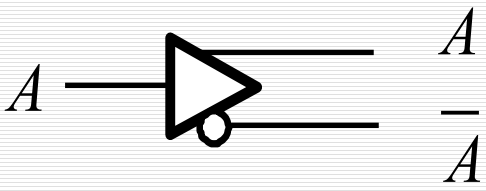




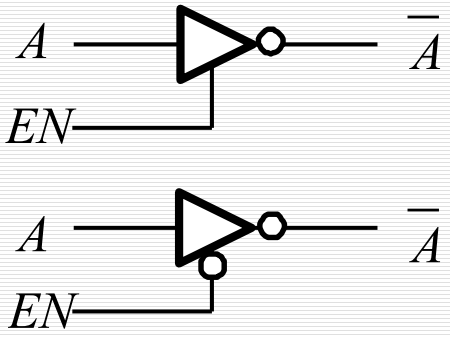

**输出恒等于0的与门**




**输出为1的与门**



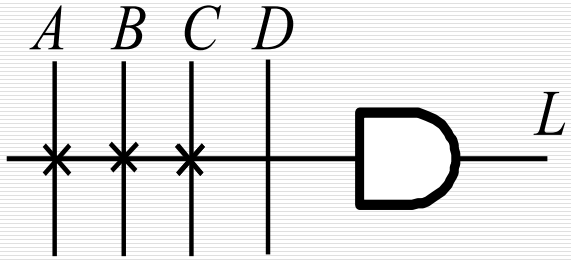

**输入缓冲器**



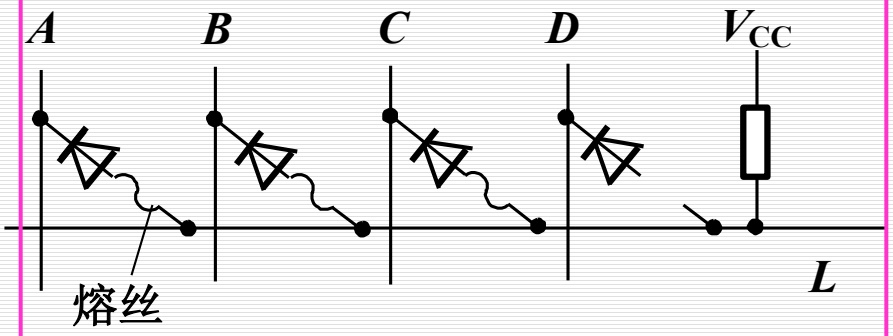

**三态输出缓冲器**

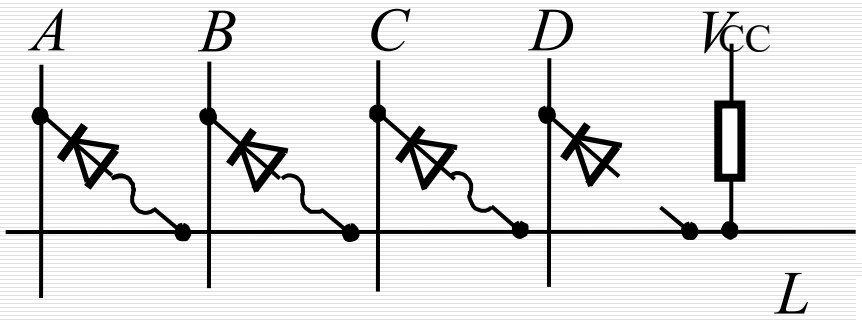
## (3) 编程连接技术

**PLD表示的与门**




**熔丝工艺的与门原理图**

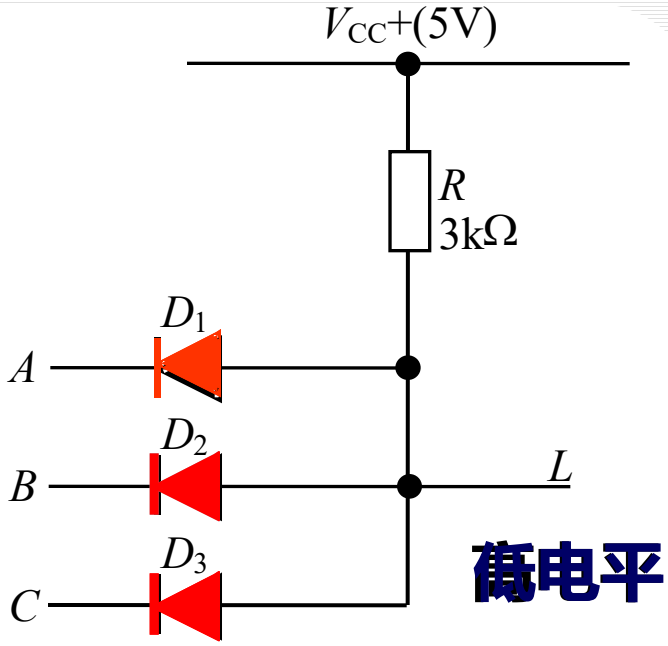





5V

5V

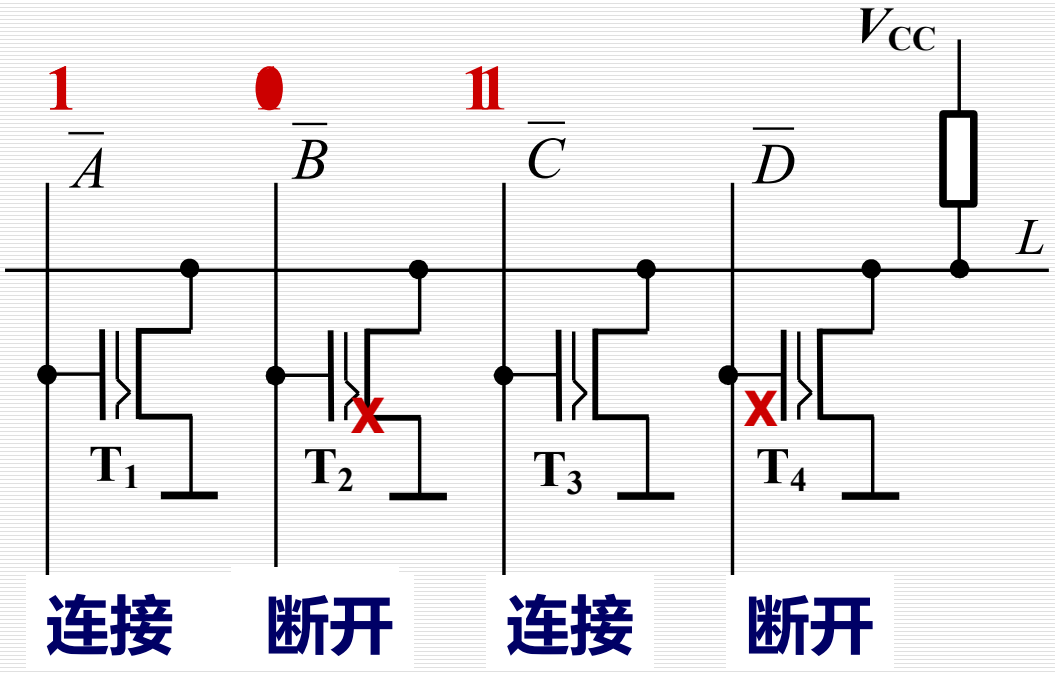
5V




A、B、C有一个输入低电平0V

A、B、C三个都输入高电平+5V

$L=A\bullet B\bullet C$




$A$、$B$、$C$ 中有一个为0

输出为0；

$A$、$B$、$C$ 都为1

输出为1。

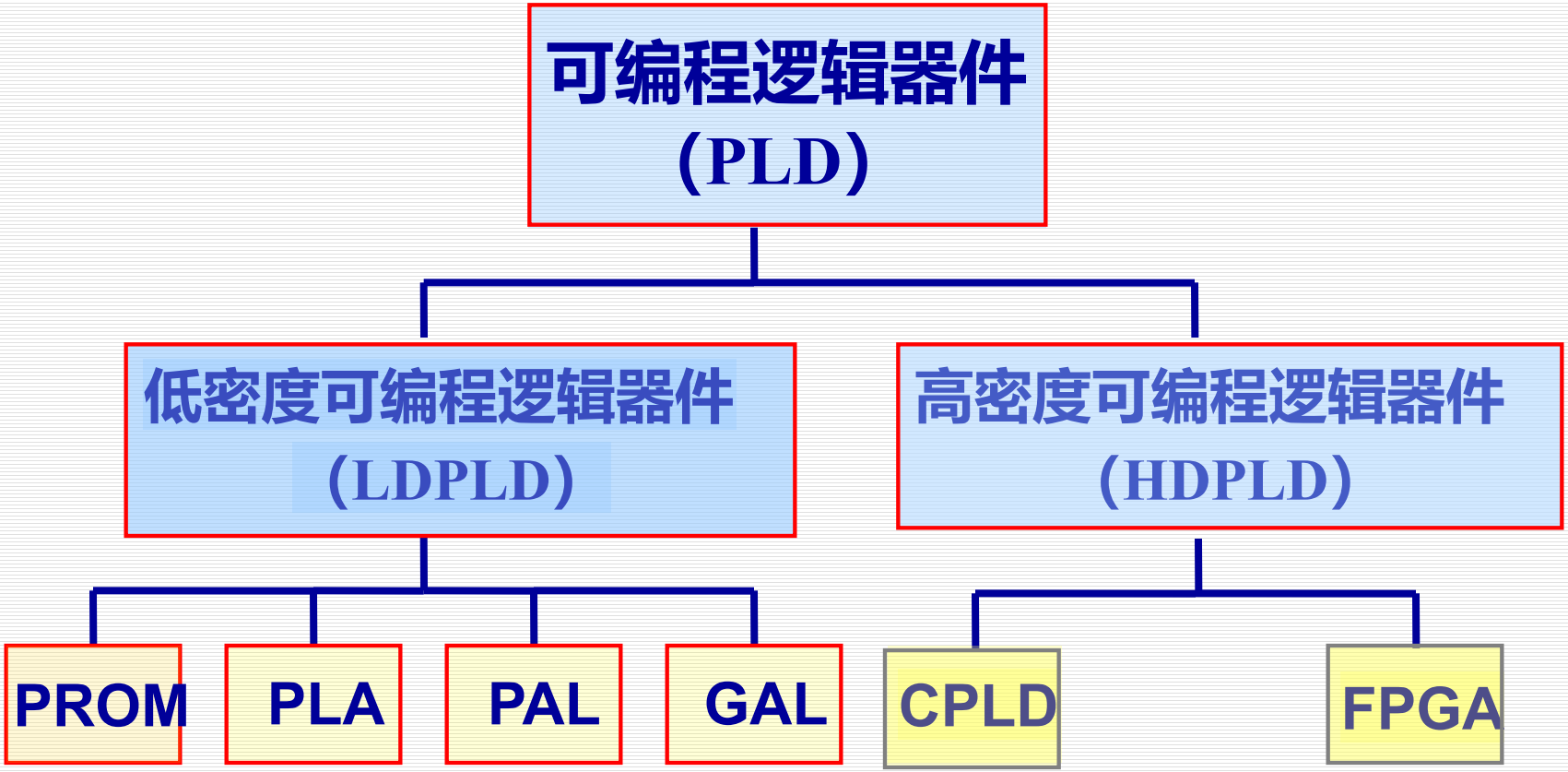
$L=ABC$

$L=AC$

器件的开关状态不同，电路实现逻辑函数也就不同

# 3.PLD的分类

## 按集成密度划分为



## 2、按结构特点划分

- 简单PLD (PAL，GAL)

- 复杂的可编程器件(CPLD)：

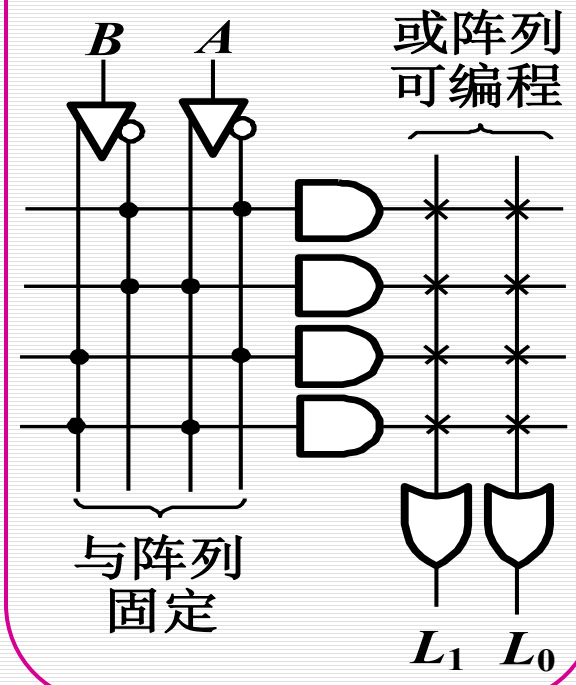
CPLD的代表芯片如：Altera的MAX系列

- 现场可编程门阵列(FPGA)

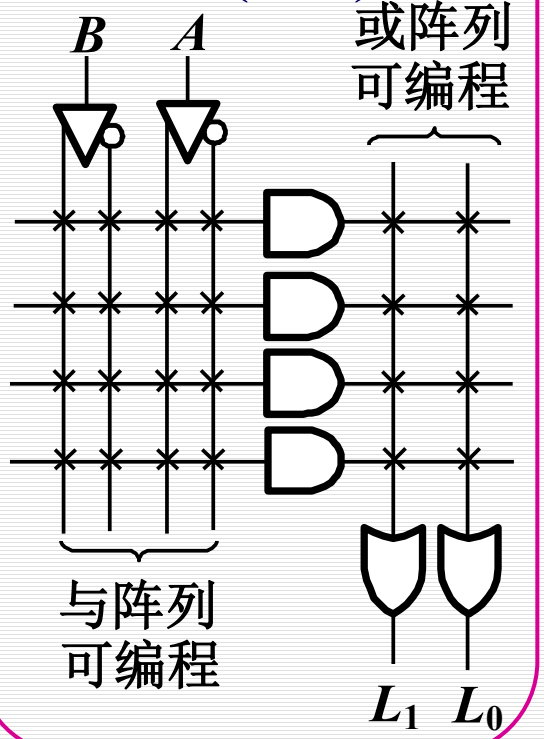
# 3. 按PLD中的与、或阵列是否编程分

## PLD中的三种与、或阵列

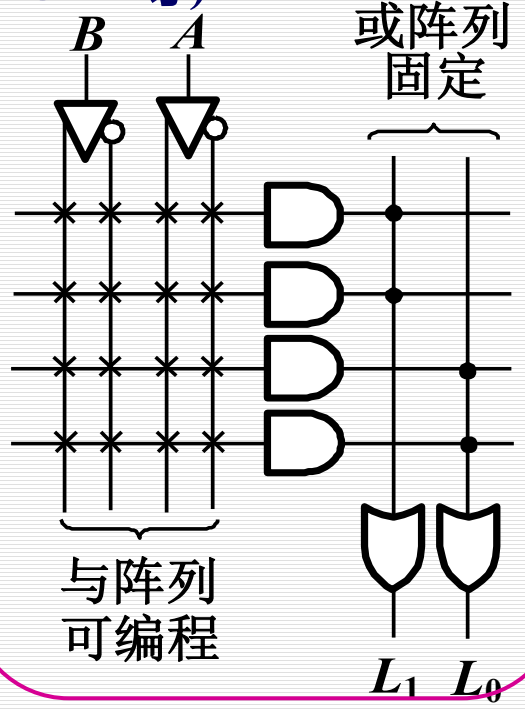
**与阵列固定，或阵列可编程(PROM)**




**与阵列、或阵列均可编程(PLA)**

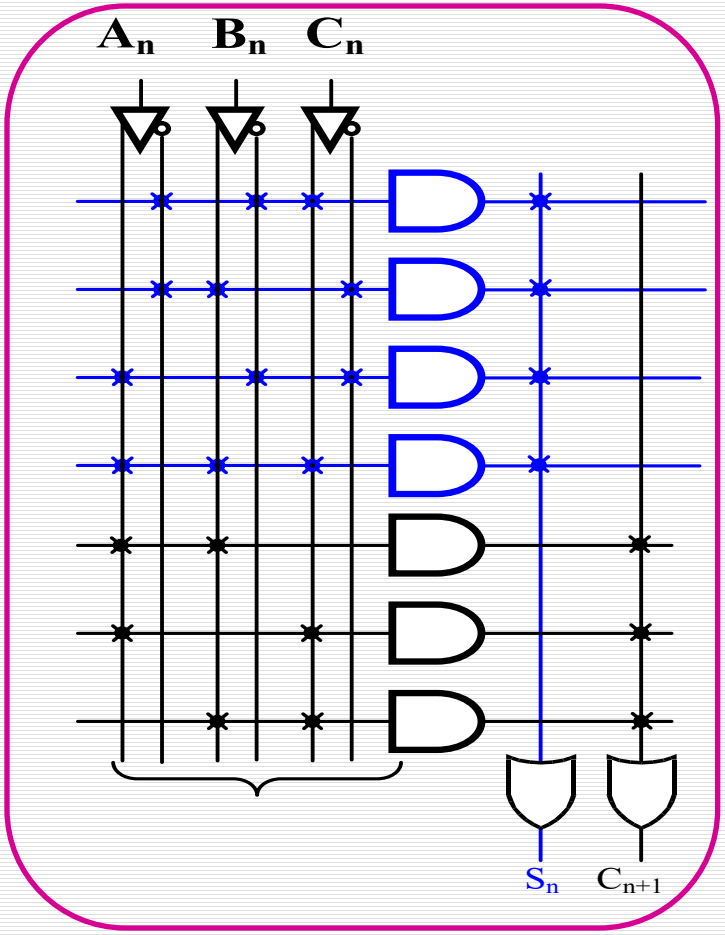



**与阵列可编程，或阵列固定(PAL和GAL等)**



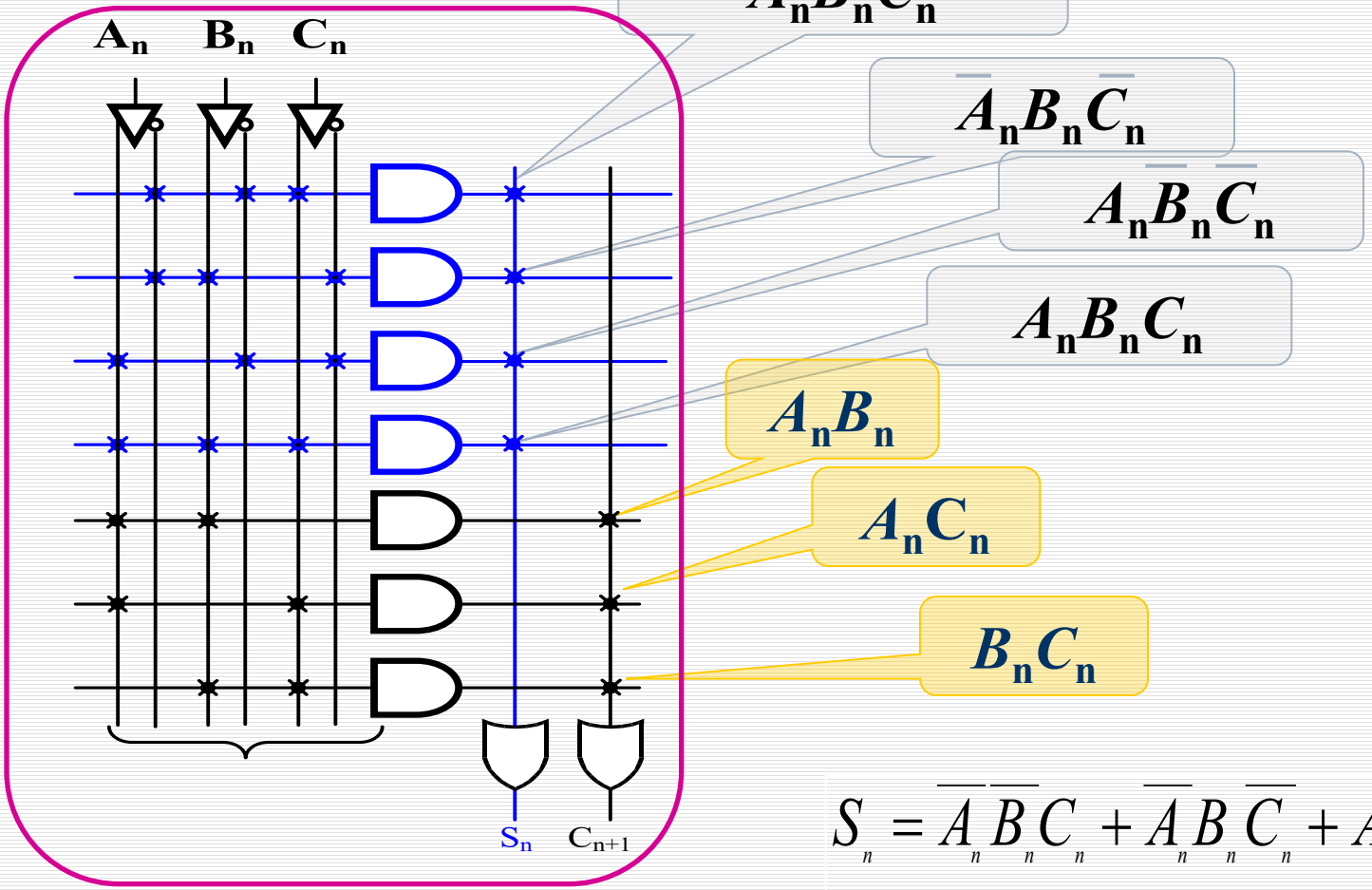
# 4.5.2 组合逻辑电路的 PLD 实现

例1 由PLA构成的逻辑电路如图所示，试写出该电路的逻辑表达式，并确定其逻辑功能。




写出该电路的逻辑表达式：

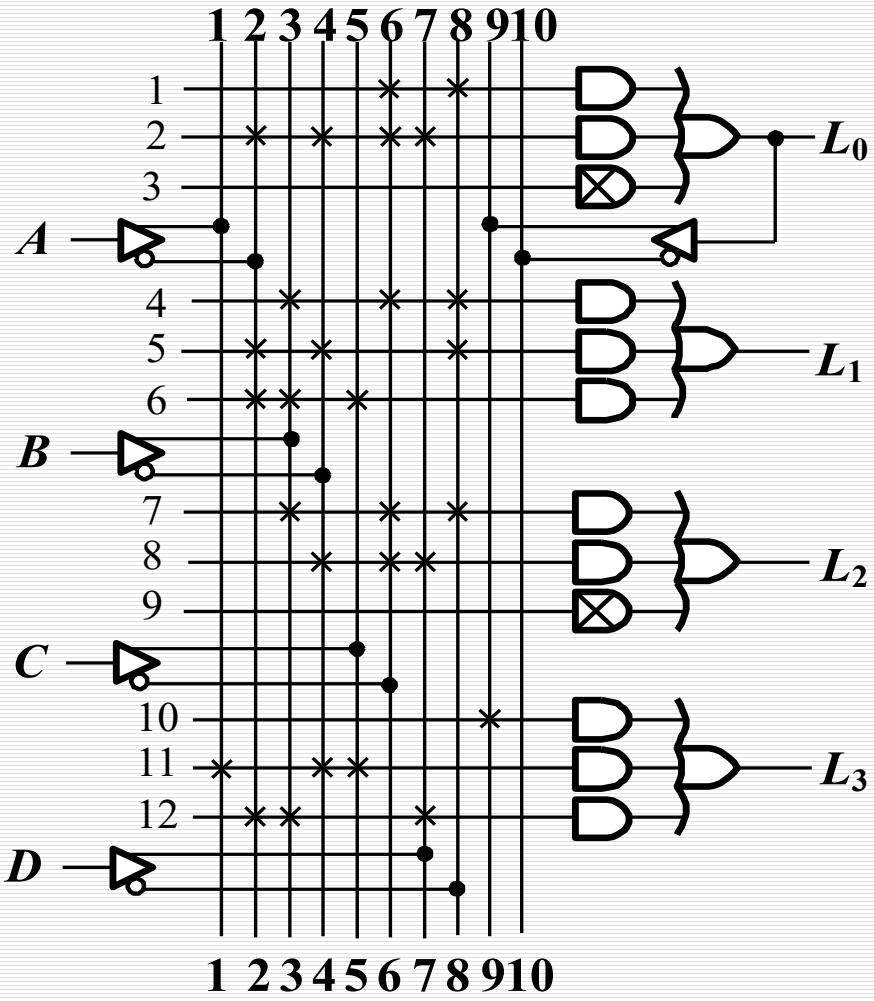
# 全加器




$$S_n = \overline{A}_n\overline{B}_nC_n + \overline{A}_nB_n\overline{C}_n + A_n\overline{B}_n\overline{C}_n + A_nB_nC_n$$

$$C_{n+1} = A_nB_n + A_nC_n + B_nC_n$$

# 例2 试写出该电路的逻辑表达式。




$$L_0 = \overline{C}\,\overline{D} + \overline{A}BC\overline{D}$$

$$L_1 = B\overline{C}\,\overline{D} + \overline{A}B\overline{D} + \overline{A}BC$$

$$L_2 = B\overline{C}\,\overline{D} + \overline{B}C\overline{D}$$

$$L_3 = L_0 + A\overline{B}C + \overline{A}BD$$

- **课堂练习：**

  试用图4.5.10（p194）所示的可编程阵列逻辑PAL实现码转换电路，输入为4位8421BCD码，输出为余3码。

# 6.6 简单的时序可编程逻辑器件(GAL)

## 6.6.1 GAL的结构

## 6.6.2 GAL的输出逻辑宏单元

## 6.6.3 GAL的控制字

# 1. 时序可编程逻辑器件的主要类型

（1） 通用阵列逻辑（GAL）

在PLA和PAL基础上发展起来的增强型器件.电路设计者可根据需要编程，对宏单元的内部电路进行不同模式的组合，从而使输出功能具有一定的灵活性和通用性。

（2）复杂可编程逻辑器件（CPLD）

集成了多个逻辑单元块，每个逻辑块就相当于一个GAL器件。这些逻辑块可以通过共享可编程开关阵列组成的互连资源，实现它们之间的信息交换，也可以与周围的I/O模块相连，实现与芯片外部交换信息。

### （3） 现场可编程门阵列（FPGA）

芯片内部主要由许多不同功能的可编程逻辑模块组成，靠纵横交错的分布式可编程互联线连接起来，可构成极其复杂的逻辑电路。它更适合于实现多级逻辑功能，并且具有更高的集成密度和应用灵活性在软件上，亦有相应的操作系统配套。这样，可使整个数字系统（包括软、硬件系统）都在单个芯片上运行，即所谓的SOC技术。

## 2. PAL的不足：

（1）由于采用的是双极型熔丝工艺，一旦编程后不能修改；

（2）输出结构类型太多，给设计和使用带来不便。

## 3. GAL的优点：

（1）采用电可擦除的$E^2$CMOS工艺可以多次编程；

（2）输出端设置了可编程的输出逻辑宏单元（OLMC）通过编程可将OLMC设置成不同的工作状态，即一片GAL便可实现PAL 的5种输出工作模式。器件的通用性强；

（3）GAL工作速度快，功耗小

第4.5节介绍了PLD实现组合逻辑电路，在此基础上，在PLD中增加触发器及反馈就可以实现时序逻辑。

例： 对于例6.3.2中的110序列脉冲检测电路，改用含D触发器的可编程逻辑器件实现，试求激励方程和输出方程，并画出编程后的逻辑图。

解（1）根据例6.3.2状态转换表，得到状态转换真值表。

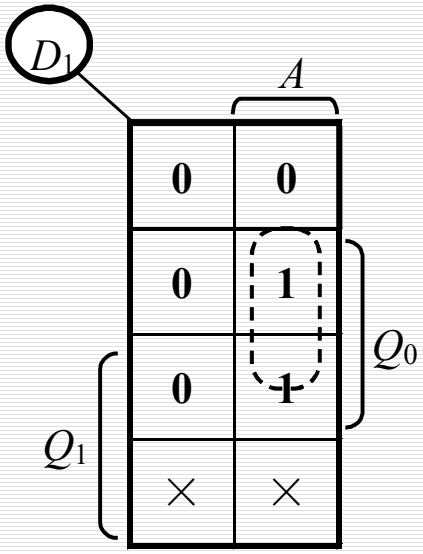
状态转换表

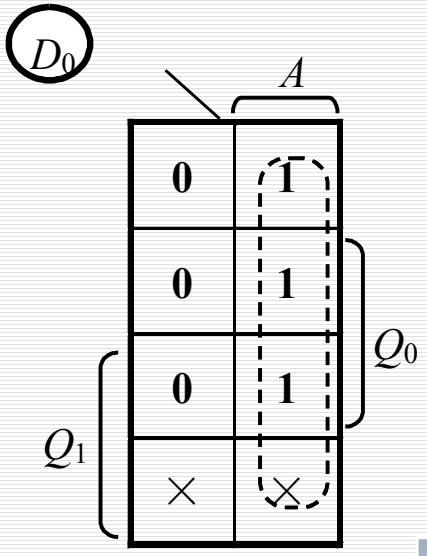
| $Q_1^nQ_0^n$ | $Q_1^{n+1}Q_0^{n+1}/Y$ | |
|---|---|---|
| | $A$=0 | $A$=1 |
| **0 0** | 0 0/0 | 0 1/0 |
| **0 1** | 0 0/0 | 1 1/0 |
| **1 1** | 0 0/1 | 1 1/0 |

状态转换真值表

| $Q_1^n$ | $Q_0^n$ | $A$ | $Q_1^{n+1}$ | $Q_0^{n+1}$ | $Y$ |
|---|---|---|---|---|---|
| **0** | **0** | **0** | 0 | 0 | 0 |
| **0** | **0** | **1** | 0 | 1 | 0 |
| **0** | **1** | **0** | 0 | 0 | 0 |
| **0** | **1** | **1** | 1 | 1 | 0 |
| **1** | **1** | **0** | 0 | 0 | 1 |
| **1** | **1** | **1** | 1 | 1 | 0 |

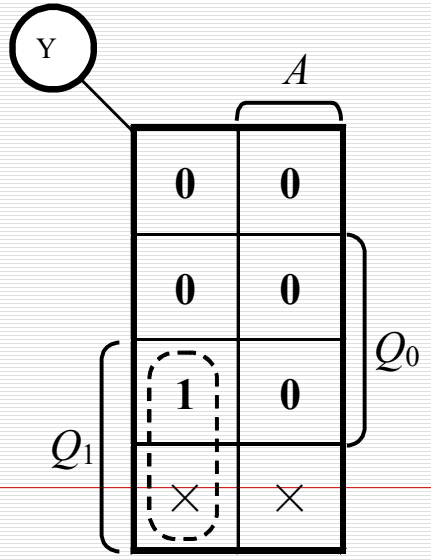
## （2）由于$D$触发器的特性方程$Q^{n+1}=D$，根据状态转换真值表，画卡诺图求激励方程。




$$D_1 = Q_0 A$$

$$D_0 = A$$




$$Y = Q_1 \bar{A}$$

**状态转换真值表**

| $Q_1^n$ | $Q_0^n$ | A | $Q_1^{n+1}$ | $Q_0^{n+1}$ | Y |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 |

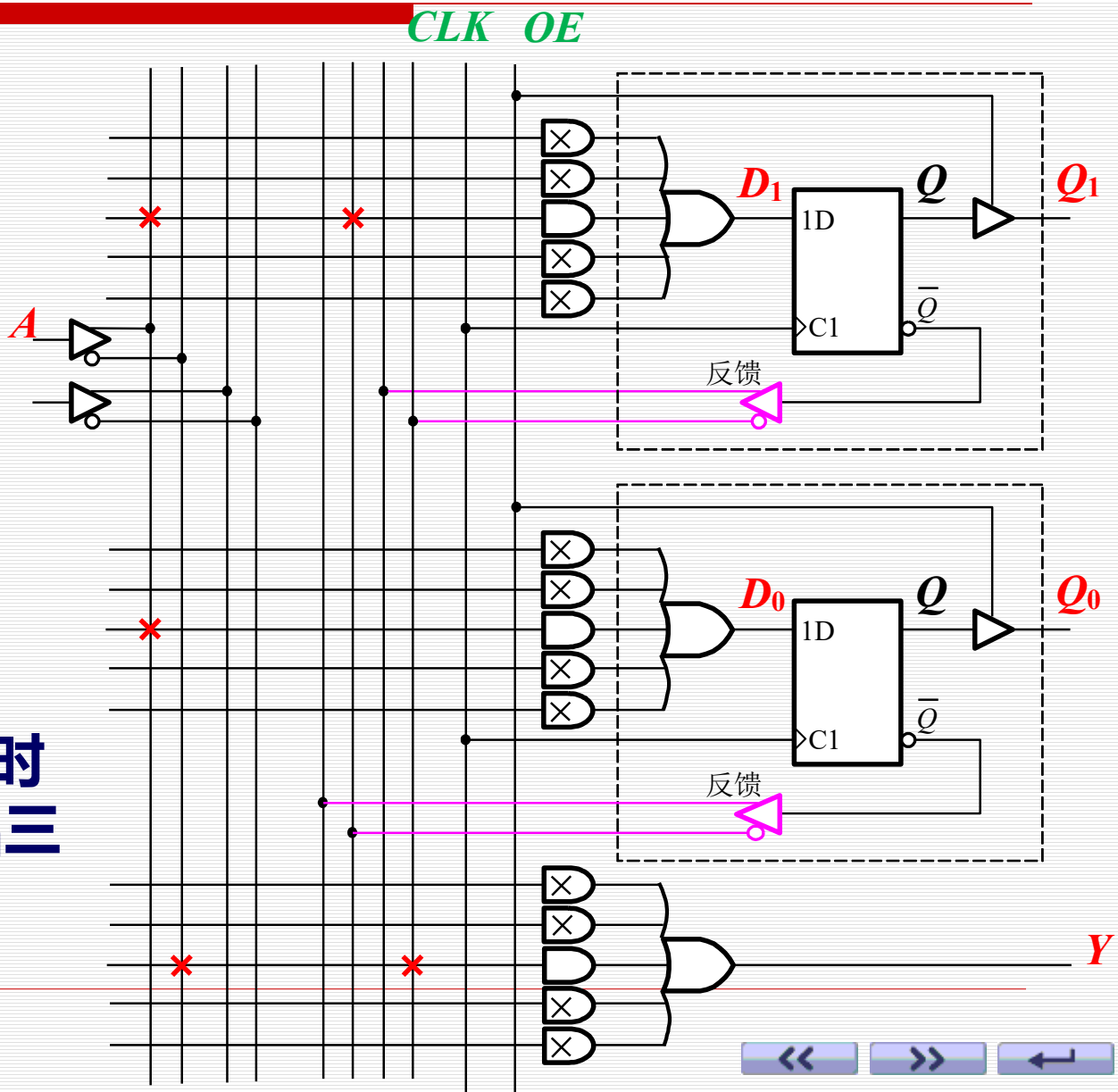
# （3）用含D触发器的PLD实现编程后的逻辑图如图。

$$D_1 = Q_0 A$$

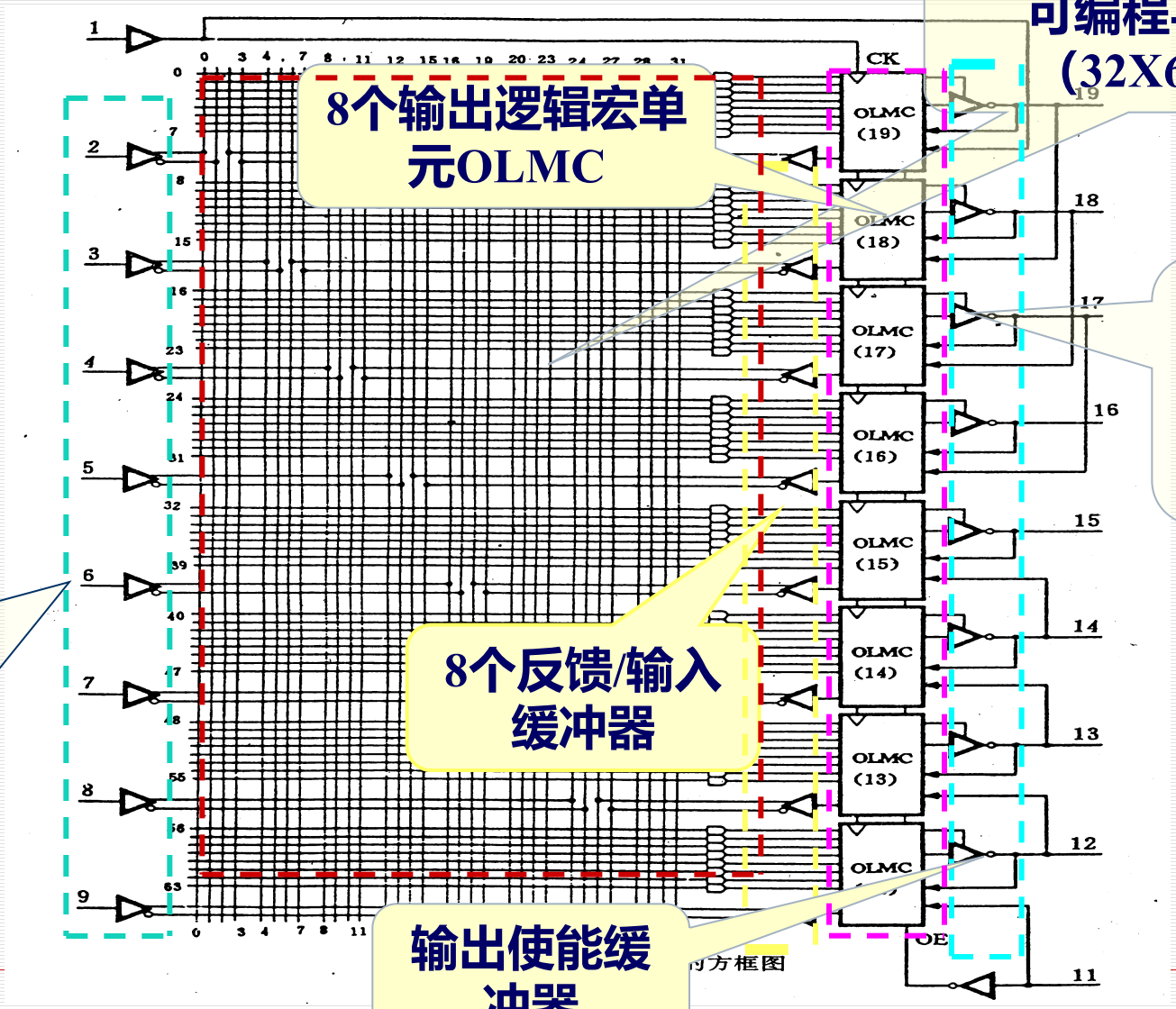
$$D_0 = A$$

$$Y = Q_1 \overline{A}$$

- CLK是统一的时钟，OE是输出三态门使能信号。

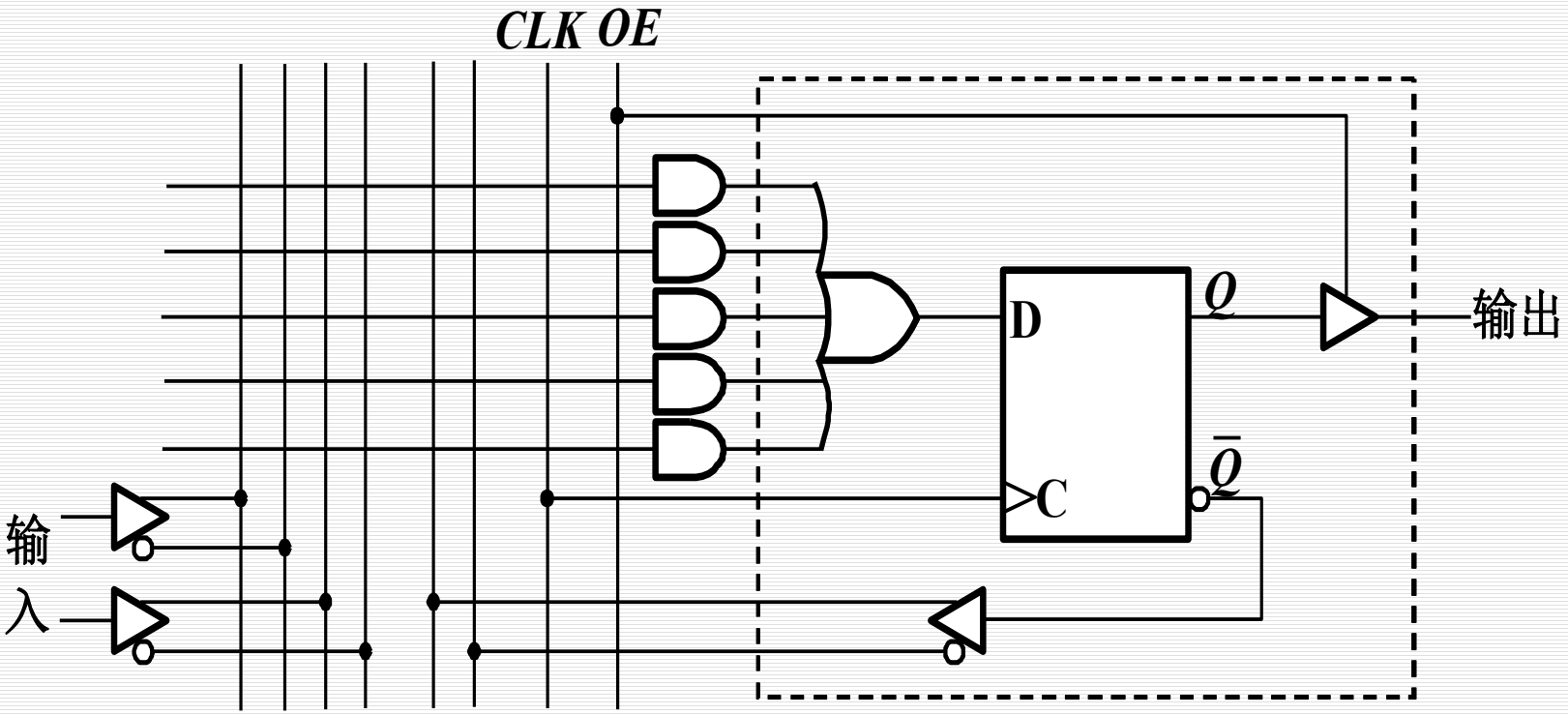


# 6.6.1 GAL的结构——GAL16V8的结构为例



# 6.6.2 GAL中的输出逻辑宏单元

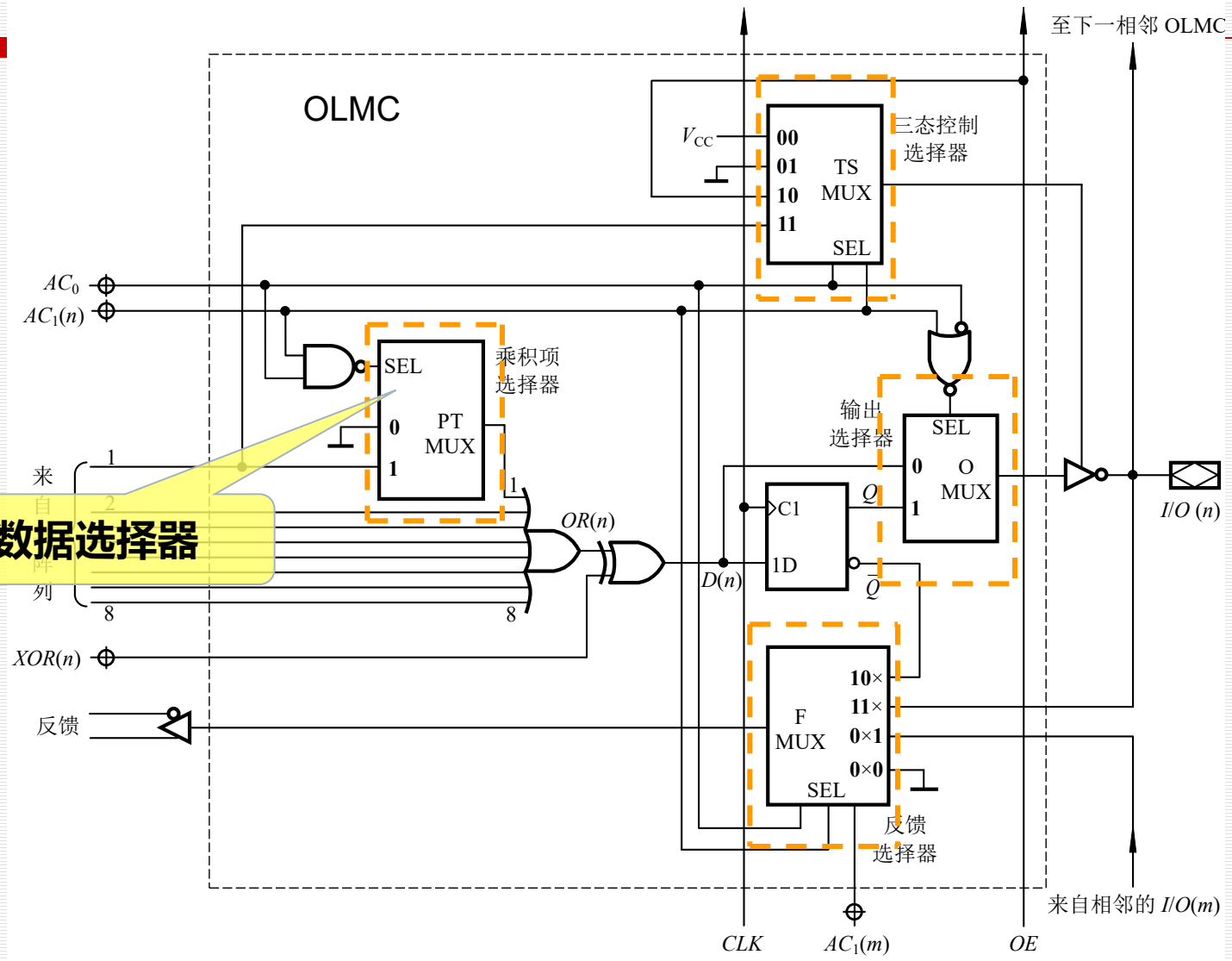
## 1. 寄存器型PAL

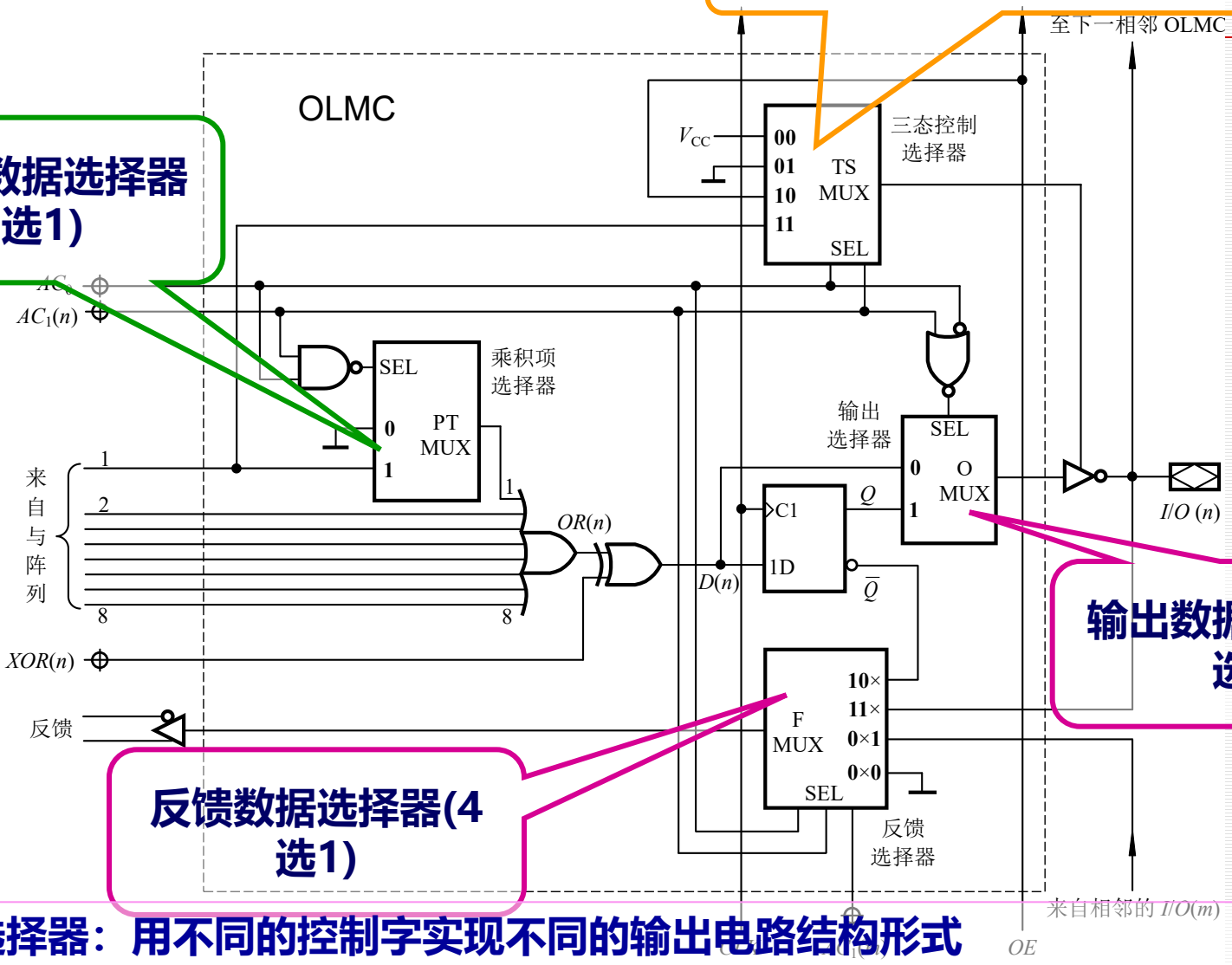



寄存器型PAL如图所示，在组合PLD基础上增加了D触发器，并反馈回到输入与阵列，满足时序电路设计要求。

## 2. GAL中的输出宏单元

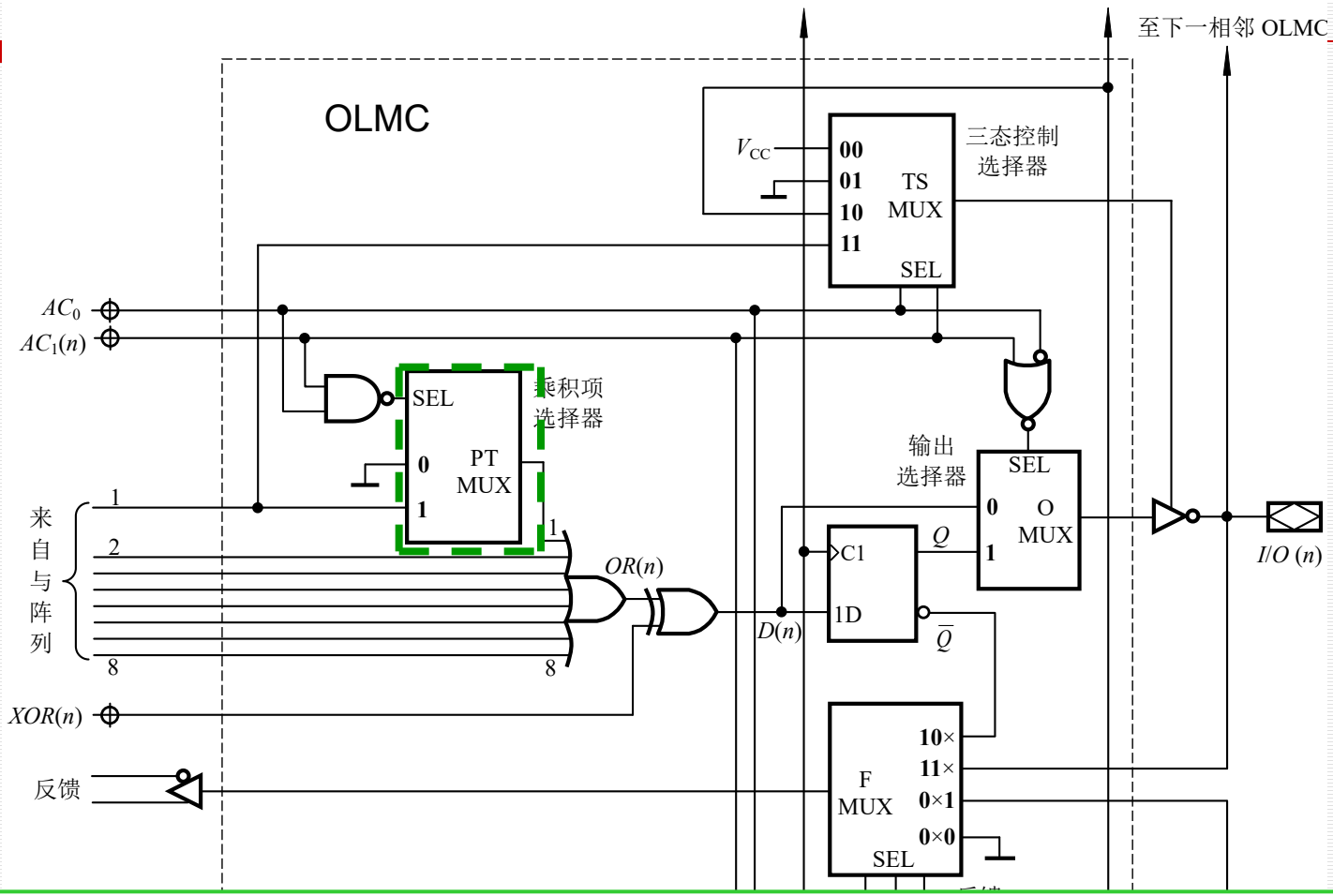
GAL的电路结构与PAL类似，由可编程的与逻辑阵列、固定的或逻辑阵列和输出电路组成，但GAL的输出端增设了可编程的的输出逻辑宏单元（OLMC）。通过编程可将OLMC设置为不同的工作状态，可实现PAL的所有输出结构，产生组合、时序逻辑电路输出。


至下一相邻 OLMC
OLMC
$V_{CC}$
00
01
10
11
TS
MUX
SEL
三态控制
选择器
$AC_0$
$AC_1(n)$
SEL
0
1
PT
MUX
乘积项
选择器
输出
选择器
SEL
0
1
O
MUX
$I/O\ (n)$
来
自
阵
列
1
2
8
8
$OR(n)$
C1
1D
Q
$\overline{Q}$
$D(n)$
$XOR(n)$
反馈
F
MUX
10×
11×
0×1
0×0
SEL
反馈
选择器
来自相邻的 $I/O(m)$
$CLK$
$AC_1(m)$
$OE$
数据选择器



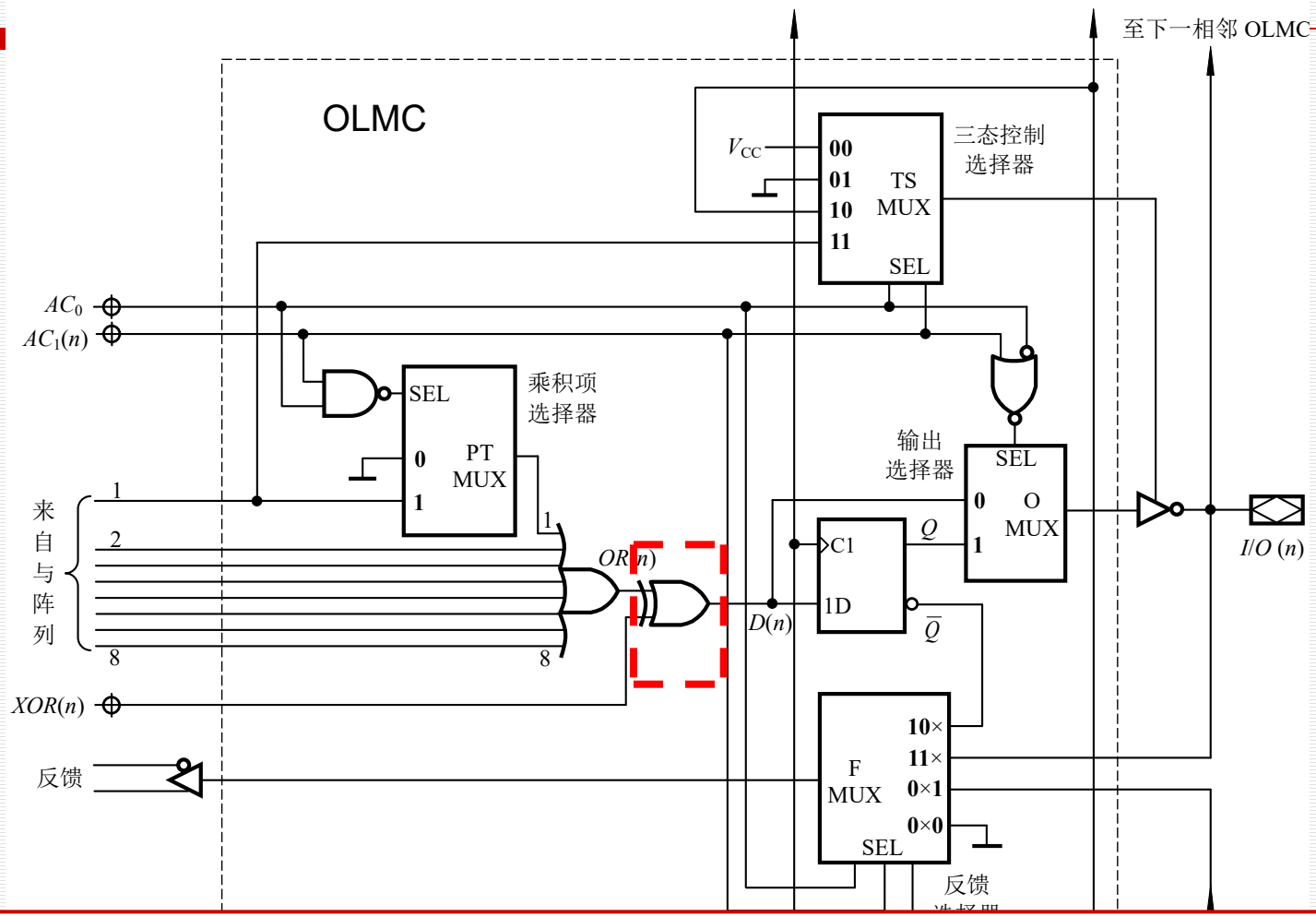

**4个数据选择器：用不同的控制字实现不同的输出电路结构形式**

# (1)输入电路—由乘积项数据选择器(2选1)PTMUX控制



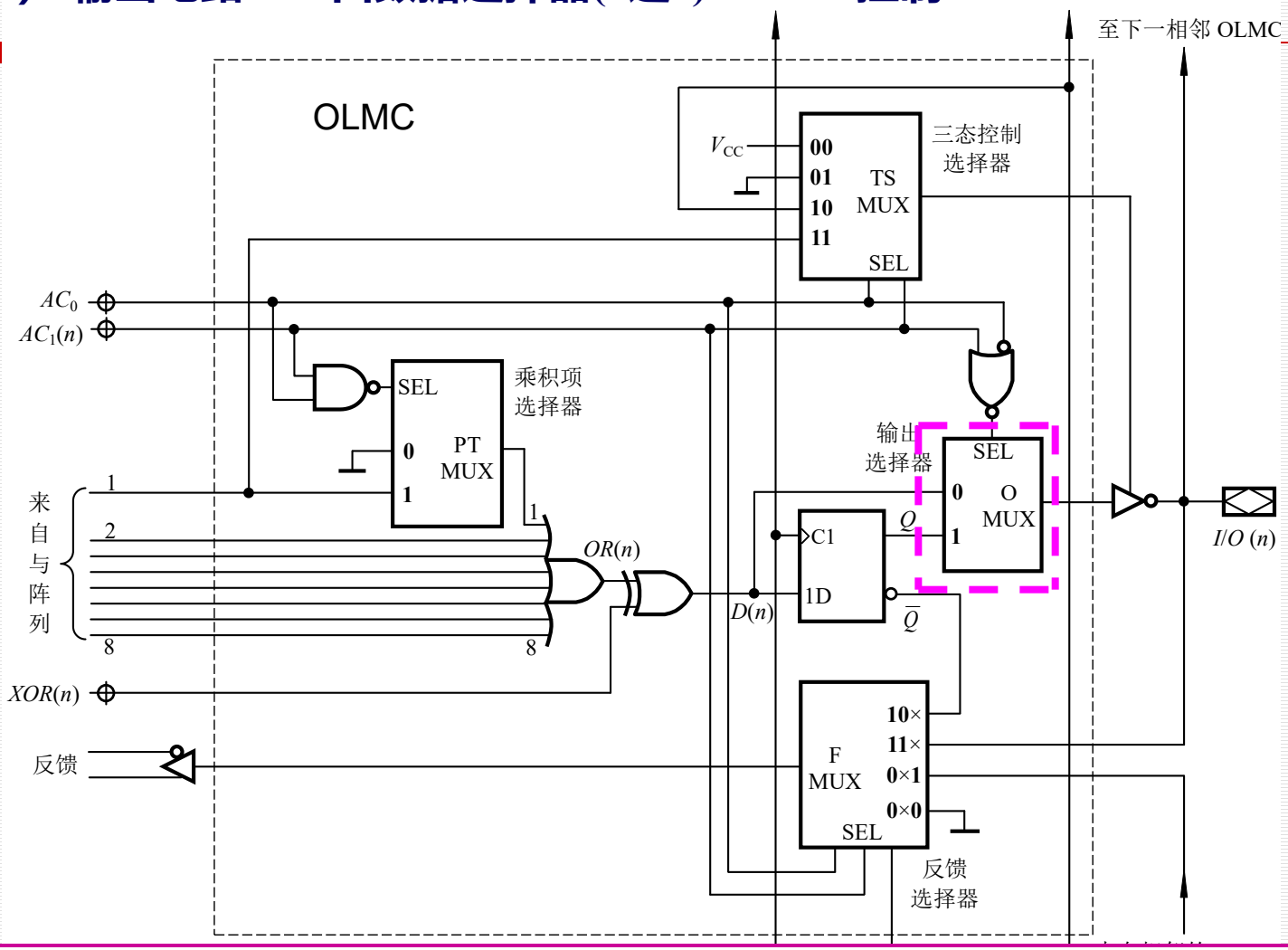

**乘积项数据选择器：根据AC0和AC1(n)决定与逻辑阵列的第一乘积项是否作为或门的一个输入端。**

# (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)=OR(n) 。**

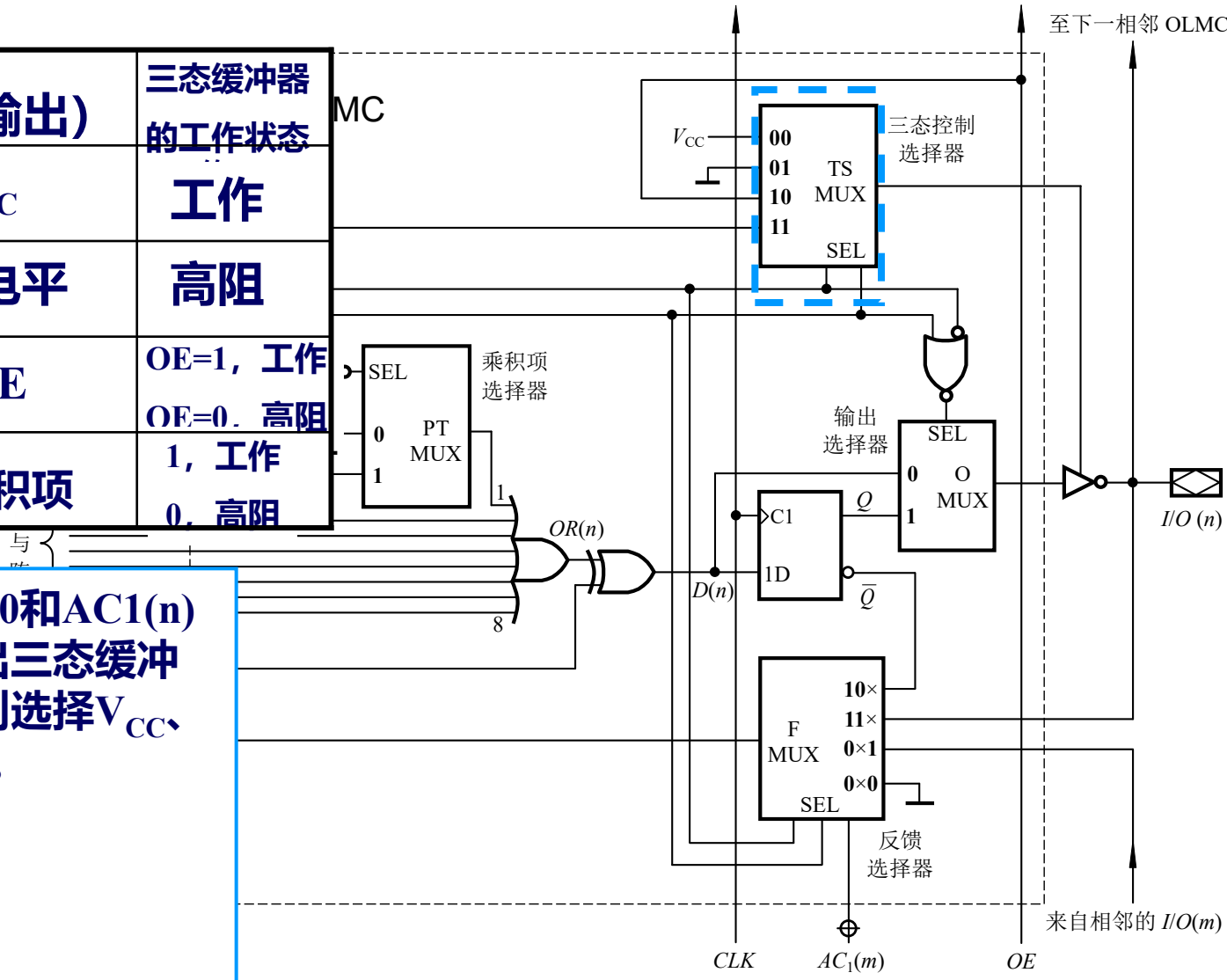
# (3) 输出电路——由数据选择器(2选1) OMUX控制




**OMUX：根据AC0和AC1(n)决定OLMC是组合输出还是寄存器输出模式**

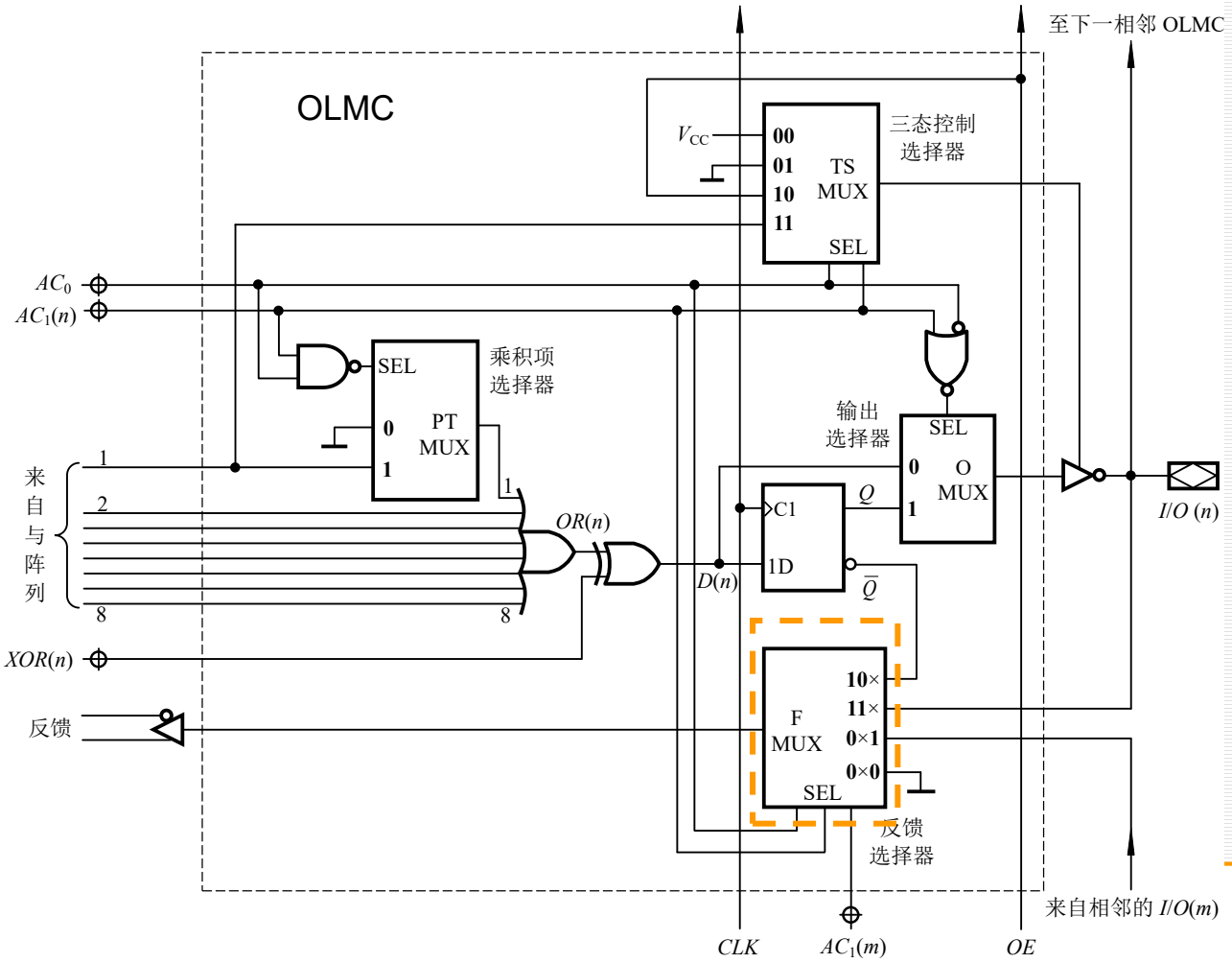
# 由三态数据选择器(4选1)控制输出选择器的选通端SEL

| 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<br>OE=0，高阻 |
| 1 1 | 第一乘积项 | 1，工作<br>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。

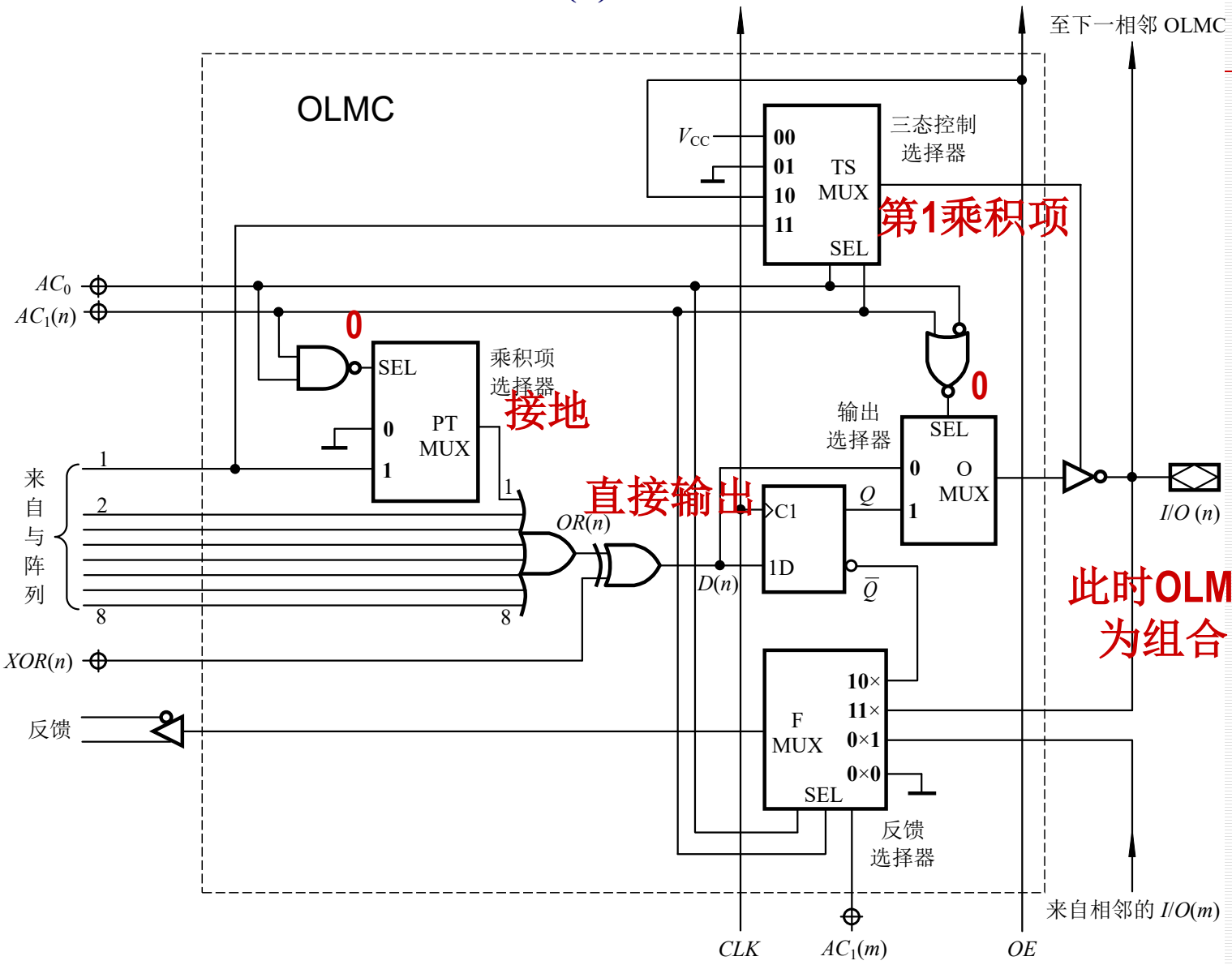
# 反馈数据选择器(4选1)——FMUX



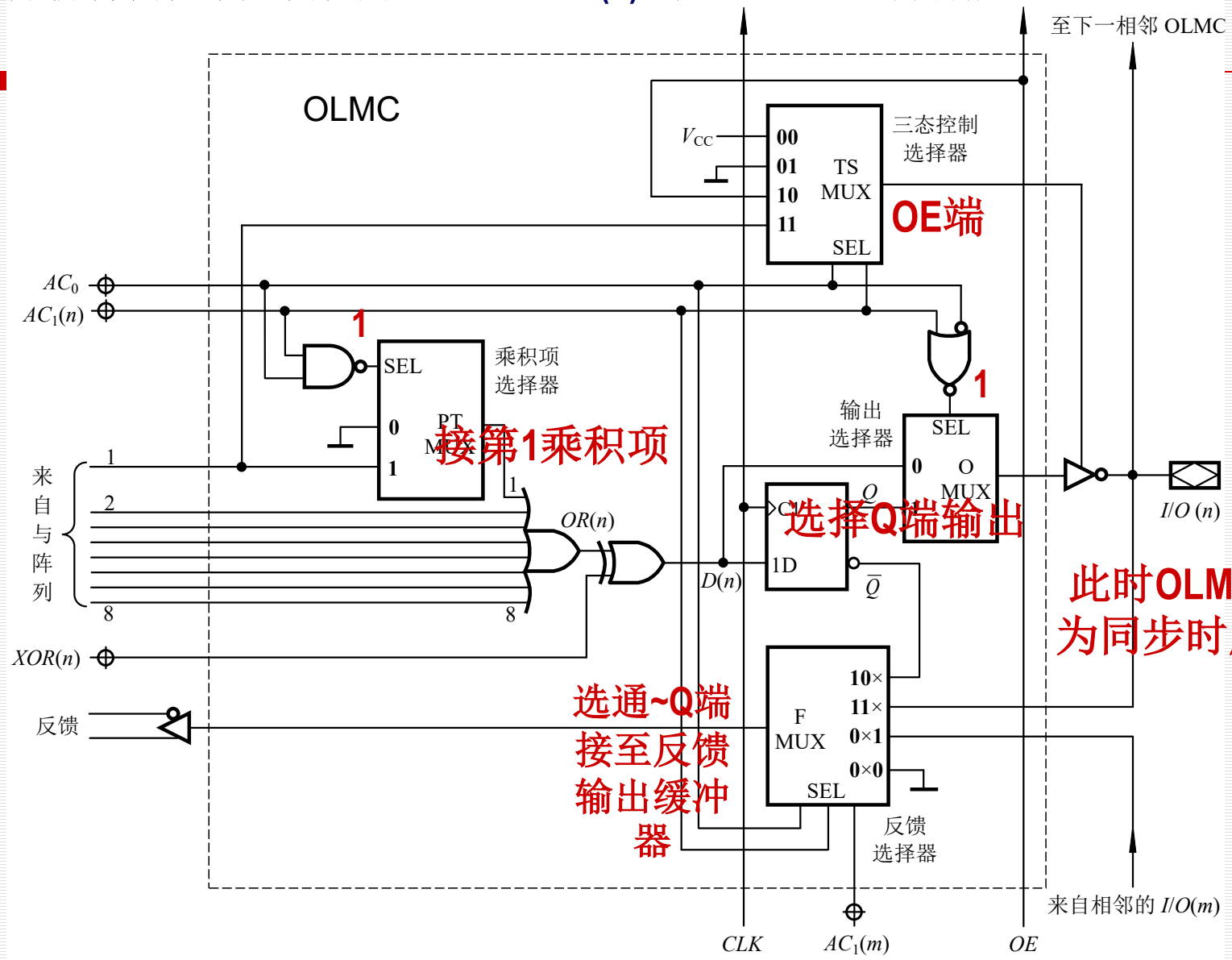

**FMUX：**

**根据AC0和AC1(n)和AC(m)的不同编码，便反向传输的电信号也对应不同，分别选择来自地（逻辑0）、相邻OLMC的输出、本级OLMC的输出或者本级D触发器的~Q送至与门阵列。**

# 例1：分析将编程单元编程为AC0=AC1(n)=1后的OLMC逻辑功能



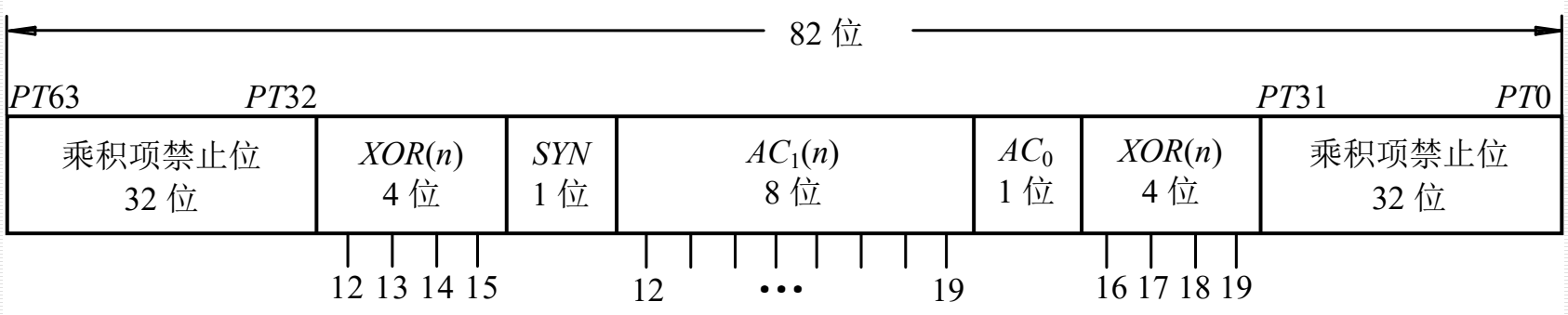
# 例2：分析将编程单元编程为AC0=1、AC1(n)=0后的OLMC逻辑功能



# 6.6.3 GAL的结构控制字

将编程单元组合在一起，构成GAL结构控制字，每个编程单元占一位。

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 。



# 8. CPLD和FPGA

8.1 复杂可编程逻辑器件(CPLD)简介

8.2 现场可编程门阵列(FPGA)

8.3 可编程逻辑器件开发过程简介简介

# 8.1　复杂可编程逻辑器件(CPLD)简介

1. 逻辑块

2. 可编程内部连线

3. I/O单元

# 8.1 复杂可编程逻辑器件(CPLD)简介

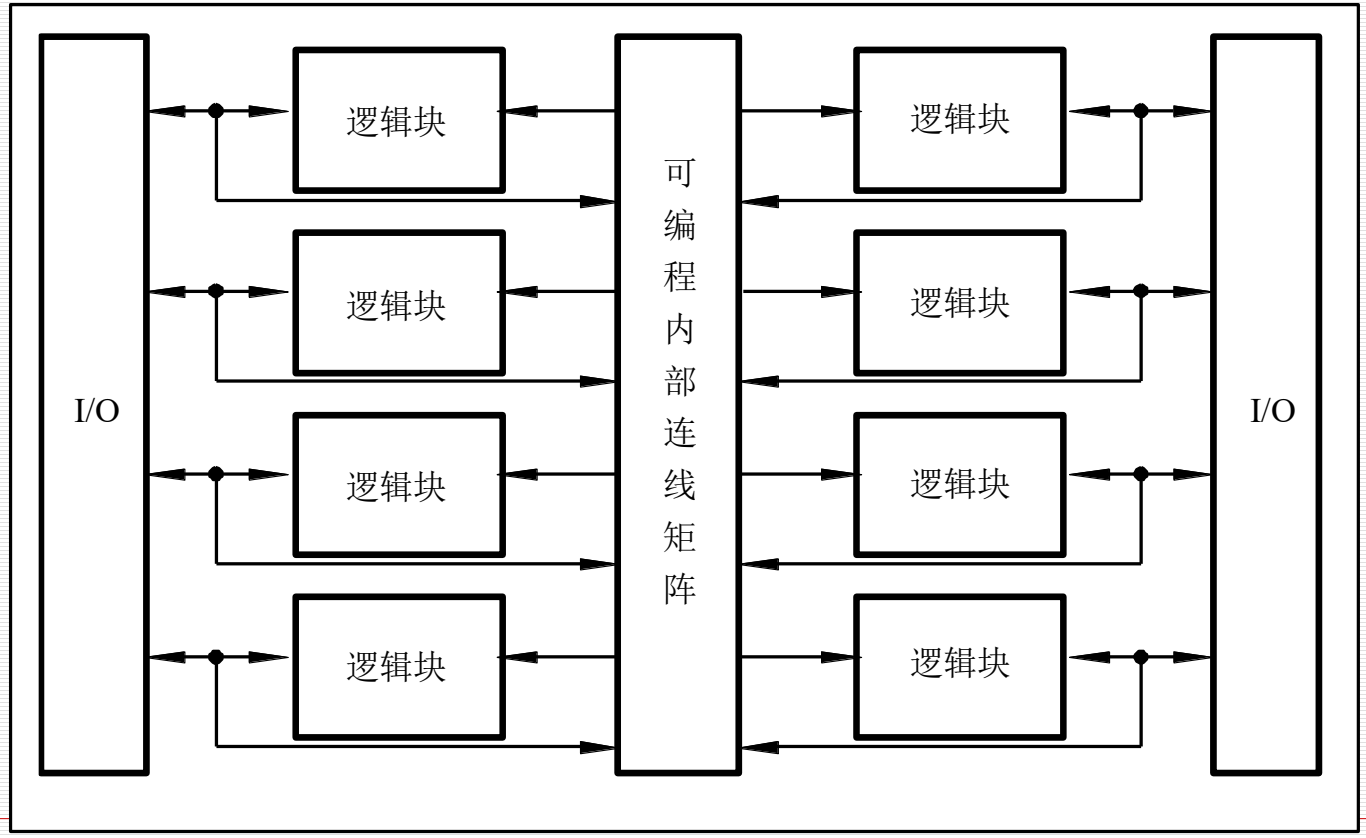
- 与PAL、GAL相比，CPLD的集成度更高，有更多的输入端、乘积项和更多的宏单元；

•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。

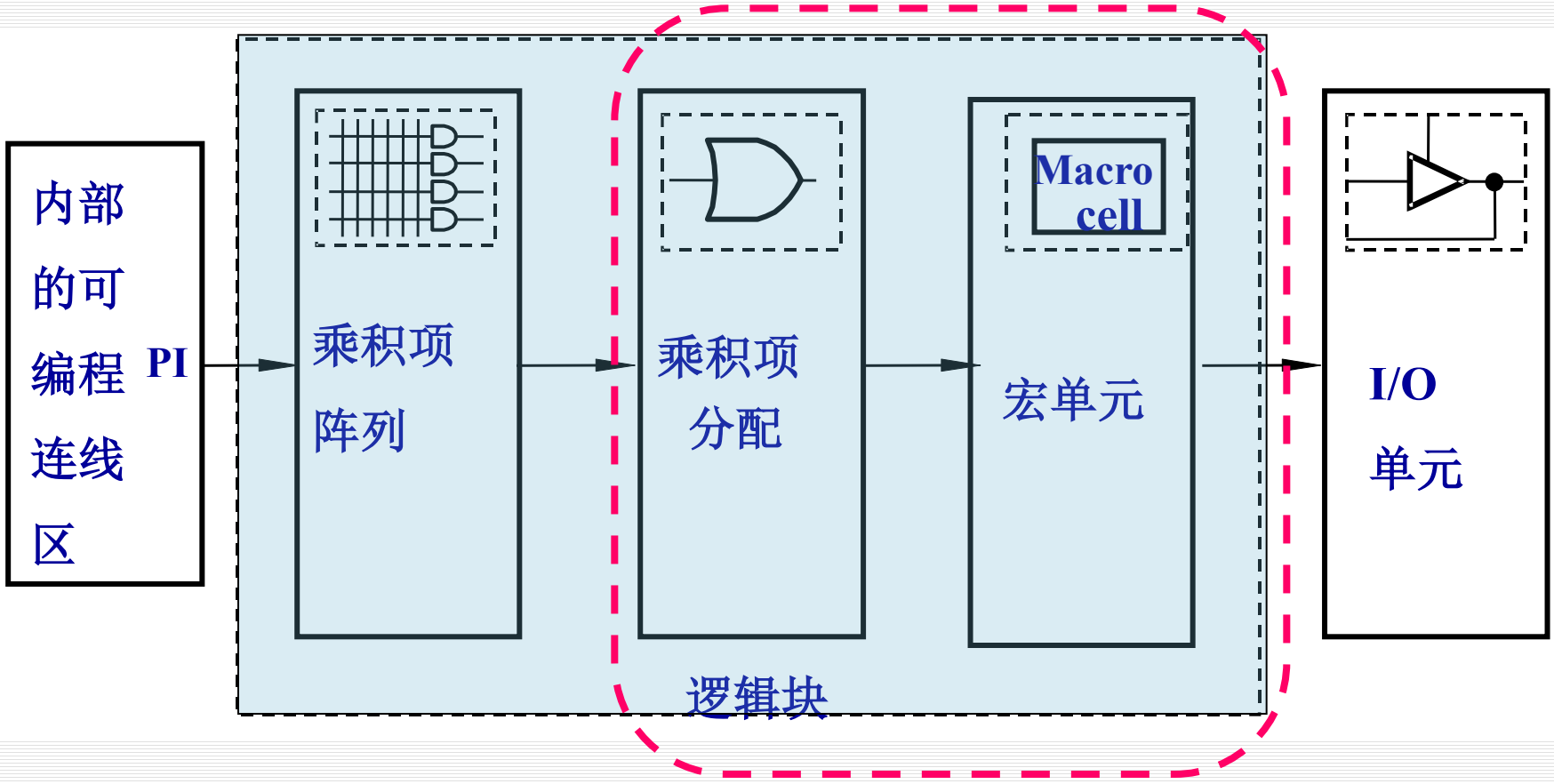
# 1. 逻辑块

**逻辑块是CPLD实现逻辑功能的核心模块。**



# （1）可编程乘积项阵列

## 通用的CPLD器件逻辑块的结构



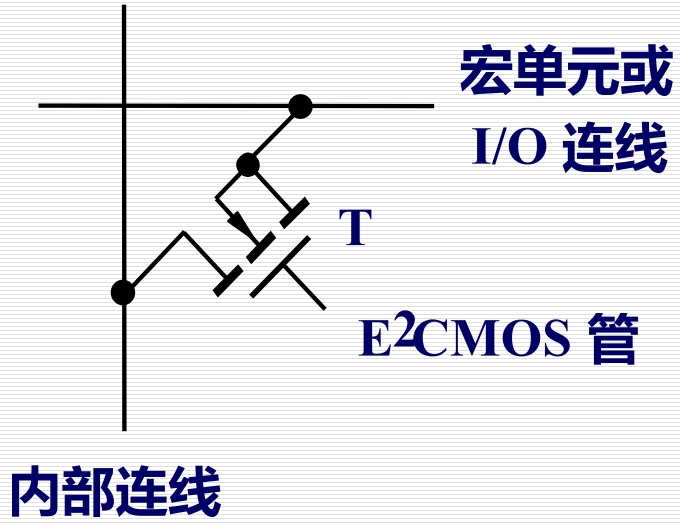

GAL中的乘积项是固定的，对应一个宏单元。但逻辑块中的乘积项可以编程，分配到不同的宏单元。灵活性大大提高。 CPLD中的宏单元与GAL中的类似。

# 2. 可编程内部连线

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。

连线区的可编程连接一般由$E^2CMOS$管实现。

当$E^2CMOS$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。

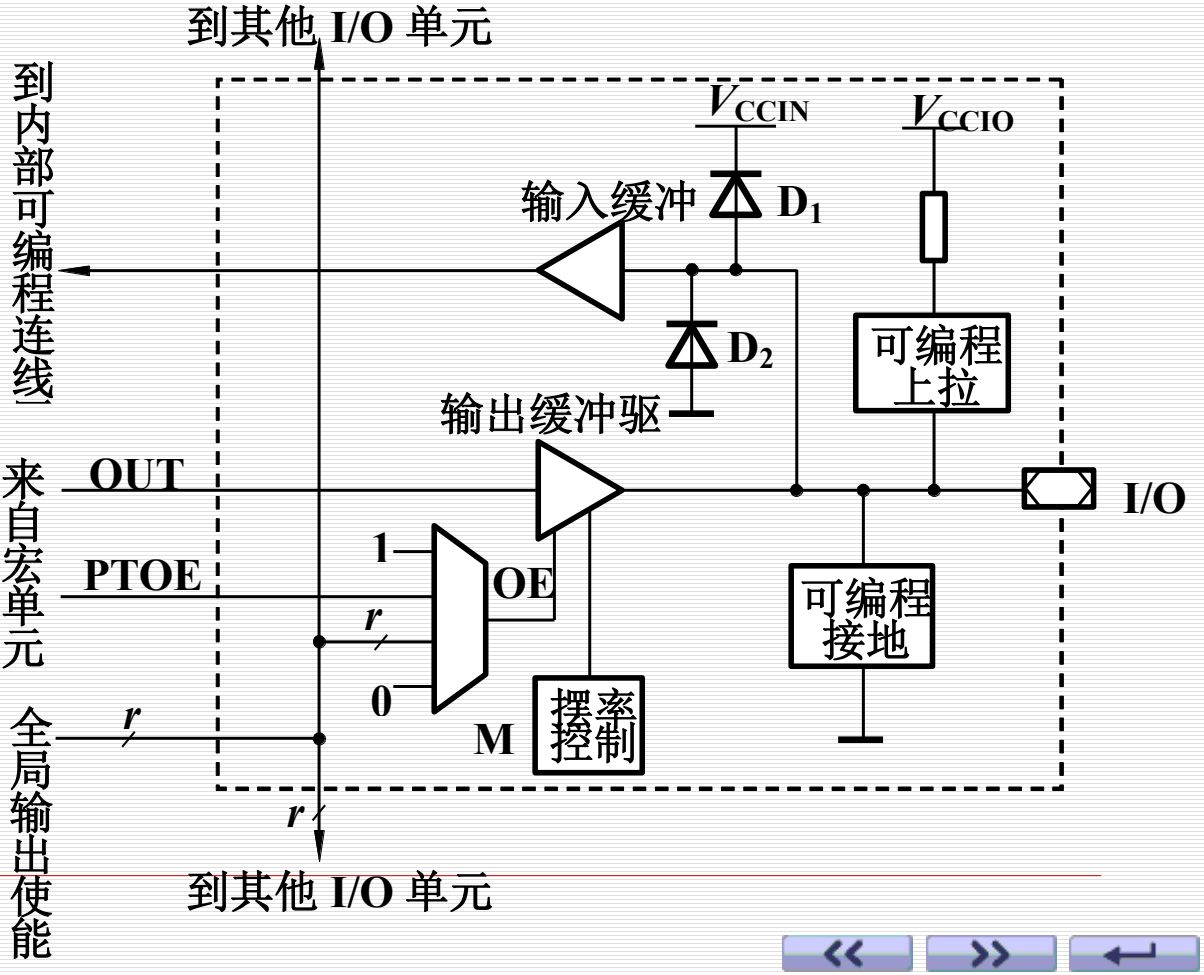



可编程连接原理图

# 3. I/O单元

I/O单元是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号。OE=1, I/O引脚为输出



# 8.2 现场可编程门阵列(FPGA)

## 8.2.1 FPGA实现逻辑功能的基本原理

## 8.2.2 FPGA结构简介

# 8.2 现场可编程门阵列(FPGA)

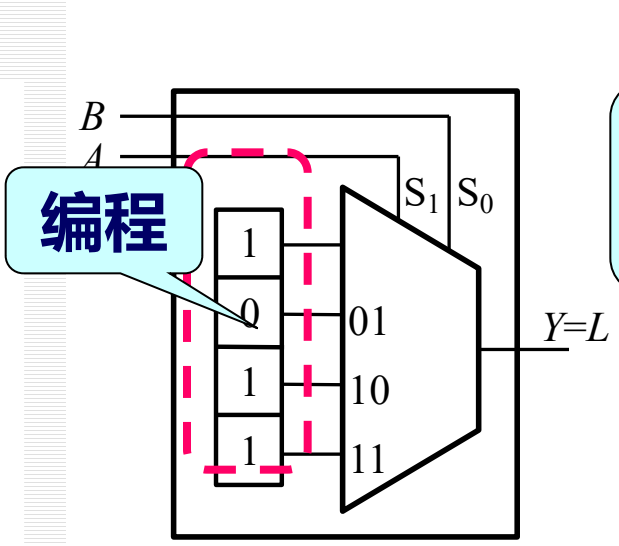
- CPLD用可编程“与-或”阵列实现逻辑函数。编程基于$E^2$PROM或快闪存储器。
- FPGA是用查找表(LUT)实现逻辑函数。复杂函数使用众多的LUT和触发器实现。编程基于SRAM。

# 8.2.1 FPGA实现逻辑功能的基本原理

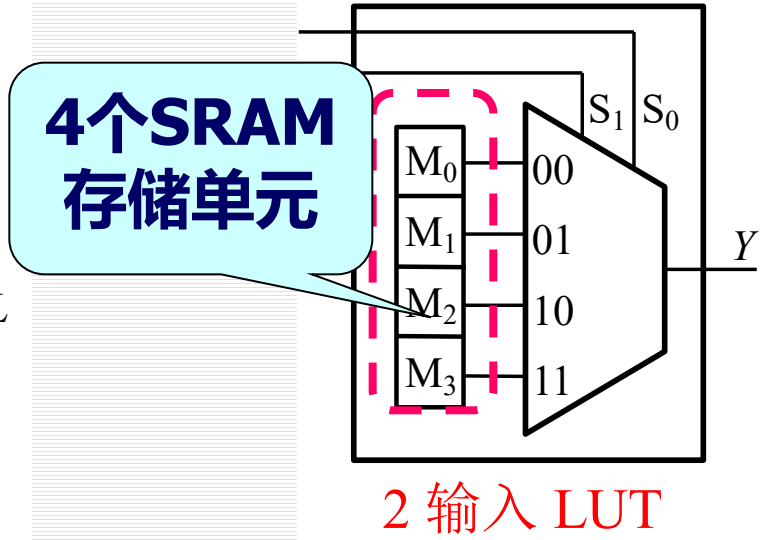
LUT是FPGA实现逻辑函数的基本单元。2输入LUT可实现任意2变量组合逻辑函数。

某函数 *L* 的真值表

| *A B* | *L* |
|---|---|
| 0 0 | 1 |
| 0 1 | 0 |
| 1 0 | 1 |
| 1 1 | 1 |




实现 *L* 的 LUT




2 输入 LUT

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。

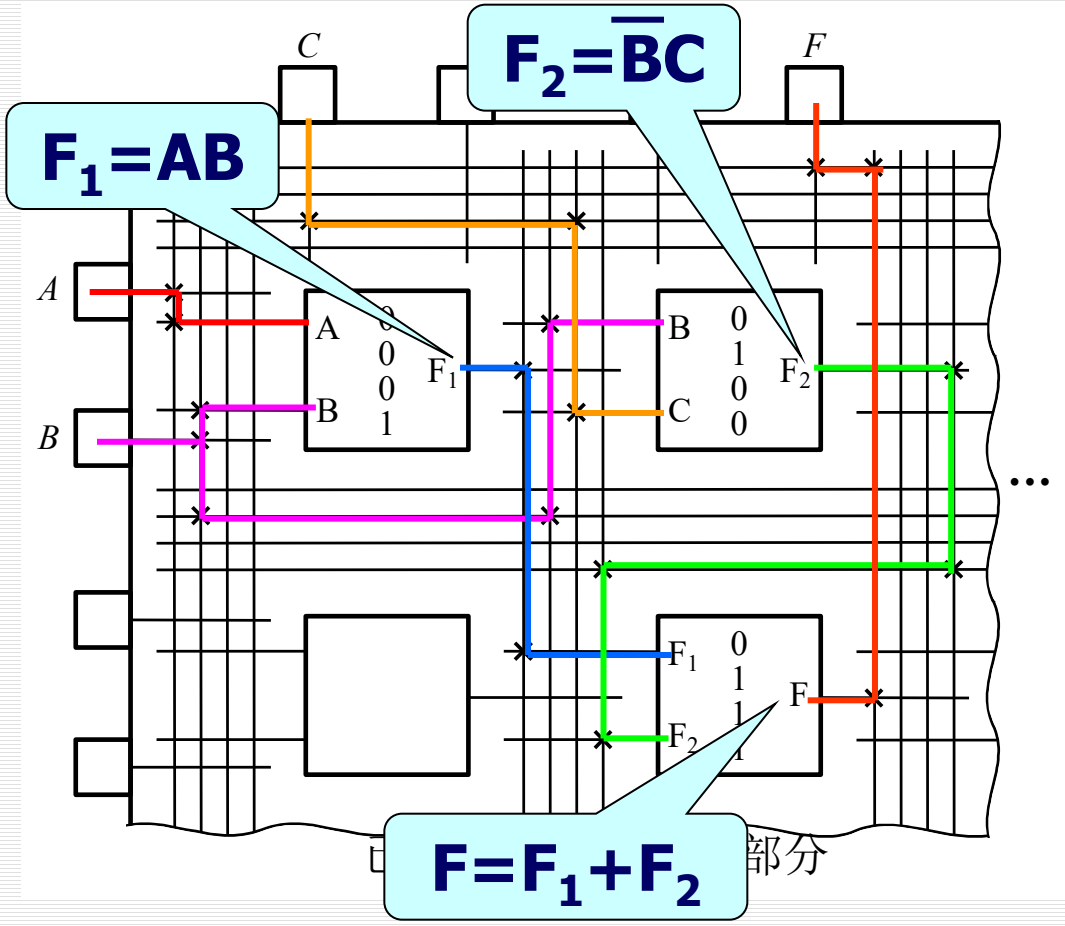
# LUT扩展--用2输入LUT实现函数 $F = AB + \overline{B}C = F_1 + F_2$

函数 $F$ 的真值表

| $A\ B$ | $F_1$ |
|---|---|
| 0 0 | 0 |
| 0 1 | 0 |
| 1 0 | 0 |
| 1 1 | 1 |

| $B\ C$ | $F_2$ |
|---|---|
| 0 0 | 0 |
| 0 1 | 1 |
| 1 0 | 0 |
| 1 1 | 0 |

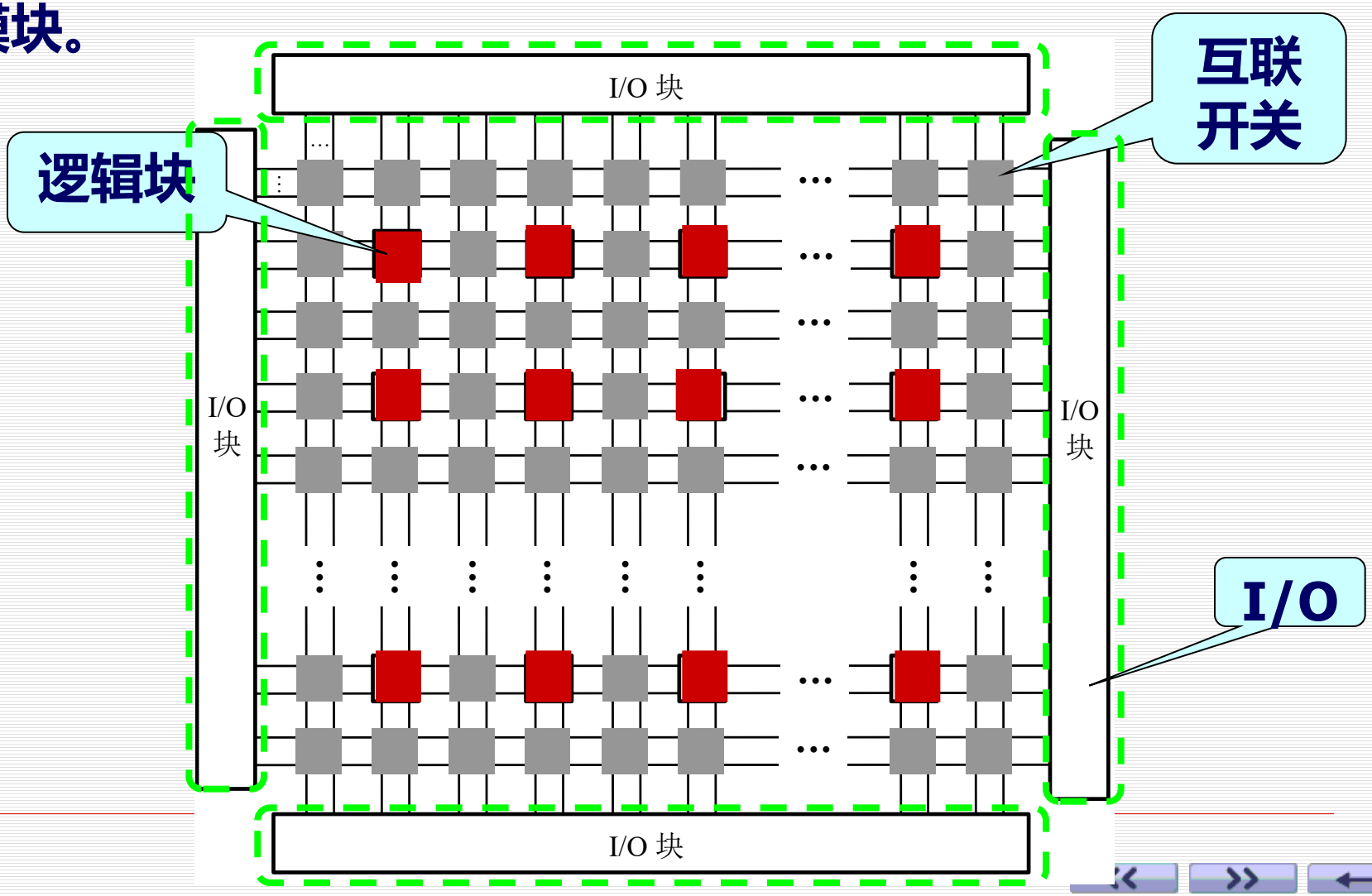
| $F_1\ F_2$ | $F$ |
|---|---|
| 0 0 | 0 |
| 0 1 | 1 |
| 1 0 | 1 |
| 1 1 | 1 |



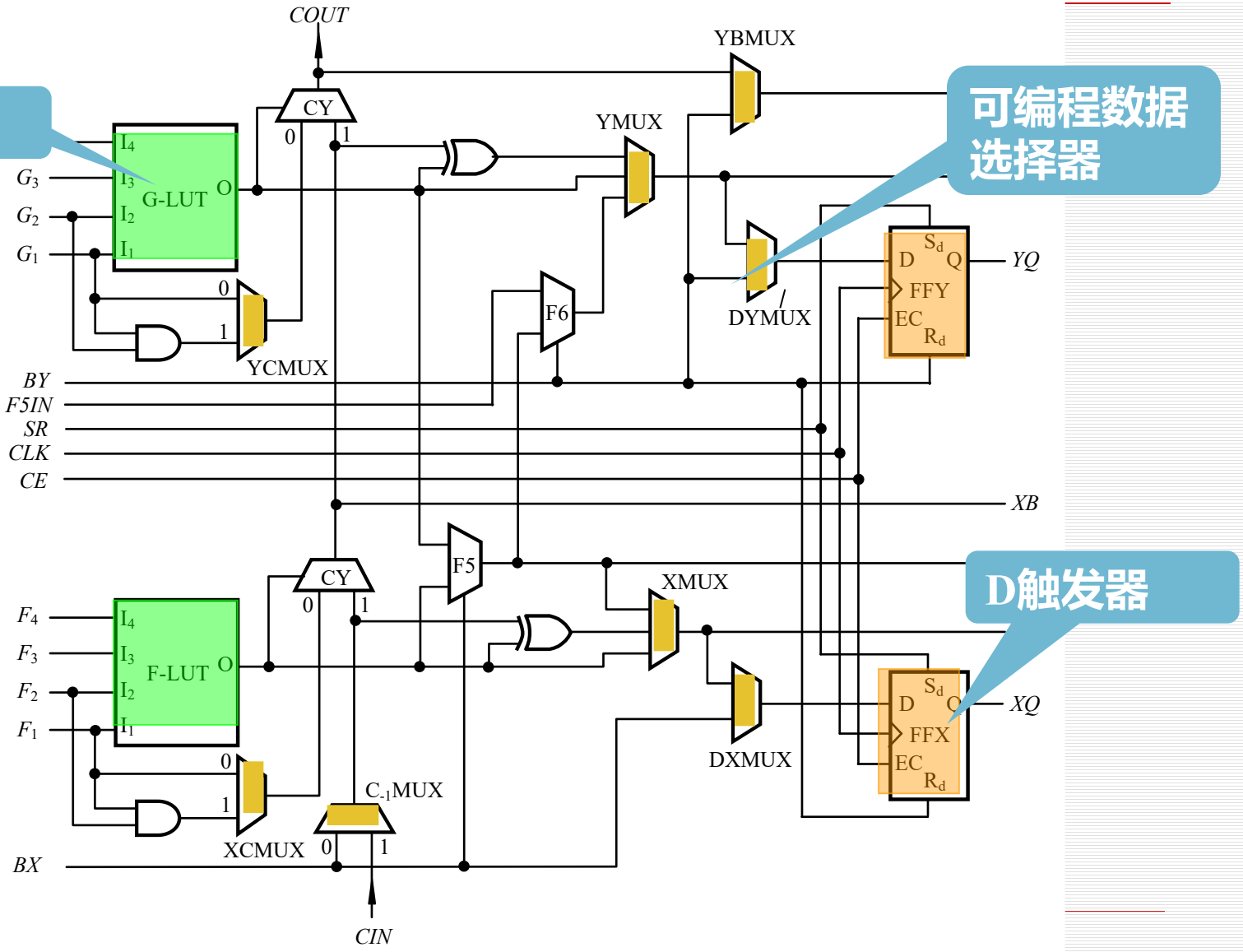

**在LUT的基础上增加触发器便可实现时序电路。**

# 8.2.2 FPGA结构简介

FPGA包括：可编程逻辑块、可编程互联开关、可编程I/O模块。



# 1. 可编程逻辑块


4输入LUT
可编程数据选择器
D触发器
COUT
YBMUX
CY
YMUX
G-LUT
$G_3$
$G_2$
$G_1$
YCMUX
F6
DYMUX
YQ
BY
F5IN
SR
CLK
CE
XB
F5
CY
XMUX
F-LUT
$F_4$
$F_3$
$F_2$
$F_1$
DXMUX
XQ
XCMUX
$C_{-1}$MUX
BX
CIN

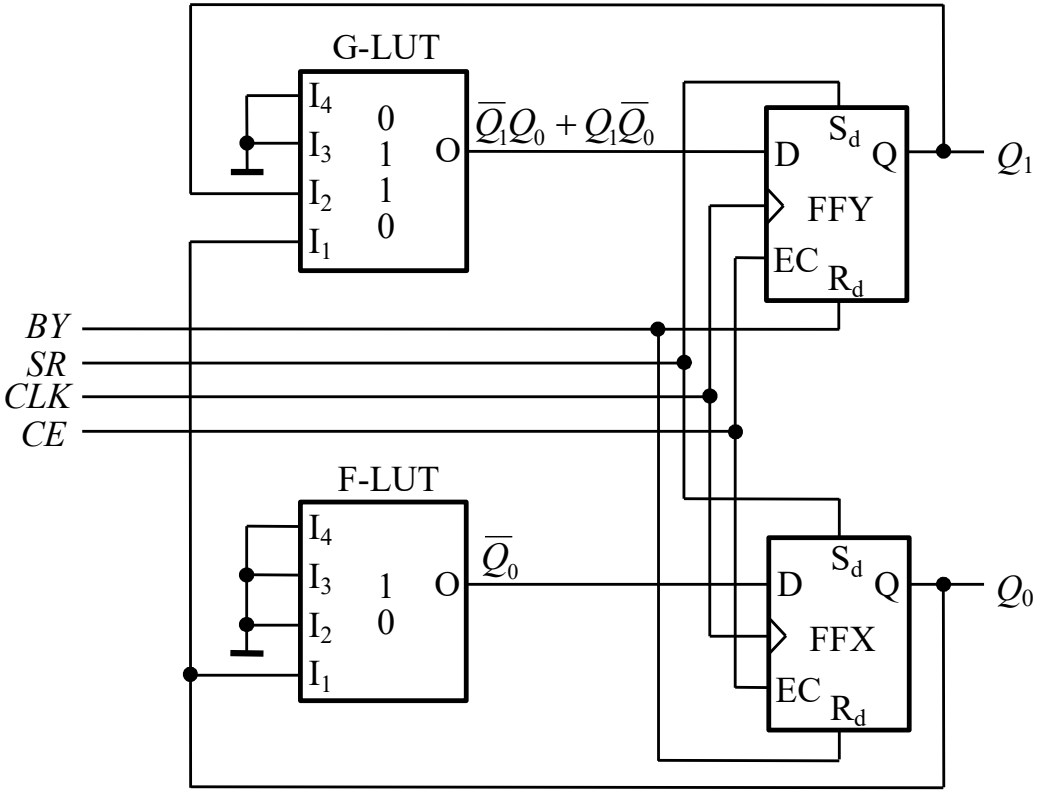
# 用可编程逻辑块实现2位二进制计数器。

## 2位二进制状态转换表

| $Q_1^n Q_0^n$ | $Q_1^{n+1}(D_1)\ Q_0^{n+1}(D_0)$ |
|---|---|
| 00 | 01 |
| 01 | 10 |
| 10 | 11 |
| 11 | 00 |

得 $D_1 = \overline{Q_1} Q_0 + Q_1 \overline{Q_0}$

$D_0 = \overline{Q_0}$

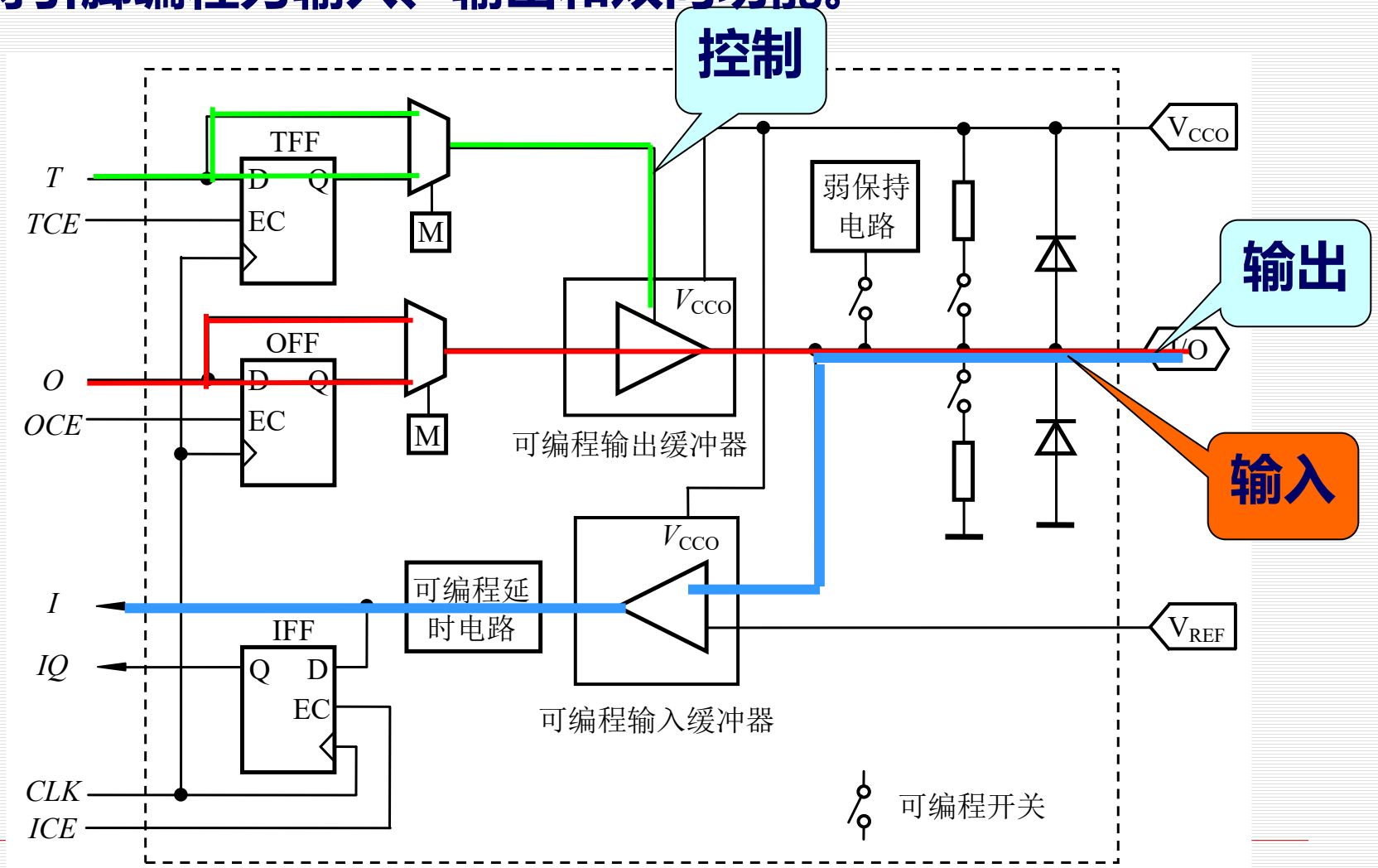



逻辑块编程实现 2 位二进制计数器

- 课堂练习：
  - 已知逻辑函数 $L = A\overline{B}\overline{C} + \overline{A}C$ ，若用2输入的LUT实现该逻辑函数，需要几个LUT？确定其中的内容。

# 2. I/O块

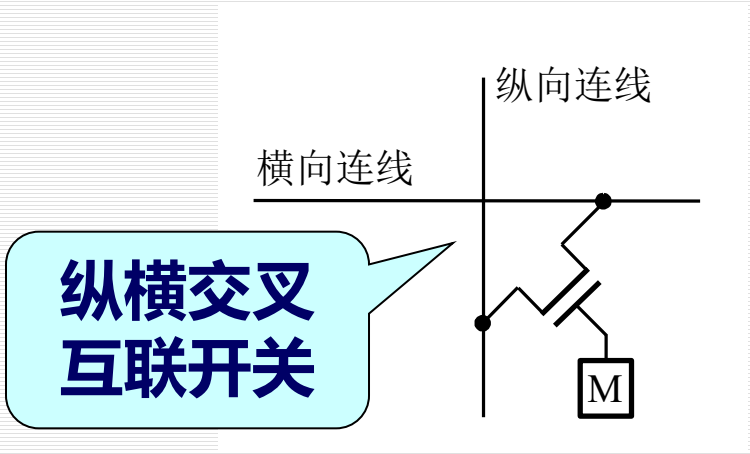
## 将引脚编程为输入、输出和双向功能。

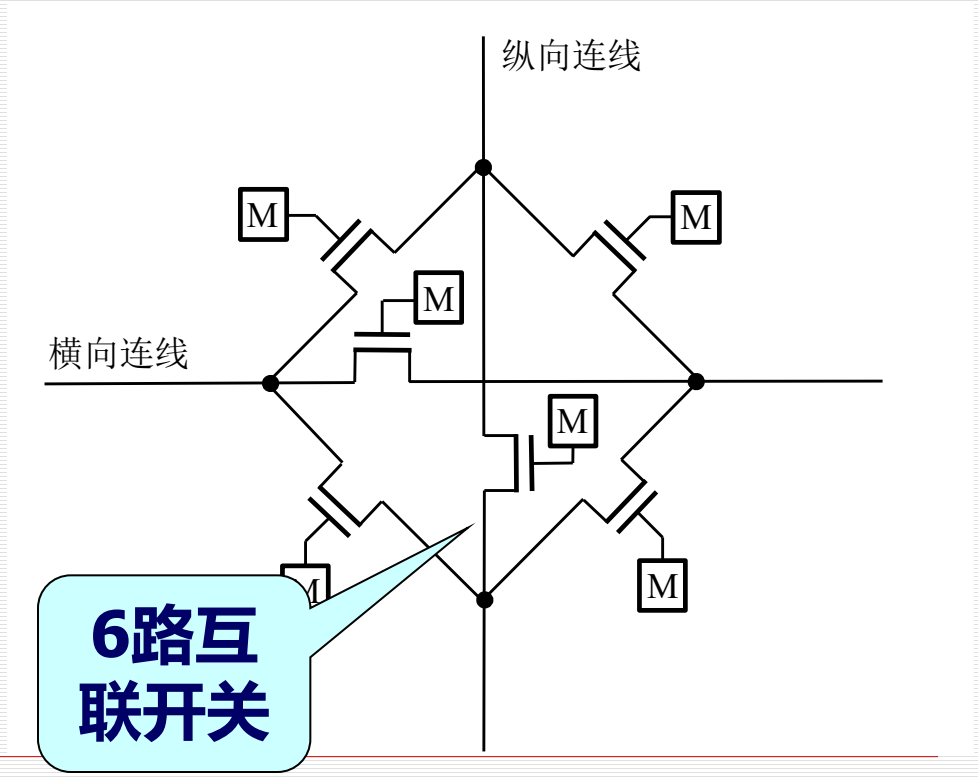


## 3. 可编程连线资源

可编程开关实现逻辑块与逻辑块之间，逻辑块与连线之间，逻辑块与I/O之间等的连接。

两种典型的互联开关结构如图。



# CPLD与FPGA的区别

FPGA采用SRAM进行功能配置，可重复编程，但系统掉电后，SRAM中的数据丢失。因此，需在FPGA外加EPROM，将配置数据写入其中，系统每次上电自动将数据引入SRAM中

| | CPLD | FPGA |
|---|---|---|
| 内部结构 | Product - term | Look - up Table |
| 程序存储 | 内部EEPROM | SRAM，外挂EEPROM |
| 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 |
| | - | EAB，锁相环 |
| | 可加密 | 一般不能保密 |

CPLD器件一般采用EEPROM存储技术，可重复编程，并且系统掉电后，EEPROM中的数据不会丢失，适于数据的保密

# FPGA与CPLD的区别

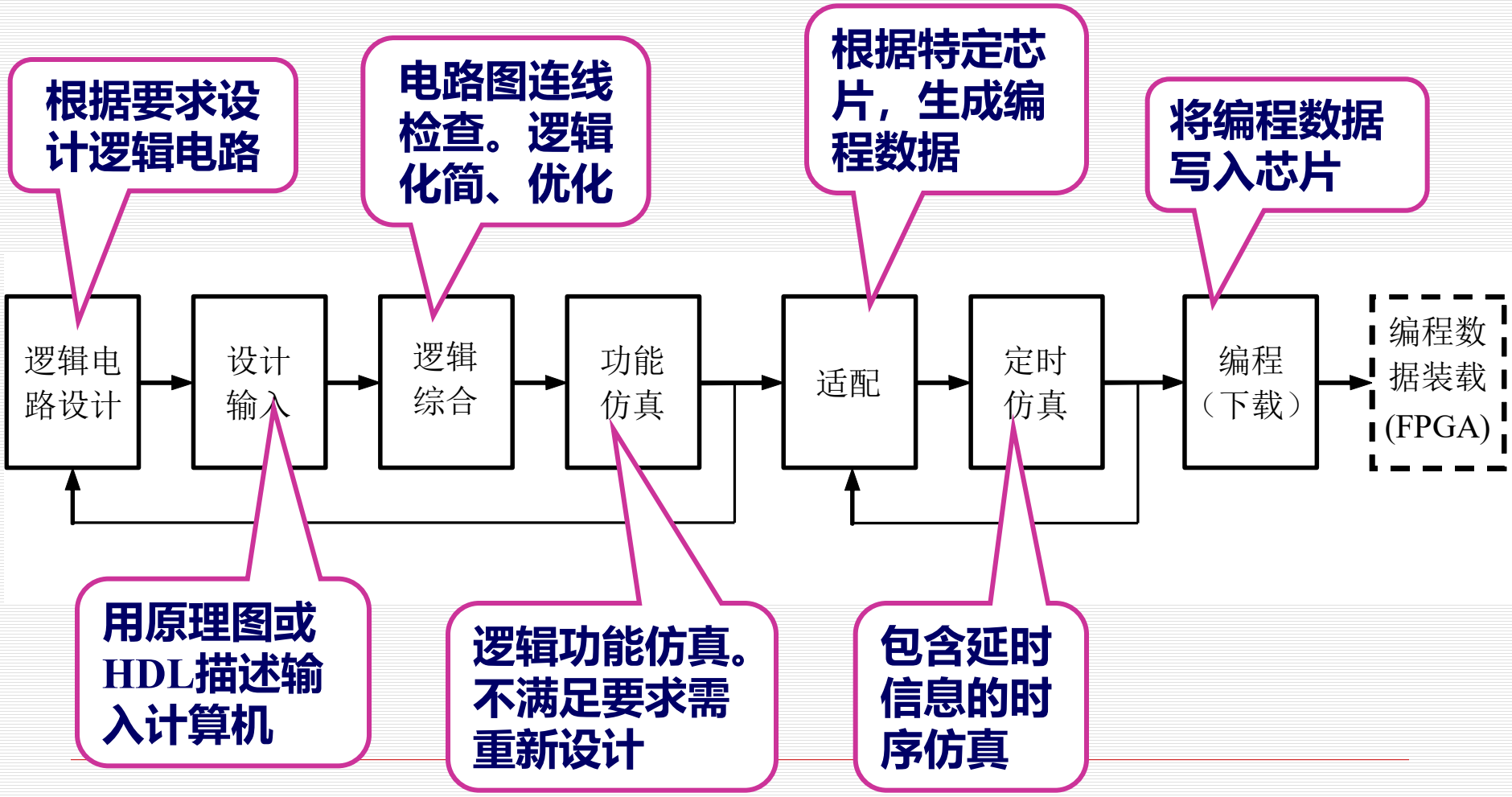
- FPGA为细粒度结构，CPLD为粗粒度结构。FPGA内部有丰富连线资源，CLB分块较小，芯片的利用率较高。CPLD的宏单元的与或阵列较大，通常不能完全被应用，且宏单元之间主要通过高速数据通道连接，其容量有限，限制了器件的灵活布线，因此CPLD利用率较FPGA器件低。

# FPGA与CPLD的区别

- FPGA为非连续式布线，CPLD为连续式布线。FPGA器件在每次编程时实现的逻辑功能一样，但走的路线不同，因此延时不易控制，要求开发软件允许工程师对关键的路线给予限制。CPLD每次布线路径一样，CPLD的连续式互连结构利用具有同样长度的一些金属线实现逻辑单元之间的互连。连续式互连结构消除了分段式互连结构在定时上的差异，并在逻辑单元之间提供快速且具有固定延时的通路。CPLD的延时较小。

# 8.3 可编程逻辑器件开发过程简介

## 可编程器件的一般开发过程



# 8.3 可编程逻辑器件开发过程简介

## 为什么FPGA需要编程数据装载?

- CPLD采用CMOS $E^2$PROM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。

# 编程条件

（1）微机；（2） CPLD编程软件；（3）专用编程电缆。

计算机根据用户编写的源程序运行开发系统软件，产生相应的编程数据和编程命令，通过五线编程电缆接口与芯片连接。

将电缆接到计算机的并行口，通过编程软件发出编程命令，将编程数据文件（*JED）中的数据转换成串行数据送入芯片。