



第八章

可编程逻辑器件





数字系统中常用的大规模集成电路可分为三大类。

- 非用户定制电路(Non custom design IC)
- 全用户定制电路(Full custom design IC)
- 半用户定制电路(Semicustom design IC)

目前在数字系统设计中广泛使用的可编程逻辑器件(Programmable Logic Device, 简称PLD) 属于LSI中的半用户定制电路。由于PLD具有结构灵活、性能优越、设计简单等特点, 因而在不同应用领域中受到广泛重视, 是构成数字系统的理想器件。



本章知识要点:

- ☆ PLD的基本概念
- ☆ 低密度可编程逻辑器件
- ☆ 复杂可编程逻辑器件
- ☆ 现场可编程门阵列
- ☆ 在系统编程技术简介





8.1 PLD 概述

PLD是20世纪70年代开始发展起来的一种新型大规模集成电路。一片**PLD**所容纳的逻辑门可达数百、数千甚至更多，其逻辑功能可由用户编程指定。

PLD特别适宜于构造小批量生产的系统，或在系统开发研制过程中使用。





8.1.1 PLD的发展

20世纪70年代初期：第一种PLD器件-----可编程只读存储器(PROM)问世。PROM由一个“与”阵列和一个“或”阵列组成，“与”阵列是固定的，“或”阵列是可编程的；

20世纪70年代中期：出现了可编程逻辑阵列(PLA)，PLA同样由一个“与”阵列和一个“或”阵列组成,但其“与”阵列和“或”阵列都是可编程的；

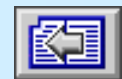
20世纪70年代末期：出现了可编程阵列逻辑(PAL)。**PAL**器件的“与”阵列是可编程的，而“或”阵列是固定的,它有多种输出和反馈结构，因而给逻辑设计带来了很大的灵活性。但PAL器件一般采用熔丝工艺，一旦编程后便不能改写。





20世纪80年代中期:先后出现了通用阵列逻辑(GAL)、复杂可编程逻辑器件(CPLD)和现场可编程门阵列(FPGA)等可编程器件。

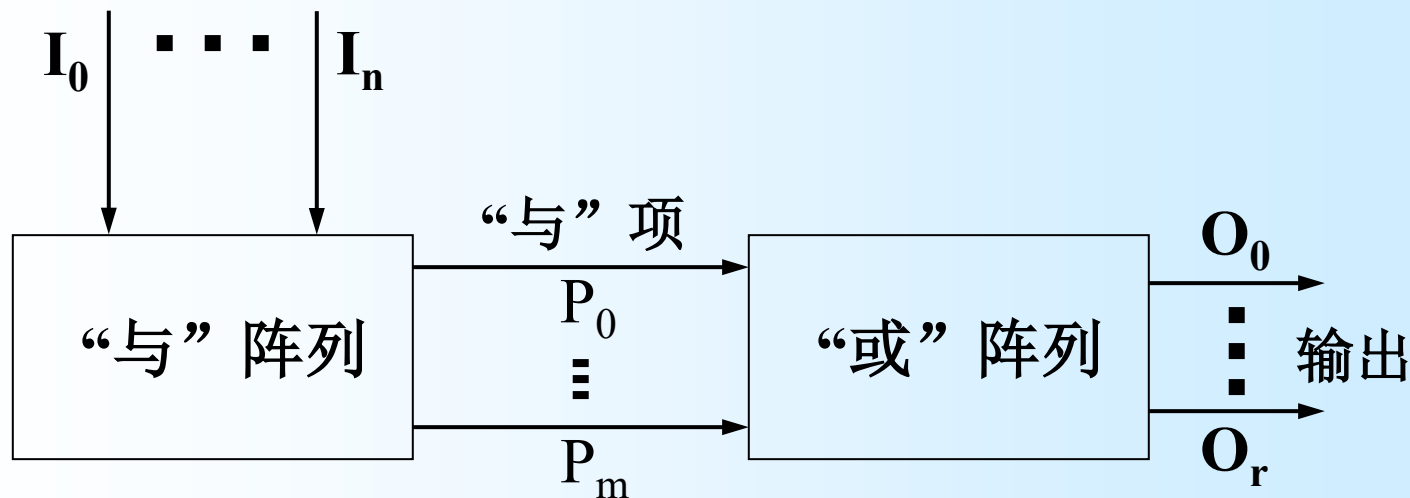
20世纪90年代:产生了在系统编程(ISP)器件。在系统编程是指用户具有在自己设计的目标系统中或线路板上为重构逻辑而对逻辑器件进行编程或反复改写的的能力。ISP器件为用户提供了传统的PLD技术无法达到的灵活性,使可编程逻辑技术发生了实质性飞跃。



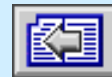


8.1.2 PLD的基本结构

PLD的基本组成为一个“与”阵列和一个“或”阵列。阵列中输入线和输出线的交点通过逻辑元件相连接。这些元件是接通还是断开，可由厂家根据器件的结构特征决定或由用户根据要求编程决定。基本结构如下图所示。



在基本结构的基础上，附加一些其他逻辑元件，如输入缓冲器、输出寄存器、内部反馈、输出宏单元等，便可构成各种不同的PLD。



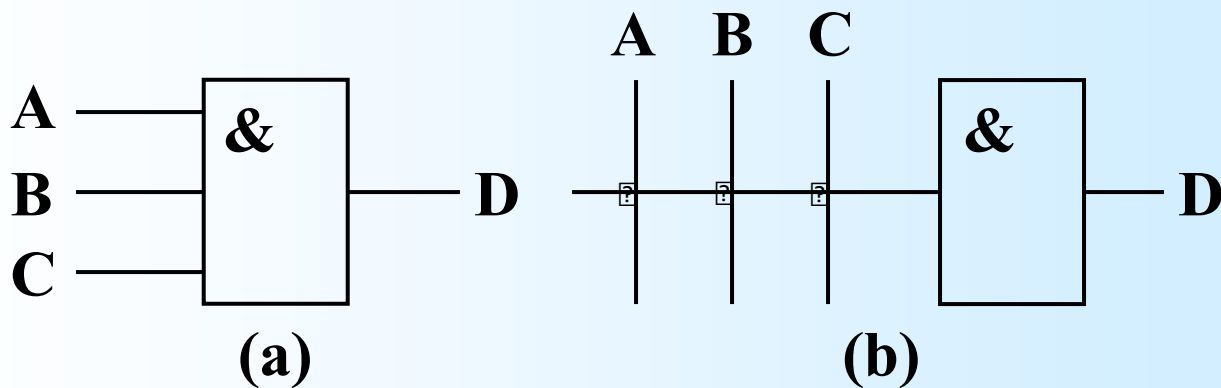


8.1.3 PLD的电路表示法

对于PLD器件，用逻辑电路的一般表示法很难描述其内部电路，这给PLD的生产和应用带来诸多不便。为此，对描述PLD基本结构的有关逻辑符号和规则作出了某些约定。

一、与门和或门

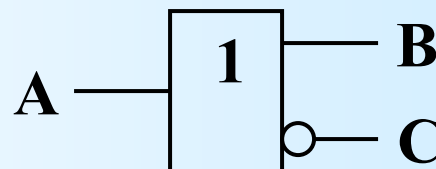
下图给出了3输入与门的两种表示法。传统表示法(图(a))中与门的3个输入A、B、C在PLD表示法(图(b))中称为3个输入项，而输出D称为“与”项。同样，或门也采用类似方法表示。





二、输入缓冲器

典型输入缓冲器的PLD表示法如右图所示。它的两个输出B、C是其输入A的原和反（见图中真值表）。

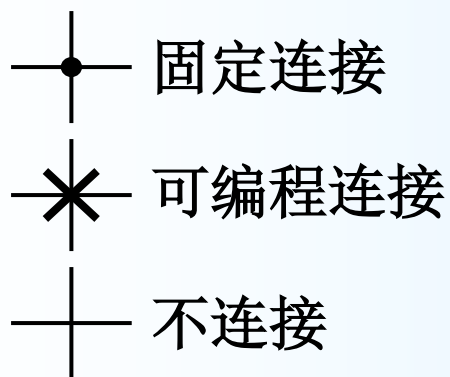


A	B	C
0	0	1
1	1	0

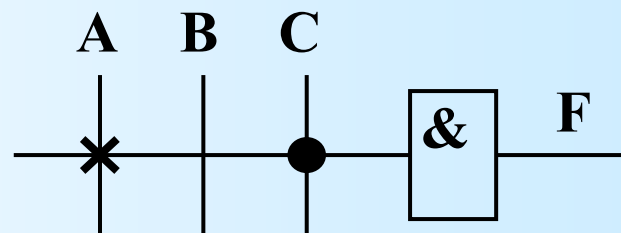
$$B = A, C = \bar{A}$$

三、连接方式

PLD阵列交叉点上的三种连接方式如图(a)所示。实点“.”表示硬线连接,即固定连接;“×”表示可编程连接;没有“×”和“.”的表示两线不连接。如图(b)中的输出 $F = A \cdot C$ 。



(a)

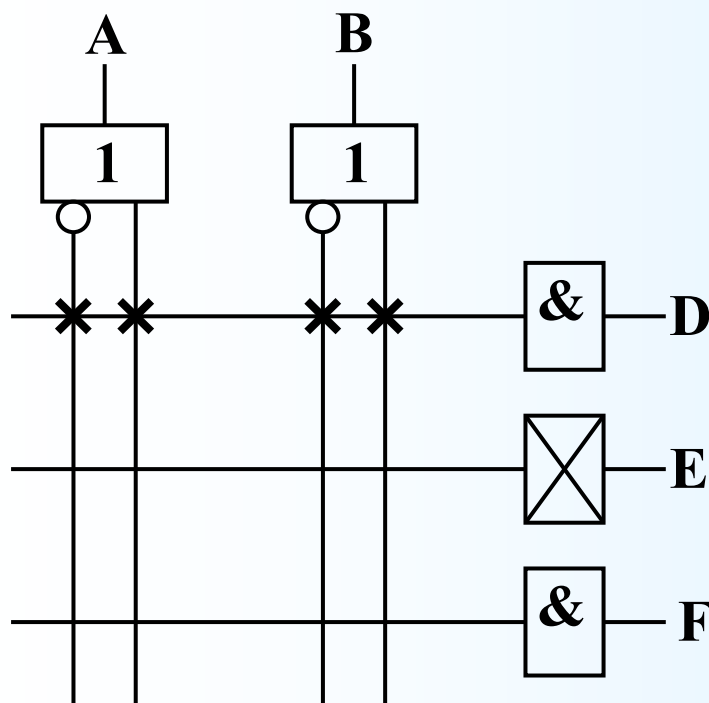


(b)





四、与门不执行任何功能时的连接表示



A	B	D	E	F
0	0	0	0	1
0	1	0	0	1
1	0	0	0	1
1	1	0	0	1

图中，输出为D的与门连接了所有的输入项,其输出方程为 $D = A \cdot \bar{A} \cdot B \cdot \bar{B} = 0$

方便起见，用标有“×”标记的与门输出来表示所有输入缓冲器输出全部连到某一“与”项的情况，如图中输出E。

与上述相反，图中输出F表示无任何输出项与其相连，因此，该“与”项总是处于“浮动”的逻辑“1”。





8.2 低密度可编程逻辑器件

根据PLD中阵列和输出结构的不同，目前常用的PLD有4种主要类型：

- 可编程只读存储器**PROM**
- 可编程逻辑阵列**PLA**
- 可编程阵列逻辑**PAL**
- 通用阵列逻辑**GAL**





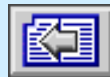
8.2.1 可编程只读存储器PROM

一、 半导体存储器的分类

存储器(Memory)是数字计算机和其他数字系统中存放信息的重要部件。随着大规模集成电路的发展, 半导体存储器因其具有集成度高、速度快、功耗小、价格低等优点而被广泛应用于各种数字系统中。

半导体存储器按功能可分为两大类。

- { 随机存取存储器RAM(Random Access Memory)
- { 只读存储器ROM(Read Only Memory)





1. 随机存取存储器RAM

RAM是一种既可读又可写的存储器，故又称为读写存储器。根据制造工艺的不同，RAM又可分为双极型和MOS型两种类型。

- 双极型RAM:** 工作速度高，但成本高、功耗大、集成度低，主要用作高速小容量存储器。
- MOS型RAM:** 功耗小、集成度高、成本低，但速度比双极型RAM低,适宜于构造大容量存储器。

MOS型RAM又可进一步分为静态RAM(SRAM)和动态RAM(DRAM)两种，相比之下DRAM的集成度更高。

RAM的优点是读写方便，使用灵活；缺点是一旦断电，所存储的信息便会丢失，它属于易失性存储器。





2. 只读存储器ROM

只读存储器ROM是一种在正常工作时只能读出、不能写入的存储器。通常用来存放那些固定不变的信息。只读存储器存入数据的过程通常称为编程。

根据编程方法的不同，可分为两类。

掩膜编程ROM（简称MROM）：存放的内容是由生产厂家在芯片制造时利用掩膜技术写入的。优点是可靠性高，集成度高，批量生产时价格便宜；缺点是用户不能重写或改写，不灵活。

用户可编程ROM（简称PROM）：存放的内容是由用户根据需要在编程设备上写入的。优点是使用灵活方便，适宜于用来实现各种逻辑功能。

只读存储器ROM属于非易失性存储器，即使切断电源，ROM中的信息也不会丢失，因而在数字系统中获得广泛应用。

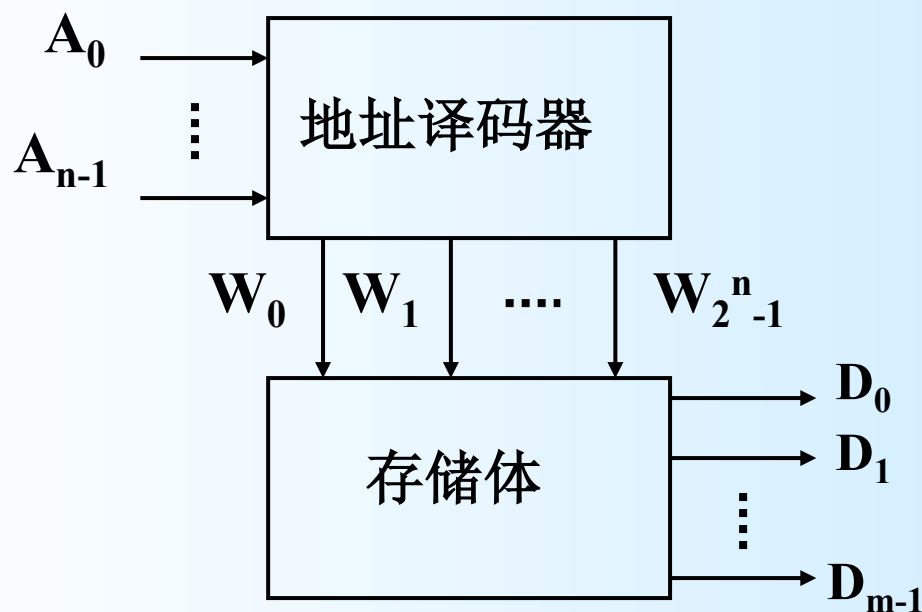


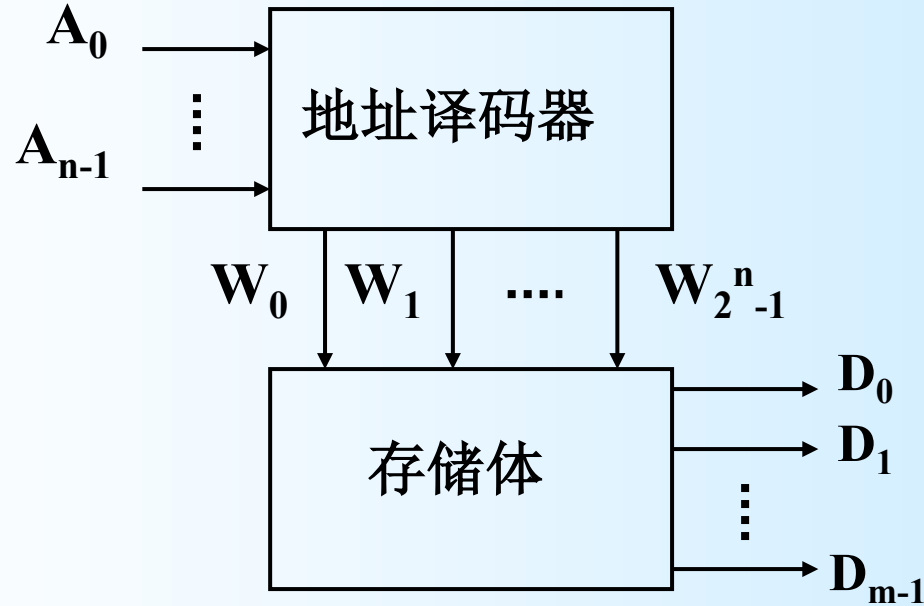


二、可编程ROM的结构与类型

1. PROM的逻辑结构

PROM的结构框图如下图所示，它主要由地址译码器和存储体两大部分组成。



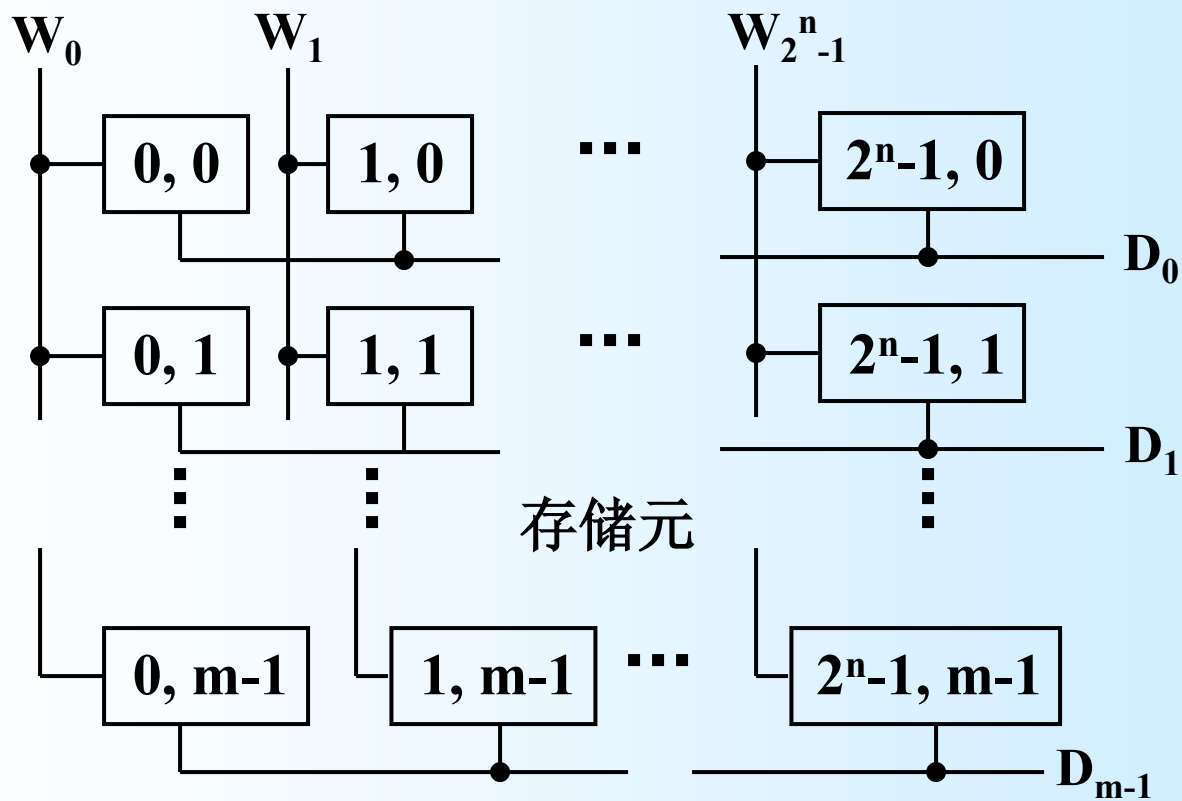


图中， $A_0 \sim A_{n-1}$ 为地址输入线； $W_0 \sim W_{2^n-1}$ 为地址译码输出线，又称为**字线**； $D_0 \sim D_{m-1}$ 为数据输出线，又称为**位线**。通常，将一个n位地址输入和m位数据输出的**PROM**的存储容量表示为 $2^n \times m$ (位)，意味着存储体中有 $2^n \times m$ 个存储元，每个存储元的状态代表一位二进制代码。



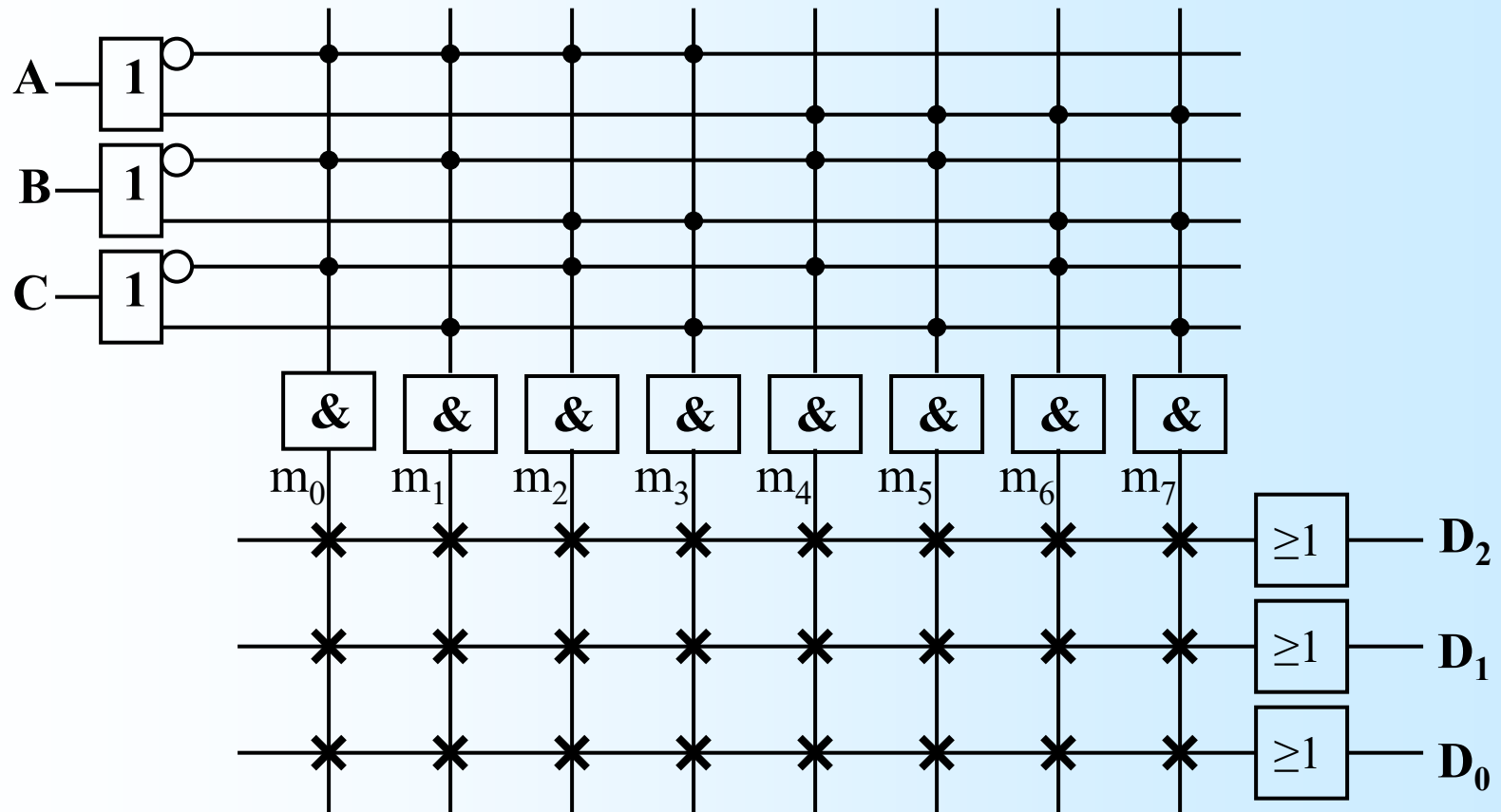


存储体的结构示意图如下：



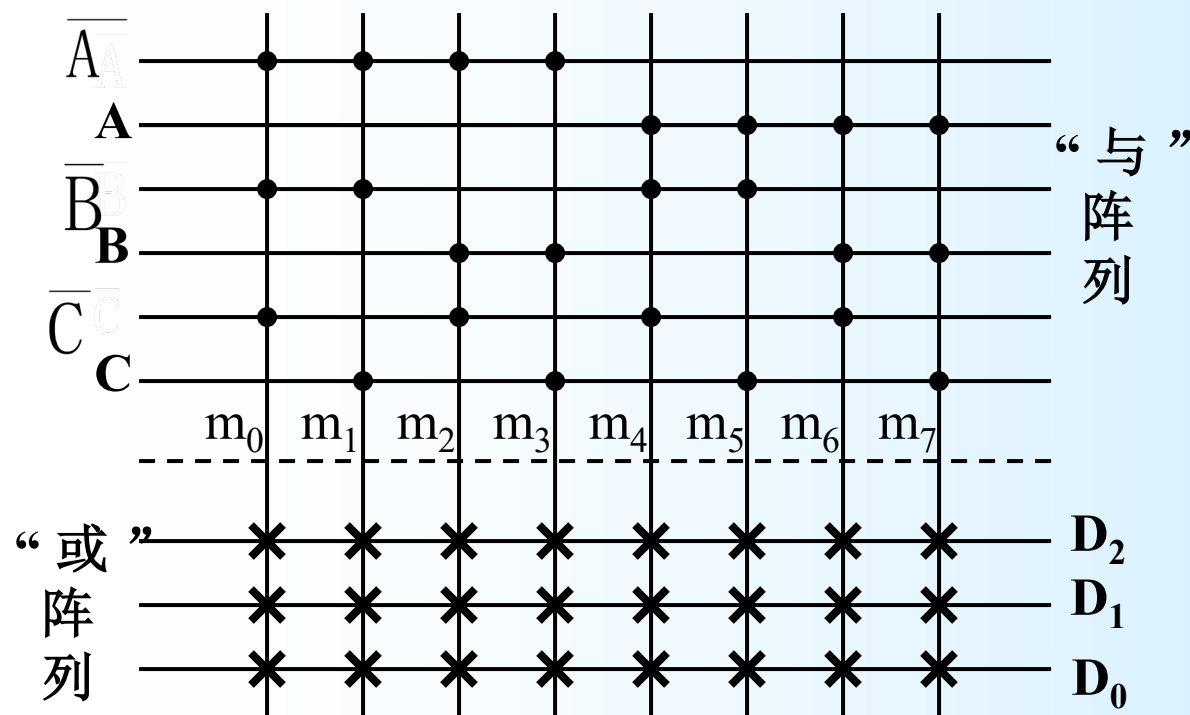


从逻辑器件的角度理解，PROM是由一个固定连接的与门阵列和一个可编程连接的或门阵列所构成的组合逻辑电路。例如，一个 8×3 (8与门 \times 3或门)PROM的逻辑结构图如下。





为了PROM设计的方便，通常将逻辑结构图简化为阵列逻辑图，简称阵列图。画阵列图时，将PROM中的每个与门和或门都简化成一根线。上图的阵列逻辑图如下图所示。



图中虚线上面6根水平线分别表示输入线 A 、 \bar{A} 、 B 、 \bar{B} 、 C 、 \bar{C} 。与门阵列的8根垂直线代表8个与门，或门阵列中标有 D_2 、 D_1 、 D_0 的3根水平线表示3个或门。





2. PROM的类型

根据存储元电路构造的不同，PROM有3种常用的类型。

(1) 一次编程的ROM(PROM)

所有存储元均被加工成同一状态“0”(或“1”), 用户可通过编程将某些存储元的状态改变成另一状态“1”(或“0”)。这种编程只能进行一次, 一旦编程完毕, 其内容便不能再改变。

PROM与固定ROM相比, 增加了灵活性。但因其可靠性较差, 加之只能一次性编程, 故目前很少使用。



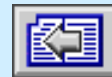


(2) 可抹可编程ROM(EPROM)

EPROM不仅可由用户编程存放指定的信息，而且可由用户通过专用的紫外线灯照射芯片上的受光窗口，将原存储内容抹去，再写入新的内容。这一特性是由EPROM中存储元的电路结构决定的。

(3) 电可抹可编程ROM(EEPROM)

EEPROM的结构与EPROM相似，但EEPROM在浮栅上增加了一个隧道二极管，编程时可使电荷通过它流向浮栅，而擦除时可使电荷通过它流走，即擦除和编程均用电完成。这种器件不仅工作电流小、擦除速度快，而且允许改写的次数大大高于EPROM，一般允许改写100次~1000次。目前，EPROM和EEPROM的应用均十分广泛。





三、PROM应用举例

由于PROM是由一个固定连接的“与”阵列和一个可编程连接的“或”阵列组成，所以，用户只要改变“或”阵列上连接点的数量和位置，就可以在输出端形成由输入变量“最小项之和”表示的任何一种逻辑函数。

采用PROM进行逻辑设计时，只需首先根据逻辑要求列出真值表，把真值表的输入作为PROM的输入，然后根据逻辑函数值确定对PROM“或”阵列进行编程的代码，画出相应的阵列图即可。





例 用PROM设计一个代码转换电路，将4位二进制码转换为Gray码。

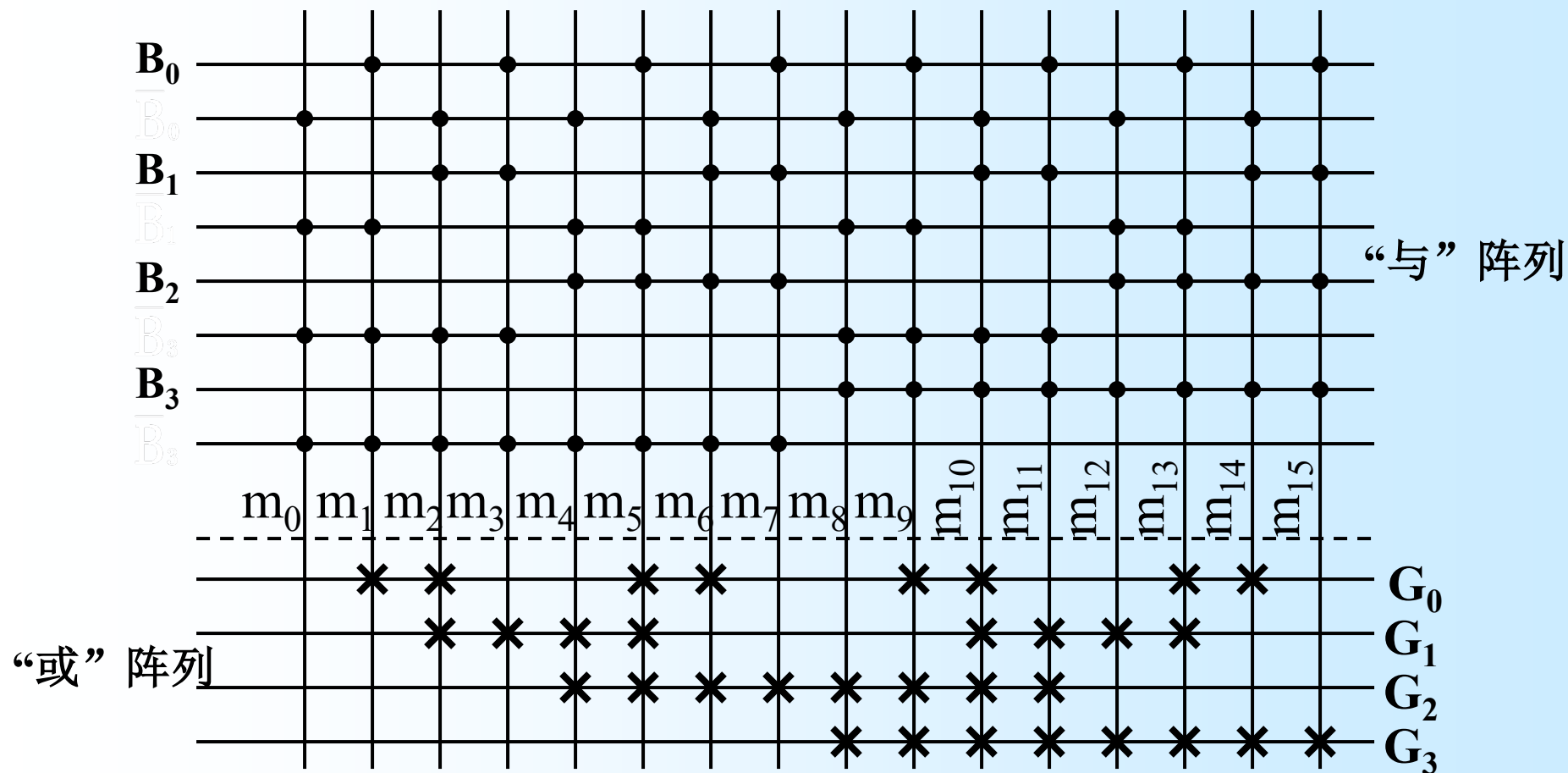
解 设4位二进制码为 B_3 、 B_2 、 B_1 、 B_0 ，4位Gray码为 G_3 、 G_2 、 G_1 、 G_0 ，其真值表如下表所示。

二进制码				Gray码				二进制码				Gray码			
B_3	B_2	B_1	B_0	G_3	G_2	G_1	G_0	B_3	B_2	B_1	B_0	G_3	G_2	G_1	G_0
0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0
0	0	0	1	0	0	0	1	1	0	0	1	1	1	0	1
0	0	1	0	0	0	1	1	1	0	1	0	1	1	1	1
0	0	1	1	0	0	1	0	1	0	1	1	1	1	1	0
0	1	0	0	0	1	1	0	1	1	0	0	1	0	1	0
0	1	0	1	0	1	1	1	1	1	0	1	1	0	1	1
0	1	1	0	0	1	0	1	1	1	1	0	1	0	0	1
0	1	1	1	0	1	0	0	1	1	1	1	1	0	0	0





将4位二进制码作为PROM的输入，Gray码作为PROM的输出，可选容量为 $2^4 \times 4$ 的PROM实现给定功能。根据真值表可画出该电路的阵列图如下图所示。





8.2.2 可编程逻辑阵列PLA

PLA是一种“与”阵列和“或”阵列均可编程的逻辑器件，即可编程逻辑阵列PLA(Programmable Logic Array)。

PLA可分为组合PLA和时序PLA两种类型。

一、组合PLA

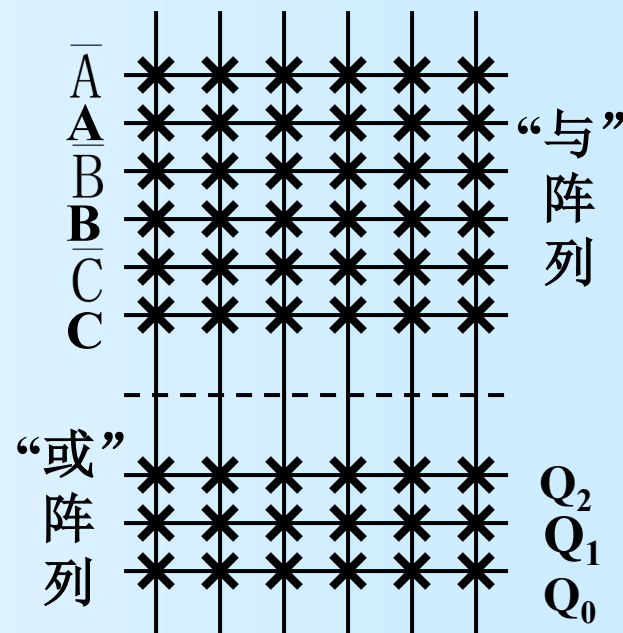
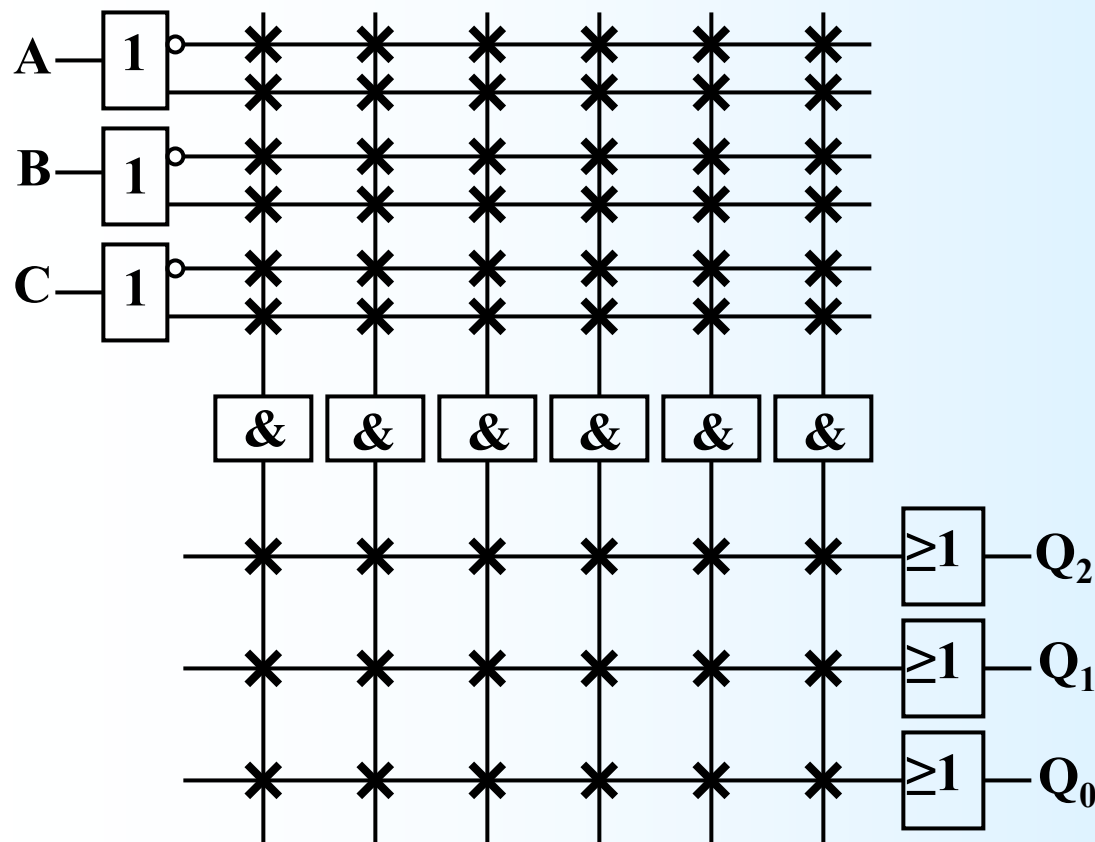
1. 组合PLA的逻辑结构

逻辑结构:由一个“与”阵列和一个“或”阵列构成,“与”阵列和“或”阵列都是可编程的。





一个具有3个输入变量、可提供6个“与”项、产生3个输出函数的PLA逻辑结构图及其相应阵列图如下图所示。





PLA的存储容量用输入变量数(n)、与项数(p)、输出端数(m)来表示。

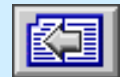
前面所示PLA的容量为3—6—3。目前常见的有容量为16—48—8和14—96—8等PLA器件。

2. 应用举例

采用PLA进行逻辑设计，可以十分有效地实现各种逻辑功能。相对PROM而言，PLA更灵活、更经济、结构更简单。

用PLA设计组合逻辑电路时，一般分为两步：

- 将给定问题的逻辑函数按多输出逻辑函数的化简方法简化成最简“与-或”表达式；
- 根据最简表达式中的不同“与项”以及各函数最简“与-或”表达式确定“与”阵列和“或”阵列，并画出阵列逻辑图。

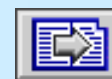




例 用PLA设计一个代码转换电路，将一位十进制数的8421码转换成余3码。

解 设ABCD-----表示8421码，WXYZ-----表示余3码，可列出转换电路的真值表如下表所示。

A	B	C	D	W	X	Y	Z	A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	1	1	1	0	0	0	1	0	1	1
0	0	0	1	0	1	0	0	1	0	0	1	1	1	0	0
0	0	1	0	0	1	0	1	1	0	1	0	d	d	d	d
0	0	1	1	0	1	1	0	1	0	1	1	d	d	d	d
0	1	0	0	0	1	1	1	1	1	0	0	d	d	d	d
0	1	0	1	1	0	0	0	1	1	0	1	d	d	d	d
0	1	1	0	1	0	0	1	1	1	1	0	d	d	d	d
0	1	1	1	1	0	1	0	1	1	1	1	d	d	d	d





根据真值表写出函数表达式，并按照多输出函数化简法则用卡诺图进行化简，可得到最简“与-或”表达式如下：

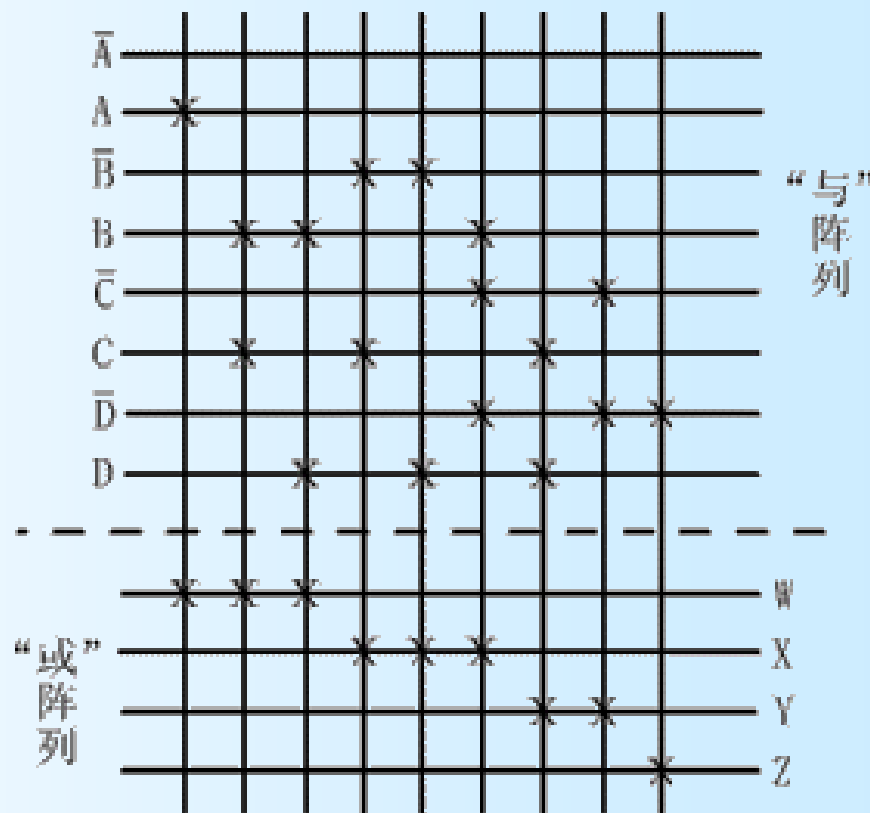
$$W = A + BC + BD$$

$$X = \bar{B}C + \bar{B}D + B\bar{C}\bar{D}$$

$$Y = CD + \bar{C}\bar{D}$$

$$Z = \bar{D}$$

由此可见，全部输出函数只包含9个不同“与”项，所以，该代码转换电路可用一个容量为4—9—4的PLA实现，其阵列图如图所示。

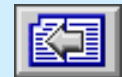
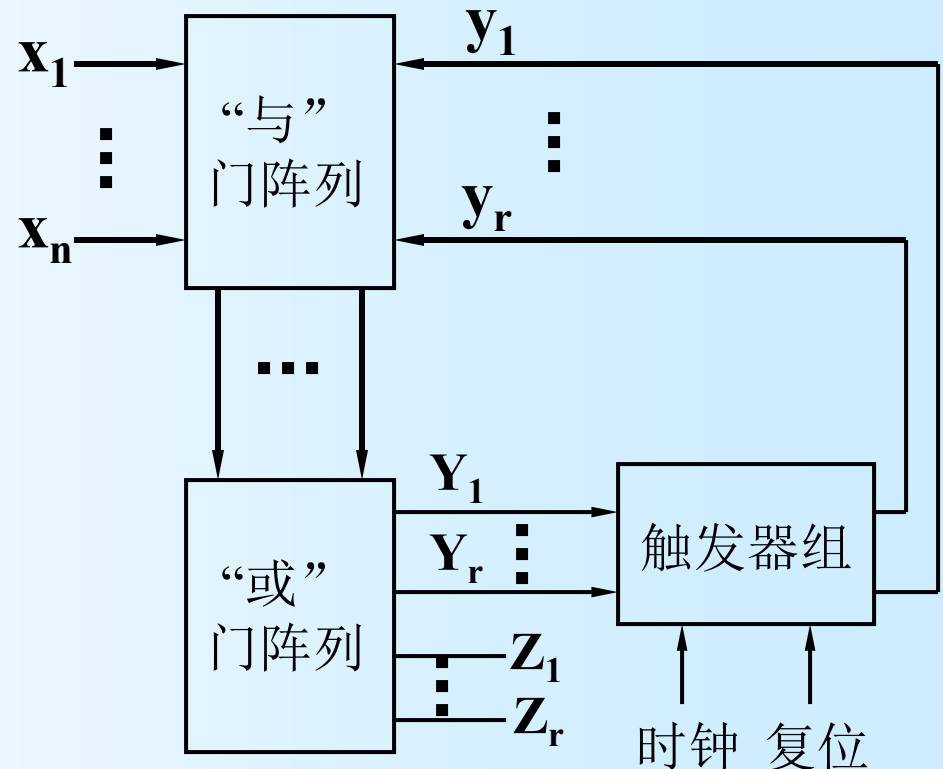




二、时序PLA

逻辑结构：由“与”阵列、“或”阵列和一个用于存储过去输入状态的触发器网络构成。

触发器网络中包含若干触发器，它们的输入接受“或”阵列输出及时钟脉冲、复位信号的控制，其输出反馈到“与”阵列，用来和当前输入一起产生“与”项输出。时序PLA的结构框图如右图所示。



例 8-4 分析图 8.6 所示时序 PLA 的阵列图,说明该电路功能。

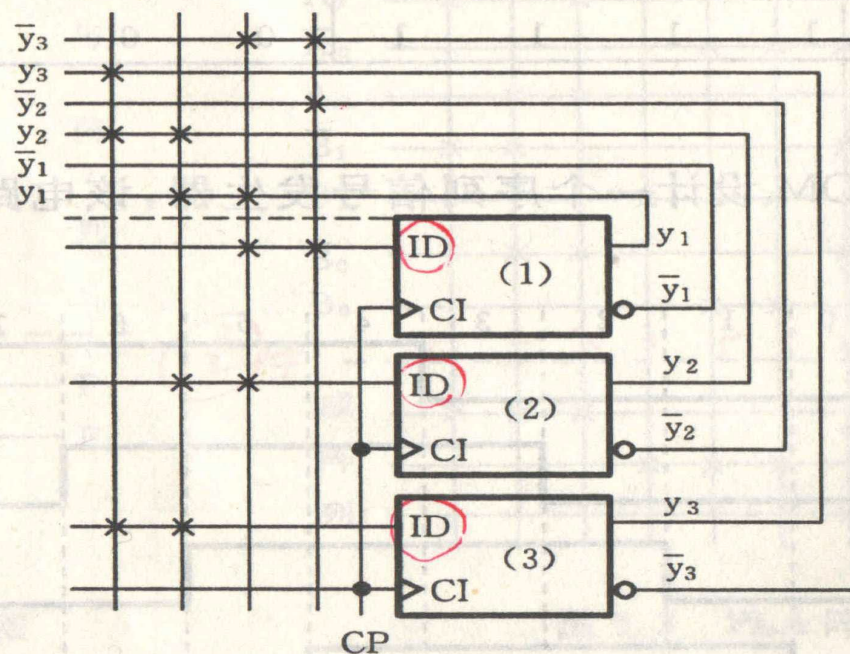


图 8.6 PLA 阵列图

解 由图 8.6 可知,该电路是 1 个由 3 个 D 触发器作为存储元件的同步时序逻辑电路。电路的激励函数表达式如下:

$$D_3 = y_3 y_2 + y_2 y_1$$

$$D_2 = \bar{y}_3 y_1 + y_2 y_1$$

$$D_1 = \bar{y}_3 \bar{y}_2 + \bar{y}_3 y_1$$

$$Q^{n+1} = D$$

根据激励函数表达式和 D 触发器的功能表可作出该电路的状态表如表 8.2 所示,状态图如图 8.7 所示。

由状态图可知,该电路是一个具有自启动功能的同步模 6 计数器。

表 8.2 状态表

现 态			次 态		
y_3	y_2	y_1	y_3^{n+1}	y_2^{n+1}	y_1^{n+1}
0	0	0	0	0	1
0	0	1	0	1	1
0	1	0	0	0	0
0	1	1	1	1	1
1	0	0	0	0	0
1	0	1	0	0	0
1	1	0	1	0	0
1	1	1	1	1	0

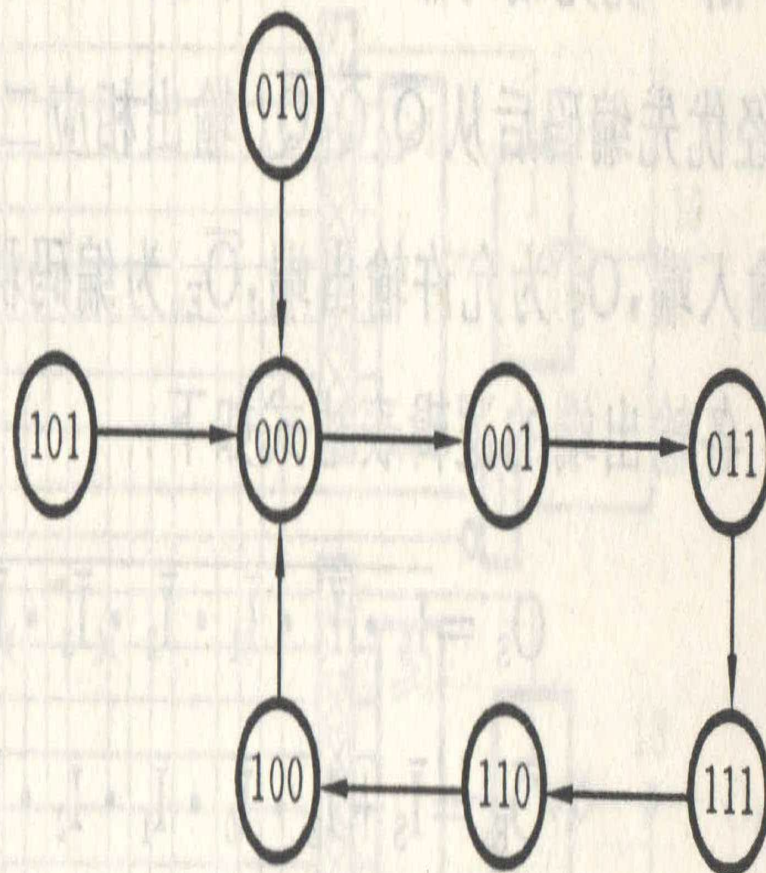


图 8.7 状态图

例 8-5 某同步时序逻辑电路有一个输入 x 和一个输出 Z , 其二进制状态表如表 8.3 所示。试用 PLA 和 T 触发器实现该电路功能。

表 8.3 状态表

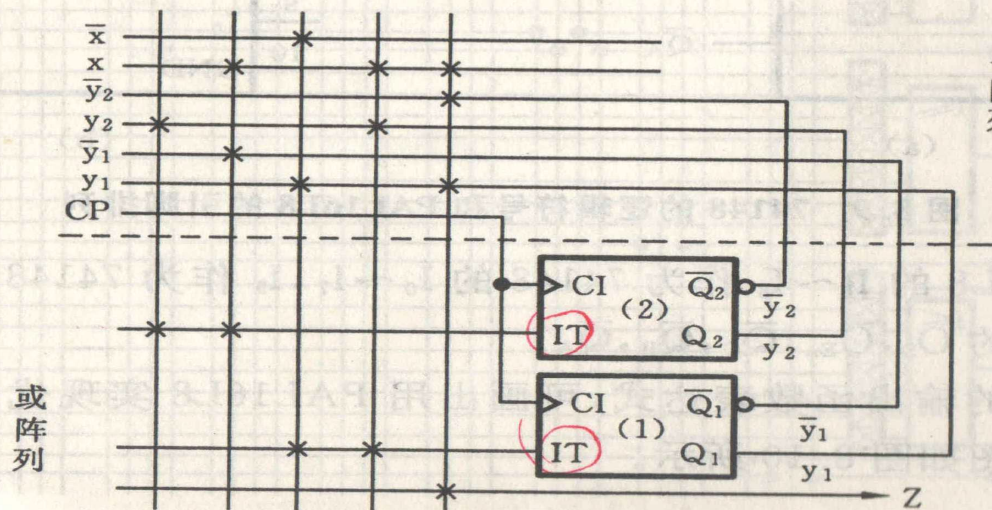
现 态		次态 $y_2^{n+1} y_1^{n+1}$ / 输出 Z	
y_2	y_1	$x=0$ T_2 T_1	$x=1$ T_2 T_1
0	0	00/0	10/0
0	1	00/0	01/1
1	0	00/0	01/0
1	1	00/0	00/0

$Q^{n+1} = T \oplus Q$

解 根据表 8.3 所示二进制状态表和 T 触发器激励表, 可求出该电路的激励函数和输出函数表达式:

$$T_2 = x \bar{y}_1 + y_2 \quad T_1 = \bar{x} y_1 + x y_2 \quad Z = x \bar{y}_2 y_1$$

据此, 可画出用 PLA 和 T 触发器实现该电路功能的阵列逻辑如图 8.8 所示。



与阵列

y_2	y_1	x	T_2	T_1
0	0	0	0	0
0	1	0	0	1
1	0	0	0	0
1	1	0	0	0
0	0	1	1	0
0	1	1	0	1
1	0	1	0	0
1	1	1	0	0

图 8.8 阵列逻辑图

$y_1(x\bar{y}_1 + y_1\bar{x} + y_1x)$

$T_2 = y_2\bar{y}_1\bar{x} + y_2y_1\bar{x} + \bar{y}_2\bar{y}_1x + y_2y_1x$



8.2.3 可编程阵列逻辑PAL

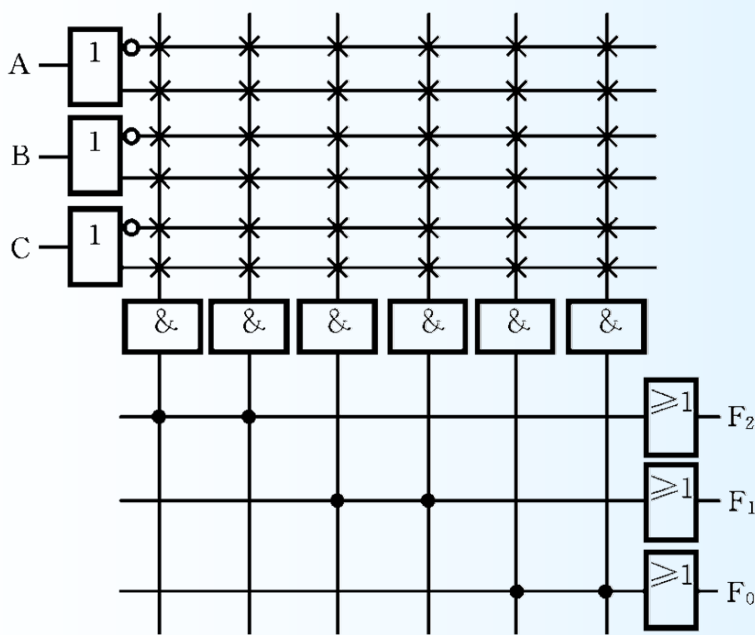
PAL (Programmable Array Logic) 是在 PROM 和 PLA 的基础上发展起来的一种可编程逻辑器件。它相对于 PROM 而言，使用更灵活，且易于完成多种逻辑功能，同时又比 PLA 工艺简单，易于实现。



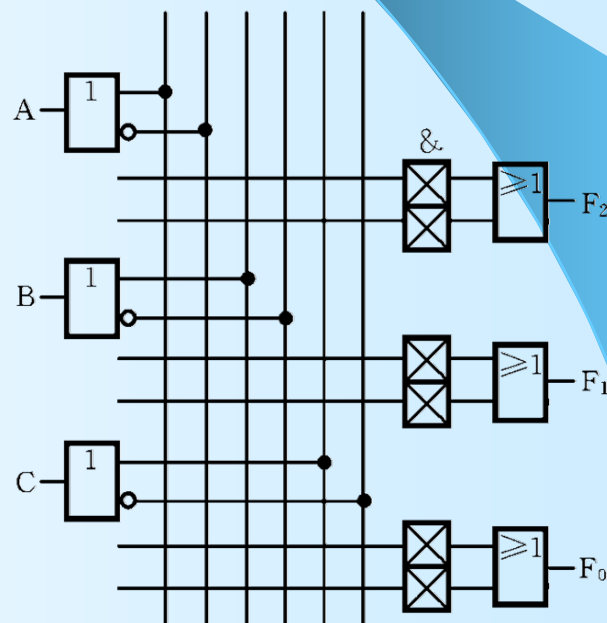


一、PAL的逻辑结构

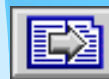
PAL由一个可编程的“与”阵列和一个固定连接的“或”阵列组成。图 (a) 给出了一个三输入三输出PAL的逻辑结构图，通常将其表示成图 (b) 所示形式。



(a)



(b)





PAL每个输出包含的“与”项数目是由固定连接的“或”阵列提供的。在典型逻辑设计中，一般函数约包含3个~4个“与”项，而现有PAL器件最多可为每个输出提供8个“与”项，因此，使用这种器件能很好地完成各种常用逻辑电路的设计。

PAL器件的结构(包括输入、输出、“与”项数目)是由生产厂家固定的。

从PAL问世至今，大约已生产出几十种不同的产品，按其输出和反馈结构，大致可将其分为5种基本类型。

(1) 专用输出的基本门阵列结构

这种结构类型适用于实现组合逻辑函数。常见产品有PAL10H8(10个输入，8个输出，输出高电平有效)，PAL12L6(12个输入，6个输出，输出低电平有效)等。



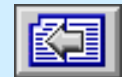


(2) 带反馈的可编程I/O结构

带反馈的可编程I/O结构通常又称为异步可编程I/O结构。该类PAL器件常见产品有PAL16L8 (10 个输入, 8个输出, 6个反馈输入) 以及PAL20L10 (12个输入, 10个输出, 8个反馈输入)。

(3) 带反馈的寄存器输出结构

带反馈的寄存器输出结构使PAL构成了典型的时序网络结构。该类器件的典型产品有PAL16R8 (8个输入、8个寄存器输出、8个反馈输入、1个公共时钟和1个公共选通)。





(4) 加“异或”、带反馈的寄存器输出结构

这种结构是在带反馈寄存器输出结构的基础上增加了一个异或门，该类电路的典型产品有PAL16RP8（8个输入，8个寄存器输出，8个反馈输入）。

(5) 算术选通反馈结构

算术PAL是在综合前几种PAL结构特点的基础上，增加了反馈选通电路，使之能实现多种算术运算功能。算术PAL的典型产品有PAL16A4（8个输入、4个寄存器输出、4个可编程I/O输出、4个反馈输入、4个算术选通反馈输入）。





二、 PAL应用举例

设计的一般步骤

用PAL进行逻辑设计的一般步骤如下：

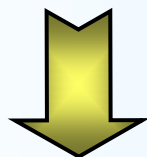
(1) 建立对给定问题的逻辑描述；



(2) 写出相应的函数表达式；



(3) 选择合适的器件；



(4) 编程实现。





8.2.4 通用阵列逻辑GAL

GAL(Generic Array Logic)器件是1985年由美国LATTICE公司开发并商品化的一种新的PLD器件。它是在PAL器件的基础上综合了E²PROM和CMOS技术发展起来的一种新型技术。

GAL器件具有PAL器件所没有的可擦除、可重写及结构可组态等特点。这些特点形成了器件的可测试性和高可靠性，且具有更大的灵活性。





一、GAL的基本逻辑结构

1. GAL16V8的基本组成

GAL16V8芯片是一种具有8个固定输入引脚、最多可达16个输入引脚，8个输出引脚，输出可编程的普通型GAL。

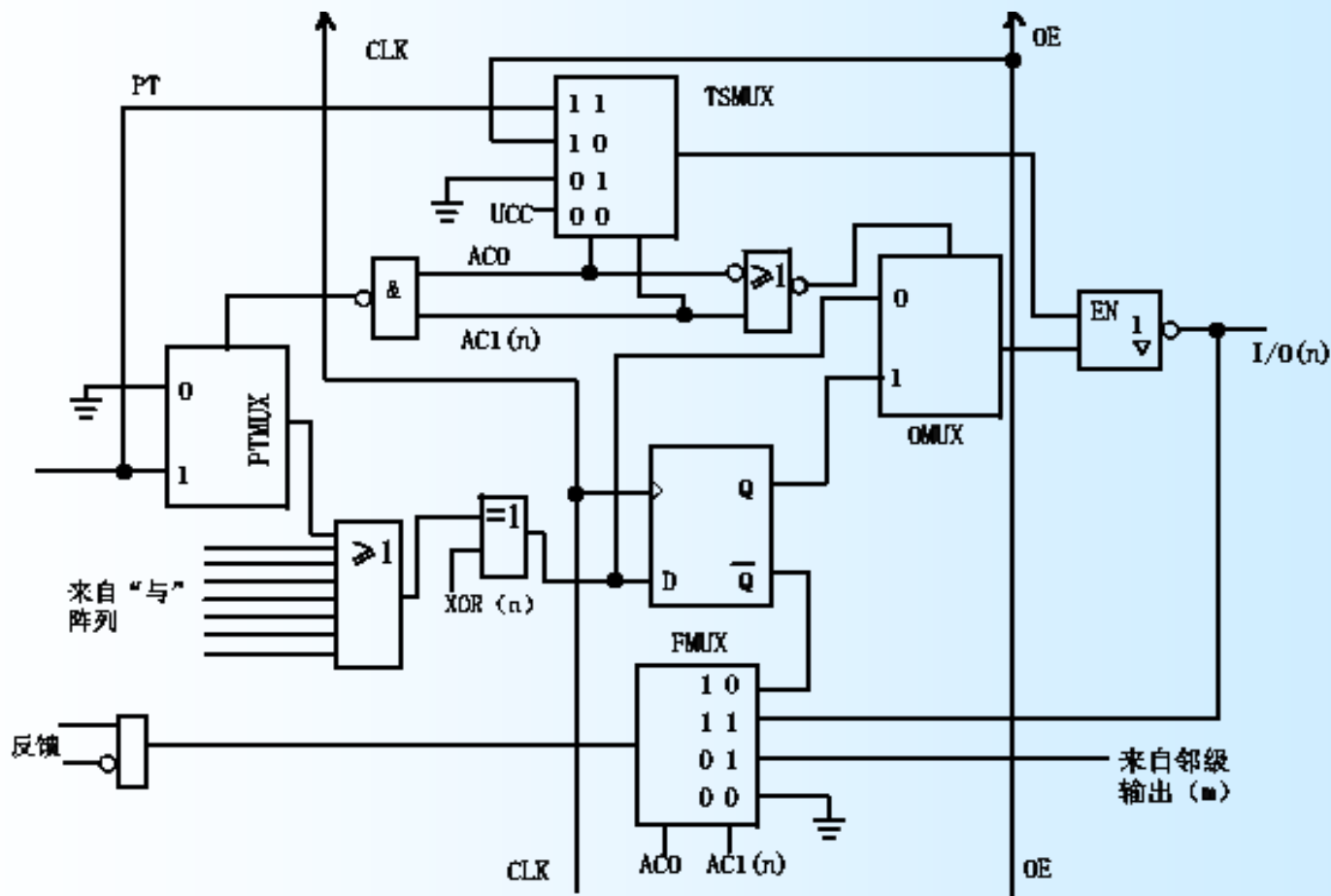
组成: 由8个输入缓冲器、8个反馈输入缓冲器、8个输出逻辑宏单元OLMC，8个输出三态缓冲器、“与”阵列以及系统时钟、输出选通信号等组成。





2. 输出逻辑宏单元OLMC

OLMC由一个8输入或门、极性选择异或门、D触发器、4个多路选择器等组成。其结构如下图所示。





3. 结构控制字

(1) 结构控制字的组成

GAL16V8由一个82位的结构控制字控制着器件的各种功能组合状态。该控制字各位功能如下。

82位							
PT63		PT32			PT31		PT0
“与”项禁止位 32位		XOR(n) 4位	SYN 1位	AC1(n) 8位	ACO 1位	XOR(n) 4位	“与”项禁止位 32位
		12 ... 15		12 ... 19		16 ... 19	

图中，XOR(n)和AC1(n)字段下面的数字分别对应器件的引脚号。





结构控制字各位的功能如下：

同步位SYN ——确定器件是寄存器输出或是纯粹的组合输出。
SYN=0时，GAL器件有寄存器输出能力；
SYN=1时，GAL为一个纯粹组合逻辑器件。

结构控制位AC0——该位对于8个OLMC是公共的，它与AC1(n)配合控制各个OLMC(n)中的多路选择器。

结构控制位AC1——共有8位，每个OLMC(n)有单独的AC1(n)。

极性控制位XOR(n)——它通过OLMC(n)中的异或门控制逻辑操作结果的输出极性。
XOR(n)=0时，输出信号O(n)低电平有效；
XOR(n)=1时，输出信号O(n)高电平有效。

“与”项(PT)禁止位——共64位，分别控制“与”阵列的64行(PT0～PT63)，以便屏蔽某些不用的“与”项。

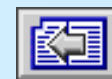




(2) OLMC(n)的5种组态

通过编程结构控制字中的SYN、AC0和AC1(n)，输出逻辑宏单元OLMC(n)可以组成以下5种组态。

- ① 专用输入方式($\text{SYN} \cdot \text{AC0} \cdot \text{AC1(n)} = 101$)
- ② 专用组合型输出方式($\text{SYN} \cdot \text{AC0} \cdot \text{AC1(n)} = 100$)
- ③ 组合型输出方式($\text{SYN} \cdot \text{AC0} \cdot \text{AC1(n)} = 111$)
- ④ 寄存器型器件中的组合逻辑输出方式
($\text{SYN} \cdot \text{AC0} \cdot \text{AC1(n)} = 011$)
- ⑤ 寄存器型输出方式($\text{SYN} \cdot \text{AC0} \cdot \text{AC1(n)} = 010$)

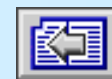
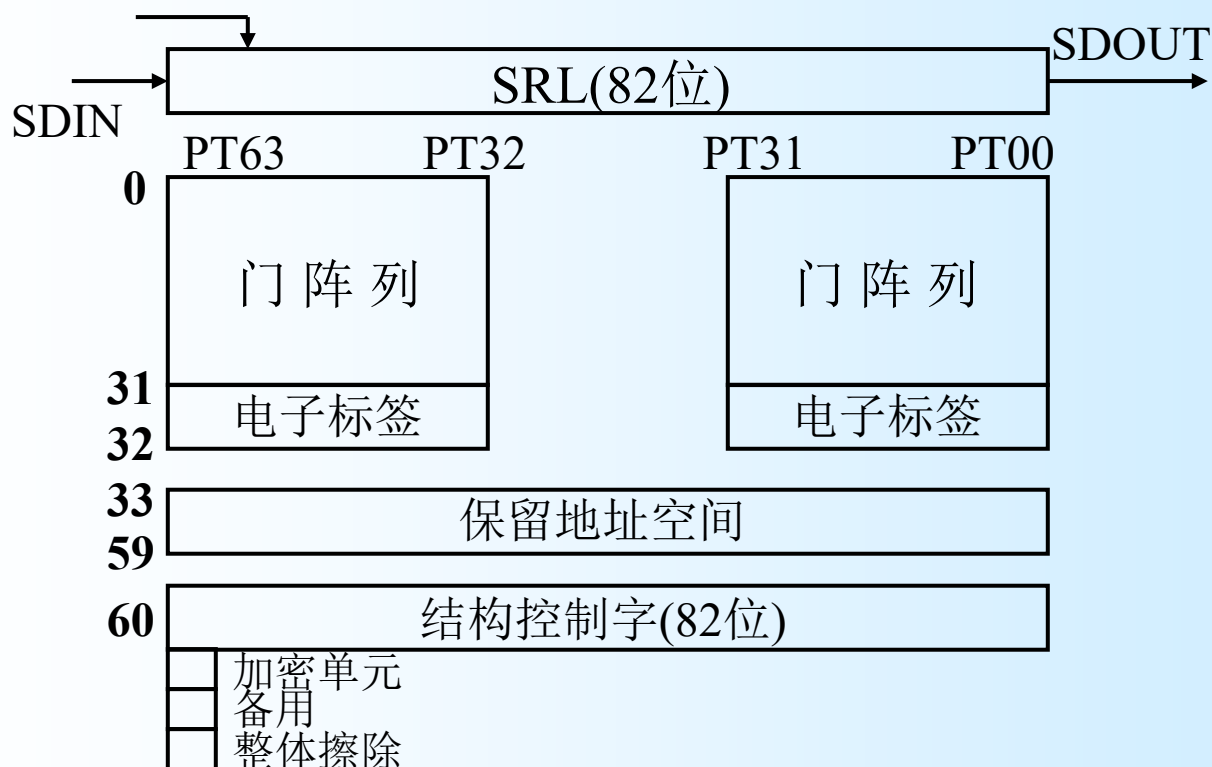




4. 行地址布局

(1) 行地址布局图

GAL器件的可编程阵列包括“与”阵列、结构控制字、保密位及整体擦除位等。对其进行编程时是由行地址进行映射的。GAL16V8的行地址布局图如图所示。





(2) 供用户使用的行地址

当对GAL16V8进行编程实现某个设计方案时，可供用户使用的行地址总共有36个，它们分别是：

① **行地址0~31**：用户用来编程制定逻辑模式的阵列。32个行地址对应逻辑结构图上“与”阵列的32个输入,每个行地址可写入64位数据，对应于64个“与”项。

② **行地址32**：器件的电子标签(ES)。它提供64位供用户定义。例如，用来识别用户身份的代码、版本号、编目管理等，这个标签中的数据与下述保密单元的状态无关，用户始终可以使用。

③ **行地址60**：82位的结构控制字，用于设计所需用途的器件。





④ **行地址61**：仅1位的保密位，用于防止复制阵列的逻辑点阵。该位一旦被编程，存取阵列的电路就不能工作，从而防止了对32位的“与”阵列再次编程或读出。

该单元只能在整体擦除时和阵列一起被擦除，所以，一旦保密位被编程，就绝对不能检查阵列的原始配置，仅电子标签除外。

⑤ **行地址63**：仅含1位，用于器件整体擦除。在器件编程期间访问该行时，执行清除功能，整个“与”阵列、结构控制字、电子标签以及保密单元统统被擦除，使编程的器件恢复到未使用的状态。





二、GAL的应用

1. 用GAL进行逻辑设计的一般步骤

一般步骤如下：

(1) 分析设计要求，确定描述给定功能的逻辑表达式。

这是关键的一步，设计者可综合运用逻辑设计技术，完成有关逻辑设计。

(2) 根据逻辑设计结果，选择GAL器件并对器件进行引脚分配。

在器件选择时，一般首先应满足输入、输出端数目的要求，其次考虑速度、功率等因素。器件确定后，根据要求给各引脚赋予变量名。





(3) 编写GAL设计说明书

在调用编程软件前,首先要根据规定的语法格式和逻辑设计结果编写设计说明书(又称为GAL源程序)。

说明书中通常包括器件型号、实现的功能、设计者姓名和设计日期、电子标签、GAL引脚表以及逻辑表达式等。

(4) 建立有关文件

建立文本文件(LST文件),熔丝图文件(PLT文件),标准装载文件(JE D文件)等。

(5) 硬件编程

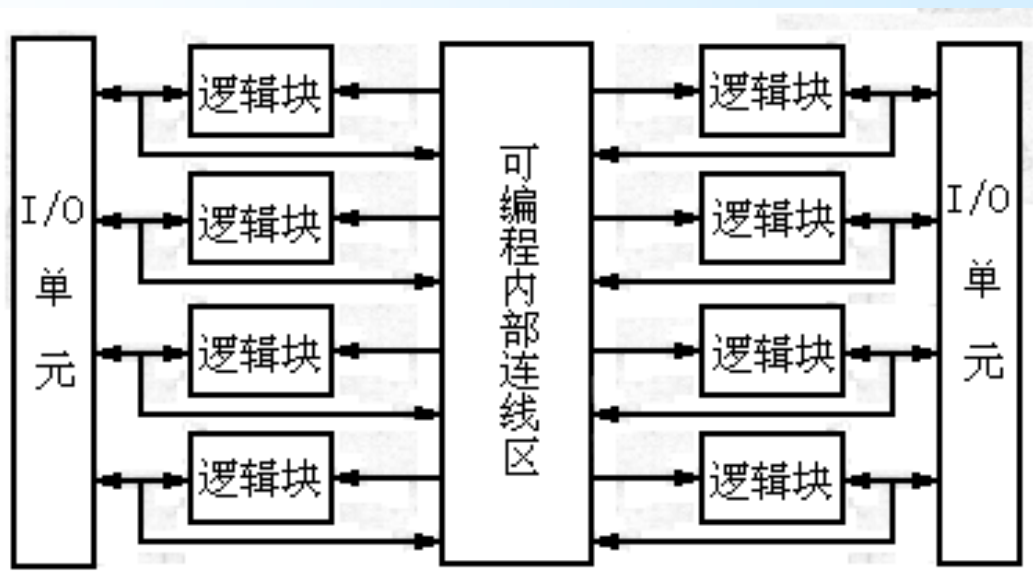
包括连接编程器,建立GAL编程系统,编程后写保密位等。





8.3 复杂可编程逻辑器件

复杂可编程逻辑器件（complex programmable logic device, 简称CPLD）是从简单PLD发展而来的高密度PLD器件。一般由逻辑块、可编程内部连线区和I/O单元组成。常见CPLD的结构示意图如下：





一、CPLD的结构

根据集成规模和设计方法的不同,各生产厂家提供的CPLD在结构上具有各自的特色。目前,常用的CPLD结构有通用互连阵列结构、大块结构和灵活逻辑单元阵列结构等不同类型。

二、典型器件

最常用的CPLD有Altera公司生产的FLEX 10K系列器件。FLEX 10K 是一种嵌入式的PLD。它采用灵活逻辑单元阵列结构和重复可构造的CMOS SRAM工艺,具有高密度、低成本、低功率等特点,现已成为ALTERA CPLD中应用前景最好的器件系列。

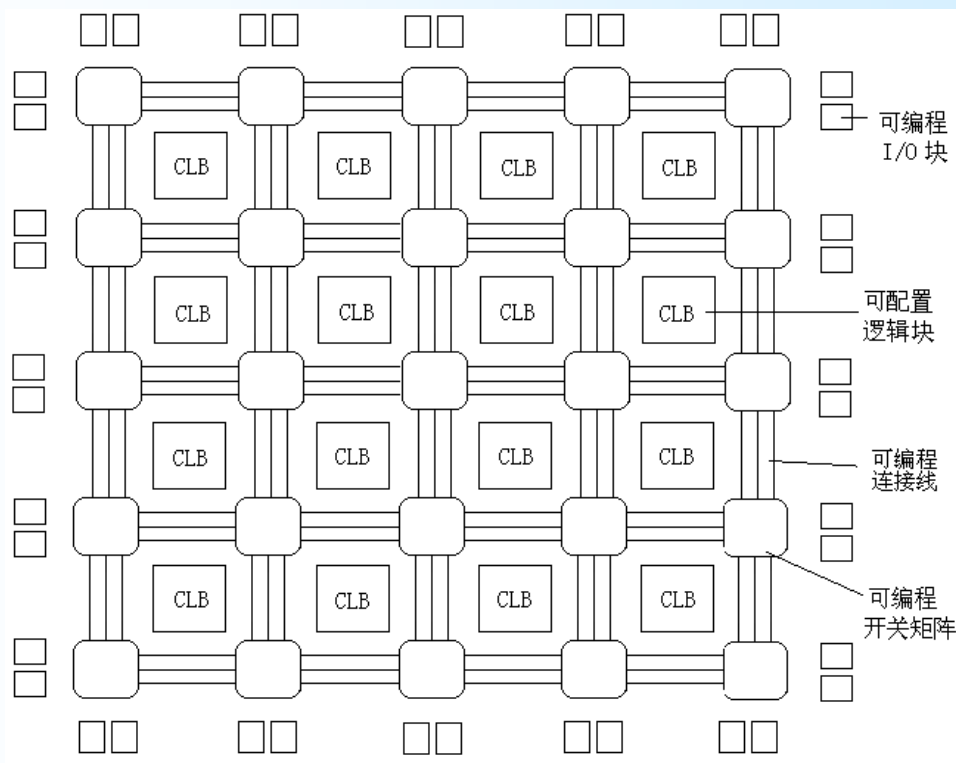
(有关CPLD的介绍见教材中相关部分)





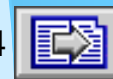
8.4 现场可编程门阵列

现场可编程门阵列FPGA（Field Programmable Gate Array）是20世纪80年代中后期发展起来的一种高密度可编程逻辑器件，它由世界著名的可编程逻辑器件供应商Xilinx公司最初提出。不同公司生产的FPGA器件在结构和性能上不尽相同，Xilinx公司的XC4000系列FPGA器件的结构示意图如下：





有关FPGA的典型器件见教材中相关部分！





8.5 在系统编程技术简介

在系统编程(In System Programmable, 简称ISP)技术是20世纪90年代发展起来的一种PLD技术。

所谓在系统编程，是指可以在用户自己设计的目标系统上、为实现预定逻辑功能而对逻辑器件进行编程或改写。

ISP器件的出现，使数字系统设计更加灵活、方便，为用户带来了显著的经济效益和时间效益。可以说,ISP技术是PLD设计技术发展中的一次重要变革。





8.5.1 ISP技术的主要特点

ISP技术的应用，对数字系统硬件设计方法、设计环境、系统调试周期、测试与维护、系统的升级以及器件的充分利用等均产生了重要影响。**主要特点如下：**

- 一、全面实现了硬件设计与修改的软件化
- 二、简化了设计与调试过程
- 三、容易实现系统硬件的现场升级
- 四、可降低系统成本，提高系统可靠性
- 五、器件制造工艺先进，性能参数好





8.5.2 ISP逻辑器件

一、ISP逻辑器件的类型

目前，市场上提供的ISP逻辑器件可分为ispLSI、ispGAL和ispGDS三种类型。

1. ispLSI逻辑器件

ispLSI(在系统编程大规模集成)逻辑器件具有集成度高、速度快、可靠性好、灵活方便等优点，能满足在高性能系统中实现各种复杂逻辑功能的需要，被广泛应用于数据处理、图形处理、空间技术、军事装备及通信、自动控制等领域。





ISP技术是美国Lattice公司率先推出的，该公司将ISP技术应用到高密度可编程逻辑器件(HDPLD)中，形成了ispLSI系列高密度在系统可编程逻辑器件。常用ispLSI器件有**4个系列**：

- ① **基本系列ispLSI1000**：适用于高速编码、总线管理等；
- ② **高速系列ispLSI2000**：I/O端口数较多,适用于高速计数、定时等场合，并可用作高速RISC/CISC微处理器的接口；
- ③ **高密系列ispLSI3000**：集成密度高，能实现非常复杂的逻辑功能，适用于数字信号处理、图形处理、数据压缩等；
- ④ **模块化系列ispLSI6000**：带有存储器和寄存器/计数器，适用于数据处理、数据通信等。





2. ispGAL器件

ispGAL系列器件,是把ISP技术引入到标准的低密度系列可编程逻辑器件中形成的ISP器件。

典型产品: ispGAL22V10

ispGAL22V10是把流行的GAL22V10与ISP技术相结合形成的产品,在功能和结构上与GAL22V10完全相同。

适应范围: 高速图形处理和高速总线管理, 状态控制、数据处理、通信工程、测量仪器以及实现诸如地址译码器之类的基本逻辑功能。





3. ispGDS器件

ispGDS(在系统可编程数字开关)是ISP技术与开关矩阵相结合的产物。它标志着ISP技术已从系统逻辑领域扩展到系统互连领域。

ispGDS器件能提供的一种独特功能是在不拨动机械开关或不改变系统硬件的情况下,快速地改变或重构印制电路板的连接关系。

ispGDS系列器件非常适合于重构目标系统的连接关系,它使系统硬件可以通过软件控制进行重构而无需人工干预。





二、ispLSI 逻辑器件的结构

ispLSI是ISP逻辑器件中最早问世、最具代表性的逻辑器件。该系列器件是基于与、或阵列结构的复杂PLD产品。

ispLSI芯片由若干巨块组成，巨块之间通过全局布线区GRP连接起来，每个巨块包括若干个通用逻辑块GLB、输出布线区ORP、若干输入输出引脚和专用输入引脚等。

典型产品：ispLSI1016

1. ispLSI1016的特性

ispLSI1016是采用E²CMOS工艺制造、PLCC(塑料有引脚芯片载体)封装的器件。芯片共有44个引脚，其中38个I/O引脚；集成密度为2000等效门，每片含64个触发器和32个锁存器；最高工作频率为110MHz。



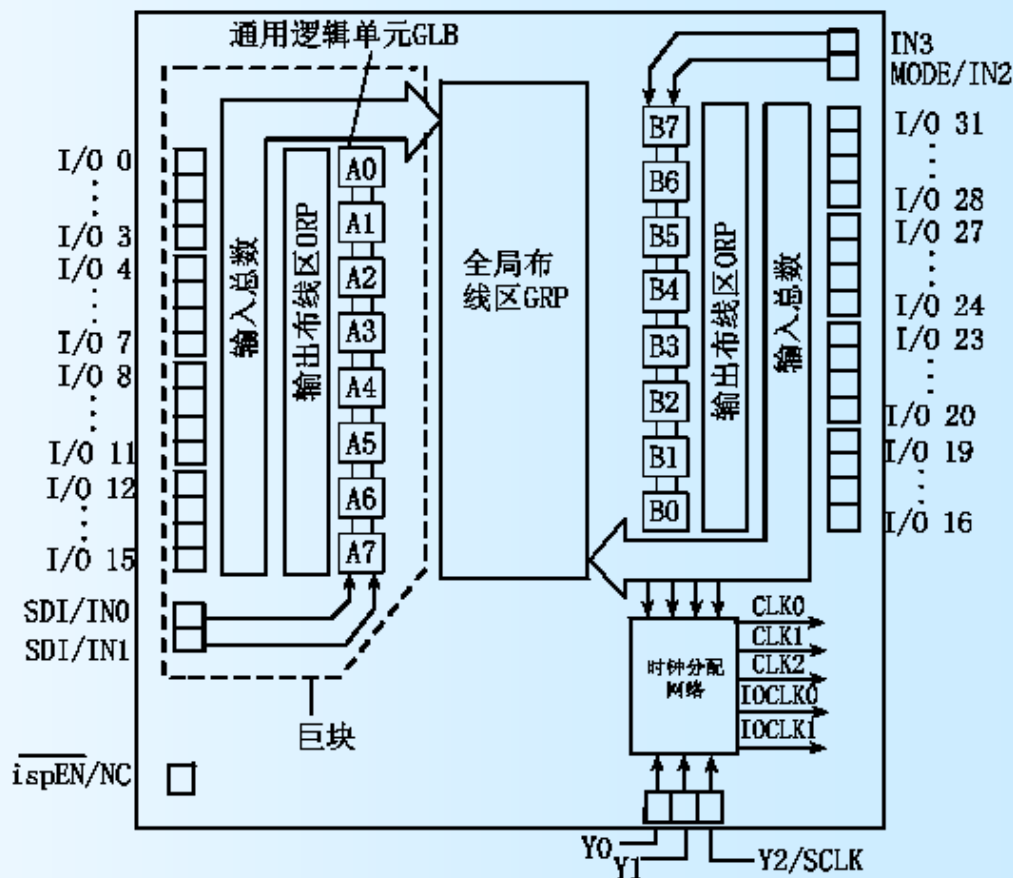


2. 组成

该器件由两个巨块、一个全局布线区和一个时钟分配网络构成。如右图所示。

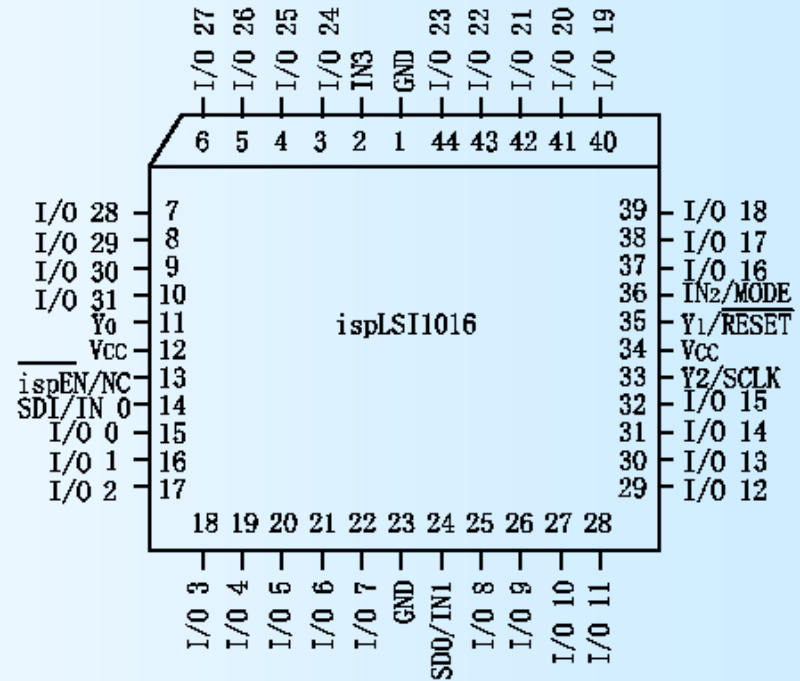
每个巨块包括:8个通用逻辑单元GLB, 16个I/O单元, 两只专用输入引脚(IN0,IN1或IN2,IN3),一个输出布线区ORP及16位输入总线。

器件型号ispLSI1016中的“16”表示器件含有16个GLB。





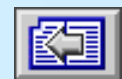
ispLSI1016 的引脚排列图如右图所示。



3. 主要模块功能

(1) 全局布线区GRP

GRP位于两个巨块之间。除了经过各个I/O单元的输入信号由16位输入总线送至GRP之外，各GLB的输出在送往输出布线区的同时也送往GRP。GRP实现上述信号和各个GLB输入之间的灵活互连,将所有片内逻辑联系在一起，供设计者使用，设计者可以根据需要方便地实现各种复杂的逻辑功能。





(2) 通用逻辑块GLB

通用逻辑块GLB是ispLSI器件最基本的逻辑单元，由与阵列、乘积项共享阵列、输出逻辑宏单元和控制逻辑电路组成。

与阵列：共有18个输入，18个输入经过输入缓冲器后形成18个输入信号的原信号和非信号，送至20个与门的输入端，通过编程可形成20个与项(又称乘积项PT)。

乘积项共享阵列：乘积项共享阵列将20个与项PT0~PT19分组送到4个或门的输入端,对4个或门的输出进行可编程“线或”后，再送到输出逻辑宏单元中的可重构触发器。

输出宏单元：输出逻辑宏单元中有4个可重构触发器，由4个数据选择器MUX分别选择GLB的4个输出O3~O0为组合输出或者寄存器输出。组合电路可有“与或”和“异或”两种方式，触发器可组态为D、T、JK等形式。





(3)输出布线区ORP

输出布线区ORP是介于GLB和输入输出单元IOC之间的可编程互连阵列。

阵列的输入是8个GLB的32个输出端，阵列有16个输出端，分别与该侧的16个IOC相连。通过对ORP编程,可以将任何一个GLB的输出灵活地送到16个IOC中的任何一个(每个GLB有4个输出，每个输出通往4个IOC中的一个)。

特点：GLB与IOC之间没有一一对应的关系，可将对GLB的编程和对外部引脚的连接分开进行,从而可以在不改变外部引脚排列的情况下，修改芯片内部的逻辑设计。

为了减少传输延迟，提高工作速度，每个GLB的4个输出中有两个输出可以跨过ORP直接通向固定的IOC，这种方式称为旁路连接。





(4)输入输出单元IOC

输入输出单元IOC用于将输入信号、输出信号或输入输出双向信号与具体的I/O管脚相连接形成输入、输出、三态输出的双向I/O口,具体由控制输出三态缓冲器使能端的MUX来选择。

IOC工作于输入状态: 包括有输入缓冲、锁存输入及寄存器输入;

IOC工作于输出状态: 包括有输出缓冲、反向输出缓冲及三态输出缓冲;

IOC工作于双向状态: 有双向I/O及带有寄存器的双向I/O。

各种I/O组态与GLB组态相组合,可构成几十种电路方式。





(5) 巨块的输出使能控制电路

巨块内8个GLB中的与项PT19都能用作输出使能控制。8个PT19接到一个8选1输出使能数据选择器OEMUX的数据输入端,通过对三个选择输入的“熔丝”值编程,可选择其中一个PT19作为巨块内16个IOC的公共输出使能控制信号。

(6) 时钟分配网络

GLB B0的4个输出O0~O3能分别作为用户定义的内部时钟CLK1, CLK2 ,IOCLK0及IOCLK1时钟,另有CLK0是外部时钟。





8.5.3 ISP器件的开发软件与设计流程

一、ISP器件的开发软件

目前，常用的ISP器件开发软件有PDS软件，Synario软件，ISP Synario System软件等。





二、设计流程

利用ISP器件开发软件进行设计一般步骤为：

逻辑设计规划



设计输入



设计检验



布局布线



逻辑功能仿真



熔丝图生成



下载编程





1. 逻辑设计规划

目的：选择合适的ispLSI器件实现预定功能。

进行设计规划时，首先定义I/O端口，以便考虑器件的I/O单元是否够用。然后进行任务划分，即将要求完成的设计任务分配到各个通用逻辑块GLB中。最后统计出所需要的GLB数目和I/O单元数目，作为器件型号选择的依据。确定了I/O单元数和GLB数之后，便可选择合适的ispLSI器件。



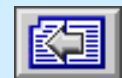


2. 设计输入

所谓设计输入就是将设计者所设计的电路,按照开发软件要求的某种形式表达出来,并输入计算机中。

3. 设计检验

- ① 语法检验
- ② 设计规划检验
- ③ 逻辑最小化
- ④ 逻辑适配
- ⑤ 全局设计规划检验



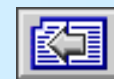


4. 布局布线

布局布线工作是由软件自动完成的,它能以最优方式对逻辑元件布局和准确地实现元件间的互连。

5. 逻辑模拟

逻辑模拟又称为逻辑功能仿真,它包括功能模拟和定时模拟。





6. 熔丝图生成

熔丝图生成是指生成JEDEC文件(熔丝图)，它是设计过程的重要目标。

7. 下载编程

下载编程就是将设计生成的熔丝图文件JEDEC装入ispLSI器件中。

有关ISP器件的编程原理和开发软件的使用等请参考相关书籍。



第八章作业

- 8.5
- 8.7