

前 言

随着集成电路制造技术的发展，在一块芯片上制造的元件越来越多。集成电路规模的扩大为新的电路设计方法提供了物质基础。

作为传统数字系统中使用的主要器件，标准逻辑器件已经使用了 30 多年。标准逻辑器件对于研究数字系统基本构成模块的工作原理具有重要的意义，它在许多基础的理论和实验教学课程中仍然占据重要的位置。目前，“数字电路逻辑设计”课程仍然以标准逻辑器件为主进行讲授。

基于标准逻辑器件的数字电路设计过程包括：定义输入和输出变量；写出描述输入信号和输出信号之间关系的真值表；由真值表可以写出描述电路工作的布尔表达式；利用布尔表达式就可以用逻辑门符号画出电路图；选择合适的数字集成电路器件组装实际电路。你也许有这样的体会，组装实际电路的过程最麻烦，既费时间又容易出错误。如果需要修改电路功能，电路还必须重新组装。

可编程逻辑器件（PLD）能使组装电路这个繁琐的步骤借助计算机和相关的开发软件来完成，因此，现在许多数字系统采用可编程逻辑器件实现以提高设计效率，同时，由于使用的器件数量的减少也提高了系统的可靠性。本书完整地介绍了基于可编程逻辑器件设计应用系统所需要的基础知识，以及利用这些基础知识来设计一个应用系统的过程。

本书是在作者多年来参与全国大学生电子设计竞赛的赛前学生训练、竞赛指导工作以及电子线路课程教学改革经验总结的基础上编写的。在赛前学生训练的教学过程中改革了以往课程的授课方式，通过一系列具有明确目的的设计任务来组织教学。通过合理地安排这些设计任务，把学生感到困难的教学内容进行分解，把一个高的台阶分解成若干个台阶，同时，方便学生从开始上课就接触实际的电路组装和软件编程，使其立刻就能体会到成功的喜悦，提高学习的兴趣。

教学中使用的可编程逻辑器件芯片为 ALTERA 公司生产的 EPM7128SLC84-15 芯片和 EPF10K10LC84-4 芯片。采用这两种芯片进行教学是因为 EPM7128SLC84-15 芯片属于复杂可编程逻辑器件（CPLD）类型，EPF10K10LC84-4 芯片属于现场可编程门阵列（FPGA）器件类型，同时它们的包装形式都具有 PLCC 形式，这种包装形式的芯片由于安装在管座上，便于拆装。

ALTERA 公司是著名的可编程逻辑器件生产厂家，它的产品在我国，尤其在高校的教学中获得广泛的应用。ALTERA 公司的可编程逻辑器件具有高性能、高集成度和高性价比的优点，此外，该公司还提供了功能全面的开发工具，例如获得广泛使用的 MAX+PLUS II 开发软件。本书介绍的 Quartus II 是 ALTERA 公司新的开发软件，它是该公司前一代可编程逻辑器件的集成开发软件 MAX+plus II 的更新换代产品。

本书是在多年来教学教案的基础上编写的。书中的内容不仅包括了许多任课教师的教学经验，也包括了许多学生的学习经验。在和同学们一起共同的学习过程中，我们教授给同学

们知识，同时也从同学们那里学习到很多东西。书中的许多硬件电路和软件程序也是同学们参与设计和调试的，在这里我们向张伟、段高飞、闫兰珍、蔡晓云和张侠等同学表示衷心的感谢。

在多年的教学和全国大学生电子设计竞赛的赛前参赛学生训练中，得到了西安邮电学院和许多老师的支持与帮助；在本书的编写过程中，参考了许多专家、学者的著作和研究成果；本书能够顺利出版得到国防工业出版社和王京涛先生的大力支持；在此向他们表示衷心的感谢。

本书的编写目的是为希望提高工程设计能力的学生以及准备参加全国大学生电子设计竞赛的学生提供一本训练指导书，它也可以作为高校相关专业的教材和工程技术人员的参考书。由于对课时、实验室设备等教学条件的考虑，本书在 Quartus II 可编程逻辑器件的集成开发软件和 VHDL 内容的完整性方面可能有所欠缺，加之作者水平有限，书中的错误与不妥之处在所难免，敬请读者批评指正。

编 著 者

2006年10月于西安

第 1 章 可编程逻辑器件

目标

通过本章的学习，应掌握以下知识：

- 数字电路和系统的特点
- 常用数字集成电路的种类和特点
- 数字电路的各种描述方法
- 组合逻辑电路
- 时序逻辑电路
- 基于标准逻辑器件的数字电路的设计步骤
- 可编程逻辑器件（Programmable Logic Device, PLD）的分类
- 可编程逻辑器件内部电路的描述
- 简单可编程逻辑器件
- 复杂可编程逻辑器件（Complex Programmable Logic Device, CPLD）
- 现场可编程门阵列（Field Programmable Gate Array, FPGA）器件

引言

按照所处理的信号，应用系统可以被划分为数字系统和模拟系统。数字系统具有容易设计、整个系统的准确度以及精度容易保持一致、信息存储方便、抗干扰能力强等优点。采用数字技术面临的最大问题是在现实世界中存在的信号主要以模拟量的形式存在，另外处理数字信号需要花费较多的时间。

数字电路和数字技术具有较多的优点，它在计算机、电信设备、自动化装置、医疗设备以及家用电器等几乎所有的生产和生活领域中获得广泛应用。实际应用的需求促进了数字技术的发展，这些发展包括描述数字系统和数字电路的方法和用来实现这些方法的技术。新方法和新技术的不断出现向我们提出这样问题，在何种程度上，是仅仅只学习新方法，还是设法用老方法去解释新问题。

本章首先对当前用于设计数字电路和系统的主要器件的特点进行讨论；接着对在“数字电路逻辑设计”课程中学习的基于标准逻辑器件对数字电路进行分析和设计的方法进行回顾；然后介绍了一种新的技术——采用可编程逻辑器件设计数字电路，这种技术克服了标准逻辑器件电路可靠性低、修改电路设计困难的缺点；最后介绍可编程逻辑器件的基本工作原理和 ALTERA 公司生产的可编程逻辑器件。

1.1 数字集成电路的分类

尽管本书的主要目的是讨论如何利用可编程逻辑器件实现要求的设计功能，但是考察可供选择的各种器件对数字系统的设计者来说还是有益的，因为它有助于我们更好地理解所有可供选择的方案，同时也可以意识到虽然描述数字系统和数字电路的方法和用来实现这些方法的技术在不断变化，但是基本原理并没有改变。

现代数字系统中所使用的数字电路几乎都是集成电路。使用集成电路实现系统功能比使用分立元件具有电路体积小，可靠性高等优点。从 20 世纪 60 年代开始，数字集成电路在集成度方面的发展经历了以下 4 个阶段：包含几十到几百个逻辑门的小规模集成电路（Small Scale Integration, SSI）；包含几百到几千个逻辑门的中规模集成电路（Medium Scale Integration, MSI）；包含几千到几万个逻辑门的大规模集成电路（Large Scale Integration, LSI）；包含几万个以上逻辑门的超大规模集成电路（Very Large Scale Integration, VLSI）。

在工作原理方面，数字集成电路又可以被划分为标准逻辑器件、微处理器和专用集成电路。

1.1.1 标准逻辑器件

标准逻辑器件在集成度方面属于中小规模集成电路。它包括各种逻辑门、触发器、译码器、多路选择器、寄存器和计数器等器件。标准逻辑器件有 3 种主要类型：TTL、CMOS 和 ECL。TTL 是一种成熟的技术，新的系统设计已经很少采用 TTL 逻辑器件，但是正在运行的系统中仍然包含这种器件。CMOS 器件是当前最流行的标准逻辑器件，它的优点是功耗低。ECL 器件主要用于高速系统中。

作为传统数字系统中使用的主要器件，标准逻辑器件已经使用了 40 多年。标准逻辑器件的产量很大，生产成本低廉，价格便宜。如果我们的设计不很复杂时，这些器件仍然是很实用的。标准逻辑器件对于研究数字系统基本构成模块的工作原理具有重要的意义，它在许多基础的理论和实验教学课程中仍然占据重要的位置。目前“数字电路逻辑设计”课程仍然以标准逻辑器件为主进行讲授。在 1.2 节将一起回顾“数字电路逻辑设计”课程的内容，并以此为基础讨论可编程逻辑器件的工作原理。

标准逻辑器件由于集成度较低，采用它们设计数字系统需要较多的器件，这就使得电路连线复杂，系统的可靠性降低。由于用户无法修改这类器件的功能，修改系统设计必须通过对电路重新设计和组装来实现。

1.1.2 微处理器

数字技术已经进入众多的领域，其中数字计算机是最著名和应用最广泛的产品。尽管计算机影响了人类生活的许多方面，但是许多人并不完全知道计算机能干些什么。简单地说，计算机是一个能完成算术运算、逻辑运算、数据处理和做出判断的数字系统。

个人计算机（PC）是最常见的计算机，它由一些数字集成电路芯片组成，这些芯片包括微处理器芯片、存储器芯片以及输入/输出接口芯片等。在大多数情况下，凡是人能做的，计算机都能做，而且计算机还能干得更快更精确。尽管事实上计算机每次只能完成所有计算中的一步，但是计算机完成每一步的速度非常快，它的高速度弥补了它的低效率。

计算机依靠所运行的软件（程序）来完成工作。这个软件是人们给计算机的一组完整的指令，指令告诉计算机其操作的每一步应该干什么。这些指令以二进制代码的形式存储在计算机的存储器中，计算机从存储器中一次读取一条指令代码，并完成由指令代码指定的操作。

通过编写软件可以控制计算机完成不同的工作，这个特点使得设计灵活性得到提高。当修改系统设计时，设计者只需要改变软件，不需要或者较少需要修改电路连线。由于计算机一次只能执行一条指令，因此它的主要局限性是工作速度。采用硬件方案设计的数字系统总是比软件方案的数字系统的工作速度快。

集成电路制造工艺的发展使得在一个芯片上制造大量的数字电路成为可能，这也促进了计算机技术的发展。把计算机中的微处理器芯片、存储器芯片以及输入/输出接口芯片等做在一块芯片上就形成单片机，有的文献上也被称作为微控制器。这种单芯片的微控制器的性能价格比非常高，它在工程中应用非常广泛。例如，仪表控制、数控机床、自动提款机、复印机、汽车的防抱死制动系统（ABS）、医疗设备等。

1.1.3 专用集成电路

专用集成电路（Application Specific Integrated Circuit, ASIC）的出现一定程度上克服了上述两种逻辑器件的缺点。专用集成电路是为满足一种或几种特定功能而专门设计和制作的集成电路芯片，它的集成度很高。一片专用集成电路芯片甚至可以构成一个完整的数字系统，因此，这使得系统的硬件规模进一步降低，可靠性进一步提高。

专用集成电路可以分为全定制（Full Custom）产品、半定制（Semi-custom）产品和可编程逻辑器件（Programmable Logic Device, PLD）。

1. 全定制产品

全定制产品是指专为特定目的设计、制造的集成电路芯片，例如电视机、电话等设备中大量使用的专用集成电路芯片。这类产品的设计是从晶体管的版图尺寸、位置和相互连线开始进行，其目的是达到半导体芯片面积利用率高、工作速度快、功耗低的优良性能。专用集成电路芯片的制作过程包括电路设计、逻辑模拟、版图设计和集成电路的全部生产工序。全定制产品的性能优越，但是它的设计制造成本高、周期长、同时还具有较大的风险，因此该产品仅适用于需要进行特大批量生产的情况。

2. 半定制产品

半定制产品内部包含基本逻辑门、触发器和具有特定功能的逻辑块所构成的标准单元。这些标准单元是由器件生产厂家预先做好，但是标准单元之间的连线有待按用户要求进行连接。应用半定制产品时，用户需要根据设计要求选择合适的产品，再由产品的结构设计出连线版图，最后交给生产厂家完成各个标准单元之间的连线。

3. 可编程逻辑器件

全定制产品和半定制产品的使用都离不开器件生产厂家的支持，这给用户带来很多麻烦。用户希望自己能设计专用集成电路芯片，并且能立即投入到实际应用之中，而且在使用中也能比较方便地对设计进行修改。可编程逻辑器件就是为满足这一需求而产生的。可编程逻辑器件内的电路和连线都是事先由器件生产厂家做好，但是其逻辑功能并没有确定。逻辑功能的确定可以由设计者借助于开发工具，通过编写软件的方法来实现。可编程逻辑器件的工作速度与标准逻辑器件工作速度相当，但目前使用它们实现信号处理比使用微处理器要复杂，

而且使用成本较高。

1.2 标准逻辑器件

标准逻辑器件是目前大学“数字电路逻辑设计”课程中用来实现数字系统的主要器件。“数字电路逻辑设计”课程中介绍的数字电路描述方法不仅适用于由标准逻辑器件组成的电路，而且也适用于在本书将要学习的由可编程逻辑器件组成的电路。

1.2.1 数字电路的描述

数字电路也称为逻辑电路。数字电路的任意一个输入和输出信号仅存在两种可能的状态：高电平或者低电平。由于二进制数也只用两个数字：0 和 1，因此它适合用来表示数字信号。布尔代数是一种描述逻辑关系的数学工具，利用这种数学工具，数字电路输入和输出之间的关系可以用代数方程（布尔表达式）来描述。布尔代数中的数只有两种可能的取值，与普通代数相比，布尔代数容易计算。布尔代数仅有 3 种基本运算是：与（AND）、或（OR）和非（NOT）。

布尔代数不仅可以作为分析和简化数字电路的工具，而且也可以作为数字电路的设计工具，用来设计满足给定输入输出关系的逻辑电路。用于数字电路分析与设计的其它方法还包括真值表、电路图、时序图以及本书将要讨论的硬件描述语言。如果对这些描述方法进行分类，则可以认为：布尔代数利用数学表达式来描述电路输入和输出之间的关系；真值表利用数字来描述电路输入和输出之间的关系；电路图利用符号来描述电路输入和输出之间的关系；时序图是利用信号波形来描述电路输入和输出之间的关系；硬件描述语言利用文本来描述电路输入和输出之间的关系。

下面以交通灯的控制电路为例，形象地回顾“数字电路逻辑设计”课程中讲述的数字电路描述方法。这里交通灯的控制电路控制东西和南北两个方向的信号灯，每个方向的信号灯包括红、黄和绿 3 盏灯。为方便行人，该系统还包括通行/等待时间显示。交通灯一个循环周期包括 16 个状态，对于每个方向红灯占 8 个状态、绿灯占 7 个状态、黄灯占 1 个状态。上述对交通灯的控制电路的要求可以使用真值表进行描述，真值表如表 1-1 所示。

表1-1 交通灯的控制电路真值表

状态	输入 $X_3 X_2 X_1 X_0$	东西方向			南北方向			时间显示
		红	黄	绿	红	黄	绿	
1	0000	0	0	1	1	0	0	7
2	0001	0	0	1	1	0	0	6
3	0010	0	0	1	1	0	0	5
4	0011	0	0	1	1	0	0	4
5	0100	0	0	1	1	0	0	3
6	0101	0	0	1	1	0	0	2
7	0110	0	0	1	1	0	0	1
8	0111	0	1	0	1	0	0	0

(续)

状态	输入 $X_3 X_2 X_1 X_0$	东西方向			南北方向			时间显示
		红	黄	绿	红	黄	绿	
9	1000	1	0	0	0	0	1	7
10	1001	1	0	0	0	0	1	6
11	1010	1	0	0	0	0	1	5
12	1011	1	0	0	0	0	1	4
13	1100	1	0	0	0	0	1	3
14	1101	1	0	0	0	0	1	2
15	1110	1	0	0	0	0	1	1
16	1111	1	0	0	0	1	0	0

把真值表中使输出为“1”的输入状态进行或运算就可以得到描述电路输出和输入之间关系的布尔表达式。以东西方向的绿灯为例，描述输出与输入关系的布尔表达式如式(1-1)所示，即

$$Y_{\text{东西-绿灯}} = \bar{X}_3 \bar{X}_2 \bar{X}_1 \bar{X}_0 + \bar{X}_3 \bar{X}_2 \bar{X}_1 X_0 + \bar{X}_3 \bar{X}_2 X_1 \bar{X}_0 + \bar{X}_3 \bar{X}_2 X_1 X_0 + \bar{X}_3 X_2 \bar{X}_1 \bar{X}_0 + \bar{X}_3 X_2 \bar{X}_1 X_0 + \bar{X}_3 X_2 X_1 \bar{X}_0$$
(1-1)

由上面的布尔表达式可以看出，通过对输入信号进行与(AND)、或(OR)和非(NOT)这3种布尔代数的基本运算就能获得需要的输出。

布尔表达式可以通过化简获得一个比较简单的形式。简单的形式可用简单的电路来实现，这个简单电路与原电路在功能上等效，但是使用较少的器件，包含较少的连线。进一步来看，这样也提高了电路的可靠性，因为相互之间的连线减少，减少了可能的潜在电路故障。常用的化简方法有代数法和卡诺图法两种。化简以后的布尔表达式如式(1-2)所示，即

$$Y_{\text{东西-绿灯}} = \bar{X}_3 \bar{X}_2 + \bar{X}_3 \bar{X}_1 + \bar{X}_3 \bar{X}_0$$
(1-2)

1.2.2 组合逻辑电路

当一个电路的逻辑功能用布尔表达式给出时，具体的逻辑电路图则可以直接由表达式画出。例如，如果需要一个电路具有 $Y_1 = X_1 \cdot X_2 \cdot X_3$ 的逻辑功能，我们就立即想到可以使用 1 个 3 输入与门；如果需要实现 $Y_2 = X_1 + \bar{X}_2$ 的逻辑电路，则可以首先使用 1 个非门实现输入变量 X_2 的反变量，再使用 1 个 2 输入的或门。适用于这些简单例子的原理可以推广到复杂的电路。

如果要设计 1 个电路以实现布尔表达式(1-2)所描述的逻辑功能，由于这个表达式包括 3 个与运算 ($\bar{X}_3 \bar{X}_2$ 、 $\bar{X}_3 \bar{X}_1$ 和 $\bar{X}_3 \bar{X}_0$) 的或运算，因此首先需要 1 个 3 输入的或门。3 输入或门的每一个输入都是电路输入信号的与运算，每一个与运算可以由 2 输入的与门来实现。完整的逻辑电路如图 1-1 所示，图中的非门用来产生输入变量的反变量。

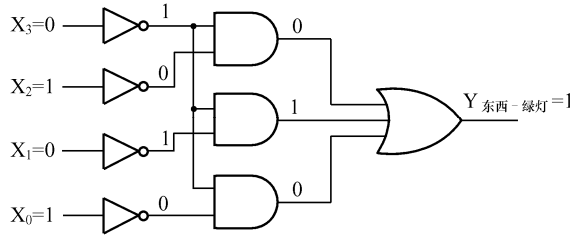


图 1-1 由布尔表达式画出逻辑电路图

当给定逻辑电路的输入信号，它的输出信号可以由电路直接确定，而不必利用布尔表达式进行计算。图 1-1 所示的逻辑电路图中给出输入信号为 $X_3=0, X_2=1, X_1=0, X_0=1$ ，从电路的输入开始分析，经过每一级门电路，逐级写出各级门电路的输出值，直到获得最终的输出值。技术人员在故障检修和系统测试时经常使用这个方法，因为这可以告诉技术人员每个门的输出以及电路最终的输出。

电路的完整时序图可以采用图 1-1 所示的方法画出，电路输入信号 X_0, X_1, X_2 和 X_3 从 0000 变到 1111，求取每一组输入对应的输出。输出信号 $Y_{\text{东西-绿灯}}$ 的波形如图 1-2 所示。

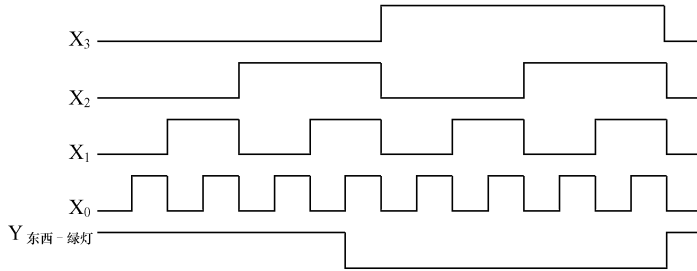


图 1-2 逻辑电路的时序图

其它信号灯的控制电路可以采用同样的方法获得，即从真值表写出布尔表达式，再由布尔表达式画出逻辑电路图。如果使用 7 段数码管来显示道路的通行/等待时间，我们也可以写出反映 7 段数码管的每个发光段和输入信号之间关系的真值表，再采用同样的方法完成它的控制电路的设计。

由于几乎所有的系统都需要数字显示，为简化电路设计，常采用一些中规模逻辑器件实现这个功能。采用 BCD 码-7 段译码器 7447 可以把 BCD 码转换为共阳极数码管的显示代码。图 1-3 给出具体的电路，图中 7447 的 BCD 码输入管脚 D 直接接地，这样当管脚 C~管脚 A 输入从 000 变化到 111 时，7447 即可分别输出共阳极数码管从 0~7 的显示代码。

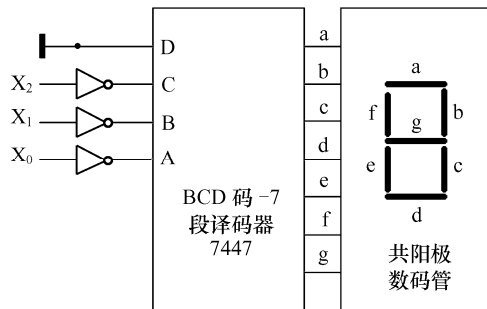


图 1-3 基于中规模逻辑器件的数据显示电路

图 1-3 中的非门把输入信号的变化从加法转换成减法，例如当输入状态 $X_2X_1X_0$ 为 000 时，通过非门后转换成 111，7447 输出共阳极数码管“7”的显示代码；当输入状态 $X_2X_1X_0$ 为 001 时，通过非门后转换成 110，7447 输出共阳极数码管“6”的显示代码；依次类推，当输入状

态 $X_2X_1X_0$ 为 111 时, 通过非门后转换成 000, 7447 输出共阳极数码管“0”的显示代码。

鉴于篇幅的关系, 这里不给出 BCD 码-7 段译码器 7447 的内部电路、功能表以及工作时序图。这些内容可以从网上下载 BCD 码-7 段译码器 7447 数据手册来获得, 从数据手册可以发现它的内部电路也是由逻辑门电路组成。

在组合逻辑电路中, 常用的中规模逻辑器件还有编码器、译码器、数据选择器、数据分配器以及加法器等。这些中规模逻辑器件也是由门电路构成, 读者可以参考相关器件的数据手册。

综上所述, 布尔代数仅有 3 种基本运算: 与 (AND)、或 (OR) 和非 (NOT), 由这 3 种布尔代数基本运算的组合就能获得需要的输出。

1.2.3 时序逻辑电路

长期以来, 逻辑电路被当作组合电路。组合电路是指在任何时刻, 输出状态只取决于该时刻电路输入状态的组合, 而与先前电路的状态无关的逻辑电路。在 1.2.2 节中, 无论是交通灯的控制电路, 还是数据显示电路, 它们的输出只与电路当时的输入有关, 与电路先前的状态无关, 因此它们都是组合逻辑电路。

许多逻辑电路的输出不仅取决于当前时刻各输入状态的组合, 而且还与先前电路的状态有关, 即需要记忆功能。这样的逻辑电路被称作为时序逻辑电路。在前面我们默认电路具有 16 个状态的输入信号, 从 0000~1111, 已经存在。产生这 16 个状态必须使用时序逻辑电路, 因为下一个状态的形式取决于上一个状态的形式。组合逻辑电路的输出只与电路当前的输入有关; 时序逻辑电路的输出则不仅与电路当前的输入有关, 而且还与电路上一个状态有关。

最基本的具有记忆功能的电路是触发器, 它也是由逻辑门组成。逻辑门本身没有记忆能力, 但是几个逻辑门组合起来就具有了记忆能力。图 1-4 给出边沿触发的 J-K 触发器的内部电路和它的电路符号。表 1-2 给出功能表。

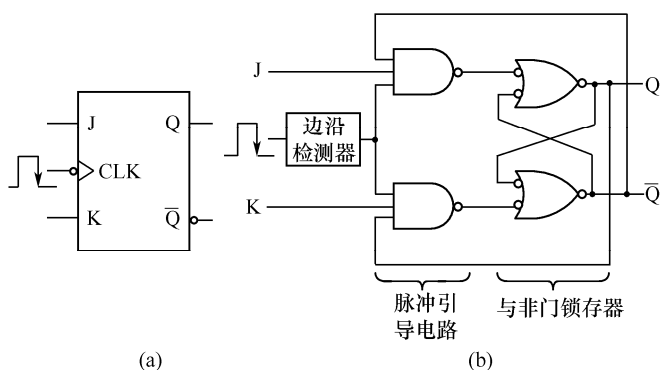


图 1-4 边沿触发的 J-K 触发器的内部电路和它的电路符号

(a) 电路符号; (b) 内部电路。

表 1-2 边沿触发器 J-K 触发器的功能表

J	K	CLK	Q
0	0	•	Q^n (状态不变)
0	1	•	0
1	0	•	1
1	1	•	\bar{Q}^n (状态翻转)

在表 1-2 的第二行, $J=0, K=0$, 时钟跳变时触发器保持原有状态; 在第三行, $J=0, K=1$, 时钟跳变时无论前一个输出状态为什么, 输出状态都置位; 在第四行, $J=1, K=0$, 时钟跳变时无论前一个输出状态为什么, 输出状态都清零; 在第五行, $J=1, K=1$, 时钟跳变时输出状态发生翻转。

把多个边沿触发的 J-K 触发器, 例如 4 个, 按照图 1-5 所示的方法连接起来就可以被用来产生交通灯的控制电路所需要的 16 个状态。

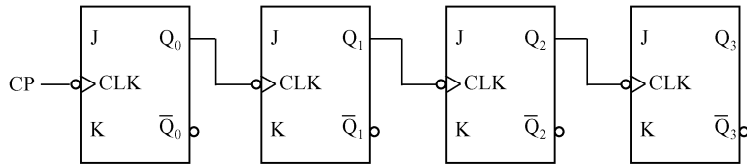


图 1-5 边沿触发的 J-K 触发器连接的状态产生电路 (模 16)

在图 1-5 中, 每个触发器的 J 和 K 端都为 1, 因此当时钟到来时触发器总会改变状态 (发生翻转)。时钟信号 CP 只作用于触发器 Q_0 的 CLK 端。输出端 Q_0 与触发器 Q_1 的 CLK 端相连接, 输出端 Q_1 与触发器 Q_2 的 CLK 端相连接, 输出端 Q_2 与触发器 Q_3 的 CLK 端相连接。图 1-6 是电路工作的时序图, 该图显示了电路在时钟的作用下产生的输出状态的变化情况。

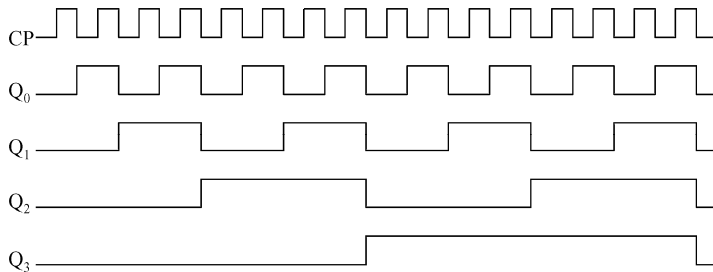


图 1-6 电路的时序图

图 1-5 所示的电路也被称作计数电路。像 BCD 码到 7 段数码管的显示代码转换一样, 这样的电路也具有对应的中规模逻辑器件, 如 4 位二进制同步计数器 74161。对于时序逻辑电路, 常用的中规模逻辑器件还有其它形式的计数器和寄存器等器件。这些中规模逻辑器件也是由门电路和触发器构成, 读者可以参考相关器件的数据手册。

任意组合逻辑电路都可以由与门、或门和非门这样的逻辑门组成。逻辑门本身没有记忆能力, 但是几个逻辑门组合起来可以组成触发器, 就具有了记忆能力。把逻辑门和触发器结合起来就可以组成各种时序逻辑电路。综上所述, 与门、或门和非门这样的逻辑门是数字电路或者数字系统的最基本单元, 换句话说, 应用与门、或门和非门这样的逻辑门就可以实现所有逻辑功能。

1.3 可编程逻辑器件

1.3.1 可编程逻辑器件基础

在介绍可编程逻辑器件之前, 我们再次回顾一下基于标准逻辑器件的数字电路设计过程。首先定义输入和输出变量, 即确定设计输入和输出信号, 并指定变量名称, 如 X_3 、 X_2 、 X_1 、 X_0 和 Y (东西-绿灯)。然后写出描述输入信号和输出信号之间关系的真值表。真值表是描述电路如何工作的方法之一, 描述电路工作原理的另一种方法是布尔表达式。利用布尔表达式就可以用逻辑门符号画出电路图。最后是选择合适的数字集成电路器件组装实际电路。你也许有这样的体会, 组装实际电路的过程最麻烦, 既费时间又容易出错误。如果需要修改电路功能, 电路还必须重新组装。

可编程逻辑器件能使组装电路这个烦人的步骤借助计算机和相关的开发软件来完成，因此现在许多数字系统采用可编程逻辑器件实现以提高设计效率。为支持计算机的工作，可编程逻辑器件提供了硬件基础。由“数字电路逻辑设计”课程的知识，可以得到以下结论：与门、或门和非门这样的基本逻辑门能够组成任何组合逻辑电路；这样的基本逻辑门也能够组成触发器使电路具有存储能力；组合电路加上存储元件就构成了时序逻辑电路。

可编程逻辑器件的基本原理结构图如图 1-7 所示。图中的输入缓冲电路用来对输入信号进行放大，同时这部分电路也为后面的与阵列提供输入信号的反变量。输入缓冲电路输出的所有输入信号和它们的反变量在与阵列中实现布尔表达式中的与项，这个与项也被称作为乘积项。与阵列输出的与项在或阵列中实现或运算。

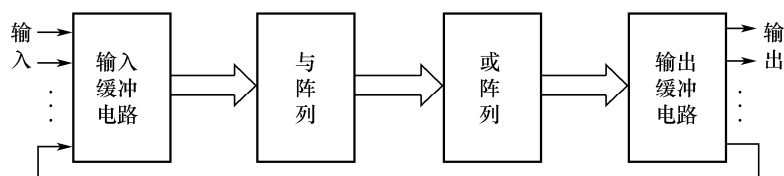


图 1-7 可编程逻辑器件的基本原理结构图

输出缓冲电路中通常包括触发器，输出信号可以直接输出，也可以通过触发器输出。触发器的输出信号还可以反馈回来以实现时序电路。

图 1-7 也可以被看作复杂可编程逻辑器件（Complex Programmable Logic Device, CPLD）的原理结构图，这种“与—或”结构组成的可编程逻辑器件的功能比较简单。可编程逻辑器件还有一种基于查找表的逻辑形成方法。查找表的功能就像真值表，对于每一组输入组合存储 0 或 1，从而产生所要求的组合函数。由于使用多个查找表构成一个查找表阵列，因此这种器件被称为现场可编程门阵列（Field Programmable Gate Array, FPGA）。

1.3.2 可编程逻辑器件内部电路的描述

图 1-8 给出一个简单的可编程逻辑器件的内部组合电路部分电路图。这个器件具有 2 个输入端，A 和 B。2 个同相缓冲器和 2 个反相缓冲器构成输入缓冲电路，输入信号 A 和 B 的每一个分别经过同相缓冲器和反相缓冲器产生原变量和反变量。这些原变量和反变量连接到与门阵列的输入线。与阵列由 4 个与门组成，它们能够产生 2 个输入信号的所有组合的与运算，为此每个与门分别连接到 2 条不同的输入线。与门的输出也被称作乘积项线，它们把与运算的结果送到或阵列。

每条乘积项线通过熔丝与每个 4 输入或门的一个输入端相连。1 个 4 输入或门能够实现一个布尔表达式，采用多个 4 输入或门能够实现多个布尔表达式。由于所有熔丝的原始状态是完好的，因此每个或门的输出为 1。以或门 1 为例，未编程前它的输出为

$$O_1 = \overline{AB} + \overline{A}B + A\overline{B} + AB = 1$$

烧断熔丝的或门输入端被认为是逻辑 0，通过有选择地烧断熔丝，每一个或门的输出都能产生变量 A 和 B 的任意函数。例如设计或门 1 的输入端 1 和输入端 4 的熔丝被烧断，则或门 4 的输出为

$$O_4 = 0 + \bar{A}B + A\bar{B} + 0 = \bar{A}B + A\bar{B}$$

例如，设计或门 2 的输入端 1、输入端 2 和输入端 3 的熔丝被烧断，则或门 2 的输出为

$$O_2 = 0 + 0 + 0 + AB = AB$$

图 1-8 给出的电路仅有 2 个输入信号，但是电路图已经很复杂了，具有许多连线。实际使用的可编程逻辑器件可能具有几十，甚至上百个输入/输出管脚，这样如果画出其内部电路将是非常复杂以致无法使用。为了简化电路的绘制，描述可编程逻辑器件内部电路时采用了不同以前基于标准逻辑器件电路绘制的特殊方法。图 1-9 使用这种特殊方法重新绘制了图 1-8。

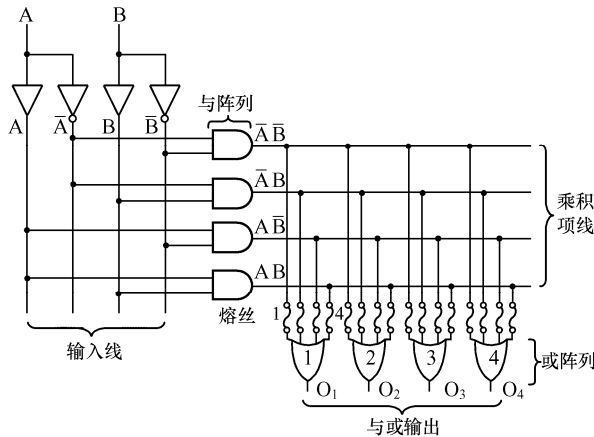


图 1-8 可编程逻辑器件内部电路图

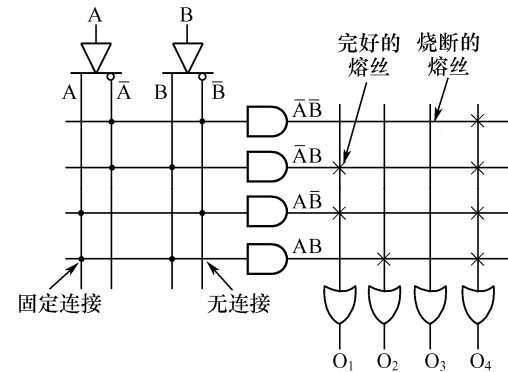


图 1-9 可编程逻辑器件内部电路图

对于图 1-9 所示的电路图，与基于标准逻辑器件电路绘制的区别为：一是用具有 2 个输出端的单个缓冲器代替原来的同相缓冲器和反相缓冲器，新缓冲器的 2 个输出端，一个表示同相，另一个表示反相；二是与门和或门虽然图上只画了一条输入线，但是它表示该逻辑门具有多个输入信号，行线和列线之间具有多少个交叉点，表示该逻辑门具有多少个输入端。

逻辑门的输入线上的交叉点表示该逻辑门的输入端，但是这些输入端的一些可能与输入信号相连接，另一些可能与输入信号不连接。如果在交叉点上具有“·”符号表示这个信号与逻辑门为固定连接；如果在交叉点上具有“×”符号表示这个信号与逻辑门为编程连接；如果在交叉点上既没有“·”符号，也没有“×”符号则表示这个信号与逻辑门不连接。以图 1-9 所示的电路图为例，它的与门输入端采用固定连接，或门输入端采用编程连接，4 个或门的输出分别为

$$O_1 = 0 + \bar{A}B + A\bar{B} + 0 = \bar{A}B + A\bar{B} \quad O_2 = 0 + 0 + 0 + AB = AB$$

$$O_3 = 0 + 0 + 0 + 0 = 0 \quad O_4 = 0 + \bar{A}B + A\bar{B} + 0 = \bar{A}B + A\bar{B}$$

1.3.3 可编程逻辑器件内部电路的分类

自从这项技术问世以来，可编程逻辑器件一直在发展之中。这个发展过程经历了可编程只读存储器(Programmable Read Only Memory, PROM)，可编程逻辑阵列(Programmable Logic Array, PLA)，通用阵列逻辑(Generic Array Logic, GAL)，直到复杂可编程逻辑器件(CPLD)

和现场可编程门阵列 (FPGA)。可编程逻辑器件也可以按器件的编程工艺来划分, 编程工艺包括熔丝型、EPROM 型、E²PROM 型、Flash 型和 SRAM 型。

1. 可编程只读存储器(PROM)

可编程只读存储器芯片的结构如图 1-10 所示。输入缓冲电路提供输入信号的原变量和反变量, 与门提供所有输入信号组合的译码, 或门的输入采用可编程连接。对于任意一个给定的输入信号变量组合, 相应的与门输出高电平, 如果或门的输入和该与门输出相连接, 则此或门输出高电平, 如果或门的输入没有和该与门输出相连接, 则此或门输出低电平。

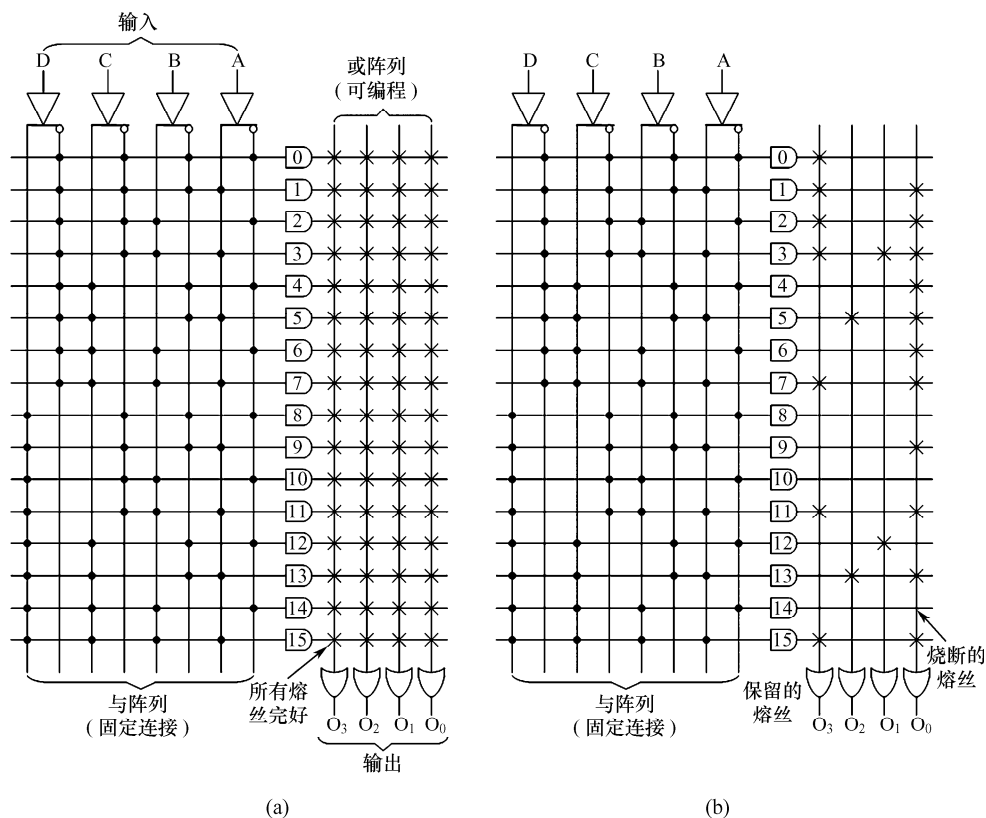


图 1-10 可编程只读存储器芯片的结构

(a) 可编程只读存储器内部电路图; (b) 通过编程实现要求的逻辑函数。

图 1-10 (b) 中, 4 个或门的输出分别为

$$O_3 = AB + \bar{C} \bar{D} \quad O_2 = ABC$$

$$O_1 = ABC \bar{D} + \bar{A} \bar{B} CD \quad O_0 = A + BD + CD$$

由于可编程只读存储器产生了输入信号的所有乘积项, 所以它能产生输入信号的任意逻辑函数。这种器件的缺点是当输入信号包括的变量数目较大时实现起来很困难, 因为每增加一个输入变量, 需要编程的连接点将增加 1 倍。

2. 可编程阵列逻辑(PAL)

考虑在实际应用时, 绝大多数组合逻辑函数并不需要所有的乘积项。可编程阵列逻辑对可编程只读存储器进行了改进, 这种芯片的结构如图 1-11 所示。

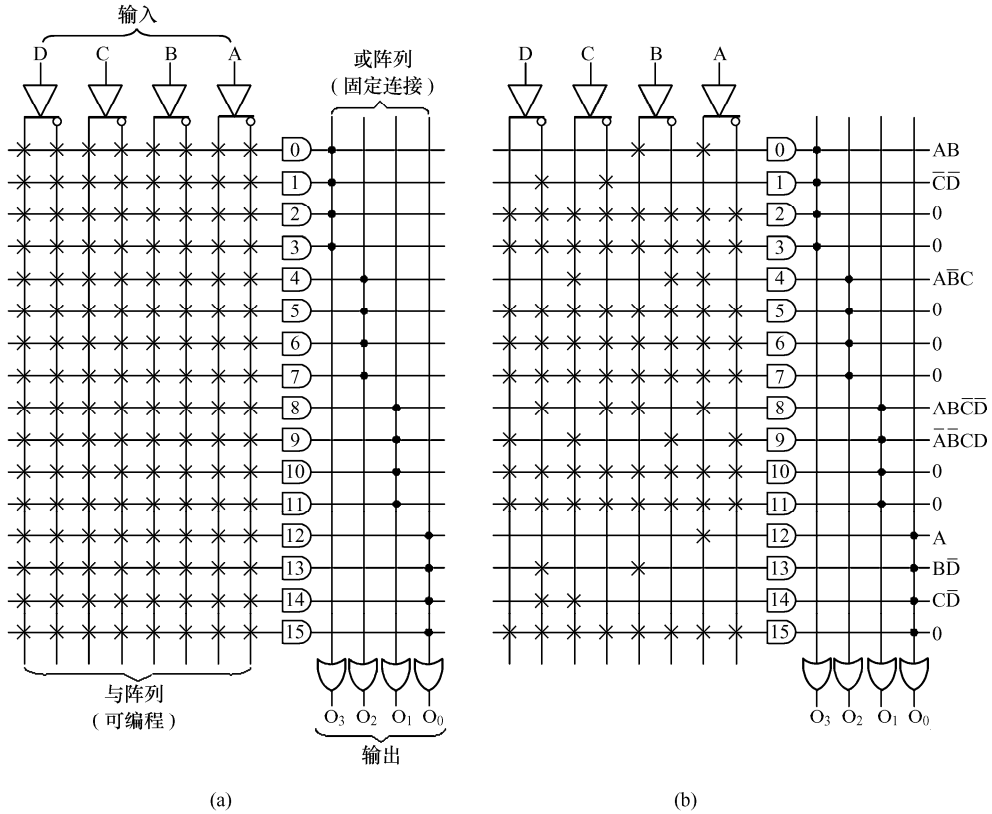


图 1-11 可编程阵列逻辑芯片的结构

(a) 可编程阵列逻辑内部电路图；(b) 通过编程实现要求的逻辑函数。

相对于如图 1-10 所示的可编程只读存储器内部电路，可编程阵列逻辑内部电路的与门的输入是可编程的，或门的输入是固定连接。它适合逻辑函数只需要较少的乘积项的情况。由于与门的输入是可编程的，因此它可以产生输入信号任意一种乘积项。每个或门的输入只和 4 个与门的输出固定连接，它限定了每个输出函数只能有 4 个乘积项。

图 1-11 (b) 中，4 个或门的输出分别为

$$O_3 = AB + \overline{CD} \quad O_2 = \overline{ABC}$$

$$O_1 = ABC\overline{D} + \overline{AB}CD \quad O_0 = A + B\overline{D} + \overline{CD}$$

上面提到的电路结构只能解决组合逻辑的可编程问题，但是对时序逻辑还是无能为力。在图 1-11 所示电路的基础上再加上输出寄存器单元就可以实现时序逻辑的可编程。为实现不同的应用需要，PAL 的输出结构很多，往往一种输出结构就是一种器件。器件种类繁多将使得生产和使用都不方便。现在 PAL 器件已不生产，关于输出寄存器单元的内容下面将会详细介绍。

3. 通用阵列逻辑(GAL)

通用阵列逻辑是在可编程阵列逻辑的基础上发展起来的，它沿用了与阵列可编程、或阵列固定的结构。它们之间的区别反映在编程工艺和输出结构方面。一般的 PAL 器件采用熔丝型编程工艺，属于一次性可编程器件；通用阵列逻辑器件采用 E²PROM 型编程工艺，允许对其进行多次编程。在输出部分增加了输出逻辑宏单元 (Output Logic Macro Cell, OLMC)，使

得一种通用阵列逻辑器件可以代替多种可编程阵列逻辑器件。

图 1-12 给出了 GAL 16V8 的结构图，它是由 Lattice Semiconductor 公司的产品。这种芯片具有 8 个专用输入管脚（管脚 2~管脚 9），2 个特殊功能输入管脚（管脚 1：时钟输入 CLK，管脚 11：使能 OE），8 个输入/输出管脚（管脚 12~管脚 19）。

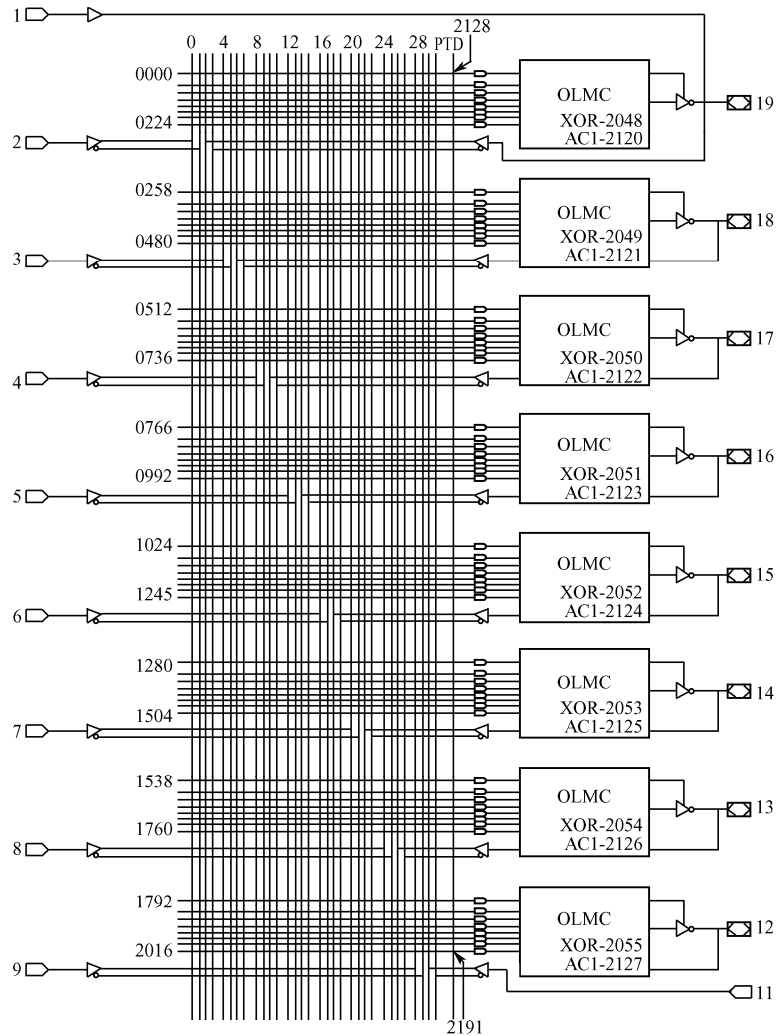


图 1-12 通用阵列逻辑 16V8 结构图

8 个专用输入管脚中每一个的输入信号经过一级缓冲以后，产生输入信号的原变量和反变量，它们分别连接到输入矩阵对应的列线上，为与阵列提供一部分输入信号。与阵列的输入信号还包括来自输出逻辑宏单元的反馈信号，这些反馈信号也分别连接到输入矩阵对应的列线上。从图 1-12 可以看出，与阵列共有 64 个与门，每个与门具有 32 个可编程的输入变量。

输出逻辑宏单元的电路图如图 1-13 所示。每个输出逻辑宏单元具有 1 个 8 输入或门，它接受 8 个固定的多输入与门的输出（乘积项），产生输入信号的“与—或”表达式。或门的输出可以经过不同的路径到达输出管脚，具体经过的路径可以通过编程来决定，既能实现组合逻辑输出，也能实现寄存器输出。

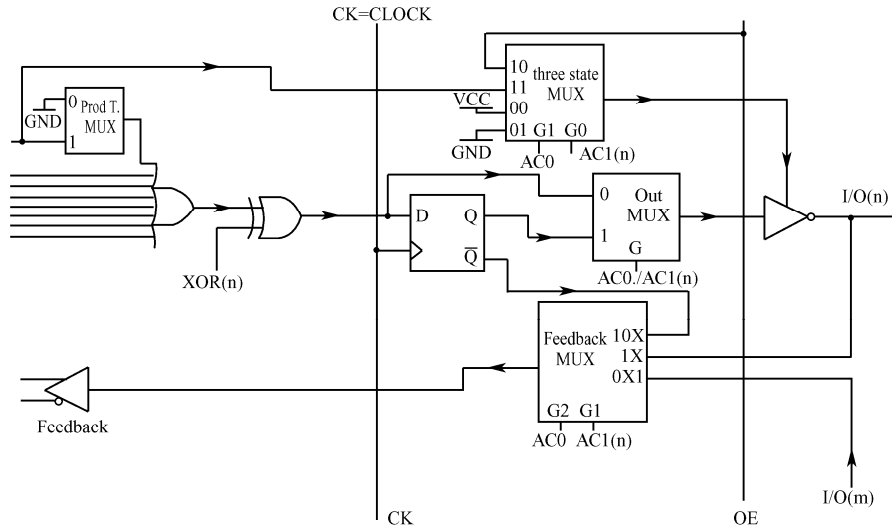


图 1-13 输出逻辑宏单元的电路图

在输出逻辑宏单元电路中，8 输入或门的信号分两类：7 个输入乘积项直接与或门的输入端连接，第 8 个输入乘积项则通过乘积项数据选择器（Prod.TMUX）连接到或门的输入端。乘积项数据选择器由 E^2 PROM 矩阵中的可编程位（AC0 和 AC1）控制，使得第 8 个输入乘积项可以送入或门，也可以不送入或门。

4 输入三态数据选择器（three state MUX）用来对输出三态反相器使能，它的 4 个输入信号为 V_{CC} 、接地、第 8 个输入乘积项以及来自管脚 OE（管脚 11）的外部信号。当三态数据选择器选择 V_{CC} 输入，三态反相器输出处于使能状态，这时输入/输出管脚可以用于输出管脚；当选择接地输入，三态反相器输出处于高阻状态，这时输入/输出管脚可以用于输入管脚。三态数据选择器的另外两个输入信号使得输出可以由第 8 个输入乘积项或者来自管脚 OE（管脚 11）的外部信号进行控制。

输出数据选择器（Out MUX）是 1 个 2 输入数据选择器，它由 E^2 PROM 矩阵中的可编程位（AC0 和 AC1）控制，在组合逻辑输出和寄存器（D 触发器）输出之间实现选择。

反馈数据选择器（Feedback MUX）也由 E^2 PROM 矩阵中的可编程位（AC0 和 AC1）控制，用来选择反馈到输入矩阵中的逻辑信号，这些反馈信号和芯片的输入信号一起作为“与-或”阵列的输入信号。由于反馈信号中包括 D 触发器的输出信号，这个特点使得 GAL 器件具有了实现时序逻辑的能力。

图 1-13 中的 2 输入异或门用来提供可编程输出信号极性的能力。2 输入异或门的一个输入信号来自 8 输入或门的输出，另一个为可编程的控制信号 CONTROL。当 CONTROL=0，8 输入或门的输出保持原来的极性；当 CONTROL=1，8 输入或门的输出被反相。

上面讨论的可编程只读存储器，可编程阵列逻辑和通用阵列逻辑（GAL）通常被称作为简单可编程逻辑器件，即使如此，通用阵列逻辑的内部电路就已经很复杂了。讨论这些内容只是想说明可编程逻辑器件仍是由在“数字电路逻辑设计”课程中学习的基本电路组成，课程中介绍的数字电路理论不仅适用于由标准逻辑器件组成的电路，而且也适用于在本书将要学习的由可编程逻辑器件组成的电路。

现在广泛使用的两种可编程逻辑器件是复杂可编程逻辑器件和现场可编程门阵列，它们也被

称作为大容量可编程逻辑器件。这两种器件多个厂家都进行生产，虽然不同厂家的产品的基本结构类似，但是具体电路仍有许多改进形式，同时由于它们的内部电路非常复杂，具体介绍起来将非常困难。在 1.4 节结合 ALTERA 公司的可编程逻辑器件将对它们的组成框图进行介绍。

1.4 ALTERA 公司的可编程逻辑器件

ALTERA 公司是著名的可编程逻辑器件生产厂家，它的产品在我国，尤其在高校的教学中获得广泛的应用。ALTERA 公司的可编程逻辑器件具有高性能、高集成度和高性价比的优点，此外该公司还提供了功能全面的开发工具，例如获得广泛使用的 MAX+PLUS II 开发软件。第 2 章将要介绍的 Quartus II 是 ALTERA 公司新的集成开发软件。

ALTERA 公司的可编程逻辑器件包括多个产品系列，按照这些产品系列推出的先后顺序依次为 Classic 系列、MAX 系列、FLEX 系列、APEX 系列、ACEX 系列、APEX II 系列、Cyclone 系列、Stratix 系列、MAX II 系列、Cyclone II 系列以及 Stratix II 系列。由于我们的教学采用了 MAX 系列的 EPM7128SLC84-15 芯片和 FLEX 系列的 EPF10K10LC84-4 芯片，因此本节将以它们为例进行介绍。采用这两种芯片进行教学是因为 EPM7128SLC84-15 芯片属于复杂可编程逻辑器件类型，EPF10K10LC84-4 芯片属于现场可编程门阵列类型，同时它们的包装形式都具有 PLCC 形式，这种包装形式的芯片安装在管座上，便于它们的拆装。

1.4.1 复杂可编程逻辑器件 (CPLD)

本节讨论 EPM7128S 的结构，这种芯片属于复杂可编程逻辑器件，它采用 E²PROM 型编程工艺，允许对其进行多次编程。EPM7128S 的组成框图如图 1-14 所示。

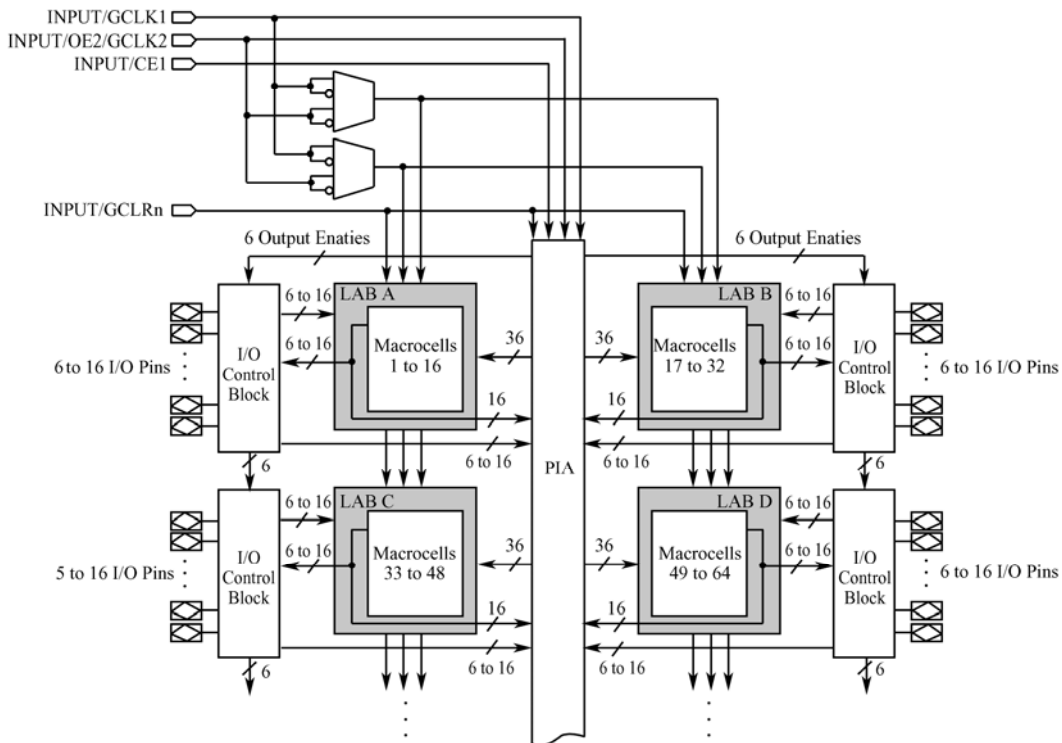


图 1-14 EPM7128S 的组成框图