

EDA 应用技术

<http://www.phei.com.cn>

基于Quartus II 的 FPGA/CPLD数字系统 设计实例 (第2版)

周润景 苏良碧 编著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY



EDA 应用技术

基于Quartus II 的 FPGA/CPLD数字系统 设计实例 (第2版)

周润景 苏良碧 编著



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书以 Altera 公司全新推出的 Quartus II 11.0 为设计平台, 结合大量的实例来介绍基于 FPGA/CPLD 数字系统的设计方法。书中的例子包含简单的数字逻辑电路实例、数字系统设计实例及复杂的数字控制系统设计实例, 由浅入深地介绍了采用 Quartus II 11.0 进行数字系统开发的设计流程、设计思想和设计技巧。本书的所有例子均在 SOPC-NIOSII EDA/SOPC 系统上验证。

本书适合从事数字系统设计的研发人员阅读, 也可作为高等学校电子、通信、自动化等相关专业的教学用书。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有, 侵权必究。

图书在版编目 (CIP) 数据

基于 Quartus II 的 FPGA/CPLD 数字系统设计实例 / 周润景, 苏良碧编著. —2 版. —北京: 电子工业出版社, 2013.1

(EDA 应用技术)

ISBN 978-7-121-18690-5

I. ①基… II. ①周… ②苏… III. ①可编程序逻辑器件—系统设计 IV. ①TP332.1

中国版本图书馆 CIP 数据核字 (2012) 第 239935 号

策划编辑: 张 剑 (zhang@phei.com.cn)

责任编辑: 刘真平

印 刷: 涿州市京南印刷厂

装 订: 涿州市京南印刷厂

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1 092 1/16 印张: 24.75 字数: 633.6 千字

印 次: 2013 年 1 月第 1 次印刷

印 数: 4 000 册 定价: 59.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前 言

随着电子技术、计算机应用技术和 EDA 技术的不断发展,利用 FPGA/CPLD 进行数字系统的开发已被广泛应用于通信、航天、医疗电子、工业控制等领域。与传统电路设计方法相比,FPGA/CPLD 具有功能强大,开发过程投资少、周期短、便于修改,以及开发工具智能化等特点。近年来,FPGA/CPLD 市场发展迅速,并且随着电子工艺不断改进,低成本、高性能的 FPGA/CPLD 器件推陈出新,从而促使了 FPGA/CPLD 成为当今硬件设计的首选方式之一。熟练掌握 FPGA/CPLD 设计技术已经是电子设计工程师的基本要求。

VHDL 语言作为国际标准的硬件描述语言,已经成为工程技术人员和高校学生的必备技能。本书例子中的文本编辑均采用 VHDL 语言编写,并且书中实例均通过仿真和硬件测试。

本书主要以实例为主来介绍以 Altera 公司推出的 Quartus II 11.0 为设计平台的 FPGA/CPLD 数字系统设计。书中的例子包含简单的数字逻辑电路实例、数字系统设计实例及复杂数字系统设计实例,由浅入深地介绍了采用 Quartus II 进行数字系统开发的设计流程、设计思想和设计技巧。

本书共分为 12 章及一个附录,第 1 章介绍了使用可编程逻辑器件进行数字系统设计的基本概念,包括数字集成电路、标准逻辑器件、可编程逻辑器件、CPLD 和 FPGA 等的基本概念;第 2 章介绍了 Quartus II 的使用方法,包括原理图编辑、文本编辑和混合编辑的设计方法;第 3~7 章介绍了简单的数字电路实例,包括门电路、组合逻辑电路、触发器、时序逻辑电路及存储器的设计方法;第 8 章介绍了课程设计中涉及的数字系统设计实例,使读者熟练地掌握 Quartus II 的使用方法和熟练运用 VHDL 语言;第 9 章介绍了宏功能模块及 IP 核的使用方法和简单的应用实例;第 10 章介绍了 ModelSim、TimeQuest、SignalTap II 与 Quartus II 的结合使用方法及简单的应用实例,VHDL 硬件调试,以及在 VHDL 设计当中使用库模块的方法;第 11~12 章介绍了两个大型数字系统的设计实例,使读者更深入地掌握数字系统地设计方法。

本书共 12 章,其中第 1 章由苏良碧负责编写,其余由周润景教授负责编写。全书由周润景统稿、定稿。参加本书编写的还有任冠中、丁莉、王志军、胡训智、李琳、解倩倩、李彦龙、李可洋、赵建凯、朱晓丽、于佳、米利国和田天,在此表示感谢。

感谢北京百科融创教学仪器设备有限公司提供的 SOPC-NIOSII EDA/SOPC 系统,本书的所有例子均在该系统验证过。

向本书参考文献的作者表示感谢。

由于作者水平有限,书中难免存在错误和不足之处,敬请读者批评指正。

目 录

第 1 章	FPGA 设计基础	1
1.1	数字集成电路的分类	1
1.2	标准逻辑器件	3
1.3	可编程逻辑器件	5
1.4	Quartus II 简介	19
1.5	VHDL 程序简介	23
第 2 章	Quartus II 的使用	27
2.1	原理图设计	27
2.2	文本编辑	49
2.3	混合编辑 (自底向上)	64
2.4	混合编辑 (自顶向下)	68
2.5	Qsim 仿真工具的详细使用方法	74
第 3 章	门电路设计范例	90
3.1	与非门电路	90
3.2	或非门电路	92
3.3	异或门电路	94
3.4	三态门电路	96
3.5	单向总线缓冲器	97
3.6	双向总线缓冲器	98
第 4 章	组合逻辑电路设计范例	100
4.1	编码器	100
4.2	译码器	105
4.3	数据选择器	110
4.4	数据分配器	113
4.5	数值比较器	114
4.6	加法器	116
4.7	减法器	120
第 5 章	触发器设计范例	126
5.1	RS 触发器	126

5.2 JK 触发器	127
5.3 D 触发器	129
5.4 T 触发器	131
第 6 章 时序逻辑电路设计范例	133
6.1 同步计数器	133
6.2 异步计数器	137
6.3 减法计数器	139
6.4 可逆计数器	140
6.5 可变模计数器	142
6.6 寄存器	144
6.7 锁存器	146
6.8 移位寄存器	148
6.9 顺序脉冲发生器	155
6.10 序列信号发生器	156
6.11 分频器	157
第 7 章 存储器设计范例	170
7.1 只读存储器 (ROM)	170
7.2 随机存储器 (RAM)	171
7.3 堆栈	173
7.4 FIFO	176
第 8 章 数字系统设计范例	178
8.1 跑马灯设计	178
8.2 8 位数码扫描显示电路设计	180
8.3 4×4 键盘扫描电路设计	183
8.4 数字频率计	186
8.5 乒乓球游戏机	188
8.6 交通控制器	194
8.7 数字钟	200
8.8 自动售货机	206
8.9 出租车计费器	213
8.10 电梯控制器	219
第 9 章 可参数化宏模块及 IP 核的使用	228
9.1 ROM、RAM、FIFO 的使用	228

9.2	乘法器、锁相环的使用	237
9.3	正弦信号发生器的设计	242
9.4	NCO IP 核的使用	243
第 10 章	深入使用 Quartus II 开发软件	248
10.1	使用 ModelSim 波形编辑器对 VHDL 设计进行仿真	248
10.2	TimeQuest 时序分析仪的用法	258
10.3	SignalTap II 嵌入式逻辑分析仪的使用	266
10.4	VHDL 硬件设计调试	279
10.5	在 VHDL 设计当中使用库模块	296
第 11 章	基于 FPGA 的射频热疗系统	311
11.1	肿瘤热疗的生物学与物理技术概论	312
11.2	温度场特性的仿真	314
11.3	射频热疗系统设计	314
11.4	系统硬件电路设计	315
11.5	软件实现	326
11.6	温度场测量与控制的实验	351
11.7	结论	355
第 12 章	基于 FPGA 的直流电动机伺服系统	356
12.1	电动机控制发展情况	356
12.2	系统控制原理	357
12.3	算法设计	359
12.4	系统硬件设计原理	361
12.5	系统软件设计原理	368
12.6	系统调试及结果分析	378
12.7	结论	382
附录 A	RC-EDA/SOPC 实验平台简介	383

第 1 章 FPGA 设计基础

近年来，电子设计的飞速发展使得数字电路的研究及应用出现了广阔的发展空间。使用 FPGA（Field Programmable Gate Array）或 CPLD（Complex Programmable Logic Device）的设计软件可以将设计好的程序烧写入 FPGA 器件中，如同自行设计集成电路一样，可节省电路开发的费用与时间。

1.1 数字集成电路的分类

尽管本书的学习目标是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及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划分为标准逻辑器件、微处理器和专用集成电路。

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。

作为传统数字系统设计中使用的主要器件，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，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。

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。

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, 有的文献上也将其称为微控制器。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。

3. 专用集成电路

专用集成电路 (Application Specific Integrated Circuit, ASIC) 的出现在一定程度上克服了上述两种逻辑器件的缺点。专用集成电路是为了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)。

- 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, 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。
- 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。
- 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的, 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“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. 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：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他方法还包括真值表、电路图、时序图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。

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。

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；如果需要一个实现的逻辑电路，则可以首先使用 1 个非门实现输入变量的反变量，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。

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、数据分配器及加法器等。这些中规模逻辑器件也由门电路构成，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。

3. 时序逻辑电路

长期以来，逻辑电路被当做组合电路。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。无论是控制电路，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。这样的逻辑电路称为时序逻辑电路。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 1-2-1 给出边沿触发的 J-K 触发器的内部电路及其电路符号,表 1-2-1 给出功能表。

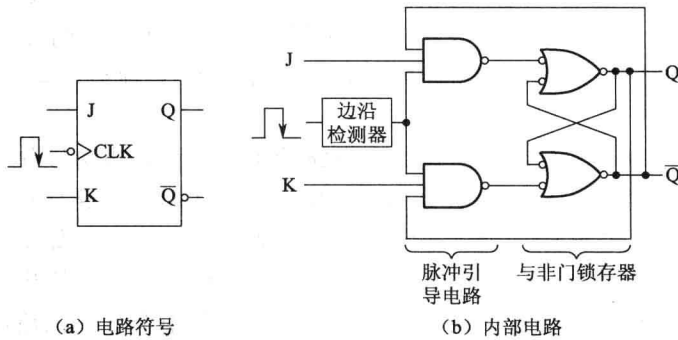


图 1-2-1 边沿触发的 J-K 触发器的内部电路及其电路符号

表 1-2-1 边沿触发的 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	↑	$\overline{Q_n}$ (状态翻转)

在表 1-2-1 的第 2 行, $J=0, K=0$, 时钟跳变时触发器保持原有状态; 在第 3 行, $J=0, K=1$, 时钟跳变时无论前一个输出状态是什么, 输出状态都清零; 在第 4 行, $J=1, K=0$, 时钟跳变时无论前一个输出状态是什么, 输出状态都置位; 在第 5 行, $J=1, K=1$, 时钟跳变时输出状态发生翻转。

对于时序逻辑电路,常用的中规模逻辑器件还有其他形式的计数器和寄存器等器件。这些中规模逻辑器件也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. 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。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。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相关的知识，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 1-3-1 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，这个与项也称为乘积项。与阵列输出的与项在或阵列中实现或运算。

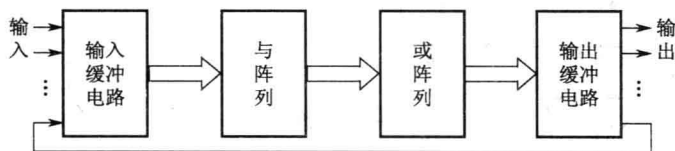


图 1-3-1 可编程逻辑器件的基本原理结构图

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 1-3-1 也可以看做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，因此这种器件称为现场可编程门阵列（Field Programmable Gate Array, FPGA）。

2. 可编程逻辑器件内部电路描述

图 1-3-2 给出一个简单的可编程逻辑器件的内部电路图。这个器件具有两个输入端：A 和 B。两个同相缓冲器和两个反相缓冲器构成输入缓冲电路，输入信号 A 和 B 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，它们能够产生两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两条不同的输入线。与门的输出也称做乘积项线，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

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

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 1 的输出为

$$O_1 = 0 + \overline{A}B + \overline{A}\overline{B} + 0 = \overline{A}B + \overline{A}\overline{B}$$

又如, 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 1-3-2 给出的电路仅有两个输入信号, 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/输出引脚, 如果画出其内部电路, 将是非常复杂的以致无法使用。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于以前的基于标准逻辑器件电路绘制的特殊方法。图 1-3-3 所示为使用这种特殊方法重新绘制的可编程逻辑器件内部电路图。

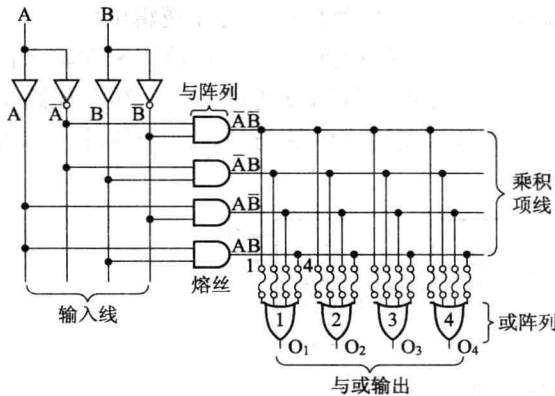


图 1-3-2 可编程逻辑器件内部电路图 (1)

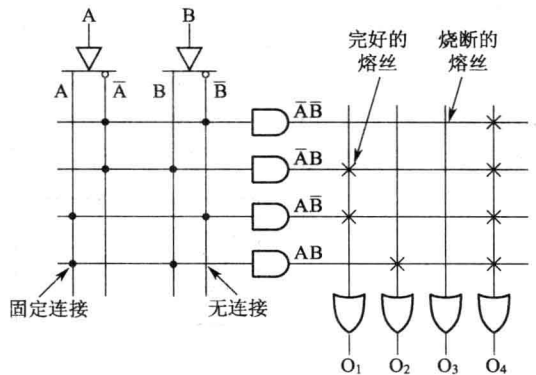


图 1-3-3 可编程逻辑器件内部电路图 (2)

对于图 1-3-3 所示的电路图, 与基于标准逻辑器件电路绘制的区别为: 一是用具有两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, 新缓冲器的两个输出端, 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, 则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, 则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 1-3-3 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

$$O_1 = 0 + \overline{A}B + \overline{A}\overline{B} + 0 = \overline{A}B + \overline{A}\overline{B}$$

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

$$O_3 = 0 + 0 + 0 + 0 = 0$$

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

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 1-3-4 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。

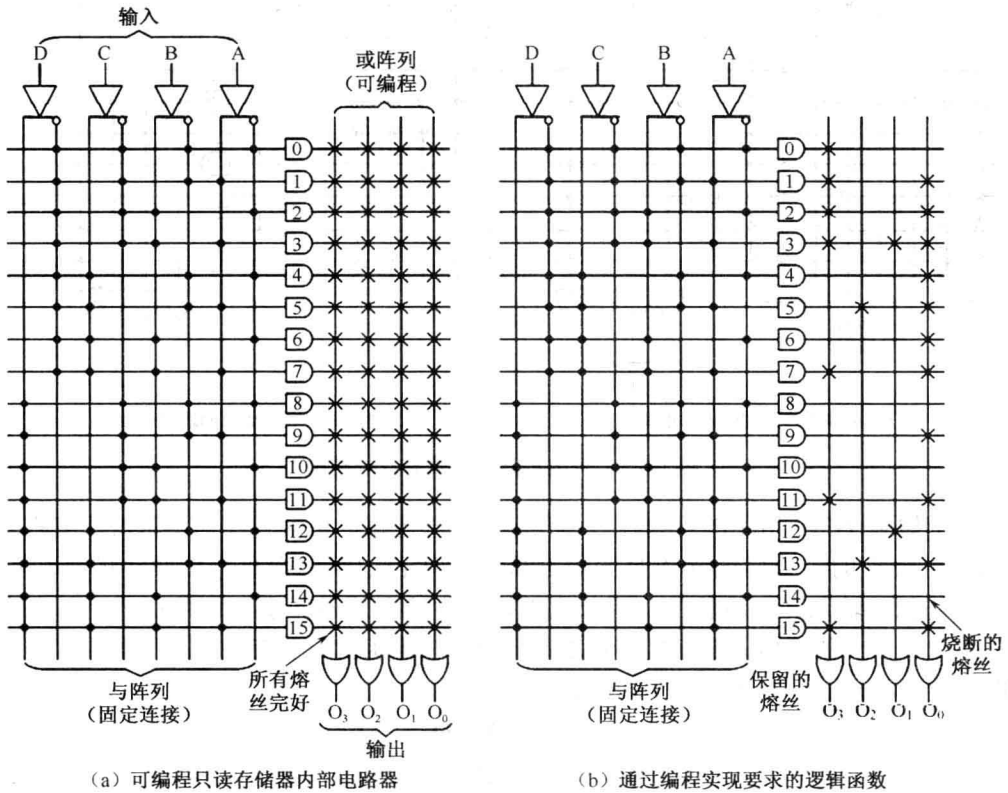


图 1-3-4 可编程只读存储器芯片的结构

图 1-3-4 (b) 中，4 个或门的输出分别为

$$\begin{aligned} O_3 &= AB + \overline{CD} \\ O_2 &= \overline{ABC} \\ O_1 &= \overline{ABCD} + \overline{ABCD} \\ O_0 &= A + B\overline{D} + \overline{CD} \end{aligned}$$

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 1-3-5 所示。

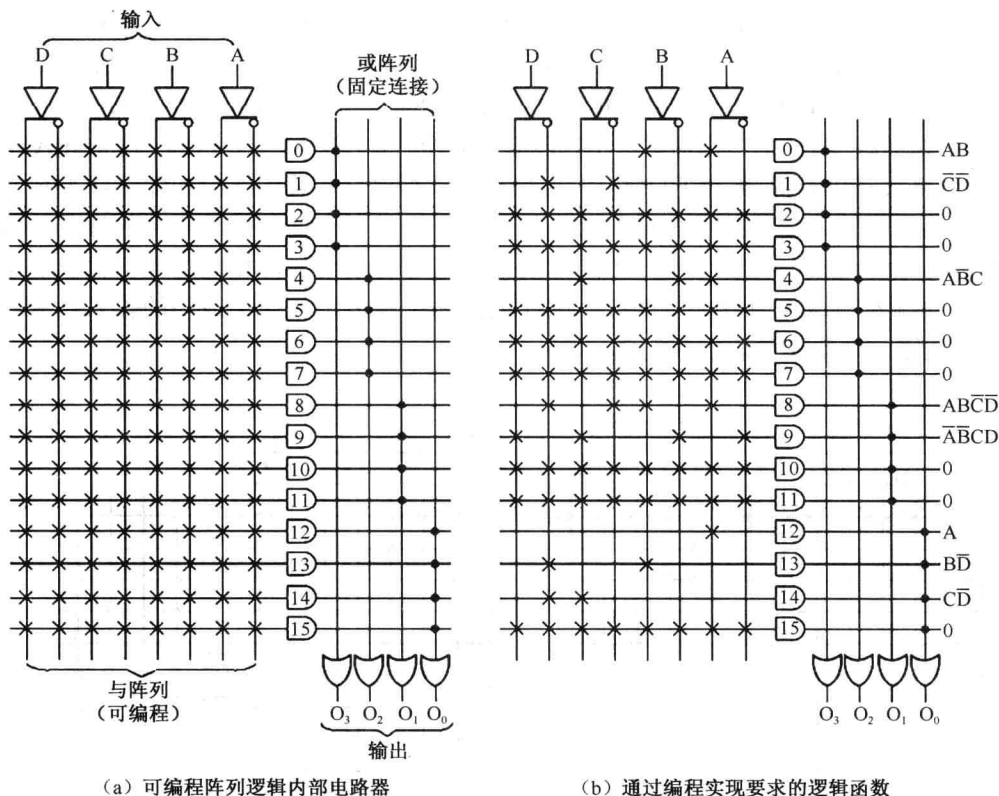


图 1-3-5 可编程阵列逻辑芯片的结构

相对于如图 1-3-4 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 1-3-5 (b) 中, 4 个或门的输出分别为

$$\begin{aligned} O_3 &= AB + \overline{CD} \\ O_2 &= \overline{ABC} \\ O_1 &= \overline{ABCD} + \overline{ABCD} \\ O_0 &= A + \overline{BD} + \overline{CD} \end{aligned}$$

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 1-3-5 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 1-3-6 给出了通用阵列逻辑 16V8 的结构图，它是 Lattice Semiconductor 公司的产品。这种芯片具有 8 个专用输入引脚（引脚 2~引脚 9）、两个特殊功能输入引脚（引脚 1：时钟输入 CLK，引脚 11：使能 OE）及 8 个输入/输出引脚（引脚 12~引脚 19）。

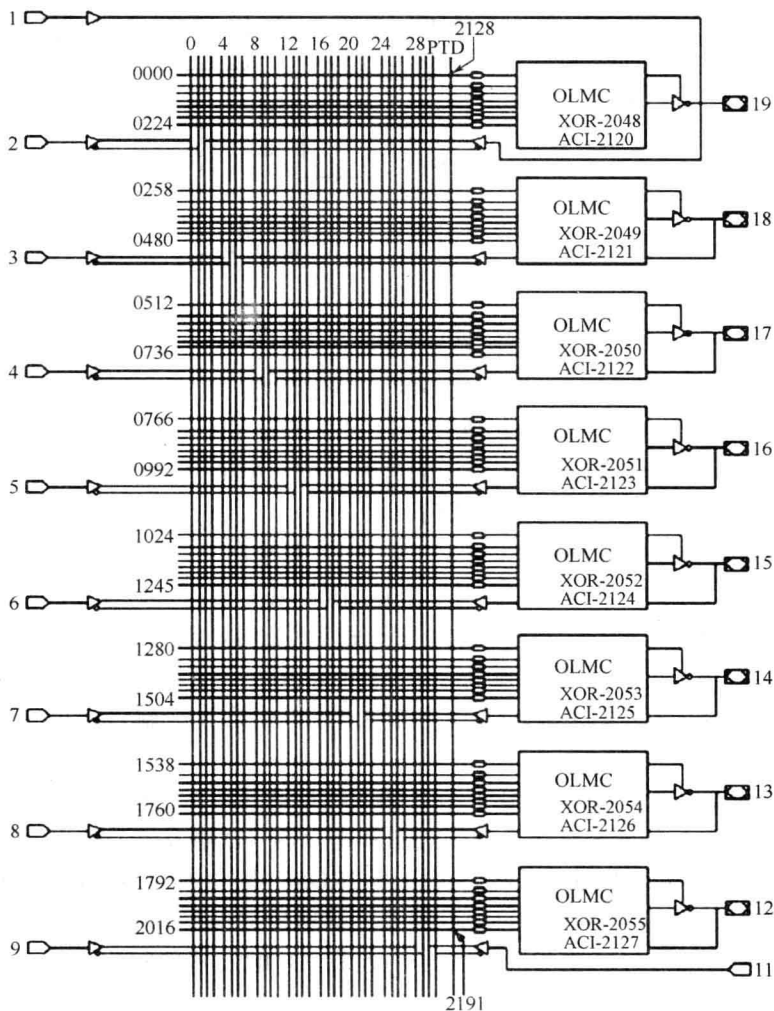


图 1-3-6 通用阵列逻辑 16V8 结构图

8 个专用输入引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 1-3-6 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 1-3-7 所示。每个输出逻辑宏单元具有 1 个 8 输入或门，它接收 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引脚，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。

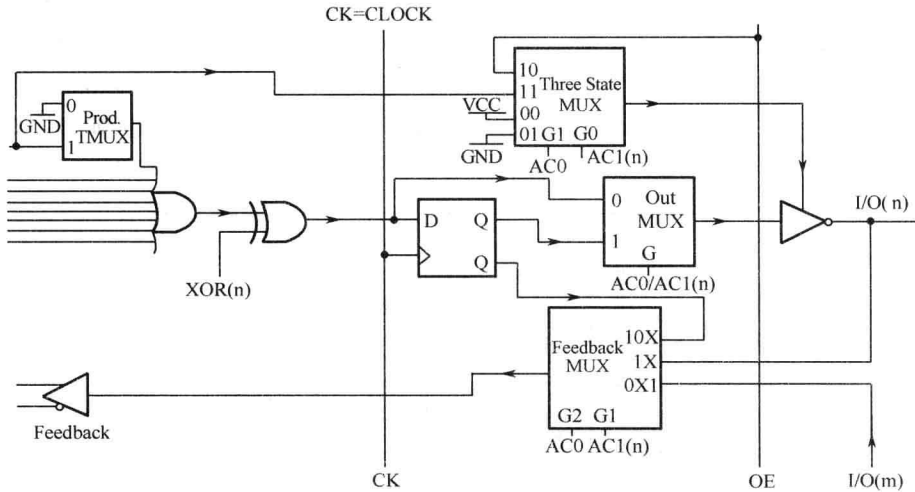


图 1-3-7 输出逻辑宏单元的电路图

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 VCC、接地、第 8 个输入乘积项及来自引脚 OE（引脚 11）的外部信号。当三态数据选择器选择 VCC 输入，三态反相器输出处于使能状态时，输入/输出引脚可以用做输出引脚；当选择接地输入，三态反相器输出处于高阻状态时，输入/输出引脚可以用做输入引脚。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 个输入乘积项或来自引脚 OE（引脚 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，它也由 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 1-3-7 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 时，8 输入或门的输出保持原来的极性；当 CONTROL=1 时，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）通常称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也已经很复杂了。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的，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。

4) 复杂可编程逻辑器件 CPLD 现在一般把所有超过某一集成度（如 1 000 门以上）