

可编程逻辑器件快速进阶丛书,

基于VHDL语言与Quartus II软件 的可编程逻辑器件应用与开发

郑燕 赫建国 党剑华 编著



国防工业出版社

National Defense Industry Press

可编程逻辑器件快速进阶丛书

基于VHDL语言与Quartus II软件的可编程逻辑器件应用与开发

郑燕 赫建国 党剑华 编著

国防工业出版社

· 北京 ·

图书在版编目(CIP)数据

基于 VHDL 语言与 Quartus 2 软件的可编程逻辑器件应用与开发/郑燕,赫建国,党剑华编著. —北京:国防工业出版社,2007.3

(可编程逻辑器件快速进阶丛书)

ISBN 978-7-118-04978-7

I. 基... II. ①郑...②赫...③党... III. ①硬件描述语言,VHDL—程序设计②可编程逻辑器件—应用软件,Quartus 2 IV. TP312 TP332.1

中国版本图书馆 CIP 数据核字(2007)第 012569 号

※

国防工业出版社 出版发行

(北京市海淀区紫竹院南路 23 号 邮政编码 100044)

天利华印刷装订有限公司印刷

新华书店经售

*

开本 787×1092 1/16 印张 13 字数 317 千字

2007 年 3 月第 1 版第 1 次印刷 印数 1—4000 册 定价 23.00 元

(本书如有印装错误,我社负责调换)

国防书店:(010)68428422

发行邮购:(010)68414474

发行传真:(010)68411535

发行业务:(010)68472764

内 容 简 介

本书内容分为两部分：第一部分（第1章~第7章）主要介绍基于可编程逻辑器件设计应用系统所需要的基础知识。其中包括：可编程逻辑器件的工作原理和类型，ALTERA公司生产的复杂可编程逻辑器件和现场可编程门阵列器件；电子设计自动化的概念，使用ALTERA公司提供的可编程逻辑器件的集成开发软件Quartus II进行电路设计和调试的整个过程；利用硬件描述语言VHDL编写设计代码的基本结构；VHDL的并行语句；VHDL的顺序语句；向可编程逻辑器件下载设计文件的模式、相关电路和操作步骤；有限状态机技术等。

第二部分（第8章~第10章）主要介绍一些比较深入的知识以及如何利用前面学习的基本知识实现应用系统的设计。其中包括：Quartus II可编程逻辑器件的集成开发软件中的原理图输入方式，开发软件提供的各种元件在电路设计中的使用，层次化设计的概念，利用这个概念可以把多个设计者完成的子系统组合成一个完整的系统；VHDL代码中实现层次化设计的方法，元件、函数及过程等，这些能使代码重复使用的方法，以提高设计效率并使得代码结构更加清晰；以及通过一个应用系统（信号产生器）的设计过程，介绍如何把前面的基础知识应用于具体的工作之中。

本书可作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的训练指导书，也可作为高校相关专业的教材和工程技术人员的参考书。

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。

作为传统数字系统设计中使用的主要器件,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

IV

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

目 录

第1章 可编程逻辑器件	1
1.1 数字集成电路的分类	2
1.1.1 标准逻辑器件	2
1.1.2 微处理器	2
1.1.3 专用集成电路	3
1.2 标准逻辑器件	4
1.2.1 数字电路的描述	4
1.2.2 组合逻辑电路	5
1.2.3 时序逻辑电路	7
1.3 可编程逻辑器件	8
1.3.1 可编程逻辑器件基础	8
1.3.2 可编程逻辑器件内部电路的描述	9
1.3.3 可编程逻辑器件内部电路的分类	10
1.4 ALTERA公司的可编程逻辑器件	15
1.4.1 复杂可编程逻辑器件(CPLD)	15
1.4.2 现场可编程门阵列(FPGA)器件	18
小结	22
习题	22
第2章 Quartus II 开发软件	24
2.1 Quartus II 简介	25
2.2 Quartus II 集成开发软件的安装	26
2.2.1 计算机系统配置	26
2.2.2 Quartus II 集成开发软件的安装	26
2.2.3 Quartus II 集成开发软件的授权	28
2.3 创建工程(Project)	30
2.4 设计的输入	34
2.5 设计的编译	37
2.6 设计的功能仿真	40
2.6.1 创建仿真波形文件	41
2.6.2 设计的功能仿真	43

小结	45
习题	45
第3章 VHDL 程序的结构	46
3.1 VHDL 的产生及发展	47
3.2 VHDL 程序的最简单结构	48
3.2.1 VHDL 程序的基本格式	48
3.2.2 VHDL 程序的仿真	49
3.3 实体	51
3.3.1 实体的格式	51
3.3.2 VHDL 的标识符	52
3.3.3 端口模式	53
3.3.4 端口的数据类型	53
3.4 结构体	55
3.5 VHDL 程序的结构	56
3.5.1 VHDL 程序的基本单元	56
3.5.2 VHDL 库	57
3.5.3 STD_LOGIC 数据类型	58
小结	60
习题	60
第4章 并行语句	61
4.1 数据类型	62
4.1.1 预定义的数据类型	62
4.1.2 数据类型转换	64
4.1 VHDL 的运算符	64
4.2.1 赋值运算符	64
4.2.2 逻辑运算符	65
4.2.3 算术运算符	65
4.2.4 关系运算符	66
4.2.5 移位操作符	66
4.2.6 并置运算符	67
4.3 基于逻辑门的组合逻辑电路设计	67
4.3.1 VHDL 中逻辑表达式的使用	67
4.3.2 VHDL 中真值表的使用	68
4.4 并行语句	71
4.4.1 并行语句概述	71
4.3.2 并行信号赋值语句	72

4.5 三态缓冲器	73
4.6 编码器	74
4.6.1 8线-3线编码器	74
4.6.2 优先编码器	76
4.7 译码器	77
4.8 数据选择器与数据分配器	79
4.8.1 数据选择器	79
4.8.2 数据分配器	80
小结	81
习题	82
第5章 顺序语句	83
5.1 顺序语句介绍	84
5.1.1 顺序信号赋值语句	84
5.1.2 条件(IF)语句	84
5.1.3 选择(CASE)语句	86
5.1.4 循环(LOOP)语句	86
5.1.5 空操作(NULL)语句	87
5.2 进程语句	87
5.2.1 进程语句的格式	87
5.2.2 进程语句的应用	88
5.2.3 WAIT 语句	91
5.3 时钟信号的描述	91
5.3.1 使用属性描述时钟	92
5.3.2 使用测定边沿的函数描述时钟	93
5.4 触发器	93
5.4.1 简单D触发器	93
5.4.2 具有异步复位和置位功能D触发器	94
5.4.3 具有同步复位和置位功能D触发器	95
5.5 计数器	96
5.5.1 加法计数器	96
5.5.2 具有同步置数和进位输出的计数器	98
5.5.3 分频器	99
5.6 循环语句	100
5.6.1 奇偶校验检测电路	100
5.6.2 连“0”检测电路	101
5.7 数据对象	102
5.7.1 常数	103

5.7.2 变量	103
5.7.3 信号	104
5.7.4 通用属性	105
小结.....	106
习题.....	106
第 6 章 可编程逻辑器件的编程/配置	108
6.1 编程/配置模式.....	108
6.2 ByteBlaster 下载电缆	110
6.2.1 ByteBlasterMV 下载电缆	111
6.2.2 JTAG 编程/配置模式	112
6.2.3 被动串行(PS)模式	113
6.3 可编程逻辑器件的编程/配置.....	114
6.3.1 编程器的设置	114
6.3.2 可编程逻辑器件管脚的设置	116
6.4 配置芯片	117
6.4.1 配置芯片的类型	117
6.4.2 配置电路	118
6.4.3 配置芯片 EPC2 的编程	119
小结.....	121
第 7 章 状态机.....	123
7.1 状态机概述	123
7.2 状态机的设计风格	124
7.2.1 组合逻辑电路控制输出	125
7.2.2 时序逻辑电路控制输出	126
7.3 增强状态机程序的可阅读性	128
7.4 单一进程的状态机程序	130
7.4.1 组合逻辑电路输出的毛刺现象	130
7.4.2 输出毛刺现象的克服	132
7.5 数字/模拟转换器 DAC7611 的使用	134
7.5.1 数字/模拟转换器 DAC7611 的介绍	134
7.5.2 数字/模拟转换器 DAC7611 的控制程序	136
小结.....	139
习题.....	140
第 8 章 深入使用 Quartus II 开发软件	142
8.1 原理图输入方法	142

8.1.1 创建原理图输入文件	143
8.1.2 Quartus II 集成开发软件提供的元件	143
8.1.3 原理图输入	144
8.2 层次化设计	146
8.3 存储器的设计	148
8.3.1 利用 VHDL 设计只读存储器	148
8.3.2 定制 LPM_ROM 初始化数据文件	150
8.3.3 定制 LPM_ROM 元件	151
小结	155
第 9 章 VHDL 的深入使用	156
9.1 包集(PACKAGES)	157
9.2 元件(COMPONENT)	158
9.2.1 元件概述	158
9.2.2 在应用程序中声明元件	159
9.2.3 在包集中声明元件	161
9.2.4 含有 GENERIC 参数元件使用	162
9.3 函数(FUNCTION)	164
9.3.1 函数概述	164
9.3.2 在应用程序中定义函数	165
9.3.3 在包集中定义函数	167
9.4 过程(PROCEDURE)	168
9.4.1 过程概述	168
9.4.2 在应用程序中定义过程	169
9.4.3 在包集中定义过程	170
小结	171
习题	171
第 10 章 信号产生器的设计	173
10.1 信号产生器概述	174
10.1.1 传统的信号产生器设计方案	174
10.1.2 基于微处理器和数字/模拟转换器的设计方案	174
10.2 直接数字合成技术	175
10.3 方案论证	176
10.3.1 DDS 信号产生器芯片	176
10.3.2 利用微处理器实现 DDS 信号产生器	178
10.3.3 利用可编程逻辑器件实现 DDS 信号产生器	179
10.4 系统设计	181

10.4.1 硬件电路的系统设计·····	181
10.4.2 系统代码框图·····	182
10.5 单元电路设计·····	182
10.5.1 数字/模拟转换器·····	182
10.5.2 波形数据表·····	184
10.5.3 相位累加器·····	185
10.5.4 显示电路·····	185
10.5.5 键盘电路·····	187
10.6 系统连调·····	191
10.6.1 各种时钟的产生·····	191
10.6.2 单元电路的组合·····	193
10.6.3 系统测试·····	195
小结·····	196
参考文献·····	197

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

$$Y_{\text{东西-绿灯}} = \overline{X_3}\overline{X_2}\overline{X_1}\overline{X_0} + \overline{X_3}\overline{X_2}\overline{X_1}X_0 + \overline{X_3}\overline{X_2}X_1\overline{X_0} + \overline{X_3}\overline{X_2}X_1X_0 + \overline{X_3}X_2\overline{X_1}\overline{X_0} + \overline{X_3}X_2\overline{X_1}X_0 + \overline{X_3}X_2X_1\overline{X_0} \quad (1-1)$$

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

$$Y_{\text{东西-绿灯}} = \overline{X_3}\overline{X_2} + \overline{X_3}\overline{X_1} + \overline{X_3}\overline{X_0} \quad (1-2)$$

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；如果需要一个实现 $Y_2 = X_1 + \overline{X_2}$ 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($\overline{X_3}\overline{X_2}$ 、 $\overline{X_3}\overline{X_1}$ 和 $\overline{X_3}\overline{X_0}$)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。