

基于VHDL与Quartus II软件 的可编程逻辑器件应用与开发 (第2版)

郑燕 赫建国 编著

本书配套源程序和资源可在 www.ndip.cn 上下载。继第一版读者热捧后的作者又一力作。



国防工业出版社

National Defense Industry Press

基于 VHDL 与 Quartus II 软件的 可编程逻辑器件应用与开发

(第 2 版)

郑燕 赫建国 编著



国防工业出版社

· 北京 ·

图书在版编目(CIP)数据

基于 VHDL 与 Quartus II 软件的可编程逻辑器件应用与开发/郑燕,赫建国编著. —2 版. —北京:国防工业出版社,2011.4

ISBN 978-7-118-07356-0

I. ①基… II. ①郑… ②赫… III. ①硬件描述语言, VHDL—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2011)第 038737 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 100048)
北京奥鑫印刷厂印刷
新华书店经售

*

开本 787 × 1092 1/16 印张 15 3/4 字数 384 千字
2011 年 4 月第 2 版第 1 次印刷 印数 1—4000 册 定价 33.00 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

第2版前言

在本书第1版出版之后的这些年里,无论 ALTERA 公司生产的可编程逻辑器件芯片,还是 Quartus II 开发软件都取得了许多进展。本书在原来的基础上,根据当前的进展做了修改,并根据本书在使用中的情况,添加了一些内容。

第1章介绍 ALTERA 公司生产的可编程逻辑器件芯片中,采用 EP2C35F672C-6 芯片代替了 EPF10K10LC84-4 芯片,因为后者已经停产。EP2C35F672C-6 芯片具有更多的逻辑资源,同时提供了 EPF10K10LC84-4 芯片没有的锁相环(PLL)和内嵌乘法器模块,另外该芯片还支持 Nios II 软核处理器系统。

第2章采用 Quartus II version 9.0 介绍开发软件的使用。新的版本不仅支持更多的可编程逻辑器件芯片的开发,提供更多的电路功能模块,而且能够使用较少的逻辑资源实现所要求的应用系统功能。

由于对使用的可编程逻辑器件芯片和开发软件进行了修改,因此,对后继章节的相关内容也做了相应的调整。

第2版添加了时钟产生电路一章。这一章介绍了能够产生较高时钟频率稳定度的晶体时钟产生电路;介绍了现场可编程阵列器件的片内锁相环模块的使用。

第2版还添加了 ALTERA 公司提出的数字系统 SOPC(System On Programmable Chip, SOPC)解决方案。该方案使得处理器能够配置到现场可编程阵列器件之中,这样的处理器被称作为 Nios II 软核处理器。这种解决方案使得一块芯片将同时获得基于 VHDL 设计的可编程逻辑器件应用电路具有数据传送速度快的优点和基于 C 语言设计的微处理器应用电路具有数据处理能力强的优点。

朱旭花、赵林森和郑慧娟以及刘娟同志参与了本书的编写工作,在这里向他们表示衷心的感谢。

书中源代码可从 www.ndip.cn 网站下载。

编著者
2011年1月

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。“数字电路逻辑设计”课程目前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。读者也许有这样的体会,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小的台阶,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器件(FPGA)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。本书将要介绍的 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。在与

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。

由于时间仓促,水平有限,书中难免存在疏漏或不妥之处,欢迎读者批评指正。

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学院许多老师的支持和帮助,在这里向他们表示衷心感谢。在本书的编写过程中,也参考了许多专家和学者的著作和研究成果,在这里也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,加之作者的水平有限,书中的错误与不妥在所难免,敬请读者批评指正。E-mail:jtawang @ ndip.cn

目 录

第1章 可编程逻辑器件	1
目标	1
引言	1
1.1 数字集成电路的分类	1
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 标准逻辑器件基础知识	3
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	6
1.3 可编程逻辑器件基础知识	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	11
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	15
1.4.2 现场可编程门阵列器件	19
小结	23
习题	23
第2章 Quartus II 开发软件	24
目标	24
引言	24
2.1 Quartus II 简介	24
2.2 Quartus II 集成开发软件的工作窗口	26
2.3 创建工程	27
2.4 设计的输入	32
2.5 设计的编译	34
2.6 设计的功能仿真	36
2.6.1 创建仿真波形文件	36

2.6.2 设计的功能仿真	39
2.6.3 设计的时序仿真	41
小结	42
习题	42
第3章 VHDL 程序的结构	44
目标	44
引言	44
3.1 VHDL 语言的产生及发展	44
3.2 VHDL 程序的最简单结构	46
3.2.1 VHDL 程序的基本格式	46
3.2.2 VHDL 程序的仿真	47
3.3 实体	49
3.3.1 实体的格式	49
3.3.2 VHDL 语言的标识符	50
3.3.3 端口模式	51
3.3.4 端口的数据类型	52
3.4 结构体	53
3.5 VHDL 程序的结构	54
3.5.1 VHDL 程序的基本单元	54
3.5.2 VHDL 库	55
3.5.3 STD_LOGIC 数据类型	56
小结	57
习题	58
第4章 并行语句	59
目标	59
引言	59
4.1 数据类型	60
4.1.1 预定义的数据类型	60
4.1.2 数据类型转换	61
4.2 VHDL 的运算符	62
4.2.1 赋值运算符	62
4.2.2 逻辑运算符	63
4.2.3 算术运算符	63
4.2.4 关系运算符	63
4.2.5 移位操作符	64
4.2.6 并置运算符	64

4.3	基于逻辑门的组合逻辑电路设计	65
4.3.1	VHDL 中逻辑表达式的使用	65
4.3.2	VHDL 中真值表的使用	66
4.4	并行语句基础知识	69
4.4.1	概述	69
4.3.2	并行信号赋值语句	69
4.5	三态缓冲器	70
4.6	编码器	71
4.6.1	8 线-3 线编码器	71
4.6.2	优先编码器	73
4.7	译码器	74
4.8	数据选择器与数据分配器	75
4.8.1	数据选择器	76
4.8.2	数据分配器	77
4.9	奇偶产生/校验器	78
	小结	79
	习题	79
第 5 章	顺序语句	81
	目标	81
	引言	81
5.1	顺序语句介绍	81
5.1.1	顺序信号赋值语句	82
5.1.2	条件 (IF) 语句	82
5.1.3	选择 (CASE) 语句	83
5.1.4	循环 (LOOP) 语句	84
5.1.5	空操作 (NULL) 语句	84
5.2	进程语句	85
5.2.1	进程语句的格式	85
5.2.2	进程语句的应用	85
5.2.3	WAIT 语句	88
5.3	时钟信号的描述	89
5.3.1	使用属性描述时钟	89
5.3.2	使用测定边沿的函数描述时钟	90
5.4	触发器	90
5.4.1	简单 D 触发器	91
5.4.2	具有异步复位和置位功能的 D 触发器	91
5.4.3	具有同步复位和置位功能的 D 触发器	92

5.4.4	具有异步复位和置位功能的 J-K 触发器	93
5.5	计数器	95
5.5.1	加法计数器	95
5.5.2	具有同步置数 and 进位输出的计数器	96
5.6	分频器	97
5.6.1	通用整数分频器	97
5.6.2	占空比为 1/2 的奇数分频器	99
5.6.3	半整数分频器	100
5.7	循环语句	102
5.7.1	奇偶校验检测电路	102
5.7.2	连“0”检测电路	103
5.8	数据对象	104
5.8.1	常数	104
5.8.2	变量	105
5.8.3	信号	105
5.8.4	通用属性	107
小结		108
习题		108
第 6 章	可编程逻辑器件的编程/配置	109
目标		109
引言		109
6.1	编程/配置模式	109
6.2	ByteBlaster 下载电缆	111
6.2.1	ByteBlasterMV 下载电缆	111
6.2.2	JTAG 编程/配置模式	113
6.2.3	被动串行方式模式	114
6.3	可编程逻辑器件的编程/配置	115
6.3.1	编程器的设置	115
6.3.2	可编程逻辑器件管脚的设置	117
6.4	配置芯片	119
6.4.1	配置芯片的类型	119
6.4.2	配置电路	120
6.4.3	配置芯片 EPC2 的编程	121
小结		124
第 7 章	状态机	125
目标		125

引言	125
7.1 介绍	125
7.2 状态机的设计风格	126
7.2.1 组合逻辑电路控制输出	127
7.2.2 时序逻辑电路控制输出	128
7.3 增强状态机程序的可阅读性	130
7.4 单一进程的状态机程序	131
7.4.1 组合逻辑电路输出的毛刺现象	131
7.4.2 输出毛刺现象的克服	133
7.4.3 序列信号检测器的设计	135
7.5 数字/模拟转换器 DAC7611 的使用	137
7.5.1 数字/模拟转换器 DAC7611 的介绍	137
7.5.2 数字/模拟转换器 DAC7611 的控制程序	139
小结	143
习题	144
第 8 章 Quartus II 开发软件深入使用	146
目标	146
引言	146
8.1 原理图输入方法	146
8.1.1 创建原理图输入文件	147
8.1.2 Quartus II 开发软件提供的元件	147
8.1.3 原理图输入	148
8.2 层次化设计	150
8.3 存储器的设计	152
8.3.1 利用 VHDL 设计 ROM	153
8.3.2 定制 LPM_ROM 初始化数据文件	154
8.3.3 定制 LPM_ROM 元件	155
小结	161
第 9 章 VHDL 的深入使用	162
目标	162
引言	162
9.1 包集	162
9.2 元件	164
9.2.1 概述	164
9.2.2 在应用程序中声明元件	165
9.2.3 在包集中声明元件	166

9.2.4 含有 GENERIC 参数元件使用	168
9.3 函数	170
9.3.1 概述	170
9.3.2 在应用程序中定义函数	171
9.3.2 在包集中定义函数	173
9.4 过程	174
9.4.1 概述	174
9.4.2 在应用程序中定义过程	175
9.4.2 在包集中定义过程	175
小结	176
习题	176
第 10 章 时钟电路	178
目标	178
引言	178
10.1 时钟产生电路	179
10.1.1 基于 555 定时器的时钟产生电路	179
10.1.2 晶体时钟产生电路	179
10.2 锁相环	181
10.2.1 锁相环的工作原理	181
10.2.2 Cyclone II 系列芯片中的锁相环模块	182
10.2.3 altpll 宏功能模块	183
10.2.4 锁相环模块的工作模式	184
10.3 锁相环模块的创建	185
小结	194
第 11 章 信号产生器的设计	196
目标	196
引言	196
11.1 概述	197
11.1.1 传统的信号产生器设计方案	197
11.1.2 基于微处理器和数字/模拟转换器的设计方案	197
11.2 直接数字合成技术	198
11.3 方案论证	199
11.3.1 DDS 信号产生器芯片	199
11.3.2 利用微处理器实现 DDS 信号产生器	201
11.3.3 利用可编程逻辑器件实现 DDS 信号产生器	202
11.4 系统设计	204

11.4.1	硬件电路的系统设计	204
11.4.2	系统代码框图	205
11.5	单元电路设计	205
11.5.1	数字/模拟转换器	205
11.5.2	波形数据表	207
11.5.3	相位累加器	208
11.5.4	显示电路	208
11.5.5	键盘电路	210
11.6	系统连调	214
11.6.1	各种时钟的产生	214
11.6.2	单元电路的组合	216
11.6.3	系统测试	219
小结		219
第 12 章	Nios II 软核处理器	220
目标		220
引言		220
12.1	概述	220
12.2	配置 Nios II 软核处理器系统	222
12.2.1	创建 Nios II 软核处理器系统	222
12.2.2	配置 Nios II 软核处理器系统	225
12.3	产生 Nios II 软核处理器系统	230
12.3.1	产生 Nios II 软核处理器系统模块	230
12.3.2	Nios II 软核处理器系统的产生	230
12.4	创建 Nios II IDE 环境下的应用工程	232
12.4.1	Nios II IDE 工程创建	232
12.4.2	C 语言源文件的编辑	233
12.4.3	C 语言源文件的编译	236
12.5	配置目标现场可编程阵列芯片器件	237
小结		237
参考文献		238

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它们的特点
- 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,我们是仅仅只学习新方法,还是设法用老方法去解决新问题。

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,同时我们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,但是作为理论基础的基本原理并没有改变。

现代数字系统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,因此它们的生产成本低廉,价格便宜。如果在设计不很复杂时,这些器件仍然是有用的。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。“数字电路逻辑设计”课程目前仍然以标准逻辑器件为主进行讲授。在 1.2 节一起回顾“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技术领域,其中数字计算机是最著名和应用最广泛的。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(I/O)接口芯片等。在大多数情况下,凡是人能干的,计算机都能干,而且计算机还能做得更快更精确。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设计的数字系统工作速度快。

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了单片机,有的文献上它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是非常高的,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 标准逻辑器件基础知识

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: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和输出之间的关系的逻辑电路。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:布尔代数是利用数学表达式来描述电路输入和输出之间的关系;真值表是利用数字来描述电路输入和输出之间的关系;电路图是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(Hardware Description Language, HDL)是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、黄灯占一个状态。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 所列。

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
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