

逻辑设计基础

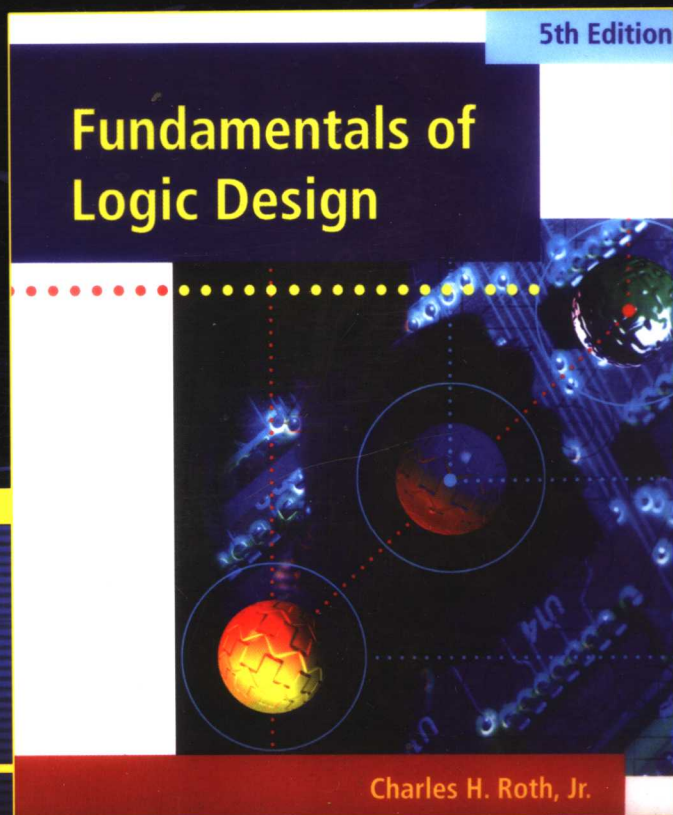
(原书第5版)

Fundamentals of Logic Design

(5th Edition)

(美) Charles H. Roth, Jr. 著

解晓萌 黎永志 王坤 等译



机械工业出版社
China Machine Press



电子与电气工程丛书

逻辑设计基础

(原书第5版)

Fundamentals of Logic Design (5th Edition)

(美) Charles H. Roth, Jr. 著

解晓萌 黎永志 王坤 等译



机械工业出版社
China Machine Press

本书详细地阐述了理解逻辑设计基本概念所必需的理论, 主要内容包括布尔代数、逻辑门设计、锁存器、触发器、状态机等基本概念。通过将触发器和逻辑门电路相结合, 讲解如何设计计数器、加法器、序列检测器以及与之类似的电路。本书还介绍了VHDL硬件描述语言在组合逻辑设计、时序逻辑设计和简单数字系统设计中的应用。

本书适合作为高等院校电子工程、计算机科学及相关专业的本科生和研究生的教材, 也适合工程专业人员参考。

Charles H. Roth, Jr., Fundamentals of Logic Design (5th Edition)

ISBN: 0-534-37804-8

Copyright © 2004 by Brooks/Cole, a division of Thomson Learning.

Original language published by Thomson Learning (a division of Thomson Learning Asia Pte Ltd). All rights reserved. 本书原版由汤姆森学习出版集团出版。版权所有, 盗印必究。

China Machine Press is authorized by Thomson Learning to publish and distribute exclusively this simplified Chinese edition. This edition is authorized for sale in the People's Republic of China only (excluding Hong Kong, Macao SARs and Taiwan). Unauthorized export of this edition is a violation of the Copyright Act. No part of this publication may be reproduced or distributed by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

本书中文简体字翻译版由汤姆森学习出版集团授权机械工业出版社独家出版发行。此版本仅限在中华人民共和国境内(不包括中国香港、澳门特别行政区及中国台湾)销售。未经授权的本书出口将被视为违反版权法的行为。未经出版者预先书面许可, 不得以任何方式复制或发行本书的任何部分。

981-265-628-6

版权所有, 侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号: 图字: 01-2003-6968

图书在版编目(CIP)数据

逻辑设计基础(原书第5版)/(美)罗思(Roth, C. H.)著; 解晓萌等译. -北京: 机械工业出版社, 2005.7

(电子与电气工程丛书)

书名原文: Fundamentals of Logic Design, 5th Edition

ISBN 7-111-16379-6

I. 逻… II. ①罗… ②解… III. 数字系统-逻辑设计-高等学校-教材 IV. TP302.2

中国版本图书馆CIP数据核字(2005)第025239号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑: 傅志红 洪海丽

北京瑞德印刷有限公司印刷·新华书店北京发行所发行

2005年7月第1版第1次印刷

787mm × 1092mm 1/16 · 36印张

印数: 0 001 - 4 000册

定价: 69.00元(附光盘)

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换
本社购书热线: (010) 68326294

译者序

Charles H.Roth的*Fundamentals of Logic Design*是有关数字逻辑设计的著作。毫无疑问，它是关于数字逻辑设计的最重要著作之一，在国外有很高的地位和影响力，是国外逻辑设计的经典教材。伴随着数字逻辑设计的发展，*Fundamentals of Logic Design*经过了多次修改和补充，现在已经是第5版了，其影印版——*Fundamentals of Logic Design*（英文版·第5版）已经由机械工业出版社出版。

本书按照数字系统逻辑设计的第一门课程进行编写。编写是基于这样的前提：学生应该理解和彻底学习本课程中的基础概念。由于集成电路技术的发展，使得更多的部件可以集成到一个芯片上，数字系统变得越来越复杂。本书介绍了用于设计这样复杂系统所需要的硬件描述语言VHDL以及VHDL在逻辑设计方面的应用，着重强调了VHDL语句与对应的数字硬件之间的关系。最后，介绍了如何用计算机辅助设计实现逻辑设计过程的自动化。

然而，作者不仅深入讨论了数字逻辑设计的原理、方法和应用，而且为了让学生或者自学者能够深刻体会数字逻辑设计的精髓，有几章包含有仿真习题或实验设计习题，这些习题提供了对逻辑电路进行设计以及对其操作进行测试的机会。光盘上提供的逻辑模拟器SimUaid可以用于逻辑设计的检验。

对于中国读者来说，本书既适合计算机科学专业的学生，也适合电子电路设计等工程方面的学生和设计人员。对于没有电子电路基础的计算机系学生或其他学生来说，附录A中包含了与逻辑门电路方面有关的内容，以便这一类的学生或工程设计人员对此有些了解。除此之外，本书既可以用于标准讲解式课程，也可用于自择内容式课程。除了标准的学习内容及习题外，书中还提供学习指导及其他辅助内容以方便自学。

本书作者在网上提供了英文版的勘误，更正了书中的错误，并修改了少量的程序示例。我们在翻译时与作者联系，对错误的修改取得了一致的意见，有关更正反映在本书中。由于这种原因，本书在个别地方的叙述可能与某些读者手中的英文原书有异。

本书由华南理工大学的解晓萌、黎永志、宋华、王坤和易成就共同完成翻译，由解晓萌统稿审校。具体分工为：解晓萌完成第1至4章，王坤完成第5至8章，黎永志完成第9至13章，宋华完成第14至18章，易成就完成第19至20章和附录。

由于译者水平有限，译文中难免出现不妥之处，敬请读者批评指正。

译者

2005年3月

前 言

学习本书之后，读者应该可以应用开关理论来解决逻辑设计问题。这意味着既要学习开关电路基础理论，也要学习如何应用这些理论。在简单介绍之后将学习布尔代数，它是分析和综合开关电路所需的基本数学工具。由对某个问题的语言表述开始，学习如何设计逻辑门电路，该逻辑门的输入端与输出端信号之间具有特定的关系。接着学习触发器的逻辑特性，它可以作为时序开关电路的存储器件。通过将触发器和逻辑门电路相结合，学习如何设计计数器、加法器、序列检测器以及与之类似的电路。还将学习VHDL硬件描述语言，及该语言在组合逻辑设计、时序逻辑设计和简单数字系统设计中的应用。

本书新版本在第4版基础上做了下面一些改进：书中的内容重新进行组织以便提供更好的教学顺序；去除了书中陈旧的内容；锁存器和触发器等章节重新进行了编写；本书更强调如何使用可编程逻辑器（PLD），包括可编程门阵列和复杂PLD；每一章增加了新的练习和习题；为了表述得更清楚重新编写了某些章节；书中增加了3章有关硬件描述语言VHDL的内容，并且重点强调它在逻辑电路仿真及计算机辅助设计中所起的作用。

本书既可以作为标准讲解式教材，也可作为自择内容式教材。除了标准的学习内容及习题外，书中还有学习指导及其他辅助内容以方便自学。本书内容分为20章，这些章形成一个逻辑顺序，也就是说掌握前一章的内容通常是学习后续章节的先决条件。每一章包含四部分，首先是学习目标，它准确说明了学习本章后应该达到的目标。接着是学习指导，它包含学习任务和学习问题，当学习某一章后，应该可以写出这些问题的答案。紧接其后的是学习内容及习题部分，该部分与传统教材类似。当结束某章的学习后，应该重新回顾学习目标，确信已经达到这些目标。

需要学习的内容分为3大主要部分，前9章讲解布尔代数和组合逻辑电路设计问题。第11章至第16章、第18章和第19章主要关注的是时序逻辑电路的分析和设计问题，这包括算术运算电路的设计。第10、17和20章介绍硬件描述语言VHDL及其在逻辑设计中的应用。

由于计算机在逻辑设计过程中发挥着重要的作用，因此将计算机应用引入到第一门逻辑设计课程中是非常重要的。本书提供的光盘中有一个计算机辅助逻辑设计程序，该程序名字为LogicAid，它可以使学生方便地根据小项、真值表及状态转换表推导出逻辑表达式，使学生从大量枯燥的计算中解脱出来，使得在较短时间内解决更复杂的逻辑设计问题成为可能。LogicAid还提供有关卡诺图及状态转换图推导的指导和帮助。

有几章包含仿真习题或实验习题，这些习题提供对逻辑电路进行设计及其操作进行测试的机会。光盘上提供的逻辑仿真器SimUaid可以用于逻辑设计的检验。测试所需的实验设备可以是带有触发器和逻辑门的面包板，也可以是带有可编程逻辑器件的电路板。若无法获得这些设备，可以用SimUaid对实验习题进行仿真，或者当作一个设计问题布置下去，这对于第8、16和20章尤为重要，因为这些章中的综合设计问题有助于复习和整合前面几章所讲的内容。

随着集成电路技术的发展，使得更多的部件可以集成到一个芯片上，数字系统会变得越来越复杂，设计这样的复杂系统需要硬件描述语言例如VHDL。本书介绍VHDL在逻辑设计方

面的应用,着重强调VHDL语句与对应的数字硬件之间的关系。VHDL允许在用逻辑部件实现数字硬件之前,先在更高层次上对数字硬件进行描述和仿真。用于综合的计算机程序可以把数字系统的VHDL描述转换成相应的逻辑器件组合及它们之间的互连。虽然使用这样的计算机辅助设计工具有助于实现逻辑设计过程的自动化,我们仍坚信在写VHDL代码之前,先理解基础逻辑器件功能及其时序是非常重要的。先让学生手工实现数字逻辑,可以使学生更全面地理解VHDL的功能及不足之处。

本书针对数字系统逻辑设计的第一门课程进行编写。本书的编写基于这样的前提:学生应该理解和彻底学习本课程中的基础概念。这些基础概念的例子有:应用布尔代数描述逻辑电路中的信号和连接、使用系统方法化简逻辑电路、用简单器件互连实现更复杂的逻辑功能、根据时序图和状态转换图分析时序逻辑电路,以及使用控制电路控制数字系统中的事件顺序。

本书尝试在理论与应用之间取得平衡,基于这个原因,本书并没有过分强调开关理论中的数学。但是,它仍然提供了正确理解逻辑设计中基本概念所需的理论。在学习本书之后,学生应该为学习更高层次的数字系统设计课程做好了准备,这些课程强调一些更直观的概念,例如数字处理算法的开发、数字系统中子系统的划分、以及使用当前已有的硬件实现数字系统等。另外学生也为学习开关理论方面的高级课程做好了准备,这些课程进一步扩展本书介绍的某些理论概念。

自本书第1版出版之后,虽然用于实现数字系统的技术有了很大变化,但用于逻辑设计的基本原理并没有改变,真值表和状态转换表依旧用于确定逻辑电路的行为,布尔代数也依旧是逻辑设计的基本数学工具。甚至当可编程逻辑器取代单独的门电路和触发器之后,为了在更小的PLD上实现逻辑等式,对逻辑等式的化简依旧是值得的。良好的状态分配依然重要,因为若没有好的分配,一个逻辑等式可能需要更大的PLD来实现。

本书既适合计算机科学专业的学生,也适合工程专业的学生。与逻辑门电路方面有关的内容包含在附录A中,以便于计算机科学的学生或其他没有电子电路基础的学生跳过这些内容。本书做了一定的安排,即使学生跳过有关Quine-McCluskey方法的第6章,也不会影响本书的连贯性。关于VHDL的三章可以按正常顺序学习,即学习其他章后再学习这些章,也可以全部跳过这些章。

虽然在开关理论或逻辑设计领域有许多书籍可供我们选择,但本书的编写最初是为了迎合自择内容课程的需要,即学生只学习与他们有关的内容。每一章都配有大量的课堂测验用于自择内容环境下的学习。根据学生的反馈意见,我们对每一章都做了修改。

学习指导和本书的内容根据学生要求做了扩充,使得学生可以在没有讲稿的情况下学习本书,也使得几乎所有学生都可以掌握目标中的所有内容。在编写本书的过程中我们还编写了一些补充材料。教师手册就是一个补充材料,该手册中包括对标准式课程和自择内容式课程的建议、每一章的小测验、对有关实验设备和方法的建议等。该教师手册中还包括习题、单元小测验及实验练习的答案。

用于自学目的的书不能因为要提高效率而简单编写,它必须经过多次试用及修订才能完成。我希望向参与此过程的许多教授和学生表示感谢。特别感谢David Brown博士,他与我一起工作并从事自择内容课程的教学,他给我提出了许多建议来丰富本书内容。我还特别感谢研究生教学助理Mark Story,他为新版书设计了大量新的习题并提供了答案,他还在提高本书内容的一致性与表达的清晰性方面提供了许多建议。

如何自学本书

如果你想学习本书的所有内容并达到精通的水平，那么对每一章我们推荐下面的学习流程：

1. 阅读每一章的学习目标。这些学习目标是对学习本章后应该掌握的内容的简明概括。

2. 完成学习指导。在学习此章的每个部分之后，写出对应的学习指导问题的答案。学习指导中的习题通常有助于强调每个部分的重点，也可以指导你更好地理解其中的难点。如果不能回答出学习指导中的某些问题，这表明在继续学习之前，你需要再次学习该问题对应章节的内容。在本书的最后给出了部分学习指导问题的答案，剩余问题的答案通常可以在书中找到。

3. 某些章（第3、5、6、11、13、14和18章）含有一个或多个循序渐进练习。每个循序渐进练习一步一步地引导你解决本书中遇到的比较难的题。

4. 完成每一章后面的习题。用提供的答案检查你的答案，若你的答案有错误则需要重做该习题。

5. 重新阅读学习目标，确保你已经达到所有目标。若不确定的话，请复习书中相关的内容。

6. 如果在自择内容式课程中使用本书，只有当你通过了为某一章准备的测验后，才能继续下一章。这些测验的目的是确定你已经掌握了该章的学习目标，这样才能继续学习下一章。测验中的问题直接与该章的学习目标有关。因此，如果你已经完成学习指导，并且做出学习指导中所有的问题及布置的习题，那么你将轻松地通过这个测验。

目 录

译者序
前言

第1章 数制系统及其转换入门	1
1.1 数字系统与开关电路	3
1.2 数制系统及其转换	5
1.3 二进制运算	8
1.4 负数的表示	10
1.4.1 补码的加法	12
1.4.2 反码的加法	13
1.5 二进制编码	14
习题	17
第2章 布尔代数	19
2.1 简介	23
2.2 基本运算	24
2.3 布尔表达式及真值表	26
2.4 基本定理	27
2.5 交换律、结合律与分配律	29
2.6 化简定理	30
2.7 展开及因式分解	32
2.8 德摩根定律	34
习题	35
布尔代数的定律及定理	39
第3章 布尔代数(续)	41
3.1 表达式的展开及因式分解	44
3.2 异或与同或运算	46
3.3 蕴涵定理	48
3.4 开关表达式的代数化简	49
3.5 等式成立的证明	51
循序渐进练习	53
习题	55
第4章 布尔代数的应用、小项与大项展开式	59
4.1 文字描述向布尔等式的转换	63

4.2 用真值表设计组合逻辑	64
4.3 小项与大项展开式	65
4.4 通用的小项与大项展开式	68
4.5 不完全给定函数	70
4.6 真值表构建举例	72
4.7 二进制加法器与减法器的设计	74
习题	78
第5章 卡诺图	85
5.1 开关函数的最简形式	92
5.2 两变量卡诺图和三变量卡诺图	93
5.3 四变量卡诺图	97
5.4 用基本首要蕴涵项确定最简表达式	99
5.5 五变量卡诺图	104
5.6 卡诺图的其他应用	105
5.7 卡诺图的其他形式	107
循序渐进练习	108
习题	112
第6章 奎因-麦克拉斯基法	117
6.1 首要蕴涵项的确定	121
6.2 首要蕴涵项表	123
6.3 Petrick法	126
6.4 不完全给定函数的化简	127
6.5 采用代入变量的卡诺图化简	128
6.6 小结	129
循序渐进练习	130
习题	133
第7章 多级门电路/与非门和或非门	135
7.1 多级门电路	138
7.2 与非门和或非门	141
7.3 使用与非门和或非门的两级电路设计	143
7.4 使用与非门和或非门的多级电路设计	146

7.5 用门的替代符号转换电路	146	11.2 S-R锁存器	233
7.6 两级、多输出电路的设计	148	11.3 门控D锁存器	236
7.7 多输出与非门和或非门电路	152	11.4 边沿触发D触发器	237
习题	153	11.5 S-R触发器	239
第8章 用门电路设计和仿真组合电路	157	11.6 J-K触发器	240
8.1 复习组合电路的设计	159	11.7 T触发器	242
8.2 使用有限扇入门设计电路	160	11.8 带有附加输入端的触发器	243
8.3 门延迟和时序图	162	11.9 小结	244
8.4 组合电路中的冒险	163	习题	245
8.5 逻辑电路的仿真与测试	166	循序渐进练习	250
习题	168	第12章 寄存器与计数器	253
设计习题	170	12.1 寄存器和寄存器传输	256
第9章 多路选择器、译码器和可编程逻辑器件	175	12.2 移位寄存器	260
9.1 简介	178	12.3 二进制计数器的设计	264
9.2 多路选择器	178	12.4 其他顺序的计数器	268
9.3 三态缓冲器	180	12.5 用S-R触发器和J-K触发器设计计数器	272
9.4 译码器和编码器	182	12.6 触发器输入方程的导出-小结	275
9.5 只读存储器	184	习题	278
9.6 可编程逻辑器件	188	第13章 时序电路分析	285
9.6.1 可编程逻辑阵列	188	13.1 时序奇偶校验器	289
9.6.2 可编程阵列逻辑	191	13.2 信号跟踪及时序图分析	291
9.7 复杂可编程逻辑器件	193	13.3 状态转换表与状态转换图	294
9.8 现场可编程门阵列	194	13.4 时序电路的通用模型	300
习题	197	循序渐进练习	303
第10章 VHDL入门	201	习题	307
10.1 组合逻辑电路的VHDL描述	204	第14章 状态转换图与状态转换表的导出	313
10.2 多路选择器的VHDL模型	206	14.1 序列检测器的设计	315
10.3 VHDL模型	208	14.2 更复杂的设计问题	319
10.4 信号与常量	212	14.3 建立状态转换图的方法	323
10.5 数组	213	14.4 串行数据代码的转换	327
10.6 VHDL运算符	215	14.5 字母数字状态转换图标注	330
10.7 包与库	217	循序渐进练习	332
10.8 IEEE标准逻辑	218	习题	337
10.9 VHDL代码的编译与仿真	220	第15章 状态转换表的化简及状态赋值	343
习题	222	15.1 冗余状态的消除	349
设计习题	224	15.2 等价状态	351
第11章 锁存器与触发器	229	15.3 用蕴涵表确定状态的等价	352
11.1 简介	232		

15.4 等价的时序电路	355	仿真习题	433
15.5 不完全给定的状态转换表	356	第18章 算术运算电路	435
15.6 触发器输入方程式的导出	357	18.1 带累加器的串行加法器	436
15.7 等价的状态赋值	360	18.2 并行乘法器的设计	439
15.8 状态赋值的方法	363	18.3 二进制除法器的设计	443
15.9 单跃变状态赋值的应用	367	循序渐进练习	447
习题	368	习题	450
第16章 时序电路设计	377	第19章 使用SM图的状态机设计	457
16.1 时序电路设计方法小结	378	19.1 状态机流程图	458
16.2 设计实例-代码转换器	379	19.2 SM图的推导	462
16.3 迭代电路的设计	381	19.3 SM图的实现	467
16.4 用ROM和PLA设计时序电路	385	习题	471
16.5 用CPLD设计时序电路	388	第20章 用于数字系统设计的VHDL	475
16.6 用FPGA设计时序电路	391	20.1 串行加法器的VHDL代码	477
16.7 时序电路的仿真与测试	393	20.2 二进制乘法器的VHDL代码	478
16.8 计算机辅助设计概述	397	20.3 二进制除法器的VHDL代码	487
设计习题	399	20.4 掷骰子游戏模拟器的VHDL代码	488
补充习题	404	20.5 要点总结	491
第17章 用于时序逻辑的VHDL	407	习题	492
17.1 用VHDL进程建立触发器模型	410	实验设计习题	494
17.2 用VHDL进程建立寄存器和计数器 模型	413	附录A MOS及CMOS逻辑	497
17.3 用VHDL进程建立组合逻辑模型	417	附录B VHDL语言小结	503
17.4 时序机建模	419	附录C 定理的证明	509
17.5 VHDL代码的综合	424	附录D 国内外逻辑门图形符号对照	511
17.6 更多关于进程和顺序语句的内容	427	参考文献	513
习题	428	部分学习指导与习题的答案	515

第1章 数制系统及其转换入门

学习目标

1. 简介

本单元的前一部分介绍了本书后面将要学到的内容。在这一部分里，你除了要从总体上把握本书将要学习的内容外，还应该能够解释：

- 模拟系统与数字系统的区别，以及为什么数字系统有更高的精确度。
- 组合电路与时序电路的区别。
- 为什么数字系统中广泛采用两值信号和二进制数。

2. 数制系统及其转换

学习完本章，应该可以解决下列类型的问题：

- 给定一个正整数、正小数或以某数（从2到16）为基的混合数，可以将其转换成以其他数为基的值，并用幂级数展开该数，以证明你的转换过程是正确的。
- 正二进制数的加法、减法、乘法和除法运算，根据进位和借位来解释加减运算的过程。
- 会用原码、反码和补码形式表示负数，会用反码和补码完成带符号二进制数的加法运算，并能证明所使用的方法，能说明什么情况下发生了溢出。
- 可以用BCD码（binary-coded-decimal）、6-3-1-1码及余3码（excess-3 code）等代码表示十进制数。当给定一组权值时可以构造出相应的有权码。

学习指导

1. 学习1.1节 数字系统与开关电路，并回答下列问题：

- 模拟系统与数字系统的基本区别是什么？
- 为什么数字系统比模拟系统有更高的精确度？
- 解释组合开关电路与时序开关电路的区别。
- 数字系统中的开关器件有哪些共同特性？
- 为什么数字系统中采用二进制系统？

2. 学习1.2节 数制系统及其转换，在学习的过程中回答下列习题：

- 当使用除法方法完成数制转换时，所得到的第一位余数是最低有效位还是最高有效位？
- 完成教材中你遇到的所有例题，确保你理解其中的每一个步骤。
- 等式（1-1）说明了一种实现二进制数与十六进制数相互转换的简易方法。为什么是从二进制小数点处而不是从最左边开始把该数分为4位一组？
- 为什么不可能像十六进制一样将十进制数逐个数字转换成二进制数？
- 完成下面的数制转换表。

二进制 (以2为基)	八进制 (以8为基)	十进制 (以10为基)	十六进制 (以16为基)
0	0	0	0
1			
10			
11			
100			
101			
110			
111			
1000			
1001			
1010			
1011			
1100			
1101			
1110			
1111			
10000	20	16	10

(f) 完成习题1.1、1.2、1.3和1.4。

3. 学习1.3节 二进制运算。

(a) 确保你可以理解所有例子，尤其是减法过程中的借位传递。

(b) 为了确保你可以理解借位处理过程，对下列例子写出以二进制运算时的详细分析过程：

$$\begin{array}{r} 1100 \\ -101 \\ \hline 111 \end{array}$$

4. 完成习题1.5、1.6和1.17(a)。

5. 学习1.4节 负数的表示。

(a) 在数字系统中，为什么通常采用反码和补码来表示负数而不采用原码表示？

(b) 说出两种不同的求 n 位二进制数反码的方法。

(c) 说出三种不同的求 n 位二进制数补码的方法。

(d) 若字长 $n = 4$ (包含符号位)，在原码形式中 1000_2 表示哪一个十进制数？

在补码形式中表示什么数？

在反码形式中表示什么数？

(e) 给出某个负数的补码形式，如何求出它的绝对值？

给出某个负数的反码形式，如何求出它的绝对值？

(f) 若字长 $n = 6$ (包含符号位)，在原码形式中 10000_2 表示哪一个十进制数？

在补码形式中表示什么数？

在反码形式中表示什么数？

(g) 溢出的含义是什么？在进行补码加法或反码加法运算中如何判断是否发生了溢出？

最后一位发生进位是否意味着发生了溢出？

- (h) 举例并完成正数与负数在不同组合下的补码与反码加法运算。
- (i) 如何证明反码加法中使用的循环进位方法的正确性?
- (j) 补码数中经常引起麻烦的一种情况是负数的一个特例, 此时该补码数由一个1和其后的所有0组成 ($1000\cdots000$)。若这个数字的长度为 n 位, 它代表什么数? 为什么? (它并不是负数零。)
- (k) 完成习题1.7和1.8。

6. 学习1.5节 二进制编码。

- (a) 用BCD码、余3码、6-3-1-1码和5中取2码 (2-out-of-5 code) 表示187。
- (b) 验证6-3-1-1编码是一种有权码。注意在该编码方案中有些十进制数有两种不同的编码组合, 例如0101和0110都表示4, 在这种情况下, 我们采用具有较小二进制值的编码组合。
- (c) 如何获得余3码?
- (d) 如何获得十进制数字的ASCII编码? ASCII编码中大写字母与小写字母的代码之间有何关系?
- (e) 完成习题1.9。

7. 如果你把本课程做为自择内容式课程, 那么你在继续学习下一章之前, 需要先通过为本章准备的测验。这个测验的目的是测定你是否已经掌握了本章的所有内容, 以及你是否已经做好学习下一章的准备, 在进行这个测验之前你需要:

- (a) 将你的习题答案与提供的答案做对照, 如果你的某些答案存在错误, 确保你明白为什么出现错误并改正这些错误。
- (b) 确保你已经实现本章开始所列出的所有学习目标。

1.1 数字系统与开关电路

数字系统广泛用于计算机、数据处理、控制系统、通讯及测量等领域。由于数字系统比模拟系统有更高的精确度和可靠性, 以前用模拟系统完成的许多任务现在都采用数字系统完成。

在数字系统中, 物理量或信号都被认为是离散值。但在模拟系统中, 物理量或信号可以在某个给定的范围内连续变化。例如数字系统中的输出电压被强制为0V或5V两个值, 但模拟系统中的输出电压允许取 $-10V$ 到 $+10V$ 内的任何值。

因为数字系统处理的是离散信号, 在许多情况下, 这些系统被设计成在给定的输入下, 输出结果恰好是完全正确的。例如, 若我们用十进制乘法器将两个5位的十进制数相乘, 结果中的10位十进制乘积数全部是准确的。但另一方面, 模拟乘法器的输出结果可能存在一个或几个百分点的误差, 这取决于构建模拟乘法器的器件的精度。此外, 如果我们要求乘积结果为20位而不是10位时, 所得到的结果也必须是正确的。这时我们可以重新设计数字乘法器, 使其可以处理更长的数字, 可以在输入端加入更长的数字。但是由于受到器件精度的限制, 对模拟乘法器在精度方面做同样的改进几乎是不可能的。

数字系统的设计可以简单地划分为三个部分——系统设计、逻辑设计和电路设计。系统设

设计包括将整个系统划分成若干子系统，并指定每个子系统的特性。例如数字计算机的系统设计包括各种存储单元、运算单元及输入输出设备的数量及类型的确定，以及这些子系统之间的互连及控制方式的确定。逻辑设计包括如何将基本的逻辑功能块连接起来，以实现特定的功能。逻辑设计的一个例子是确定如何将所需的逻辑门及触发器连接起来，实现一个二进制加法器。电路设计包括如何确定特定器件之间的连接，例如连接寄存器、二极管和晶体管来形成一个门电路、触发器或其他逻辑功能块。当代的电路设计主要采用集成电路的形式，也就是使用合适的计算机辅助工具在一个硅片上设计和互连各个部件。这本书主要讲解逻辑设计及理解逻辑设计过程所必需的基础理论。第18章和第20章涉及系统设计的某些方面，逻辑门的电路设计主要在附录A中讨论。

数字系统中的许多子系统以开关电路(图1-1)的形式出现。数字电路具有一个或多个输入端、一个或多个输出端，输入端和输出端都取离散值。在本书中我们将学习两种类型的开关电路—组合电路和时序电路。在组合电路中，输出值仅与当前的输入值有关，而与以前的输入值无关。在时序电路中，输出值既与当前的输入值有关，也与以前的输入值有关。换句话说，为了决定一个时序电路的输出，必须确定输入的时序。因为时序电路必须记住以前的输入时序，因此有人说它具有记忆功能。但组合电路并没有记忆功能。通常，时序电路由组合电路及附加的存储元件组成。组合电路比时序电路简单，我们将首先学习组合电路。



图1-1 开关电路

构成组合电路的基本功能块是逻辑门。逻辑设计者必须决定如何将这此逻辑门互连起来，以便将电路的输入信号转变成所需的输出信号。这些输入信号与输出信号之间的关系可以使用布尔代数来描述。本书的第2章和第3章介绍了布尔代数中的基本定律与定理，并讲述了如何用布尔代数描述逻辑门电路的功能。

从给定的命题开始，组合逻辑电路设计的第一步是推导出一个真值表或逻辑表达式，该真值表或表达式可以把电路输出表示为电路输入的一个函数(第4章)。为了设计一个可以经济地实现此函数的电路，描述电路输出的逻辑表达式必须经过化简。我们在第3章讲述代数化简方法，其他化简方法(卡诺图和Quine-McCluskey过程)我们在第5章和第6章介绍。第7章我们讲述使用几种门电路实现化简过的逻辑表达式，另一种使用可编程逻辑器件的设计方法我们在第9章讨论。

时序电路设计中的基本存储元件被称为触发器(第11章)，这些触发器与门电路经过互连后可以形成计数器和寄存器(第12章)。第13章介绍时序图、状态转换图及状态转换表等常用的时序电路分析方法。设计时序开关电路的第一步是构造用于描述输入输出序列之间关系的状态转换图或状态转换表(第14章)。第15章介绍如何由状态转换图或状态转换表得到由逻辑门和触发器组成的电路。用可编程逻辑器件实现时序电路的方法我们在第16章讨论。在第18章中，组合电路及时序电路设计技巧应用于实现二进制加法、乘法及除法功能的系统。本书中设计的时序电路被称为同步时序电路，因为这些电路使用一个共同的被称为时钟的时间信

号, 该信号用来同步存储元件的工作。

第10、17和20章中介绍如何在组合逻辑、时序逻辑和数字系统设计中使用硬件描述语言VHDL。VHDL用于对数字硬件进行描述、仿真和综合。在编写好VHDL代码之后, 设计者可以使用计算机辅助设计软件对这些硬件描述进行编译, 从而完成数字逻辑的设计。这使得我们无需手工完成由门电路和触发器设计的所有电路的细节, 就可以实现某些复杂的设计过程。

数字系统中所使用的开关器件通常是具有两种状态的器件, 也就是说, 可以认为它们的输出值只能在两个不同的离散值之中取值。开关器件有继电器、二极管和晶体管等。继电器可以被认为只有两种状态——闭合或开启, 这取决于线圈中是否有电流流过。二极管可以处于导通状态或非导通状态, 晶体管可以处于截止状态或饱和状态, 相应的输出为高电平或低电平。当然晶体管也可以作为线性放大器使用, 相应的输出电压在一定范围内连续变化。但在数字应用中, 若把晶体管用做两态器件操作的话, 可以获得更高的可靠性。由于大多数开关器件的输出被认为只能取两个不同的值, 因此在数字系统内部使用二进制也就是很自然的事了。基于这个原因, 在继续学习开关电路设计之前, 我们先讨论二进制数及数制系统。

1.2 数制系统及其转换

当我们写一个十进制数(基数为10)时, 我们在采用一种位置相关的概念: 每个数字要被某个适当的10的幂相乘, 这个幂与该数字的位置相关。例如:

$$953.78_{10} = 9 \times 10^2 + 5 \times 10^1 + 3 \times 10^0 + 7 \times 10^{-1} + 8 \times 10^{-2}$$

同理, 对于一个二进制数(基数为2), 每个二进制数字要被某个适当的2的幂相乘:

$$\begin{aligned} 1011.11_2 &= 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} \\ &= 8 + 0 + 2 + 1 + \frac{1}{2} + \frac{1}{4} = 11\frac{3}{4} = 11.75_{10} \end{aligned}$$

注意二进制小数点将2的正数幂与负数幂分开, 就像十进制小数点将十进制数中的正数幂与负数幂分开一样。

任何一个正整数 $R(R > 1)$ 可以被用做为某数制系统的根或基数。若基数为 R , 则共使用 R 个数字(0, 1, ..., $R-1$)。例如若 $R=8$, 则使用的数字是0、1、2、3、4、5、6和7。一个用位置相关法写出的数可以展开成 R 的幂级数形式。例如:

$$\begin{aligned} N &= (a_4 a_3 a_2 a_1 a_0 a_{-1} a_{-2} a_{-3})_R \\ &= a_4 \times R^4 + a_3 \times R^3 + a_2 \times R^2 + a_1 \times R^1 + a_0 \times R^0 \\ &\quad + a_{-1} \times R^{-1} + a_{-2} \times R^{-2} + a_{-3} \times R^{-3} \end{aligned}$$

这里的 a_i 是 R 的系数并且 $0 < a_i < R-1$ 。如果幂级数展开式中以10为基数运算, 那么所得结果就是 N 的十进制等效值, 例如:

$$\begin{aligned} 147.3_8 &= 1 \times 8^2 + 4 \times 8^1 + 7 \times 8^0 + 3 \times 8^{-1} = 64 + 32 + 7 + \frac{3}{8} \\ &= 103.375_{10} \end{aligned}$$

当基数大于10时, 为了表示一个数字需要10个以上的符号, 在这种情况下一般用字符代表大于9的数字。例如在十六进制(以16为基数)中, A 代表 10_{10} , B 代表 11_{10} , C 代表 12_{10} , D 代表 13_{10} , E 代表 14_{10} , F 代表 15_{10} 。因此:

$$A2F_{16} = 10 \times 16^2 + 2 \times 16^1 + 15 \times 16^0 = 2560 + 32 + 15 = 2607_{10}$$

接下来我们讨论用除法将十进制整数转换成以 R 为基数的数值的方法。十进制整数 N 的以 R 为基数的等效值可以表示为:

$$N = (a_n a_{n-1} \cdots a_2 a_1 a_0)_R = a_n R^n + a_{n-1} R^{n-1} + \cdots + a_2 R^2 + a_1 R^1 + a_0$$

如果我们用 N 除以 R , 设余数为 a_0 :

$$\frac{N}{R} = a_n R^{n-1} + a_{n-1} R^{n-2} + \cdots + a_2 R^1 + a_1 = Q_1, \text{ 余数 } a_0$$

接着我们用商 Q_1 除以 R 得:

$$\frac{Q_1}{R} = a_n R^{n-2} + a_{n-1} R^{n-3} + \cdots + a_3 R^1 + a_2 = Q_2, \text{ 余数 } a_1$$

接着我们用商 Q_2 除以 R 得:

$$\frac{Q_2}{R} = a_n R^{n-3} + a_{n-1} R^{n-4} + \cdots + a_3 = Q_3, \text{ 余数 } a_2$$

这个过程继续下去, 直到我们最后得到 a_n 。注意每个除法步骤中所得的余数就是我们想要的某位数字, 并且最先得到的是最低有效位的数字。

例题1 将 53_{10} 转换成二进制

$$\begin{array}{r} 2 \overline{)53} \\ 2 \overline{)26} \text{ 余数 } = 1 = a_0 \\ 2 \overline{)13} \text{ 余数 } = 0 = a_1 \\ 2 \overline{)6} \text{ 余数 } = 1 = a_2 \\ 2 \overline{)3} \text{ 余数 } = 0 = a_3 \\ 2 \overline{)1} \text{ 余数 } = 1 = a_4 \\ 0 \text{ 余数 } = 1 = a_5 \end{array} \quad 53_{10} = 110101_2$$

我们可以用连续乘以 R 的方法将一个十进制小数转换成以 R 为基的等效值。十进制小数可以表示为:

$$F = (a_{-1} a_{-2} a_{-3} \cdots a_{-m})_R = a_{-1} R^{-1} + a_{-2} R^{-2} + a_{-3} R^{-3} + \cdots + a_{-m} R^{-m}$$

乘以 R 得出:

$$FR = a_{-1} + a_{-2} R^{-1} + a_{-3} R^{-2} + \cdots + a_{-m} R^{-m+1} = a_{-1} + F_1$$

这里的 F_1 代表结果的小数部分, a_{-1} 代表结果的整数部分, F_1 乘以 R 得出:

$$F_1 R = a_{-2} + a_{-3} R^{-1} + \cdots + a_{-m} R^{-m+2} = a_{-2} + F_2$$

接着给 F_2 乘以 R 得到:

$$F_2 R = a_{-3} + \cdots + a_{-m} R^{-m+3} = a_{-3} + F_3$$

这个过程一直进行下去，直到我们得到足够位数的数字。请注意每一步所得整数部分就是我们想要的某位数字，并且我们最先得到的是最高有效位。

例题2 将 0.625_{10} 转换为二进制

$$\begin{array}{r}
 F = 0.625 \\
 \times 2 \\
 \hline
 1.250 \\
 (a_{-1} = 1)
 \end{array}
 \qquad
 \begin{array}{r}
 F_1 = 0.250 \\
 \times 2 \\
 \hline
 0.500 \\
 (a_{-2} = 0)
 \end{array}
 \qquad
 \begin{array}{r}
 F_2 = 0.500 \\
 \times 2 \\
 \hline
 1.000 \\
 (a_{-3} = 1)
 \end{array}
 \qquad
 0.625_{10} = 0.101_2$$

这个转换过程通常不能终止，当过程不能终止时，则表示结果是一个循环小数。

例题3 将 0.7_{10} 转换为二进制

$$\begin{array}{r}
 0.7 \\
 \underline{2} \\
 (1)0.4 \\
 \underline{2} \\
 (0)0.8 \\
 \underline{2} \\
 (1)0.6 \\
 \underline{2} \\
 (1)0.2 \\
 \underline{2} \\
 (0)0.4 \\
 \underline{2} \\
 (0)0.8
 \end{array}
 \quad \leftarrow \begin{array}{l}
 \text{因为前面得出过} 0.4, \text{ 所以} \\
 \text{运算过程从这里开始循环}
 \end{array}$$

$0.7_{10} = 0.1 \underline{0110} \underline{0110} \underline{0110} \dots_2$

在两个以非10为基数的数值之间转换时，可以直接使用刚才给出的转换过程。但是转换时的进位必须用新的基数来运算而不是使用10。通常比较简单的办法是先将其转换成十进制数，然后将这个十进制数转换成另外一个基数下的数。

例题4 转换 231.3_4 为以7为基的数

$$231.3_4 = 2 \times 16 + 3 \times 4 + 1 + \frac{3}{4} = 45.75_{10}$$

$$\begin{array}{r}
 7 \overline{)45} \\
 \underline{7 \overline{)6}} \quad \text{余数} 3 \\
 \quad \quad \underline{0} \quad \text{余数} 6
 \end{array}
 \qquad
 \begin{array}{r}
 0.75 \\
 \underline{7} \\
 (5)0.25 \\
 \underline{7} \\
 (1)0.75 \\
 \underline{7} \\
 (5)0.25 \\
 \underline{7} \\
 (1)0.75
 \end{array}
 \qquad
 45.75_{10} = 63.5151 \dots_7$$

将二进制数转换成十六进制数（及相反转换）时，可以通过观察这个数直接转换。因为每位十六进制数恰好对应四位二进制数。从二进制数的小数点开始，每四位二进制数划分为