

数字原理与设计



Digital Principles and Design

Donald D. Givone 原著
罗 嵘 汪 玉 刘勇攀 等 编译

清华大学出版社



双语版教材

数字原理与设计

Digital Principles and Design

Donald D. Givone 原著

罗嵘 汪玉 刘勇攀 等 编译

清华大学出版社

北京

Donald D. Givone

Digital Principles and Design

EISBN: 0-07-119520-3(ISE)

Copyright © 2003 by The McGraw-Hill Companies, Inc.

Original language published by The McGraw-Hill Companies, Inc. All Rights reserved. 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.

Simplified Chinese translation edition is published and distributed exclusively by Tsinghua University Press under the authorization by McGraw-Hill Education(Asia)Co. , within the territory of the People's Republic of China only(excluding Hong Kong, Macao SAR and Taiwan). Unauthorized export of this edition is a violation of the Copyright Act. Violation of this Law is subject to Civil and Criminal Penalties.

本书中文简体字翻译版由美国麦格劳-希尔教育出版(亚洲)公司授权清华大学出版社在中华人民共和国境内(不包括中国香港、澳门特别行政区和中国台湾地区)独家出版发行。未经许可之出口视为违反著作权法,将受法律之制裁。未经出版者预先书面许可,不得以任何方式复制或抄袭本书的任何部分。

北京市版权局著作权合同登记号 图字: 01-2006-3566

版权所有,翻印必究。举报电话: 010-62782989 13501256678 13801310933

本书封面贴有 McGraw-Hill 公司防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

数字原理与设计/(美)吉沃恩(Givone, D. D.)原著;罗嵘,汪玉,刘勇攀等编译. —北京:清华大学出版社, 2006. 10

书名原文: Digital Principles and Design

ISBN 7-302-13404-9

I. 数… II. ①吉…②罗…③汪…④刘… III. ①数字系统—理论—双语教学—高等学校—教材—汉、英 ②数字系统—系统设计—双语教学—高等学校—教材—汉、英 IV. TP271

中国版本图书馆 CIP 数据核字(2006)第 078147 号

出 版 者: 清华大学出版社

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

社 总 机: 010-62770175

地 址: 北京清华大学学研大厦

邮 编: 100084

客户服务: 010-62776969

组稿编辑: 邹开颜 王敏稚

文稿编辑: 李玮琪

印 刷 者: 清华大学印刷厂

装 订 者: 三河市新茂装订有限公司

发 行 者: 新华书店总店北京发行所

开 本: 185×230 印张: 49.75 字数: 1677 千字

版 次: 2006 年 10 月第 1 版 2006 年 10 月第 1 次印刷

书 号: ISBN 7-302-13404-9/TP·8417

印 数: 1~3000

定 价: 69.00 元

译者的话

21世纪是国际化的信息时代,为培养学生具备优秀的查阅外文资料、获取最新信息的本领,提高与国外同行进行口头和书面交流能力,双语教学成为我国高等院校课程改革的一个特色。

这本中英文对照的双语版教材是以原版的英文教材(Donald D. Givone的著作 Digital Principles and Design,2003年由 McGraw-Hill 公司出版)为主,附上其中重点章节的汉语译文。该书详细、系统地论述了数字系统实现涉及的基本原理和分析设计方法,其内容与我国高等院校开设的数字电路类专业课程的教学大纲基本一致。原书概念准确、论述严谨、思路清晰,是一本易教易学的优秀教材,已被美国众多大学作为教学参考书。

全书分为9章:第1章绪论;第2章数制系统、算术和编码;第3章布尔代数和组合网络;第4章布尔表达式的化简;第5章使用 MSI 元件和可编程逻辑器件的逻辑设计;第6章触发器及其简单应用;第7章同步时序网络;第8章算法状态机;第9章异步时序网络。在进行英文影印时,删减了如下章节:第2章的2.9节和2.12节;第4章的4.7节~4.14节;第9章的9.4节~9.8节;附录A;索引。

限于篇幅,且根据我国普通高等院校相关课程的教学基本要求,在译文中,除了影印删减的内容没有翻译以外,还有如下内容和章节没有翻译:第2章的2.5.2小节和2.5.4小节;第8章;第9章;每章后的习题;参考文献等。为了方便读者,本书增加了中英文专用名词索引。本书的翻译工作是由清华大学电子工程系罗嵘副教授负责完成。罗嵘副教授一直从事清华大学本科生专业基础课“数字逻辑电路”的教学工作。本书不仅可以作为数字电路类课程双语教学的教材,还可以作为进行数字系统设计的工程师的基础理论参考书。

罗嵘副教授负责第1章~第3章、第7章的翻译以及全书的审校,汪玉负责第4章和第5章的翻译,刘勇攀负责第6章的翻译,王桂颖负责全书的文字校审。参加本书翻译工作的还有刘伟、罗洪、丁潜、应蓓华等。他们牺牲了大量业余时间,在此表示衷心感谢。此外,刘宝琴教授、杨华中教授、汪蕙教授、罗淑云教授以及王希勤教授对本书提出了很多宝贵意见,在此一并表示衷心感谢。

在本书出版的过程中,还得到了原作者、McGraw-Hill 出版公司和清华大学出版社的大力支持,在此表示我们诚挚的感谢。由于译者水平有限,译文中难免有误,敬请读者批评指正。

译者

2006年于清华大学

《数字原理与设计》

教师反馈卡

感谢您购买本书！本书系我社获美国 McGraw-Hill 公司授权出版。McGraw-Hill 公司是美国著名的教育图书出版公司，出版了很多著名的计算机、工程类以及经管类图书。我社与 McGraw-Hill 公司为了使中国的学生和教师更快地了解国外高校教材，特别联合出版系列教材，以求为中国高等教育水平的提高尽一份力量。

同时，我们十分重视教师手册等教学课件以及网上资源的使用，如果您确认将本书作为指定教材，请填好下面的表格并经系主任签字盖章后寄回我们的联系地址，McGraw-Hill 公司将免费向您提供英文原版的教师手册或其他教学课件。

联系地址：



清华大学出版社
清华大学学研大厦 A604 室

邮编：100084
Tel: 010-62770175-3208
Fax: 010-62770278
E-mail: hanbh@tup.tsinghua.edu.cn

姓名			
院校			
系			
专业			
您所教课程的名称			
学生人数/学期		学时	
您目前采用的教材	作者		
	书名		
	出版社		
联系地址			
邮政编码			
联系电话			
E-mail			
您的建议(或意见)			系主任签字 盖章

PREFACE

With the strong impact of digital technology on our everyday lives, it is not surprising that a course in digital concepts and design is a standard requirement for majors in computer engineering, computer science, and electrical engineering. An introductory course is frequently encountered in the first or second year of their undergraduate programs. Additional courses are then provided to refine and extend the basic concepts of the introductory course.

This book is suitable for an introductory course in digital principles with emphasis on logic design as well as for a more advanced course. With the exception of the appendix, it assumes no background on the part of the reader. The intent of the author is not to just present a set of procedures commonly encountered in digital design but, rather, to provide justifications underlying such procedures. Since no background is assumed, the book can be used by students in computer engineering, computer science, and electrical engineering.

The approach taken in this book is a traditional one. That is, emphasis is on the presentation of basic principles of logic design and the illustration of each of these principles. The philosophy of the author is that a first course in logic design should establish a strong foundation of basic principles as provided by a more traditional approach before engaging in the use of computer-aided design tools. Once basic concepts are mastered, the utilization of design software becomes more meaningful and allows the student to use the software more effectively. Thus, it is the understanding of basic principles on which this book focuses and the application of these principles to the analysis and design of combinational and sequential logic networks. Each topic is approached by first introducing the basic theory and then illustrating how it applies to design.

SCOPE OF THE BOOK

Chapter 1 discusses the differences between continuous, i.e., analog, and discrete, i.e., digital, networks and devices. Then, the basic operation of the digital computer is introduced as an example of a system that utilizes most of the concepts presented in the remaining chapters. The chapter concludes with an overview of the topics that will be introduced.

In Chapter 2 the general concepts of positional number systems, arithmetic, and conversion techniques are introduced. This material is developed for arbitrary positive integer bases rather than simply for the binary number system to emphasize the similarity of all positional number systems and their manipulations. Then, various codes and their properties are discussed with emphasis on error detection and correction.

The two-valued Boolean algebra is introduced in Chapter 3. How Boolean expressions are written, manipulated, and simplified is presented. Since there is a one-

to-one correspondence between the two-valued Boolean algebra and logic networks, it is then shown how the algebra can serve as a mathematical model for the behavior and structure of combinational logic networks. The chapter concludes with a discussion of the gate properties that are relevant to logic networks, i.e., noise margin, fan-out, propagation delays, and power dissipation.

An important application of the Boolean algebra is to obtain those expressions which can best be associated with optimal networks. Under the assumption that the reduction in the delay time of a network is of paramount importance, it is possible to obtain efficient networks by systematic procedures. Two methods for obtaining minimal expressions are presented in Chapter 4. The first method, Karnaugh maps, is a graphical procedure that permits minimal expressions to be obtained very rapidly. However, since the procedure relies upon the recognition of patterns, there is a limit to the complexity of a problem for which the procedure is effective. This limit seems to be problems of six variables. A second method for obtaining minimal expressions is the Quine-McCluskey method. This method involves just simple mathematical manipulations. For problems with many variables, the Quine-McCluskey method can be carried out on a digital computer. The concepts of both approaches are then extended to a set of Boolean expressions describing multiple-output networks. The chapter concludes with a variation of the Karnaugh map concept, called variable-entered Karnaugh maps, in which Boolean functions can appear as map entries.

Chapter 5 is concerned with several MSI and LSI components. The intent of this chapter is to investigate combinational networks that are commonly encountered in digital systems. Several types of adders and subtracters are discussed. These include binary and decimal adders as well as high-speed adders using the carry lookahead concept. Also included in this chapter are discussions on comparators, decoders, encoders, and multiplexers. In the case of decoders and multiplexers, attention is given to their use as generic logic design devices. The final part of the chapter involves the three basic structures of programmable logic devices: programmable read-only memories, programmable logic arrays, and programmable array logic devices. Emphasis is placed on their utilization for the realization of logic networks, with special attention given to their strengths and weaknesses and the constraints placed on a logic design utilizing them.

Chapter 6 begins the presentation on sequential logic networks. In this chapter, various types of flip-flops, i.e., *JK*, *D*, *T*, and *SR* flip-flops, are introduced. The operational behavior of the three categories of flip-flops, i.e., latches, edge-triggered, and master-slave flip-flops, is discussed in detail. The remainder of the chapter is concerned with some simple flip-flop applications, in particular, registers and counters. Ripple and synchronous counters are presented and compared. The chapter concludes with a general design procedure for synchronous counters. This procedure serves as a basis for synchronous sequential logic design, which is elaborated upon in the next two chapters.

Chapters 7 and 8 involve clocked synchronous sequential networks. In Chapter 7 the classic Mealy and Moore models of a synchronous sequential network are presented. First, these networks are analyzed to establish various tabular representations of network behavior. Then, the process is reversed and synthesis is discussed.

Chapter 8 also involves the design of clocked synchronous sequential networks; however, this time using the algorithmic state machine model. The relationship between the classic Mealy/Moore models and the algorithmic state machine model is discussed as well as the capability of the algorithmic state machine model to handle the controlling of an architecture of devices.

In Chapter 9 asynchronous sequential networks are studied. Paralleling the approach taken for synchronous sequential networks, the analysis of asynchronous sequential networks is first undertaken and then, by reversing the analysis procedure, the synthesis of these networks is presented. Included in this chapter is also a discussion on static and dynamic hazards. Although these hazards occur in combinational networks, their study is deferred to this chapter, since these hazards can have a major effect on asynchronous network behavior. A great deal of attention is given to the many design constraints that must be satisfied to achieve a functional design of an asynchronous network. In addition to the static and dynamic hazards, the concepts of races, the importance of the state assignment, and the effects of essential hazards are addressed.

An appendix on digital electronics is included for completeness. It is not intended to provide an in-depth study on digital electronics, since such a study should be reserved for a course in itself. Rather, its inclusion is to provide the interested reader an introduction to actual circuits that can occur in digital systems and the source of constraints placed upon a logic designer. For this reason, the appendix does not delve into circuit design but, rather, only into the analysis of electronic digital circuits. Emphasis is placed on the principles of operation of TTL, ECL, and MOS logic circuits. Since circuits are analyzed, the appendix does assume the reader has an elementary knowledge of linear circuit analysis. In particular, the reader should be familiar with Ohm's law along with Kirchhoff's current and voltage laws.

Another appendix with software tutorials is also included. These tutorials, provided by two contributors, include one on Altera MAX+plus II 10.1 Student Edition and one on LogicWorksTM4. The tutorials are meant to provide basic introductions to these tools for those people who are using them in their course.

HOMEWORK PROBLEMS

With the exception of Chapter 1, each chapter includes a set of problems. Some of these problems provide for reinforcement of the reader's understanding of the material, some extend the concepts presented in the chapter, and, finally, some are applications-oriented.

ADDITIONAL RESOURCES

The expanded book website at <http://www.mhhe.com/givone> includes a downloadable version of the *Solutions Manual* for instructors only and PowerPoint slides. There are also a variety of labs using both the Altera Software and LogicWorks. A CD-ROM containing Altera's MAX+plus II CAD software and Multisim 2001 is included free with every copy of the book.

WHAT CAN BE COVERED IN A COURSE

More material is included in this book than can be covered in a one-semester course. This allows the instructor to tailor the book to the background of the students and the time available. Different ways the book can be used include:

- A possible one-semester course based on this book would include Chapters 1 to 3, Sections 4.1 to 4.7, and Chapters 5 to 7. Sections 4.1 to 4.7 involve the simplification of Boolean functions using Karnaugh maps.
- Sections 4.8 to 4.11, involving the Quine-McCluskey method, might also be included in a slightly more quickly paced course.
- The material of Sections 4.12 to 4.14 and Chapters 8 and 9 can serve as the basis for a second semester course.

A few formal proofs have been included for the interested reader. However, these proofs are clearly delineated and can be skipped without loss of continuity.

ACKNOWLEDGMENTS

I would like to thank the reviewers of the manuscript for their comments and suggestions. These include:

- Kenneth J. Breeding, The Ohio State University
- Kirk W. Cameron, University of South Carolina
- Mehmet Celenk, Ohio University
- Travis E. Doom, Wright State University
- Richard W. Freeman, Iowa State University
- Bruce A. Harvey, Florida State University
- Raj Katti, North Dakota State University
- Larry Kinney, University of Minnesota
- Wagdy H. Mahmoud, Tennessee Technological University
- Jeffery P. Mills, Illinois Institute of Technology
- Debashis Mohanty, Texas A&M University
- Richard G. Molyet, The University of Toledo
- Jane Moorhead, Mississippi State University
- Suku Nair, Southern Methodist University
- Emil C. Neu, Stevens Institute of Technology
- Tatyana D. Roziner, Boston University
- Salam Salloum, California State Polytechnic University, Pomona
- Susan Schneider, Marquette University
- Charles B. Silio, Jr., University of Maryland
- Dan Stanzione, Clemson University
- A. J. Thomas, Jr., Tennessee State University

- Massood Towhidnejad, Embry-Riddle Aeronautical University
- Murali Varanasi, University of South Florida
- Donald C. Wunsch II, University of Missouri-Rolla

Also, I would like to acknowledge the efforts of the staff at McGraw-Hill, particularly Betsy Jones, Michelle Flomenhoft, and Susan Brusch. In addition, I want to express my appreciation to Donna and David, who helped with the typing and artwork in the early drafts of the manuscript. I would also like to thank David for his work that led to the image on the cover of this book. Finally, I am grateful to my wife Louise for her support and patience during this project.

Donald D. Givone

由于数字技术对日常生活的强大影响,有关数字概念和设计的课程已经成为计算机工程、计算机科学和电机工程专业的基础课。在大学本科的第一年或者第二年经常会遇到相关的绪论课程。在此基础上,会设置附加课程来细化和扩展绪论课程的基本概念。

本书适合作为介绍数字原理的绪论课程(重点强调逻辑设计)以及更高级课程的教材。除附录以外,本书假定读者没有任何相关的背景知识。本书作者的意图是不仅仅给出数字设计中普遍遇到的一系列过程,还更希望提供给读者过程背后的原因。本书假定读者没有任何背景知识,因此可供学习计算机工程、计算机科学和电机工程的学生使用。

本书采用常用的介绍方法。也就是说,重点介绍逻辑设计的基本原理以及每个原理的例证。作者的观点是在使用计算机辅助设计工具之前,逻辑设计的第一门课应该建立有关基本原理的扎实基础。一旦掌握了基本概念,运用设计软件就变得更有意义,且学生可以更有效地使用这些软件。因此,本书强调基本原理的理解以及用这些原理来分析和设计组合及时序逻辑网络。每个主题都是先介绍基本理论,然后阐明如何利用它来进行设计。

本书内容

第1章讨论连续(即模拟)和离散(即数字)网络及设备之间的差别。然后,作为一个综合了本书其他章给出的大多数概念的系统实例,介绍了数字计算机的基本操作。最后,概述了即将讨论的主题。

第2章介绍位置数制系统、算术和转换技术的一般概念。该章详述任意正整数基数,而不只是介绍二进制数制系统,以强调所有位置数制系统以及对其处理的相似性。然后,讨论各种编码和它们的特点,重点介绍错误检测和修正。

第3章引入二值布尔代数。该章给出了如何写、处理和化简布尔表达式。因为二值布尔代数和逻辑网络之间存在一一对应的关系,随后可以看到代数是如何在作为描述组合逻辑网络的行为和结构的数学模型中应用的。该章还对与逻辑网络相关的门特性(即噪声容限、扇出、传输延时和功耗)进行了讨论。

布尔代数的一个重要应用是那些获得最佳表示优化网络的表达式。假定网络的延长时间减少是最重要的,就可能通过系统化过程获得高效的网络。第4章给出了两种获得最简表达式的方法。第一种方法(卡诺图)是一种图形化的过程,可以快速地获得最简表达式。但是,由于卡诺图化简过程依赖于图形的识别,所以为了保证化简过程的有效性,需要对问题的复杂度进行限制。这个限制看起来就是六变量问题。第二种获得最简表达式的方法是

Quine-McCluskey 法。这种方法只涉及简单的数学处理。对于包含许多变量的问题, Quine-McCluskey 法可在数字计算机上执行。然后,将这两种方法的概念扩展到描述多输出网络的一组布尔表达式。该章最后给出了卡诺图概念的变化形式,称为输入变量的卡诺图,此图单元格中可以出现布尔函数。

第 5 章涉及若干 MSI 和 LSI 器件。该章的目的是研究数字系统中常常遇到的组合网络。讨论了若干类型的加法器和减法器。这些器件包括二进制和十进制加法器,以及使用超前进位概念的高速加法器。这一章的内容还包括对比较器、译码器、编码器和多路选择器的讨论。对于译码器和多路选择器,只关注它们作为通用逻辑设计器件的使用。该章最后一部分涉及可编程逻辑器件的 3 种基本结构:可编程只读存储器、可编程逻辑阵列和可编程阵列逻辑器件。重点放在使用它们来实现逻辑网络,特别注意它们的优势和弱点以及用它们完成逻辑设计时的约束。

第 6 章开始介绍时序逻辑网络。该章介绍了各种类型的触发器,即 JK、D、T 和 SR 触发器。该章还详细讨论了 3 类触发器的工作特性,即锁存器、边沿触发器和主从型触发器。剩余章节则关注一些简单的触发器应用,尤其是寄存器和计数器。提出了波状和同步计数器,并进行了比较。该章以同步计数器的通用设计过程结束。这个过程可作为同步时序逻辑设计的基础,在接下来的两章中将详尽进行介绍。

第 7 章涉及钟控同步时序网络。该章提出了经典的同步时序网络的米利和摩尔模型。首先,分析这些网络以建立网络行为的各种表格化表示。然后,将此过程反过来讨论综合过程。

课程覆盖内容

本书的内容比一个学期的课程内容要多。这样,教师可以根据学生的背景和讲课时间限制来裁剪本书的内容。使用本书的不同方式包括:

- 基于本书的可能的一学期课程可以包括第 1 章~第 3 章,第 4.1 节~第 4.7 节,以及第 5 章~第 7 章。第 4.1 节~第 4.7 节涵盖了用卡诺图进行布尔函数的简化。
- 涵盖 Q-M 表格法的第 4.8 节~第 4.11 节也可以包括在进度稍快的课程中。
- 第 4.12 节~第 4.14 节以及第 8 章和第 9 章的内容可作为下一学期课程的基础。

对于有兴趣的读者,本书给出了几个定理的正式证明。此外,本书完整地阐述了这些证明,读者可以跳过它们但不影响阅读的连续性。

致谢

我要感谢对本书书稿给出宝贵意见和建议的审稿专家们。他们是:
Kenneth J. Breeding, 俄亥俄州立大学

Kirk W. Cameron, 南卡罗来纳大学
Mehmet Celenk, 俄亥俄大学
Travis E. Doom, 莱特州立大学
Richard W. Freeman, 爱荷华州立大学
Bruce A. Harvey, 佛罗里达州立大学
Raj Katti, 北达科他州立大学
Larry Kinney, 明尼苏达大学
Wagdy H. Mahmoud, 田纳西科技大学
Jeffery P. Mills, 伊利诺伊理工学院
Debashis Mohanty, 德州 A&M 大学
Richard G. Molyet, 托莱多大学
Jane Moorhead, 密西西比州立大学
Suku Nair, 南方卫理公会大学
Emil C. Neu, 史蒂文斯理工学院
Tatyana D. Roziner, 波士顿大学
Salam Salloum, 加利福尼亚州立理工大学波莫纳分校
Susan Schneider, 马奎特大学
Charles B. Silio, 马里兰大学
Dan Stanzione, 克莱姆森大学
A. J. Thomas, 田纳西州立大学
Massood Towhidnejad, 艾姆伯里-利德尔航空大学
Murali Varanasi, 南佛罗里达大学
Donald C. Wunsch II, 密苏里-罗拉大学

同时,我要向 McGraw-Hill 出版社的全体成员表示衷心感谢,尤其是 Besty Jones、Michelle Flomenhoft 和 Susan Brusch。此外,我希望向 Donna 和 David 表示我的感激之情,他们帮助完成了书稿的早期草稿的录入和加工。我还要感谢 David 完成了本书封面图片的设计。最后,我非常感激我的妻子 Louise,感谢她在我写作期间给予的支持和爱。

Donald D. Givone

CONTENTS

Chapter 1

Introduction 1

- 1.1 The Digital Age 1
- 1.2 Analog and Digital Representations of Information 2
- 1.3 The Digital Computer 2
 - 1.3.1 The Organization of a Digital Computer 3
 - 1.3.2 The Operation of a Digital Computer 5
- 1.4 An Overview 5

Chapter 2

Number Systems, Arithmetic, and Codes 7

- 2.1 Positional Number Systems 7
- 2.2 Counting in a Positional Number System 9
- 2.3 Basic Arithmetic Operations 11
 - 2.3.1 Addition 11
 - 2.3.2 Subtraction 11
 - 2.3.3 Multiplication 14
 - 2.3.4 Division 16
- 2.4 Polynomial Method of Number Conversion 16
- 2.5 Iterative Method of Number Conversion 19
 - 2.5.1 Iterative Method for Converting Integers 20
 - 2.5.2 Verification of the Iterative Method for Integers 21
 - 2.5.3 Iterative Method for Converting Fractions 22

2.5.4 Verification of the Iterative Method for Fractions 23

2.5.5 A Final Example 23

- 2.6 Special Conversion Procedures 24
- 2.7 Signed Numbers and Complements 26
- 2.8 Addition and Subtraction with r 's-Complements 31
 - 2.8.1 Signed Addition and Subtraction 33
- 2.9* Addition and Subtraction with $(r - 1)$'s-Complements
 - 2.9.1 Signed Addition and Subtraction
- 2.10 Codes 36
 - 2.10.1 Decimal Codes 37
 - 2.10.2 Unit-Distance Codes 40
 - 2.10.3 Alphanumeric Codes 42
- 2.11 Error Detection 43
- 2.12* Error Correction
 - 2.12.1 Hamming Code
 - 2.12.2 Single-Error Correction plus Double-Error Detection
 - 2.12.3 Check Sum Digits for Error Correction
- Problems 45

Chapter 3

Boolean Algebra and Combinational Networks 53

- 3.1 Definition of a Boolean Algebra 54
 - 3.1.1 Principle of Duality 55
- 3.2 Boolean Algebra Theorems 55
- 3.3 A Two-Valued Boolean Algebra 62
- 3.4 Boolean Formulas and Functions 65
 - 3.4.1 Normal Formulas 67

注：带“*”章节影印从略。

- 3.5 Canonical Formulas 68
 - 3.5.1 Minterm Canonical Formulas 68
 - 3.5.2 m -Notation 70
 - 3.5.3 Maxterm Canonical Formulas 72
 - 3.5.4 M -Notation 73
- 3.6 Manipulations of Boolean Formulas 75
 - 3.6.1 Equation Complementation 75
 - 3.6.2 Expansion about a Variable 76
 - 3.6.3 Equation Simplification 76
 - 3.6.4 The Reduction Theorems 78
 - 3.6.5 Minterm Canonical Formulas 79
 - 3.6.6 Maxterm Canonical Formulas 80
 - 3.6.7 Complements of Canonical Formulas 81
- 3.7 Gates and Combinational Networks 83
 - 3.7.1 Gates 84
 - 3.7.2 Combinational Networks 84
 - 3.7.3 Analysis Procedure 85
 - 3.7.4 Synthesis Procedure 86
 - 3.7.5 A Logic Design Example 87
- 3.8 Incomplete Boolean Functions and Don't-Care Conditions 89
 - 3.8.1 Describing Incomplete Boolean Functions 91
 - 3.8.2 Don't-Care Conditions in Logic Design 91
- 3.9 Additional Boolean Operations and Gates 93
 - 3.9.1 The Nand-Function 94
 - 3.9.2 The Nor-Function 95
 - 3.9.3 Universal Gates 95
 - 3.9.4 Nand-Gate Realizations 97
 - 3.9.5 Nor-Gate Realizations 100
 - 3.9.6 The Exclusive-Or-Function 103
 - 3.9.7 The Exclusive-Nor-Function 105
- 3.10 Gate Properties 105
 - 3.10.1 Noise Margins 107
 - 3.10.2 Fan-Out 108
 - 3.10.3 Propagation Delays 109
 - 3.10.4 Power Dissipation 110
- Problems 110

Chapter 4

Simplification of Boolean Expressions 119

- 4.1 Formulation of the Simplification Problem 119
 - 4.1.1 Criteria of Minimality 120
 - 4.1.2 The Simplification Problem 121
- 4.2 Prime Implicants and Irredundant Disjunctive Expressions 121
 - 4.2.1 Implies 121
 - 4.2.2 Subsumes 122
 - 4.2.3 Implicants and Prime Implicants 123
 - 4.2.4 Irredundant Disjunctive Normal Formulas 125
- 4.3 Prime Implicates and Irredundant Conjunctive Expressions 125
- 4.4 Karnaugh Maps 127
 - 4.4.1 One-Variable and Two-Variable Maps 127
 - 4.4.2 Three-Variable and Four-Variable Maps 128
 - 4.4.3 Karnaugh Maps and Canonical Formulas 130
 - 4.4.4 Product and Sum Term Representations on Karnaugh Maps 133
- 4.5 Using Karnaugh Maps to Obtain Minimal Expressions for Complete Boolean Functions 137
 - 4.5.1 Prime Implicants and Karnaugh Maps 137
 - 4.5.2 Essential Prime Implicants 142
 - 4.5.3 Minimal Sums 143
 - 4.5.4 Minimal Products 147
- 4.6 Minimal Expressions of Incomplete Boolean Functions 149
 - 4.6.1 Minimal Sums 150
 - 4.6.2 Minimal Products 151
- 4.7* Five-Variable and Six-Variable Karnaugh Maps
 - 4.7.1 Five-Variable Maps
 - 4.7.2 Six-Variable Maps

- 4.8* The Quine-McCluskey Method of Generating Prime Implicants and Prime Implicates
 - 4.8.1 Prime Implicants and the Quine-McCluskey Method
 - 4.8.2 Algorithm for Generating Prime Implicants
 - 4.8.3 Prime Implicates and the Quine-McCluskey Method
 - 4.9* Prime-Implicant/Prime-Implicate Tables and Irredundant Expressions
 - 4.9.1 Petrick's Method of Determining Irredundant Expressions
 - 4.9.2 Prime-Implicate Tables and Irredundant Conjunctive Normal Formulas
 - 4.10* Prime-Implicant/Prime-Implicate Table Reductions
 - 4.10.1 Essential Prime Implicants
 - 4.10.2 Column and Row Reductions
 - 4.10.3 A Prime-Implicant Selection Procedure
 - 4.11* Decimal Method for Obtaining Prime Implicants
 - 4.12* The Multiple-Output Simplification Problem
 - 4.12.1 Multiple-Output Prime Implicants
 - 4.13* Obtaining Multiple-Output Minimal Sums and Products
 - 4.13.1 Tagged Product Terms
 - 4.13.2 Generating the Multiple-Output Prime Implicants
 - 4.13.3 Multiple-Output Prime-Implicant Tables
 - 4.13.4 Minimal Sums Using Petrick's Method
 - 4.13.5 Minimal Sums Using Table Reduction Techniques
 - 4.13.6 Multiple-Output Minimal Products
 - 4.14* Variable-Entered Karnaugh Maps
 - 4.14.1 Constructing Variable-Entered Maps
 - 4.14.2 Reading Variable-Entered Maps for Minimal Sums
 - 4.14.3 Minimal Products
 - 4.14.4 Incompletely Specified Functions
 - 4.14.5 Maps Whose Entries Are Not Single-Variable Functions
- Problems 152
- Chapter 5**
Logic Design with MSI Components and Programmable Logic Devices 160
- 5.1 Binary Adders and Subtracters 161
 - 5.1.1 Binary Subtracters 163
 - 5.1.2 Carry Lookahead Adder 166
 - 5.1.3 Large High-Speed Adders Using the Carry Lookahead Principle 168
 - 5.2 Decimal Adders 172
 - 5.3 Comparators 176
 - 5.4 Decoders 178
 - 5.4.1 Logic Design Using Decoders 179
 - 5.4.2 Decoders with an Enable Input 186
 - 5.5 Encoders 190
 - 5.6 Multiplexers 192
 - 5.6.1 Logic Design with Multiplexers 196
 - 5.7 Programmable Logic Devices (PLDs) 206
 - 5.7.1 PLD Notation 209
 - 5.8 Programmable Read-Only Memories (PROMs) 209
 - 5.9 Programmable Logic Arrays (PLAs) 213
 - 5.10 Programmable Array Logic (PAL) Devices 222
 - Problems 224
- Chapter 6**
Flip-Flops and Simple Flip-Flop Applications 231
- 6.1 The Basic Bistable Element 232
 - 6.2 Latches 233
 - 6.2.1 The SR Latch 234
 - 6.2.2 An Application of the SR Latch: A Switch Debouncer 235
 - 6.2.3 The \overline{SR} Latch 237

- 6.2.4 The Gated *SR* Latch 238
 - 6.2.5 The Gated *D* Latch 239
 - 6.3 Timing Considerations 240
 - 6.3.1 Propagation Delays 240
 - 6.3.2 Minimum Pulse Width 242
 - 6.3.3 Setup and Hold Times 242
 - 6.4 Master-Slave Flip-Flops (Pulse-Triggered Flip-Flops) 243
 - 6.4.1 The Master-Slave *SR* Flip-Flop 244
 - 6.4.2 The Master-Slave *JK* Flip-Flop 247
 - 6.4.3 0's and 1's Catching 249
 - 6.4.4 Additional Types of Master-Slave Flip-Flops 250
 - 6.5 Edge-Triggered Flip-Flops 251
 - 6.5.1 The Positive-Edge-Triggered *D* Flip-Flop 251
 - 6.5.2 Negative-Edge-Triggered *D* Flip-Flops 254
 - 6.5.3 Asynchronous Inputs 254
 - 6.5.4 Additional Types of Edge-Triggered Flip-Flops 256
 - 6.5.5 Master-Slave Flip-Flops with Data Lockout 258
 - 6.6 Characteristic Equations 259
 - 6.7 Registers 262
 - 6.8 Counters 267
 - 6.8.1 Binary Ripple Counters 267
 - 6.8.2 Synchronous Binary Counters 270
 - 6.8.3 Counters Based on Shift Registers 275
 - 6.9 Design of Synchronous Counters 277
 - 6.9.1 Design of a Synchronous Mod-6 Counter Using Clocked *JK* Flip-Flops 278
 - 6.9.2 Design of a Synchronous Mod-6 Counter Using Clocked *D*, *T*, or *SR* Flip-Flops 282
 - 6.9.3 Self-Correcting Counters 286
- Problems 288

Chapter 7

Synchronous Sequential Networks 297

- 7.1 Structure and Operation of Clocked Synchronous Sequential Networks 298

7.2 Analysis of Clocked Synchronous Sequential Networks 301

- 7.2.1 Excitation and Output Expressions 303
- 7.2.2 Transition Equations 304
- 7.2.3 Transition Tables 305
- 7.2.4 Excitation Tables 307
- 7.2.5 State Tables 309
- 7.2.6 State Diagrams 310
- 7.2.7 Network Terminal Behavior 312

7.3 Modeling Clocked Synchronous Sequential Network Behavior 315

- 7.3.1 The Serial Binary Adder as a Mealy Network 315
- 7.3.2 The Serial Binary Adder as a Moore Network 318
- 7.3.3 A Sequence Recognizer 320
- 7.3.4 A 0110/1001 Sequence Recognizer 323
- 7.3.5 A Final Example 326

7.4 State Table Reduction 328

- 7.4.1 Determining Equivalent Pairs of States 329
- 7.4.2 Obtaining the Equivalence Classes of States 335
- 7.4.3 Constructing the Minimal State Table 336
- 7.4.4 The 0110/1001 Sequence Recognizer 340

7.5 The State Assignment 345

- 7.5.1 Some Simple Guidelines for Obtaining State Assignments 348
- 7.5.2 Unused States 352

7.6 Completing the Design of Clocked Synchronous Sequential Networks 354

- 7.6.1 Realizations Using Programmable Logic Devices 362
- Problems 366

Chapter 8

Algorithmic State Machines 374

- 8.1 The Algorithmic State Machine 374
- 8.2 ASM Charts 377
 - 8.2.1 The State Box 378
 - 8.2.2 The Decision Box 379
 - 8.2.3 The Conditional Output Box 380