

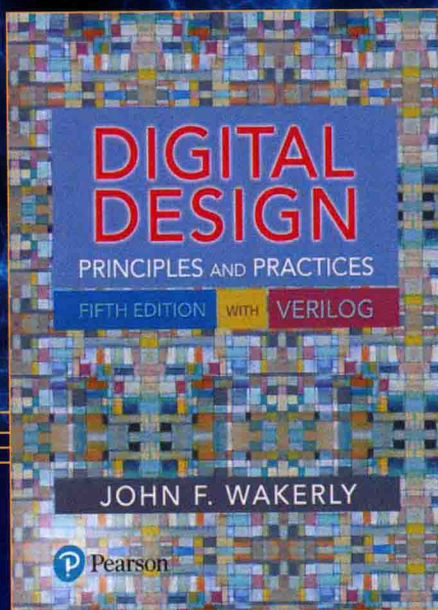
数字设计

原理与实践

(英文版·第5版)

[美] 约翰·F. 韦克利 (John F. Wakerly) 著
斯坦福大学

Digital Design
Principles and Practices
Fifth Edition



国外电子与电气工程技术丛书

数字设计

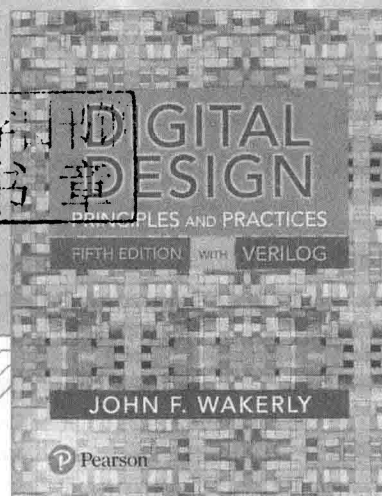
原理与实践

(英文版·第5版)

[美] 约翰·F. 韦克利 (John F. Wakerly) 著
斯坦福大学

Digital Design
Principles and Practices
Fifth Edition

常州大学图书馆
藏书章



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

数字设计: 原理与实践 (英文版·第5版) / (美) 约翰·F. 韦克利 (John F. Wakerly) 著.
—北京: 机械工业出版社, 2018.5

(国外电子与电气工程技术丛书)

书名原文: Digital Design: Principles and Practices, Fifth Edition

ISBN 978-7-111-59941-8

I. 数… II. 约… III. 数字电路—电路设计—教材—英文 IV. TN79

中国版本图书馆 CIP 数据核字 (2018) 第 091764 号

本书版权登记号: 图字 01-2017-7338

Authorized Reprint from the English language edition, entitled *Digital Design: Principles and Practices*, 5E, John F. Wakerly, published by Pearson Education, Inc., Copyright © 2018, 2006, 2000 by Pearson Education, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

English language edition published by China Machine Press, Copyright © 2018.

本书英文影印版由 Pearson Education Inc. 授权机械工业出版社独家出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

此影印版仅限于中华人民共和国境内 (不包括香港、澳门特别行政区及台湾地区) 销售发行。

本书封面贴有 Pearson Education (培生教育出版集团) 激光防伪标签, 无标签者不得销售。

出版发行: 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: 曲 熠

责任校对: 殷 虹

印 刷: 北京文昌阁彩色印刷有限责任公司

版 次: 2018 年 6 月第 1 版第 1 次印刷

开 本: 186mm×240mm 1/16

印 张: 55.75

书 号: ISBN 978-7-111-59941-8

定 价: 139.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzjsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光/邹晓东

出版者的话

文艺复兴以来,源远流长的科学精神和逐步形成的学术规范,使西方国家在自然科学的各个领域取得了垄断性的优势;也正是这样的优势,使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中,美国的产业界与教育界越来越紧密地结合,信息学科中的许多泰山北斗同时身处科研和教学的最前线,由此而产生的经典科学著作,不仅擘划了研究的范畴,还揭示了学术的源变,既遵循学术规范,又自有学者个性,其价值并不会因年月的流逝而减退。

近年,在全球信息化大潮的推动下,我国的信息产业发展迅猛,对专业人才的需求日益迫切。这对我国教育界和出版界都既是机遇,也是挑战;而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下,美国等发达国家在其信息科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此,引进一批国外优秀教材将对我国教育事业的发展起到积极的推动作用,也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章公司较早意识到“出版要为教育服务”。自1998年开始,我们就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力,我们与Pearson、McGraw-Hill、Elsevier、John Wiley & Sons、CRC、Springer等世界著名出版公司建立了良好的合作关系,从他们现有的数百种教材中甄选出Alan V. Oppenheim、Thomas L. Floyd、Charles K. Alexander、Behzad Razavi、John G. Proakis、Stephen Brown、Allan R. Hambley、Albert Malvino、Peter Wilson、H. Vincent Poor、Hassan K. Khail、Gene F. Franklin、Rex Miller等大师名家的经典教材,以“国外电子与电气工程技术丛书”和“国外工业控制与智能制造丛书”为系列出版,供读者学习、研究及珍藏。这些书籍在读者中树立了良好的口碑,并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也越来越多被实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑,这些因素使我们的图书有了质量的保证。随着电气与电子信息学科建设的不断完善和教材改革的逐渐深化,教育界对国外电气与电子信息教材的需求和应用都将步入一个新的阶段,我们的目标是尽善尽美,而反馈的意见正是我们达到这一终极目标的重要帮助。华章公司欢迎老师和读者对我们的工作提出建议或给予指正,我们的联系方式如下:

华章网站: www.hzbook.com

电子邮件: hzjsj@hzbook.com

联系电话: (010) 88379604

联系地址: 北京市西城区百万庄南街1号

邮政编码: 100037



华章科技图书出版中心

前言

本书写给所有需要设计和构建真正的数字电路的读者。为达到这个目的，读者必须掌握数字电路的基本原理，同时理解它们在真实世界中是如何运转的。正是基于这一理念，我们选取了“原理与实践”这两大主题。

在过去的 30 年里，随着集成电路的速度和集成度的快速提高，数字设计实践经历了重大转变。过去，数字设计者用成千甚至上万的门电路和触发器来构建系统，因此专业课程的重点就是如何最小化和有效地利用芯片及板级资源。

今天，一个芯片可以包含几千万个晶体管，并且可以通过编程来构建片上系统。过去要实现这样的系统，需要用几百个包含上百万单个门电路和触发器的分立芯片。如今，产品开发能否成功更多地取决于设计团队正确、完整地定义产品详细功能的能力，而不是将需要的所有电路集成到一块电路板或芯片上的能力。因此，现代专业课程的重点是设计方法论和软件工具，包括硬件描述语言（HDL）。设计团队利用 HDL 便可以完成超大型分层数字系统的设计。

借助 HDL，一方面可以看到典型设计的抽象层次变得更高，超越了单个门电路和触发器的层次。但与此同时，芯片级和电路板级数字电路的速度和集成度的提高，也迫使设计者必须在较低电子电路级提升竞争力。

大多数优秀的数字设计师都精通或者至少熟悉上述两个抽象层次。本书提供了高级（HDL）、低级（电子电路）以及“广泛中间级”（门电路、触发器和一些较高级的数字设计构件）的多层次基础知识。

目标读者

本书可以作为电气工程、计算机工程或计算机专业数字逻辑设计课程的入门教材或中级教材。对于计算机科学专业的学生，如果不熟悉基本电子学概念或者对数字器件的电气特性不感兴趣，可以跳过第 14 章，书中其他章节基本上都不需要这一章内容作为预备知识。另一方面，具有基本电子学基础的读者，则可通过阅读第 14 章来快速掌握数字电子学知识。

虽然本书是入门级的，但却比普通入门教材包含更多内容。我希望常规课程采用书中不超过三分之二的材料，但是，每门课程所用到的的是不同的三分之二。因此，请各位教师或读者按照自己的需要决定阅读内容。为了帮助大家做选择，我已经在可选章节的标题上打了星号。一般情况下，可以跳过这些章节而不影响后续必选章节的内容连贯性。此外，矩形文本框（如下所示）中的材料也是可以跳过的。

这是一个简单的文本框示例

一些读者抱怨之前的版本太厚了，我缩减了这一版的篇幅，但依然要提醒大家注意：

- 不必一字不落地读完所有内容，打星号的章节都可以略过。
- 写在这种框里的内容通常也可以略过。
- 我要求出版社调大了字号（本书采用 11 磅字，而大部分科技类教材都采用 10 磅字），这样不仅阅读起来更加舒服，而且图表的位置也更加合理。随手翻开一页，你会发现大部分图表和对图表内容的讲解都位于同一页。（我亲自做的排版工作，着实花了一番心血。）
- 本书的主题非常全面，在学习后续课程时，你可以再回头翻看相关知识点，甚至参加工作后，依然能从书里学到新东西。不过，教科书总会过时，难逃被丢进垃圾桶的厄运，但也有些“幸运儿”成了书架上的藏品，它的命运最终将系于读者之手。

有些课程会把本书当作中级教程或实验课本来使用。高年级学生可以跳过基础内容而直接进入感兴趣的章节。一旦掌握了基础知识，你会发现 Verilog 数字设计实例中包含更多重要的和有趣的知识。

建议读者好好利用贯穿全书的页边注释，这将帮助你把握重要的定义和知识点。

各章内容概述

接下来简短介绍一下本书 15 章的主要内容。这可能会让你想起一般软件指南中所说的，“写给不喜欢阅读手册的人”。看过这部分后，也许你就不必阅读本书的剩余内容了。

第 1 章给出了一些基本定义和重要主题的预览，还有一点关于数字电路的内容，使读者在不深入阅读第 14 章的情况下可以顺利学习其他内容。

第 2 章介绍二进制数制和编码。在软件课程中已经熟悉二进制数制的读者，仍需要阅读 2.10 ~ 2.13 节，以便理解硬件是如何使用二进制编码的。高年级学生可以阅读 2.14 节和 2.15 节，其中清楚地介绍了检错码。每个读者都应该阅读 2.16.1 节，因为在许多现代系统中都要用到它。

第 3 章讲解组合逻辑设计原理，包括开关代数、组合电路分析、综合与最小化。

第 4 章从文档标准开始介绍各种数字设计实践，文档标准可能是设计者开始实践时的第一步，也是很重要的一步。然后介绍定时的概念，特别是组合电路的定时，最后以关于 HDL、设计流程和工具的讨论结束。

第 5 章是关于 Verilog 的教程和参考资料，Verilog 是本书剩余部分所采用的硬件描述语言。

所有读者都应该阅读前几节内容,但是部分读者可能希望跳过其余内容,直到需要时再开始阅读。这也是没问题的,因为新的 Verilog 结构集中在后续章节(主要是第 6 章),第一次用到时才会进行讲述。

第 6 章描述了两个通用的组合构件 ROM 和 PLD。然后介绍两个最常用的功能构件——译码器和多路复用器,每一个都会给出门级和基于 Verilog 的设计。读者可以从这里直接跳到第 9 章的状态机,然后再回到第 7 章和第 8 章。

第 7 章继续讨论门级和用 Verilog 实现的组合构件,包括三态器件、优先编码器、异或门和奇偶函数以及比较器,然后用一个非平凡“随机逻辑”函数的 Verilog 设计实例引出结论。

第 8 章讨论实现算术功能的组合电路,包括加法和减法、移位、乘法和除法。

第 9 章介绍使用 D 触发器的传统状态机,包括采用状态表、状态图、ASM 图和 Verilog 的状态机的分析和综合。

第 10 章介绍其他时序元件,包括锁存器、边沿触发器件及其 Verilog 行为模型。这一章还描述了用典型的 FPGA 实现的时序元件,并为感兴趣的读者准备了关于时序型 PLD 和反馈时序电路的章节。

第 11 章重点讲解两个最常用的时序电路构件——计数器和移位寄存器,包括它们的应用,以及门级和基于 Verilog 的例子。

第 12 章包含关于如何采用 Verilog 对状态机建模的更详细的内容,以及更多例子。

第 13 章讨论时序电路设计的重要实际概念,包括同步系统结构、时钟和时钟偏移、异步输入和亚稳定性,以及一个用 Verilog 实现的双时钟同步的详细例子。

第 14 章介绍数字电路运算,重点在于逻辑器件的外部特性。学习起点是基础的电子学知识,包括电压、电流和欧姆定律。对于如何使真实电路运作起来不感兴趣的读者,或者有权把这项辛苦活交给别人完成的读者可以跳过本章。

第 15 章全部都是关于存储器件和 FPGA 的内容。存储器方面包括只读存储器以及静态、动态读/写存储器的内部电路和功能行为特性。最后一节会详细介绍 Xilinx 7 系列的 FPGA 结构。

大多数章节都包含参考资料、训练题和练习题。训练题通常是简答题或“轻而易举”的问题,可以直接根据书中材料给出答案,而练习题通常需要花时间仔细思考。第 14 章的训练题尤其广泛,是为了帮助非电子工程师能较容易地理解本章内容而专门设计的。

与第 4 版的区别

对于用过本书之前版本的读者和教师而言,除了普通的更新之外,第 5 版还有以下关键的不同之处。

- 新版只涉及 Verilog,没有 VHDL。在不同语言之间跳转只会使人分神。另外,Verilog 及

其后继 SystemVerilog 是目前非官方背景下所选择的 HDL。推荐大家阅读 Steve Golson 和 Leah Clark 的论文,“再论 21 世纪的语言之争: Verilog 与 VHDL”(2016 Synopsys Users Group Conference)。这是一篇精彩纷呈且论据充足的论文,如果你不想阅读全文的话,可以直接跳到最后一节。

- 新版有更多 HDL 的例子,更加强调设计流程和测试平台,包括纯粹的激励和自检信号。
- 对于并非旨在培养电子工程师的计算机工程专业的学生,为了使本书更容易阅读,关于 CMOS 电路的详细内容移到了第 14 章,而在第 1 章中加入了最少量的电子学知识,这样,学生就可以跳过整个关于 CMOS 的章节。
- 关于 TTL、SSI、MSI、74 系列逻辑、PLD 以及 CPLD 的内容都删除了。
- 简化了卡诺图化简的内容。
- 尽管第 5 章依然是关于 Verilog 的综合性教程和参考资料,但 Verilog 的概念也散布在第 6 章和第 7 章的文本框中。它们会在需要相关知识点的地方及时出现,学生可以直接在那里找到“好东西”。
- 更强调基于 FPGA 的设计、FPGA 结构特性、综合结果以及权衡。
- 原来关于组合逻辑元件的一章被分成了三章,以便有需要的读者可以直接从一开始就跳到状态机的内容。而且,还可以在最后讲解更多算术运算电路的内容。
- 用一整章内容讲解如何用 Verilog 实现状态机,包括许多例子。
- 关于同步设计方法论的章节目前包含一个详细的控制单元加数据通路的例子,以及一个关于采用异步 FIFO 的交叉时钟域的综合例子。
- 希望我在书里开的玩笑不会太糟糕。

数字设计软件工具

书中所有 Verilog 的例子都是采用 Xilinx Vivado 套件编译和测试的,这个套件包括以 Xilinx 7 系列 FPGA 为目标器件的 Verilog、SystemVerilog 以及 VHDL。然而,这些例子一般并不特别要求采用 Vivado 编译,甚至不要求目标器件是 Xilinx 或任何其他 FPGA。本书不包含关于 Vivado 的教程,而 Xilinx 有丰富的在线资源可供参考。因此,读者可以将本书与任何 Verilog 工具一起使用,包括下面提及的工具。

可以从 Xilinx 下载免费的 Vivado “Webpack” 版本。这个版本支持较小型的 7 系列 FPGA,带有 Zynq Soc 的 FPGA 以及评估板。这是一个庞大的文件,超过 10GB,但也是一个综合工具套件。支持前 7 系列 FPGA 以及较小型 Zynq FPGA 的 Xilinx ISE (Integrated Software Environment) 也包含在免费的 “Webpack” 版本中。注意,“legacy” 模式支持 ISE,而自从 2013 年后,ISE 就再也没有更新过。要获取任何一种套件,可以登录 www.xilinx.com,搜索 “Webpack download”。

如果你正在使用 Altera (现在是 Intel 的一部分了) 器件, 他们还提供了不错的“大学计划”和工具。你可以搜索“Altera university support”, 然后导航到“For Student”网页。免费的工具包括 Quartus Prime Lite 版本, 用于以入门级 FPGA 和 CPLD 为目标器件的 Verilog、SystemVerilog 和 VHDL 设计。还有一个配套的 ModelSim 模拟软件初始版本, 这一版本也是符合工业标准的。

Altera 和 Xilinx 都提供廉价的评估板, 适用于直接或通过第三方等效实现基于 FPGA 的学生项目。这样的评估板可以包括开关、LED、模拟/数字转换器以及运动传感器, 甚至包括 USB 和 VGA 接口。如果参与制造商的大学计划, 你的总花费将少于 100 美元。

另一个长期支持大学计划的专业数字设计工具是 Aldec (www.aldec.com)。他们提供一个流行的 Active-HDL 学生版本, 用于设计入门和模拟。其中不仅包含常规的 HDL, 还包括方框图和状态机的图形编辑器。模拟器中还包括波形编辑器, 用于创建交互激励信号。为利用这些特性, 可以将 Active-HDL 模拟器作为 Vivado 的一个插件, 取代 Vivado 模拟器。

上述工具以及大多数其他工程设计工具都是在 Windows PC 上运行的, 所以, 如果你是一个 Mac 迷, 就必须习惯使用 Window PC! 你可以在 Mac 的 Windows 仿真环境 (比如 VMware) 中运行, 但是成功与否取决于具体的软件工具。要想使一款工具在你的 PC 上“快速运行”, 最重要的条件就是配置固态硬盘驱动器而不是旋转硬盘驱动器。

即使你并未打算完成原创设计, 也可以利用上述工具中的任何一个来测试和改进书中的例子, 因为各章中的源代码都是在线提供的。

工程资源和 www.ddpp.com

可以从 Pearson 的“Engineering Resource”网站上获得本书的丰富资源。在出版时, Pearson 的链接是 media.pearsoncmg.com/bc/abp/engineering-resources。但是, 你知道输入一长串网址的麻烦。直接登录作者网站 www.ddpp.com 更为方便, 其中包含到 Pearson 网站的链接。而且, 作者网站中还包含最新的勘误表和其他增改资料, 甚至某天还可能会写博客。

Pearson 网站上的资源包括书中所有 Verilog 模块的源代码下载、部分训练题和练习题答案以及补充材料, 例如, 针对非电子工程人员提供的 20 页电子学基础概念介绍。

教师资源^①

Pearson 维护着一个专供教师使用的附加材料的集合。登录上述工程资源网站, 导航到这本书, 然后点击“Instructor Resources”链接。这个网站要求注册, 可能需要花费几天时间等待

① 关于本书教辅资源, 只有使用本书作为教材的教师才可以申请, 需要的教师请联系机械工业出版社华章公司, 电话: 136 0115 6823, 邮箱: wanguang@hzbook.com。——编辑注

以获得访问权限。所提供的资源包括附加的训练题和练习题答案、附加的源代码、更多的练习题，以及用于授课的简图和图表。本书之前版本的材料也会根据需求发布在网站上，以协助教师实现从旧版技术到新版课程的转换。

其他的教师资源还包括作者网站 www.ddpp.com，以及 Xilinx、Altera 和 Aldec 的大学计划，登录 www.ddpp.com 可以找到这些资料的最新链接。制造商的网站提供各种产品资料和课程资料，还有芯片和电路板的折扣信息，其中包含很多数字设计实验课常用的器件。有些功能齐全的工具套件也会参与促销，折扣力度很大，开设相关高级课程的教师可以多多关注。

勘误

警告：本书可能包含错误。作者和出版社对于因错误引起的问题免责，包括偶然事故、精神损害以及其他方面的问题。

这样可以让律师感到轻松。现在，为了让你感到快乐，我向你保证我们为本书做了精心准备，尽可能地不出错。我渴望获知仍然存在的错误，以便在重印、再版以及改编版本中改正。我会通过 PayPal 向第一个发现本书印刷版中某个错误——技术错误、印刷错误或是其他错误——的人支付 5 美元。请通过 www.ddpp.com 上的相关链接将你的意见通过电子邮件发给我。

www.ddpp.com 上提供最新的勘误列表，希望这个列表不会很长。

致谢

由于许多人的帮助才使得本书顺利出版。大多数人都对前四版的出版给予了帮助，在那里我已经表示了感谢。关于本书的“原理”方面，我还是要特别感谢我的老师、研究生导师以及朋友 Ed McCluskey。关于本书的“实践”方面，几位良师益友提供了很多好建议，包括我的朋友 Jesse Jenkins，Xilinx 的职员 Parimal Patel 和 Trevor Bauer，以及 McCluskey 的同事——斯坦福大学的 Subhasish Mitra 教授。

自本书第 4 版出版以来，我从读者那里收到了许多有益的意见。除了提出改进建议外，读者还指出了大量印刷上的和技术上的错误，所有这些都在第 5 版中一并改正。

对这个版本最具实质性影响和贡献的是匿名（对我而言）的学术评审，他们都是使用本书第 4 版或其他同类书籍作为教材的教师。我尽量接受他们的建议，这通常意味着要删去一些像我这样的“老前辈”所过于固守的材料，而增加大量现代概念，例如与基于 HDL 的设计流程、测试平台、综合等相关的内容。

感谢 Pearson 的责任编辑 Julie Bai 在过去几年为这个项目所做的精心且细致的工作，实际上，她是采用本书之前版本作为教材教授过数字设计课程的第一个编辑。遗憾的是，她在项目

差不多完成时离职了，我已经记不清这种状况是第四次还是第五次发生。这也使我更加确信，与我合作的编辑要么“不堪折磨逃离魔掌”，要么“历尽劫难浴火重生”，或者二者兼而有之。特别感谢她的老板的老板 Marcia Horton，二三十年来她一直关注我的项目。还要感谢 Scott Disanno 和 Michelle Bayman，他们负责这个版本的生产和发行过程。

还要感谢艺术家 Peter Crowell，我在 eBay 上发现了他的画作，当时，编辑 Julie Bai 建议我们基于 Piet Mondrian 的作品设计封面，她说，Piet 的某些作品“看起来几乎就像是逻辑电路的抽象”。Crowell 的“Tuesday Matinee”完美地契合了我们的需求。他的画作“铺设”在封面和每章开篇的地方，与铺设在 FPGA 上的逻辑模块及其连接非常相像。封面设计师 Marta Samsel 采纳了我这个工程主义的观点，最终成品非常漂亮。

最后，妻子 Joanne Jacobs 非常支持这个项目，让我在楼上安静地工作，而她则在楼下处理教育博客。她甚至从未抱怨圣诞树在家里一直放到二月份还没有搬走。

John F. Wakerly

加利福尼亚州，洛思阿图斯

Contents

1	INTRODUCTION	1
1.1	About Digital Design	1
1.2	Analog versus Digital	3
1.3	Analog Signals	7
1.4	Digital Logic Signals	7
1.5	Logic Circuits and Gates	9
1.6	Software Aspects of Digital Design	13
1.7	Integrated Circuits	16
1.8	Logic Families and CMOS	19
1.9	CMOS Logic Circuits	20
1.10	Programmable Devices	25
1.11	Application-Specific ICs	27
1.12	Printed-Circuit Boards	28
1.13	Digital-Design Levels	29
1.14	The Name of the Game	33
1.15	Going Forward	34
	Drill Problems	34
2	NUMBER SYSTEMS AND CODES	35
2.1	Positional Number Systems	36
2.2	Binary, Octal, and Hexadecimal Numbers	37
2.3	Binary-Decimal Conversions	39
2.4	Addition and Subtraction of Binary Numbers	42
2.5	Representation of Negative Numbers	44
	2.5.1 Signed-Magnitude Representation	2.5.2 Complement Number Systems
	2.5.3 Two's-Complement Representation	
	2.5.4 Ones'-Complement Representation	2.5.5 Excess Representations
2.6	Two's-Complement Addition and Subtraction	48
	2.6.1 Addition Rules	2.6.2 A Graphical View
	2.6.3 Overflow	
	2.6.4 Subtraction Rules	2.6.5 Two's-Complement and Unsigned Binary Numbers

2.7	Ones'-Complement Addition and Subtraction	52
2.8	Binary Multiplication	54
2.9	Binary Division	56
2.10	Binary Codes for Decimal Numbers	57
2.11	Gray Code	60
2.12	Character Codes	62
2.13	Codes for Actions, Conditions, and States	64
2.14	n-Cubes and Distance	66
2.15	Codes for Detecting and Correcting Errors	67
	2.15.1 Error-Detecting Codes	
	2.15.2 Error-Correcting and Multiple-Error-Detecting Codes	
	2.15.3 Hamming Codes	2.15.4 CRC Codes
	2.15.5 Two-Dimensional Codes	2.15.6 Checksum Codes
	2.15.7 m-out-of-n Codes	
2.16	Codes for Transmitting and Storing Serial Data	78
	2.16.1 Parallel and Serial Data	2.16.2 Serial Line Codes
	References	82
	Drill Problems	83
	Exercises	85
3	SWITCHING ALGEBRA AND COMBINATIONAL LOGIC	89
3.1	Switching Algebra	91
	3.1.1 Axioms	3.1.2 Single-Variable Theorems
	3.1.3 Two- and Three-Variable Theorems	3.1.4 n-Variable Theorems
	3.1.5 Duality	3.1.6 Standard Representations of Logic Functions
3.2	Combinational-Circuit Analysis	104
3.3	Combinational-Circuit Synthesis	110
	3.3.1 Circuit Descriptions and Designs	3.3.2 Circuit Manipulations
	3.3.3 Combinational-Circuit Minimization	3.3.4 Karnaugh Maps
3.4	Timing Hazards	122
	3.4.1 Static Hazards	3.4.2 Finding Static Hazards Using Maps
	3.4.3 Dynamic Hazards	3.4.4 Designing Hazard-Free Circuits
	References	126
	Drill Problems	128
	Exercises	129
4	DIGITAL DESIGN PRACTICES	133
4.1	Documentation Standards	133
	4.1.1 Block Diagrams	4.1.2 Gate Symbols
	4.1.3 Signal Names and Active Levels	4.1.4 Active Levels for Pins
	4.1.5 Constant Logic Signals	4.1.6 Bubble-to-Bubble Logic Design
	4.1.7 Signal Naming in HDL Models	4.1.8 Drawing Layout
	4.1.9 Buses	4.1.10 Additional Schematic Information
4.2	Circuit Timing	154
	4.2.1 Timing Diagrams	4.2.2 Propagation Delay
	4.2.3 Timing Specifications	4.2.4 Sample Timing Specifications
	4.2.5 Timing Analysis Tools	

4.3	HDL-Based Digital Design	165
4.3.1	<i>HDL History</i>	4.3.2 <i>Why HDLs?</i>
4.3.3	<i>EDA Tool Suites for HDLs</i>	4.3.4 <i>HDL-Based Design Flow</i>
	References	172
	Drill Problems	174
	Exercises	176
5	VERILOG HARDWARE DESCRIPTION LANGUAGE	177
5.1	Verilog Models and Modules	179
5.2	Logic System, Nets, Variables, and Constants	184
5.3	Vectors and Operators	189
5.4	Arrays	193
5.5	Logical Operators and Expressions	194
5.6	Compiler Directives	197
5.7	Structural Models	198
5.8	Dataflow Models	203
5.9	Behavioral Models (Procedural Code)	205
5.9.1	<i>Always Statements and Blocks</i>	5.9.2 <i>Procedural Statements</i>
5.9.3	<i>Inferred Latches</i>	5.9.4 <i>Assignment Statements</i>
5.9.5	<i>begin-end Blocks</i>	5.9.6 <i>if and if-else Statements</i>
5.9.7	<i>case Statements</i>	5.9.8 <i>Looping Statements</i>
5.10	Functions and Tasks	220
5.11	The Time Dimension	224
5.12	Simulation	225
5.13	Test Benches	226
5.14	Verilog Features for Sequential Logic Design	232
5.15	Synthesis	232
	References	233
	Drill Problems	234
	Exercises	235
6	BASIC COMBINATIONAL LOGIC ELEMENTS	237
6.1	Read-Only Memories (ROMs)	240
6.1.1	<i>ROMs and Truth Tables</i>	
6.1.2	<i>Using ROMs for Arbitrary Combinational Logic Functions</i>	
6.1.3	<i>FPGA Lookup Tables (LUTs)</i>	
6.2	Combinational PLDs	246
6.2.1	<i>Programmable Logic Arrays</i>	
6.2.2	<i>Programmable Array Logic Devices</i>	
6.3	Decoding and Selecting	250
6.3.1	<i>A More Mathy Decoder Definition</i>	6.3.2 <i>Binary Decoders</i>
6.3.3	<i>Larger Decoders</i>	6.3.4 <i>Decoders in Verilog</i>
6.3.5	<i>Custom Decoders</i>	6.3.6 <i>Seven-Segment Decoders</i>
6.3.7	<i>Binary Encoders</i>	
6.4	Multiplexing	281
6.4.1	<i>Gate-Level Multiplexer Circuits</i>	6.4.2 <i>Expanding Multiplexers</i>
6.4.3	<i>Multiplexers, Demultiplexers, and Buses</i>	
6.4.4	<i>Multiplexers in Verilog</i>	

	References	294
	Drill Problems	295
	Exercises	296
7	MORE COMBINATIONAL BUILDING BLOCKS	301
7.1	Three-State Devices	302
	7.1.1 Three-State Buffers	7.1.2 Standard MSI Three-State Buffers
	7.1.3 Three-State Outputs in Verilog	7.1.4 Three-State Outputs in FPGAs
7.2	Priority Encoding	312
	7.2.1 Cascading Priority Encoders	7.2.2 Priority Encoders in Verilog
7.3	Exclusive-OR Gates and Parity Functions	320
	7.3.1 Exclusive-OR and Exclusive-NOR Gates	
	7.3.2 Parity Circuits	7.3.3 Parity-Checking Applications
	7.3.4 Exclusive-OR Gates and Parity Circuits in Verilog	
7.4	Comparing	331
	7.4.1 Comparator Structure	7.4.2 Iterative Circuits
	7.4.3 An Iterative Comparator Circuit	7.4.4 Magnitude Comparators
	7.4.5 Comparators in HDLs	7.4.6 Comparators in Verilog
	7.4.7 Comparator Test Benches	
	7.4.8 Comparing Comparator Performance	
7.5	A Random-Logic Example in Verilog	356
	Drill Problems	363
	Exercises	364
8	COMBINATIONAL ARITHMETIC ELEMENTS	371
8.1	Adding and Subtracting	372
	8.1.1 Half Adders and Full Adders	8.1.2 Ripple Adders
	8.1.3 Subtractors	8.1.4 Carry-Lookahead Adders
	8.1.5 Group Ripple Adders	8.1.6 Group-Carry Lookahead
	8.1.7 MSI Arithmetic and Logic Units	8.1.8 Adders in Verilog
	8.1.9 Parallel-Prefix Adders	8.1.10 FPGA CARRY4 Element
8.2	Shifting and Rotating	403
	8.2.1 Barrel Shifters	8.2.2 Barrel Shifters in Verilog
8.3	Multiplying	416
	8.3.1 Combinational Multiplier Structures	8.3.2 Multiplication in Verilog
8.4	Dividing	426
	8.4.1 Basic Unsigned Binary Division Algorithm	
	8.4.2 Division in Verilog	
	References	433
	Drill Problems	433
	Exercises	434
9	STATE MACHINES	439
9.1	State-Machine Basics	440
9.2	State-Machine Structure and Analysis	443
	9.2.1 State-Machine Structure	9.2.2 Output Logic
	9.2.3 State-Machine Timing	
	9.2.4 Analysis of State Machines with D Flip-Flops	

9.3	State-Machine Design with State Tables	455
9.3.1	<i>State-Table Design Example</i>	
9.3.2	<i>State Minimization</i>	
9.3.3	<i>State Assignment</i>	
9.3.4	<i>Synthesis Using D Flip-Flops</i>	
9.3.5	<i>Beyond State Tables</i>	
9.4	State-Machine Design with State Diagrams	472
9.4.1	<i>T-Bird Tail Lights Example</i>	
9.5	State-Machine Design with ASM Charts	478
9.5.1	<i>T-Bird Tail Lights with ASM Charts</i>	
9.6	State-Machine Design with Verilog	483
	References	486
	Drill Problems	487
	Exercises	490
10	SEQUENTIAL LOGIC ELEMENTS	495
10.1	Bistable Elements	496
10.1.1	<i>Digital Analysis</i>	
10.1.2	<i>Analog Analysis</i>	
10.1.3	<i>Metastable Behavior</i>	
10.2	Latches and Flip-Flops	499
10.2.1	<i>S-R Latch</i>	
10.2.2	<i>$\overline{S}\text{-}\overline{R}$ Latch</i>	
10.2.3	<i>D Latch</i>	
10.2.4	<i>Edge-Triggered D Flip-Flop</i>	
10.2.5	<i>Edge-Triggered D Flip-Flop with Enable</i>	
10.2.6	<i>T Flip-Flops</i>	
10.3	Latches and Flip-Flops in Verilog	508
10.3.1	<i>Instance Statements and Library Components</i>	
10.3.2	<i>Behavioral Latch and Flip-Flop Models</i>	
10.3.3	<i>More about clocking in Verilog</i>	
10.4	Multibit Registers and Latches	522
10.4.1	<i>MSI Registers and Latches</i>	
10.4.2	<i>Multibit Registers and Latches in Verilog</i>	
10.5	Miscellaneous Latch and Bistable Applications	525
10.5.1	<i>Switch Debouncing</i>	
10.5.2	<i>Bus-Holder Circuits</i>	
10.6	Sequential PLDs	528
10.7	FPGA Sequential Logic Elements	531
10.8	Feedback Sequential Circuits	534
10.8.1	<i>Basic Analysis</i>	
10.8.2	<i>Analyzing Circuits with Multiple Feedback Loops</i>	
10.8.3	<i>Feedback Sequential-Circuit Design</i>	
10.8.4	<i>Feedback Sequential Circuits in Verilog</i>	
	References	544
	Drill Problems	545
	Exercises	547
11	COUNTERS AND SHIFT REGISTERS	553
11.1	Counters	554
11.1.1	<i>Ripple Counters</i>	
11.1.2	<i>Synchronous Counters</i>	
11.1.3	<i>A Universal 4-Bit Counter Circuit</i>	
11.1.4	<i>Decoding Binary-Counter States</i>	
11.1.5	<i>Counters in Verilog</i>	

11.2	Shift Registers	566
11.2.1	<i>Shift-Register Structure</i>	11.2.2 <i>Shift-Register Counters</i>
11.2.3	<i>Ring Counters</i>	11.2.4 <i>Johnson Counters</i>
11.2.5	<i>Linear Feedback Shift-Register Counters</i>	
11.2.6	<i>Shift Registers in Verilog</i>	11.2.7 <i>Timing-Generator Examples</i>
11.2.8	<i>LFSR Examples</i>	
11.3	Iterative versus Sequential Circuits	593
	References	596
	Drill Problems	596
	Exercises	599
12	STATE MACHINES IN VERILOG	605
12.1	Verilog State-Machine Coding Styles	606
12.1.1	<i>Basic Coding Style</i>	12.1.2 <i>A Verilog State-Machine Example</i>
12.1.3	<i>Combined State Memory and Next-State Logic</i>	12.1.4 <i>Reset Inputs</i>
12.1.5	<i>Pipelined Moore Outputs in Verilog</i>	
12.1.6	<i>Direct Verilog Coding Without a State Table</i>	
12.1.7	<i>State-Machine Extraction</i>	
12.2	Verilog State-Machine Test Benches	616
12.2.1	<i>State-Machine Test-Bench Construction Methods</i>	
12.2.2	<i>Example Test Benches</i>	
12.2.3	<i>Instrumenting Next-State Logic for Testing</i>	
12.2.4	<i>In Summary</i>	
12.3	Ones Counter	626
12.4	Combination Lock	628
12.5	T-Bird Tail Lights	632
12.6	Reinventing Traffic-Light Controllers	637
12.7	The Guessing Game	642
12.8	"Don't-Care" State Encodings	646
12.9	Decomposing State Machines	648
12.9.1	<i>The Guessing Game Again</i>	
12.10	The Trilogy Game	656
	References	664
	Drill Problems	664
	Exercises	666
13	SEQUENTIAL-CIRCUIT DESIGN PRACTICES	673
13.1	Sequential-Circuit Documentation Practices	674
13.1.1	<i>General Requirements</i>	13.1.2 <i>Logic Symbols</i>
13.1.3	<i>State-Machine Descriptions</i>	
13.1.4	<i>Timing Diagrams and Specifications</i>	
13.2	Synchronous Design Methodology	681
13.2.1	<i>Synchronous System Structure</i>	
13.2.2	<i>A Synchronous System Design Example</i>	
13.3	Difficulties in Synchronous Design	691
13.3.1	<i>Clock Skew</i>	13.3.2 <i>Gating the Clock</i>
13.3.3	<i>Asynchronous Inputs</i>	