HZ BOOKS

电子与电气工程丛书

PEARSON Prentice Hall

数字设计

原理与实践

（第4版·本科教学版）

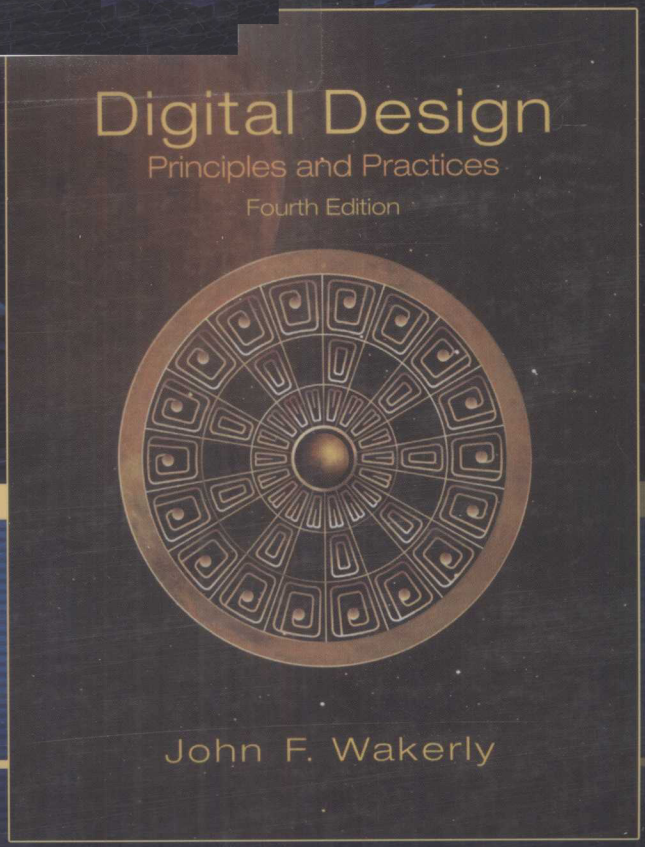
Digital Design

Principles and Practices

(Fourth Edition)

(美) John F. Wakerly 著
思科系统公司&斯坦福大学

林生 葛红 金京林 译
林生 审校
金京林 改编



机械工业出版社
China Machine Press

电子与电气工程丛书

数字设计

原理与实践

（第4版·本科教学版）

（美）John F. Wakerly 著
思科系统公司&斯坦福大学

林生 葛红 金京林 译
林生 审校
金京林 改编

机械工业出版社
China Machine Press

本书结合作者严谨的学术风范与丰富的实践背景，讲述了插件板级和VLSI系统中的数字设计基本原理和实践需求，提供了广泛的逻辑设计实践，给出了大量实际应用，并配有丰富的练习题。全书共分7章，主要内容包括：数制和编码，组合逻辑设计原理和实践，硬件描述语言（HDL），时序逻辑设计原理和实践，存储器、CPLD和FPGA。

本书条理清晰、简明易懂，可作为电气工程、计算机工程或计算机科学专业数字逻辑设计课程的教材，同时也可作为数字设计者的参考书。

本书版权登记号：图字：01-2009-5637

图书在版编目（CIP）数据

数字设计：原理与实践（第4版·本科教学版）/（美）韦克利（Wakerly, J. F.）著；林生，葛红，金京林译. —北京：机械工业出版社，2010.1

（电子与电气工程丛书）

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

ISBN 978-7-111-28973-9

Ⅰ. 数…　Ⅱ. ① 韦…　② 林…　③ 葛…　④ 金…　Ⅲ. 数字电路－电路设计　Ⅳ. TN79

中国版本图书馆CIP数据核字（2009）第199279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：曾　珊

北京京师印务有限公司印刷

2010年1月第1版第1次印刷

184mm × 260mm · 21.25印张

标准书号：ISBN 978-7-111-28973-9

定价：45.00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

购书热线：（010）68326294；88379649；68995259

投稿热线：（010）88379604

读者信箱：hzjsj@hzbook.com

译 者 序

继2003年我们翻译出版了这本教科书的第3版后，时隔才4年第4版又要跟广大读者见面了。对于一本教科书的作者来说，“如何能帮助学生去适应不可避免地要面临的各种变化，这才是最困难的。”主要是因为“在这个领域的一般教科书都因摩尔定律而缩短了它的适用期”。本书仍然在这个方面体现了它值得被推荐给广大读者的理由。

与前面的各个版本一样，本书仍然保留着共同的特色：条理清楚、生动有趣，增强了读者阅读的目的性和主动性；生动的比喻、有趣的描述和讨论，使抽象的概念和方法更加容易理解和掌握；内容上全面详尽，讲解上细致入微、循序渐进，习题丰富。

本书在内容安排上，较之上一个版本，由原来的11章删减到9章，但是整体篇幅并没有太多减少。被删减的部分主要是那些非核心、非原理性的内容，而增加的部分却是重点加强培养硬件编程能力方面的内容。正如作者在前言中所说的：“本书最重要和最精彩的部分，就是对硬件描述语言ABEL、VHDL和Verilog的讲解。”读者会发现，这些内容对提高硬件编程能力大有裨益，充分体现出本书“与时俱进”的新特点。

本书不仅适合作为计算机、电子、电气及控制等专业学生的教材，而且对于打算自学这方面内容的读者和技术人员，它也是一本不可多得的好书。

本书由华南师范大学计算机学院林生教授、葛红副教授和金京林副教授共同翻译完成，全书由林生教授审校和统稿。值得说明的是，本书的翻译工作是在上一版的基础上进行的，经征得原来所有译者同意，内容相同的部分仍然引用原来的译文。因此，我们要衷心感谢原来的所有译者。

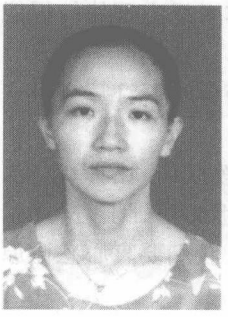
由于时间和水平有限，书中难免存在错误，敬请读者指正。

译 者

2007年1月

译者简介

林 生 男，华南师范大学计算机学院教授。先后任教于西安电子科技大学信息工程系和华南师范大学计算机科学系，多年从事数字逻辑与数字系统、计算机网络方向的课程教学与科研。在数字逻辑设计方面，编著有《时序逻辑电路设计原理》，译著有《数字系统设计基础》和《数字设计：原理与实践（原书第3版）》。在计算机网络方面，编著有《计算机通信网原理》和《计算机通信与网络教程》（第1版、第2版），译著有《计算机网络与因特网（原书第4版）》。



葛 红 女，华南师范大学计算机学院副教授。1989年，于重庆大学自动化系本科毕业。1997年，于华南理工大学自动化系获得硕士学位，2004年获得博士学位。多年从事数字逻辑和数字系统方向的课程教学和实验指导，主要译著有《数字设计：原理与实践（原书第3版）》和《VHDL设计指南》。

金京林 女，华南师范大学计算机学院副教授。1984年于吉林大学本科毕业，1987年于中国科学院长春光学精密机械和物理研究所获得硕士学位。先后在北京化工大学计算机系和华南师范大学计算机系从事教学和科研工作，主要研究方向是计算机体系结构。主要译著有《数字设计：原理与实践（原书第3版）》。

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，读者必须掌握基本原理，同时还必须理解它们在真实世界中的工作情况。本书正是在这种观念的基础上写作而成的，因此确定了“原理与实践”这个主题。

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非常大的转变。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，专业课程的重点就是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利用编程的方式创建片上系统。过去要实现这样的系统，需要用几百个包含了上百万的单个门电路和触发器的分立芯片来构造。如今，成功的产品开发更多地受限于设计团队正确、完整地定义产品详细功能的能力，而不是受限于团队将需要的所有电路集成到一个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可以完成非常大型的分层数字系统的设计。

一方面，利用HDL，我们看到典型设计的抽象层次移向更高的、单个门电路和触发器之上的层次。但是与此同时，芯片级和电路板级的数字电路的速度和集成度的提高，迫使许多数字设计者在较低的电子电路级更具竞争力。

大多数有价值的以及非常成功的数字设计者，都能够熟练地使用或者至少是精通上述两个抽象层次。本书提供了高级（HDL）、低级（电子电路）以及全面的“广泛的中间级”（门电路、触发器和一些较高级的数字设计构件）层次的基础知识。

目标读者

本书可以作为电气工程、计算机工程或计算机科学专业数字逻辑设计课程的入门教材。那些不具备电子学基础的学生，可以通过阅读20页的电子教材（在DDPPonline的Elec节中）而获得基础知识。DDPPonline是这本书基于Web的辅助材料的缩写。

虽然本书是入门级的，但比起一般的普通入门教材，它却包含更多的内容。在写作过程中，我发现有许多重要的东西要写进去，而这些东西又不适于斯坦福大学的一学期[⊖]课程或400页课本。因此，凭我个人的实践经验，把我认为比较重要的所有东西都写进去，由教师或读者自己去决定哪些是最需要的和最重要的。但为了有助于判断，我已经在一些可选节的标题上打了星号。在一般情况下，可以跳过这些节而不影响连贯性。在DDPPonline上可以找到更多的可选材料。

毫无疑问，有些人把本书当做高级教程和实验课本来使用。能力强的学生可以跳过基础部分而直接进入到所喜欢的那部分。本书中最重要和最精彩的部分，就是对硬件描述语言ABEL、VHDL和Verilog的讲解。你会发现，这些内容对提高硬件编程能力大有裨益。

另外，本书也可作为数字设计者的自学参考书。这一部分读者可分为两类：

⊖ 这是指每学年分为四学期制中的一学期。——编辑注

- 新手——刚刚开始从事数字设计，并且在学校里学过很“理论的”逻辑设计课程。这类读者应该集中学习相关知识为接触实际课题打基础，做准备。
- 老手——有经验的数字设计者。这类读者可能并不需要本书中的“实践性”材料，但是第1、2、5章的原理部分却有助于读者梳理思维，并且讨论了哪些内容是重要的。第4章和第6章中的举例应该会让这部分读者洞察和欣赏到各种各样的设计方法。最后，第3～6章中对ABEL、VHDL以及Verilog语言的描述和举例，都是专门为这部分读者学习基于HDL设计的方法而组织的。

给评论者和其他人的一些话

过去几年里，几个为本书写过评论的计算机科学专业学生和其他一些人对本书前一个版本所涉及的内容和讲述方式提出了意见。对于讲述方式，我只能为我没能力做得更好而道歉。但是，所选择的内容都是我认为一个未来的数字设计者应该知道的内容。由于我也雇佣过很多数字设计者，所以内容方面应该没什么可抱怨的，除非你读本书的目的不是想找一份工作或保留某份工作。

我并非故意回避计算机科学（CS）专业的学生或计算机科学本身。我喜欢CS，而且我本人也接受过很多CS专业教育，有很多CS方面的经验——我可以分析算法，了解编译器和操作系统的内部原理，可以用LISP等语言编写代码。但是今天，我从数字设计公司经理那里听到的最多抱怨，就是刚毕业的学生知道如何很好地写出Verilog或VHDL代码，但对于如何产生好的硬件结果却毫无认识，甚至根本不知道什么叫做“好”！

数字设计不仅仅是编写HDL代码——至少目前还不是。一个芯片设计公司的经理告诉我，他向自称知道如何完成基于HDL的硬件设计的应聘者提出的第一个问题就是，“在一个16位的计数器中有多少个触发器？”这并非是一个很难的问题，可是不能够回答这个问题的应聘者的数目却会让你惊讶！

因此，如果仅仅是为了满足一个CS或其他学位的需要而使用本书，那么我要先道歉，你也许应该选择更复杂的、更正式的、更枯燥的或充满术语的书，或者包含Frisbee语言的书。对前一版本真正有抱怨的都是来自于CS专业或其他非EE-CE的评论者。

另一方面，如果你有一天会成为电路级到系统级的数字设计者，那么务必请你学习本书中所有非选学的内容，包括VHDL或Verilog。学习这些内容应该不困难，因为许多喜欢前一版本的评论者称本书是精确的、逻辑性强的、清晰的、易于理解的、深刻但非常直观的、实用的、完整的、非常好的、非常完美的、有趣的、过渡自然的、诙谐的。当然，我承认本书有点傻或有时（比如现在）有点老土。而且，顺便说一下，甚至有一个CS评论者也说，“我喜欢这本书。”

并不像看起来那么长

有少数评论者认为本书的前一版本太长。目前推出的新版本要稍微短一些，但是，如果包括DDPPonline上的材料的话，实际内容就更多了。（确实如此，前一版本的页数[⊖]可以用10位二进制编码，而新版加上DDPPonline的页数则需要用11位二进制编码。）但是，请记住：

⊖ 这是指英文原书。——编辑注

- 并不需要阅读所有内容。标有星号的节和小节对于大多数读者而言是选读内容。
- 不必学习所有的HDL。HDL部分是相互独立的，各节和各小节的HDL部分也都与其他内容独立。因此，可以忽略任何或全部HDL的内容。但是，我还是建议你学习VHDL或Verilog。
- 书中加方框的内容通常是选读的。

各章说明

下面对本书7章的内容做一简短的说明。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”。看过这些简要的说明之后，也许你就不必阅读本书的其他内容了。

- 第1章介绍二进制数制和编码。已经从软件课程中熟悉了二进制数制的读者，仍需要阅读1.4和1.5节，以便理解硬件是如何使用二进制编码的。
- 第2章讲述组合逻辑设计原理，包括开关代数、组合电路分析、综合与最小化。
- 第3章先对基于VHDL的设计进行了一般性的介绍。
- 第4章首先介绍可编程逻辑器件（PLD），重点是它们实现组合逻辑函数的能力。该章剩下的部分描述常用的组合逻辑函数和应用。对于每一个函数，都介绍标准的MSI构件以及基于VHDL的设计。
- 第5章讲述时序逻辑设计原理，首先介绍锁存器和触发器。该章的重点是时钟同步状态机的分析与设计。但是，本章也大胆地介绍了基本模式电路以及反馈时序电路的分析与设计。在本章最后，为满足时序电路设计的需要，介绍了VHDL的特性和编码风格。
- 第6章讲述同步时序电路的设计实践。本章重点放在常用的函数方面，并给出应用MSI构件、VHDL的设计例子。6.6节、6.7节讨论在进行完全同步设计中不可避免的障碍，并阐述在异步环境中如何使时序同步这个重要问题。
- 第7章介绍存储器件、CPLD和FPGA。存储器部分包括只读存储器、静态和动态读/写存储器，而且是从内部电路和功能特性的观点进行介绍的。最后两节介绍CPLD和FPGA的体系结构。

大多数章末都含有参考资料、训练题和练习题。训练题都是些简短回答或启发性的问题，根据课文中的材料就能够直接回答出来；而练习题则可能要求多做一点思考。

许多附加的材料，包括本书第3版和该本科教学版中删除的内容，都可以在DDPPonline上找到。

www.ddpp.com、OneKey和DDPPonline

本书丰富的支持材料可以从网站www.ddpp.com上获得，其中包括免费的材料，例如一些习题的解答和最新的勘误；还有受保护的材料，例如补充章节和附加练习及解答。任何人都可以从本书的网站上获得免费材料。

只有使用OneKey的注册用户才可以获得受保护的材料。OneKey是由Prentice Hall在www.prenhall.com/OneKey上专为教师和学生提供的新的在线资源。每本书[⊖]均包含一个惟一的OneKey访问码。购买旧版的学生也可以单独购买访问码。

*在本书中，使用DDPPonline作为OneKey网站的别名，可以通过*www.prenhall.com/OneKey

⊖ 这是指英文原书。——编辑注

访问这个网站。必须使用访问码在这个网站上注册，然后每次使用这个网站都必须登录。另外，还可以通过www.ddpp.com或www.DDPPonline.com上的专门链接来访问OneKey。

DDPPonline包含下列学生资源：

- 超过300页的补充材料和设计举例，组织成20多节，每节从几页到30页。有些节还包含了附加的练习题。
- 如果教师订购了OneKey，还会得到附加的习题解答。教师可以选择要公布的习题解答。请注意，这里并没有给出每道题的答案，所以，如果教师没有给出某些习题的答案的话，请不要责怪他们。
- 本书以及补充章节中所有示例的C、ABEL、VHDL和Verilog程序的源文件。

在2005年秋季英文书出版之时，DDPPonline网站中包括下列补充章节（名称和标题）：

ABEL	各种各样的ABEL主题
BiPLD	双极型PLD
BJT	双极结型晶体管
CAD	计算机辅助设计工具
Cntr	计数器设计主题
Dec	译码器设计主题
DFT	可测试的设计
Diode	二极管和二极管逻辑
ECL	发射极耦合逻辑
Elec	Bruce M. Fleischer的电子电路评论
Enc	编码器设计主题
IEEE	IEEE标准符号
JKSM	利用J-K触发器的状态机的分析与综合
Min	组合逻辑最小化的附加主题
Mux	多路复用器设计主题
Pin	SSI、MSI、PLD以及ROM/RAM引脚
Pmin	可编程组合逻辑最小化主题
Rel	可靠性评估
Sreg	移位寄存器设计主题
TTL	关于TTL逻辑的附加主题
XCabl	利用ABEL进行组合逻辑设计举例
XCbb	利用MSI模块进行组合逻辑设计举例
XCver	利用Verilog进行组合逻辑设计举例
XCvhd	利用VHDL进行组合逻辑设计举例
XPLD	X系列PLD
XSabl	利用ABEL进行时序逻辑设计举例
XSbb	利用MSI模块进行时序逻辑设计举例
XSver	利用Verilog进行时序逻辑设计举例
XSvhd	利用VHDL进行时序逻辑设计举例
Zo	发布平台和反应

敬告教师

在DDPPonline上有一些专供给任课教师用的辅助材料，但只有通过OneKey系统注册的教师才可以使用这部分内容。要注册或了解更多关于OneKey的内容，请访问www.prenhall.com/OneKey或联系Prentice Hall代理商。

在这个专为教师设置的站点，包含了本书中的所有图表。可以利用这些文件直接制作演示幻灯片，或者选择合适的材料嵌入到你自己设计的其他演示材料中。

这个网站上还包含部分习题解答，这些习题占书中习题的半数以上，约200页。利用OneKey的课程管理系统，可以选择其中一部分开放给在OneKey上注册了的学生。注意，有些解答（基本上是第3版的全部学生解答），无论你是否可访问OneKey，都可以在www.ddpp.com上找到。这个网址告诉你是哪些解答，另外，还包含从第3版到第4版的交叉引用。

对教师来说，还有其他资源，包括Xilinx的大学计划（www.xilinx.com/ programs/univ）和Aldec的教育计划（www.aldec.com/education/university）。Xilinx网站提供了大量的产品资料、课程资料以及用于数字设计实验课程的芯片和插件。Aldec网站提供了Aldec自己的软件包和第三方的兼容工具以及原型系统。

关于本书的市场信息和相关资源，以及最新的订购信息都可以在www.prenhall.com/wakerlyinfo上找到。

致谢

由于许多人的帮助才使得本书顺利出版。大多数人都是在前三版中给予了帮助，在那里我已经表示了感谢。虽然第4版的准备过程是比较孤单地进行的，但也得到了思科系统公司的朋友Prem Jain和Mike Volpi的帮助，这使得我的工作更加顺利。他们以及思科公司削减了我所应承担的任务，把本来需要10个月的本书修订时间缩短到5个月之内。

关于本书“原理”方面，我还是要特别感谢我的老师、研究生导师和朋友Ed McCluskey。关于“实践”方面，我发起的“数字设计师名人斋”的主要成员之一Dave Raaum对新的Verilog材料进行了审核并提出许多建议。

自本书第3版出版以来，我从读者那里收到了许多有益的意见。除了提出许多建议和改进的意见外，一些读者还指出了许多印刷上和技术上的错误。在第4版中，这些错误都一并进行了改正。

我十分感谢Prentice Hall的责任编辑Tom Robbins，他在过去的几年里为本书做了很多工作。他是在（基本）完成本书的一个项目后更换工作的第3或第4个编辑，这使我怀疑和我一起工作是否让人不可避免地崩溃或成功，或者二者兼而有之。特别感谢Tom的老板——Marcia Horton，她在Tom离开后接管了他的工作。如果你正在阅读这些内容的话，那么她所做的一项工作就是从火里救出了这一页！

印刷编辑和校对Jennie Kaufman也做了精彩的工作，他确保了本书的一致性并发现了印刷问题，包括第2版和第3版中几处其他人（包括我自己）都没有发现的问题。还要感谢产品编辑Scott Disanno，承蒙他在我与出版社之间进行了很顺利的沟通，并且在项目的最后阶段给予我很大的帮助。

感谢艺术家Ken Bakeman，我几年前用Google搜索“wakerly”时就发现了他的作品。他最初的全电子“画作”出现在本书的封底，后来封面也采用了这个作品。这种圆形图案据说

是2001年6月出现在英格兰北安普敦郡的Barrowden、靠近Wakerly Woods的庄稼地里的怪圈。现在，“Wakerly Woods”常常被错拼成“Wakerley Woods”，但我没什么可抱怨的。我过去居住在Waverley街，人们也总是会混淆这两个拼写。无论如何，非常感谢Ken提供他的作品作为本版的封面。

在前几版中，此时我会感谢我的妻子Kate，但我非常难过地告诉你，经过7个写书项目和近34年的婚姻生活，在与乳腺癌长期搏斗之后，Kate于2004年初去世了。因此，我提到在为本版书做准备时非常孤独，这一点也不夸张。然而，对于我、我的孩子、我们的家庭以及朋友而言，Kate永远在我们的心里。

在本书出版之前，最后的感谢送给我的新老朋友和助理作者——未婚妻Joanne Jacobs，她在本书的准备过程中给予我极大的支持和鼓励。

John F. Wakerly

于伊利诺伊州，Oakbrook Terrace

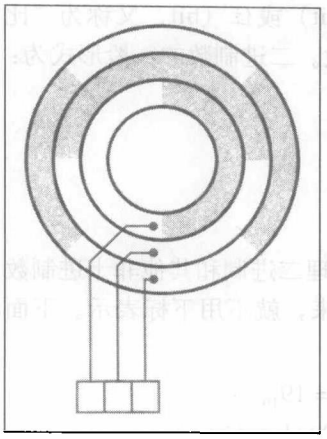
目 录

译者序
译者简介
前言

第1章 数制和编码 ……1
1.1 按位计数制 ……1
1.2 八进制和十六进制 ……2
1.3 常用按位计数制的转换 ……3
1.4 十进制数的二进制编码 ……5
*1.5 字符编码 ……7
参考资料 ……8
训练题 ……8
练习题 ……9
第2章 组合逻辑设计原理 ……10
2.1 开关代数 ……11
2.1.1 公理 ……11
2.1.2 单变量定理 ……13
2.1.3 二变量定理和三变量定理 ……13
2.1.4 *n*变量定理 ……14
2.1.5 对偶性 ……16
2.1.6 逻辑函数的标准表示法 ……18
2.2 组合电路分析 ……21
2.3 组合电路的综合 ……24
2.3.1 电路描述与设计 ……25
2.3.2 电路处理 ……27
2.3.3 组合电路最小化 ……28
2.3.4 卡诺图 ……30
2.3.5 最小化“积之和”表达式 ……31
2.3.6 其他最小化问题 ……37
2.3.7 程序化的最小化方法 ……37
*2.4 定时冒险 ……38
2.4.1 静态冒险 ……39
2.4.2 利用卡诺图发现静态冒险 ……40
2.4.3 动态冒险 ……41
2.4.4 设计无冒险电路 ……41
参考资料 ……42
训练题 ……43
练习题 ……44
第3章 硬件描述语言 ……48
3.1 基于HDL的数字设计 ……48
3.1.1 为什么用HDL ……48
3.1.2 HDL工具组 ……49
3.1.3 基于HDL的设计流程 ……50
3.2 VHDL硬件描述语言 ……52
3.2.1 程序结构 ……52
3.2.2 类型、常量和数组 ……55
3.2.3 函数和过程 ……58
3.2.4 库和包 ……61
3.2.5 结构形式的设计元素 ……62
3.2.6 数据流形式的设计元素 ……65
3.2.7 行为形式的设计元素 ……67
3.2.8 时间尺度 ……71
3.2.9 模拟 ……72
3.2.10 测试平台 ……73
3.2.11 时序逻辑设计的VHDL特性 ……74
3.2.12 综合 ……74
参考资料 ……75
训练题 ……76
练习题 ……76
第4章 组合逻辑设计实践 ……78
4.1 组合型PLD ……79
4.1.1 可编程逻辑阵列 ……79
4.1.2 可编程阵列逻辑器件 ……80
4.1.3 通用阵列逻辑器件 ……83
4.1.4 复杂型可编程逻辑器件 ……84
4.2 译码器 ……85
4.2.1 二进制译码器 ……85
4.2.2 大规模元件的逻辑符号 ……86
4.2.3 3-8译码器74x138 ……87
4.2.4 级联二进制译码器 ……90

4.2.5 用VHDL实现译码器 ……91
4.3 编码器 ……95
4.3.1 优先级编码器 ……96
4.3.2 优先级编码器74x148 ……96
4.3.3 用VHDL实现编码器 ……97
4.3.4 用Verilog实现编码器 ……99
4.4 三态器件 ……100
4.4.1 三态缓冲器 ……100
4.4.2 标准MSI三态缓冲器 ……102
*4.4.3 用VHDL实现三态输出 ……104
4.5 多路复用器 ……107
4.5.1 标准MSI多路复用器 ……108
4.5.2 扩展多路复用器 ……110
4.5.3 多路复用器、多路分配器和总线 …112
4.5.4 用VHDL实现多路复用器 ……114
4.6 “异或”门和奇偶校验电路……115
4.6.1 “异或”门和“异或非”门 ……115
4.6.2 奇偶校验电路 ……116
4.6.3 9位奇偶校验发生器74x280 ……117
4.6.4 奇偶校验的应用 ……118
4.6.5 用VHDL实现“异或”门和奇偶校验电路 ……119
4.7 比较器 ……121
4.7.1 比较器结构 ……122
4.7.2 迭代电路 ……123
4.7.3 迭代比较器电路 ……123
4.7.4 标准MSI大小比较器 ……124
4.7.5 用HDL实现比较器 ……127
4.7.6 用ABEL和PLD实现比较器 ……127
4.7.7 用VHDL实现比较器 ……128
4.7.8 用Verilog实现比较器 ……130
*4.8 加法器、减法器和ALU ……133
4.8.1 半加器和全加器 ……134
4.8.2 串行进位加法器 ……134
4.8.3 减法器 ……135
4.8.4 先行进位加法器 ……136
4.8.5 MSI加法器 ……137
4.8.6 MSI算术逻辑单元 ……140
4.8.7 组间先行进位 ……141
4.8.8 用VHDL实现加法器 ……142
参考资料……145
训练题……146
练习题……147
第5章 时序逻辑设计原理 ……151
5.1 双稳态元件 ……152
5.1.1 数字分析 ……153
5.1.2 模拟分析 ……153
5.1.3 亚稳态特性 ……153
5.2 锁存器与触发器 ……154
5.2.1 S-R锁存器 ……155
5.2.2 $\overline{S}$ - $\overline{R}$锁存器 ……157
5.2.3 具有使能端的S-R锁存器 ……157
5.2.4 D锁存器 ……158
5.2.5 边沿触发式D触发器 ……159
5.2.6 具有使能端的边沿触发式D触发器…161
5.2.7 扫描触发器 ……162
*5.2.8 主从式S-R触发器 ……163
*5.2.9 主从式J-K触发器 ……164
*5.2.10 边沿触发式J-K触发器 ……165
5.2.11 T触发器 ……166
5.3 时钟同步状态机分析 ……167
5.3.1 状态机结构 ……167
5.3.2 输出逻辑 ……168
5.3.3 特征方程 ……169
5.3.4 使用D触发器的状态机分析 ……169
5.4 时钟同步状态机设计 ……176
5.4.1 状态表设计举例 ……177
5.4.2 状态最小化 ……180
5.4.3 状态赋值 ……181
5.4.4 采用D触发器的综合 ……183
*5.4.5 采用J-K触发器的综合 ……185
5.4.6 采用D触发器的其他设计例子 ……186
5.5 用状态图设计状态机 ……189
5.6 用VHDL设计时序电路……194
5.6.1 时钟电路 ……194
5.6.2 用VHDL设计状态机 ……196
5.6.3 VHDL状态机举例 ……197
5.6.4 VHDL中的状态赋值 ……199
5.6.5 VHDL中的流水线型输出 ……200
5.6.6 不用状态表的直接VHDL编程 ……201

5.6.7 更多VHDL状态机例子 ……202
5.6.8 用VHDL定义触发器 ……204
5.6.9 VHDL状态机测试平台 ……205
5.6.10 反馈时序电路 ……208
参考资料……209
训练题……210
练习题……212
第6章 时序逻辑设计实践 ……215
6.1 锁存器和触发器 ……215
6.1.1 SSI型锁存器和触发器 ……215
*6.1.2 开关消颤 ……216
*6.1.3 最简单的开关消颤电路 ……217
*6.1.4 总线保持电路 ……218
6.1.5 多位寄存器和锁存器 ……219
6.1.6 用VHDL实现寄存器和锁存器 ……221
6.2 时序型PLD……224
6.2.1 时序型GAL器件 ……224
6.2.2 PLD定时规格说明 ……228
6.3 计数器 ……230
6.3.1 行波计数器 ……230
6.3.2 同步计数器 ……231
6.3.3 MSI型计数器及应用 ……231
6.3.4 二进制计数器状态的译码 ……235
6.3.5 用VHDL实现计数器 ……236
6.4 移位寄存器 ……239
6.4.1 移位寄存器结构 ……239
6.4.2 MSI移位寄存器 ……241
6.4.3 移位寄存器计数器 ……242
6.4.4 环形计数器 ……243
6.4.5 用VHDL实现移位寄存器 ……245
6.5 同步设计方法 ……248
6.6 同步设计中的障碍 ……250
6.6.1 时钟偏移 ……251
6.6.2 选通时钟 ……253
6.6.3 异步输入 ……254
6.9 同步器故障和亚稳定性 ……256
6.7.1 同步器故障 ……257
6.7.2 亚稳定性分辨时间 ……257
6.7.3 可靠同步器设计 ……258
6.7.4 亚稳定的定时分析 ……258
6.7.5 更好的同步器 ……260
6.7.6 其他同步器设计 ……261
6.7.7 同步高速数据传输 ……263
参考资料……271
训练题……272
练习题……273
第7章 存储器、CPLD和FPGA……277
7.1 只读存储器 ……277
7.1.1 ROM用于“随机”组合逻辑函数…278
*7.1.2 ROM的内部结构……280
*7.1.3 二维译码 ……282
7.1.4 商用ROM的类型……284
7.1.5 ROM的控制输入和定时……287
7.1.6 ROM的应用……289
7.2 读/写存储器 ……293
7.3 静态RAM ……294
7.3.1 静态RAM的输入和输出……294
7.3.2 静态RAM的内部结构……294
7.3.3 静态RAM的定时……296
*7.3.4 标准静态RAM……297
*7.3.5 同步SRAM ……299
7.4 动态RAM ……302
7.4.1 动态RAM的结构……302
7.4.2 SDRAM的定时 ……304
7.4.3 DDR SDRAM ……306
7.5 复杂可编程逻辑器件 ……307
7.5.1 Xilinx XC9500 CPLD系列……308
7.5.2 功能块体系结构 ……309
7.5.3 输入/输出块体系结构……311
7.5.4 开关矩阵 ……312
7.6 现场可编程门阵列 ……314
7.6.1 Xilinx XC4000 FPGA系列……314
7.6.2 可配置逻辑块 ……315
7.6.3 输入/输出块……317
7.6.4 可编程互连 ……318
参考资料……320
训练题……321
练习题……321



第1章

数制和编码

数字系统是由用来处理二进制数码0和1的电路所构建的，然而现实中的东西很少是完全基于二进制数的，所以数字系统的设计者必须在数字电路处理的二进制数码和实际的数字、事件、条件等事物之间建立某种对应的关系。本章的目的，就是要阐述在数字系统中大家所熟悉的数字量是如何被表示和处理的，以及那些非数值数据、事件、条件等事物又是如何被表示的。

前面的1.1～1.3节先叙述二进制数制，说明用这种数制怎样进行加、减、乘、除运算。1.4和1.5节阐述如何用二进制数码串对其他事物（例如，十进制数、文本字符，等）进行编码。

1.1 按位计数制

按位计数制（positional number system）是大家在学校学习过且在商务活动中每天都使用的传统计数制。在这种计数制中，用一串数码来表示一个数，每个数码的位置对应有一个相关的*权*（weight），该数的值就等于所有数码按权展开相加之和，例如：

$$1734 = 1 \times 1000 + 7 \times 100 + 3 \times 10 + 4 \times 1$$

与数码位置相对应，每个权均为10的幂，幂次可以为正也可以为负，由小数点决定：

$$5185.68 = 5 \times 1000 + 1 \times 100 + 8 \times 10 + 5 \times 1 + 6 \times 0.1 + 8 \times 0.01$$

一般而言，形如$d_1d_0.d_{-1}d_{-2}$的数D，其值为：

$$D = d_1 \times 10^1 + d_0 \times 10^0 + d_{-1} \times 10^{-1} + d_{-2} \times 10^{-2}$$

在此，10为计数制的*基数*（base或radix）。在一般的按位计数制中，基数r可以是任何大于等于2的整数，即$r \geqslant 2$。位置i上的数字其权值为r^i，所以数的一般形式为：

$$d_{p-1}d_{p-2}\cdots d_1d_0 \,.\, d_{-1}d_{-2}\cdots d_{-n}$$

这里，*小数点*（radix point）的左边有p位数码，右边有n位数码。如果没标出小数点，则假定其在最右边数码的右侧。数D的值为每个数码乘以其对应的权再求和：

$$D = \sum_{i=-n}^{p-1} d_i \cdot r^i$$

在按位计数制中，除去可能出现的前缀零和后缀零以外，数的表示是惟一的（显然，0185.6300等于185.63，等等）。在这样的数中，*最左边的数码称做最高有效数字或高阶数字*（most significant digit or high-order digit, MSD），*最右边的数码称做最低有效数字或低阶数字*（least significant digit or low-order digit, LSD）。

数字电路处理的信号通常是在诸如低或高、充电或放电、关或开等仅有的两个状态中取

一，这些信号代表着只有0和1两个取值的二进制数字（binary digit）或位（bit，又称为“比特”），因而数字系统中通常用二进制基数（binary radix）来表示数。二进制数的一般形式为：

$$b_{p-1}b_{p-2}\cdots b_1b_0 \,.\, b_{-1}b_{-2}\cdots b_{-n}$$

其值为：

$$B=\sum_{i=-n}^{p-1} b_i \cdot 2^i$$

在二进制中，小数点称为二进制小数点（binary point）。在处理二进制和其他非十进制数时，用下标表示每个数的基数。若基数从上下文中能明显地看出来，就不用下标表示。下面给出了二进制数及其等效的十进制数：

$$10011_2 = 1\times 16 + 0\times 8 + 0\times 4 + 1\times 2 + 1\times 1 = 19_{10}$$

$$100010_2 = 1\times 32 + 0\times 16 + 0\times 8 + 0\times 4 + 1\times 2 + 0\times 1 = 34_{10}$$

$$101.001_2 = 1\times 4 + 0\times 2 + 1\times 1 + 0\times 0.5 + 0\times 0.25 + 1\times 0.125 = 5.125_{10}$$

二进制数最左边的位是最高有效位或高阶位（most significant bit or high-order bit, MSB），最右边的是最低有效位或低阶位（least significant bit or low-order bit, LSB）。

1.2 八进制和十六进制

由于日常生活中使用的是十进制，基数10很重要；又因为数字电路中直接处理的是二进制数，基数2很重要。其他基数的计数制虽然常常不直接处理，但对文档编制或其他用途还是很重要的。尤其是基数8和基数16，方便了数字系统中多位数的简写。

八进制（octal number system）使用基数8，而十六进制（hexadecimal number system）使用基数16。表1-1列出了二进制整数0到1111及其等效的八进制数、十进制数、十六进制数。八进制需要8个数码，使用十进制数码0～7；十六进制需要16个数码，在十进制数码0～9的基础上增加字母A～F。

表1-1 二进制、十进制、八进制和十六进制数

二进制	十进制	八进制	3位二进制串	十六进制	4位二进制串
0	0	0	000	0	0000
1	1	1	001	1	0001
10	2	2	010	2	0010
11	3	3	011	3	0011
100	4	4	100	4	0100
101	5	5	101	5	0101
110	6	6	110	6	0110
111	7	7	111	7	0111
1000	8	10	—	8	1000
1001	9	11	—	9	1001
1010	10	12	—	A	1010
1011	11	13	—	B	1011
1100	12	14	—	C	1100
1101	13	15	—	D	1101
1110	14	16	—	E	1110
1111	15	17	—	F	1111

八进制和十六进制的基数均是2的幂，因而在表示多位二进制时很有用。如表1-1中第3列和第4列所示，1位八进制数码可以惟一地表示3位二进制串，这是因为3位二进制串有8种不同

的组合；同理，如表1-1中第5列和第6列所示，1位十六进制数码可以惟一地表示4位二进制串。

这样，很容易将二进制数转换为八进制。从二进制小数点开始向左每3位二进制数码划为一组，每组用其等效的八进制数码代替：

$$100011001110_2 = 100\ 011\ 001\ 110_2 = 4316_8$$

$$11101101110101001_2 = 011\ 101\ 101\ 110\ 101\ 001_2 = 355651_8$$

二进制到十六进制的转换也是类似的，只是要将每4位二进制划为一组：

$$100011001110_2 = 1000\ 1100\ 1110_2 = 8CE_{16}$$

$$11101101110101001_2 = 0001\ 1101\ 1011\ 1010\ 1001_2 = 1DBA9_{16}$$

在这些例子中，为使二进制数码是所需要的3或4的倍数，在最左边按需补充零。

如果二进制小数点右边也有数码，则从小数点开始向右每3位或4位一组，将二进制数转换成八进制或十六进制数。在转换过程中，不管左边还是右边都可以添零以使二进制数的位数为3或4的倍数，如下所示：

$$10.1011001011_2 = 010.101\ 100\ 101\ 100_2 = 1.5454_8$$

$$= 0010.1011\ 0010\ 1100_2 = 2.B2C_{16}$$

相反地，很容易将八进制或十六进制转换为二进制，只需简单地将每个八进制或十六进制数码用对应的3或4位二进制串替代即可，如下所示：

$$1357_8 = 001\ 011\ 101\ 111_2$$

$$2046.17_8 = 010\ 000\ 100\ 110.001\ 111_2$$

$$BEAD_{16} = 1011\ 1110\ 1010\ 1101_2$$

$$9F.46C_{16} = 1001\ 1111.0100\ 0110\ 1100_2$$

30年前，八进制数已经相当流行，因为某些微型计算机面板的控制灯和开关是按3位一组排列的。然而，现今八进制数用得不多了，因为计算机的优势在于处理字节（byte），一个字节等于8位二进制数，很难从八进制表示的多字节量中抽取单字节的值。例如，在八进制表示的32位二进制数12345670123$_8$中，其4个字节对应的八进制值分别是多少？

在我64岁的时候

随着年龄的增长，你会发现十六进制更有用。40岁那年，我告诉朋友我刚刚过了28_{16}岁了。当然，要压着嗓音低声说“十六进制”。在50岁时，那就是32_{16}岁了。

人们在20、30、40、50等10周年生日时都很高兴，但是应该使你的朋友确信：十进制并不具有基本意义。当在你的岁数上加一位最高有效位（MSB）时，更多重要的生活变化就会发生在2、4、8、16、32和64等周年上了。你认为Beatles为什么要歌唱“当我64岁的时候”？

在十六进制中，两个数码表示一个8位字节，2n个数码表示具有n个字节的字；每一对数码刚好组成一个字节。例如，32位的十六进制数5678ABCD$_{16}$由4个字节构成，其值分别为56_{16}、78_{16}、AB$_{16}$和CD$_{16}$。结合上下文，一个4位的十六进制数码有时也称做半字节（nibble），32位（4字节）含8个半字节。十六进制数经常用来描述计算机存储器的地址空间，如：有16位地址的计算机，随机存储器所占据的空间为0～EFFF$_{16}$，只读存储器所占据的空间为F000～FFFF$_{16}$。许多计算机编程语言用前缀“0x”表示十六进制数，如0xBFC0000。

1.3 常用按位计数制的转换

两个基数间的转换一般不能通过简单的替换来完成，需要算术操作。在这一节将说明怎