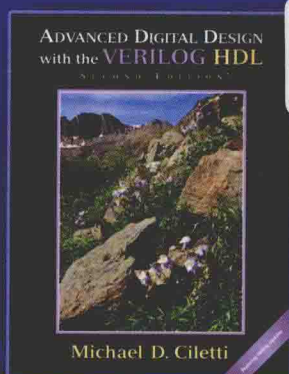


国外电子与通信教材系列

PEARSON

Verilog HDL 高级数字设计 (第二版)

Advanced Digital Design with the Verilog HDL
Second Edition



[美] Michael D. Ciletti 著

李广军 林水生 阎波 等译



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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

国外电子与通信教材系列

Verilog HDL 高级数字设计 (第二版)

Advanced Digital Design with the Verilog HDL
Second Edition

[美] Michael D. Ciletti 著

李广军 林水生 阎波 等译

电子工业出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书依据数字集成电路系统工程开发的要求与特点,利用 Verilog HDL 对数字系统进行建模、设计与验证,对 ASIC/FPGA 系统芯片工程设计开发的关键技术与流程进行了深入讲解,内容包括:集成电路芯片系统的建模、电路结构权衡、流水线技术、多核微处理器、功能验证、时序分析、测试平台、故障模拟、可测性设计、逻辑综合、后综合验证等集成电路系统的前后端工程设计与实现中的关键技术及设计案例。书中以大量设计实例叙述了集成电路系统工程开发须遵循的原则、基本方法、实用技术、设计经验与技巧。

本书既可作为电子与通信、电子科学与技术、自动控制、计算机等专业领域的高年级本科生和研究生的教材或参考资料,也可用于电子系统设计及数字集成电路设计工程师的专业技术培训。

Authorized translation from the English language edition, entitled Advanced Digital Design with the Verilog HDL, Second Edition, 9780136019282 by Michael D. Ciletti, published by Pearson Education, Inc., publishing as Prentice Hall, Copyright ©2011 Pearson Higher Education.

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.

CHINESE SIMPLIFIED language edition published by PEARSON EDUCATION ASIA LTD., and PUBLISHING HOUSE OF ELECTRONICS INDUSTRY Copyright ©2014.

本书中文简体版专有版权由 Pearson Education(培生教育出版集团)授予电子工业出版社。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。

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。

版权贸易合同登记号 图字:01-2010-0894

图书在版编目(CIP)数据

Verilog HDL 高级数字设计:第2版/(美)西勒提(Ciletti, M. D.)著;李广军等译.

北京:电子工业出版社,2014.2

书名原文:Advanced Digital Design with the Verilog HDL

国外电子与通信教材系列

ISBN 978-7-121-22193-4

I. ①V… II. ①西… ②李… III. ①硬件描述语言-程序设计-高等学校-教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2013)第 304091 号

策划编辑:马 岚

责任编辑:周宏敏

印 刷:三河市鑫金马印装有限公司

装 订:三河市鑫金马印装有限公司

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本:787×1092 1/16 印张:41.5 字数:1286 千字

印 次:2014 年 2 月第 1 次印刷

定 价:95.00 元



凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010)88254888。

质量投诉请发邮件至 zlt@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010)88258888。

前 言

精炼、明晰化与验证

用硬件描述语言(HDL)建立行为级模型是现代专用集成电路设计的关键技术。如今,大多数设计者使用基于硬件描述语言的设计方法,创建基于语言的高层、抽象的电路描述,以验证其功能和时序。在本书第一版的使用过程中,讲授设计方法学所用的语言(IEEE 1464-1995)已经历了两次修改,分别是IEEE 1364-2001及2005年的修订版,即Verilog-2001和Verilog-2005,以提高其有效性和效率。

这一版的编写动机和第一版基本是相同的。对那些准备在产品研发团队做出成绩的学生们来说,必须了解如何在设计流程的关键阶段使用硬件描述语言。因此,需要有一门在内容上超越先修课程“数字设计”中学习过的基本原则和方法的课程,本书就是为该课程而著的。

现在,市面上讨论硬件描述语言的书籍的数量已远远超过本书第一版出版时的数量。但是,这些书大部分都定位于解释语法,而不是如何运用语言进行设计,不太适合于课堂教学。本书的重点是硬件描述语言的设计方法学,因此语言本身只是一个配角。这一版中强化了如何通过实例证明,将一个数字系统描述并划分为数据通路、状态(反馈)信号和控制器(有限状态机)系统结构的重要性。我们认为,这种描述可使设计和验证复杂数字系统的方法更加清楚、直接、明了。本书给出了大量的仿真结果和注释,以帮助学生掌握时序机的操作过程,并深入理解由控制器产生的信号间的时序互动关系,数据通路的操作,以及从数据通路回馈给控制器的信号。其目的都是为了开发出可综合、无锁存且无竞争的设计。

Verilog 2001和2005的语言增强功能已用于重新描述和简化书中模型的代码。我们强调工业界通用的规范和风格,但并不鼓励不考虑模型能否被综合的学术模型风格。本书第二版已把第一版中处理同步FIFO的部分改为同步和异步FIFO,并给出了精心设计的例子,以解释使用异步FIFO来同步跨越时钟域的数据传输问题。

书中的设计实例已多次优化和改进^①。从设计方法学的角度,对一个嵌入式控制器,用C语言建模和用Verilog建模,这两种设计方法学之间存在着竞争和互补的关系。基于C的方法执行陈述性语句,而Verilog HDL模拟了某个机器的多个并发的行为动作。后一种设计方法对硬件进行编译,而前一种是编译预先存储在硬件单元中的语句。对于某个特定应用,Verilog模型编译的硬件在主机接口处生成了等效的I/O信号。对于嵌入式代码而言,其区别是不会产生等效的硬件。本书的目标就是讲授硬件建模/编译的范例,并预测综合实现后的结果。C语言编程是预测程序产生的数据,而状态机/处理器的应用却显而易见。作为对比,用Verilog描述的模型预测该硬件将产生应用所需求的I/O信号,因此需要开发者根据寄存器操作时序控制进行思考和设计。Verilog的模型鼓励学习者理解一个数字电路和系统的本质。

本书要求学生已学过逻辑设计的入门课程,本书的目标是:(1)简要复习组合时序逻辑的基本原理,(2)介绍HDL在设计中的应用,(3)强调的是快速设计通过ASIC和/或FPGA实现的电路设计描述风格,(4)提供具有一定难度的设计实例。章末习题的目的是鼓励学生精炼、明晰化

^① 登录华信教育资源网 www.hxedu.com.cn 可注册下载本书相关配套资源。

并验证他们自己的设计。从本质上讲,许多习题均为开放式的设计,要求验证以达到所要求的设计规范。

广泛使用的 Verilog 硬件描述语言(IEEE 1364 标准),作为一个公共框架为本书的设计实例的讨论提供了支持。第一版重点关注数字电路的设计、验证和综合,而不是 Verilog 语言本身的语法,本版仍然保持这种风格。

选修数字设计中高级课程的大多数学生至少应该熟悉一种编程语言,并且在阅读本书时能够将其作为可以借鉴的背景知识。本书仅讨论 Verilog 的核心设计方法及其广泛使用的特性。为了强调在面向综合的设计环境中使用该语言,我们还特意将许多语法的细节、特点和解释放在附录中,以便于读者参考。附录中也提供了 Verilog 的所有形式化语法。

大部分数字设计的入门课程都介绍过通过状态转移图表示的有限状态机及算法状态机(ASM)图。同样,本书中也大量使用了 ASM 图,演示了其在设计时序状态机的行为模型中的功用。对利用 ASMD 图(即通过标注显示出被控数据通道的寄存器操作的 ASM 图)系统地设计有限状态机来控制数字状态机中复杂数据通道的重要问题,进行了深入论述。并将精简指令集计算机中央处理器(RISC CPU)和其他重要硬件单元的设计作为实例给出。我们的支持网站上包含了 RISC 计算机的源代码和可用于应用程序开发的汇编程序。这个汇编程序也可作为研究鲁棒性更好的指令集和其他派生架构的基础。

本书完整地引入了 Verilog 语言,但仅在支持设计实例的需要时才进行详细说明。正文中使用了大量的实例,讲解使用 Verilog 硬件描述语言进行 VLSI 电路设计时的重要和关键设计步骤。设计实例的源代码都经过了验证,并且所有实例的源代码和测试平台都可以从出版社的网站下载。

读者对象

本书适用于学习高级数字系统设计课程的学生,以及那些想通过实例学习 Verilog 的现代集成电路设计专业工程师。本书适合电子工程、计算机工程和计算机科学等专业的高年级本科生和低年级研究生,也适合学习过逻辑设计入门课程的专业工程师使用。本书假定读者具有布尔代数及其在逻辑电路设计中应用的背景知识,并熟悉同步时序有限状态机。在此基础上,本书讨论了一些应用于计算机系统、数字信号处理、图像处理、跨时钟域的数据传输、内建自测试(BIST)和一些其他应用的重要电路的设计实例。这些实例涵盖了建模、架构的设计折中、流水线技术、多处理器执行、功能验证、定时分析、测试生成、故障模拟、可测性分析、逻辑综合和综合后验证的关键设计问题。

本版的新颖之处

- 探索了 Verilog 2001 和 2005 的主要特点
- 阐述并推广基于 Verilog 2001 和 2005 且可综合的寄存器传输级(RTL)描述和算法建模的设计风格
- 深入讨论基于 Verilog 2001 和 2005 的数字处理系统(如图像处理器、数字滤波器和环形缓冲器)算法和架构
- 给出了基于 Verilog 2001 和 2005 语言的综合设计实例(如 RISC 计算机和各种数据通道控制器)
- 提供了大量有评注和解释的仿真结果的图形化描述
- 给出了 150 多个经过完全验证的基于 Verilog 2001 和 2005 的设计实例
- 含有利用 Verilog 2001 和 2005 编写的具备 JTAG 和 BIST 可测功能的实用设计案例

- 附录中给出了 Verilog 2001 和 2005 HDL 的语法形式
- 讨论了异步和同步 FIFO 设计

本书特色

- 简要回顾了组合时序逻辑设计的基本原则
- 重点讨论现代数字设计方法
- 说明了行为级建模中 ASM 和 ASMD 图的作用
- 明确指出了可综合和不可综合循环的区别
- 通过实例对时序分析、故障模拟、测试和可测性设计进行切合实际的讨论
- 每章后均设计了一些涉及面广且难度高的习题^①

课程讲授次序

本书首先对组合逻辑设计进行简要介绍和回顾,接着描述了一个 ASIC 或 FPGA 的设计流程。按照书中内容的顺序,第 1 章至第 6 章利用综合的方法来研究设计了一些题目和内容。但是,阅读第 7 章至第 10 章时,则不必按照书中的顺序。课后作业具有挑战性,而且基于 FPGA 的实验练习适于同步实验或学期末的课题。第 10 章列出了一些算术运算的架构,覆盖了较多的应用范围。第 11 章介绍了后综合设计验证、时序分析、故障模拟和可测试性设计。根据课程教学的深度和重点,本章涵盖的内容和范围也可省略。

说明

我们没有坚持常规使用大写和小写字体,或使用代码清单专用字体。本书的选择一直是基于最大化的整体视觉效果及所列代码的可读性。我们认为,设计实例中的代码得到正确表达才是至关重要的。模块框图已被简化,以减少视觉混乱。所以,我们通常只显示信号的实际外部名称,而省略其形式化的内在对应名称。由于 D 触发器在现代 EDA 工具的综合中起着主导作用,因此书中几乎唯一性地使用了 D 触发器。

各章概述

第 1 章简要论述了硬件描述语言在基于库单元的 ASIC 和 FPGA 设计流程中的作用。第 2 章和第 3 章则根据传统的教学方式(例如卡诺图算法),回顾了数字设计先修课程中涉及的主要知识。这些资料可以奠定读者的数字设计的背景知识,便于之后利用实例介绍许多基于硬件描述语言的数字设计方法。第 4 章和第 5 章介绍了组合电路与时序电路的 Verilog 语言建模方法,重点强调了行为级建模中的代码编写风格。第 6 章着重基于库单元的 ASIC 综合,介绍了组合逻辑与时序逻辑的综合。这一章追求两个主要目标:(1) 提出可综合的代码描述风格;(2) 建立能够让读者预测综合结果的基础知识和能力。尤其在对时序状态机综合时,通常会把时序状态机分成数据通道和控制通道两部分来编写。第 7 章介绍了一些例子,这些例子描述了怎样设计一个数据通道的控制器,包括带有从数据通道反馈给控制器的状态信号的状态机设计。而 RISC CPU 设计和通用异步收发器(UART,用于系统间传输数据的电路)的设计作为这个例子的应用平台。第 8 章讲述了可编程逻辑器件(PLD)、复杂 PLD、只读存储器(ROM)和静态随机存储器(SRAM)的知识,并将综合目标扩展为 FPGA 的综合。第 9 章主要涉及计算机结构、数字滤波器和其他信

^① 相关教辅的申请(PPT,习题解答)请与电子工业出版社联系,联系电话:010-88254555,E-mail:Te_service@phei.com.cn。

号处理器中有关计算单元和算法的建模和综合。第 10 章研究并描述了数字状态机中计算单元的算法和结构。第 11 章使用 Verilog 语言, 结合故障仿真器和时序分析器, 重新审查了之前设计的状态机选择方案, 并考虑性能、时序问题及可测性问题, 来优化和完善这个主要取决于设计者的设计流程和任务。本章建模的测试访问端口 (TAP) 控制器由 IEEE 1149.1 标准定义 (即俗称的 JTAG 标准), 并提出了其应用实例。另外还给出了一个内建自测试 (BIST) 的详细实例。

致谢

本书作者非常感谢曾为本书做出贡献并提出宝贵意见的同事和学生们的支持。本书是我的研发经验和在科罗拉多大学教学经历的综合成果, 也包括我在惠普、福特微电子公司和 Prisma 公司的工作经历, 在荷兰的 Delft 技术大学的教学经验, 以及在欧洲和亚洲的短期课程的教学经验。虽然其中有的公司如今已成回忆, 但是我仍然深深感谢这些公司和科罗拉多大学对我进行 VLSI 电路设计研究工作的支持。本书手稿的第一版审稿人也提出了鼓励、关键内容的调整与许多有益的建议。我非常感谢 Jim Tracey 博士和 Rodger Ziemer 博士, 他们支持并肯定了我 VLSI 电路设计方面的努力和成就, 我也十分感谢福特微电子公司的 Deepak Goel 先生, 他向我介绍了后来成为最先进的 VLSI 设计平台的福特微电子的 Daisy 工作站。感谢 Simucad 公司的 Bill Fuchs 先生, 他帮助我获取了工业级的 Verilog 仿真器。感谢惠普公司的 Tom Saponas 和 Dave Ritchey 先生, 他们给我机会领导完成一个动态时序分析器的反向设计工程, 两名学生 David Uranek 和 Jerry Barnett 参与并获成功。十分感谢我在 Prisma 公司暑期工作的主管 Dave Still 先生, 提供了设计环境与精神鼓励, 使我完成了高性能多核系统中建模的难题。感谢 Sutherland HDL 的 Stu Sutherland 帮助我理解并更深入研究了数字系统建模中的竞争条件问题, 这些观点使我坚持使用非阻塞赋值来进行边缘敏感的行为级建模和使用阻塞赋值进行电平敏感的行为级建模的描述风格, 让我更好地帮助学生理解同步数字系统的操作和设计。谢谢我的朋友兼同行、瑞士联邦理工学院的 Hubert Kaeslin 博士, 与他进行的有意义的讨论让我能更深入钻研数字处理器的算法和结构。感谢 Kirk Sprague 和 Scott Kukel 帮助研发了一个可用于 UART 的汉明编码器。同时感谢 Cris Hagan, 他的论文提供了本书第 9 章的数字信号处理器中的抽取器和其他功能单元建模。非常感谢 Rex Anderson 先生帮助校正了几章的内容, 并对第一版进行了修改。谢谢我的学生 Terry Hansen 和 Lisa Horton, 他们提供了咖啡自动贩卖机例子的灵感, 并开发了支持 RISC CPU 的汇编代码。同时我还要感谢科罗拉多大学的 Greg Tumbush 教授和 Temple 大学的 Chen-Huan Chiang 教授为本书第二版提供了重要建议, 也谢谢许多学生的课堂讨论, 他们的发言为第二版提供了帮助。谢谢 Scott Disanno 和 Irwin Zucker 领导了第二版的出版, 谢谢 Haseen Khan 精心策划本书的结构。我向所有给予本书支持的朋友表示衷心的感谢!

目 录

第 1 章 数字设计方法概论	1
1.1 设计方法简介	1
1.1.1 设计规格	3
1.1.2 设计划分	3
1.1.3 设计输入	3
1.1.4 仿真与功能验证	4
1.1.5 设计整合与验证	5
1.1.6 预综合完成	5
1.1.7 门级综合与工艺映射	5
1.1.8 后综合设计确认	5
1.1.9 后综合时序验证	6
1.1.10 测试生成与故障模拟	6
1.1.11 布局与布线	6
1.1.12 物理和电气设计规则检查	6
1.1.13 提取寄生参量	7
1.1.14 设计完成	7
1.2 IC 工艺选择	7
1.3 后续内容概览	8
参考文献	8
第 2 章 组合逻辑设计回顾	10
2.1 组合逻辑与布尔代数	10
2.1.1 ASIC 库单元	10
2.1.2 布尔代数	12
2.1.3 狄摩根定律	13
2.2 布尔代数化简定理	14
2.3 组合逻辑的表示	15
2.3.1 积之和表示法	16
2.3.2 和之积表示法	17
2.4 布尔表达式的化简	18
2.4.1 异或表达式的化简	23
2.4.2 卡诺图(积之和形式)	23
2.4.3 卡诺图(和之积形式)	25
2.4.4 卡诺图与任意项	25
2.4.5 扩展的卡诺图	26
2.5 毛刺与冒险	27
2.5.1 静态冒险的消除(积之和形式)	28
2.5.2 消除两级电路静态冒险的小结	30

2.5.3	多级电路中的静态冒险	30
2.5.4	消除多级电路静态冒险的小结	32
2.5.5	动态冒险	32
2.6	逻辑设计模块	34
2.6.1	与非-或非结构	34
2.6.2	多路复用器	37
2.6.3	多路解复用器	38
2.6.4	编码器	38
2.6.5	优先编码器	39
2.6.6	译码器	40
2.6.7	优先译码器	41
	参考文献	41
	习题	41
第3章	时序逻辑设计基础	43
3.1	存储元件	43
3.1.1	锁存器	43
3.1.2	透明锁存器	44
3.2	触发器	45
3.2.1	D 触发器	45
3.2.2	主从触发器	46
3.2.3	J-K 触发器	48
3.2.4	T 触发器	48
3.3	总线与三态器件	49
3.4	时序机设计	50
3.5	状态转移图	52
3.6	设计举例: BCD 码到余 3 码的转换器	53
3.7	数据传输的串行线码转换器	57
3.7.1	设计举例: 用 Mealy 型 FSM 实现串行线性码转换	58
3.7.2	设计举例: 用 Moore 型 FSM 实现串行线码转换	60
3.8	状态化简与等价状态	61
	参考文献	63
	习题	64
第4章	Verilog 逻辑设计介绍	65
4.1	组合逻辑的结构化模型	65
4.1.1	Verilog 原语和设计封装	66
4.1.2	Verilog 结构化模型	67
4.1.3	模块端口	68
4.1.4	语言规则	68
4.1.5	自顶向下的设计和模块嵌套	69
4.1.6	设计层次和源代码结构	71
4.1.7	Verilog 矢量	71
4.1.8	结构化连接	72

4.2	逻辑系统设计验证及测试方法	75
4.2.1	Verilog 中的四值逻辑和信号解析	75
4.2.2	测试方法	75
4.2.3	测试平台的信号发生器	77
4.2.4	事件驱动仿真	78
4.2.5	测试模板	79
4.2.6	定长数	79
4.3	传播延时	80
4.3.1	惯性延时	81
4.3.2	传输延时	82
4.4	组合与时序逻辑的 Verilog 真值表模型	82
	参考文献	87
	习题	87
第 5 章	用组合与时序逻辑的行为级模型进行逻辑设计	89
5.1	行为建模	89
5.2	行为级建模的数据类型的简要介绍	90
5.3	基于布尔方程的组合逻辑行为级模型	90
5.4	传播延时与连续赋值	92
5.5	Verilog 中的锁存器和电平敏感电路	93
5.6	触发器和锁存器的周期性行为模型	94
5.7	周期性行为和边沿检测	95
5.8	行为建模方式的比较	96
5.8.1	连续赋值模型	96
5.8.2	数据流/寄存器传输级模型	97
5.8.3	基于算法的模型	99
5.8.4	端口名称: 风格问题	100
5.8.5	用行为级模型仿真	100
5.9	多路复用器、编码器和译码器的行为模型	101
5.10	线性反馈移位寄存器的数据流模型	106
5.11	用循环算法的数字机模型	107
5.11.1	IP(知识产权)的复用和参数化模型	110
5.11.2	时钟发生器	111
5.12	多循环操作状态机	112
5.13	设计文件中的函数和任务: 是精明还是愚蠢?	113
5.13.1	任务	113
5.13.2	函数	114
5.14	行为建模的算法状态机图	116
5.15	ASMD 图	117
5.16	计数器、移位寄存器和寄存器组的行为级模型	120
5.16.1	计数器	120
5.16.2	移位寄存器	124
5.16.3	寄存器组和寄存器(存储器)阵列	127

5.17	用于异步信号的去抖动开关、亚稳定性和同步装置	129
5.18	设计实例：键盘扫描器和编码器	133
	参考文献	138
	习题	139
第6章	组合逻辑与时序逻辑的综合	144
6.1	综合简介	144
6.1.1	逻辑综合	145
6.1.2	RTL 综合	150
6.1.3	高级综合	150
6.2	组合逻辑的综合	151
6.2.1	优先级结构的综合	153
6.2.2	利用逻辑无关紧要条件	154
6.2.3	ASIC 单元与资源共享	157
6.3	带锁存器的时序逻辑综合	158
6.3.1	锁存器的无意综合	159
6.3.2	锁存器的有意综合	162
6.4	三态器件和总线接口的综合	164
6.5	带有触发器的时序逻辑综合	165
6.6	显式状态机的综合	168
6.6.1	BCD 码/余 3 码转换器的综合	168
6.6.2	设计举例：Mealy 型 NRZ 码/Manchester 线性码转换器的综合	171
6.6.3	设计举例：Moore 型 NRZ 码/Manchester 线性码转换器的综合	172
6.6.4	设计举例：序列检测器的综合	174
6.7	寄存器逻辑	181
6.8	状态编码	185
6.9	隐式状态机、寄存器和计数器的综合	187
6.9.1	隐式状态机	187
6.9.2	计数器综合	187
6.9.3	寄存器综合	189
6.10	复位	192
6.11	门控时钟与时钟使能的综合	194
6.12	预测综合结果	195
6.12.1	数据类型综合	195
6.12.2	运算符分组	195
6.12.3	表达式替代	196
6.13	循环的综合	198
6.13.1	不带内嵌定时控制的静态循环	198
6.13.2	带内嵌定时控制的静态循环	200
6.13.3	不带内嵌定时控制的非静态循环	202
6.13.4	带内嵌定时控制的非静态循环	203
6.13.5	用状态机替代不可综合的循环	205
6.14	要避免的设计陷阱	209

6.15	分割与合并: 设计划分	209
	参考文献	210
	习题	211
第7章	数据通路控制器的设计与综合	216
7.1	时序状态机的划分	216
7.2	设计实例: 二进制计数器	217
7.3	RISC 存储程序机的设计与综合	221
7.3.1	RISC SPM: 处理器	221
7.3.2	RISC SPM: ALU	222
7.3.3	RISC SPM: 控制器	222
7.3.4	RISC SPM: 指令集	223
7.3.5	RISC SPM: 控制器设计	224
7.3.6	RISC SPM: 程序执行	234
7.4	设计实例: UART	236
7.4.1	UART 的操作	236
7.4.2	UART 发送器	237
7.4.3	UART 接收器	246
	参考文献	255
	习题	255
第8章	可编程逻辑及存储器件	267
8.1	可编程逻辑器件	268
8.2	存储器件	268
8.2.1	只读存储器	268
8.2.2	可编程 ROM(PROM)	270
8.2.3	可擦除 ROM	271
8.2.4	基于 ROM 的组合逻辑实现	272
8.2.5	用于 ROM 的 Verilog 系统任务	272
8.2.6	ROM 的比较	274
8.2.7	基于 ROM 的状态机	274
8.2.8	闪存	276
8.2.9	静态随机存储器(SRAM)	276
8.2.10	铁电非易失性存储器	291
8.3	可编程逻辑阵列(PLA)	291
8.3.1	PLA 最小化	293
8.3.2	PLA 建模	295
8.4	可编程阵列逻辑(PAL)	297
8.5	PLD 的可编程性	298
8.6	复杂可编程逻辑器件	298
8.7	现场可编程门阵列	298
8.7.1	FPGA 在 ASIC 市场中的角色	299
8.7.2	FPGA 技术	300
8.7.3	Xilinx 公司 Virtex 系列 FPGA	301

8.8	片上系统(SoC)的嵌入式可编程 IP 核	302
8.9	基于 Verilog 的 FPGA 设计流程	302
8.10	FPGA 综合	303
	参考文献	305
	相关网站	305
	习题及基于 FPGA 的设计训练	305
第 9 章	数字处理器的算法和架构	330
9.1	算法、循环嵌套程序和数据流图	330
9.2	设计实例:半色调像素图像转换器	332
9.2.1	半色调像素图像转换器的原型设计	334
9.2.2	基于 NLP 的半色调像素图像转换器结构	337
9.2.3	半色调像素图像转换器的最小并行处理器结构	342
9.2.4	半色调像素图像转换器:设计权衡	353
9.2.5	带反馈数据流图的结构	353
9.3	数字滤波器和信号处理器	358
9.3.1	FIR 滤波器	360
9.3.2	数字滤波器设计过程	362
9.3.3	IIR 滤波器	364
9.4	构建信号处理器的基本运算单元模型	367
9.4.1	积分器(累加器)	367
9.4.2	微分器	369
9.4.3	抽样和插值滤波器	369
9.5	流水线结构	373
9.5.1	设计实例:流水线型加法器	375
9.5.2	设计实例:流水线型 FIR 滤波器	380
9.6	环形缓冲器	380
9.7	异步 FIFO——跨越时钟域的同步问题	383
9.7.1	简化异步 FIFO	384
9.7.2	异步 FIFO 的时钟同步	391
	参考文献	406
	习题	406
第 10 章	算术处理器架构	412
10.1	数的表示方法	412
10.1.1	负整数的原码表示	412
10.1.2	负整数的反码表示方法	413
10.1.3	正数和负数的补码表示方法	414
10.1.4	小数的表示	415
10.2	加减法功能单元	415
10.2.1	行波进位加法器	415
10.2.2	超前进位加法器	415
10.2.3	上溢出和下溢出	419
10.3	乘法运算功能单元	419

10.3.1	组合(并行)二进制乘法器	419
10.3.2	时序二进制乘法器	422
10.3.3	时序乘法器设计:层次化分解	423
10.3.4	基于 STG 的控制器设计	424
10.3.5	基于 STG 的高效二进制时序乘法器	428
10.3.6	基于 ASMD 的时序二进制乘法器	433
10.3.7	基于 ASMD 的高效二进制时序乘法器	437
10.3.8	基于 ASMD 数据通路和控制器设计的总结	441
10.3.9	精简寄存器时序乘法器	441
10.3.10	隐式状态机二进制乘法器	445
10.3.11	Booth 算法时序乘法器	454
10.3.12	比特对编码	464
10.4	有符号二进制数乘法	470
10.4.1	有符号数的乘积:被乘数为负,乘数为正	470
10.4.2	有符号数的乘积:被乘数为正,乘数为负	470
10.4.3	有符号数的乘积:被乘数、乘数均为负	472
10.5	小数乘法	472
10.5.1	有符号小数:被乘数、乘数均为正	473
10.5.2	有符号小数:被乘数为负,乘数为正	473
10.5.3	有符号小数:被乘数为正,乘数为负	474
10.5.4	有符号小数:被乘数、乘数均为负	474
10.6	除法功能单元	474
10.6.1	无符号二进制数的除法	475
10.6.2	无符号二进制数的高效除法	480
10.6.3	精简寄存器时序除法器	487
10.6.4	有符号二进制数(补码)的除法	491
10.6.5	带符号的计算	491
	参考文献	493
	习题	494
第 11 章	后综合设计任务	498
11.1	后综合设计验证	498
11.2	后综合时序验证	500
11.2.1	静态时序分析	502
11.2.2	时序规范	503
11.2.3	影响时序的因素	505
11.3	ASIC 中时序违约的消除	508
11.4	虚假路径	509
11.5	用于时序验证的系统任务	511
11.5.1	时序检查:建立时间条件	511
11.5.2	时序检查:保持时间约束	512
11.5.3	时序检查:建立时间和保持时间约束	512
11.5.4	时钟检查:脉冲宽度约束	513

11.5.5	时序检查: 信号偏移约束	513
11.5.6	时序检查: 时钟周期	513
11.5.7	时序检查: 恢复时间	514
11.6	故障模拟及制造测试	514
11.6.1	电路缺陷和故障	515
11.6.2	故障检测与测试	517
11.6.3	D 标记法	518
11.6.4	组合电路的自动测试模板生成	520
11.6.5	故障覆盖和缺陷级别	522
11.6.6	时序电路的测试生成	522
11.7	故障模拟	524
11.7.1	故障解析	525
11.7.2	串行故障模拟	525
11.7.3	并行故障模拟	526
11.7.4	并发性故障模拟	526
11.7.5	概率性故障模拟	526
11.8	JTAG 端口和可测性设计	526
11.8.1	边界扫描和 JTAG 端口	527
11.8.2	JTGA 操作模式	528
11.8.3	JTAG 寄存器	528
11.8.4	JTAG 指令	530
11.8.5	TAP 结构	531
11.8.6	TAP 控制器状态机	532
11.8.7	设计实例: JTAG 测试	535
11.8.8	设计实例: 内建自测试	553
	参考文献	564
	习题	565
附录 A	Verilog 原语	569
附录 B	Verilog 关键词	574
附录 C	Verilog 数据类型	575
附录 D	Verilog 运算符	580
附录 E	Verilog 语言形式化语法 (I)	587
附录 F	Verilog 语言形式化语法 (II)	588
附录 G	Verilog 语言的附加特性	608
附录 H	触发器和锁存器类型	615
附录 I	Verilog 2001, 2005	616
附录 J	编程语言接口	631
附录 K	相关网站	632
	中英文术语对照表	633

第 1 章 数字设计方法概论

电路设计的经典设计方法是依赖于电路原理图的人工设计方法，而现在的大规模复杂电路广泛采用基于计算机语言的现代设计方法。这种实践变革有几方面的原因，其中最重要的原因是没有任何一支设计工程师团队能够用人工方法有效、全面、正确地设计和管理含数百万门级的现代集成电路(IC)。但使用硬件描述语言(HDL)，工程师们能很容易地实现对大型复杂电路系统的设计和管理。即使小规模电路的设计也更多地依赖于基于语言的描述，因为工程师们必须快速设计生产出满足瞬息万变的市场需求的产品。

基于语言的设计易于移植且不依赖于工艺，设计团队也可以重用或修改以前的设计，以保持与更先进工艺的一致性。随着器件物理尺寸的缩小，电路密度的提高，基于原有 HDL 模型进行综合生成的电路同样具有更高的性能。

硬件描述语言也是将各种设计专利成果集成为知识产权核(IP)的一种方便而有效的工具和手段。通过使用这种通用设计语言的描述，电路模块可以根据需要单独或合并进行综合和测试，以缩短设计周期。有些仿真工具还支持基于多种语言的混合描述。

采用 HDL 最显著的优点在于：基于语言描述的电路及其优化可以自动地进行综合，而不用经历人工设计方法中那些费力的步骤(如用卡诺图化简逻辑函数)。

目前，基于 HDL 的综合方法是工业界普遍采用的主流设计方法。设计者可以通过构建一个软件原型或模型来验证其功能，然后利用综合工具自动对所设计的电路进行优化，并且可以生成针对某物理工艺技术的网表(netlist)。

HDL 和综合工具的应用使得工程师们更关注有关功能的设计，而不是具体的单个晶体管或逻辑门的设计；综合得到的电路可以实现预期的功能，并满足面积和/或性能的约束要求。无论是功能模型还是行为模型的 HDL 模型描述，都可综合出不同的结构，并可据此快速对设计进行评估和折中。

HDL 可作为多种设计工具的平台，包括：设计输入、设计验证、测试向量生成、故障分析和仿真、时序分析和/或验证、综合和原理图的自动生成等任务。HDL 这种宽范围的覆盖使得设计者的设计工作通过工具链路时，由于不再需要考虑设计描述在不同工具间的转换过程而大大提高了设计流程的工作效率。

Verilog^[1]和 VHDL^[2]两种语言受到工业界的广泛支持，这两种语言都成为了 IEEE(电气和电子工程师协会)标准，并都得到 ASIC(专用集成电路)和 FPGA(现场可编程门阵列)相关综合工具的支持。模拟电路设计语言，如 Spice^[3]，在验证电路的关键时序路径上扮演着重要角色。但由于这些模拟电路描述语言对大型设计来说需要大得惊人的计算量，而且也不支持抽象设计，使得它们在大规模电路设计应用中变得很不实际。混合语言(如 Verilog-A 语言^[4])用于设计兼有数字和模拟电路的混合信号系统。近几年还出现了 SystemC^[5]和 Superlog^[6]这样的系统级设计语言，它们能够支持比 Verilog 或 VHDL 语言更高抽象级别的设计。

1.1 设计方法简介

系统级设计 ASIC 和 FPGA 电路的目的是最大限度地确保设计正确，使设计没有致命缺陷并能够进行生产制造。设计者可以按照图 1.1 所示的设计流程进行电路设计。该流程中给出了数

字电路的设计、验证、综合和测试等几个主要步骤的次序。ASIC 设计流程包括了从设计规范和设计输入开始,直到芯片级布局布线以及时序收敛等几个设计进程。在设计中,当所有的信号通路都满足由接口电路、电路时序单元和系统时钟所产生的时序约束条件时,即达到时序收敛。虽然设计流看起来呈现线性关系,但实际上则不然。当发现设计出现错误,设计需求改变,或有不符设计性能要求及设计约束改变等情况发生时,有可能要重新进行设计流程中的多个步骤。例如,如果一个设计不能满足时序约束,就不得不重新进行布局布线设计,还有可能要对一些关键路径进行重新设计。

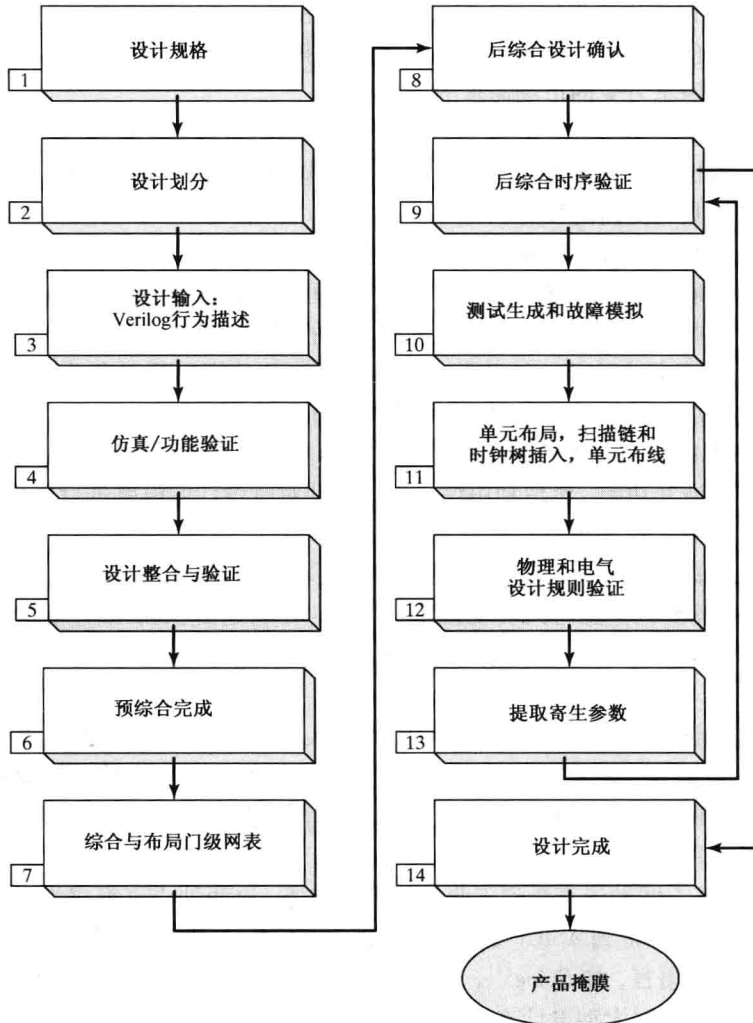


图 1.1 基于 HDL 的 ASIC 设计流程

由于 ASIC 的结构是非固定的,其设计所实现的电路性能取决于晶片上单元电路的物理布局与布线以及底层器件的特性等因素,因此基于标准单元的 ASIC 设计流程要比基于 FPGA 的设计流程更为复杂。在小于 $0.18\ \mu\text{m}$ 的亚微米工艺中,互连线延时对电路性能的影响起着关键的作用,其预布局布线的时序估计并不能确保满足布线路径设计的时序收敛要求。下面几节将详细阐述图 1.1 中描述的设计流程的各个步骤。