高等学校电子信息类“十三五”规划教材

西安电子科技大学研究生精品教材

Verilog HDL Advanced Digital Integrated Circuit Design

Verilog HDL 数字集成电路高级程序设计

蔡觉平　翁静纯　褚洁　冯必先　编著

西安电子科技大学出版社
http://www.xduph.com

高等学校电子信息类“十三五”规划教材
西安电子科技大学研究生精品教材

Verilog HDL 数字集成电路高级程序设计

蔡觉平　翁静纯　褚 洁　冯必先　编著

西安电子科技大学出版社

内 容 简 介

本书系统地对 Verilog HDL 程序设计方法进行说明，明确了数字可综合逻辑设计和测试仿真程序设计在 Verilog HDL 语言中的不同，通过对典型的组合逻辑电路、时序逻辑电路、混合电路和测试程序的设计举例，较为完整地说明了 Verilog HDL 语言在数字集成电路中的设计方法。

全书共分 10 章。第 1 章是 Verilog HDL 数字集成电路设计方法概述；第 2 章是 Verilog HDL 模块和结构化建模；第 3 章是 Verilog HDL 数据流描述和运算符；第 4 章是 Verilog HDL 行为级描述；第 5 章是 Verilog HDL 测试和仿真；第 6 章是 Verilog HDL 组合电路设计；第 7 章是 Verilog HDL 时序电路设计；第 8 章是 Verilog HDL 存储器设计；第 9 章是 Verilog HDL 设计风格；第 10 章是 Verilog HDL 高级程序设计。

学习本书需要具备数字电路和 Verilog HDL 基础知识。

本书可作为集成电路设计和 HDL 课程的研究生教材及本科生的辅导和设计参考教材，也可以作为数字集成电路设计工程师的参考书。

图书在版编目(CIP)数据

Verilog HDL **数字集成电路高级程序设计**/蔡觉平等编著. —西安：西安电子科技大学出版社，2015.10
高等学校电子信息类“十三五”规划教材
ISBN 978-7-5606-3858-4

Ⅰ. ① V… Ⅱ. ① 蔡… Ⅲ. ① 数字集成电路—电路设计—高等学校—教材 ② VHDL 语言—程序设计—高等学校—教材 Ⅳ. ① TN431.2 ② TP312

中国版本图书馆 CIP 数据核字(2015)第 219585 号

策　　划　李惠萍　戚文艳
责任编辑　李惠萍　宁晓蓉
出版发行　西安电子科技大学出版社(西安市太白南路 2 号)
电　　话　(029)88242885　88201467　　邮　　编　710071
网　　址　www.xduph.com　　电子邮箱　xdupfxb001@163.com
经　　销　新华书店
印刷单位　陕西天意印务有限责任公司
版　　次　2015 年 10 月第 1 版　　2015 年 10 月第 1 次印刷
开　　本　787 毫米×1092 毫米　1/16　　印　张　29.5
字　　数　706 千字
印　　数　1～3000 册
定　　价　53.00 元
ISBN 978-7-5606-3858-4 / TN
XDUP　4150001-1

前　言

随着集成电路技术的飞速发展，集成电路的制造工艺已经达到 14 nm 甚至更小尺寸，数字集成电路的规模越来越大，复杂度越来越高。为了提高设计的效率和可靠性，融合了电子技术、计算机技术和智能化技术的 EDA(Electronics Design Automation)工具已经在高速复杂数字集成电路设计中得到了广泛应用。

硬件描述语言(HDL)是现代专用集成电路(ASIC)EDA 设计的重要设计和仿真语言。目前，大部分数字集成电路设计者都在使用 HDL 创建高层次、结构化、基于语言的抽象电路描述，利用已有的设计技术综合出所需硬件电路，并对其进行功能验证和时序分析。

对于准备从事集成电路设计和 FPGA 设计的研究生和工程师来说，需要了解如何在设计流程的关键阶段正确使用 HDL，从而在综合后获得期望的电路。为此需要在了解 HDL 基本语法结构的基础上，深入理解电路的设计方法、综合特性和测试仿真方法。本书就是为这样一个目标而撰写的。

Verilog HDL 是被广泛采用的一种硬件描述语言，目前许多有关 Verilog HDL 的书籍重点关注的是讲解语言和语法，较少分析 Verilog HDL 语言和相应数字电路的关系，以及如何通过设计得到与目标相符合的电路系统。与这些书籍不同，本书着眼点主要放在 Verilog HDL 的设计方法上，这是编写本书的基本出发点。

本书主要根据 Verilog HDL 国际标准 IEEE 1364，对使用 HDL 进行数字集成电路设计、验证和综合的方法进行讲解和分析；对于基于 IP 的设计及方式、可综合代码风格、系统程序设计架构等高级程序设计方法也进行了规范化说明。通过 HDL 设计方法和大量的实用电路的设计，使读者能够对 Verilog HDL 数字集成电路设计技术有一个全面了解。

本书重点集中在如何在数字电路设计中的设计、综合和验证阶段合理使用 Verilog HDL。由于 Verilog HDL 本质上是对数字电路的一种描述方法，因此学习本书时需要深入了解数字电路设计基础知识，同时至少熟悉一种编程语言，这有助于通过阅读获取有用知识，并提高设计能力。本书通过典型的设计例程，讨论了 Verilog HDL 核心设计方法和验证方法，以便帮助读者快速掌握相关知识内容，并希望借助于这些典型例程，为读者在设计复杂电路时提供帮助。

数字电路中通常采用真值表、状态转移图和算法状态图对组合电路和时序电路进行分析和表示，在本书中将这些方法用于 Verilog HDL 的设计和分析，可以提高对设计方法的理解。同时，对于目前在信号处理、自动控制、数值计算等应用中所采用的一些设计方法，如查找表(LUT)、级数展开和有限状态机进行了说明和举例，希望能够帮助读者扩展设计思路。

目前数字集成电路普遍采用基于 IP 的设计方式，以提高集成电路的设计效率、规范设计方式、形成商业化的集成电路设计模式。本书对于集成电路和 FPGA 设计中 IP 的使用、综合和测试仿真进行了完整的讲解，通过学习可以初步掌握相关的设计方法和流程。

在 Verilog HDL 高级程序设计章节中，例举了一个完整的采用 BPSK 调制解调的无线通信系统设计，该方案已经用于 ZigBee 芯片中。通过该例程，可以帮助设计人员建立系统级设计的概念，有助于了解大规模集成电路的设计工作。

本书的另外一个特点是总结、归纳和分析了 HDL 设计代码风格和可综合电路的关系。通过典型例程及分析，初步建立程序设计代码风格的概念，对于实际设计过程中设计代码的编写和程序代码分析，会起到重要的作用。

本书列举大量实例的目的主要是希望读者在使用 Verilog HDL 进行超大规模集成(VLSI)电路设计时，学习如何应用关键步骤进行设计和验证。书中所列举的实例是完整的，并在 Modelsim 和 Synplify 软件中进行了编译、综合和仿真。

本书重点对于设计方法、测试方法和代码风格等进行讲解，对于 Verilog HDL 基本语法和不常用的概念未作介绍。本书适合作为集成电路设计和 HDL 课程的研究生教材，以及相应本科生的辅导和设计参考教材，对于希望通过实例学习 Verilog HDL，并将这种语言应用于集成电路设计和测试的专业工程师，也会起到一定的帮助。本书假定读者已具有布尔代数和数字逻辑设计等背景知识，并具有一定的数字电路设计经验。

全书共分 10 章。第 1 章是 Verilog HDL 数字集成电路设计方法概述；第 2 章是 Verilog HDL 模块和结构化建模；第 3 章是 Verilog HDL 数据流描述和运算符；第 4 章是 Verilog HDL 行为级描述；第 5 章是 Verilog HDL 测试和仿真；第 6 章是 Verilog HDL 组合电路设计；第 7 章是 Verilog HDL 时序电路设计；第 8 章是 Verilog HDL 存储器设计；第 9 章是 Verilog HDL 设计风格；第 10 章是 Verilog HDL 高级程序设计。

十分感谢对于本书的出版作出贡献的老师和学生们。感谢湘潭大学黄嵩人教授、西安交通大学张鸿教授、北京工业大学候立刚教授、西北工业大学张盛兵教授对本书提出的建设性意见。在本书中，蔡觉平完成了第 1 章和第 9 章的内容和程序验证，冯必先和褚洁完成了第 2～4 章的内容和程序验证，翁静纯完成了第 5 章的内容，国际留学生阮文长和王科完成了第 6～7 章内容，李娇完成了第 8 章的内容和程序验证，杨云锋完成了第 10 章的内容和程序验证。感谢马原、徐维佳、宋喆喆、同亚娜和温凯林在集成电路设计流程、代码质量评估等方面的大量实际工作。感谢课题组其他同学对于本书出版所作的努力。我们非常高兴能够与负责本书出版工作的西安电子科技大学出版社李惠萍编辑一起工作。她的支持和鼓励，以及对本书创作过程的指导，确保了本书的出版质量。

希望通过本书的出版，为致力于集成电路设计的同学和工程师提供帮助。

编　者

2015 年 8 月

目　录

第 1 章

Verilog HDL 数字集成电路设计方法概述

1.1　数字集成电路的发展和设计方法的演变

集成电路起步于 20 世纪 60 年代，随着数字集成电路和计算机技术的飞速发展，数字系统也得到了飞速发展。最早的数字电路由真空管和电子管构成，后来出现了以硅基半导体为主的集成电路。第一代集成电路是只有几十个逻辑门的小规模集成电路(Small Scale Integrated，SSI)。随着技术的发展，先后经历了中规模、大规模、超大规模集成电路，甚至发展到单芯片上有数千万个逻辑门的极大规模集成电路(Ultra Large Scale Integrated，ULSI)，如图 1.1-1 所示。集成电路的规模越来越大，集成密度越来越高，相应的设计也越来越复杂。芯片制造商生产的芯片上所集成的晶体管数量已达到了空前的水平。例如，NVIDIA 公司 2013 年发布的单芯显卡 GeForce GTX 780Ti 所搭载的 GK110-425-B1 芯片拥有 71 亿的晶体管规模；AMD 公司 2014 年生产的 Tonga 显卡芯片单芯片集成了超过 50 亿只晶体管；NVIDIA 公司 2015 年发布的单芯显卡 GeForce GTX Titan X 所采用的 GM200-400-A1 芯片晶体管数量达到 80 亿。集成电路产业的主流技术推进到了 22 nm 工艺，甚至 Intel 已经量产 14 nm 工艺，下一步先进技术还将导入到 10 nm 领域。

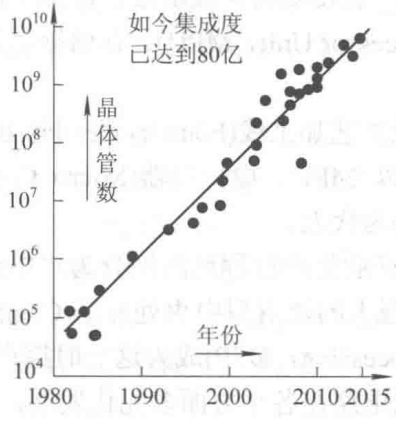

图 1.1-1　数字集成电路复杂度趋势

设计单元从起初的分立元件发展到 IP(Intellectual Property，知识产权)复用技术；系统

级别由早期的印刷版系统到片上系统(System on Chip，SoC)以及系统级封装(System in Package，SiP)；功能方面也从开始的简单布尔逻辑运算发展到可以每秒处理数十亿次计算的复杂运算。这一切都使得数字集成电路被广泛应用于计算机、通信、图像等多个领域。

随着半导体工艺尺寸逐步逼近硅工艺物理极限，单芯片集成晶体管数量超过了几十亿只。数字集成电路不断引入新技术以推动超大规模集成电路设计的发展，最关键的几项技术包括 PLD(Programmable Logic Device，可编程逻辑器件)技术、SoC 技术和 IP 复用技术。

集成电路工艺制造水平的提高和芯片集成度、复杂度的日益增加，使芯片的设计方法和设计技术发生了深刻的变化，如图 1.1-2 所示。早期的数字系统规模尚小，采用的是基于原理图的设计方法，即用一些固定功能的器件加上一定的外围电路搭成电路板，进一步构成电子系统。但是随着电路规模的不断增大，这种设计方法灵活性差、设计效率低的问题变得更加明显。

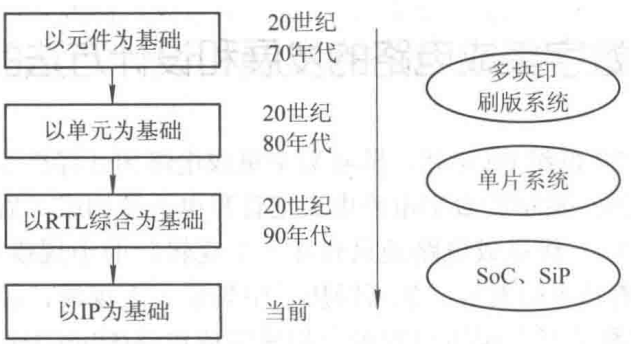

图 1.1-2 数字集成电路设计方法的演变

硬件描述语言(Hardware Description Language，HDL)的出现和发展逐渐改变了用传统的原理图设计电路的方法。HDL 以文本的形式描述硬件电路的功能、信号连接以及时序关系，相比原理图的描述方式，HDL 更便于保存管理，易于修改维护和设计重用，能够更高效灵活地实现大规模复杂数字系统的设计。以 HDL 语言为基础的 IP 复用技术的出现，使得功能模块得以重用，进一步提高了设计效率。

回顾多年来集成电路的发展，可将其分为三个阶段：

(1) 20 世纪 70 年代——IC 产业发展的初级阶段。以加工制造为主导，设计只作为附属部门，简单微处理器(Micro Processor Unit，MPU)、存储器以及标准通用逻辑电路是这一时期的标志性产物。

(2) 20 世纪 80 年代——标准工艺加工线(Foundry)公司与 IC 设计公司相互结合、共同发展的阶段。这一时期主流产品以 MPU、微控制器(Micro Control Unit，MCU)以及专用 IC(Application-Specific IC，ASIC)为代表。

(3) 20 世纪 90 年代——IC 产业生产过程逐渐细分为“电路设计、芯片制造、电路封装、电路测试”四大领域。功能强大的通用型中央处理器(Central Processing Unit，CPU)和数字信号处理器(Digital Signal Processing，DSP)成为这一时期产业发展的一个主要方面。

到 21 世纪的今天，数字集成电路在各个方面多元化发展：

(1) 芯片的市场需求方面，通用型芯片被具有特定功能的差异化专用芯片取代，以应对多媒体技术和移动通信等应用的飞速发展。

(2) 技术方面，原本单纯依靠提升频率的发展方式已经跟不上单芯片规模扩大的速度，这种情况下，采用大规模多内核处理器结构设计芯片成为新的主流模式。

(3) 设计方法方面，基于 IP 核的功能模块重用的设计方式，极大地提高了芯片的设计效率和可扩展性，有利于在商业化竞争中取得优势。

目前集成电路的规模越来越大，复杂度越来越高，芯片设计和制造成本不断提高，设计、测试和制造工艺中的环节增加。这些变化使电子设计任务的难度、复杂度和工作量都迅速加大，整个设计任务要求有很高的协调性和整体性。面对日益庞大的硬件规模，设计者需要从更高的抽象层次上进行设计，提高元件模型的可重用性，并且需要用更自动化、规范化和高效化的方式来描述系统，以解决超大规模集成电路发展所面临的一系列问题：

(1) 功能模块的可重用性。

(2) 综合，特别是高层次综合和数模混合电路模型的综合。

(3) 验证，如形式验证和仿真验证等。

(4) 数字电路的超深亚微米效应。

为了使复杂的芯片易于描述理解，很有必要用一种高级语言来表达其功能，隐藏具体实现细节。基于上述原因，HDL 应运而生。现在 HDL 不仅应用于数字集成电路设计阶段，在经过改进和发展后，也能很好地应用于设计的建模、仿真验证和综合等各个阶段。

Verilog HDL 作为一种常用的硬件描述语言，从一种专用语言发展成 IEEE 标准，不断地修正扩展，在其基础上发展出了模拟硬件描述语言 Verilog-A，为模拟集成电路的程序化设计提供了支持。这个扩展使得 Verilog HDL 可以对集成了模拟和混合信号的系统进行建模。之后为了将数字模型的建立和电路设计加以统一，在其基础上又诞生了 System Verilog，进一步提高了集成电路的设计效率。

1.2　Verilog HDL 的发展和国际标准

Verilog HDL 是目前设计界通常采用的一种硬件描述语言，被广泛地应用于数字 ASIC 和可编程逻辑器件的设计开发工作。Verilog HDL 按照一定的规则和风格编写代码，可以从系统级、电路级、门级到开关级等抽象层次，进行数字电路系统的建模、设计和验证工作。被建模的数字系统对象可以简单到一个门级电路，也可以复杂到一个功能完整的数字电子系统。

Verilog HDL 从设计开始到目前的广泛应用经历了 30 多年的发展历程，功能也由最初的数字集成电路设计发展到数字和模拟电路设计，如图 1.2-1 所示。Verilog HDL 已经成为数字电路和数字集成电路中广泛使用的设计语言。

Verilog HDL 语言诞生于 1983 年，最初是由 Gateway Design Automation 公司为其模拟器产品开发的硬件建模语言。1987 年，Synopsys 公司将 Verilog HDL 语言作为综合工具的输入，为在数字集成电路上的应用提供了 EDA 综合工具，更加高效地实现电路的描述性设计。

1989 年 GDA 公司被 Cadence 公司并购，Verilog HDL 语言成为 Cadence 公司的私有财产。1990 年 Cadence 公司正式公开发表 Verilog HDL 语言，以便于 Verilog HDL 大范围的

推广和使用。随后成立的 OVI(Open Verilog International)组织负责 Verilog HDL 语言的发展并制定有关标准。

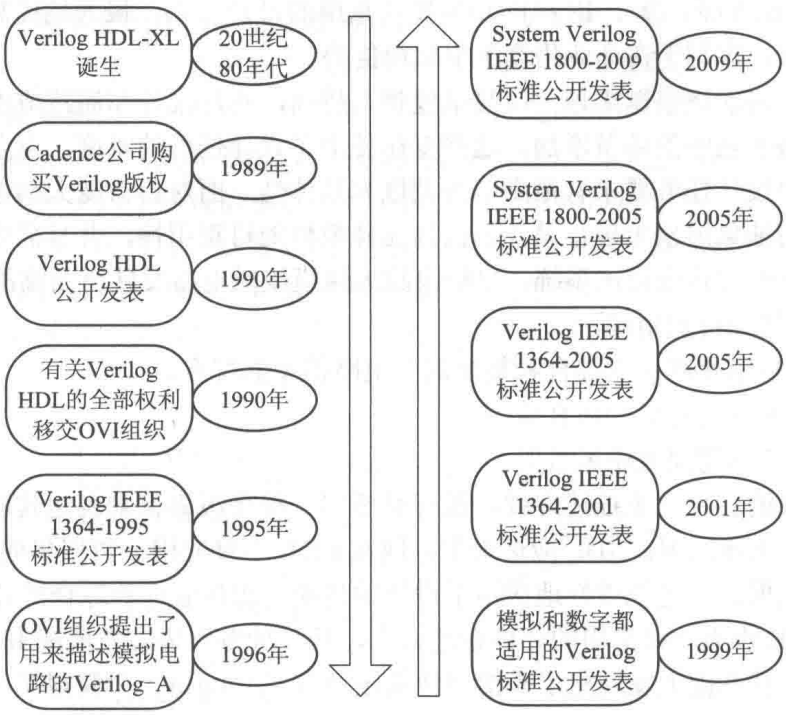

图 1.2-1 Verilog HDL 的发展历史

1992 年，OVI 开始致力于将 Verilog OVI 标准推广成为 IEEE 标准。1993 年，几乎所有 ASIC 厂商都开始支持 Verilog HDL，并且认为 Verilog HDL-XL 是最好的仿真器。同时，OVI 推出 2.0 版本的 Verilog HDL 规范，IEEE 则将 OVI 的 Verilog HDL2.0 作为 IEEE 标准的提案。从此，Verilog HDL 正式成为数字集成电路的设计语言标准，见表 1.2-1。

表 1.2-1 Verilog HDL 国际标准

名 称	时 间	备 注
Verilog IEEE 1364-1995	1995 年 12 月	基于 Verilog HDL 的优越性，IEEE 将 Verilog HDL 制定为标准，即 Verilog HDL 1364-1995
Verilog-A	1996 年	Verilog-A 是由 OVI 提出的一种基于 IEEE 1364 Verilog 规范的硬件描述语言，用于模拟电路行业的建模
	1999 年	模拟和数字都适用的 Verilog 标准公开发表
Verilog IEEE 1364-2001	2001 年	IEEE 制定了 Verilog IEEE 1364-2001 标准并公开发表；其中 HDL 部分相对于 1995 标准有较大增强，PLI 部分变化不大
Verilog IEEE 1364-2005	2005 年	对上一版本的细微修正。该版本还包括了一个相对独立的新部分，即 Verilog-AMS
SystemVerilog IEEE 1800-2005	2005 年	基于 Verilog HDL 语言，对 Verilog IEEE 1364-2001 标准的扩展增强，是新一代硬件设计和验证语言
SystemVerilog IEEE 1800-2009	2009 年	将 IEEE 1364-2005 和 IEEE 1800-2005 两个部分合并，成为一个新的、统一的 SystemVerilog

1995 年 12 月，IEEE 制定了第一个 Verilog HDL 语言标准 Verilog IEEE 1364-1995。在此基础上，2001 年 IEEE 增加了部分功能，并制定了较为完善的标准 Verilog IEEE 1364-2001。目前在数字集成电路方面主要采用的就是这两个标准所规定的程序语法和设计规范。2005 年，Verilog 再次进行了更新，即 Verilog IEEE 1364-2005 标准。该版本只是对上一版本的细微修正。这个版本还包括了一个相对独立的新部分，即 Verilog-AMS。这个扩展使得传统的 Verilog 可以对集成的模拟和混合信号系统进行建模。

由于 Verilog HDL 在数字集成电路设计上的优越性，众多程序开发人员希望其能在硬件设计领域得到更为广泛的应用和发展。

在模拟电路设计方面，基于 IEEE 1364 Verilog HDL 规范发展出了 Verilog-A，作为模拟电路行业的标准建模语言，更有效地实现了模拟集成电路的程序化设计。

在系统级设计方面，传统的设计方法将数字模型的建立和电路设计过程分割开来。数字模型的建立和分析是用 C 语言等高级软件语言来实现的，通过定点化设计将数学模型转变成电路模型，而电路设计部分是用 HDL 语言来实现的。这种方法无法将数字模型直接用于数字集成电路的设计中，导致设计所需的周期加长、耗时耗力，且存在重复性工作等问题。现有的语言无法解决这一问题，为此研究和开发人员迫切希望统一模型建立和电路设计过程，使集成电路设计更加灵活高效，这就给 EDA 工具厂商提出了新的要求。为了满足这一要求，一种新的工程语言 System Verilog 应运而生，这个统一的语言使得工程师可以建模大型复杂的设计，并且验证设计功能的正确性。

SystemVerilog 是一种硬件描述和验证语言(Hardware Description and Verification Language，HDVL)，由 Accellera 开发，它主要定位在芯片的实现和验证流程上，并为系统级的设计流程提供了强大的连接能力。System Verilog 是对 Verilog IEEE 1364-2001 的扩展，其由 Accellera 标准组织维护并提交标准化，在 2005 年 12 月被标准化为 IEEE 1800-2005。System Verilog 的扩展主要针对两个方面：

(1) 对硬件建模的扩展，主要整合了 SUPERLOG 和 C 语言的许多优秀特性。

(2) 对验证和断言方面的扩展，主要整合了 SUPERLOG、VERA C、C++ 及 VHDL 语言的特性，同时包括了 OVA 和 PSL 断言。

作为 Verilog HDL 的扩展，System Verilog 综合了一些已验证过的硬件设计和验证语言的特性，这些扩展使得 SystemVerilog 在一个更高的抽象层次上提高了在 RTL 级、系统级及结构级进行硬件建模的能力，以及验证模型功能的一系列丰富特征。虽然 System Verilog 是一个整合体，但它大大超越了各分立部分的总和。

2009 年，IEEE 1364-2005 和 IEEE 1800-2005 两个部分合并为 IEEE 1800-2009，成为了一个新的、统一的 SystemVerilog 硬件描述验证语言。

Verilog HDL 语言是完全独立于目标器件芯片物理结构的硬件描述语言，Verilog 中描述所有的硬件组件和测试平台的基本结构被称为模块，通过模块的相互连接调用来构建复杂的电路。使用 Verilog HDL 语言进行数字集成电路和系统的功能设计时，采用的是描述性的建模方式，通过数据流描述、行为描述和结构描述等方式，分模块、分层次地进行硬件组件的描述以便进行仿真、综合，规范编写用于指定测试数据和监控电路响应的测试平台，从而完成电路的设计和验证工作。同时，Verilog HDL 还提供编程语言接口，方便在模拟、验证期间通过该接口从设计外部访问设计，包括模拟具体控制和运行。

Verilog HDL 不仅定义了完善的语法规则，而且对每个语法结构都定义了清晰的模拟、仿真语义。它从 C 编程语言中继承了多种操作符和结构，提供了较强的扩展建模能力。它的使用大大简化了硬件设计的过程，提高了设计的效率和可靠性。设计者可以专注于其功能的实现，而不需要对不影响功能的与工艺有关的因素花费过多的时间和精力。完整的硬件描述语言足以对从最复杂的芯片到完整的电子系统进行描述。

本书主要针对 Verilog HDL 基本语法规则和数字集成电路设计进行讲述，以应对越来越复杂的数字集成电路芯片的设计和验证工作。

1.3 Verilog HDL 语言的设计思想和可综合特性

在数字集成电路设计过程中，设计者使用 Verilog HDL 硬件描述语言进行关键性步骤的开发和设计。其基本过程是首先使用 Verilog HDL 对硬件电路进行描述性设计，利用 EDA 综合工具将其综合成一个物理电路，然后进行功能验证、定时验证和故障覆盖验证。

与计算机软件所采用的高级程序语言(C 语言)类似，Verilog HDL 是一种高级程序设计语言，程序编写较简单，设计效率很高。然而，它们面向的对象和设计思想却完全不相同。

软件高级程序语句是对通用型处理器(如 CPU)的编程，主要是在固定硬件体系结构下的软件化程序设计。处理器的体系结构和功能决定了可以用于程序编程的固定指令集，设计人员的工作是调用这些指令，在固化的体系结构下实现特定的功能。

Verilog HDL 和 VHDL 等硬件描述语言是对电路的设计，将基本的最小数字电路单元(如门单元、寄存器、存储器等)通过连接方式构成具有特定功能的硬件电路。在数字集成电路中，这种最小的单元是工艺厂商提供的设计标准库或定制单元；在 FPGA 中，这种最小单元是芯片内部已经布局的基本逻辑单元。设计人员通过描述性语言调用和组合这些基本单元实现特定的功能，其基本的电路是灵活的。

Verilog HDL 给设计者提供了几种描述电路的方法。设计者可以使用结构性描述方式把逻辑单元互连在一起进行电路设计，也可以采用抽象性描述方式对大规模复杂电路进行设计，如对有限状态机、数字滤波器、总线和接口电路的描述等。

由于硬件电路的设计目标是最终产生的电路，因此 Verilog HDL 程序设计的正确性需要通对综合后电路的正确性来验证。逻辑上相同的电路在物理电路中的形式却有可能完全不同。对于 Verilog HDL 程序设计而言，数字电路的程序描述性设计具有一定的设计模式，这与 C 语言等高级软件程序设计是显著不同的。

例 1.3-1 是对模 256(8 bit)计数器的两种描述。程序(1)是通常的 Verilog HDL 语言对计数器的描述方式，通过改变计数器状态寄存器组的位宽和进位条件，可以实现不同计数器的硬件电路设计。程序(2)是初学者经常出现的一种错误描述方式，刚开始编写 Verilog HDL 程序时经常会套用 C 语言等高级程序设计的模式，这样往往得不到目标数字电路功能。

例 1.3-1 用 Verilog HDL 设计模 256(8 bit)计数器。

(1) 可综合程序描述方式。

```
module counter(clk, rst_n, cnt);
```

```
    input clk, rst_n;
    output [7:0] cnt;
    reg [7:0] cnt;
    always@(posedge clk or negedge rst_n)
        if(!rst_n) cnt<=8'b00000000;
        else cnt<=cnt+1'b1;
endmodule
```

(2) 常见的错误描述方式。

```
module counter(clk, rst_n, cnt);
    input clk, rst_n;
    output [7:0] cnt;
    reg [7:0] cnt;
    integer i;
    always@(posedge clk or negedge rst_n)
      begin
        if(!rst_n) cnt<=8'b00000000;
        else
          for(i=0; i<=255; i=i+1)
              cnt<=cnt+1'b1;
      end
endmodule
```

Verilog HDL 的电路描述方式具有多样性，这也决定了电路设计的多样性。例 1.3-2 是对一个多路选择器的设计，程序(1)采用的是真值表的形式；程序(2)采用的是逻辑表达式的形式；程序(3)采用的是基本逻辑单元的结构性描述形式。

例 1.3-2　用 Verilog HDL 设计数字多路选择器。

(1) 采用真值表形式的代码。

```
module MUX (data, sel, out);
    input [3:0] data;
    input [1:0] sel;
    output out;
    reg out;
    always @(data or sel)
      case (sel)
        2'b00 : out=data[0];
        2'b01 : out=data[1];
        2'b10 : out=data[2];
        2'b11 : out=data[3];
      endcase
endmodule
```

(2) 采用逻辑表达式形式的代码。

```
module MUX (data, sel, out);
    input [3:0] data;
    input [1:0] sel;
    output out;
    wire w1, w2, w3, w4;
    assign w1=(~sel[1])&(~sel[0])&data[0];
    assign w2=(~sel[1])&sel[0]&data[1];
    assign w3=sel[1]&(~sel[0])&data[2];
    assign w4=sel[1]&sel[0]&data[3];
    assign out=w1|w2|w3|w4;
endmodule
```

(3) 采用结构性描述的代码。

```
module MUX (data, sel, out);
    input [3:0] data;
    input [1:0] sel;
    output out;
    wire w1, w2, w3, w4;
    not  U1(w1, sel[1]);
         U2(w2, sel[0]);
    and  U3(w3, w1, w2, data[0]);
         U4(w4, w1, sel[0], data[1]);
         U5(w5, sel[1], w2, data[2]);
         U6(w6, sel[1], sel[0], data[3]);
    or   U7(out, w3, w4, w5, w6);
endmodule
```

Verilog HDL 语言主要用于电路设计和验证，部分语言是为电路的测试和仿真制定的，因此其语言分为用于电路设计的可综合性语言和用于测试仿真的不可综合性语言。对于可综合性语言，EDA 综合工具可以将其综合为物理电路。而对于部分语言，EDA 工具综合性很差，设计人员往往得不到与设计思想相符合的物理电路。

正是由于 Verilog HDL 语言的特殊性，初学者往往很难把握可综合电路的设计方法，得不到最终期望的电路，这也是掌握 Verilog HDL 设计方法所面临的一个困难。为了解决这一问题，降低 Verilog HDL 的设计门槛，EDA 工具厂商正努力设计综合性工具，使之能够适应 C 语言程序设计思想，但这需要一个很长的过程。

在现阶段，作为设计人员熟练掌握 Verilog HDL 程序设计的多样性和可综合性，是至关重要的。作为数字集成电路的基础，基本数字逻辑电路设计是进行复杂电路设计的前提。本章通过对数字电路中基本逻辑电路的 Verilog HDL 程序设计进行讲述，使读者掌握基本逻辑电路的可综合性设计，为具有特定功能的复杂电路的设计打下基础。

逻辑电路可以分成两大类：一类是组合逻辑电路，简称组合电路；另一类是时序逻辑