

## <<Verilog HDL数字系统设计与验证>>

### 图书基本信息

书名：<<Verilog HDL数字系统设计与验证>>

13位ISBN编号：9787121082924

10位ISBN编号：7121082926

出版时间：2009-4

出版时间：乔庐峰、王志功 电子工业出版社 (2009-04出版)

作者：乔庐峰 著

页数：255

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## <<Verilog HDL数字系统设计与验证>>

### 前言

Verilog HDL通常也直接称其为Vefilog，是一种通用的硬件描述语言。

使用VefilogHDL可以对电子电路和系统的行为进行描述。

基于这种描述，结合相关的EDA软件工具，可以进行电路的设计和仿真，并最终得到所期望的实际的电路与系统。

VefilogHDL出现于20世纪80年代初，由于其使用的方便性和实用性而逐渐被众多设计者所接受，影响力不断扩大，成为工业界的设计标准。

美国电气和电子工程师协会（IEEE：Insfitutc ofElectrical and Electronics Engineers）于1995年12月制定了Verilog的国际标准IEEE 1364-1995。

此后，IEEE在2001年又发布了更为完善和丰富的IEEE 1364-2001标准。

这两个标准的发布极大地推动了Verilog在全球的发展，使之成为进行集成电路设计和可编程逻辑器件设计的应用最广泛的语言之一。

目前，使用VefilogHDL（或另一种硬件描述语言VHDL）进行电路设计是从事相关领域科研和开发工作的工程师、设计师必须具备的基本能力。

笔者长期工作在教学、科研一线，使用Vefilog设计了很多具有一定规模和复杂度的电路。

在学习和使用Verilog过程中，也遇到过种种难题和困惑，这些难题和困惑随着专业知识的积累和广泛的交流学习，有的得到了解决，有的理清了思路。

此后，在一些教学和讲座活动中，笔者将这些经验匀很多人共同分享，不少学员都觉得颇有受益并建议在现有讲义的基础上编写一本相关的教材，于是便有厂这本书。

VerilogHDL语法内容比较庞杂，可以在算法级、RTL级、门级和晶体管级建立电路模型和进行电路仿真。

本书从电路设计与验证的实际需要出发，重点介绍的是可用于电路综合、实现的RTL级语法和与电路仿真、验证密切相关的语法要点。

在进行语法学习时，本书力求以简洁清晰的方式对语法要点进行文字说明，重点是提供典型例题加以辅助分析。

书中的多数例题都具有典型性和代表性，部分例题直接源于工程设计实践，具有一定的参考价值。

全书共有16章，分成了两个组成部分，其中第一部分讨论语法基础与基本电路单元设计，第二部分讨论系统设计与验证。

最后给出了5个附录。

## <<Verilog HDL数字系统设计与验证>>

### 内容概要

全面介绍如何使用Verilog HDL进行数字电路设计、仿真和验证。

全书共分为VerilogHDL语法基础与基本电路单元设计、系统设计与验证和附录三个组成部分。

《VerilogHDL数字系统设计与验证》以Verilog-1995和Verilog-2001标准为基础，重视电路仿真与验证，紧密结合设计实践，可以帮助读者掌握规范的电路设计方法。

书中大量的例题可直接用于读者的设计实践，具有良好的参考价值。

《VerilogHDL数字系统设计与验证》适合通信工程、电子工程及相关专业的高年级本科生、硕士生作为教材使用，同时也可供进行集成电路设计和可编程逻辑器件设计的工程师参考使用。

## 书籍目录

第一部分 语法基础与基本电路单元设计 第1章 引言1.1 VerilogHDL语言的产生与发展1.2 设计流程1.3 VerilogHDL在电路仿真中的应用1.4 VerilogHDL在电路综合中的应用思考与练习第2章 Verilog代码结构2.1 模块的结构2.2 电路功能描述方式思考与练习第3章 Verilog中的常量、变量与数据类型3.1 常量3.2 变量3.3 块语句与变量的赋值思考与练习第4章 操作符 / 运算符4.1 算术操作符4.2 关系操作符4.3 相等关系操作符4.4 逻辑操作符4.5 按位操作符4.6 缩位(归约)操作符4.7 移位操作符4.8 条件操作符4.9 并位(位拼接)操作符4.10操作符的优先级思考与练习第5章 条件语句与循环语句5.1 if-else语句5.1.1 if-else语句的语法结构5.1.2 if-else语句与锁存器5.2 case、casez和casex语句5.2.1 case语句5.2.2 casez和casex语句5.2.3 case语句与锁存器5.3 循环语句5.3.1 forever循环语句5.3.2 repeat循环语句5.3.3 while循环语句5.3.4 for循环语句思考与练习第6章 任务与函数6.1 任务6.1.1 任务定义6.1.2 任务调用6.1.3 任务定义与调用举例6.2 函数6.2.1 函数的定义6.2.2 函数的调用6.2.3 函数定义与调用举例6.3 任务与函数的异同小结思考与练习第7章 用户定义的原语7.1 UDP的定义7.2 组合电路UDP7.3 时序电路UDP第8章 状态机8.1 引言8.2 设计风格18.3 设计风格28.4 设计风格38.5 状态机编码方式：二进制编码和独热编码思考与练习第9章 系统任务与编译预处理9.1 与仿真相关的系统任务9.1.1 \$display和\$write9.1.2 \$monitor和\$strobe9.1.3 \$time和\$realtime9.1.4 \$finish和\$stop9.1.5 \$readmemh和\$readmemb9.1.6 \$random9.2 与波形和定时检查相关的系统任务9.3 编译预处理语句9.3.1 宏定义define9.3.2 文件包含处理9.3.3 仿真时间标度timescale9.4 条件编译命令思考与练习第10章 常用基本电路单元设计10.1 Verilog代码的综合10.2 算术逻辑单元10.3 并 / 串变换电路10.4 简单自动售货机控制电路10.5 7段数码显示器控制电路10.6 逐级进位和超前进位加法器10.6.1 逐级进位加法器实现方法10.6.2 超前进位加法器10.7 同步FIFO的设计思考与练习第二部分 系统设计与验证第11章 静态定时分析、时钟域与同步化设计11.1 前仿真与后仿真11.2 静态定时分析11.2.1 静态定时分析与门延迟11.2.2 时钟抖动对静态定时分析的影响11.2.3 时钟偏移对静态定时分析的影响11.3 时钟域与同步化设计11.3.1 同步器结构11.3.2 时钟域的划分11.3.3 单一跨时钟域信号的有效传递11.3.4 多个跨时钟域信号的有效传递11.4 采用异步FIFO进行时钟域隔离11.4.1 异步FIFO的电路结构11.4.2 格雷码计数器11.4.3 AFIFO的设计与应用11.5 通过高速采样实现异步信号的同步化设计思考与练习第12章 Verilog设计验证技术12.1 电路验证的基本概念12.2 验证的全面性与代码覆盖率分析12.3 随机化测试12.4 定时验证12.5 自动测试testbench12.5.1 以太网桥接器的工作原理12.5.2 电路的模块级验证12.5.3 电路的系统级验证思考与练习第13章 典型复杂电路设计与分析13.1 乘法器13.1.1 串-并型乘法器13.1.2 并行乘法器13.1.3 使用“\*”实现乘法器13.2 除法器13.2.1 除法电路的算法13.2.2 VerilogHDL除法器的实现13.3 数字滤波器13.4 检错码编码电路思考与练习第14章 通信系统中的异步复用电路14.1 同步复用电路14.2 异步复用电路14.2.1 异步复用的基本概念14.2.2 正码速调整14.2.3 全同步设计方法第15章 通用异步收发器的设计与验证15.1 通用异步收发器规范15.2 电路结构设计15.3 UART控制电路模块代码设计与分析15.4 UART发送电路模块代码设计与仿真分析15.5 UART接收电路模块代码设计与仿真分析15.6 系统仿真15.7 UART自动测试testbench第16章 Viterbi译码器电路16.1 卷积码编码器的工作原理16.2 Viterbi译码器的工作原理16.2.1 分支度量单元的设计16.2.2 ACS单元的设计16.2.3 幸存路径信息存储和回溯单元的设计16.3 Viterbi译码器电路实现附录A 可编程逻辑器件附录B ModelSimSE使用指南附录C XilinxISE+ModelSim使用指南附录D AlteraQuartusII+SynplifyPro+ModelSim使用指南附录E Verilog(IEEEStd-1364-1995)关键字参考文献

## &lt;&lt;Verilog HDL数字系统设计与验证&gt;&gt;

## 章节摘录

插图：第1章引言1.1 Verilog HDL语言的产生与发展Verilog HDL ( Verilog HDL : Verilog Hardware Description Language ) 是一种硬件描述语言，可以对电子电路和系统的行为进行描述。

基于这种描述，结合相关的软件工具，可以得到所期望的实际的电路与系统。

Verilog HDL从20世纪80年代初由GDA ( Gateway Design Automation ) 公司最早推出，到现在被全球范围内的众多设计者所接受，已经经历了20多年的时间。

Verilog HDL ( 经常又称为Verilog ) 最初是GDA公司为其数字逻辑仿真器产品配套开发的硬件描述语言，用于建立硬件电路的模型。

那时它只是一种专用语言，但随着这种仿真器产品及其后续版本Verilog-XL的出现和广泛应用，Verilog也因其使用的方便性和实用性而逐渐被众多设计者所接受，影响力不断扩大。

1987年，著名的电子设计自动化 ( EDA : Electronic Design Automation ) 厂商Synopsys公司开始使用Verilog语言作为其综合工具的标准输入语言。

1989年，另一个著名的EDA厂商Cadence公司收购了GDA公司，然后把Verilog HDL进行了公开发布。随后，一个名为OVI ( Open Verilog HDL International ) 的组织成立了，专门负责Verilog的发展和标准化推动工作。

到了1993年，几乎所有专用集成电路设计厂商都开始支持Verilog，并且认为Verilog-XL是最好的电路仿真软件。

同时，OVI推出2.0版本的Verilog规范。

美国电气和电子工程师协会 ( IEEE : Institute of Electrical and Electronics Engineers ) 接受了将OVI的Verilog HDL 2.0作为IEEE标准的提案，并于1995年12月制定了Verilog的国际标准IEEE 1364.1995。

此后，IEEE在2001年又发布了更为完善和丰富的IEEE 1364-2001标准。

这两个标准的发布极大地推动了Verilog在全球的发展。

Verilog语言被广泛使用的基本原因在于它是一种标准语言，与设计工具和实现工艺无关，从而可以方便地进行移植和重用。

Verilog语言的两个最直接的应用领域是可编程逻辑器件和专用集成电路 ( ASIC : Application Specific Integrated Circuits ) 的设计，其中可编程逻辑器件包括复杂可编程逻辑器件 ( CPLD : Complex Programmable Logic Devices ) 和现场可编程门阵列 ( FPGA : Field Programmable Gate Arrays ) 。

一段Verilog代码编写完成后，用户可以使用Altera或Xilinx等厂商生产的可编程逻辑器件来实现整个电路，或者将其提交给专业的代工厂用于ASIC的生产，这也是目前许多复杂的商用芯片 ( 例如微控制器 ) 所采用的实现方法。

关于Verilog语言，最后需要说明的是，它不同于常规的顺序执行的计算机程序 ( program ) ，Verilog从根本上讲是并发执行的，因此我们通常称之为Verilog代码 ( code ) ，而不是Verilog程序。

编辑推荐

《VerilogHDL数字系统设计与验证》特色：· 语法介绍清晰简洁，配套例题针对性强，包括必要的顶层电路图、设计代码、电路综合结果、验证代码和仿真结果，有助于读者全面理解。

- 将状态机的设计独立成章，总结了3种常用状态机设计风格，并通过典型例题进行对照分析。
- 加强了对常用系统函数和任务的内容讲述，给出典型例题并辅以解释说明，使读者易于理解。
- 重视数字系统的设计验证，采用专门的章节进行全面分析。
- 深入讨论数字电路设计中的时钟问题。

详细分析静态定时分析方法的原理，并讨论多时钟并存时的时钟域划分和同步化设计等问题。

- 部分例题源于实际的工程设计项目，可供工程技术人员直接参考使用。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>