

<<轻松成为设计高手>>

图书基本信息

书名：<<轻松成为设计高手>>

13位ISBN编号：9787512407909

10位ISBN编号：7512407904

出版时间：2012-6

出版时间：北京航空航天大学出版社

作者：EDA先锋工作室

页数：239

字数：350000

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>

<<轻松成为设计高手>>

内容概要

《轻松成为设计高手——Verilog HDL实用精解》主要分为3部分：第1~3章描述了Verilog HDL语言的基本概念、设计流程、语法与建模方式；第4~6章讨论如何合理使用Verilog HDL描述高性能的可综合电路；第7、8章重点描述了如何编写测试激励以及Verilog的仿真原理。另外，第9章对Verilog HDL语言的发展趋势做了展望。

《轻松成为设计高手——Verilog HDL实用精解》可作为高等院校通信工程、电子工程、计算机、微电子与半导体学等理工专业的教材，也可作为FPGA / CPLD逻辑设计工程师、IC工程师、硬件电路设计工程师的实用工具书。本书由EDA先锋工作室编著。

书籍目录

第1章 HDL设计方法简介

- 1.1 设计方法的变迁
- 1.2 Verilog语言的特点
 - 1.2.1 Verilog的由来
 - 1.2.2 HDL与原理图
 - 1.2.3 Verilog和HDL
 - 1.2.4 Verilog和C
- 1.3 HDL设计与验证流程
- 1.4 问题与思考

第2章 Verilog语言基础

- 2.1 Top-Down和Bottom-Up
- 2.2 从一个实例开始
 - 2.2.1 实例
 - 2.2.2 3种描述方法
- 2.3 基本词法
- 2.4 模块和端口
- 2.5 编译指令
- 2.6 逻辑值与常量
 - 2.6.1 逻辑值
 - 2.6.2 常量
- 2.7 变量类型
 - 2.7.1 线网类型
 - 2.7.2 寄存器类型
 - 2.7.3 变量的物理含义
 - 2.7.4 驱动和赋值
- 2.8 参数
- 2.9 Verilog 中的并发与顺序
- 2.10 操作数、操作符和表达式
 - 2.10.1 操作符
 - 2.10.2 二进制数值
 - 2.10.3 操作数
- 2.11 系统任务和系统函数
 - 2.11.1 显示任务
 - 2.11.2 文件输入 / 输出任务
 - 2.11.3 其他系统任务和函数
- 2.12 小结
- 2.13 问题与思考

第3章 描述方式和设计层次

- 3.1 描述方式
- 3.2 数据流描述
 - 3.2.1 数据流
 - 3.2.2 连续赋值语句
 - 3.2.3 延时
 - 3.2.4 多驱动源线网
- 3.3 行为描述

<<轻松成为设计高手>>

- 3.3.1 行为描述的语句格式
- 3.3.2 过程赋值语句
- 3.3.3 语句组
- 3.3.4 高级编程语句
- 3.4 结构化描述
 - 3.4.1 实例化模块的方法
 - 3.4.2 参数化模块
- 3.5 设计层次
 - 3.5.1 系统级和行为级
 - 3.5.2 RTL级
 - 3.5.3 门级
 - 3.5.4 晶体管级
 - 3.5.5 混合描述
- 3.6 CRC计算与校验电路实例
 - 3.6.1 CRC10校验, 行为级
 - 3.6.2 CRC10 机算电路, RTL级
- 3.7 小结
- 3.8 问题与思考
- 第4章 RTL概念与常用RTL建模
 - 4.1 RTL和综合的概念
 - 4.2 RTL级的基本要素和设计步骤
 - 4.3 常用RTL级建模
 - 4.3.1 非阻塞赋值、阻塞赋值、连续赋值
 - 4.3.2 寄存器电路建模
 - 4.3.3 组合逻辑建模
 - 4.3.4 双向端口与三态信号建模
 - 4.3.5 mux建模
 - 4.3.6 存储器建模
 - 4.3.7 简单的时钟分频电路
 - 4.3.8 串 / 并转换建模
 - 4.3.9 同步复位和异步复位
 - 4.3.10 用case和if...else建模
 - 4.3.11 可综合的Verilog语法子集
 - 4.4 CPU读 / 写PLD寄存器接口设计实例
 - 4.5 小结
 - 4.6 问题与思考
- 第5章 RTL设计与编码指导
 - 5.1 一般性指导原则
 - 5.1.1 面积和速度的平衡与互换原则
 - 5.1.2 硬件原则
 - 5.1.3 系统原则
 - 5.2 同步设计原则和多时钟处理
 - 5.2.1 同步设计原则
 - 5.2.2 亚稳态
 - 5.2.3 异步时钟域数据同步
 - 5.3 代码风格
 - 5.3.1 Coding Style的分类

<<轻松成为设计高手>>

- 5.3.2 Coding Style的重要性
- 5.4 结构层次设计和模块划分
 - 5.4.1 结构层次化编码
 - 5.4.2 模块划分的技巧
- 5.5 组合逻辑的注意事项
 - 5.5.1 always组合逻辑信号敏感表
 - 5.5.2 组合逻辑环路
 - 5.5.3 脉冲产生器
 - 5.5.4 慎用锁存器
- 5.6 时钟设计的注意事项
 - 5.6.1 内部逻辑产生的时钟
 - 5.6.2 Ripple Counter
 - 5.6.3 时钟选择
 - 5.6.4 门控时钟
 - 5.6.5 时钟同步使能端
- 5.7 RTL代码优化技巧
 - 5.7.1 使用Pipelining技术优化时序
 - 5.7.2 模块复用与Resource Sharing
 - 5.7.3 逻辑复制
 - 5.7.4 香农扩展运算
- 5.8 小结
- 5.9 问题与思考
- 第6章 如何写好状态机
 - 6.1 状态机的基本概念
 - 6.1.1 状态机是一种思想方法
 - 6.1.2 状态机的基本要素与分类
 - 6.1.3 状态机的基本描述方式
 - 6.2 如何写好状态机
 - 6.2.1 什么是好的RTL级FSM描述
 - 6.2.2 RTL级状态机描述常用语法
 - 6.2.3 推荐的状态机描述方法
 - 6.2.4 状态机设计的其他技巧
 - 6.3 使用Synplify Pro分析FSM
 - 6.4 小结
 - 6.5 问题与思考
- 第7章 逻辑验证与testbench编写
 - 7.1 概述
 - 7.1.1 仿真和验证
 - 7.1.2 什么是testbench
 - 7.2 建立testbench, 仿真设计
 - 7.2.1 编写仿真激励
 - 7.2.2 搭建仿真环境
 - 7.2.3 确认仿真结果
 - 7.2.4 写testbench要注意什么
 - 7.3 CPU接口仿真实例
 - 7.3.1 设计简介
 - 7.3.2 一种testbench

<<轻松成为设计高手>>

- 7.3.3 另一种testbench
- 7.4 结构化testbench思想
 - 7.4.1 任务和函数
 - 7.4.2 总线模型重用
 - 7.4.3 测试套件
 - 7.4.4 测试用例
 - 7.4.5 结构化testbench
- 7.5 实例：结构化testbench的编写
 - 7.5.1 单顶层testbench
 - 7.5.2 多顶层testbench
- 7.6 扩展Verilog的高层建模能力
- 7.7 小结
- 7.8 问题与思考
- 第8章 Verilog语义和仿真原理
 - 8.1 从一个问题说起
 - 8.2 电路与仿真
 - 8.2.1 电路是并行的
 - 8.2.2 Verilog是并行语言
 - 8.2.3 仿真器串行执行，Verilog仿真语义
 - 8.3 仿真原理
 - 8.3.1 Verilog如何仿真
 - 8.3.2 仿真时间
 - 8.3.3 事件驱动
 - 8.3.4 进程
 - 8.3.5 调度
 - 8.3.6 时序控制
 - 8.3.7 进程、事件和仿真时间的关系
 - 8.3.8 Verilog语言的不确定性
 - 8.4 分层事件队列与仿真参考模型
 - 8.4.1 分层事件队列
 - 8.4.2 仿真参考模型
 - 8.5 时序模型与延时
 - 8.5.1 仿真模型
 - 8.5.2 时序模型
 - 8.5.3 案例分析
 - 8.5.4 如何在Verilog语言中正确的增加延时
 - 8.6 再谈阻塞与非阻塞赋值
 - 8.6.1 本质
 - 8.6.2 案例分析
 - 8.7 如何提高代码的仿真效率
 - 8.8 防止仿真和综合结果不一样
 - 8.9 小结
 - 8.10 问题与思考
- 第9章 设计与验证语言发展趋势
 - 9.1 设计与验证语言发展历程
 - 9.1.1 HDL作为设计和验证语言
 - 9.1.2 C / C++和私有的验证语言

<<轻松成为设计高手>>

- 9.1.3 Accellera和IEEE的标准化工作
 - 9.2 硬件设计语言发展的现状和走向
 - 9.2.1 HDL的竞争
 - 9.2.2 一些尝试
 - 9.2.3 下一代的Verilog语言
 - 9.2.4 SystemC
 - 9.3 验证语言发展现状和走向
 - 9.3.1 验证方法
 - 9.3.2 HVL标准化进程
 - 9.3.3 HVL的新需求
 - 9.4 总结和展望
 - 9.5 小结
 - 9.6 问题与思考
- 参考文献

<<轻松成为设计高手>>

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>