

Quartus II

Verilog HDL与FPGA 数字系统设计

罗杰 主编

谭力 刘文超 王贞炎 张大卫 参编



机械工业出版社
China Machine Press

机械工业出版社·北京·邮政编码：100071

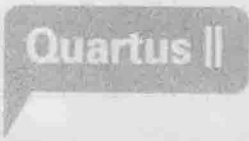
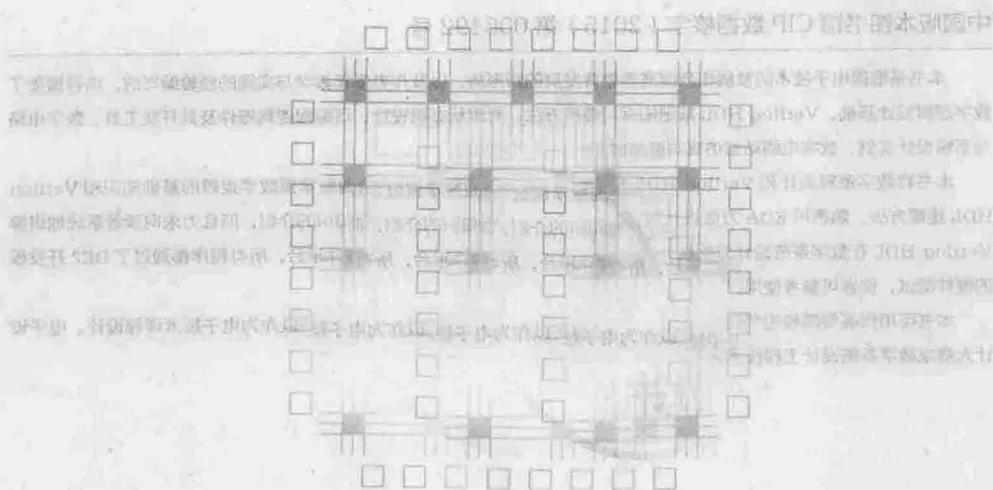
语言 (CIP)

EDA与IC设计

1-15084-1 141-4821-1

1-15084-1 141-4821-1

中国版本图书馆CIP数据核字(2012)第113331号



Verilog HDL与FPGA 数字系统设计

罗杰 主编
谭力 刘文超 王贞炎 张大卫 参编

机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

Verilog HDL 与 FPGA 数字系统设计 / 罗杰主编. —北京: 机械工业出版社, 2015.3
(高等院校电子信息与电气学科系列规划教材)

ISBN 978-7-111-48951-1

I. V… II. 罗… III. ① 硬件描述语言—程序设计—高等学校—教材 ② 可编程逻辑器件—系统设计—高等学校—教材 IV. ① TP312 ② TP332.1

中国版本图书馆 CIP 数据核字 (2015) 第 006492 号

本书是根据电子技术的发展和我国高等教育发展的新形势, 以及作者多年教学与实践的经验编写的。内容覆盖了数字逻辑设计基础、Verilog HDL 基础知识与建模方法、有限状态机设计、可编程逻辑器件及其开发工具、数字电路与系统设计实例、数字电路动态仿真与静态时序分析等相关知识。

本书将数字逻辑设计和 Verilog HDL 有机地结合在一起, 方便读者快速地掌握数字逻辑的基础知识和 Verilog HDL 建模方法, 熟悉用 EDA 方法设计数字系统的技巧。本书不仅注重基础知识的介绍, 而且力求向读者系统地讲解 Verilog HDL 在数字系统设计方面的实际应用。本书以 Quartus II 9.1 为软件平台, 所有程序都通过了 DE2 开发板的硬件测试, 读者可参考使用。

本书可用作高等院校电气信息类专业本、专科生的教材或教学参考书, 也可以作为电子技术课程设计、电子设计大赛或数字系统设计工程技术人员学习 EDA 技术的参考书。

出版发行: 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: 张国强

责任校对: 董纪丽

印刷: 三河市宏图印务有限公司

版次: 2015 年 4 月第 1 版第 1 次印刷

开本: 186mm × 240mm 1/16

印张: 25

书号: ISBN 978-7-111-48951-1

定价: 69.00 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) 88379210 88361066

投稿热线: (010) 88379007

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzjg@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

前 言

随着数字技术的高速发展,人们已经不再采用各种功能固定的通用中、小规模集成电路和电路图输入方法设计数字系统,而是广泛地采用硬件描述语言对数字电路的行为进行建模,并使用电子设计自动化(Electronic Design Automation, EDA)软件自动地对所设计的电路进行优化和仿真,然后使用逻辑综合工具将设计转化成物理实现的网表文件,最后用可编程逻辑器件或者专用集成电路(Application Specific Integrated Circuit, ASIC)完成数字系统。因此,掌握硬件描述语言、EDA技术和可编程逻辑器件已成为当今数字系统设计者的重要任务。

目前,符合IEEE标准的硬件描述语言(Hardware Description Language, HDL)有VHDL和Verilog HDL。两者的应用广泛,都能够通过程序描述电路的功能,从而进行数字电路的设计。由于Verilog HDL在ASIC设计领域占有重要的地位,并且它是在C语言的基础上发展起来的,语法较自由,易学易用,因此本书选取Verilog HDL进行电路设计。同时,本书还介绍了ModelSim软件和Quartus II软件的使用方法,读者可以使用它们进行仿真和综合Verilog HDL代码。

本书是作者根据多年的教学科研经验以及指导学生参加全国电子设计竞赛经验编写而成的。在内容上,将数字逻辑设计和Verilog HDL有机结合在一起,方便读者快速进入现代数字逻辑设计领域。按照“数字逻辑设计基础、Verilog HDL建模技术、可编程逻辑器件的结构原理、EDA设计工具软件、数字电路系统设计实践”的体系结构编写。为了让大家更容易掌握Verilog HDL知识,本书在介绍数字电路设计的过程中列举了Verilog HDL的很多例程,并假定读者没有任何数字逻辑基础知识。

全书共11章。首先介绍了数字逻辑运算、逻辑门、组合电路设计等基础知识,接着重点介绍了Verilog HDL基础知识与建模方法,对状态机的建模方法进行了深入讨论;然后讨论各种可编程逻辑器件的组成、结构特点和开发流程,以及Quartus II软件的使用方法和静态时序分析方法;最后通过大量的例程介绍Verilog HDL在数字系统设计方面的应用,有助于读者理解书中的基本概念并掌握从简单电路到复杂模块的设计技术。

本书力求做到通俗易懂，适教适学。为方便读者学习，每章开头均有“本章目的”，介绍该章将要学习的主要内容，每章后面均安排有小结，部分章节后面配有习题。理论学习要和上机实验相结合，从第7章开始通过精选的例程进行引导，读者可以按照这些例程进行实际操作，将HDL代码“写入”FPGA芯片，对设计的电路进行实际测试，以方便读者掌握FPGA开发的整个流程。

参加本书编写工作的有华中科技大学的罗杰(第1、2、3、4、5章)、张大卫(第6、7章、附录C)、谭力(第8、10章)、王贞炎(第9章)和湖北大学的刘文超(第11章、附录A、B)等，罗杰担任主编，负责全书的策划、组织整理和定稿工作。

本书在编写过程中，得到了华中科技大学电工电子科技创新基地的大力支持；得到了华中科技大学“教学改革工程”教材建设基金资助；还得到康华光教授的热情支持和鼓励，在此表示衷心的感谢。

由于作者知识水平有限，书中难免有疏漏、不妥或错误之处，敬请各位专家、同行和读者批评指正。您可以通过Luojiewh@gmail.com给作者发送邮件，我们会阅读所有来信，并尽可能尽快回复。

编者

2014年11月

教学建议

学 时	教 学 内 容	教 学 要 求	学 时	
			理 论	实 验
6	第 1 章 数字逻辑设计基础	<ul style="list-style-type: none"> 了解数字电路历史和集成电路逻辑系列 掌握十进制数、二进制数、十六进制数之间的相互转换 掌握常见的逻辑运算规则,了解逻辑门的图形符号 掌握逻辑代数的原理和逻辑函数化简方法 掌握组合逻辑电路分析与设计方法 	6	0
2	第 2 章 Verilog HDL 入门与 功能仿真	<ul style="list-style-type: none"> 了解 Verilog HDL 模块程序的基本组成 了解用 Verilog HDL 描述数字逻辑电路的几种不同风格 掌握逻辑功能仿真过程以及测试平台(testbench)的写法 	2	2
6	第 3 章 Verilog HDL 基础语法与 组合逻辑电路建模	<ul style="list-style-type: none"> 掌握 Verilog HDL 基础语法及逻辑门原语 掌握组合逻辑电路建模的方法 掌握常用组合电路及其设计 掌握分模块、分层次的电路设计方法 	6	4
6	第 4 章 时序逻辑电路建模	<ul style="list-style-type: none"> 掌握锁存器、触发器、计数器的逻辑功能 掌握同步时序电路的设计方法 掌握用 Verilog HDL 描述时序逻辑电路的方法 	6	4
4	第 5 章 有限状态机设计	<ul style="list-style-type: none"> 掌握状态机的基本结构及其表示方法 掌握用 Verilog HDL 描述状态图的方法 	4	2
4	第 6 章 可编程逻辑器件	<ul style="list-style-type: none"> 了解可编程逻辑器件类型及其简化符号 了解 CPLD 和 FPGA 组成及其结构特点 掌握 PLD 器件开发流程 了解用与/或阵列和 LUT 实现逻辑函数的原理 	4	0
2	第 7 章 FPGA 开发工具的使用 (自学与实验)	<ul style="list-style-type: none"> 掌握 Quartus II 软件的使用方法 掌握嵌入式逻辑分析仪 SignalTap II 的使用 了解在 Quartus II 软件中直接调用 ModelSim 软件进行仿真的流程 	2	4
0	第 8 章 数字电路与系统的设计 实践(自学与实验)	<ul style="list-style-type: none"> 掌握计数器、分频器、定时器、数字钟、频率计、信号发生器等电路设计与实现 了解 Quartus II 软件对状态机的状态进行设置的方法 	0	24
2	第 9 章 静态时序分析工具 TimeQuest 的使用(讲解+实验)	<ul style="list-style-type: none"> 了解静态时序分析的基础知识 掌握 TimeQuest Timing Analyzer 的使用 掌握对时序报告进行分析的方法 	2	8

(续)

教 学 内 容	教 学 要 求	学 时	
		理 论	实 验
第 10 章 异步串口通信及 UART 实现(选讲)	<ul style="list-style-type: none"> 了解异步串行通信接口的实现原理 掌握一个完整数字系统的开发流程 	2	12
第 11 章 VGA 接口控制器的设计(选讲)	<ul style="list-style-type: none"> 了解 VGA 接口原理、工作时序和接口电路 掌握 VGA 彩条信号发生器和二十四位位图显示设计与实现 	2	12
总课时	第 1~9 章建议课时	32	48
	综合实训(第 10~11 章)建议课时	4	24

说明:

- 1) 建议课堂教学工作全部在配有多媒体的实验室完成, 实现“讲—练”结合。
- 2) 可将教学内容分为两个模块: 核心知识技能模块(前 9 章)和技能提高模块(第 10~11 章), 其中核心知识技能模块建议教学学时为 32(理论)+48(实验), 技能提高模块建议学时为 4(理论)+24(实验), 不同学校可以根据各自的教學要求和计划学时数对教学内容进行取舍。

	<ul style="list-style-type: none"> 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 	第 7 章	并行接口芯片 I/O 控制信号
	<ul style="list-style-type: none"> 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 	第 8 章	并行接口芯片 I/O 控制信号
	<ul style="list-style-type: none"> 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 	第 9 章	并行接口芯片 I/O 控制信号
	<ul style="list-style-type: none"> 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 	第 10 章	并行接口芯片 I/O 控制信号
	<ul style="list-style-type: none"> 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 	第 11 章	并行接口芯片 I/O 控制信号
	<ul style="list-style-type: none"> 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 	第 12 章	并行接口芯片 I/O 控制信号
	<ul style="list-style-type: none"> 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 	第 13 章	并行接口芯片 I/O 控制信号
	<ul style="list-style-type: none"> 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 	第 14 章	并行接口芯片 I/O 控制信号
	<ul style="list-style-type: none"> 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 	第 15 章	并行接口芯片 I/O 控制信号
	<ul style="list-style-type: none"> 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 	第 16 章	并行接口芯片 I/O 控制信号
	<ul style="list-style-type: none"> 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 	第 17 章	并行接口芯片 I/O 控制信号
	<ul style="list-style-type: none"> 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 	第 18 章	并行接口芯片 I/O 控制信号
	<ul style="list-style-type: none"> 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 	第 19 章	并行接口芯片 I/O 控制信号
	<ul style="list-style-type: none"> 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 	第 20 章	并行接口芯片 I/O 控制信号
	<ul style="list-style-type: none"> 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 	第 21 章	并行接口芯片 I/O 控制信号
	<ul style="list-style-type: none"> 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 	第 22 章	并行接口芯片 I/O 控制信号
	<ul style="list-style-type: none"> 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 	第 23 章	并行接口芯片 I/O 控制信号
	<ul style="list-style-type: none"> 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 	第 24 章	并行接口芯片 I/O 控制信号
	<ul style="list-style-type: none"> 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 	第 25 章	并行接口芯片 I/O 控制信号
	<ul style="list-style-type: none"> 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 	第 26 章	并行接口芯片 I/O 控制信号
	<ul style="list-style-type: none"> 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 	第 27 章	并行接口芯片 I/O 控制信号
	<ul style="list-style-type: none"> 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 	第 28 章	并行接口芯片 I/O 控制信号
	<ul style="list-style-type: none"> 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 	第 29 章	并行接口芯片 I/O 控制信号
	<ul style="list-style-type: none"> 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 掌握 8255 并行接口芯片的 I/O 控制信号 	第 30 章	并行接口芯片 I/O 控制信号

目 录

前 言 教学建议

第一篇 数字系统基础

第 1 章 数字逻辑设计基础 / 2

- 1.1 数字电路的发展历史及分类 / 2
- 1.2 逻辑运算及逻辑门 / 5
 - 1.2.1 基本逻辑运算及对应的逻辑门 / 5
 - 1.2.2 常用复合逻辑运算及对应的逻辑门 / 9
 - 1.2.3 集成逻辑门电路简介 / 10
 - 1.2.4 三态门 / 10
- 1.3 逻辑代数的基本公式和规则 / 12
 - 1.3.1 逻辑代数的基本公式 / 12
 - 1.3.2 逻辑代数的基本规则 / 13
 - 1.3.3 基本公式的应用 / 14
- 1.4 逻辑函数的代数化简法 / 16
- 1.5 逻辑函数的卡诺图化简法 / 18
 - 1.5.1 最小项的定义和性质 / 18
 - 1.5.2 逻辑函数的最小项表达式 / 18
 - 1.5.3 用卡诺图表示逻辑函数 / 19
 - 1.5.4 用卡诺图化简逻辑函数 / 22
 - 1.5.5 用卡诺图化简含无关项

的逻辑函数 / 24

1.6 组合逻辑电路设计 / 25

- 1.6.1 组合逻辑电路设计的一般步骤 / 25
- 1.6.2 组合逻辑电路设计举例 / 26

小结 / 30

习题 / 30

第 2 章 Verilog HDL 入门与功能仿真 / 32

- 2.1 硬件描述语言简介 / 32
 - 2.1.1 硬件描述语言的起源 / 32
 - 2.1.2 硬件描述语言的特点 / 33
- 2.2 Verilog HDL 程序的基本结构 / 34
 - 2.2.1 Verilog HDL 程序的基本概述 / 34
 - 2.2.2 简单 Verilog HDL 程序实例 / 35
- 2.3 逻辑功能的仿真验证过程 / 37
 - 2.3.1 激励块 / 38
 - 2.3.2 仿真过程简介 / 39
- 2.4 ModelSim 仿真软件的使用 / 40
 - 2.4.1 创建工作目录 / 40
 - 2.4.2 输入源文件 / 40

- 2.4.3 建立工作库 / 40
- 2.4.4 编译设计文件 / 41
- 2.4.5 将设计文件载入
仿真器 / 41
- 2.4.6 运行仿真器 / 42
- 2.5 Verilog HDL 功能仿真常用
命令 / 45
 - 2.5.1 系统任务 / 45
 - 2.5.2 编译器指令 / 47
- 小结 / 50
- 习题 / 50

第 3 章 Verilog HDL 基础语法与组合 逻辑电路建模 / 51

- 3.1 Verilog HDL 基本语法
规则 / 51
 - 3.1.1 词法规定 / 51
 - 3.1.2 逻辑值集合 / 52
 - 3.1.3 常量及其表示 / 52
 - 3.1.4 数据类型 / 55
- 3.2 Verilog HDL 门级建模 / 61
 - 3.2.1 多输入门 / 62
 - 3.2.2 多输出门 / 63
 - 3.2.3 三态门 / 64
 - 3.2.4 门级建模举例 / 65
- 3.3 Verilog HDL 数据流建模与
运算符 / 67
 - 3.3.1 数据流建模 / 67
 - 3.3.2 表达式与操作数 / 68
 - 3.3.3 运算符 / 69
 - 3.3.4 运算符的优先级别 / 74
- 3.4 组合电路的行为级建模 / 75
- 3.5 分层次的电路设计方法 / 84
 - 3.5.1 设计方法 / 84

- 3.5.2 模块实例引用语句 / 85
- 3.6 常用组合电路及其设计 / 87
 - 3.6.1 编码器 / 87
 - 3.6.2 二进制译码器 / 89
 - 3.6.3 七段显示译码器 / 91
 - 3.6.4 二进制数与 8421 BCD 码
的转换 / 94
- 小结 / 100
- 习题 / 100

第 4 章 时序逻辑电路建模 / 102

- 4.1 锁存器 / 102
 - 4.1.1 基本 SR 锁存器 / 102
 - 4.1.2 门控 D 锁存器 / 103
 - 4.1.3 门控 D 锁存器的 Verilog
HDL 建模 / 105
- 4.2 时序电路建模基础 / 106
 - 4.2.1 阻塞型赋值语句和非阻塞
型赋值语句 / 106
 - 4.2.2 事件控制语句 / 106
- 4.3 触发器 / 107
 - 4.3.1 D 触发器的逻辑功能 / 107
 - 4.3.2 有清零输入和预置输入
的 D 触发器 / 108
 - 4.3.3 有使能端的 D 触发器 / 110
 - 4.3.4 D 触发器及其应用电路
的 Verilog HDL 建模 / 110
- 4.4 寄存器和移位寄存器 / 114
 - 4.4.1 寄存器及 Verilog HDL
建模 / 114
 - 4.4.2 移位寄存器及 Verilog HDL
建模 / 115
 - 4.4.3 移位寄存器的应用电路 / 118
- 4.5 同步计数器 / 121

4.5.1	同步计数器的设计 / 122
4.5.2	同步计数器的 Verilog HDL 建模 / 125
4.6	Verilog HDL 函数与任务的 使用 / 128
4.6.1	函数说明语句 / 129
4.6.2	任务说明语句 / 132
4.7	m 序列码产生电路设计 / 134
	小结 / 138
	习题 / 138

第 5 章 有限状态机设计 / 140

5.1	状态机的基本概念 / 140
5.1.1	状态机的基本结构及 类型 / 140
5.1.2	状态机的状态图 表示法 / 141
5.1.3	状态机的设计步骤 / 141
5.2	基于 Verilog HDL 的状态机 描述方法 / 142
5.2.1	状态图的建立过程 / 142
5.2.2	状态图的描述方法 / 143
5.3	状态机设计中的关键技术 / 147
5.3.1	状态编码 / 147
5.3.2	消除输出端产生的 毛刺 / 147
5.3.3	使用 One-Hot 编码方案 设计状态机 / 150
5.4	状态机设计举例 / 151
5.4.1	汽车尾灯控制电路 设计 / 151
5.4.2	十字路口交通灯控制 电路设计 / 154
	小结 / 160

习题 / 161

第 6 章 可编程逻辑器件 / 163

6.1	概述 / 163
6.1.1	PLD 的历史 / 163
6.1.2	PLD 开发流程简介 / 164
6.1.3	PLD 器件的符号 / 165
6.2	简单可编程逻辑器件 / 166
6.2.1	PLA / 167
6.2.2	PAL / 168
6.3	复杂可编程逻辑器件 / 175
6.3.1	CPLD 的基本结构 / 175
6.3.2	逻辑块 / 176
6.3.3	I/O 块 / 177
6.3.4	可编程内部互连线 资源 / 178
6.4	现场可编程门阵列 / 179
6.4.1	FPGA 实现逻辑函数的 基本原理 / 179
6.4.2	FPGA 的一般结构 / 182
6.4.3	基于 LUT 的逻辑块 / 182
6.4.4	可编程布线资源 / 183
6.4.5	I/O 块 / 184
	小结 / 186
	习题 / 187

第二篇 数字系统设计实践

第 7 章 FPGA 开发工具的使用 / 190

7.1	Quartus II 软件介绍 / 190
7.1.1	Quartus II 9.1 软件 主界面 / 190
7.1.2	Quartus II 的设计 流程 / 191
7.1.3	USB-Blaster 驱动

- 安装 / 193
 - 7.2 基于原理图的电路仿真 / 194
 - 7.2.1 建立新的设计项目 / 195
 - 7.2.2 输入电路原理图 / 196
 - 7.2.3 编译设计项目 / 197
 - 7.2.4 仿真验证设计项目 / 198
 - 7.2.5 分析信号的延迟特性 / 201
 - 7.2.6 实验任务 / 201
 - 7.3 基于 Verilog HDL 的电路设计与实现 / 202
 - 7.3.1 半加器的设计与 Verilog HDL 建模举例 / 203
 - 7.3.2 输入设计文件 / 204
 - 7.3.3 建立新的设计项目 / 204
 - 7.3.4 编译设计文件 / 205
 - 7.3.5 仿真验证设计项目 / 206
 - 7.3.6 分配引脚 / 208
 - 7.3.7 对目标器件编程与硬件电路测试 / 210
 - 7.3.8 使用电路网表观察器查看电路图 / 212
 - 7.3.9 实验任务 / 212
 - 7.4 基于原理图和 Verilog HDL 的层次化设计 / 213
 - 7.4.1 编码、译码、显示电路 / 213
 - 7.4.2 建立新的设计项目 / 215
 - 7.4.3 输入 HDL 底层文件并完善原理图 / 216
 - 7.4.4 分配引脚并编译设计 / 216
 - 7.4.5 仿真验证设计项目 / 217
 - 7.4.6 对目标器件编程与硬件电路测试 / 218
 - 7.4.7 实验任务 / 219
 - 7.5 嵌入式逻辑分析仪 SignalTap II 的使用 / 220
 - 7.5.1 SignalTap II 的实现原理与使用流程 / 220
 - 7.5.2 SignalTap II 的基本使用方法 / 221
 - 7.5.3 实验任务 / 227
 - 7.6 宏功能模块的调用 / 227
 - 7.6.1 计数器模块 LPM_COUNTER 的配置与调用 / 227
 - 7.6.2 嵌入式锁相环模块 ALTPLL 的配置与调用 / 230
 - 7.6.3 先进先出模块 FIFO 的配置与调用 / 236
 - 7.6.4 存储器模块 LPM_ROM 的配置与调用 / 239
 - 7.6.5 实验任务 / 245
 - 7.7 在 Quartus II 中调用 ModelSim 进行仿真 / 246
 - 7.7.1 乘法器模块 LPM_MULT 的配置与调用 / 246
 - 7.7.2 仿真流程 / 247
 - 7.7.3 实验任务 / 250
 - 小结 / 250
- ## 第 8 章 数字电路与系统的设计实践 / 251
- 8.1 变模计数器设计 / 251
 - 8.1.1 功能要求 / 251
 - 8.1.2 设计分析 / 251
 - 8.1.3 逻辑设计 / 251
 - 8.1.4 设计实现 / 253
 - 8.1.5 实验任务 / 253
 - 8.2 移动显示字符的设计 / 253

- 8.2.1 功能要求 / 253
 - 8.2.2 设计分析 / 254
 - 8.2.3 逻辑设计 / 255
 - 8.2.4 设计实现 / 256
 - 8.2.5 实验任务 / 256
 - 8.3 分频器设计 / 258
 - 8.3.1 功能要求 / 258
 - 8.3.2 设计分析 / 258
 - 8.3.3 逻辑设计 / 259
 - 8.3.4 设计仿真 / 260
 - 8.3.5 实际运行结果 / 260
 - 8.3.6 实验任务 / 260
 - 8.4 篮球竞赛 30 秒定时器设计 / 261
 - 8.4.1 功能要求 / 261
 - 8.4.2 设计分析 / 261
 - 8.4.3 逻辑设计 / 261
 - 8.4.4 设计实现 / 264
 - 8.4.5 实验任务 / 264
 - 8.5 多功能数字钟设计 / 265
 - 8.5.1 功能要求 / 265
 - 8.5.2 设计分析 / 265
 - 8.5.3 数字钟主体电路逻辑设计 / 266
 - 8.5.4 功能扩展电路逻辑设计 / 270
 - 8.5.5 顶层电路设计 / 271
 - 8.5.6 实验任务 / 274
 - 8.6 频率计设计 / 274
 - 8.6.1 功能要求 / 274
 - 8.6.2 设计分析 / 275
 - 8.6.3 逻辑设计 / 277
 - 8.6.4 顶层电路设计 / 282
 - 8.6.5 实验任务 / 282
 - 8.7 DDS 函数信号发生器的设计 / 283
 - 8.7.1 功能要求 / 283
 - 8.7.2 DDS 产生波形的原理 / 283
 - 8.7.3 设计分析 / 286
 - 8.7.4 顶层电路设计 / 289
 - 8.7.5 设计实现 / 290
 - 8.7.6 D/A 转换电路及放大电路设计 / 291
 - 8.7.7 实验任务 / 292
 - 8.8 有限状态机实验 / 294
 - 8.8.1 功能要求 / 294
 - 8.8.2 设计分析 / 295
 - 8.8.3 逻辑设计 / 296
 - 8.8.4 设计实现 / 297
 - 8.8.5 实验任务 / 298
 - 小结 / 299
- 第 9 章 静态时序分析工具 TimeQuest 的使用 / 300**
- 9.1 静态时序分析基础 / 301
 - 9.1.1 同步路径的分析 / 301
 - 9.1.2 异步路径的分析 / 305
 - 9.1.3 外部同步路径的分析 / 305
 - 9.1.4 不同的时序模型 / 309
 - 9.2 TimeQuest 时序分析器的使用 / 309
 - 9.2.1 TimeQuest 的使用流程 / 309
 - 9.2.2 两级流水线乘法器设计 / 309
 - 9.2.3 设定时序要求 / 312
 - 9.2.4 全编译并完成布局布线 / 315
 - 9.2.5 验证时序 / 316

小结 / 318

第 10 章 异步串口通信及 UART

实现 / 319

10.1 UART 接口实现原理 / 319

10.1.1 串行通信的概念 / 319

10.1.2 基本的 UART 通信
协议 / 320

10.2 UART 接口模块的层次化 设计 / 321

10.2.1 UART 接口的功能模块
划分 / 321

10.2.2 配置文件 / 322

10.2.3 顶层模块的功能
描述 / 323

10.2.4 接收模块的功能
描述 / 324

10.2.5 发送模块的功能
描述 / 327

10.2.6 波特率变换模块的
功能描述 / 331

10.2.7 微处理器接口模块的
功能描述 / 331

10.3 对 UART 接口模块的 功能仿真 / 338

10.3.1 对接收模块的功能
仿真 / 338

10.3.2 对发送模块的功能
仿真 / 340

10.3.3 对波特率变换模块的
功能仿真 / 341

10.3.4 对微处理器接口模块的
功能仿真 / 341

10.3.5 对 UART 接口模块的
功能仿真 / 343

10.4 逻辑综合与时序仿真 / 345

10.5 下载与验证测试 / 345

10.5.1 验证系统概述 / 345

10.5.2 验证结果 / 346

小结 / 347

第 11 章 VGA 接口控制器的设计 / 348

11.1 VGA 接口标准和接口
电路 / 348

11.1.1 VGA 接口标准 / 348

11.1.2 VGA 接口电路 / 350

11.2 VGA 彩条信号发生器 / 352

11.2.1 功能要求 / 352

11.2.2 设计分析 / 352

11.2.3 逻辑设计 / 353

11.2.4 顶层电路设计 / 357

11.2.5 对目标器件编程与硬件
电路测试 / 358

11.2.6 使用 Signal Tap II 观察
VGA 工作时序 / 359

11.2.7 实验任务 / 361

11.3 24 位位图显示 / 362

11.3.1 功能要求 / 362

11.3.2 设计分析 / 362

11.3.3 逻辑设计 / 362

11.3.4 顶层电路设计 / 365

11.3.5 对目标器件编程与硬件
电路测试 / 367

11.3.6 实验任务 / 368

小结 / 368

附录 A Verilog HDL 关键字 / 369

附录 B 常用 FPGA 开发板介绍 / 370

附录 C Cyclone II 系列器件结构 / 380

参考文献 / 386

第1章 数字逻辑设计基础

第一篇

数字系统基础

第1章 数字逻辑设计基础

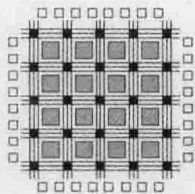
第2章 Verilog HDL入门与功能仿真

第3章 Verilog HDL基础语法与组合逻辑电路建模

第4章 时序逻辑电路建模

第5章 有限状态机设计

第6章 可编程逻辑器件



第 1 章

数字逻辑设计基础

本章目的

本章介绍数字逻辑运算、逻辑门、组合电路设计的一些基本知识，为后续学习做准备。具体内容如下：

- 了解数字集成电路的逻辑系列及常用逻辑门
- 说明与、或、非三种基本的逻辑运算和常用的复合逻辑运算
- 介绍布尔代数的基本公式和规则
- 说明逻辑函数的化简方法——代数化简法和卡诺图化简法
- 阐述使用逻辑门设计组合逻辑电路的方法

1.1 数字电路的发展历史及分类

数字电路是数字计算机和自动控制系统的基础，它经历了电子管、晶体管等半导体分立器件到集成电路(Integrated Circuit, IC)的发展历程。以计算机的发展为例，在 1946 年美国宾夕法尼亚大学研制的世界上第一台电子数字积分计算机 (Electronic Numerical Integrator And Computer, ENIAC)是用电子管构成的。

在 1947 年贝尔(Bell)实验室的 John Bardeen、Walter H. Brattain 及 William Shockley 共同发明了晶体管之后，人们逐渐开始用晶体管取代电子管构成数字系统，20 世纪 50 年代出现的第二代计算机(例如，IBM 7000 系列)是用分立的晶体管和磁芯存储器构成的。1959 年美国德州仪器(Texas Instruments, TI)公司的 Jack Kilby(杰克·基尔比)研制出世界上第一个集成电路(包括 4 个双极性晶体管、三个电阻和一个电容器)。此后，随着生产工艺的改进，IC 产量增大且集成度(芯片上包含的晶体管数目)提高，价格大幅度降低。在 2000 年，基尔比获得了瑞典皇家科学院授予的诺贝尔物理学奖以表彰他在集成电路领域的贡献。由于集成电路的发展非常迅速，很快占据了主导地位，因此，现在数字电路的主流形式是数字集成电路。

从 20 世纪 60 年代开始，各种不同型号的逻辑门、触发器以及能完成一些特定功能的集成电路(如译码器、加法器、寄存器、计数器、乘法器等)不断涌现，到 20 世纪 70 年代，包含 200~200 000 个等效逻辑门的大规模集成电路(Large Scale Integration, LSI)得以发展，出现了微处理器、小型存储器、可编程逻辑器件和定制器件。20 世纪 80 年代以后，各种不同类型的简单可编程逻辑器件得以发展，在 20 世纪 90 年代初，已经可以制造包含几百万个晶体管的微处理器，如今的芯片已经可以集成超过十亿个晶体管，可以将复杂的电子系统全部集成在一个芯片上，使集成电路设计向集成系统设计转变，这就是片上系统(System on Chip, SoC)。

伴随着器件的发展,从 20 世纪 60 年代中期开始,人们使用集成电路构成计算机,从此计算机得到飞速发展。20 世纪 70 年代后,计算机一方面向功能极强的大型和巨型机发展,另一方面,则向价格便宜的微型计算机发展,1971 年 Intel 公司发明了世界上第 1 台微处理器(Intel 4004)芯片,该芯片上集成了 1200 个晶体管。此后,出现了 8 位、16 位、32 位和 64 位等一系列的微处理器,以微处理器为主构成的计算机的性能也得到大幅度提升。

这里介绍一下摩尔定律(Moore's Law)。摩尔是 Intel 公司的创始人之一,他在 1965 年提出来半导体芯片上所能集成的晶体管数量每两年翻倍的预测,后来修正为每 18 个月翻一番。此后的实际情况证实了这个预测是正确的,直到今天仍在延续。

相对于使用分立器件组装的电路,集成电路把构成具有一定功能电路所需的晶体管、电阻、电容等元器件及它们之间的连接导线全部集成在一小块硅片上,然后焊接封装在一个管壳内,其封装外壳有圆壳式、双列直插式、扁平式或球形栅格阵列式等多种形式。这样提高了电路可靠性,减小了体积和功耗。

图 1.1.1 是带有塑料封装外壳的 IC 截面图,这种封装形式称为双列直插封装(Dual-Inline Package, DIP)。所有电路都集成在内部的芯片上,芯片通过细导线与外部引脚相连接。有时也称集成电路为芯片。

IC 的分类方法很多,根据芯片内部集成的逻辑门数目,早期把数字集成电路分为大、中、小三类,如表 1.1.1 所示。最简单的商用 IC 称为小规模集成(Small-Scale Integration, SSI)电路,芯片内部含有的逻辑门数目小于 12 个。中规模集成(Medium-Scale Integration, MSI)电路芯片内部含有的逻辑门数目在 12~100 个之间;大规模集成(Large-Scale Integration, LSI)电路芯片内部含有的逻辑门数目在 100~9999 个之间。随着技术的进步,后来出现的规模更大的集成电路称为超大规模集成(Very Large-Scale Integration, VLSI)电路。实际上,LSI 与 VLSI 之间的界限有些模糊不清,并且后来趋向于以晶体管的个数而不是以逻辑门的个数来界定 IC,凡是超过 100 万个晶体管的 IC 就一定是 VLSI,例如微处理器。

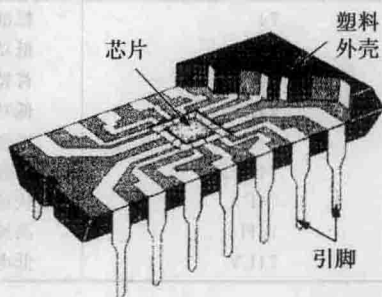


图 1.1.1 IC 封装示意图

表 1.1.1 数字集成电路的集成度分类

分 类	门的个数	典型集成电路
小规模	最多 12 个	逻辑门、触发器
中规模	12~99 个	计数器、加法器
大规模	100~9 999 个	小型存储器、门阵列
超大规模	10 000 个以上	大型存储器、微处理器、可编程逻辑器件等

由于晶体管有双极结型管(Bipolar Junction Transistor, BJT)和金属-氧化物-半导体场效应管(Metal-Oxide-Semiconductor Field Effect Transistor, MOSFET)两种类型,根据芯片内部所使用的晶体管类型不同,将使用 BJT 的芯片称为双极型集成电路,将使用 MOSFET 的芯片称为单极型集成电路。晶体管-晶体管逻辑(Transistor-Transistor Logic, TTL)是双极型集成电路的典型代表,基于互补金属氧化物半导体(Complementary MOS, CMOS)技术的 4000 系列则是单极型集成电路的典型代表。每一种类型的 IC 都有其优点和

缺点,选用时要考虑其工作温度范围、电源电压、运行速度、消耗功率及成本等很多因素。

TTL 是 1964 年由 TI 公司作为标准产品推出的, TI 公司称之为 54/74 逻辑系列。54 系列为军用型产品,其工作温度范围是 $-55\sim 125^{\circ}\text{C}$,工作电压范围是 $5\text{V}\pm 10\%$;而 74 系列为商用型产品,其工作温度范围是 $0\sim 70^{\circ}\text{C}$,工作电压范围是 $5\text{V}\pm 5\%$ 。所谓逻辑系列是一些功能不同的 IC 的集合,这些 IC 有类似的输入、输出特性。同一系列的 IC 可以通过外部的相互连线实现任意逻辑功能,而不同系列的 IC 可能采用不同的电源电压,或输入、输出逻辑电平值不同,需要使用接口电路才能相互连接。

为了进一步提高集成电路的工作速度和降低功耗,许多半导体公司对 74 系列 IC 的电路结构和制作工艺进行了改进,先后出现了很多不同的逻辑系列产品,如表 1.1.2 所示。其中,74LS 系列的使用范围最为广泛。

表 1.1.2 TTL 逻辑系列分类表

TTL 系列	说 明	缩写字母注释
74	标准 TTL (出现得最早)	无
74L	低功耗型	Low-power
74S	肖特基型	Schottky
74LS	低功耗肖特基型(广泛使用的)	Low-power Schottky
74AS	增强型肖特基型	Advanced Schottky
74ALS	增强型低功耗肖特基型	Advanced low-power Schottky
74F	快速型	Fast
74H	高速型	High-speed
74LV	低电源电压型	Low-voltage

第一个 CMOS 集成电路在 1968 年就被研发出来,且功耗也比 TTL 低,但由于其工作速度较慢的缘故,其应用范围受到一定的限制。经过长期研究与改良,CMOS 芯片的性能大大提高,现在 CMOS IC 无论在芯片使用的面积、工作的速度、耗损的功率,还是制造的成本上都比 TTL 要有优势。其中,低功耗和高集成度是 CMOS 电路的突出优点,非常适合制造大规模和超大规模集成电路。到 20 世纪 90 年代后期,CMOS 电路便逐渐取代 TTL 电路而成为数字集成电路的主流产品。

早期 CMOS IC 典型代表是 4000 系列,其供电电源在 $3\sim 18\text{V}$ 之间,后来为了能与 TTL 芯片兼容,多数 CMOS 芯片使用 5V 或者更低的电源。现在,CMOS 有 4000、74HC、74AC、74HCT 等系列,如表 1.1.3 所示。

表 1.1.3 CMOS 逻辑系列分类表

CMOS 系列	说 明
4000	最早出现的 CMOS,供电电源为 $3\sim 18\text{V}$
74HC	与 TTL 芯片的引脚兼容、编号相同的高速 CMOS,供电电源为 $2\sim 6\text{V}$
74HCT	类似于 74HC,并能与 TTL 直接相连,供电电源为 $4.5\sim 5.5\text{V}$
74AC	增强型 CMOS,供电电源为 $3.0\sim 5.5\text{V}$
74ACT	类似于 74AC,并能与 TTL 直接相连,供电电源为 $4.5\sim 5.5\text{V}$
74AHC	增强型高速 CMOS,供电电源为 $2.0\sim 5.5\text{V}$
74AHCT	类似于 74AHC,并能与 TTL 直接相连,供电电源为 $4.5\sim 5.5\text{V}$
74FCT	具有 TTL 输入电平的快速 CMOS,供电电源为 $4.75\sim 5.25\text{V}$
74LVC	低电源电压型,供电电源为 $2.0\sim 3.6\text{V}$