

电工电子科技创新人才培养系列教材



Verilog HDL 与数字ASIC设计基础

华中科技大学电工电子科技创新中心

主编 罗 杰

华中科技大学出版社

<http://www.hustp.com>

Verilog HDL

与数字ASIC设计基础

华中科技大学电工电子科技创新中心

主编 罗 杰

编著 罗 杰 肖 看 谭 力
柯志武 龚 军

华中科技大学出版社

<http://www.hustp.com>

中国·武汉

图书在版编目(CIP)数据

Verilog HDL 与数字 ASIC 设计基础/罗 杰 主编. —武汉:华中科技大学出版社, 2008 年 3 月

ISBN 978-7-5609-4404-3

I. V… II. 罗… III. ①硬件描述语言, Verilog HDL-程序设计-高等学校-教材
②数字电路-电路设计-高等学校-教材 IV. TP312 TN79

中国版本图书馆 CIP 数据核字(2008)第 032695 号

Verilog HDL 与数字 ASIC 设计基础

罗 杰 主编

责任编辑:沈旭日

封面设计:潘 群

责任校对:周 娟

责任监印:周治超

出版发行:华中科技大学出版社(中国·武汉)

武昌喻家山 邮编:430074 电话:(027)87557437

录 排:华中科技大学惠友文印中心

印 刷:湖北恒泰印务有限公司

开本:787mm×1092mm 1/16

印张:18.75

字数:456 000

版次:2008 年 3 月第 1 版

印次:2008 年 3 月第 1 次印刷

定价:28.00 元

ISBN 978-7-5609-4404-3/TP·650

(本书若有印装质量问题,请向出版社发行部调换)

内 容 简 介

本书从实用的角度介绍了硬件描述语言 Verilog HDL 基础知识,并重点介绍了可以综合成为硬件电路的语法结构、语句与建模方法。以 Altera 公司的产品为例,介绍了 CPLD/FPGA 器件的结构、配置方法、下载电路和 Quartus II 等 EDA 软件的使用方法。

本书的特点是内容先进、方法实用、易读易懂、实践性强,能够使读者快速入门,逐步掌握 Verilog HDL 和 CPLD/FPGA 的基础知识、设计流程和建模方法,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。

本书可用作高等学校电气信息类专业高年级本、专科生的教材或教学参考书,也可以作为电子线路课程设计、电子设计大赛、电子系统设计工程技术人员学习 EDA 技术的参考书。

序

随着微电子技术和半导体工艺的迅速发展,电子器件不断朝着高速、大规模、小型的方向发展。近年来,电子器件更新换代更可谓日新月异。因此,加快集成电路领域人才的培养是高等教育改革和发展的一项重要任务。

专用集成电路(Application-Specific IC, ASIC)技术是在集微电子技术、半导体集成电路技术发展的基础上,结合电路与系统的设计方法发展起来的一种电路系统集成化的方法。当今,日趋进步和完善的 ASIC 技术正在逐步取代传统的电子设计方法而成为现代电子设计技术的核心,ASIC 的应用已进入社会各行各业,给我们的生活和工作带来了日新月异的变化,例如,在移动通信、仪器仪表、自动控制系统等领域得到广泛的应用。作者从数字系统设计的角度,介绍了 ASIC 设计的基础知识,具体包括 HDL 语言、RTL 设计、逻辑仿真与综合、FPGA 验证等;以 EDA 技术为背景,以数字电路为基础,侧重于从电路与系统的角度深入研讨新形势下的电子设计技术。其特点概括如下:

(1) 内容先进。Verilog HDL 是一种较流行的硬件描述语言,在数字 ASIC 设计中占主导地位,而用可编程逻辑器件 PLD(这里特指 CPLD、FPGA)和 EDA 工具来进行 ASIC 设计是目前最为流行的方式之一。本书通过大量实例,深入浅出地介绍了 Verilog HDL 各种建模技术,同时还回顾了可编程逻辑器件的发展历程,并以 Altera 公司的产品为例,讨论了 PLD 器件的结构原理、器件配置方法以及 EDA 工具的使用方法。PLD 具有用户现场可编程的特性,不需要经过集成电路工艺线加工,也没有定制掩膜层或逻辑单元的麻烦,用户利用 EDA 设计工具对 PLD 器件进行编程,就能得到满足用户特定需求 ASIC 电路。因此它特别适合于从事电子系统设计的工程人员进行 ASIC 设计。

(2) 适应教学。着眼于实用,强调在逻辑综合环境使用 Verilog HDL 方法,略去了有关行为仿真语句的内容,能够使学生尽快地熟悉可综合为硬件电路的 HDL 语言的应用。学生可以在计算机上对所设计的电路进行仿真,也有机会在实验室把自己设计的逻辑配置到真实的芯片中,观察电路的实际行为,这有利于提高学生的学习兴趣与学习热情。全书概念清晰、逻辑性强,内容循序渐进,语言流畅,实例丰富,并备有思考题和实验,便于自学。

(3) 实践性强。本书将理论教学和实验研究结合起来,书中列举了大量的开发实例,并将 HDL 语言、EDA 软件和 CPLD/FPGA 器件结合起来进行实验,通过学习,读者能够用 Verilog HDL 对数字电路进行建模和模拟仿真,用 CPLD/FPGA 器件验证自己设计的电路,熟悉用 EDA 方法设计数字系统的流程和常用的设计技巧,真正做到学以致用。

(4) 在第 9 章以“异步串口通信及 UART 实现”为例,详细介绍了分层次、分模块的设计方法,并给出了各模块的设计代码与仿真过程,以便学生了解一个实际数字系统的完整设计过程。

在附录中还介绍了目前在 PC 机上较为流行的仿真工具和逻辑综合工具,以便开阔学生的视野。

2008 年 2 月 20 日于华中科技大学

前 言

随着电子技术的发展,数字电路系统正朝着速度快、容量大、体积小、重量轻的方向发展。推动该潮流迅猛发展的引擎就是日趋完善的专用集成电路 ASIC (Application Specific Integrated Circuit) 设计技术。ASIC 通常包括三种:全定制、半定制(含标准单元和门阵列)及可编程逻辑器件(包括简单的 SPLD、复杂的 CPLD 和 FPGA)。对于前两种,由于生产中需要支付一次性工程费用 NRE (Non Recurring Engineering), 所以小批量生产、实验项目等不可能采用这两种方法实现。因此,可编程逻辑器件已成为实现 ASIC 的主要手段,在数字电路系统原型(指研发的产品样机或者试验性产品)的实现和 ASIC 原型的验证方面发挥了重要作用。

可编程逻辑器件具有保密性强、体积小、重量轻、可靠性高等 ASIC 的共同优点,同时它还具有现场可编程的特性。与普通的集成电路相比,它还具有设计开发周期短、设计制造成本低、开发工具先进等优点,因而现在被广泛地应用在数字通信、图形图像、仪表、兵器等系统中。

目前,硬件描述语言在可编程逻辑器件开发中发挥着越来越重要的作用,符合 IEEE 标准的硬件描述语言有 VHDL 和 Verilog HDL。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 作为电路设计语言。

本书按照“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例与实践”的思路编写。全书共 10 章,分为 4 个部分:第一部分在简述 EDA 技术的发展情况之后,重点介绍了 Verilog HDL 基础知识与建模方法,对可以综合成为硬件电路的语法结构和语句进行重点介绍,对只能用于仿真的语句介绍得较少,对状态机的建模方法进行了深入讨论;第二部分以 Altera 公司的 CPLD/FPGA 为背景,介绍了典型器件的结构原理、性能特点及器件的配置技术;第三部分介绍 Quartus II 软件的使用方法;第四部分是应用实例与实践,通过大量的开发实例介绍 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到复杂模块的设计技术。

本书的特点是:内容先进、方法实用、易读易懂、实践性强,有助于读者逐步掌握 Verilog HDL 和 CPLD/FPGA 的基础知识、设计流程和建模方法,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。本书还安排了一定数量的设计性实验,实验内容易到难,先安排了一些常用单元电路的设计实验,然后逐步过渡到复杂数字电路系统的实验,并要求读者将 HDL 代码“写”入 FPGA 芯片,对设计的电路进行在系统测试,以便掌握 FPGA 开发的整个流程。在每章末均安排了思考题与习题,便于自学。

本书由罗杰主编。参加编写的还有肖看、谭力、龚军和柯志武等老师,研究生曾炼、刘芳、刘红亮、田世坤等也参加了部分章节的编写和代码的验证工作。在编写本书的过程中,参考了许多同行专家的专著和文章,得到华中科技大学“新世纪教学改革工程”教材建设基金资助,还得到康华光教授、瞿安连教授和谢自美教授的热情支持和鼓励,在此表示衷心的感谢。

在本书出版之际,感谢华中科技大学国家电工电子教学基地老师们的支持和帮助,同时对 Altera 公司大学计划部经理徐平波(Bob Xu)先生的大力支持也深表谢意。

由于知识水平有限,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。可以通过 Luojiwh@gmail.com 给作者发送电子邮件。作者会认真阅读所有来信,并尽可能及时给予回复。

编 者

2007 年 12 月

目 录

第 1 章 概述	(1)
1.1 EDA 技术的发展	(1)
1.2 硬件描述语言	(2)
1.2.1 HDL 的发展	(2)
1.2.2 HDL 的内容与特点	(3)
1.3 可编程逻辑器件与专用集成电路	(4)
1.3.1 可编程逻辑器件	(4)
1.3.2 专用集成电路	(5)
1.3.3 基于 IP 核复用技术的 SOC 芯片	(6)
1.3.4 FPGA/CPLD 与 ASIC 设计流程	(7)
1.4 数字系统设计方法	(9)
1.4.1 数字系统的组成	(9)
1.4.2 自下而上的设计方法	(10)
1.4.3 自上而下的设计方法	(11)
1.5 EDA 设计工具的选择	(12)
思考题和习题一	(13)
第 2 章 Verilog HDL 基础	(14)
2.1 Verilog HDL 的基本语法规则	(14)
2.1.1 词法规定	(14)
2.1.2 逻辑值集合	(15)
2.1.3 常量及其表示	(15)
2.1.4 变量的数据类型	(17)
2.2 Verilog HDL 运算符	(21)
2.2.1 算术运算符	(22)
2.2.2 相等与全等运算符	(22)
2.2.3 逻辑运算符	(23)
2.2.4 位运算符	(23)
2.2.5 缩位运算符	(24)
2.2.6 位拼接运算符	(24)
2.2.7 运算符的优先级别	(24)
2.3 Verilog HDL 程序的基本结构	(25)
2.3.1 Verilog HDL 程序的基本结构	(25)
2.3.2 简单 Verilog HDL 程序实例	(26)
2.4 逻辑功能的仿真与测试	(28)
2.4.1 使用 Quartus II 软件进行仿真	(28)

2.4.2 使用 ModelSim 软件进行仿真	(29)
思考题和习题二	(30)
第 3 章 Verilog HDL 常用建模方式	(31)
3.1 Verilog HDL 结构级建模	(31)
3.1.1 多输入门	(31)
3.1.2 多输出门	(32)
3.1.3 三态门	(33)
3.1.4 门级建模设计举例	(34)
3.1.5 分层次的电路设计方法简介	(35)
3.2 Verilog HDL 数据流建模	(36)
3.2.1 数据流建模的基本语句	(37)
3.2.2 数据流建模举例	(37)
3.3 Verilog HDL 行为级建模	(38)
3.3.1 行为级建模的基本语句	(38)
3.3.2 触发器和移位寄存器的建模	(45)
3.3.3 计数器的建模	(48)
3.3.4 FIFO 的建模	(52)
3.4 Verilog HDL 函数与任务的使用	(54)
3.4.1 函数(function)说明语句	(54)
3.4.2 任务(task)说明语句	(56)
思考题和习题三	(58)
第 4 章 有限状态机设计	(61)
4.1 状态机的基本概念	(61)
4.1.1 状态机的基本结构及类型	(61)
4.1.2 状态机的状态图表示法	(62)
4.1.3 状态机的设计步骤	(63)
4.2 基于 Verilog HDL 的状态机描述方法	(63)
4.2.1 状态图的建立过程	(63)
4.2.2 推荐的状态图描述方法	(64)
4.3 状态机设计中的关键技术	(68)
4.3.1 状态编码	(68)
4.3.2 消除输出端产生的毛刺	(69)
4.3.3 使用 One-hot 编码方案设计状态机	(71)
4.4 状态机设计举例	(73)
4.4.1 汽车尾灯控制电路设计	(73)
4.4.2 十字路口交通灯控制电路设计	(76)
4.4.3 串行 D/A 转换器逻辑控制电路设计	(82)
思考题和习题四	(93)
第 5 章 Altera 公司的 CPLD/FPGA	(96)
5.1 可编程逻辑器件综述	(96)

5.1.1	PLD 发展简史	(96)
5.1.2	PLD 的表示方法	(98)
5.1.3	PLD 的与-或阵列结构	(99)
5.1.4	与-或阵列实现组合逻辑函数的原理	(100)
5.1.5	PLD 器件实现时序逻辑电路的基本原理	(101)
5.1.6	通用型 GAL 器件的基本结构	(101)
5.2	MAX7000S 系列器件结构	(103)
5.2.1	Altera 公司的产品简介	(103)
5.2.2	MAX7000S 系列器件结构	(106)
5.3	MAX3000A 系列器件结构	(110)
5.3.1	器件概述	(110)
5.3.2	MAX3000A 系列器件的结构特点	(111)
5.4	FLEX10K 系列器件结构	(112)
5.4.1	FPGA 实现逻辑函数的基本原理	(112)
5.4.2	FLEX10K 系列器件结构	(114)
5.5	Cyclone 系列器件结构	(119)
5.5.1	功能描述	(119)
5.5.2	Cyclone 系列器件结构	(120)
5.6	MAX II 系列器件	(122)
5.6.1	功能描述	(122)
5.6.2	MAX II 系列器件结构	(124)
5.7	FPGA 最小系统电路设计	(127)
5.7.1	电源电路和 LED 指示灯电路	(127)
5.7.2	复位电路	(128)
5.7.3	外部时钟振荡电路	(128)
5.7.4	FPGA 的下载配置电路	(128)
5.7.5	用户扩展接口	(129)
5.7.6	FPGA 芯片与各模块的接口电路	(131)
	思考题和习题五	(133)
第 6 章	Altera FPGA 器件的配置	(134)
6.1	配置方式与配置过程	(134)
6.1.1	配置方式	(134)
6.1.2	配置过程	(135)
6.2	配置接口电路	(136)
6.2.1	主动串行(AS)配置	(137)
6.2.2	被动串行(PS)配置	(139)
6.2.3	快速被动并行(FPP)配置	(141)
6.2.4	被动并行异步(PPA)配置	(142)
6.2.5	JTAG 配置	(142)
6.2.6	配置电路设计注意事项	(144)

6.3	FPGA 的配置存储器与 ByteBlaster II 下载电缆	(144)
6.3.1	FPGA 的配置存储器	(144)
6.3.2	ByteBlaster II 下载电缆的使用	(145)
6.3.3	ByteBlaster II 下载电缆的电路原理图	(147)
6.4	配置软件的设置与配置文件	(149)
6.4.1	配置模式与配置存储器的选择	(149)
6.4.2	复用配置引脚的处理	(150)
6.4.3	通用配置选项	(150)
6.4.4	配置文件	(151)
	思考题和习题六	(154)
第 7 章	Quartus II 6.0 软件的使用	(155)
7.1	设计流程概述	(155)
7.2	设计与仿真的过程	(156)
7.2.1	使用向导建立新工程	(156)
7.2.2	输入设计文件	(157)
7.2.3	编译前的设置	(158)
7.2.4	编译设计文件	(160)
7.2.5	设计项目的仿真验证	(161)
7.2.6	应用 RTL 观察器查看电路图	(164)
7.3	引脚分配与器件编程	(165)
7.3.1	引脚分配	(165)
7.3.2	器件编程	(166)
7.4	Altera 宏功能模块的使用	(168)
7.4.1	嵌入式锁相环 ALTPLL 宏功能模块的调用	(168)
7.4.2	嵌入式 LPM_FIFO 宏功能模块的调用	(173)
	思考题和习题七	(177)
第 8 章	数字电路与系统的设计实例	(178)
8.1	篮球竞赛 30 秒定时器设计与实现	(178)
8.1.1	定时器的功能要求	(178)
8.1.2	设计分析	(178)
8.1.3	逻辑设计	(179)
8.1.4	设计实现	(179)
8.2	多位 LED 显示器的动态扫描译码电路设计	(180)
8.2.1	功能要求	(180)
8.2.2	设计分析	(181)
8.2.3	逻辑设计	(181)
8.2.4	设计实现	(182)
8.3	CPLD 与矩阵式键盘接口电路的设计	(183)
8.3.1	接口电路的功能要求	(183)
8.3.2	接口电路的分析	(183)

8.3.3 接口电路的逻辑设计	(184)
8.4 多功能数字钟电路的分层次设计	(188)
8.4.1 数字钟的功能要求	(188)
8.4.2 设计分析	(188)
8.4.3 数字钟主体电路逻辑设计	(189)
8.4.4 功能扩展电路逻辑设计	(193)
8.4.5 多功能数字钟顶层电路设计	(195)
8.5 频率计的设计	(196)
8.5.1 频率计的功能要求	(196)
8.5.2 设计分析	(196)
8.5.3 频率计主体电路逻辑设计	(198)
8.5.4 频率计顶层电路设计	(203)
8.6 DDS 函数信号发生器的设计	(204)
8.6.1 DDS 的功能要求	(204)
8.6.2 设计分析	(204)
8.6.3 各主要模块的逻辑设计	(208)
8.6.4 DDS 函数信号发生器顶层设计	(213)
第 9 章 异步串口通信及 UART 实现	(215)
9.1 UART 接口实现原理	(215)
9.1.1 串行通信的概念	(215)
9.1.2 基本的 UART 通信协议	(216)
9.2 UART 接口模块的层次化设计	(217)
9.2.1 UART 接口的功能模块划分	(217)
9.2.2 配置文件 UART_INC. H	(218)
9.2.3 顶层模块的功能描述 UART. V	(219)
9.2.4 接收模块的功能描述 U_REC. V	(222)
9.2.5 发送模块的功能描述 U_XMIT. V	(226)
9.2.6 波特率变换模块的功能描述 BAUD. V	(230)
9.2.7 微处理器接口模块的功能描述	(231)
9.3 对 UART 接口模块的功能仿真	(239)
9.3.1 对接收模块的功能仿真	(239)
9.3.2 对发送模块的功能仿真	(241)
9.3.3 对波特率变换模块的功能仿真	(243)
9.3.4 对微处理器接口模块的功能仿真	(244)
9.3.5 对 UART 接口模块的功能仿真	(247)
9.4 逻辑综合与时序仿真	(251)
9.5 下载与验证测试	(251)
9.5.1 验证系统概述	(251)
9.5.2 验证结果	(252)

第 10 章 数字电路与数字系统实验	(253)
实验一 基本数字电路设计	(253)
实验二 跑马灯电路设计	(254)
实验三 有限状态机设计	(255)
实验四 多功能数字钟设计	(257)
实验五 FPGA 与 PC 串口通信实验	(258)
实验六 基于 FPGA 的 DDS 信号源设计	(259)
实验七 数字频率相位测量仪的设计	(260)
附录 A Verilog HDL 关键字	(262)
附录 B 常用 EDA 软件使用指南	(263)
B.1 仿真软件 ModelSim 6.0 的使用	(263)
B.1.1 设计与仿真流程	(263)
B.1.2 对层次化设计的仿真	(268)
B.1.3 布线后的时序仿真	(272)
B.2 逻辑综合软件 LeonardoSpectrum 的使用	(274)
附录 C Altera DE2 开发板的使用说明	(279)
C.1 Altera DE2 开发板的结构与测试	(279)
C.1.1 DE2 开发板的结构	(279)
C.1.2 DE2 开发板的测试	(280)
C.2 DE2 开发板上 FPGA 的引脚与其他资源的连接	(281)
C.3 FPGA 芯片 EP2C35 的配置方法	(284)
参考文献	(286)

第 1 章

概 述

随着电子设计自动化(Electronic Design Automation,简称 EDA)技术的不断普及和广泛应用,传统的“标准集成芯片+连线”的设计方法正在逐步退出历史舞台,而基于专用集成电路(Application Specific Integrated Circuit,简称 ASIC)的设计方法正在成为电子系统设计的主流。在这种设计方法中,可编程逻辑器件和硬件描述语言发挥了重要作用。本章将从 EDA 技术的发展和数字系统设计方法方面对目前这一领域的发展状况进行简要介绍。

1.1 EDA 技术的发展

EDA 技术是伴随计算机技术和微电子技术发展起来的一门新技术,大致可以分为 CAD(Computer Aided Design,计算机辅助设计)、CAE(Computer Aided Engineering,计算机辅助工程)和 EDA 三个发展阶段。

CAD 技术始于 20 世纪 70 年代初,当时电子设计主要以 SSI、MSI 标准元件的 PCB 设计构造电子系统。由于手工布设 PCB 和 IC 版图劳动强度大、效率低,人们开始在产品的设计过程的末端把高度重复性的繁杂劳动(如布图)借助于计算机软件工具完成,于是出现了 CAD 软件工具。当时,受计算机平台的制约,CAD 工具支持的设计工作有限,性能也较差,只有一些拥有大型计算机的公司开发了一些能做逻辑仿真工作的 CAD 软件工具。

到了 20 世纪 80 年代初期,CAD 技术进入支持设计过程的分析阶段,称为 CAE 阶段,出现了以仿真(逻辑模拟、定时分析和故障仿真)和自动布局布线为核心的 EDA 工具。到了 20 世纪 80 年代后期,CAD 技术在支持单元库的基础上,已能提供逻辑图输入、逻辑综合和优化、版图设计和编辑、设计结果验证等工作。

但是,上述的设计方法都是在原理图的基础之上进行的。随着电子产品复杂程度的增加和市场竞争的加剧,人们希望缩短产品的开发周期,这样一来,原理图设计方法已经不能满足设计的需求了。因为从原理图出发的 CAD 工具的设计受单纯从图形提取“信息”的约束,而且具体化的元件图形是对优化设计的一种约束,所以原理图设计成了电子系统设计自动化的“瓶颈”,于是 EDA 技术应运而生。

EDA 技术的基本特征是:采用硬件描述语言 HDL(Hardware Description Language)作为设计输入,具有系统级仿真和综合能力。硬件描述语言就是一种描述硬件电路的功能、信号连接关系及时序关系的语言。人们利用这种语言可以从上层到下层(从抽象到具体)逐层描述自己的设计思想,用一系列分层次的模块来表示极其复杂的数字系统;接着,利用 EDA 工具,逐层进行仿真验证,把其中需要变为实际电路的模块组合,经过自动综合工具转换成门级电路网表;然后,用专用集成电路 ASIC 或现场可编程门阵列 FPGA 自动布局布线工具,把网表转换为要实现的具体电路布线结构。目前,EDA 设计方法已经非常成熟,设计过程的各个阶段:系统级设计—行为级综合—功能模拟—逻辑综合—时序模拟—版图综合—后仿真等,都有许

多优秀的软件工具支撑。

综上所述,EDA 技术就是以大规模集成电路设计为载体,以硬件描述语言为主要表达方式,以计算机为设计环境,利用软件开发工具自动完成设计系统的逻辑编译、逻辑化简、逻辑综合、逻辑仿真、布局布线,直至完成对特定芯片的适配、映射、编程下载,最终将设计系统集成到特定的芯片中,完成专用集成电路芯片的设计。随着电子系统向集成化、大规模和高速度的方向发展,HDL 语言必将成为电子系统硬件设计工程师必须掌握的语言。

1.2 硬件描述语言

1.2.1 HDL 的发展

自从 20 世纪 70 年代 TTL(Transistor-Transistor Logic,晶体管-晶体管逻辑)标准化数字集成电路面市以来,数字电路设计便进入了所谓逻辑门层次设计(Gate-Level Design)时代。工程师能以便宜的价格买到标准的逻辑门元件,进行电路的组装及验证。从此,逻辑元件设计与数字电路设计分了家。逻辑元件设计者,可专心致力于提供性能更加优异的集成电路,而数字电路工程师则专注在数字电路的逻辑分析与设计方面。

随着集成电路制造技术的进步,集成电路向大规模、高密度、高速度的方向发展,电子系统的设计愈来愈复杂,人们希望运用计算机强大的运算能力来协助人们设计电路和分析电路,于是各种计算机辅助设计等工具应运而生,它们使得数字电路的设计进入了高层次设计的新时代——采用硬件描述语言设计数字电路与系统。

早在 1980 年,美国国防部为方便管理有关武器承包商的电子电路技术文件,使其遵循统一的设计描述界面,以便在将来有新技术推出时,仍能重复使用原设计,因而发展了名为 VHSIC(Very High Speed Integrated Circuit,超高速集成电路)的计划。最初,其目的是希望能制定一个标准的文件格式及语法,各武器承包商及其分包商,均须遵循此统一标准语法格式描述设计的电路。最终目的是希望创造出下一代高速集成电路的设计界面,以期能突破各种大型集成电路在设计上的不便。

1981 年 6 月成立了 VHDL(VHSIC Hardware Description Language,超高速集成电路硬件描述语言)小组。1983 年第三季度,由 IBM 公司、TI 公司、Intermetric 公司成立开发小组。1986 年 IEEE 标准化组织开始讨论 VHDL 语言标准。1987 年 IEEE 接受 VHDL 为标准 HDL,即 IEEE STD 1076—1987 标准。1993 年进一步修订,定为 ANSI/IEEE STD 1076—1993 标准。

与此同时,一家以提供 EDA 工具为主业的公司 GDA(Gateway Design Automation),于 1983 发表了一种相似的硬件描述语言 Verilog HDL(简称 Verilog)。由于 GDA 了解到当时 ASIC 半导体厂需要电路模拟器(Simulator),以便对复杂的数字电路进行模拟验证,所以 Verilog HDL 在发展之初,便试图以程序语言界面(Programming Language Interface, PLI)为基础,创造一个以计算机辅助设计为导向的自动化电路设计环境。

在 1984—1985 年间,GDA 公司的 Phil Moorby 设计出了第一个关于 Verilog_XL 的仿真器。1986 年他提出了用于门级仿真的快速算法并获得成功。1987 年,Synopsys 公司发表了第一个以 Verilog HDL 为界面的数字电路综合工具之后,提高了数字电路设计的效率。1989

年,Cadence 公司收购了 GDA 公司,并于 1990 年公开发表了 Verilog HDL,成立了 OVI(Open Verilog International)组织来负责 Verilog HDL 的发展。1992 年 OVI 决定致力于推广 Verilog HDL 成为 IEEE 标准,并于 1995 年成为 IEEE 标准,即 IEEE STD 1364—1995。Verilog 的增强版本是 Verilog 2001,于 2001 年正式被批准为 IEEE 标准,称为 1364—2001。它支持原始 Verilog 版本的所有特性,并引入了一些新特性。

目前,这两种硬件描述语言都已经被指定为 IEEE(电气与电子工程师协会)标准。很多 EDA 供应商都把 VHDL 和 Verilog HDL 作为其 EDA 软件输入/输出的标准。例如,Cadence、Synopsys 等厂商都提供了对 VHDL 和 Verilog HDL 的支持。这两种硬件描述语言除了语法上的差异及限制之外,支持两者的 EDA 工具所提供的功能几乎是相同的。就目前版本的 Verilog HDL 和 VHDL 而言,一般认为 Verilog HDL 在系统级的抽象建模方面比 VHDL 略差一些,而在门级开关电路建模方面则比 VHDL 强得多。因此,比较哪一种语言更好是没有任何实际意义的。

Verilog HDL 是一种非常类似 C 的语言,因此它受到了学过 C 语言的工程师的欢迎。VHDL 是一种类似 Ada 的语言,但大多数的工程师都没有使用 Ada 语言的经验,掌握 VHDL 设计技术难度较大。

事实上,除了 VHDL 和 Verilog HDL 两种标准的硬件描述语言外,在 20 世纪 80 年代许多高校、科研单位和大型计算机厂商都相继推出了各自的 HDL 语言,例如,M、BLM、UDL/I、ABEL、PALASM、CUPL、DSL、AHDL 等,但都未能得到推广。近几年,还出现了 System C 和 Superlog 等新的硬件描述语言,它们能够支持数字系统的软件和硬件的协同设计,比 Verilog HDL 和 VHDL 语言支持的抽象级别高。

1.2.2 HDL 的内容与特点

硬件描述语言(HDL)是为描述数字系统的行为而设计且经过优化的一种编程语言。它是硬件电路设计人员与电子设计自动化(EDA)工具之间沟通的桥梁,其主要目的是用来编写设计文件、建立电子系统行为级的仿真模型,对用 Verilog HDL 或 VHDL 建模的复杂数字逻辑进行仿真,然后利用逻辑综合工具自动生成符合要求且在电路结构上可以实现的数字逻辑网表(Netlist)。根据网表和某种工艺进行版图设计,然后生成该工艺条件下电路的延时模型,仿真验证无误后用于制造 ASIC 芯片或者写入可编程逻辑器件中。

HDL 的主要特点如下:

- ① HDL 支持数字电路的设计、验证、综合和测试,但不支持模拟电路的描述。
- ② HDL 既包含一些高级程序设计语言的结构形式,同时也兼顾描述硬件电路连接的具体构件。
- ③ HDL 是并发的,即具有在同一时刻执行多任务的能力。一般来讲,程序设计语言是串行的,但在实际硬件中许多操作都是在同一时刻发生的,所以 HDL 语言具有并发的特征。
- ④ HDL 有时序概念。一般地说,程序设计语言是没有什么时序概念的,但在硬件电路中,从输入到输出总是有延时存在的。为描述这些特征,HDL 需要建立时序的概念,因此使用 HDL 除了可以描述硬件电路的功能外,还可以描述其时序关系。

1.3 可编程逻辑器件与专用集成电路

早期电子系统硬件设计采用分立器件,随着集成电路的出现和应用,人们选用功能固定的标准集成电路(例如各种逻辑门、编码器、译码器、触发器、寄存器和计数器等)构成硬件系统,后来,又以微处理器为核心构成系统,现在则广泛采用专用集成电路(ASIC)来构成系统,一个复杂的数字系统只要一片或几片 ASIC 即可实现。

制作 ASIC 的方法大体可分为两种,一种是掩膜处理^[1]方法,由半导体厂家制造;另一种是使用现场可编程逻辑器件实现,用户通过计算机和 EDA 开发工具,将所设计的电路或系统“编程”到芯片上,就可以得到一块专用集成电路。

1.3.1 可编程逻辑器件

可编程逻辑器件 PLD(Programmable Logic Device)就是一种可以由用户定义和设置逻辑功能的数字集成电路,属于可编程 ASIC。PLD 的种类较多,目前广泛使用的 PLD 器件主要是复杂可编程逻辑器件 CPLD(Complex Programmable Logic Device)和现场可编程门阵列 FPGA(Field Programmable Gate Array)。

CPLD 器件内部含有许多逻辑块和连线资源,而逻辑块由与-或阵列和触发器等构成,逻辑块的功能由用户编程决定。逻辑块之间、逻辑块与芯片外部可以通过可编程的连线资源实现信息交换。通常 CPLD 器件采用 CMOS E²PROM 工艺制作,当用户的逻辑写入后,即使断电也不会丢失。部分 CPLD 内部还集成了 E²PROM、FIFO 或双口 RAM,以适应不同功能的数字系统设计。

FPGA 是另一类可编程逻辑器件,在结构上与 CPLD 有很大差别,电路设计不受与-或阵列结构的两级组合逻辑限制。芯片内部主要由许多可编程逻辑模块组成,靠纵横交错的分布式可编程互连线连接起来,可构成极其复杂的逻辑电路。它更适合于实现多级逻辑功能,并且具有更高的集成密度和应用灵活性。目前,主要有基于 CMOS SRAM 工艺制造的 FPGA 和基于反熔丝工艺制造的 FPGA 两种类型。由于 SRAM 中的数据理论上可以进行无限次写入,所以,基于 SRAM 技术的 FPGA 可以进行无限次编程。但是,SRAM 具有数据易失性的特点,即一旦断电,其原有的逻辑功能将消失,所以在使用这类 FPGA 时,需要一个外部的 PROM 保存编程数据,上电后,FPGA 首先从 PROM 中读入编程数据进行初始化,然后才开始正常工作。基于反熔丝技术的 FPGA 则只能编程一次,此类 FPGA 比较适合定型产品和大批量应用。

可编程逻辑器件是组成数字系统的理想器件,设计时只需通过定义器件内部的逻辑和输入/输出引脚,就可以实现各种逻辑功能,而且由于引脚定义的灵活性,大大减轻了电路图和电路板设计的工作量和难度,从而有效地增强了设计的灵活性,提高了工作效率。和 ASIC 相比,这种方法的缺点是单片成本较高,电路的性能受到 PLD 器件的限制,很难实现高性能或有

[1] 掩膜处理是 IC 的一种制作工艺,指在 IC 的不同制造阶段需要处理晶片上的不同位置,这就需要遮住其他的不需要处理的地方,称为不同的掩膜层。