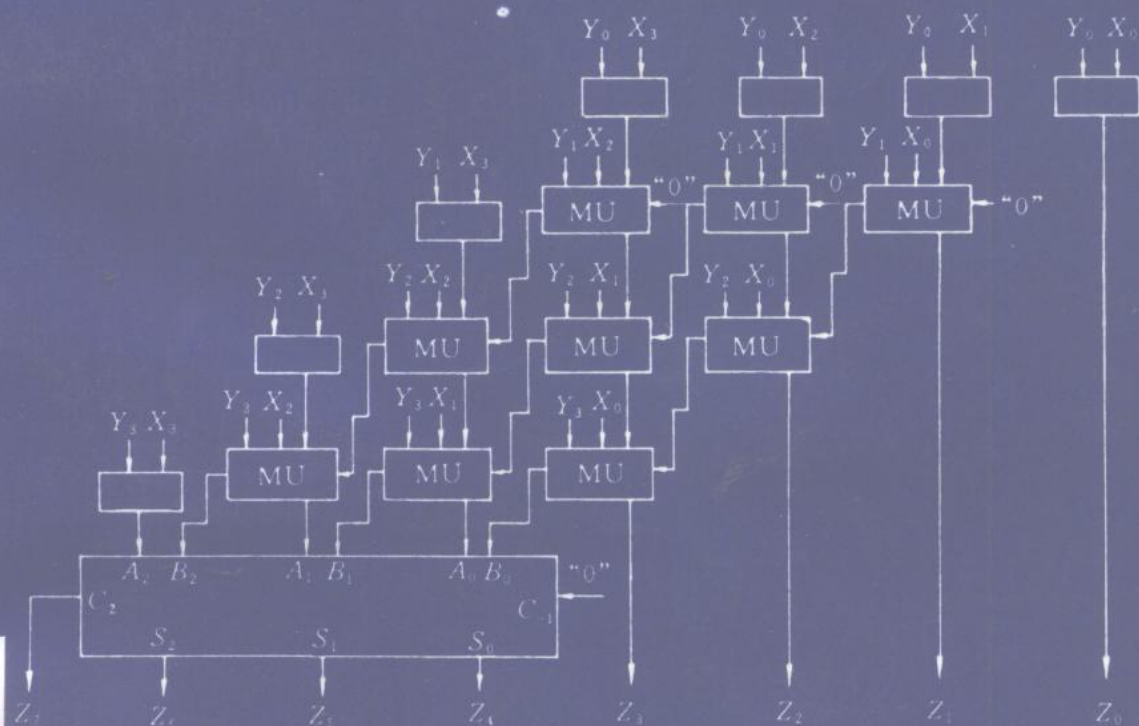


复杂数字电路与系统的 *Verilog HDL* 设计技术

夏宇闻 编著



北京航空航天大学出版社

7N79
X31

复杂数字电路与系统的 Verilog HDL 设计技术

夏宇闻 编著

北京航空航天大学出版社

内 容 提 要

本书讲述的是 90 年代才开始在美国和其他先进的工业国家逐步推广的利用硬件描述语言 (Verilog HDL) 设计复杂数字逻辑电路与系统的技术和方法。掌握了这种基本方法之后, 就可以设计极其复杂的硬线 (hard-wired) 数字逻辑电路与系统, 如实时数字信号处理 (DSP) 电路系统等。因为本书的内容是独立于开发环境的, 所以书中并不介绍具体工具的使用, 只介绍有关 Verilog HDL 建模、仿真、综合以及 TOP-DOWN 等现代设计思想、技术、方法和需要注意的要点。全书共分为六章, 第一章为 Verilog HDL 设计方法概述; 第二章介绍 Verilog HDL 的基本语法; 第三章介绍不同抽象级别的 Verilog HDL 模型; 第四章讲述有限状态机和可综合风格的 Verilog HDL; 第五章为可综合的 Verilog HDL 设计实例 (简化的 RISC CPU 设计简介); 第六章介绍虚拟器件和虚拟接口模块。书中各章都有大量的例题, 每章后还附有思考题, 可以帮助读者理解书中的基本概念并掌握设计从简单到非常复杂的各种风格模块的技术。本书面向的对象是大学电子类和计算机工程类本科高年级学生和研究生, 以及在专用数字电路与系统设计领域工作的工程师们。阅读本书所需的基础知识是数字电子技术基础和 C 语言编程基础知识。

图书在版编目 (CIP) 数据

复杂数字电路与系统的 VerilogHDL 设计技术/夏宇闻编
著. —北京: 北京航空航天大学出版社, 1998. 9

ISBN 7-81012-808-6

I. 复… II. 夏… III. ①数字电路-硬件描述语言, VerilogHDL-程序设计②逻辑电路-硬件描述语言, VerilogHDL-程序设计 IV. TN79

中国版本图书馆 CIP 数据核字 (98) 第 14357 号

复杂数字电路与系统的 Verilog HDL 设计技术

夏宇闻 编著

责任编辑 王海虹

责任校对 陈 坤

北京航空航天大学出版社出版发行

(北京市学院路 37 号, 邮编 100083, 发行部电话 62015720)

河北涿州市新华印刷厂印装

各地书店经销

*

开本: 787×1092 1/16 印张: 12.75 字数: 326 千字

1998 年 8 月第 1 版 1999 年 1 月第 2 次印刷 印数: 1 001~3 000 册

ISBN 7-81012-808-6 / TN·030 定价: 12.00 元

前 言

国内数字逻辑电路与系统的设计多年来一直采用传统的方法,即先在纸上画出真值表,做布尔代数化简,通过画波形图、有限状态机流程图、动态和静态卡诺图等方法来分析和综合电路,然后在实验板上用 CMOS 或 74 系列集成电路块构成数字逻辑电路。当电路比较复杂时设计和调试这样的实验系统需要花费很多的时间和精力。

二十年前国内开始使用微处理机,中规模集成电路的使用也逐步普及,大学里的电子和计算机类学科普遍开设了汇编语言课程和一些常用的中规模集成电路的使用课程。这些技术大大缩短了开发复杂专用数字系统所需的时间。

近十年来,国外先进工业国家由于计算机电路辅助设计技术和半导体集成工艺技术的快速进步,在生产的电子系统中,专用集成电路(ASIC)和 FPGA 的使用越来越多,特别在先进的电讯设备、计算机系统和网络设备中更是如此。这不仅因为有不少实时的 DSP(数字信号处理)芯片是一般微处理机所无法替代的,而且也因为市场对电子产品的要求越来越高。在电子设计和制造领域,我们与国外先进国家的技术差距越来越大。

作为一名在大学讲授复杂专用数字电路与系统设计课程的教师深深感到身上责任的重大。我个人觉得这与大学的课程设置和教学条件有关,因为我们没有及时把国外最先进的设计技术介绍给学生,也没有给他们创造实践的机会。

1992 年我受学校和系领导的委托,筹建世界银行贷款的电子设计自动化(EDA)实验室。其中 CADENCE 设计环境中数字设计部分由我负责。我们首先掌握了利用电路图输入的方法,并逐步掌握了利用 Verilog HDL 设计复杂数字电路的仿真和综合技术。在此基础上我们为有关单位设计了三万门左右的复杂数字电路,提供给他们经仿真验证的 Verilog HDL 源代码,为此得到很高的评价。我们也为自己的科研项目——小波(Wavelet)图像压缩设计了小波卷积器和改进的零修剪树(EZW)算法(即 SPIHT 算法)的硬线逻辑的 Verilog HDL 模型,并成功地进行了仿真和综合。

从 1994 年找到一些有关 Verilog HDL 的资料起,我就在研究生课程“复杂专用数字系统设计”中,逐步增加有关利用 Verilog HDL 进行复杂数字系统设计的内容。1996 年春,我受张凤言教授的邀请,在国家教委电路教学委员会召集的华北区讨论会上作了三个小时的有关 EDA 和 HDL 设计方法的讲座。会后,张凤言教授就一直鼓励我写一本有关 HDL 设计方法的书。现在,这本书终于出版了。因为我们使用 Verilog HDL 设计复杂数字逻辑电路总共也只有两年的时间,水平有限,书中谬误之处在所难免,敬请读者及时把意见反馈给我。之所以匆匆把这本书推出,是想把我们在采用 Verilog HDL 设计方法上积累的一些经验与读者分享,在大学生和研究生中加快 Verilog HDL 设计技术的推广,尽快培养出一批掌握先进设计技术的跨世纪人才。期望本书能在这一过程中起到抛砖引玉的作用。

本书是我们实验室全体老师和同学的共同劳动成果。EDA 实验室的研究生张琰、山岗、王静璇、田玉文和冯文楠等做了许多基础工作,如部分素材的翻译、录入和一些 Verilog HDL 模块的设计和验证。我只是收集了全书的素材,翻译和理解素材中一些较难的概念,最后对全书

文稿进行组织、整理和补充。实验室的董金明和杨惠军老师也给了我许多帮助和鼓励,特别是董金明老师一直以他自己努力工作的实际行动给我以最有力的鞭策,使我不能懈怠。在出版之际,我衷心地感谢在本书编写过程中所有给过我帮助和鼓励的老师和同学们!

编 者

1997 年 12 月于北京航空航天大学 EDA 实验室

电子邮箱: xyw@dept2.buaa.edu.cn

通信地址: 北京 100083 北京航空航天大学 205 信箱 夏宇闻

电话: 010-62017251-7253

目 录

第一章 Verilog HDL 设计方法概述	(1)
1.1 硬件描述语言(HDL)	(1)
1.2 Verilog HDL 的历史	(1)
1.2.1 什么是 Verilog HDL	(1)
1.2.2 Verilog HDL 的产生及发展	(2)
1.3 Verilog HDL 和 VHDL 的比较	(2)
1.4 Verilog HDL 目前的应用情况和适用的设计	(3)
1.5 采用 Verilog HDL 设计复杂数字电路的优点	(4)
1.5.1 传统设计方法——电路原理图输入法	(4)
1.5.2 Verilog HDL 输入法与传统的电路原理图输入法的比较	(4)
1.5.3 Verilog HDL 的标准化与软核的重用	(5)
1.5.4 软核、固核和硬核的概念以及它们的重用	(5)
1.6 Verilog HDL 的设计流程简介	(5)
1.6.1 自顶向下(TOP-DOWN)设计的基本概念	(5)
1.6.2 层次管理的基本概念	(6)
1.6.3 具体模块的设计编译和仿真的过程	(6)
1.6.4 对应具体工艺器件的优化、映象和布局布线	(7)
1.7 小 结	(7)
思考题	(8)
第二章 Verilog HDL 的基本语法	(9)
2.1 简单的 Verilog HDL 模块	(10)
2.1.1 简单的 Verilog HDL 程序介绍	(10)
2.1.2 模块的结构	(11)
2.1.3 模块的端口定义	(11)
2.1.4 模块内容	(11)
2.2 数据类型及其常量、变量	(12)
2.2.1 常 量	(13)
2.2.2 变 量	(15)
2.3 运算符及表达式	(18)
2.3.1 基本的算术运算符	(18)
2.3.2 位运算符	(19)
2.3.3 逻辑运算符	(20)
2.3.4 关系运算符	(21)
2.3.5 等式运算符	(21)

2.3.6	移位运算符	(22)
2.3.7	位拼接运算符	(22)
2.3.8	缩减运算符	(23)
2.3.9	优先级别	(23)
2.3.10	关键词	(24)
2.4	赋值语句和块语句	(24)
2.4.1	赋值语句	(24)
2.4.2	块语句	(26)
2.5	条件语句	(28)
2.5.1	if_else 语句	(28)
2.5.2	case 语句	(31)
2.5.3	使用条件语句不当生成锁存器的情况	(34)
2.6	循环语句	(35)
2.6.1	forever 语句	(35)
2.6.2	repeat 语句	(35)
2.6.3	while 语句	(36)
2.6.4	for 语句	(36)
2.7	结构说明语句	(38)
2.7.1	initial 语句	(38)
2.7.2	always 语句	(39)
2.7.3	task 和 function 说明语句	(39)
2.8	系统函数和任务	(43)
2.8.1	\$display 和 \$write 任务	(44)
2.8.2	系统任务 \$monitor	(47)
2.8.3	时间度量系统函数 \$time	(48)
2.8.4	系统任务 \$finish	(49)
2.8.5	系统任务 \$stop	(49)
2.8.6	系统任务 \$readmemb 和 \$readmemh	(49)
2.8.7	系统任务 \$random	(51)
2.9	编译预处理	(51)
2.9.1	宏定义 'define	(52)
2.9.2	“文件包含”处理 'include	(54)
2.9.3	时间尺度 'timescale	(56)
2.9.4	条件编译命令 'ifdef, 'else, 'endif	(58)
2.10	小结	(59)
	思考题	(59)
第三章	不同抽象级别的 Verilog HDL 模型	(71)
3.1	门级结构描述	(71)
3.1.1	与非门、或门和反向器等及其说明语法	(71)

3.1.2 用门级结构描述 D 触发器	(72)
3.1.3 由已经设计成的模块构成更高一层的模块	(72)
3.2 Verilog HDL 的行为描述建模	(74)
3.2.1 仅用于产生仿真测试信号的 Verilog HDL 行为描述建模	(74)
3.2.2 Verilog HDL 建模在 TOP-DOWN 设计中的作用和行为 建模的可综合性问题	(76)
3.3 用 Verilog HDL 建模进行 TOP-DOWN 设计的实例	(77)
3.4 小 结	(86)
思考题	(86)
第四章 有限状态机和可综合风格的 Verilog HDL	(87)
4.1 有限状态机	(87)
4.1.1 用 Verilog HDL 语言设计可综合的状态机的指导原则	(92)
4.1.2 典型的状态机实例	(93)
4.1.3 综合的一般原则	(94)
4.1.4 语言指导原则	(95)
4.2 可综合风格的 Verilog HDL 模块实例	(96)
4.2.1 组合逻辑电路设计实例	(96)
4.2.2 时序逻辑电路设计实例	(101)
4.2.3 状态机的置位与复位	(103)
4.2.4 复杂时序逻辑电路设计实践	(106)
第五章 可综合的 Verilog HDL 设计实例——简化的 RISC_CPU 设计简介	(137)
5.1 什么是 CPU	(137)
5.2 RISC_CPU 的结构	(138)
5.2.1 时钟发生器	(138)
5.2.2 指令寄存器	(141)
5.2.3 累加器	(142)
5.2.4 算术运算器	(143)
5.2.5 数据控制器	(144)
5.2.6 地址多路器	(145)
5.2.7 程序计数器	(145)
5.2.8 状态控制器	(146)
5.2.9 外围模块	(152)
5.3 RISC_CPU 的操作和时序	(153)
5.3.1 系统的复位和启动操作	(153)
5.3.2 总线读操作	(153)
5.3.3 写总线操作	(155)
5.4 RISC_CPU 的寻址方式和指令系统	(157)
5.5 RISC_CPU 模块的调试	(157)
5.5.1 RISC_CPU 模块的前仿真	(157)

5.5.2 RISC_CPU 模块的综合	(169)
5.5.3 RISC_CPU 模块的优化和布局布线	(170)
思考题.....	(172)
第六章 虚拟器件和虚拟接口模型	(173)
6.1 虚拟器件和虚拟接口模块的供应商	(173)
6.2 虚拟接口模块的实例	(174)
参考文献	(196)

第一章 Verilog HDL 设计方法概述

随着电子设计技术的飞速发展,专用集成电路(ASIC)和用户现场可编程门阵列(FPGA)的复杂度越来越高,数字通信、工业自动化控制等领域所用的数字电路及系统的复杂程度也越来越高。设计这样复杂的电路及系统已不再是简单的个人劳动,而需要综合许多专家的经验 and 知识才能够完成。在数字逻辑设计领域,迫切需要一种共同的工业标准来统一对数字逻辑电路及系统的描述,把专家们设计的各种常用数字逻辑电路和系统部件建成宏单元(Megcell)或软核(Soft-Core)库供设计者引用,以减少重复劳动,提高工作效率。

VHDL 和 Verilog HDL 这两种工业标准的产生顺应了历史的潮流,因而得到了迅速的发展。作为跨世纪的中国大学生应该尽早掌握这种新的设计方法,成为我国 21 世纪深亚微米百万门级的复杂数字逻辑电路及系统设计的技术骨干,使我国在复杂数字电路及系统设计的竞争中逐步缩小与美国等工业发达国家的差距。

1.1 硬件描述语言(HDL)

硬件描述语言(HDL——Hardware Description Language)是一种用形式化方法来描述数字电路和设计数字逻辑系统的语言。数字逻辑电路设计者可以利用这种语言来描述自己的设计思想,然后利用电子设计自动化(下面简称为 EDA)工具进行仿真,再自动综合到门级电路,最后用 ASIC 或 FPGA 实现其功能。目前,这种被称为高层次设计(High-Level-Design)的方法已被广泛采用。据统计,在美国硅谷目前约有 80% 的 ASIC 和 FPGA 已采用硬件描述语言进行设计。

硬件描述语言发展至今已有二十多年的历史,并成功地应用于设计的各个阶段:仿真、验证、综合等。到 80 年代,已出现了上百种硬件描述语言,它们对设计自动化起到了极大的促进作用。但是,这些语言一般各自面向特定的设计领域与层次,而且众多的语言使用户无所适从,因此急需一种面向设计的多领域、多层次并得到普遍认同的标准硬件描述语言。进入 80 年代后期,硬件描述语言向着标准化的方向发展。最终,VHDL 和 Verilog HDL 语言适应了这种趋势的要求,先后成为 IEEE 标准。

1.2 Verilog HDL 的历史

1.2.1 什么是 Verilog HDL

Verilog HDL 是硬件描述语言的一种,用于数字电子系统设计。设计者可用它进行各种级别的逻辑设计,可用它进行数字逻辑系统的仿真验证、时序分析和逻辑综合。它是目前应用最广泛的一种硬件描述语言。据有关文献报道,目前在美国使用 Verilog HDL 进行设计的工程师大约有 15 000 人,预计到 2000 年将发展到 60 000 人。

1.2.2 Verilog HDL 的产生及发展

Verilog HDL 是在 1983 年,由 GDA (GateWay Design Automation) 公司的 Phil Moorby 首创的。Phil Moorby 后来成为 Verilog-XL 的主要设计者和 Cadence 公司 (Cadence Design System) 的第一个合伙人。在 1984—1985 年间, Moorby 设计出了第一个关于 Verilog-XL 的仿真器; 1986 年, 他对 Verilog HDL 的发展又作出了另一个巨大贡献——提出了用于快速门级仿真的 XL 算法。

随着 Verilog-XL 算法的成功, Verilog HDL 语言得到迅速发展。1989 年, Cadence 公司收购了 GDA 公司, Verilog HDL 语言成为 Cadence 公司的私有财产。1990 年, Cadence 公司决定公开发表 Verilog HDL 语言, 于是成立了 OVI (Open Verilog International) 组织来负责 Verilog HDL 语言的发展。基于 Verilog HDL 的优越性, IEEE 于 1995 年制定了 Verilog HDL 的 IEEE 标准, 即 Verilog HDL1364—1995。

图 1.1 显示出 Verilog 的发展历史和未来。

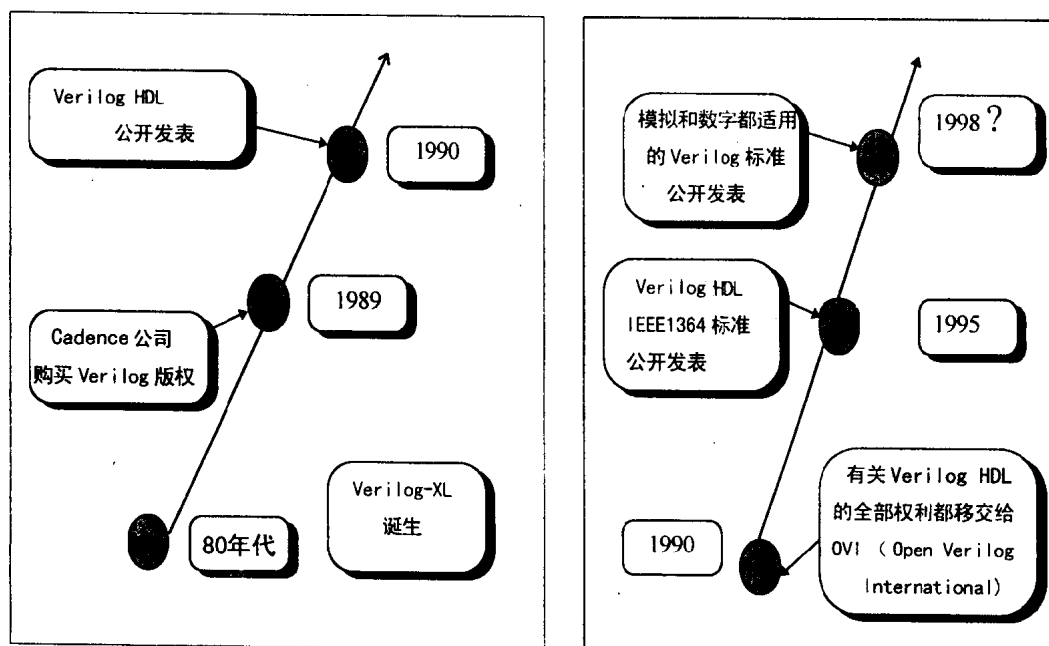


图 1.1 Verilog HDL 的发展历史和未来

1.3 Verilog HDL 和 VHDL 的比较

Verilog HDL 和 VHDL 都是用于逻辑设计的硬件描述语言, 并且都已成为 IEEE 标准。VHDL 是在 1987 年成为 IEEE 标准的, Verilog HDL 则在 1995 年才正式成为 IEEE 标准。之所以 VHDL 比 Verilog HDL 早成为 IEEE 标准, 这是因为 VHDL 是美国军方组织开发的, 而 Verilog HDL 则是从一个普通民间公司的私有财产转化而来, 基于 Verilog HDL 的优越性, 才成为 IEEE 标准, 因而有更强的生命力。

VHDL 的英文全名为 VHSIC Hardware Description Language, 而 VHSIC 则是 Very High Speed Integrated Circuit 的缩写词, 意为甚高速集成电路, 故 VHDL 准确的中文译名为甚高速集成电路的硬件描述语言。

Verilog HDL 和 VHDL 作为描述硬件电路设计的语言, 其共同的特点在于: 能形式化地抽象表示电路的结构和行为; 支持逻辑设计中层次与领域的描述; 可借用高级语言的精巧结构来简化电路的描述; 具有电路仿真与验证机制以保证设计的正确性; 支持电路描述由高层到低层的综合转换; 硬件描述与实现工艺无关(有关工艺参数可通过语言提供的属性包括进去); 便于文档管理; 易于理解 and 设计重用。

但是 Verilog HDL 和 VHDL 又各有其自己的特点。由于 Verilog HDL 早在 1983 年就已推出, 至今已有 14 年的应用历史, 因而 Verilog HDL 拥有更广泛的设计群体, 资源也远比 VHDL 丰富。与 VHDL 相比, Verilog HDL 的最大优点为: 它是一种非常容易掌握的硬件描述语言, 只要有 C 语言的编程基础, 通过 20 学时的学习, 再加上实际操作, 一般可在 2~3 个月内掌握这种设计技术。而掌握 VHDL 设计技术就比较困难, 因为 VHDL 不很直观, 需要有 Ada 编程基础, 一般认为至少需要半年以上的专业培训, 才能掌握 VHDL 的基本设计技术。目前版本的 Verilog HDL 和 VHDL 在行为级抽象建模的覆盖范围方面也有所不同。一般认为 Verilog HDL 在系统级抽象方面比 VHDL 略差一些, 而在门级开关电路描述方面比 VHDL 强得多。图 1.2 为 Verilog HDL 和 VHDL 建模能力的比较, 供读者参考。

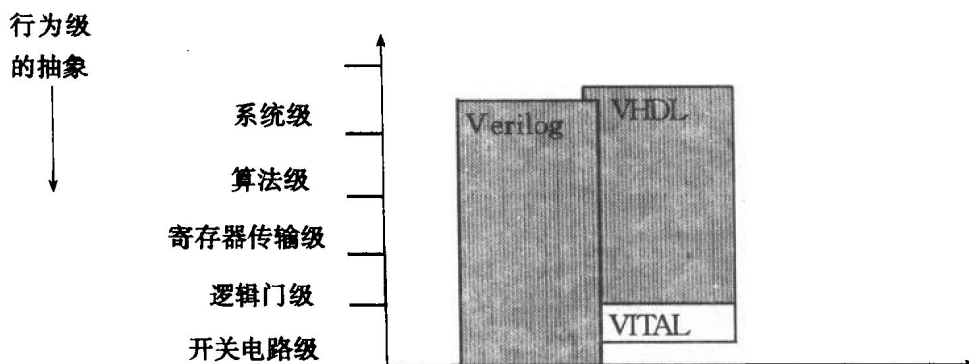


图 1.2 Verilog HDL 与 VHDL 建模能力的比较

但这两种语言也在不断地完善, 因此, Verilog HDL 作为学习 HDL 设计方法的入门和基础是比较合适的。学习掌握 Verilog HDL 建模、仿真和综合技术不仅可以对数字电路设计技术有更进一步的了解, 而且可以为学习高级的系统综合打下坚实的基础。

1.4 Verilog HDL 目前的应用情况和适用的设计

几年以来, EDA 界一直争论不休的两种硬件描述语言已初见端倪。在美国, 高层逻辑电路设计领域中 Verilog HDL 和 VHDL 的应用比率是 60% 和 40%; 在我国台湾地区各为 50%; 在我国大陆目前由于 Verilog HDL 和 VHDL 的使用才刚刚开始, 具体应用比率还没有统计。Verilog HDL 是专门为复杂数字逻辑电路和系统的设计仿真而开发的, 本身就非常适合复杂

数字逻辑电路和系统的仿真和综合。由于 Verilog HDL 在其门级描述的底层,也就是在晶体管开关的描述方面比 VHDL 有更强的功能,所以即使是 VHDL 的设计环境,在底层实质上也是由 Verilog HDL 描述的器件库所支持的。另外,目前 Verilog HDL-A 标准还支持模拟电路的描述,1998 年即将通过的 Verilog HDL 新标准,将把 Verilog HDL-A 并入 Verilog HDL 新标准,使其不仅支持数字逻辑电路的描述还支持模拟电路的描述,因此在混合信号电路系统的设计中,它必将会有更广泛的应用。在亚微米和深亚微米 ASIC 及高密度 FPGA 已成为电子设计主流的今天,Verilog HDL 的发展前景是非常大的。笔者本人的意见是,若要推广采用硬件描述语言的设计方法,则应首先从推广 Verilog HDL 开始,然后再推广 VHDL。

Verilog HDL 较为适合系统级(System)、算法级(Algorithem)、寄存器传输级(RTL)、门级(Gate)、开关级(Switch)设计;而对于特大型(几百万门级以上)的系统级(System)设计,VHDL 则更为适合。由于这两种 HDL 语言还在发展过程中,它们都会逐步地完善自己。

1.5 采用 Verilog HDL 设计复杂数字电路的优点

1.5.1 传统设计方法——电路原理图输入法

几十年前,当时所做的复杂数字逻辑电路及系统的设计规模比较小也比较简单,其中所用到的 FPGA 或 ASIC 设计工作往往只能采用厂家提供的专用电路图输入工具来进行。为了满足设计性能指标,工程师往往需要花费几天或更长的时间进行艰苦的手工布线。工程师还得非常熟悉所选器件的内部结构和外部引线特点,才能达到设计要求。这种低水平的设计方法延长了设计周期。

近年来,FPGA 和 ASIC 的设计规模不断扩大,复杂度也越来越高,而对逻辑电路及系统的设计时间要求却越来越短。这些因素促使设计人员采用高水准的设计工具,如硬件描述语言(Verilog HDL 或 VHDL)来进行设计。

1.5.2 Verilog HDL 输入法与传统的电路原理图输入法的比较

如 1.5.1 节所述,采用电路原理图输入法进行设计具有设计周期长、需要专门的设计工具、需手工布线等缺陷。而采用 Verilog HDL 输入法时,由于 Verilog HDL 的标准化,可以很容易地把完成的设计移植到不同厂家的不同芯片中去,并在不同规模应用时可以较容易地作修改。这是因为用 Verilog HDL 所完成的设计,其信号位数容易改变,可以很方便地对其进行修改,以适应不同规模的应用;在仿真验证时,仿真测试矢量还可以用同一种描述语言来完成;另外,采用 Verilog HDL 综合器生成的数字逻辑是一种标准的电子设计互换格式(EDIF)文件,独立于所采用的实现工艺。有关工艺参数的描述可以通过 Verilog HDL 提供的属性包括进去,然后利用不同厂家的布局布线工具,在不同工艺的芯片上实现。

采用 Verilog HDL 输入法最大的优点是其与工艺无关性。这使得工程师在功能设计、逻辑验证阶段,可以不必过多考虑门级及工艺实现的具体细节,只需要利用系统设计时对芯片的要求,施加不同的约束条件,即可设计出实际电路。实际上这是利用了计算机的巨大能力并在 EDA 工具的帮助下,把逻辑验证与具体工艺库匹配、布线及时延计算分成不同的阶段来实现,从而减轻了人们的繁琐劳动。

1.5.3 Verilog HDL 的标准化与软核的重用

Verilog HDL 是在 1983 年由 GDA 公司首先开发成功的,经过诸多改进,于 1995 年 11 月正式被批准为 IEEE 标准 1364。

Verilog HDL 的标准化大大加快了 Verilog HDL 的推广和发展。由于 Verilog HDL 设计方法的与工艺无关性,大大提高了 Verilog HDL 模型的可重用性。我们把功能经过验证的、可综合的、实现后电路结构总门数在 5 000 门以上的 Verilog HDL 模型称之为“软核”(Soft Core),而把由软核构成的器件称为虚拟器件。在新电路的研制过程中,软核和虚拟器件可以很容易地借助 EDA 综合工具与其他外部逻辑结合为一体。这样,软核和虚拟器件的可重用性就可大大缩短设计周期,加快了复杂电路的设计。目前国际上有一个叫作“虚拟接口联盟”(Virtual Socket Interface Alliance)的组织来协调这方面的工作。

1.5.4 软核、固核和硬核的概念以及它们的重用

1.5.3 节中已介绍了软核的概念,下面再介绍一下固核(Firm Core)和硬核(Hard Core)的概念。

我们把在某一种现场可编程门阵列(FPGA)器件上实现的、经验证是正确的、总门数在 5 000 门以上电路结构编码文件称之为“固核”。

把在某一种专用半导体集成电路工艺的(ASIC)器件上实现的、经验证是正确的、总门数在 5 000 门以上的电路结构掩膜称之为“硬核”。

显而易见,在具体实现手段和工艺技术尚未确定的逻辑设计阶段,软核具有最大的灵活性,它可以很容易地借助 EDA 综合工具与其他外部逻辑结合为一体。当然,由于实现技术的不确定性,有可能要作一些改动以适应相应的工艺。相比之下,固核和硬核与其他外部逻辑结合为一体的灵活性要差得多,特别是电路实现工艺技术改变时更是如此。而近年来电路实现工艺技术的发展是相当迅速的,为了逻辑电路设计成果的积累和更快更好地设计更大规模的电路,发展软核的设计和推广软核的重用技术是非常必要的。新一代的数字逻辑电路设计师必须掌握这方面的知识和技术。

1.6 Verilog HDL 的设计流程简介

1.6.1 自顶向下(TOP-DOWN)设计的基本概念

现代集成电路制造工艺技术的改进,使得在一个芯片上集成数十万乃至数百万个器件成为可能,但很难设想仅由一个设计师独立设计如此大规模的电路而不出现错误。利用层次化、结构化的设计方法,一个完整的硬件设计任务首先由总设计师划分为若干个可操作的模块,编制出相应的模型(行为的或结构的),通过仿真加以验证后,再把这些模块分配给下一层的设计师。这就允许多个设计者同时设计一个硬件系统中的不同模块,其中每个设计者负责自己所承担的部分,而由上一层设计师对其下层设计者完成的设计用行为级上层模块进行验证。图 1.3 为自顶向下(TOP-DOWN)的示意图,以设计树的形式绘出。

自顶向下的设计是从系统级开始,把系统划分为基本单元,然后再把每个基本单元划分为下一层次的基本单元,一直这样做下去,直到可以直接用 EDA 元件库中的元件来实现为止。

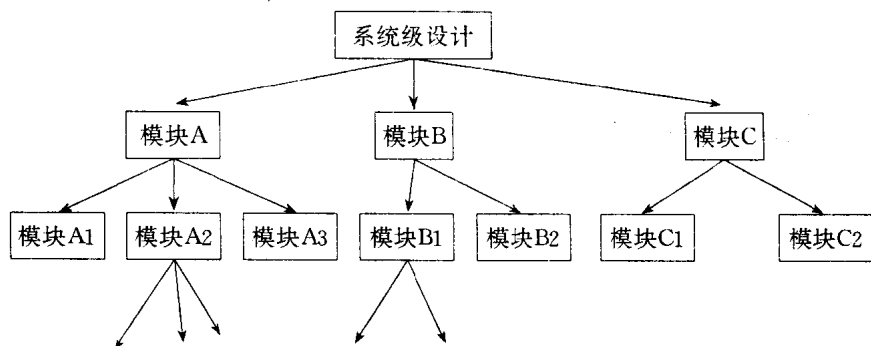


图 1.3 TOP-DOWN 设计思想

对于设计开发整机电子产品的单位和个人来说,新产品的开发总是从系统设计入手,进行方案的总体论证、功能描述、任务和指标的分配。随着系统变得复杂和庞大,特别需要在样机问世之前,对产品的全貌有一定的预见性。目前,EDA 技术的发展使得设计师有可能实现真正的自顶向下的设计。

1.6.2 层次管理的基本概念

复杂数字逻辑电路和系统的层次化、结构化设计隐含着硬件设计方案的逐次分解。在设计过程中的任意层次,硬件至少有一种描述形式。硬件的描述特别是行为描述通常称为行为建模。在集成电路设计的每一层次,硬件可以分为一些模块,该层次的硬件结构由这些模块的互连描述,该层次的硬件行为由这些模块的行为描述。这些模块称为该层次的基本单元,而该层次的基本单元又由下一层次的基本单元互连而成,如此下去,完整的硬件设计就可以由图 1.3 所示的设计树描述。在这个设计树上,节点对应着该层次上基本单元的行为描述,树枝对应着基本单元的结构分解。在不同的层次都可以进行仿真以对设计思想进行验证。EDA 工具提供了有效的手段来管理错综复杂的层次,即可以很方便地查看某一层某模块的源代码或电路图以改正仿真时发现的错误。

1.6.3 具体模块的设计编译和仿真的过程

在不同层次做具体模块的设计所用的方法有所不同。在高层次上往往需要编写一些行为级的模块,通过仿真加以验证,其主要目的是对系统性能的总体考虑和各模块的指标分配,并非具体电路的实现,因而综合及其以后的步骤往往不需进行。而当设计的层次比较接近底层时,行为描述往往需要用电路逻辑来实现,这时的模块不仅需要通过仿真加以验证,还需进行综合、优化、布线和后仿真。总之,具体电路是从底向上逐步实现的。EDA 工具往往不仅支持 HDL 描述也支持电路图输入,有效地利用这两种方法是提高设计效率的办法之一。图 1.4 简要地说明了模块的编译和测试过程。

从图 1.4 可以看出,模块设计流程主要由两大主要功能部分组成。

(1) 设计开发:即从编写设计文件→综合→布局布线→投片生成这样一系列步骤。

(2) 设计验证:也就是进行各种仿真的一系列步骤,如果在仿真过程中发现问题就返回设计输入进行修改。

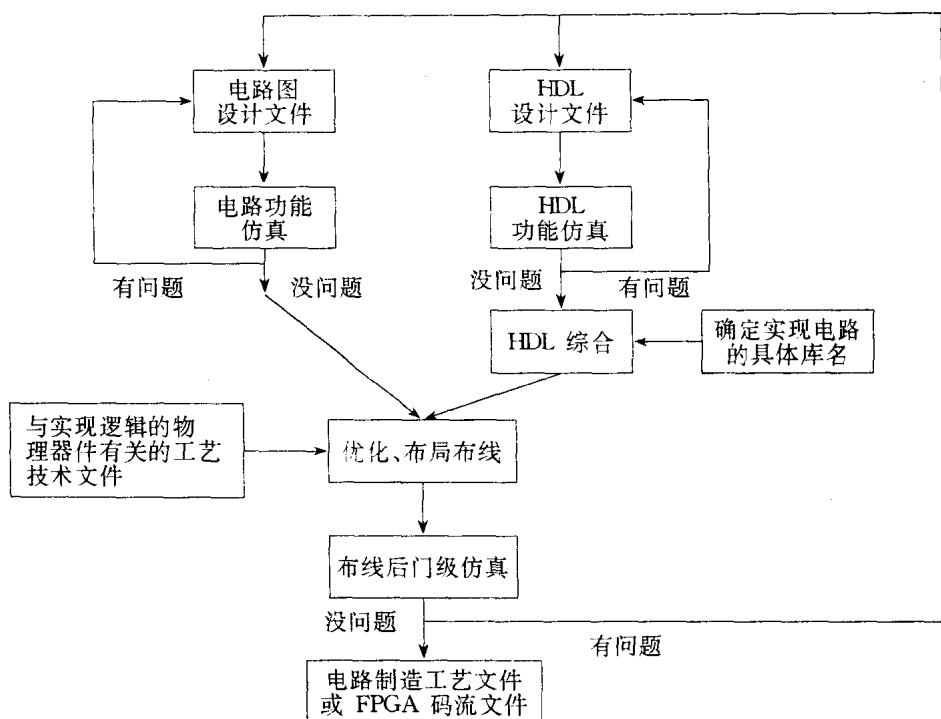


图 1.4 HDL 设计流程图

1.6.4 对应具体工艺器件的优化、映象和布局布线

由于各种 ASIC 和 FPGA 器件的工艺各不相同,因而当用不同厂家的不同器件来实现已验证的逻辑网表(EDIF 文件)时,需要不同的基本单元库与布线延迟模型与之对应才能进行准确的优化、映象和布局布线。基本单元库与布线延迟模型由熟悉本厂工艺的工程师提供,再由 EDA 厂商的工程师编入相应的处理程序,而逻辑电路设计师只需用一文件说明所用的工艺器件和约束条件,EDA 工具就会自动地根据这一文件选择相应的库和模型进行准确的处理,从而大大提高设计效率。

1.7 小结

采用 Verilog HDL 设计方法比采用电路图输入的方法更有优越性,这就是为什么美国等先进工业国家在进入 90 年代以后纷纷采用 HDL 设计方法的原因。在两种符合 IEEE 标准的硬件描述语言中,Verilog HDL 与 VHDL 相比,更加基础、更易学习。掌握 HDL 设计方法应从学习 Verilog HDL 设计方法开始。Verilog HDL 可用于复杂数字逻辑电路和系统的总体仿真、子系统仿真和具体电路综合等各个设计阶段。

由于 TOP-DOWN 的设计方法是首先从系统设计入手的,因而从顶层进行功能划分和结构设计。系统的总体仿真是顶层进行功能划分的重要环节,这时的设计是与工艺无关的。由于设计的主要仿真和调试过程是在高层次完成的,所以能够早期发现结构设计上的错误,避免设计工作的浪费,同时也减少了逻辑仿真的工作量。自顶向下的设计方法方便了从系统级划分和管理整个项目,使得几十万门甚至几百万门规模的复杂数字电路的设计成为可能,并可减少设

计人员,避免不必要的重复设计,提高了设计的一次成功率。

从底向上的设计在某种意义上讲可以看作上述 TOP-DOWN 设计的逆过程。虽然设计也是从系统级开始,即从设计树的树根开始对设计进行逐次划分,但划分时首先考虑的是单元是否存在,即设计划分过程必须从存在的基本单元出发。设计树最末枝上的单元要是已经制造出的单元,要是其他项目已开发好的单元或者是可外购得到的单元。

自顶向下的设计过程中,在每一层次划分时都要对某些目标作优化, TOP-DOWN 的设计过程是理想的设计过程,它的缺点是得到的最小单元不标准,制造成本可能很高。从底向上的设计过程全部采用标准基本单元,通常比较经济,但有时可能不能满足一些特定的指标要求。复杂数字逻辑电路和系统的设计过程通常是这两种设计方法的结合,设计时需要考虑多个目标的综合平衡。

思考题

1. 什么是硬件描述语言?它的主要作用是什么?
2. 目前世界上符合 IEEE 标准的硬件描述语言有哪两种?它们各有什么特点?
3. 什么情况下需要采用硬件描述语言的设计方法?
4. 采用硬件描述语言设计方法的优点是什么?有什么缺点?
5. 简单叙述一下利用 EDA 工具并采用硬件描述语言的设计方法和流程。
6. 硬件描述语言可以用哪两种方式参与复杂数字电路的设计?
7. 用硬件描述语言设计的数字系统需要经过哪些步骤才能与具体的电路相对应?
8. 为什么说用硬件描述语言设计的数字逻辑系统具有最大的灵活性,可以映射到任何工艺的电路路上?
9. 软核是什么?虚拟器件是什么?它们的作用是什么?
10. 固核是什么?硬核是什么?与软核相比它们各有什么优缺点?
11. 简述 TOP-DOWN 设计方法和硬件描述语言的关系。