



普通高等教育“十二五”规划教材

◎ 电子信息科学与工程类专业 规划教材

EDA技术与应用

(第4版)

◎ 江国强 编著



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

电子信息科学与工程类专业规划教材

EDA 技术与应用

(第4版)

江国强 编著

电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书分为7章,包括EDA技术概述、EDA工具软件的使用方法、VHDL、Verilog HDL、常用EDA工具软件、可编程逻辑器件和EDA技术的应用。另外,附录部分介绍了伟福EDA6000和友晶DE2等EDA实验开发系统的使用方法,供具有不同实验设备的读者学习或参考。

EDA是当今世界上最先进的电子电路设计技术,它的重要作用逐步被我国的产业界、科技界和教育界认可。本书可作为高等院校工科电子类、通信信息类、自动化类专业“EDA技术”课程的教材,也可供相关专业的技术人员参考。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

EDA技术与应用/江国强编著. —4版. —北京:电子工业出版社,2013.1

电子信息科学与工程类专业规划教材

ISBN 978-7-121-19315-6

I. ①E… II. ①江… III. ①电子电路—电路设计—计算机辅助设计—高等学校—教材 IV. ①TN702

中国版本图书馆CIP数据核字(2012)第309787号

责任编辑:凌 毅 特约编辑:张 莉

印 刷:北京市李史山胶印厂

装 订:北京市李史山胶印厂

出版发行:电子工业出版社

北京市海淀区万寿路173信箱 邮编:100036

开 本:787×1092 1/16 印张:20.5 字数:525千字

印 次:2013年1月第1次印刷

印 数:4000册 定价:38.00元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010)88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010)88258888。

第4版前言

在20世纪90年代,国际上电子和计算机技术先进的国家,一直在积极探索新的电子电路设计方法,在设计方法、工具等方面进行了彻底的变革,并取得巨大成功。在电子设计技术领域,可编程逻辑器件(如CPLD、FPGA)的应用,已得到很好的普及,这些器件为数字系统的设计带来了极大的灵活性。由于可编程逻辑器件可以通过软件编程而对其硬件结构和工作方式进行重构,使得硬件的设计可以如同软件设计那样方便快捷。这一切极大地改变了传统的数字系统设计方法、设计过程和设计观念,促进了EDA技术的迅速发展。

EDA是电子设计自动化(Electronic Design Automation)的缩写,在20世纪90年代初从计算机辅助设计(CAD)、计算机辅助制造(CAM)、计算机辅助测试(CAT)和计算机辅助工程(CAE)的概念发展而来的。EDA技术是以计算机为工具,设计者在EDA软件平台上,用硬件描述语言(HDL)完成设计文件,然后由计算机自动地完成逻辑编译、化简、分割、综合、优化、布局、布线、仿真,直至对于特定目标芯片的适配编译、逻辑映射和编程下载等工作。EDA技术的出现,极大地提高了电路设计的效率和可靠性,减轻了设计者的劳动强度。

本书是在《EDA技术与应用》(第3版)基础上编写的,跟随EDA技术的发展,本书将各种最新版本的EDA工具软件的使用方法展示给读者,包括Quartus II 12.0及Qsys、ModelSim-Altera 10.0d、Matlab R2012a等。

本书共7章。第1章EDA技术概述,介绍EDA技术的发展、EDA设计流程及EDA技术涉及的领域。

第2章介绍EDA工具软件的使用方法。EDA的核心是利用计算机完成电路设计的全程自动化,因此基于计算机环境下的EDA工具软件是不可缺少的。利用EDA技术进行电路设计的大部分工作是在EDA开发设计平台上进行的,离开了EDA工具,电路设计的自动化是不可能实现的。因此,掌握EDA工具软件的使用方法,应该是EDA技术学习的第一步。至今Altera已公布了Quartus II 12.0版本的EDA软件,本章以Altera公司的Quartus II 12.0为主介绍EDA工具软件的使用方法。

第3章和第4章分别介绍VHDL和Verilog HDL两种常用硬件描述语言的基础知识。VHDL和Verilog HDL作为IEEE标准的硬件描述语言,经过二十多年的发展、应用和完善,它们以其强大的系统描述能力、规范的程序设计结构、灵活的语言表达风格和多层次的仿真测试手段,在电子设计领域受到了普遍的认同和广泛的接受,成为现代EDA领域的首选硬件描述语言。专家认为,在本世纪VHDL与Verilog HDL语言将承担起几乎全部的数字系统设计任务。

第5章介绍几种目前世界上最流行和实用的EDA工具软件,包括ModelSim、Quartus II 9.0版本的使用方法、Matlab、Nios II和Qsys,以适应不同读者的需要。这些软件主要是基于PC平台的,面向PLD、SOPC和ASIC设计,比较适合学校教学、项目开发和相关的科研。自从Quartus II 10.0版本出现后,已经取消了自带的仿真工具,采用第三方软件ModelSim进行设计仿真,考虑到部分设计者对Quartus II 9.0(及以前)版本的自带仿真编辑工具情有独钟,因此本章仍然介绍Altera公司的Quartus II 9.0的使用方法,作为各种版本使用方法的补充。

第6章介绍PLA、PAL、GAL、EPLD和FPGA等各种类型可编程逻辑器件的电路结构、

工作原理、使用方法、编程方法和 Altera 公司可编程逻辑器件。

第 7 章介绍 EDA 技术在组合逻辑、时序逻辑电路设计, 以及基于 EDA 技术的数字系统设计中的应用。

为了方便读者能较系统和较完整地学习 EDA 技术, 本书从教学的目的出发, 尽量将有关 EDA 技术的内容编入书中, 并力求内容精练, 语言通俗易懂。读者可以根据实际需要, 节选学习书中的部分内容, 尽快掌握 EDA 基本技术, 然后通过相关 EDA 技术书籍的学习, 达到精通 EDA 技术的目的。

本书的教学可安排 32 学时, 其中第 1 章占 2 学时, 第 2 章占 4 学时, 第 3 章占 8 学时, 第 4 章占 8 学时, 第 5 章作为选学内容 (需 6~8 学时), 第 6 章占 2 学时, 第 7 章占 8 学时。另外, 还需要安排 4~8 学时的实验, 第 1 个实验安排 EDA 工具软件的使用方法, 其余的实验可安排 HDL 的编程实验。

本书提供配套的电子课件, 可登录华信教育资源网: www.hxedu.com.cn, 注册后免费下载。

本书由桂林电子科技大学信息科技学院的江国强编著, 对于书中的错误和不足之处, 恳请读者指正。

江国强

2012 年 12 月

目 录

第 1 章 EDA 技术概述	1
1.1 EDA 技术及发展	1
1.2 EDA 设计流程	2
1.2.1 设计准备	2
1.2.2 设计输入	3
1.2.3 设计处理	3
1.2.4 设计校验	4
1.2.5 器件编程	4
1.2.6 器件测试和设计验证	5
1.3 硬件描述语言	5
1.3.1 VHDL	5
1.3.2 Verilog HDL	6
1.3.3 AHDL	6
1.4 可编程逻辑器件	6
1.5 常用 EDA 工具	7
1.5.1 设计输入编辑器	7
1.5.2 仿真器	8
1.5.3 HDL 综合器	8
1.5.4 适配器 (布局布线器)	8
1.5.5 下载器 (编程器)	9
本章小结	9
思考题和习题	9
第 2 章 EDA 工具软件的使用方法	10
2.1 Quartus II 软件的主界面	10
2.2 Quartus II 的图形编辑输入法	12
2.2.1 编辑输入图形设计文件	12
2.2.2 编译设计文件	16
2.2.3 仿真设计文件	18
2.2.4 编程下载设计文件	22
2.3 Quartus II 宏功能模块的使用方法	28
2.3.1 设计原理	28
2.3.2 编辑输入顶层设计文件	28
2.3.3 仿真顶层设计文件	34
2.3.4 图形文件的转换	35
2.4 嵌入式逻辑分析仪的使用方法	37

2.4.1	打开 SignalTap II 编辑窗口	37
2.4.2	调入节点信号	38
2.4.3	参数设置	39
2.4.4	文件存盘	39
2.4.5	编译与下载	39
2.4.6	运行分析	39
2.5	嵌入式锁相环的设计方法	40
2.5.1	嵌入式锁相环的设计	40
2.5.2	嵌入式锁相环的仿真	43
2.5.3	使用嵌入式逻辑分析仪观察嵌入式锁相环的设计结果	43
2.6	设计优化	44
2.6.1	面积与速度的优化	44
2.6.2	时序约束与选项设置	45
2.6.3	Fitter 设置	45
2.7	Quartus II 的 RTL 阅读器	45
	本章小结	46
	思考题和习题	47
第 3 章	VHDL	48
3.1	VHDL 设计实体的基本结构	48
3.1.1	库、程序包	49
3.1.2	实体	49
3.1.3	结构体	50
3.1.4	配置	50
3.1.5	基本逻辑器件的 VHDL 描述	51
3.2	VHDL 语言要素	54
3.2.1	VHDL 文字规则	54
3.2.2	VHDL 数据对象	56
3.2.3	VHDL 数据类型	57
3.2.4	VHDL 的预定义数据类型	57
3.2.5	IEEE 预定义的标准逻辑位和矢量	59
3.2.6	用户自定义数据类型方式	59
3.2.7	VHDL 操作符	59
3.2.8	VHDL 的属性	62
3.3	VHDL 的顺序语句	63
3.3.1	赋值语句	64
3.3.2	流程控制语句	64
3.3.3	WAIT 语句	70
3.3.4	ASSERT (断言) 语句	71
3.3.5	NULL (空操作) 语句	71
3.4	并行语句	71

3.4.1	PROCESS (进程) 语句	72
3.4.2	块语句	73
3.4.3	并行信号赋值语句	74
3.4.4	子程序和并行过程调用语句	76
3.4.5	元件例化 (COMPONENT) 语句	78
3.4.6	生成语句	80
3.5	VHDL 的库和程序包	82
3.5.1	VHDL 库	82
3.5.2	VHDL 程序包	83
3.6	VHDL 设计流程	84
3.6.1	编辑 VHDL 源程序	84
3.6.2	设计 8 位计数显示译码电路顶层文件	86
3.6.3	编译顶层设计文件	87
3.6.4	仿真顶层设计文件	87
3.6.5	下载顶层设计文件	88
3.7	VHDL 仿真	88
3.7.1	VHDL 仿真支持语句	88
3.7.2	VHDL 测试平台软件的设计	90
	本章小结	94
	思考题和习题	94
第 4 章	Verilog HDL	97
4.1	Verilog HDL 设计模块的基本结构	97
4.1.1	模块端口定义	97
4.1.2	模块内容	98
4.2	Verilog HDL 的词法	100
4.2.1	空白符和注释	100
4.2.2	常数	100
4.2.3	字符串	101
4.2.4	关键词	101
4.2.5	标识符	101
4.2.6	操作符	101
4.2.7	Verilog HDL 数据对象	105
4.3	Verilog HDL 的语句	107
4.3.1	赋值语句	107
4.3.2	条件语句	108
4.3.3	循环语句	111
4.3.4	结构声明语句	113
4.3.5	语句的顺序执行与并行执行	116
4.4	不同抽象级别的 Verilog HDL 模型	117
4.4.1	Verilog HDL 的门级描述	118

4.4.2	Verilog HDL 的行为级描述	118
4.4.3	用结构描述实现电路系统设计	120
4.5	Verilog HDL 设计流程	122
4.5.1	编辑 Verilog HDL 源程序	122
4.5.2	设计 BCD 加法器电路顶层文件	124
4.5.3	编译顶层设计文件	125
4.5.4	仿真顶层设计文件	125
4.5.5	下载顶层设计文件	125
4.6	Verilog HDL 仿真	125
4.6.1	Verilog HDL 仿真支持语句	125
4.6.2	Verilog HDL 测试平台软件的设计	128
	本章小结	131
	思考题和习题	132
第 5 章	常用 EDA 工具软件	134
5.1	ModelSim	134
5.1.1	ModelSim 的图形用户交互方式	134
5.1.2	ModelSim 的交互命令方式	138
5.1.3	ModelSim 的批处理工作方式	140
5.1.4	ModelSim 与 Quartus II 的接口	141
5.2	Quartus II 9.0 软件的使用方法	142
5.2.1	Quartus II 软件的安装	142
5.2.2	Quartus II 软件的主界面	144
5.2.3	Quartus II 的仿真方法	145
5.2.4	Quartus II 宏功能模块的使用方法	149
5.2.5	在 Quartus II 9.0 中使用 ModelSim 仿真	151
5.3	基于 Matlab/DSP Builder 的 DSP 模块设计	156
5.3.1	设计原理	157
5.3.2	建立 Matlab 设计模型	157
5.3.3	Matlab 模型仿真	162
5.3.4	Signal Compiler 使用方法	164
5.3.5	使用 ModelSim 仿真	166
5.3.6	硬件实现与测试	167
5.3.7	DSP Builder 的层次设计	169
5.4	Nios II 嵌入式系统开发软件	169
5.4.1	Nios II 的硬件开发	170
5.4.2	调整 SDRAM 地址	177
5.4.3	生成 Nios II 硬件系统	177
5.4.4	Nios II 系统的调试	193
5.4.5	Nios II 的常用组件与编程	198
5.4.6	SOPC 的应用	206

5.5	Qsys 系统集成软件	216
5.5.1	Qsys 的硬件开发	216
5.5.2	Qsys 系统的编译与下载	220
5.5.3	Qsys 系统的调试	222
	本章小结	224
	思考题和习题	225
第 6 章	可编程逻辑器件	226
6.1	PLD 的基本原理	226
6.1.1	PLD 的分类	226
6.1.2	阵列型 PLD	229
6.1.3	现场可编程门阵列 FPGA	232
6.1.4	基于查找表 (LUT) 的结构	234
6.2	PLD 的设计技术	236
6.2.1	PLD 的设计方法	236
6.2.2	在系统可编程技术	237
6.2.3	边界扫描技术	240
6.3	PLD 的编程与配置	240
6.3.1	CPLD 的 ISP 方式编程	241
6.3.2	使用 PC 的并口配置 FPGA	241
6.4	Altera 公司的 PLD 系列产品简介	243
6.4.1	Altera 高端 Stratix FPGA 系列	243
6.4.2	Altera 中端 FPGA 的 Arria 系列	244
6.4.3	Altera 低成本 FPGA 的 Cyclone 系列	245
6.4.4	Altera 低成本 CPLD 的 MAX 系列	245
6.4.5	Altera 硬件拷贝 HardCopy ASIC 系列	246
	本章小结	246
	思考题和习题	247
第 7 章	EDA 技术的应用	248
7.1	组合逻辑电路设计应用	248
7.1.1	运算电路设计	248
7.1.2	编码器设计	249
7.1.3	译码器设计	251
7.1.4	数据选择器设计	253
7.1.5	数据比较器设计	254
7.1.6	ROM 的设计	256
7.2	时序逻辑电路设计应用	258
7.2.1	触发器设计	258
7.2.2	锁存器设计	260
7.2.3	移位寄存器设计	261
7.2.4	计数器设计	263

7.2.5 随机读写存储器 RAM 的设计	265
7.3 基于 EDA 的数字系统设计	267
7.3.1 计时器的设计	267
7.3.2 万年历的设计	271
7.3.3 8 位十进制频率计设计	275
本章小结	281
思考题和习题	282
附录 A EDA6000 实验开发系统	284
A.1 EDA6000 的特点	284
A.2 EDA6000 的使用方法	285
A.2.1 EDA6000 的硬件结构	285
A.2.2 EDA6000 软件平台的使用方法	288
附录 B Altera DE2 开发板使用方法	298
B.1 Altera DE2 开发板的结构	298
B.2 DE2 开发板的实验模式与目标芯片的引脚连接	298
B.3 DE2 开发板实验的操作	303
B.3.1 编辑	303
B.3.2 编译	306
B.3.3 仿真	306
B.3.4 引脚锁定	306
B.3.5 编程下载	307
B.3.6 硬件验证	308
B.4 DE2 开发板的控制嵌板	308
B.4.1 打开控制嵌板	308
B.4.2 设备检测	308
附录 C Quartus II 的宏函数和强函数	310
C.1 宏函数	310
C.2 强函数	316
参考文献	318

第 1 章 EDA 技术概述

本章概要：本章介绍 EDA 技术的发展、EDA 设计流程及 EDA 技术涉及的领域。

知识要点：(1) EDA 设计流程；

(2) 设计处理包含的过程；

(3) “自顶向下”的设计流程；

(4) EDA 工具各模块的主要功能。

教学安排：本章教学安排 2 学时，重点让读者熟悉 EDA 设计“自顶向下”的流程，了解 EDA 工具各模块的主要功能。

1.1 EDA 技术及发展

20 世纪末，数字电子技术的飞速发展，有力地推动了社会生产力的发展和社会信息化的提高。目前，数字电子技术的应用已经渗透到人类生活的各个方面。从计算机到手机，从数字电话到数字电视，从家用电器到军用设备，从工业自动化到航天技术，都尽可能采用了数字电子技术。

微电子技术，即大规模集成电路加工技术的进步是现代数字电子技术发展的基础。目前，在硅片的单位面积上集成的晶体管数量越来越多，1978 年推出的 8086 微处理器芯片集成的晶体管数是 4 万只，到 2000 年推出的 Pentium 4 微处理器芯片的集成度上升到 4200 万只晶体管，2005 年生产可编程逻辑器件 (PLD) 的集成度达到 5 亿只晶体管，包含的逻辑元件 (Logic Elements, LEs) 有 18 万个，2009 年生产的 PLD 中的 LEs 达到 84 万个，集成度达到 25 亿只晶体管，2011 年生产的 PLD 中的 LEs 达到 95.2 万个。原来需要成千上万只电子元器件组成的计算机主板或彩色电视机电路，现在仅用一片或几片超大规模集成电路就可以代替，现代集成电路已经能够实现单片电子系统 SOC (System On a Chip) 的功能。

现代电子设计技术的核心是 EDA (Electronic Design Automation) 技术。EDA 技术就是依靠功能强大的电子计算机，在 EDA 软件工具平台上，对以硬件描述语言 HDL (Hardware Description Language) 为系统逻辑描述手段完成的设计文件，自动地完成逻辑编译、化简、分割、综合、优化、仿真，直至下载到可编程逻辑器件 CPLD/FPGA 或专用集成电路 ASIC (Application Specific Integrated Circuit) 芯片中，实现既定的电子电路设计功能。EDA 技术使得电子电路设计者的工作仅限于利用硬件描述语言和 EDA 软件平台来完成对系统硬件功能的实现，极大地提高了设计效率，缩短了设计周期，节省了设计成本。

EDA 是在 20 世纪 90 年代初从计算机辅助设计 (CAD)、计算机辅助制造 (CAM)、计算机辅助测试 (CAT) 和计算机辅助工程 (CAE) 的概念发展而来的。一般把 EDA 技术的发展分为 CAD、CAE 和 EDA 这 3 个阶段。

CAD (Computer Aided Design) 是 EDA 技术发展的早期阶段，在这个阶段，人们开始利用计算机取代手工劳动。但当时的计算机硬件功能有限，软件功能较弱，人们主要借助计算机对所设计的电路进行一些模拟和预测，辅助进行集成电路版图编辑、印制电路板 (Printed Circuit

Board, PCB) 布局布线等简单的版图绘制工作。

CAE (Computer Aided Engineering) 是在 CAD 的工具逐步完善的基础上发展起来的, 尤其是人们在设计方法学、设计工具集成化方面取得了长足的进步, 可以利用计算机作为单点设计工具, 并建立各种设计单元库, 开始用计算机将许多单点工具集成在一起使用, 大大提高了工作效率。

20 世纪 90 年代以来, 微电子工艺有了惊人的发展, 2006 年工艺水平已经达到了 60nm, 2011 年达到 28nm。在一个芯片上已经可以集成上百万只乃至数亿只晶体管, 芯片速度达到了 Gb/s 量级。大容量的可编程逻辑器件陆续面世, 对电子设计的工具提出了更高的要求, 提供了广阔的发展空间, 促进了 EDA 技术的形成。特别重要的是, 世界各 EDA 公司致力推出兼容各种硬件实现方案和支持标准硬件描述语言的 EDA 工具软件, 有效地将 EDA 技术推向成熟。

今天, EDA 技术已经成为电子设计的重要工具, 无论是设计芯片还是设计系统, 如果没有 EDA 工具的支持, 都将是难以完成的。EDA 工具已经成为现代电路设计师的重要武器, 正在发挥着越来越重要的作用。

1.2 EDA 设计流程

利用 EDA 技术进行电路设计的大部分工作是在 EDA 软件工作平台上进行的, EDA 设计流程如图 1.1 所示。EDA 设计流程包括设计准备、设计输入、设计处理和器件编程 4 个步骤, 以及相应的功能仿真、时序仿真和器件测试 3 个设计验证过程。

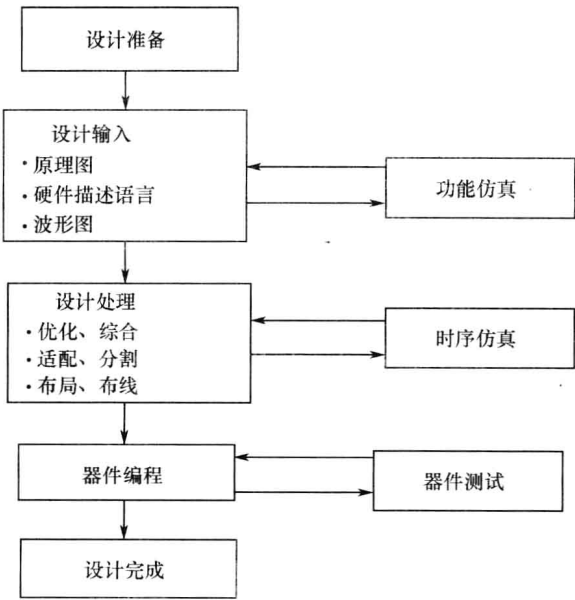


图 1.1 EDA 设计流程

1.2.1 设计准备

设计准备是设计者在进行设计之前, 依据任务要求, 确定系统所要完成的功能及复杂程度, 器件资源的利用、成本等所要做的准备工作, 如进行方案论证、系统设计和器件选择等。

1.2.2 设计输入

设计输入是将设计的电路或系统按照 EDA 开发软件要求的某种形式表示出来,并送入计算机的过程。设计输入有多种方式,包括采用硬件描述语言(如 VHDL 和 Verilog HDL)进行设计的文本输入方式、图形输入方式和波形输入方式,或者采用文本、图形两者混合的设计输入方式。也可以采用自顶向下(Top-Down)的层次结构设计方法,将多个输入文件合并成一个设计文件等。

1. 图形输入方式

图形输入也称为原理图输入,这是一种最直接的设计输入方式。它使用软件系统提供的元器件库及各种符号和连线画出设计电路的原理图,形成图形输入文件。这种方式大多用在对系统及各部分电路很熟悉的情况,或在系统对时间特性要求较高的场合。优点是容易实现仿真,便于信号的观察和电路的调整。

2. 文本输入方式

文本输入是采用硬件描述语言进行电路设计的方式。硬件描述语言有普通硬件描述语言和行为描述语言,它们用文本方式描述设计和输入。普通硬件描述语言有 AHDL、CUPL 等,它们支持逻辑方程、真值表、状态机等逻辑表达方式。

行为描述语言是目前常用的高层硬件描述语言,有 VHDL、Verilog HDL 等,它们具有很强的逻辑描述和仿真功能,可实现与工艺无关的编程与设计,可以使设计者在系统设计、逻辑验证阶段就确立方案的可行性,而且输入效率高,在不同的设计输入库之间转换也非常方便。运用 VHDL 或 Verilog HDL 硬件描述语言进行设计已是当前的趋势。

3. 波形输入方式

波形输入主要用于建立和编辑波形设计文件及输入仿真向量和功能测试向量。波形设计输入适合用于时序逻辑和有重复性的逻辑函数,系统软件可以根据用户定义的输入/输出波形自动生成逻辑关系。

波形编辑功能还允许设计者对波形进行复制、剪切、粘贴、重复与伸展,从而可以用内部节点、触发器和状态机建立设计文件,并将波形进行组合,显示各种进制的状态值。还可以通过将一组波形重叠到另一组波形上,对两组仿真结果进行比较。

1.2.3 设计处理

设计处理是 EDA 设计中的核心环节。在设计处理阶段,编译软件对设计输入文件进行逻辑化简、综合和优化,并适当地用一片或多片器件自动地进行适配,最后产生编程用的编程文件。设计处理主要包括设计编译和检查、设计优化和综合、适配和分割、布局和布线、生成编程数据文件等过程。

1. 设计编译和检查

设计输入完成之后,立即进行编译。在编译过程中,首先进行语法检验,如检查原理图的信号线有无漏接、信号有无双重来源、文本输入文件中关键词有无错误等各种语法错误,并及时标出错误的类型及位置,供设计者修改。然后进行设计规则检验,检查总的设计有无超出器件资源或规定的限制并将编译报告列出,指明违反规则和潜在不可靠电路的情况以供设计者纠正。

2. 设计优化和综合

设计优化主要包括面积优化和速度优化。面积优化的结果使得设计所占用的逻辑资源（门数或逻辑单元数）最少；时间优化的结果使得输入信号经历最短的路径到达输出，即传输延迟时间最短。综合的目的是将多个模块化设计文件合并为一个网表文件，并使层次设计平面化（即展平）。

3. 适配和分割

在适配和分割过程，确定优化以后的逻辑能否与下载目标器件 CPLD 或 FPGA 中的宏单元和 I/O 单元适配，然后将设计分割为多个便于适配的逻辑小块形式映射到器件相应的宏单元中。如果整个设计不能装入一片器件时，可以将整个设计自动分割成多块并装入同一系列的多片器件中去。

分割工作可以全部自动实现，也可以部分由用户控制，还可以全部由用户控制。分割时应使所需器件数目和用于器件之间通信的引脚数目尽可能少。

4. 布局和布线

布局和布线工作是在设计检验通过以后由软件自动完成的，它能以最优的方式对逻辑元件布局，并准确地实现元件间的布线互连。布局和布线完成后，软件会自动生成布线报告，提供有关设计中各部分资源的使用情况等信息。

5. 生成编程数据文件

设计处理的最后一步是产生可供器件编程使用的数据文件。对 CPLD 来说，是产生熔丝图文件，即 JEDEC 文件（电子器件工程联合会制定的标准格式，简称 JED 文件）；对于 FPGA 来说，是生成位流数据文件（Bit-stream Generation，简称 BG 文件）。

1.2.4 设计校验

设计校验过程包括功能仿真和时序仿真，这两项工作是在设计处理过程中同时进行的。功能仿真是在设计输入完成之后，选择具体器件进行编译之前进行的逻辑功能验证，因此又称为前仿真。此时的仿真没有延时信息或者只有由系统添加的微小标准延时，这对于初步的功能检测非常方便。仿真前，要先利用波形编辑器或硬件描述语言等建立波形文件或测试向量（即将所关心的输入信号组合成序列），仿真结果将会生成报告文件和输出信号波形，从中便可以观察到各个节点的信号变化。若发现错误，则返回设计输入中修改逻辑设计。

时序仿真是在选择了具体器件并完成布局、布线之后进行的时序关系仿真，因此又称为后仿真或延时仿真。由于不同器件的内部延时不一样，不同的布局、布线方案也会给延时造成不同的影响，因此在设计处理以后，对系统和各模块进行时序仿真，分析其时序关系，估计设计的性能及检查和消除竞争冒险等，是非常有必要的。

1.2.5 器件编程

器件编程是指将设计处理中产生的编程数据文件通过软件植入具体的可编程逻辑器件中去的操作。对 CPLD 器件来说，是将 JED 文件下载（Down Load）到 CPLD 器件中；对 FPGA 来说，是将位流数据 BG 文件配置到 FPGA 中。

器件编程需要满足一定的条件，如编程电压、编程时序和编程算法等。普通的 CPLD 器件和一次性编程的 FPGA 需要专用的编程器来完成器件的编程工作。基于 SRAM 的 FPGA 可以由 EPROM 或其他存储体进行配置。在系统可编程器件（ISP-PLD）则不需要专门的编程器，只要一根与计算机互连的下载编程电缆就可以了。

1.2.6 器件测试和设计验证

在完成器件编程之后,可以用编译时产生的文件对器件进行检验、加密,或采用边界扫描测试技术进行功能测试,测试成功后设计工作才算告一段落。

设计验证可以在 EDA 硬件开发平台上进行。EDA 硬件开发平台的核心部件是一片可编程逻辑器件、FPGA 或 CPLD,再附加一些输入/输出设备,如按键、数码显示器、指示灯、扬声器等,还提供时序电路需要的脉冲源。将设计电路编程下载到 FPGA 或 CPLD 中后,根据 EDA 硬件开发平台的操作模式要求,进行相应的输入操作,然后检查输出结果,验证设计电路。

1.3 硬件描述语言

硬件描述语言 HDL 是 EDA 技术中的重要组成部分,常用的硬件描述语言有 AHDL、VHDL 和 Verilog HDL,而 VHDL 和 Verilog HDL 是当前最流行并已成为 IEEE 标准的硬件描述语言。

1.3.1 VHDL

VHDL 是超高速集成电路硬件描述语言 (Very High Speed Integrated Circuit Hardware Description Language) 的缩写,在美国国防部的支持下于 1985 年正式推出,是目前标准化程度最高的硬件描述语言。IEEE (The Institute of Electrical and Electronics Engineers) 于 1987 年将 VHDL 采纳为 IEEE 1076 标准(VHDL 1987 版本),并于 1993 年和 2008 年分别升级为 VHDL 1993 版本和 VHDL 2008 版本。VHDL 经过 20 多年的发展、应用和完善,以其强大的系统描述能力、规范的程序设计结构、灵活的语言表达风格和多层次的仿真测试手段,在电子设计领域受到了普遍的认同和广泛的接受,成为现代 EDA 领域的首选硬件描述语言。目前,流行的 EDA 工具软件全部支持 VHDL,它在 EDA 领域的学术交流、电子设计的存档、专用集成电路 (ASIC) 设计等方面,担当着不可缺少的角色。专家认为,在 21 世纪,VHDL 与 Verilog HDL 语言将承担起几乎全部的数字系统设计任务。显然,VHDL 是现代电子设计师必须掌握的硬件设计计算机语言。

概括起来,VHDL 有以下几个特点。

① VHDL 具有强大的功能,覆盖面广,描述能力强。VHDL 支持门级电路的描述,也支持以寄存器、存储器、总线及运算单元等构成的寄存器传输级电路的描述,还支持以行为算法和结构的混合描述为对象的系统级电路的描述。

② VHDL 有良好的可读性。它可以被计算机接受,也容易被读者理解。用 VHDL 书写的源文件,既是程序又是文档;既可作为工程技术人员之间交换信息的文件,又可作为合同签约者之间的文件。

③ VHDL 具有良好的可移植性。作为一种已被 IEEE 承认的工业标准,VHDL 事实上已成为通用的硬件描述语言,可以在各种不同的设计环境和系统平台中使用。

④ 使用 VHDL 可以延长设计的生命周期。用 VHDL 描述的硬件电路与工艺无关,不会因工艺变化而使描述过时。与工艺有关的参数可以通过 VHDL 提供的属性加以描述,工艺改变时,只需要修改相应程序中的属性参数即可。

⑤ VHDL 支持对大规模设计的分解和已有设计的再利用。VHDL 可以描述复杂的电路系

统,支持对大规模设计的分解,由多人、多项目组来共同承担和完成。标准化的规则和风格,为设计的再利用提供了有力的支持。

⑥ VHDL 有利于保护知识产权。用 VHDL 设计的专用集成电路 (ASIC),在设计文件下载到集成电路时可以采用一定的保密措施,使其不易被破译和窃取。

1.3.2 Verilog HDL

Verilog HDL 也是目前应用最为广泛的硬件描述语言,并被 IEEE 采纳为 IEEE 1364-1995 标准(Verilog-1995 版本),2001 年升级为 Verilog-2001 版本,2005 年升级为 System Verilog-2005 版本。Verilog HDL 可以用来进行各种层次的逻辑设计,也可以进行数字系统的逻辑综合、仿真验证和时序分析。Verilog HDL 适合算法级 (Algorithm)、寄存器传输级 (RTL)、逻辑级 (Logic)、门级 (Gate) 和版图级 (Layout) 等各个层次的电路设计和描述。

采用 Verilog HDL 进行电路设计的最大优点是设计与工艺无关性,这使得设计者在进行电路设计时可以不必过多考虑工艺实现时的具体细节,只需要根据系统设计的要求施加不同的约束条件,即可设计出实际电路。实际上,利用计算机的强大功能,在 EDA 工具的支持下,把逻辑验证与具体工艺库相匹配,将布线及延迟计算分成不同的阶段来实现,可减少设计者的繁重劳动。

Verilog HDL 和 VHDL 都是用于电路设计的硬件描述语言,并且都已成为 IEEE 标准。Verilog HDL 也具有与 VHDL 类似的特点,稍有不同的是,Verilog HDL 早在 1983 年就已经推出,应用历史较长,拥有广泛的设计群体,设计资源比 VHDL 丰富。另外,Verilog HDL 是在 C 语言的基础上演化而来的,因此只要具有 C 语言的编程基础,设计者就很容易学会并掌握这种语言。

1.3.3 AHDL

AHDL (Altera Hardware Description Language) 是 Altera 公司根据自己公司生产的 MAX 系列器件和 FLEX 系列器件的特点,专门设计的一套完整的硬件描述语言。

AHDL 是一种模块化的硬件描述语言,它完全集成于 Altera 公司的 MAX+PLUS II 和 Quartus II 的软件开发系统中。AHDL 特别适合于描述复杂的组合电路、组 (group) 运算及状态机、真值表和参数化的逻辑。用户可以通过 MAX+PLUS II 或 Quartus II 的软件开发系统对 AHDL 源程序进行编辑,并通过对源文件的编译建立仿真、时域分析和器件编程的输出文件。

AHDL 的语句和元素种类齐全、功能强大,而且易于应用。用户可以使用 AHDL 建立完整层次的工程设计项目,或者在一个层次的设计中混合其他类型的设计文件,如 VHDL 设计文件或 Verilog HDL 设计文件。

1.4 可编程逻辑器件

可编程逻辑器件 (Programmable Logic Device, PLD) 是一种半定制集成电路,在其内部集成了大量的门和触发器等基本逻辑单元电路 (LEs),用户通过编程来改变 PLD 内部电路的逻辑关系或连线,就可以得到所需要的设计电路。可编程逻辑器件的出现,改变了传统的数字系统设计方法,其设计方法为采用 EDA 技术开创了广阔的发展空间,并极大地提高了电路设计的效率。