



Xilinx大学计划指定教材

高等院校信息技术规划教材

EDA原理及Verilog实现

何 宾 编著

INFORMATION TECHNOLOGY

清华大学出版社

高等院校信息技术规划教材

EDA原理及Verilog实现

何 宾 编著

清华大学出版社
北京

内 容 简 介

本书是为高等学校信息类和其他相关专业编著的教材。本书共分为 11 章。主要介绍了 EDA 设计导论,可编程逻辑器件设计方法,Verilog HDL 语言基础,数字逻辑单元设计,Verilog HDL 高级设计技术,基于 HDL 的设计输入,基于原理图的设计输入,设计综合和行为仿真,设计实现和时序仿真,设计下载和调试,数字系统设计实例。

根据 EDA 课程的教学要求和实际的教学实践的体会,本书不仅系统地介绍了 EDA 的设计理论,而且将理论和实践相结合,给出了大量的设计实例,并反映了最新的 EDA 设计技术及发展趋势。

本书可作为大学本科生和研究生教材,也可作为从事 Xilinx 可编程逻辑器件设计的设计人员的参考用书,同时也可作为与 Xilinx 相关的培训教材。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

EDA 原理及 Verilog 实现/何宾编著. —北京:清华大学出版社,2010.7

(高等院校信息技术规划教材)

ISBN 978-7-302-22489-1

I. ①E… II. ①何… III. ①电子电路—电路设计:计算机辅助设计—高等学校—教材②硬件描述语言,Verilog HDL—程序设计—高等学校—教材 IV. ①TN702②TP312

中国版本图书馆 CIP 数据核字(2010)第 067443 号

责任编辑:战晓雷 李玮琪

责任校对:梁 毅

责任印制:王秀菊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

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

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62795954,jsjic-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015,zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京市清华园胶印厂

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185×260 印 张:22.5

字 数:531 千字

版 次:2010 年 7 月第 1 版

印 次:2010 年 7 月第 1 次印刷

印 数:1~3000

定 价:33.00 元

产品编号:035791-01

前言

foreword

随着半导体技术和数字化处理技术的飞速发展,以及新电子产品上市周期的缩短,基于可编程逻辑器件的复杂数字系统设计成为电子设计自动化技术 EDA 中一个重要的研究方向和应用领域。采用可编程逻辑器件 PLD 比采用专用集成电路 ASIC 和专用标准部件 ADDP 的成本低。基于 EDA 技术和可编程逻辑器件的数字系统设计技术,可极大地缩短系统设计周期,满足市场对产品竞争力的要求。

随着可编程逻辑器件功能的日趋强大和相关设计软件性能的不断完善,基于 EDA 技术的设计方法越来越受到电子设计人员的重视,硬件描述语言 HDL、原理图、IP 核和网表等设计方法成为可编程逻辑器件设计中需要掌握的设计技术。可编程逻辑器件已经从单纯的数字逻辑设备发展到了片上可编程系统 SOPC 阶段,因此要求 EDA 设计人员必须能够实现软件和硬件的协同工作。

本书力图全面系统地介绍基于 Xilinx 可编程逻辑器件的设计原理和方法。通过系统介绍其设计原理和方法,使读者能够系统地掌握可编程逻辑器件的设计方法和技巧。本书主要包括以下几个部分。

EDA 的设计导论部分,该部分主要介绍 EDA 技术的发展历史、EDA 技术所涉及的内容、设计流程和 HDL 硬件描述语言概念。

可编程逻辑器件设计方法部分,该部分主要介绍可编程逻辑的分类,主要工艺,复杂可编程逻辑器件 CPLD 和现场可编程逻辑阵列 FPGA 的结构,并对 Xilinx 芯片的性能和结构进行了详细介绍。

Verilog HDL 语言基础部分,该部分介绍了 Verilog 程序结构,Verilog 语言要素,Verilog 常量,Verilog 数据类型,Verilog 名字空间,Verilog 语言表达式,Verilog 行为描述语句,Verilog 门级描述语句,Verilog 生成语句及结构,Verilog 编译指示语句,Verilog 系统任务和函数,Verilog 用户定义任务和函数,Verilog 语言模块描述方式等内容。

逻辑单元设计部分,该部分的介绍分成简单逻辑单元设计和复

杂逻辑单元设计两个部分。在简单逻辑单元设计部分重点介绍了组合逻辑电路设计和时序逻辑电路设计;在复杂逻辑单元设计部分重点介绍了存储器设计、运算单元设计和有限自动状态机设计。

Verilog HDL 高级设计技术部分,重点介绍了提高 Verilog HDL 设计性能的几种常用方法,其中包括逻辑复制和复用、并行和流水线技术、系统同步和异步单元、逻辑结构设计方法和模块的划分原则。通过该部分的学习,可以增强读者熟练运用 HDL 语言进行设计的能力。

IP 核设计技术部分,该部分重点介绍了 IP 核的分类、优化、生成和应用。

EDA 设计流程部分,基于 Xilinx 的 ISE 软件平台,首先介绍了基于 HDL 的设计流程,然后介绍了基于原理图的设计流程,最后介绍了设计综合、行为仿真、设计实现、时序仿真、设计下载和设计调试的完整设计过程。

数字系统设计实例部分,给出了 17 个不同数字系统的设计实例,每个实例都给出了设计原理,设计实现与验证,设计代码。这些设计实例分别进行了软件仿真,并在硬件平台进行了验证。这部分内容对于读者掌握数字系统设计方法和技巧非常重要。

在讲授和学习本书内容时,可以根据教学时数和内容的侧重点不同,适当地将相关章节的内容进行调整和删减。为了让读者更好地掌握相关的内容,本书还给出了大量设计示例程序和习题详细的设计代码。

本书不仅可以作为大学信息类专业讲授 EDA 相关课程的教学用书,也可以作为从事 EDA 教学和科研工作者的参考用书。

在本书的编写过程中引用和参考了许多著名学者和专家的研究成果,同时也参考了 Xilinx 公司的技术文档和手册,在此向他们表示衷心的感谢。在本书编写的过程中,刘静负责第 3 章和第 4 章的编写,王瑜负责第 6 章和第 10 章的编写,王纲领负责第 11 章的编写和验证工作,在此向他们表示感谢。在本书的出版过程中,得到了 Xilinx 公司大学合作计划的大力帮助,同时也得到了清华大学出版社各位编辑的帮助和指导,在此也表示深深的谢意。

由于编者水平有限,编写时间仓促,书中难免有疏漏之处,敬请读者批评指正。

作 者

2010 年 4 月

目录

Contents

第 1 章 EDA 设计导论	1
1.1 EDA 技术综述	1
1.1.1 EDA 技术发展历史	1
1.1.2 EDA 技术含义	3
1.1.3 EDA 技术主要内容	3
1.2 PLD 设计方法学	7
1.2.1 PLD 设计概论	7
1.2.2 PLD 设计流程	8
1.2.3 SOPC 设计流程	11
1.3 HDL 硬件描述语言	11
1.3.1 HDL 硬件描述语言概念	11
1.3.2 HDL 语言特点和比较	12
1.3.3 HDL 语言最新发展	13
习题	14
第 2 章 可编程逻辑器件设计方法	15
2.1 可编程逻辑器件基础	15
2.1.1 可编程逻辑器件概述	15
2.1.2 可编程逻辑器件的发展历史	16
2.2 PLD 芯片制造工艺	17
2.3 PLD 芯片结构	18
2.3.1 CPLD 原理及结构	18
2.3.2 FPGA 原理及结构	20
2.3.3 CPLD 和 FPGA 比较	24
2.3.4 PLD 选择原则	25
2.4 Xilinx 公司芯片简介	26
2.4.1 Xilinx CPLD 芯片介绍	26
2.4.2 Xilinx FPGA 芯片介绍	29

2.4.3 Xilinx PROM 芯片介绍	40
习题	41

第3章 Verilog HDL 语言基础

42

3.1 Verilog 语言概述	42
3.1.1 Verilog HDL 语言发展历史	42
3.1.2 Verilog HDL 硬件描述语言功能	43
3.2 Verilog 程序结构	44
3.2.1 模块声明	44
3.2.2 模块端口定义	45
3.2.3 信号类型声明	46
3.2.4 逻辑功能定义	46
3.3 Verilog 语言要素	47
3.4 Verilog 常量	48
3.4.1 整数型常量	48
3.4.2 实数型常量	50
3.4.3 字符串常量	50
3.5 Verilog 数据类型	51
3.5.1 网络和变量	52
3.5.2 参数	56
3.5.3 向量	57
3.6 Verilog 名字空间	59
3.7 Verilog 语言表达式	60
3.7.1 操作符	60
3.7.2 延迟表达式	66
3.7.3 表达式的位宽	66
3.7.4 有符号表达式	67
3.8 Verilog 行为描述语句	67
3.8.1 过程语句	68
3.8.2 语句块	70
3.8.3 赋值语句	72
3.8.4 分支语句	75
3.8.5 循环控制语句	78
3.9 Verilog 门级描述语句	80
3.9.1 内置基本门级元件	81
3.9.2 用户自定义基本元件	85
3.10 Verilog 生成语句及结构	89
3.11 Verilog 编译指示语句	90

3.12 Verilog 系统任务和函数	94
3.13 Verilog 用户定义任务和函数	103
3.13.1 任务	103
3.13.2 函数	106
3.14 Verilog 语言模块描述方式	108
3.14.1 层次化设计方法	108
3.14.2 模块的行为级描述	109
3.14.3 模块的数据流描述	109
3.14.4 模块的结构描述	111
习题	112
第 4 章 数字逻辑单元设计	113
4.1 组合逻辑电路设计	113
4.1.1 基本逻辑门电路设计	113
4.1.2 编码器和译码器设计	114
4.1.3 数据选择器设计	116
4.1.4 数字比较器设计	117
4.1.5 数据运算单元设计	117
4.1.6 总线缓冲器设计	119
4.2 时序逻辑电路设计	120
4.2.1 时钟和复位设计	120
4.2.2 触发器设计	121
4.2.3 锁存器设计	122
4.2.4 计数器设计	122
4.2.5 移位寄存器设计	124
4.3 存储器设计	125
4.3.1 ROM 设计	125
4.3.2 RAM 设计	127
4.4 有限自动状态机设计	127
4.4.1 有限状态机原理	128
4.4.2 有限状态机分类	128
4.4.3 有限状态机设计	131
习题	136
第 5 章 Verilog HDL 高级设计技术	138
5.1 Verilog HDL 代码风格	138

5.1.1	逻辑复制和复用技术	139
5.1.2	并行和流水线技术	141
5.1.3	同步和异步单元处理技术	143
5.1.4	逻辑处理技术	147
5.1.5	模块划分的设计原则	152
5.2	IP 核技术	154
5.2.1	IP 核分类	154
5.2.2	IP 核优化	155
5.2.3	IP 核生成	156
5.2.4	IP 核应用	156
习题		162
第 6 章	基于 HDL 的设计输入	163
6.1	软件环境	163
6.2	综合工具介绍	165
6.3	工程建立	165
6.4	设计描述	167
6.5	添加设计和检查	168
6.6	创建基于 HDL 的模块	169
6.7	IP 核产生和例化	170
6.7.1	IP 核的生成	171
6.7.2	IP 核的例化	172
习题		174
第 7 章	基于原理图的设计输入	175
7.1	工程建立	175
7.2	设计描述	176
7.3	创建原理图模块	178
7.3.1	原理图编辑器操作	178
7.3.2	定义模块符号	178
7.3.3	创建模块符号	182
7.4	创建状态图模块	182
7.4.1	添加状态	184
7.4.2	添加迁移	184
7.4.3	添加行为	184
7.4.4	添加复位条件	185

7.4.5 设计输出和添加	186
7.5 设计完成	186
习题	186
第 8 章 设计综合和行为仿真	187
8.1 设计综合	187
8.1.1 行为综合描述	187
8.1.2 综合属性设置	188
8.1.3 基于 XST 的综合概述	188
8.1.4 约束及设计综合的实现	189
8.1.5 RTL 符号的查看	189
8.2 行为仿真的实现	190
8.2.1 生成测试向量	190
8.2.2 基于 Modelsim 行为仿真实现	193
8.2.3 基于 ISE 行为仿真实现	197
习题	200
第 9 章 设计实现和时序仿真	201
9.1 实现过程概述及约束	201
9.1.1 实现过程概述	201
9.1.2 建立约束文件	201
9.2 实现属性参数设置	202
9.3 创建分区	203
9.4 创建时序约束	204
9.5 设计翻译	204
9.6 设计约束	205
9.6.1 时序约束	205
9.6.2 管脚和面积约束	208
9.7 设计映射及时序分析	210
9.7.1 设计映射	210
9.7.2 使用时序分析评估块延迟	210
9.8 布局布线验证	212
9.8.1 用 FPGA Editor 验证布局布线	213
9.8.2 评估布局后时序	214
9.8.3 改变分区 HDL	215
9.9 时序仿真实现	216

9.9.1	时序仿真概述	216
9.9.2	使用 ModelSim 进行时序仿真	216
9.9.3	使用 ISE 仿真器进行时序仿真	221
习题		224
第 10 章	设计下载和调试	225
10.1	可编程逻辑器件配置接口	225
10.2	创建配置数据	233
10.2.1	配置属性设置	233
10.2.2	创建 PROM 文件	234
10.3	下载实现	236
10.3.1	下载环境	236
10.3.2	下载实现	237
10.3.3	JTAG 诊断	242
10.3.4	建立 SVF 文件	243
10.4	可编程逻辑器件调试	245
10.4.1	多路复用技术的应用	245
10.4.2	虚拟逻辑分析工具概述	246
10.4.3	ChipScope Pro 调试工具概述	246
习题		248
第 11 章	数字系统设计实例	249
11.1	7 段数码管显示功能设计	249
11.1.1	7 段数码管基本显示功能设计	249
11.1.2	自动扫描复用 7 段数码管的设计	253
11.2	二进制码转换设计	256
11.2.1	4 位二进制码到 BCD 码变换设计	256
11.2.2	8 位二进制码到 BCD 码变换设计	257
11.2.3	4 位二进制码到 Gray 码变换设计	260
11.2.4	4 位 Gray 码到二进制码变换设计	261
11.3	基于移位相加运算的乘法器设计	262
11.3.1	设计原理	262
11.3.2	设计实现及验证	263
11.3.3	设计代码	264
11.4	基于移位相减运算的除法器设计	265
11.4.1	除法器设计原理	265

11.4.2	设计实现及验证	267
11.4.3	设计代码	267
11.5	4 位 ALU 单元设计	269
11.5.1	设计原理	269
11.5.2	设计实现和验证	269
11.5.3	设计代码	270
11.6	伪随机序列产生器设计	271
11.6.1	伪随机序列生成原理	271
11.6.2	设计实现及验证	272
11.7	滚动 7 段数码显示设计	273
11.7.1	设计原理	273
11.7.2	设计实现和验证	274
11.7.3	设计代码	274
11.8	序列检测器设计	277
11.8.1	设计原理	277
11.8.2	设计实现及验证	278
11.8.3	设计代码	278
11.9	有限脉冲响应 FIR 滤波器设计	280
11.9.1	有限脉冲响应 FIR 滤波器设计原理	280
11.9.2	设计实现及验证	281
11.9.3	设计代码	282
11.10	异步先进先出队列 FIFO 设计	284
11.10.1	异步先进先出 FIFO 原理	284
11.10.2	设计实现及验证	284
11.10.3	设计代码	286
11.11	数字时钟的设计	288
11.11.1	数字时钟原理	288
11.11.2	设计实现及验证	289
11.11.3	设计代码	290
11.12	液晶显示模块应用设计	294
11.12.1	液晶显示模块原理	295
11.12.2	设计实现及验证	298
11.12.3	设计代码	298
11.13	VGA 接口设计	300
11.13.1	设计原理	300
11.13.2	设计实现及验证	303
11.13.3	设计代码	304

11.14	PS/2 键盘接口设计	309
11.14.1	设计原理	309
11.14.2	设计实现及验证	312
11.14.3	设计代码	312
11.15	串行 A/D 转换器应用设计	316
11.15.1	系统设计原理	316
11.15.2	设计实现及验证	318
11.15.3	设计代码	318
11.16	数字电压表的设计	324
11.16.1	数字电压表设计原理	324
11.16.2	设计实现及验证	327
11.16.3	设计代码	327
11.17	函数信号发生器的设计	332
11.17.1	函数信号发生器设计原理	332
11.17.2	设计实现及验证	333
11.17.3	设计代码	334
附录 A XST 支持的 Verilog 结构		337
附录 B XST 支持的 Verilog 语句		339
附录 C XST 支持的 Verilog 系统任务和函数		341
附录 D XST 支持的 Verilog 原语		342
附录 E XST 支持的 Verilog 关键字		343
参考文献		344

EDA 设计导论

本章主要介绍 EDA 技术综述、PLD 设计方法学和 HDL 硬件描述语言。在 EDA 技术综述部分,重点介绍 EDA 技术发展历史、EDA 技术含义、EDA 技术主要内容;在 PLD 设计方法学部分,介绍 PLD 设计概论、PLD 设计流程、SOPC 设计流程;在 HDL 硬件描述语言部分,介绍 HDL 硬件描述语言概念、HDL 语言特点和比较、HDL 语言最新发展。本章的主要目的是让读者对 EDA 技术有一个初步了解,为学习后续章节的内容打下良好的基础。

1.1 EDA 技术综述

1.1.1 EDA 技术发展历史

EDA 技术伴随着计算机、集成电路、电子系统设计的发展,经历了计算机辅助设计(Computer Assist Design, CAD),计算机辅助工程设计(Computer Assist Engineering, CAE)和电子设计自动化(Electronic Design Automation, EDA)三个发展阶段。

1. 计算机辅助设计阶段

早期的电子系统硬件设计采用的是分立元件,随着集成电路的出现和应用,硬件设计进入到发展的初级阶段。初级阶段的硬件设计大量选用中小规模标准集成电路,人们将这些器件焊接在电路板上,做成初级电子系统,对电子系统的调试是在组装好的 PCB (Printed Circuit Board)板上进行的。

由于设计师对图形符号使用数量有限,传统的手工布图方法无法满足产品复杂性的要求,更不能满足工作效率的要求。这时,人们开始将产品设计过程中高度重复性的繁杂劳动,如布图布线工作,用二维图形编辑与分析的 CAD 工具替代,最具代表性的产品就是美国 ACCEL 公司开发的 Tango 布线软件。20 世纪 70 年代,是 EDA 技术发展初期,由于 PCB 布图布线工具受到计算机工作平台的制约,其支持的设计工作有限且性能比较差。

2. 计算机辅助工程设计阶段

初级阶段的硬件设计是用大量的标准芯片实现电子系统设计。随着微电子工艺的

发展,相继出现了集成上万只晶体管的微处理器、集成几十万直到上百万储存单元的随机存储器和只读存储器。此外,支持定制单元电路设计的硅编辑、掩膜编程的门阵列,如标准单元的半定制设计方法以及可编程逻辑器件(PAL 和 GAL)等一系列微结构和微电子学的研究成果都为电子系统的设计提供了新天地。因此,可以用少数几种通用的标准芯片实现电子系统的设计。

伴随计算机和集成电路的发展,EDA 技术进入到计算机辅助工程设计阶段。20 世纪 80 年代初,推出的 EDA 工具则以逻辑模拟、定时分析、故障仿真、自动布局和布线为核心,重点解决电路设计没有完成之前的功能检测等问题。利用这些工具,设计师能在产品制作之前预知产品的功能与性能,能生成产品制造文件,在设计阶段对产品性能的分析前进了一大步。

如果说 20 世纪 70 年代的自动布局布线的 CAD 工具代替了设计工作中绘图的重重复劳动,那么,20 世纪 80 年代出现的具有自动综合能力的 CAE 工具则代替了设计师的部分工作,对保证电子系统的设计,制造出最佳的电子产品起着关键的作用。到了 20 世纪 80 年代后期,EDA 工具已经可以进行设计描述、综合与优化和设计结果验证,CAE 阶段的 EDA 工具不仅为成功开发电子产品创造了有利条件,而且为高级设计人员的创造性劳动提供了方便。但是,大部分从原理图出发的 EDA 工具仍然不能适应复杂电子系统的设计要求,而具体化的元件图形制约着优化设计。

3. 电子系统设计自动化阶段

为了满足千差万别的系统用户提出的设计要求,最好的方法是自己设计芯片,把想设计的电路直接设计在专用芯片上。微电子技术的发展,特别是可编程逻辑器件的发展,使微电子厂家可以为用户提供各种规模的可编程逻辑器件,使设计者通过设计芯片实现电子系统功能。EDA 工具的发展,又为设计师提供了全线 EDA 工具。这个阶段发展起来的 EDA 工具,目的是在设计前期将设计师从事的许多高层次设计用工具来完成,如可以将用户要求转换为设计技术规范,有效地处理可用的设计资源与理想的设计目标之间的矛盾,按具体的硬件、软件和算法分解设计等。由于电子技术和 EDA 工具的发展,设计师可以在不太长的时间内使用 EDA 工具,通过一些简单标准化的设计过程,利用微电子厂家提供的设计库来完成数万门 ASIC 和集成系统的设计与验证。硬件描述语言 HDL 的出现是这个阶段最重要的成果,HDL 语言的出现使 EDA 设计进入到抽象描述的设计层次。

20 世纪 90 年代,设计师逐步从使用硬件转向设计硬件,从单个电子产品开发转向系统级电子产品开发(即片上系统集成, System on a Chip)。因此,EDA 工具是以系统级设计为核心,包括系统行为级描述与结构综合,系统仿真与测试验证,系统划分与指标分配,系统决策与文件生成等一整套的电子系统设计自动化工具。这时的 EDA 工具不仅具有电子系统设计的能力,而且能提供独立于工艺和厂家的系统级设计能力,具有高级抽象的设计构思手段。例如,提供方框图、状态图和流程图的编辑能力,具有适合层次描述和混合信号描述的硬件描述语言(如 VHDL、AHDL 或 Verilog-HDL),同时含有各种工艺的标准元件库。只有具备上述功能的 EDA 工具,才可能使电子系统工程在不熟

悉各种半导体工艺的情况下,完成电子系统的设计。

21 世纪初,随着微电子技术的进一步发展,EDA 设计进入了更高的阶段,即片上系统设计(System on Programmable Chip, SOPC)阶段,在这个阶段,可编程逻辑器件内集成了数字信号处理器的内核、微处理器的内核等,使得可编程逻辑器件不再只是完成复杂的逻辑功能,而且具有了强大的信号处理和控制在功能。SOPC 技术的进一步发展必将给电子系统的设计带来一场深刻的变革。

1.1.2 EDA 技术含义

EDA 技术是一门迅速发展的新技术,涉及面广,内容丰富,理解各异,目前尚无统一的看法。从一般认识上,将 EDA 技术分成狭义 EDA 技术和广义 EDA 技术。

狭义 EDA 技术,就是指以大规模可编程逻辑器件为设计载体,以硬件描述语言为系统逻辑描述的主要表达方式,以计算机、大规模可编程逻辑器件的开发软件及实验开发系统为设计工具,通过有关的开发软件,自动完成用软件方式设计的电子系统到硬件系统的逻辑编译、逻辑化简、逻辑分割、逻辑综合及优化、逻辑布局布线、逻辑仿真,直至对于特定目标芯片的适配编译、逻辑映射、编程下载等工作,最终形成集成电子系统或专用集成芯片的一门新技术,或称为 IES/ASIC 自动设计技术。狭义 EDA 技术也就是使用 EDA 软件进行数字系统的设计,这也是本书所要介绍的内容。

广义 EDA 技术,是通过计算机及其电子系统的辅助分析和设计软件,完成电子系统某一部分的设计过程。因此,广义 EDA 技术除了包含狭义的 EDA 技术外,还包括计算机辅助分析 CAA 技术(如 PSPICE, EWB, MATLAB 等),印刷电路板计算机辅助设计 PCB-CAD 技术(如 PROTEL, ORCAD 等)和其他高频和射频设计和分析的工具等。

不论是广义 EDA 技术还是狭义 EDA 技术,它们都有以下共同的特点。

- (1) 通过使用相应的电路分析和设计软件,完成电子系统某部分的设计;
- (2) 在电子系统设计中所使用的 EDA 软件基本都符合自顶向下的设计流程的理念;
- (3) 使用 EDA 软件设计电子系统,都需要分工设计,团体协作;
- (4) 使用 EDA 软件设计电子系统,提高了设计的效率,缩短了设计周期;
- (5) 使用 EDA 软件设计电子系统,采用了模块化和层次化的设计方法;
- (6) 大多数 EDA 软件都具有仿真和模拟功能。

综上所述,EDA 技术的不断发展为日益庞大的电子系统设计提供了强大的动力和技术保障。

1.1.3 EDA 技术主要内容

基于狭义 EDA 技术进行可编程逻辑器件的设计应掌握以下几个方面的内容。

- 大规模可编程逻辑器件(Programmable Logic Device, PLD), PLD 是利用 EDA 技术进行电子系统设计的载体;
- 硬件描述语言(Hardware Description Language, HDL), HDL 语言是利用 EDA

技术进行电子系统设计的主要表达手段;

- EDA 设计软件(Electronic Design Automation Software, EDAS), EDA 设计软件是利用 EDA 技术进行电子系统设计的自动化设计工具;
- 相关的硬件平台, 硬件平台是利用 EDA 技术进行电子系统设计的下载工具及硬件验证工具。

下面对这几个问题进行详细描述。

1. 大规模可编程逻辑器件

PLD 是一种由用户编程以实现某种逻辑功能的新型逻辑器件。现在所说的 PLD 器件一般包含现场可编程门阵列(FPGA)和复杂可编程逻辑器件(CPLD)。PLD 器件的应用十分广泛, 它们将随着 EDA 技术的发展而成为电子设计领域的重要角色。

FPGA 在结构上主要分为可编程逻辑单元, 可编程输入输出单元和可编程连线三个部分。CPLD 在结构上主要包括三个部分, 即可编程逻辑宏单元, 可编程输入输出单元和可编程内部连线。

由于 PLD 的集成规模非常大, 因此可利用先进的 EDA 工具进行电子系统设计和产品开发。由于开发工具的通用性、设计语言的标准化以及设计过程几乎与所用器件的硬件结构无关, 因而设计开发成功的各类逻辑功能块软件有很好的兼容性和可移植性。它几乎可用于任何型号和规模的 PLD 中, 从而使产品设计效率大幅度提高, 可以在很短时间内完成十分复杂的系统设计, 这正是产品快速进入市场最重要的特征。

与 ASIC 设计相比, PLD 显著的优势是开发周期短、投资风险小、产品上市速度快、市场适应能力强和硬件升级回旋余地大, 而且当产品定型和产量扩大后, 可将在生产中达到充分检验的 VHDL 设计迅速实现 ASIC 投产。

2. 硬件描述语言

常用硬件描述语言有 VHDL、Verilog 和 ABEL 语言。VHDL 起源于美国国防部的 VHSIC, Verilog 起源于集成电路的设计, ABEL 则来源于可编程逻辑器件的设计。

下面从使用方面对这三种语言进行简要的对比。

(1) 逻辑描述层次: 一般的硬件描述语言可以在三个层次上进行电路描述, 其层次由高到低依次可分为行为级、RTL 级和门电路级。VHDL 语言是一种高级描述语言, 适用于行为级和 RTL 级的描述, 最适于描述电路的行为; Verilog 语言和 ABEL 语言是较低级的描述语言, 适用于 RTL 级和门电路级的描述, 最适于描述门级电路。

(2) 设计要求: VHDL 进行电子系统设计时可以不了解电路的结构细节, 设计者所做的工作较少; Verilog 和 ABEL 语言进行电子系统设计时需了解电路的结构细节, 设计者需做大量的工作。

(3) 综合过程: 任何一种语言源程序, 最终都要转换成门电路级才能被布线器或适配器所接受。因此, VHDL 语言源程序的综合通常要经过行为级→RTL 级→门电路级的转化, VHDL 几乎不能直接控制门电路的生成。而 Verilog 语言和 ABEL 语言源程序的综合过程稍简单, 即经过 RTL 级→门电路级的转化, 易于控制电路资源。