



Xilinx大学计划指定教材

高等院校信息技术规划教材

EDA原理及应用

何 宾 编著

INFORMATION TECHNOLOGY
INFORMATION TECHNOLOGY
INFORMATION TECHNOLOGY

清华大学出版社



高等院校信息技术规划教材

EDA原理及应用

何 宾 编著

清华大学出版社
北京

内 容 简 介

本书系统而又全面地介绍了基于 EDA 技术的数字系统设计的方法、理论和应用。全书共分 13 章, 内容包括 EDA 设计导论、可编程逻辑器件设计方法、VHDL 语言基础、数字逻辑单元设计、VHDL 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、数字时钟设计及实现、通用异步接收发送器、数字电压表设计及实现。本书参考了大量最新的设计资料, 内容新颖, 理论与应用并重, 充分反映了基于 EDA 技术的数字系统设计的最新方法和技术, 可以帮助读者尽快掌握 EDA 设计方法和技术。

本书可作为相关专业开设 EDA 原理及应用课程的本科教学参考书, 亦可作为从事 EDA 数字系统设计的相关教师、研究生和科技人员自学参考书, 也可作为 Xilinx 公司的培训教材。

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 原理及应用/何宾编著. —北京: 清华大学出版社, 2009. 6

(高等院校信息技术规划教材)

ISBN 978-7-302-20021-5

I. E… II. 何… III. 电子电路—电路设计: 计算机辅助设计—高等学校—教材
IV. TP702

中国版本图书馆 CIP 数据核字(2009)第 063664 号

责任编辑: 战晓雷 赵晓宁

责任校对: 焦丽丽

责任印制: 李红英

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: 010-62776969, c-service@tup.tsinghua.edu.cn

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

印 装 者: 北京国马印刷厂

经 销: 全国新华书店

开 本: 185×260 印 张: 18.25 字 数: 445 千字

版 次: 2009 年 6 月第 1 版 印 次: 2009 年 6 月第 1 次印刷

印 数: 1~4000

定 价: 27.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系调换。联系电话: 010-62770177 转 3103 产品编号: 032603-01

前言

foreword

随着半导体技术的飞速发展,新电子产品上市周期的缩短,以及数字化处理技术的不断提高,大规模可编程逻辑器件(PLD)的设计成为电子系统设计中一个重要的研究方向和应用领域。在现阶段,必须依赖于高性能电子设计自动化 EDA 技术,才能完成基于 PLD 复杂数字系统的设计。采用 PLD 比采用专用集成电路 ASIC 和专用标准部件 ADDP 的成本低。通过使用 EDA 技术完成 PLD 设计,大大缩短了设计周期,适应市场对产品竞争力的要求。

随着 PLD 复杂度的提高和 EDA 软件性能的不断完善,基于 EDA 的设计原理和方法,越来越受到 EDA 设计人员的重视。基于硬件描述语言 HDL、原理图、IP 核等混合设计方法成为 PLD 设计中主要采用的方法。对 PLD 的设计已经发展到了片上可编程系统 SOPC 阶段,因此要求 EDA 设计人员能够实现软件和硬件的协同设计。

本书力图全面系统地介绍基于 PLD 的 EDA 设计原理和方法。通过系统介绍 EDA 的原理和方法,使读者能够系统、全面地掌握 EDA 设计方法和应用技巧。本书主要分为以下几个部分:

(1) EDA 的设计概论部分。该部分主要是介绍 EDA 技术的发展历史、EDA 技术所涉及的内容、设计流程和 HDL 硬件描述语言概要。

(2) 可编程逻辑器件设计方法部分。该部分主要介绍了可编程逻辑器件的分类,主要工艺,重点介绍了复杂可编程逻辑器件 CPLD 和现场可编程逻辑阵列 FPGA 的结构,并进行了比较。本部分也对 Xilinx 芯片的性能和结构进行了比较详细的介绍。

(3) 硬件描述语言 VHDL 部分。该部分介绍了 VHDL 语言的结构和风格,VHDL 语言的数据类型和运算符,VHDL 语言的主要描述语句等。在介绍这些内容时为了便于学习,给出了大量单元模块的 VHDL 设计代码。

(4) 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。通过这样的结构安排使学生更好地掌握不同层次模块 HDL 的书写规范。

(5) 高性能代码设计部分。该部分作为前面 VHDL 语言内容的补充,重点介绍了提高 HDL 设计性能的几种常用方法:逻辑复制和复用、并行和流水技术、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。

(6) IP 核设计技术部分。该部分虽然篇幅不多,但是所涉及的内容非常重要,该部分重点介绍了 IP 核的分类、优化、生成和应用。通过该部分的学习,读者能更好地了解 IP 核设计技术的各个方面,为今后从事 IP 核设计打下良好基础。

(7) 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,在介绍这些内容时,采用了混合设计方法,对于掌握整个设计流程有很大的帮助。在介绍混合设计的基础上,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,该部分也是本书最重要的内容之一。

作为对本书内容的重要补充,在本书的结尾给出三个比较典型的数字系统的设计实例,使读者从中能够进一步体会 EDA 设计技巧和方法,并且从中可以了解设计原理、设计输入、设计验证的工程化设计方法。《EDA 原理及应用实验教程》作为本书的配套用书,帮助读者通过软件和硬件的实验,进一步掌握使用 Xilinx 软件和硬件平台进行 EDA 设计的方法和技巧。

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。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。在本书的编写过程中,由杨青青、李宝敏负责部分章节文字录入工作,在此向她们表示感谢。在本书的出版过程中,得到了 Xilinx 公司“大学计划”项目的大力支持与帮助,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。

编 者

2008 年 10 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	14
习题 1	15
第 2 章 可编程逻辑器件设计方法	16
2.1 可编程逻辑器件基础	16
2.1.1 可编程逻辑器件概述	16
2.1.2 可编程逻辑器件的发展历史	17
2.2 PLD 芯片制造工艺	18
2.3 PLD 芯片结构	19
2.3.1 CPLD 原理及结构	19
2.3.2 FPGA 原理及结构	21
2.3.3 CPLD 和 FPGA 比较	25
2.3.4 PLD 选择原则	26
2.4 Xilinx 公司芯片简介	28
2.4.1 Xilinx CPLD 芯片介绍	28



2.4.2	Xilinx FPGA 芯片介绍	30
2.4.3	Xilinx PROM 芯片介绍	39
习题 2		43
第 3 章	VHDL 语言基础	44
3.1	VHDL 程序结构	44
3.1.1	VHDL 程序结构概述	44
3.1.2	VHDL 程序实体	44
3.1.3	VHDL 结构体	47
3.2	VHDL 语言描述风格	49
3.2.1	结构体行为描述	49
3.2.2	结构体数据流描述	50
3.2.3	结构体结构化描述	51
3.3	设计资源共享	53
3.3.1	库	53
3.3.2	包集合	54
3.3.3	子程序和函数	56
3.3.4	元件配置	58
3.4	VHDL 语言的文字规则	59
3.4.1	数字型文字	59
3.4.2	字符型文字	60
3.4.3	标识符	61
3.4.4	下标名及下标段名	62
3.5	VHDL 语言数据对象、类型和属性	62
3.5.1	VHDL 中的数据对象	62
3.5.2	VHDL 中的数据类型	64
3.5.3	VHDL 中的预定义属性	71
3.6	VHDL 语言的操作符	74
3.7	VHDL 的顺序描述语句	76
3.7.1	对象赋值语句	76
3.7.2	转向控制语句	78
3.7.3	断言语句	85
3.8	VHDL 的并发描述语句	86
3.8.1	进程描述语句	87
3.8.2	并行信号赋值语句	88
3.8.3	条件信号赋值语句	88
3.8.4	选择信号赋值语句	89
3.8.5	并行过程调用语句	90

3.8.6 块语句	90
3.9 VHDL 元件声明及例化语句	92
3.9.1 层次化设计	92
3.9.2 元件声明	92
3.9.3 元件例化	93
3.9.4 生成语句	95
3.10 VHDL 的文件操作	96
习题 3	98
第 4 章 数字逻辑单元设计	100
4.1 组合逻辑电路设计	100
4.1.1 基本逻辑门电路设计	100
4.1.2 编码器和译码器设计	101
4.1.3 数据选择器设计	103
4.1.4 数字比较器设计	105
4.1.5 数据运算单元设计	105
4.1.6 总线缓冲器设计	107
4.2 时序逻辑电路设计	109
4.2.1 时钟和复位设计	109
4.2.2 触发器设计	111
4.2.3 锁存器设计	114
4.2.4 计数器设计	114
4.2.5 移位寄存器设计	116
4.3 存储器设计	119
4.3.1 ROM 设计	119
4.3.2 RAM 设计	120
4.3.3 FIFO 设计	121
4.4 有限自动状态机设计	123
4.4.1 有限状态机原理	123
4.4.2 有限状态机分类	124
4.4.3 有限状态机设计	127
习题 4	131
第 5 章 VHDL 高级设计技术	133
5.1 VHDL 代码风格	133
5.1.1 逻辑复制和复用技术	134
5.1.2 并行和流水线技术	135

5.1.3	同步和异步单元处理技术	138
5.1.4	逻辑处理技术	142
5.1.5	模块划分的设计原则	146
5.2	IP 核设计技术	147
5.2.1	IP 核分类	147
5.2.2	IP 核优化	148
5.2.3	IP 核生成	149
5.2.4	IP 核应用	150
习题 5		155
第 6 章	基于 HDL 的设计输入	157
6.1	软件环境	157
6.2	综合工具介绍	159
6.3	工程建立	159
6.4	设计描述	161
6.5	添加设计和检查	162
6.6	创建基于 HDL 的模块	163
6.7	IP 核产生和例化	165
6.7.1	IP 核的生成	165
6.7.2	IP 核的例化	167
习题 6		168
第 7 章	基于原理图的设计输入	169
7.1	工程建立	169
7.2	设计描述	170
7.3	创建原理图模块	171
7.3.1	原理图编辑器操作	171
7.3.2	定义模块符号	172
7.3.3	创建模块符号	175
7.4	创建状态图模块	175
7.4.1	添加状态	175
7.4.2	添加迁移	177
7.4.3	添加行为	177
7.4.4	添加复位条件	177
7.4.5	设计输出和添加	178
7.5	设计完成	178
习题 7		179

第 8 章 设计综合和行为仿真	180
8.1 设计综合	180
8.1.1 行为综合描述	180
8.1.2 基于 XST 的综合概述	181
8.1.3 约束及设计综合的实现	181
8.1.4 RTL 符号的查看	182
8.2 行为仿真的实现	183
8.2.1 生成测试向量	183
8.2.2 基于 ModelSim 行为仿真实现	185
8.2.3 基于 ISE 行为仿真实现	190
习题 8	192
第 9 章 设计实现和时序仿真	193
9.1 实现过程概述及约束	193
9.1.1 实现过程概述	193
9.1.2 建立约束文件	193
9.2 设置实现属性参数	194
9.3 创建分区	195
9.4 创建时序约束	196
9.5 设计翻译	196
9.6 设计约束	197
9.6.1 时序约束	197
9.6.2 管脚和面积约束	200
9.7 设计映射及时序分析	201
9.7.1 设计映射	201
9.7.2 使用时序分析评估块延迟	202
9.8 布局布线验证	204
9.8.1 用 FPGA Editor 验证布局布线	205
9.8.2 评估布局后时序	205
9.8.3 改变分区 HDL	206
9.9 时序仿真实现	208
9.9.1 时序仿真概述	208
9.9.2 使用 ModelSim 进行时序仿真	208
9.9.3 使用 ISE 仿真器进行时序仿真	213
习题 9	216

第 10 章 设计下载和调试	217
10.1 PLD 的配置	217
10.1.1 配置单元	217
10.1.2 配置端口	218
10.1.3 配置电路	218
10.2 创建配置数据	221
10.2.1 配置属性	221
10.2.2 创建 PROM 文件	222
10.3 下载实现过程	224
10.3.1 下载环境	224
10.3.2 下载实现	225
10.3.3 JTAG 诊断	231
10.3.4 建立 SVF 文件	232
10.3.5 其他配置模式	234
10.4 PLD 调试	234
10.4.1 多路复用技术的应用	234
10.4.2 虚拟逻辑分析工具	235
习题 10	235
第 11 章 数字时钟设计及实现	237
11.1 数字时钟的功能要求和结构	237
11.1.1 数字时钟的功能要求	237
11.1.2 数字时钟的整体结构	237
11.2 模块设计	238
11.2.1 数字时钟控制信号	238
11.2.2 控制模块结构	238
11.3 设计实现	240
11.3.1 设计输入	240
11.3.2 设计约束	246
习题 11	247
第 12 章 通用异步接收发送器设计及实现	248
12.1 UART 设计原理	248
12.1.1 UART 原理和设计描述	248
12.1.2 接收模块设计	249
12.1.3 发送模块设计	251

12.1.4	UART 的 VHDL 设计代码	251
12.2	UART 设计验证	258
12.2.1	验证原理	258
12.2.2	验证代码	258
习题 12		261
第 13 章	数字电压表设计及实现	262
13.1	数字电压表的功能要求和结构	262
13.1.1	数字电压表的功能要求	262
13.1.2	数字电压表的整体结构	262
13.2	模块设计	263
13.2.1	数字电压表控制信号	263
13.2.2	ADC 转换原理	263
13.2.3	控制模块结构	265
13.3	设计实现	266
13.3.1	ADC 控制模块原理及实现	266
13.3.2	显示控制模块原理及实现	268
13.3.3	程序包的设计	270
13.3.4	顶层模块设计	272
13.3.5	设计约束文件	274
习题 13		275
参考文献		276

EDA 设计导论

本章主要介绍 EDA 技术综述、PLD 设计方法学、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。这章主要目的是让读者通过对本章内容的学习对 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 and 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,到了 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,最好的办法是自己设计芯片,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,使得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由工具来完成,如可以将用户要求转换为设计技术规范,有效的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,由于 HDL 语言的出现使得 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 工具是以系统机设计为核心,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师在不熟

悉各种半导体工艺的情况下,完成电子系统的设计。

21 世纪开始,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,而是具有强大的信号处理和控制功能。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 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 软件都具有仿真或模拟功能。

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(Field Programmable Gate Array,FPGA)和复杂可编程逻辑器件(Complex Programmable Logic Device,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 在结构上主要分为三个部分,可编程逻辑单元、可编程输入输出单元和可编程连线。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,从而使得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 HDL 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 语言是一种较低级的描述语言,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要稍简单,即经过 RTL 级→门电路级的转化,易于控制电路资源。

(4) 对综合器的要求。VHDL 描述语言层次较高,不易控制底层电路,因而对综合器的性能要求较高,Verilog 和 ABEL 对综合器的性能要求较低。

(5) 支持的 EDA 工具。支持 VHDL 和 Verilog 的 EDA 工具很多,但支持 ABEL 的综合器仅仅 Dataio 一家。

(6) 国际化程度。VHDL 和 Verilog 已成为 IEEE 标准,而 ABEL 正朝国际化标准努力。在新世纪中,VHDL 与 Verilog 语言将承担几乎全部的数字系统设计任务。

3. 软件开发工具

基于高复杂度 PLD 器件的开发,在很大程度上要依靠 EDA 软件完成。PLD 的 EDA 工具以计算机软件为主,将典型的单元电路封装起来形成固定模块并形成标准的硬件开发语言(如 HDL 语言)供设计人员使用。设计人员考虑如何将可组装的软件库和软件包搭建出满足需求的功能模块甚至完整的系统。PLD 开发软件需要自动地完成逻辑编译、化简、分割、综合及优化、布局布线、仿真以及对于特定目标芯片的适配编译和编程下载等工作。典型的 EDA 工具中必须包含两个特殊的软件包,即综合器和适配器。

综合器的功能就是将设计者在 EDA 平台上完成的针对某个系统项目的 HDL、原理图或状态图形描述,针对给定的硬件系统组件,进行编译、优化、转换和综合。

随着开发规模的级数性增长,就必须减短 PLD 开发软件的编译时间,并提高其编译性能以及提供丰富的知识产权 IP(Intellectual Property)核资源供设计人员调用。

此外,PLD 开发界面的友好性以及操作的复杂程度也是评价其性能的重要因素。目前在 PLD 产业领域中,PLD 厂商的开发工具的性能已成为其 PLD 产品能否大规模应用的重要影响因素。只有全面做到芯片技术领先、文档完整和 PLD 开发软件性能优良,芯片提供商才能获得客户的认可。

1) 主流厂家的 EDA 软件工具

目前比较流行的、主流厂家的 EDA 的软件工具有 Xilinx 的 Foundation Series、ISE/ISE-WebPACK Series。Altera 的 MAX+plus II、Quartus II, Lattice 的 ispEXPERT、这些软件的基本功能相同,主要差别是面向的目标器件不一样和性能各有优劣。

(1) Foundation Series。它是 Xilinx 公司集成开发的 EDA 工具。它采用自动化的、完整的集成设计环境。Foundation 项目管理器集成了 Xilinx 实现工具,并包含了强大的 Synopsys FPGA Express 综合系统,是业界最强大的 EDA 设计工具之一。

(2) ISE/ISE-WebPACK Series。它是 Xilinx 公司新近推出的全球性能最高的 EDA 集成软件开发环境(Integrated Software Environment, ISE)。Xilinx ISE 操作简易方便,其提供的各种最新改良功能能解决以往各种设计上的瓶颈,加快了设计与检验的流程,如 Project Navigator(先进的设计流程导向专业管理程式)让设计人员能在同一设计工程中使用 Synplicity 与 Xilinx 的合成工具,混合使用 VHDL 及 Verilog HDL 源程序,让设计人员能使用固有的 IP 与 HDL 设计资源达至最佳的结果。使用者亦可链结与启动 Xilinx Embedded Design Kit (EDK) XPS 专用管理器,以及使用新增的 Automatic Web Update 功能来监视软件的更新状况向使用者发送通知,及让使用者进行下载更新档案,以令其 ISE 的设定维持最佳状态。