

普通高等教育“十三五”电子信息类规划教材

可编程逻辑器件 与 EDA 技术

PROGRAMMABLE LOGIC DEVICES AND EDA TECHNOLOGY

丁山◎编

非
外
借



免费电子课件

普通高等教育“十三五”电子信息类规划教材

可编程逻辑器件与 EDA 技术

丁山 编

机械工业出版社

本书由浅入深系统地介绍了常用的可编程逻辑器件的基本工作原理,详细介绍了 VHDL 硬件描述语言、新一代 FPGA 设计套件 Vivado 的性能和使用方法以及 FPGA 的开发方法。全书内容新颖,举例充实。读者通过本书的学习可以初步掌握 EDA 的基本内容及实用技术。

本书共 14 章,主要内容包括绪论、CPLD 与 FPGA 的结构原理、VHDL 入门基础、VHDL 硬件描述语言、有限状态机设计、VHDL 优化设计、Vivado 集成设计环境导论、Vivado 工程模式下设计基础、创建和封装用户 IP 核、数字电子系统的设计实现、键控流水灯实验设计、抢答器实验设计、数字钟实验设计、UART 实验设计,内容丰富,叙述上浅显易懂,程序实例具有典型性。本书免费提供所有例题的源代码、电子课件,有很大的参考价值,欢迎选用本书作为教材的教师登录 www.cmpedu.com 下载或发邮件到 wangkang_maizi9@126.com 索取。

本书可作为高等院校电子信息类、计算机类等相关专业的教材,也可以作为电子技术工程技术人员的参考用书。

图书在版编目 (CIP) 数据

可编程逻辑器件与 EDA 技术/丁山编. —北京:机械工业出版社, 2017. 11

普通高等教育“十三五”电子信息类规划教材

ISBN 978-7-111-58375-2

I. ①可… II. ①丁… III. ①可编程逻辑器件-高等学校-教材 ②电子电路-电路设计-计算机辅助设计-高等学校-教材 IV. ①TP332.1 ②TN702

中国版本图书馆 CIP 数据核字 (2017) 第 263582 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策划编辑:王 康 责任编辑:王 康 刘丽敏

责任校对:潘 蕊 封面设计:张 静

责任印制:常天培

涿州市京南印刷厂印刷

2018 年 1 月第 1 版第 1 次印刷

184mm×260mm · 18 印张 · 437 千字

0001—3000 册

标准书号:ISBN 978-7-111-58375-2

定价:45.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

服务咨询热线:010-88379833

机工官网:www.cmpbook.com

读者购书热线:010-88379649

机工官博:weibo.com/cmp1952

教育服务网:www.cmpedu.com

封面防伪标均为盗版

金书网:www.golden-book.com

前言

随着半导体产业进入深纳米的时代,可编程逻辑器件向高密度、高速度、低价格方向迅速发展,EDA技术在电子信息、通信、自动控制及计算机应用等领域的重要性日益突出。目前EDA技术已经成为电子信息类专业一门重要的专业基础课程,是电子信息类专业学生必须掌握的专业基础知识和基本技能。为了使學生掌握EDA基本设计工具和设计方法,在EDA开发软件上本书使用在FPGA市场占有率第一的Xilinx公司推出和发布的Vivado设计套件。该套件是一款基于业界标准的开放式开发环境,可以利用Xilinx公司推出的领先一代的硬件、软件和I/O全面可编程的SoC-Zynq7000系列实现数字系统、DSP系统和嵌入式系统的设计。作者力图将EDA技术最新发展成果、现代电子设计最前沿理论和技术、国际上业界普遍接受和认可的EDA软硬件开发平台的使用方法奉献给广大读者。

本书力求全面、实用,对例题做到详细分析和解释,既可以帮助读者学习理解知识和概念,降低学习难度,又具有启发性,帮助读者更加轻松、迅速地理解和掌握本书内容。

本书在内容的组织上共分14章,各章的具体内容如下:

第1章为绪论,主要概述了EDA技术及其重要性,EDA包含的知识体系结构,如HDL、EDA的工作软件等,比较了传统电子设计方法与EDA技术各自的特点。同时对EDA技术的发展历程、特点和优势,以及利用EDA进行工程设计的流程进行了简要介绍。

第2章主要介绍了可编程逻辑器件的基本结构和工作原理,以及相关的编程、测试和配置方法。首先对可编程逻辑器件进行了概述,主要介绍了可编程逻辑器件的发展历史,并对可编程逻辑器件通过不同的划分方式进行了分类;接着重点介绍了高密度可编程逻辑器件CPLD和FPGA的结构原理和工作特点;然后详细地阐述了JTAG边界扫描技术的硬件测试原理,并对CPLD的编程方法和配置方式进行了介绍;最后介绍了本书使用的Basys3开发板。

第3章主要讲述了VHDL语言的基本语法知识,是使用VHDL进行EDA设计的基础。首先介绍了VHDL程序的基本结构,一个完整的VHDL设计由库、程序包、实体、结构体和配置组成,其中实体和结构体是基本组成部分;然后介绍了VHDL语言中的文字规则、数据对象、数据类型和操作符等内容;最后介绍了VHDL中预定义的属性。

第4章主要讲述了VHDL语句的基本内容。VHDL中的语句可以分为两大类:顺序语句和并行语句。在此基础上介绍了各种语句的语法以及使用方法、程序包的构成和配置、子程序的概念及其使用方法,并给出常用设计举例。

第5章主要讲述了有限状态机的基本概念、特点和基本结构等基础内容。在此基础上,对Moore型状态机和Mealy型状态机的结构、特性和设计方法进行了详细的举例说明;然后介绍了状态位置直接输出型编码、顺序编码、枚举类型编码及一位热码编码四种不同的状态编码方式,以及程序直接导引法及状态编码检测法两种安全状态机的设计方法;最后比较全面地对有限状态机进行了介绍。

第6章介绍了EDA的硬件系统设计中VHDL的优化设计。首先介绍了FPGA/CPLD的

资源利用优化。资源优化主要包括资源共享、逻辑优化和串行化。由于对于大多数的设计来说,速度优化比资源优化更重要,所以介绍了速度优化,并依次阐述了流水线设计、寄存器配平、关键路径法、乒乓操作法和加法树法。最后详细介绍了如何排除和避免毛刺或随机干扰信号,主要包括延时方式去毛刺、逻辑方式去毛刺和定时方式去毛刺。

第7章介绍了 Vivado 设计套件的基本知识以及 Vivado 设计套件的界面信息。首先,简单介绍了 Vivado 设计套件的特性;其次介绍了使用 Vivado 设计套件的系统级设计流程;然后介绍了 Vivado 设计套件的安装过程;之后介绍了各个工程文件夹存放的文件类型以及网表文件的相关知识;最后介绍了 Vivado 设计套件的基本界面信息。

第8章介绍了在 Vivado 集成开发环境的工程模式下设计工程的基本设计实现流程。工程模式下的基本设计实现主要步骤包括:创建一个新的设计工程、创建并添加新的设计文件、RTL 详细描述和分析、设计综合、行为级仿真、建立约束、设计实现和分析、静态时序分析、设计时序仿真以及生成编程文件并下载到目标芯片。

第9章介绍了 Vivado 集成设计环境下创建和封装用户 IP 的基本流程。创建和封装用户 IP 的主要步骤包括:创建一个用于定制用户 IP 的工程、设置定制 IP 的库名和目录以及封装 IP。

第10章介绍了在 Vivado 集成开发环境下基于 IP 的简单系统的设计实现流程。基于 IP 的系统设计实现主要步骤包括:创建一个新的设计工程、创建基于 IP 的系统、行为级仿真、设计综合、建立约束、设计实现和分析、静态时序分析、设计时序仿真以及生成编程文件并下载到目标芯片。

第11章介绍了如何使用 Vivado 集成开发环境和 Basys3 开发板设计实现键控流水灯实验的设计。首先介绍了键控流水灯的设计要求和功能描述;其次介绍了键控流水灯的层次化设计方案,主要包括三部分内容,分别是分频模块;流水灯显示模块和按键控制模块;最后对键控流水灯设计进行了硬件测试。

第12章介绍了抢答器实验设计。首先介绍了抢答器的设计要求和功能描述;其次制订了三人抢答器的层次化设计方案,主要包括三部分内容,分别是分频器模块、抢答鉴别器模块和数码管显示模块;然后对抢答器进行了顶层设计和仿真;最后对抢答器的设计进行了硬件测试。

第13章主要介绍了一个简单的数字钟实验设计,首先介绍了数字钟的设计要求和功能描述;然后介绍了数字钟的层次化设计方案,主要包括三部分内容,分别是分频器模块、计数模块和数码管显示模块;之后对数字钟进行了顶层设计和仿真;最后对本次设计进行了硬件测试。

第14章介绍了如何使用 Vivado 集成开发环境和 Basys3 开发板进行简单的 UART 实验设计,实验分为两部分,一部分是接收器的设计实现,另一部分是发送器的设计实现。在两部分中分别介绍了接收器和发送器的层次化设计方案,并对其进行了硬件测试。

本书内容充实,系统全面,重点突出,阐述循序渐进,由浅入深。书中所有例题均在 Vivado 环境下运行通过。本书配有免费的电子课件,欢迎选用本书作为教材的教师登录 www.cmpedu.com 下载或发邮件到 wangkang_maizi9@126.com 索取。

参加本书编写、校对及程序测试工作的还有吴金辉、王辉等,在此表示感谢。

由于作者水平有限,书中难免有错误和不足之处,恳请各位专家和读者批评指正。

编 者

目 录

前 言

第1章 绪论	1
1.1 EDA 技术概要	1
1.1.1 EDA 技术的含义	1
1.1.2 EDA 技术的发展历程	1
1.1.3 EDA 的基本特征	3
1.1.4 EDA 技术的优势	5
1.2 EDA 技术的实现目标	7
1.3 硬件描述语言	8
1.3.1 VHDL	8
1.3.2 Verilog HDL	9
1.3.3 VHDL 和 Verilog HDL 的比较	9
1.4 常用的 EDA 工具	10
1.4.1 设计输入编辑器	11
1.4.2 综合器	11
1.4.3 仿真器	13
1.4.4 适配器	13
1.4.5 编程下载	14
1.5 EDA 的工程设计流程	14
1.6 Vivado 概述	17
1.7 EDA 技术的发展趋势	17
本章小结	19
习题	19

第2章 CPLD 与 FPGA 的结构原理

2.1 PLD 概述	20
2.1.1 PLD 入门	20
2.1.2 常见的 PLD	21
2.1.3 PLD 的优点	22
2.1.4 PLD 的发展趋势	23
2.2 简单 PLD 的结构原理	24
2.2.1 逻辑符号表示方法	25
2.2.2 PROM 的结构原理	25
2.2.3 PLA 的结构原理	26
2.2.4 PAL 的结构原理	26
2.2.5 GAL 的结构原理	31

2.3 CPLD 的结构原理	35
2.3.1 CPLD 的基本结构	35
2.3.2 基于乘积项的可编程逻辑器件	36
2.4 FPGA 的结构原理	39
2.4.1 查找表型 FPGA 的逻辑结构	41
2.4.2 Xilinx 公司 7 系列 FPGA 简介	43
2.4.3 FPGA 与 CPLD 的区别	47
2.5 硬件测试	48
2.6 CPLD/FPGA 的编程与配置	49
2.7 Basys3 开发板介绍	50
2.7.1 电源电路	51
2.7.2 LED 灯电路和数码管电路	51
2.7.3 按键电路和拨码开关电路	52
2.7.4 VGA 显示电路和 I/O 扩展电路	53
本章小结	53
习题	54

第3章 VHDL 入门基础

3.1 VHDL 的基本结构	55
3.1.1 实体	56
3.1.2 结构体	57
3.1.3 库、程序包和配置	57
3.2 VHDL 的文字规则	59
3.2.1 数字型文字	60
3.2.2 字符串型文字	60
3.2.3 标识符	60
3.2.4 下标	61
3.3 VHDL 的数据对象	62
3.3.1 常数	62
3.3.2 变量	62
3.3.3 信号	62
3.3.4 文件	63
3.4 VHDL 的数据类型	63
3.4.1 VHDL 预定义数据类型	64
3.4.2 用户自定义数据类型	67
3.4.3 数据类型间的转换	68

3.5 VHDL 的操作符	69	5.3.1 单进程 Moore 型状态机	120
3.6 VHDL 预定义属性	73	5.3.2 多进程 Moore 型状态机	122
本章小结	75	5.4 Mealy 型状态机的设计	125
习题	76	5.5 状态编码	127
第4章 VHDL 硬件描述语言	77	5.5.1 顺序编码	127
4.1 进程语句	77	5.5.2 枚举类型编码	128
4.2 赋值语句	78	5.5.3 状态位直接输出型编码	128
4.3 顺序描述语句	80	5.5.4 一位热码编码	129
4.3.1 IF 语句	80	5.6 安全状态机设计	129
4.3.2 CASE 语句	82	5.6.1 程序直接导引法	129
4.3.3 LOOP 语句	83	5.6.2 状态编码检测法	130
4.3.4 NEXT 语句和 EXIT 语句	84	本章小结	130
4.3.5 WAIT 语句	85	习题	130
4.3.6 RETURN 语句	86	第6章 VHDL 优化设计	132
4.3.7 NULL 语句	87	6.1 资源优化	132
4.4 元件例化语句	87	6.1.1 资源共享	132
4.5 生成语句	90	6.1.2 逻辑优化	135
4.6 块语句	91	6.1.3 串行化	136
4.7 程序包和配置	93	6.2 速度优化	138
4.7.1 程序包	93	6.2.1 流水线设计	138
4.7.2 配置	94	6.2.2 寄存器配平	141
4.8 子程序	95	6.2.3 关键路径法	142
4.8.1 过程	96	6.2.4 乒乓操作法	142
4.8.2 函数	97	6.2.5 加法树法	144
4.9 其他语句	99	6.3 硬件毛刺剔出	144
4.10 常用设计举例	100	6.3.1 延时方式	145
4.10.1 结构体的三种描述方式	100	6.3.2 逻辑方式去毛刺	147
4.10.2 组合逻辑电路设计	102	6.3.3 定时方式去毛刺	147
4.10.3 时序逻辑电路设计	107	本章小结	149
本章小结	113	习题	149
习题	113	第7章 Vivado 集成设计环境导论	150
第5章 有限状态机设计	114	7.1 Vivado 设计套件	150
5.1 概述	114	7.1.1 单一的、共享的、可扩展的 数据模型	150
5.1.1 状态机的特点	114	7.1.2 标准化 XDC 约束文件 SDC	150
5.1.2 状态机的分类	115	7.1.3 多维度解析布局器	151
5.2 VHDL 状态机的一般形式	116	7.1.4 IP 封装器、集成器和目录	151
5.2.1 一般状态机的结构	116	7.1.5 Vivado HLS	152
5.2.2 状态机的设计流程	118	7.1.6 Tcl 特性	152
5.2.3 状态机的状态转移图描述	118	7.2 Vivado 系统级设计流程	153
5.2.4 状态机的状态说明部分	119	7.3 Vivado 设计套件的安装	154
5.3 Moore 型状态机的设计	120		

7.3.1 下载	154	10.7 创建实现约束	211
7.3.2 安装	154	10.8 设计实现和分析	213
7.4 Vivado 中工程数据的目录结构	157	10.9 静态时序分析	221
7.5 Vivado 网表文件	158	10.10 设计时序仿真	222
7.6 Vivado 集成设计环境主界面	159	10.11 生成编程文件并下载到目标芯片	222
7.6.1 “Quick Start” 分组	159	本章小结	224
7.6.2 “Tasks” 分组	160	习题	224
7.6.3 “Information Center” 分组	160		
7.7 Vivado 设计主界面	161	第 11 章 键控流水灯实验设计	225
7.7.1 流程处理主界面	161	11.1 设计要求	225
7.7.2 工程管理器主界面	162	11.2 功能描述	225
7.7.3 工作区窗口	163	11.3 键控流水灯的层次化设计方案	225
7.7.4 设计运行窗口	163	11.3.1 分频模块	226
本章小结	165	11.3.2 流水灯显示模块	227
		11.3.3 按键控制模块	229
第 8 章 Vivado 工程模式下设计基础	166	11.3.4 键控流水灯的设计	232
8.1 创建一个新的设计工程	166	11.3.5 引脚约束	233
8.2 创建并添加新的设计文件	169	11.3.6 硬件测试	235
8.3 RTL 详细描述和分析	172	本章小结	235
8.4 设计综合和分析	173		
8.5 设计行为级仿真	176	第 12 章 抢答器实验设计	236
8.6 创建实现约束	178	12.1 设计要求	236
8.7 设计实现和分析	181	12.2 功能描述	236
8.8 静态时序分析	182	12.3 抢答器的层次化设计方案	237
8.9 设计时序仿真	185	12.3.1 分频器模块	237
8.10 生成编程文件并下载到目标芯片	185	12.3.2 抢答鉴别器模块的设计	239
本章小结	188	12.3.3 数码管显示模块的设计	242
习题	189	12.3.4 抢答器的顶层设计	244
		12.3.5 引脚约束	246
第 9 章 创建和封装用户 IP 核	190	12.3.6 硬件测试	248
9.1 IP 核概述	190	本章小结	248
9.2 创建用于定制用户 IP 的工程	191		
9.3 设置定制 IP 的库名和目录	192	第 13 章 数字钟实验设计	249
9.4 封装定制 IP 的实现	193	13.1 设计要求	249
本章小结	197	13.2 功能描述	249
习题	197	13.3 数字钟的层次化设计方案	250
		13.3.1 分频器模块	250
第 10 章 数字电子系统的设计实现	198	13.3.2 计数模块的设计	251
10.1 创建一个新的设计工程	198	13.3.3 数码管显示模块的设计	253
10.2 设置调用 IP 的路径	198	13.3.4 数字钟的顶层设计	256
10.3 创建基于 IP 的系统	199	13.3.5 引脚约束	258
10.4 系统行为级仿真	203	13.3.6 硬件测试	259
10.5 RTL 详细描述和分析	205	本章小结	259
10.6 系统设计综合与分析	207		

第14章 UART实验设计	260	14.4 发送器的层次化设计方案	271
14.1 设计要求	260	14.4.1 分频模块	272
14.2 原理描述	260	14.4.2 发送器模块	272
14.3 接收器的层次化设计方案	262	14.4.3 发送器	276
14.3.1 分频模块	263	14.4.4 引脚约束	277
14.3.2 接收器模块	264	14.4.5 硬件测试	278
14.3.3 接收器	268	本章小结	279
14.3.4 引脚约束	269	参考文献	280
14.3.5 硬件测试	270		



1.1 EDA 技术概要

1.1.1 EDA 技术的含义

电子设计自动化 (Electronics Design Automation, EDA) 是一种以计算机为基础的工作平台, 是利用电子技术、计算机技术、智能化技术等多种应用学科的最新成果进行电子产品设计的自动设计技术; 是一种帮助电子设计工程师从事电子元件产品和系统设计的综合技术。

1.1.2 EDA 技术的发展历程

在计算机技术的推动下, 20 世纪末电子技术获得了飞速发展, 现代电子产品几乎渗透于社会的各个领域, 有力地推动了社会生产力的发展和社会信息化程度的提高, 同时又促使现代电子产品性能的进一步提高, 产品更新换代的节奏也越来越快。

EDA 技术作为现代电子设计技术的核心, 它依赖功能强大的计算机, 在 EDA 工具软件平台上, 对以硬件描述语言 (Hardware Description Language, HDL) 为系统逻辑描述手段完成的设计文件, 自动地完成逻辑简化、逻辑分割、逻辑综合、结构综合 (布局布线), 以及逻辑优化和仿真测试等功能, 直至实现既定性能的电子线路系统功能。EDA 技术使得设计者的工作几乎仅限于利用软件的方式, 即利用硬件描述语言 HDL 和 EDA 软件来完成对系统硬件功能的实现。

在现代高新电子产品的设计和生产中, 微电子技术和现代电子设计技术是相互促进、相互推动又相互制约的两个技术环节。前者代表了物理层在广度和深度上硬件电路实现的发展, 后者则反映了现代先进的电子理论、电子技术、仿真技术、设计工艺和设计技术与最新的计算机软件技术有机的融合和升华。因此, 可以说 EDA 技术是这两者的结合。

EDA 技术在硬件方面融合了大规模集成电路制造技术、IC 版图设计技术、ASIC 测试和封装技术、FPGA (Field Programmable Gate Array) 和 CPLD (Complex Programmable Logic Device) 编程下载技术、自动测试技术等; 在计算机辅助技术工程方面融合了计算机辅助设计 (CAD)、计算机辅助制造 (CAM)、计算机辅助测试 (CAT)、计算机辅助工程 (CAE) 技术以及多种计算机语言的设计概念; 而在现代电子学方面则容纳了更多的内容, 如电子线路设计理论、数字信号处理技术、嵌入式系统和计算机设计技术、数字系统建模和优化技术及微波技术等。因此 EDA 技术为现代电子理论和设计的表达与实现提供了可能性。在现代技术的所有领域中, 许多得以飞速发展的科学技术, 多属计算机辅助技术, 而非自动化技

术。显然,最早进入真正的设计自动化的技术领域非电子技术莫属,这就是电子技术始终处于所有科学技术发展的最前列的原因之一。

EDA 技术融合多学科于一体,渗透于各学科之中,已不是某一学科的分支或某种新的技术,而是一门综合性学科。它打破了软硬件间的壁垒,使计算机的软件技术与硬件实现、软件性能和硬件指标、设计效率和产品性能合二为一,它代表了电子设计技术和应用技术的发展方向。

EDA 技术的发展经历了一个由浅入深的过程。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(CAD)、计算机辅助工程设计(CAE)和电子系统设计自动化(ESDA)三个发展阶段。

20 世纪 70 年代到 80 年代初为 CAD 阶段,也是 EDA 技术发展的初级阶段。随着中小规模集成电路的开发应用,传统的手工制图设计印制电路板和集成电路的方法已无法满足设计精度和效率的要求,因此工程师们开始进行二维平面图形的计算机辅助设计,以便解脱复杂、机械的版图设计工作,这就产生了第一代 EDA 工具。这一阶段由于受到计算机的运行速度、存储量和图形功能等方面的限制,电子 CAD 和 EDA 技术没有形成系统,仅是一些孤立的软件程序。这些软件程序在逻辑仿真、印制电路板布局布线和 IC 版图编辑等方面取代了计算机辅助设计的概念。但这些软件一般只有简单的人机交互能力,能处理的电路规模不是很大,计算和绘图的速度都受到限制,而且由于没有采用统一的数据库管理技术,程序之间的数据传输和交换也不方便。

20 世纪 80 年代中后期为 CAE 阶段,也是 EDA 技术发展的中级阶段。这一阶段计算机与集成电路技术得到了高速发展,CAD 软件主要用来实现模拟与数字电路仿真、集成电路的布线布局、IC 版图参数提取与验证、印制电路板的布图与检验、设计文档制作等各设计阶段的自动设计。将这些工具软件集成为一个有机的 EDA 系统,在工作站或超级微机上运行,它具有直观、友好的图形界面,可以用电路原理图的形式输入,以图形菜单的方式选择各种仿真工具和不同的规模功能。每个工具软件都有自己的元器件库,工具之间由统一的数据库进行数据存放、传输和管理。与初期的 CAD 相比,这一阶段的软件除了能进行纯粹的图形绘制功能外,又增加了电路功能设计和结构设计,并且通过电气连接网络表将两者结合在一起,以实现工程设计,这就是计算机辅助工程(CAE)的概念。

20 世纪 90 年代以后是设计自动化阶段,也是 EDA 技术发展的高级阶段。这个时期微电子技术以惊人的速度发展,一个芯片上可以集成几千万只晶体管,超高速数字集成电路的工作效率已经达到 10Gbit/s,射频集成电路的最高工作频率已超过 6GHz,电子系统朝着多功能、高速度、智能化的趋势发展。另一方面,随着集成度的提高,一个复杂的电子系统可以在一个集成电路芯片上实现,这就要求 EDA 系统能够从电子系统的功能和行为描述开始,综合设计出逻辑电路,并自动地映射成可供生产的 IC 版图,这一过程称为集成电路的高级设计。因此,20 世纪 90 年代后的 EDA 系统真正具有了自动化设计能力,EDA 技术被推向成熟和实用,用户只要给出电路的性能指标要求,EDA 系统就能对电路结构和参数进行自动化处理和综合,寻找最佳设计方案,通过自动布局布线功能将电路直接形成集成的电路版图,并对版图的面积及电路延时特性进行优化处理。

EDA 技术在进入 21 世纪后,得到了更大的发展,突出表现在以下几个方面:

(1) 使电子设计成果以自主知识产权(Intellectual Property, IP)的方式得以明确表达

和确认成为可能。系统芯片的设计思想有别于普通的 IC 设计,它是以 IP 核为基础,以硬件描述语言 HDL 为主要设计手段,借助于以计算机为平台的 EDA 工具而进行的。IP 的原来含义是知识产权、著作权等。在 IC 设计领域可将其理解为实现某种功能的设计。美国著名的 Dataquest 咨询公司则将半导体产业的 IP 定义为用于 ASIC 或 FPGA/CPLD 中的预先设计好的电路功能模块。

随着信息技术的飞速发展,用传统的手段来设计高复杂度的系统级芯片,设计周期将变得冗长,设计效率降低。解决这一设计危机的有效方法是复用以前的设计模块,即充分利用已有的或第三方的功能模块作为宏单元,进行系统集成,形成一个完整的系统,这就是集成电路设计复用的概念。这些已有的或由第三方提供的具有知识产权的模块(或内核)称为 IP 核,它在现代 EDA 技术和开发中具有十分重要的地位。

(2) 在仿真验证和设计两方面都支持标准硬件描述语言的功能强大的 EDA 软件不断推出。

(3) 电子技术全方位进入 EDA 时代。除了日益成熟的数字技术外,传统的电路系统设计建模理念发生了重大的改变:软件无线电技术的崛起;模拟电路系统硬件描述语言的表达和设计的标准化;系统可编程模拟器件的出现;软硬件技术;软硬件功能机器结构的进一步融合等。

(4) EDA 使得电子技术领域各学科的界限更加模糊,学科之间更加包容,如模拟与数字、软件与硬件、系统与器件、ASIC 与 FPGA 等。

(5) 更大规模的 FPGA 和 CPLD 器件的不断推出。

(6) 基于 EDA 工具的同于 ASIC 设计的标准单元已涵盖大规模电子系统及复杂 IP 核模块。

(7) 软硬件 IP 核在电子行业的产业领域、技术领域和设计应用领域得到了广泛的应用。

(8) SoC 高效低成本设计技术的成熟。

(9) 系统级、行为验证级硬件描述语言,如 SystemC、System Verilog 等的出现,使复杂电子系统的设计,特别是验证趋于更加高效和简单。

1.1.3 EDA 的基本特征

现代 EDA 技术的基本特征是采用高级语言描述,具有系统级仿真和综合能力、开放式的设计环境、丰富的元器件模型库等。EDA 技术就是依赖功能强大的计算机,在 EDA 工具软件的平台上,对以硬件描述语言 HDL 为系统逻辑描述手段完成的设计文件,自动完成逻辑编译、逻辑化简、逻辑分割、逻辑综合、布局布线和仿真测试,直至实现既定的电子线路系统功能。EDA 技术使得设计者的工作仅限于利用软件的方式,即利用硬件描述语言和 EDA 软件来完成对系统硬件功能的实现。

1. 硬件描述语言设计输入

用硬件描述语言进行电路与系统的设计是当前 EDA 技术的一个重要特征,硬件描述语言输入是现代 EDA 系统的主要输入方式。统计资料表明,在硬件描述语言和原理图两种输入方式中,前者约占 70% 以上,并且这个趋势还在继续增长。与传统的原理图输入设计方法相比,硬件描述语言更适用于规模日益增大的电子系统,它还是进行逻辑综合优化的重要

工具。硬件描述语言使得设计者在比较抽象的层次上描述设计的结构和内部特征，其突出优点是：语言的公开可利用性；设计与工艺的无关性；宽范围的描述能力；便于组织大规模系统的设计；便于设计的复用和继承等。

2. “自顶向下”设计方法

近10年来，电子系统的设计方法发生了很大的变化。过去，电子产品设计的基本思路一直是先选用标准通用集成电路芯片，再用这些芯片和其他元器件自上而下地构成电路、子系统和系统。这样设计出的电子系统所用元器件的种类和数量均较多、体积功耗大、可靠性差。随着集成电路技术的不断进步，半导体集成电路也由早期的单元集成、部件电路集成，发展到整机电路集成和系统电路集成。电子系统的设计方法也由过去的集成电路厂家提供芯片，整机系统用户采用这些芯片组成电子系统的自底向上（Bottom-up）设计方法改变为一种新的自顶向下（Top-down）设计方法。在这种新的设计方法中，由整机系统用户对整个系统进行方案设计和功能划分，系统的关键电路用一片或几片专用集成电路来实现，而且这些专用集成电路是由系统和电路设计师亲自参与设计的，直至完成电路到芯片版图的设计，再交由IC工厂投片加工，或者用可编程ASIC（CPLD和FPGA）现场编程实现。如图1.1所示为电子系统两种不同的设计步骤。

“自顶向下”法是一种概念驱动的设计方法。该方法要求在整个设计过程中尽量运用概念（即抽象）去描述和分析设计对象，而不要过早地考虑实现该设计具体电路、元器件和工艺，以便抓住主要矛盾，避免纠缠在具体细节上，这样才能控制住设计的复杂性。整个设计在概念上的演化从顶层到底层应当逐步由概括到展开，由粗略到精细。只有当整个设计在概念上得到验证与优化后，才能考虑具体问题。

在进行“自顶向下”的设计时，首先从系统级设计入手，在顶层进行功能框图的划分和结构设计；在框图一级进行仿真、纠错，并用硬件描述语言对高层次的系统行为进行描述；在功能一级进行验证，然后用逻辑综合优化工具生成具体的门级逻辑电路的网表，其对应的物理实现级可以是印制电路板或专用集成电路。而“自底向上”的设计方法一般是在系统划分和分解的基础上先进行单元设计，在单元的精心设计后逐步向上进行功能块设计，然后进行子系统的设计，最后完成系统的总体设计。“自顶向下”的设计方法有利于在早期发现结构设计中的错误，提高设计的一次成功率，因而在现代EDA系统中被广泛采用。

3. 逻辑综合与优化

逻辑综合是20世纪90年代电子学领域兴起的一种新的设计方法，是以系统级设计为核心的高层次设计。逻辑综合是将最新的算法与工程界多年积累的设计经验结合起来，自动地将用真值表、状态图或VHDL硬件描述语言等所描述的数字系统转化为满足设计性能指标要求的逻辑电路，并对电路进行速度、面积等方面的优化。

逻辑综合的作用是根据一个系统的逻辑功能与性能要求，在一个包含众多结构、功能和



图1.1 “自顶向下”与“自底向上”设计

性能均已知的逻辑元器件的逻辑单元库的支持下,寻找出一个逻辑网络结构最佳的实施方案。

逻辑综合的过程主要包含以下两个方面。

(1) 逻辑结构的生成与优化:主要是进行逻辑简化与优化,达到尽可能地用较少的元器件和连线形成一个逻辑网络结构(逻辑图),满足系统逻辑功能的要求。

(2) 逻辑网络的性能优化:利用给定的逻辑单元库,对已生成的逻辑网络进行元器件配置,进而估算实现该逻辑网络的芯片的性能与成本。性能主要指芯片的速度,成本主要指芯片的面积与功耗。速度与面积或速度与功耗是矛盾的。这里有一步,允许使用者对速度与面积或速度与功耗相矛盾的指标进行性能与成本的折中,以确定合适的元器件配置,完成最终的、符合要求的逻辑网络结构。

4. 开放性和标准化

开放式的设计环境也称为框架结构(Framework)。框架是一种软件平台结构,它在EDA系统中负责协调设计过程和管理设计数据,实现数据与工具的双向流动,为EDA工具提供合适的操作环境。框架结构的核心是可以提供与硬件平台无关的图形用户界面,工具之间的通信、设计数据和设计流程的管理等,以及各种与数据库相关的服务项目。

任何一个EDA系统只要建立一个符合标准的开放式框架结构,就可以接纳其他厂家的EDA工具一起进行设计工作。框架结构的出现,使国际上许多优秀的EDA工具可以合并到一个统一的计算机平台上,成为一个完整的EDA系统,充分发挥每个设计工具的技术优势,实现资源共享。在这种环境下,设计者可以更有效地运用各种工具,提高设计质量和效率。

近年来,随着硬件描述语言等设计数据格式的逐步标准化,不同设计风格和应用的要求导致各具特色的EDA工具被集成在同一个工作站上,从而使EDA框架标准化。新的EDA系统不仅能够实现高层次的自动逻辑综合、版图综合和测试码生成,而且可以使各个仿真器对同一个设计进行协同仿真,从而进一步提高EDA系统的工作效率和设计的正确性。

5. 库

EDA工具必须配有丰富的库,包括元器件图形符号库、元器件模型库、工艺参数库、标准单元库、可复用的电路模块库、IP库等,才能够具有强大的设计能力和较高的设计效率。

在电路设计的每个阶段,EDA系统需要各种不同层次、不同种类的元器件模型库的支持。例如:原理图输入时需要元器件外形库;逻辑仿真时需要逻辑单元的功能模型库;电路仿真时需要模拟单元和器件的模型库;版图生成时需要使用不同层次和不同工艺的底层版图库;测试综合时需要各种测试向量库等。每一种库又分为不同层次的单元或元素库,例如,逻辑仿真的库又按照行为级、寄存器级和门级分别设库。而VHDL输入所需的库更为庞大和齐全,几乎包含了上述所有库的内容。各种模拟库的规模和功能是衡量EDA工具优劣的一个重要标识。

1.1.4 EDA技术的优势

传统的数字电子系统或IC设计中,手工设计占了较大的比例。手工设计一般先按电子系统的具体功能要求进行功能划分,然后对每个子模块画出真值表,用卡诺图进行手工逻辑

简化, 写出布尔表达式, 画出相应的逻辑线路图, 再据此选择元器件和设计电路板, 最后进行实测与调试。手工设计方法的缺点包括:

- (1) 复杂电路的设计和调试都十分困难。
- (2) 由于无法进行硬件系统仿真, 如果某一过程存在错误, 则查找和修改都十分困难。
- (3) 设计过程中产生大量文档, 不易管理。
- (4) 对于 IC 设计而言, 设计实现过程与具体生产工艺直接相关, 因此可移植性差。
- (5) 只有在设计出样机或生产出芯片后才能进行实测。

相比之下, 采用 EDA 技术进行电子系统的设计有很大的优势。

(1) 用 HDL 对数字系统进行抽象的行为与功能描述以及具体的内容线路结构描述, 从而可以在电子设计的各个阶段、各个层次进行计算机模拟验证, 保证设计过程的正确性, 可以大大地降低设计成本、缩短设计周期。

(2) EDA 工具之所以能够完成各种自动设计过程, 关键是有各类库的支持, 如逻辑仿真时的模拟库、逻辑综合时的综合库、版图综合时的版图库、测试综合时的测试库等。这些库都是 EDA 公司与半导体生产厂商紧密合作、共同开发的。

(3) 某些 HDL 也是文档型的语言 (如 VHDL), 极大地简化了设计文档的管理。

(4) EDA 技术中最为瞩目的功能, 即最具现代电子设计技术特征的功能是日益强大的逻辑设计仿真测试技术。EDA 仿真测试技术只需通过计算机就能对所设计的电子系统从各种不同层次的系统性能特点完成一系列准确的测试与仿真操作, 在完成实际系统的安装后, 还能对系统上的目标器件进行逻辑边界扫描测试。这一切都极大地提高了大规模系统电子设计的自动化程度。

(5) 无论传统的应用电子系统设计得如何完美, 使用了多么先进的功能器件, 都掩盖不了一个无情的事实, 即该系统对于设计者来说, 没有任何自主知识产权可言, 因为系统中的关键性器件往往并非出自设计者之手, 这将导致该系统在许多情况下的应用直接受到限制。基于 EDA 技术的设计则不同, 由于 HDL 表达成功的专用功能设计在实现目标方面有很大的可选性, 它既可以用不同来源的通用 FPGA/CPLD 实现, 也可以直接以 ASIC 来实现, 设计者拥有完全的自主权。

(6) 传统的电子设计方法至今没有任何标准规范加以约束, 因此, 设计效率低、系统性能差、开发成本高、市场竞争小。而 EDA 技术的设计语言是标准化的, 不会由于设计对象的不同而改变; 它的开发工具是规范化的, EDA 软件平台支持任何标准化的设计语言; 它的设计成果是通用性的, IP 核具有规范的接口协议。良好的可移植与可测试性为系统开发提供了可靠的保证。

(7) 从电子设计方法学来看, EDA 技术最大的优势就是能将所有设计环节纳入统一的自顶向下的设计方案中。

(8) EDA 不但在整个设计流程上充分利用计算机的自动设计能力、在各个设计层次上利用计算机完成不同内容的仿真模拟, 而且在系统板设计结束后仍可利用计算机对硬件系统进行完整全面的测试。而传统的设计方法, 如单片机仿真器, 只能在最后完成的系统上进行局部的且仅限于软件的仿真调试, 而在整个设计的过程是无能为力的。至于硬件系统测试, 由于现在的许多系统主板层数多, 而且许多器件是 BGA (Ball - Grid Array) 封装, 所有引脚都在芯片的内部, 焊接后普通的仪器仪表无法接触到所需的信号点, 因此无法测试。

1.2 EDA 技术的实现目标

一般地，利用 EDA 技术进行电子系统设计，最后的目标是完成专用集成电路（ASIC）或印制电路板（PCB）的设计与实现。其中 PCB 设计指的是电子系统的印制电路板设计，从电路原理图到 PCB 上元件的布局、布线、阻抗匹配、信号完整性分析及板级仿真，到最后的电路板机械加工文件生成，这些都需要相应的计算机 EDA 工具软件辅助设计者来完成，这是早期 EDA 技术最基本的应用。ASIC 作为最终的物理平台，集中容纳了用户通过 EDA 技术将电子应用系统的既定功能和技术具体实现的硬件实体。一般而言，专用集成电路就是具有专门用途和特定功能的独立集成电路器件。根据这个定义，作为 EDA 技术最终实现目标的 ASIC，可以通过三种途径完成，如图 1.2 所示。

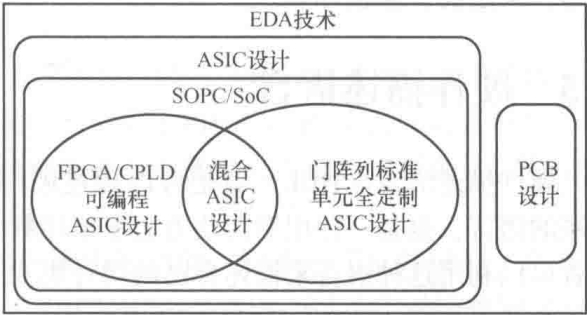


图 1.2 EDA 技术实现目标

1. 超大规模可编程逻辑器件

FPGA 和 CPLD 是实现这一途径的主流器件，其特点是直接面向用户、具有极大的灵活性和通用性、使用方便、硬件测试和实现快捷、开发效率高、成本低、技术维护简单、工作可靠性高等。FPGA 和 CPLD 的应用是 EDA 技术有机融合软硬件电子设计技术、SoC 和 ASIC 设计，以及对自动化设计与自动实现最经典的诠释。由于 FPGA 和 CPLD 的开发工具、开发流程和使用方法与 ASIC 有类似之处，因此这类器件通常也被称为可编程专用 IC 或可编程 ASIC。

2. 半定制或全定制 ASIC

基于 EDA 设计技术的半定制或全定制 ASIC，根据其实实现工艺，可统称为掩膜 ASIC，或直接称 ASIC。可编程 ASIC 与掩膜 ASIC 相比，不同之处在于前者具有面向用户的灵活多样的可编程性。

掩膜 ASIC 大致分为门阵列 ASIC、标准单元 ASIC 和全定制 ASIC。

(1) 门阵列 ASIC 门阵列芯片包括预定制的相连的 PMOS 和 NMOS 晶体管行。设计中，用户可以借助 EDA 工具将原理图或硬件描述语言模型映射为相应门阵列晶体管配置，创建一个指定金属互联路径文件，从而完成门阵列 ASIC 的开发。由于有掩膜的创建过程，门阵列有时也称掩膜可编程逻辑门阵列（MPGA）。但是 MPGA 本身与 FPGA 完全不同，它不是用户可编程的，也不属于可编程逻辑范畴，而是实际的 ASIC。MPGA 出现在 FPGA 之前，FPGA 技术源自 MPGA。

(2) 标准单元 ASIC 目前大部分 ASIC 是使用库中不同大小的标准单元设计的，这类芯片一般称作基于单元的集成电路（CBIC）。在设计者一级，库包括不同复杂性的逻辑元件：SSI 逻辑块、MSI 逻辑块、数据通道模块、存储器、IP 乃至系统级模块。库包含每个逻辑单元在硅片级的完整布局，使用者只需利用 EDA 软件工具与逻辑块描述打交道即可，完全不必关心深层次电路布局的细节。标准单元布局中，所有扩散、接触点、过孔、多晶通道及金属通道都已完全确定。当该单元用于设计时，通过 EDA 软件产生的网表文件将单元布局块

“粘贴”到芯片布局之上的单元行上。标准单元 ASIC 设计与 FPGA 设计的开发流程相近。

(3) 全定制芯片 全定制芯片中,在针对特定工艺建立的设计规则下,设计者对于电路的设计有完全的控制权,如线的间隔和晶体管大小的确定。该领域的一个例外是混合信号设计,使用通信电路的 ASIC 可以定制设计其模拟部分。

3. 混合 ASIC

混合 ASIC (不是指数模混合 ASIC) 主要指既具有面向用户的 FPGA 可编程功能和逻辑资源,同时也含有可方便调用和配置的硬件标准单元模块,如 CPU、RAM、ROM、硬件加法器、乘法器、锁相环等。

1.3 硬件描述语言

硬件描述语言 (HDL) 就是可以描述硬件电路的功能、信号连接关系及定时 (时序) 关系的语言,也是一种用形式化方法来描述数字电路和设计数字系统的语言。数字系统的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 EDA 工具进行仿真,自动综合到门级电路,再用 ASIC 或 FPGA 实现其功能。

HDL 的发展至今已有 30 多年的历史,它是 EDA 技术的重要组成部分,也是 EDA 技术发展 to 高级阶段的一个重要标志。目前已经存在许多硬件描述语言,其中 VHDL 和 Verilog HDL 是影响最为广泛的两种,并已成为 IEEE 的工业标准硬件描述语言,得到众多 EDA 公司的支持,在电子工程领域,已经成为事实上的通用硬件描述语言。

1.3.1 VHDL

VHDL 诞生于 1983 年,由美国国防部 (DOD) 发起创建。后来 IEEE (The Institute of Electrical and Electronics Engineers) 对其进一步发展,于 1987 年作为“IEEE 标准 1076”发布,从而正式成为硬件描述语言的业界标准之一。随着 VHDL 标准版本 (IEEE Std 1076) 的公布,各 EDA 公司相继推出了自己的 VHDL 设计环境,或宣布自己的设计工具可以使用和支持 VHDL。此后,VHDL 在电子设计领域得到了广泛应用,并逐步取代了原有的非标准硬件描述语言。1993 年,IEEE 对 VHDL 进行了修订,从更高的抽象层次和系统描述能力上扩展了 VHDL 的内容,公布了新版本 VHDL,即 IEEE1076—1993 版本。现在公布的最新 VHDL 标准版本是 IEEE1076—2008。

VHDL 主要用于描述数字系统的结构、行为、功能和接口。除了含有许多具有硬件特征的语句外,VHDL 的语言形式和描述风格与句法与一般的计算机高级语言十分类似。应用 VHDL 进行工程设计的优点是多方面的,具体如下。

(1) 与其他的硬件描述语言相比,VHDL 具有更强的行为描述能力,从而决定了它成为系统设计领域最佳的硬件描述语言。强大的行为描述能力是避开具体的器件结构,从逻辑行为上描述和设计大规模电子系统的重要保证。

(2) VHDL 最初是作为一种仿真标准格式出现的,因此 VHDL 既是一种硬件电路描述和设计语言,也是一种标准的网表格式,还是一种仿真语言。它有丰富的仿真语句和库函数,设计者可以在任何系统的设计早期随时对设计进行仿真模拟,查验所设计系统的功能特性,从而对整个工程设计的结构和功能的可行性做出决策。