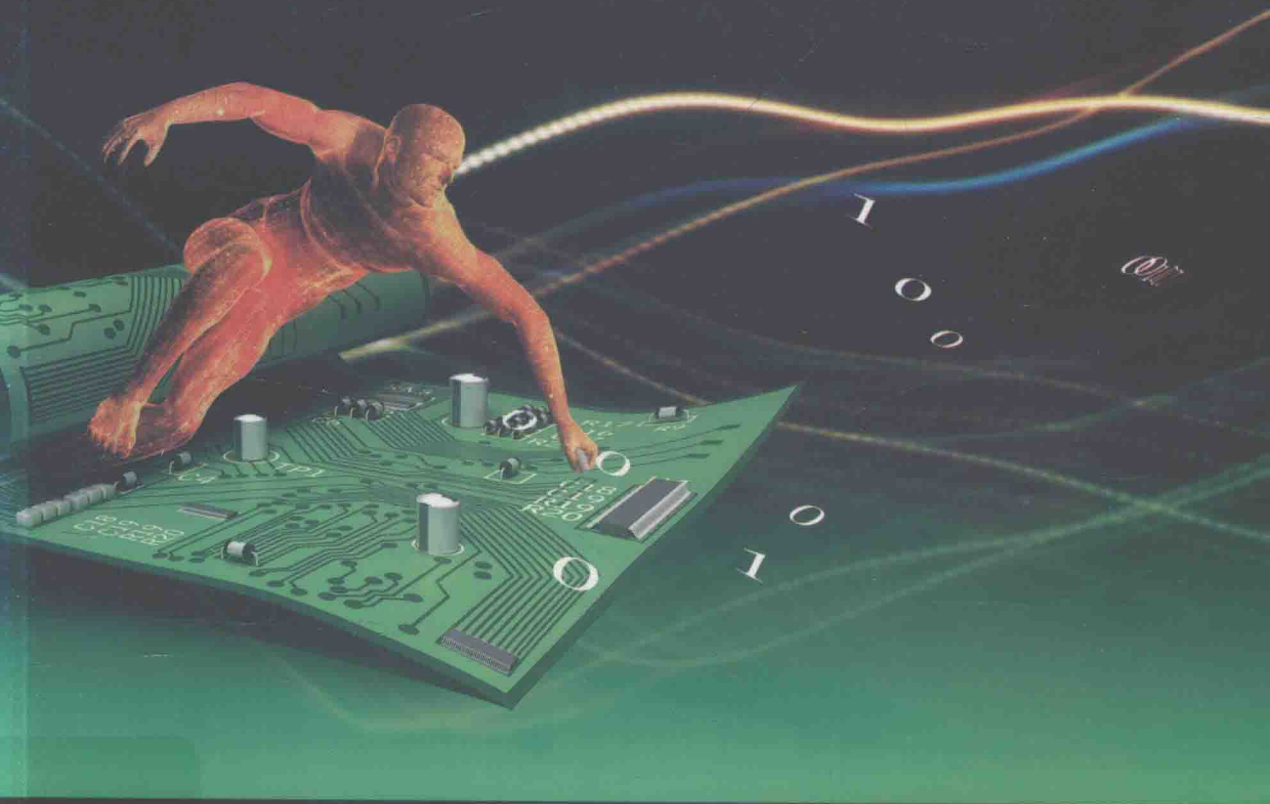


Quartus II
FPGA/CPLD



实例讲解 基于Quartus II的 FPGA/CPLD 数字系统设计快速入门

赵艳华 温利 佟春明 编著



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

实例讲解

基于 Quartus II 的 FPGA/CPLD

数字系统设计快速入门

赵艳华 温 利 佟春明 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书系统介绍了基于 Quartus II 9.0 的 FPGA/CPLD 数字系统设计, 内容包括 FPGA/CPLD 基础知识、VHDL 基本语法介绍和设计实例、复杂时序逻辑的状态机设计、结合硬件实验设备的基本设计实例、Quartus II 9.0 的开发流程和使用技巧、综合性设计实例、EDA 实验系统以及 VHDL 硬件描述语言和 Verilog HDL 语言的语法要素和语法格式, 供读者在学习和实验过程中速查参考。

本书适合从事 FPGA/CPLD 数字系统开发的工程技术人员阅读使用, 也可作为高等院校相关专业的教学用书。

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

实例讲解基于 Quartus II 的 FPGA/CPLD 数字系统设计快速入门/赵艳华, 温利, 佟春明编著. —北京: 电子工业出版社, 2017. 1

ISBN 978-7-121-30156-8

I. ①实… II. ①赵… ②温… ③佟… III. ①可编程序逻辑阵列-系统设计 IV. ①TP332.1

中国版本图书馆 CIP 数据核字 (2016) 第 251746 号

策划编辑: 张 剑 (zhang@phei.com.cn)

责任编辑: 苏颖杰

印 刷: 涿州市京南印刷厂

装 订: 涿州市京南印刷厂

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 × 1 092 1/16 印张: 21.25 字数: 544 千字

版 次: 2017 年 1 月第 1 版

印 次: 2017 年 1 月第 1 次印刷

印 数: 3 000 册 定价: 59.00 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, 88258888。

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

本书咨询联系方式: zhang@phei.com.cn。

前 言

随着 EDA 技术的发展, CPLD/FPGA 器件在电子信息、通信、自动控制及计算机领域的应用已日渐普及。CPLD/FPGA 器件的运用, 能够提升数字系统设计的集成度和自由度, 并且有助于缩短开发周期。熟练运用 CPLD/FPGA 器件成为电子类工程师的必备技能。

EDA 技术的初学者面临着软件操作复杂、理论知识和语法知识均需储备、实验开展需要熟悉硬件特征等问题, 一本能够辅助其顺利开展入门学习与实践的参考书籍显得尤为重要。

本书第 1 章为 EDA 技术和 CPLD/FPGA 基本知识概述; 第 2~4 章为 VHDL 基本语法介绍和设计实例讲解; 第 5 章为结合硬件实验设备进行的基本设计实例; 第 6 章是针对复杂时序逻辑的状态机设计方法介绍; 第 7~9 章详细介绍了 Quartus II 9.0 的开发流程和使用技巧, 可供初学者进行基础操作和入门学习; 第 10 章为综合性设计实例; 第 11 章对康芯 KX-DN 系列 EDA 实验系统进行简要介绍。附录简明扼要地列出了 VHDL 硬件描述语言和 Verilog HDL 语言的语法要素和语法格式, 供读者在学习和实验过程中速查参考。

本书针对 EDA 的初步学习的特点与需求, 简明扼要地介绍了 EDA 软件 Quartus II 9.0 的开发流程。除了基本操作流程, 还对设计中常见的软件问题进行了说明, 能够帮助读者尽快掌握软件使用方法。语法介绍方面, 力求语言通俗易懂, 简单明了, 易于初学者快速掌握。书中所选实验实例均为易于进行测试和验证, 具有一定趣味性的实验项目。

囿于作者水平, 书中难免疏漏和不妥之处, 恳请读者不吝指正。

本书面向的是电子设计自动化 (CPLD/FPGA 应用) 领域的学生和工程技术人员, 包括:

- Quartus II 9.0 软件的初级用户;
- CPLD/FPGA 的入门应用者;
- 具有一定基础知识的数字系统设计爱好者;
- 高等院校相关专业的在校学生。

为了方便读者学习, 书中所有实例和练习的源文件, 以及用到的素材都能够从零点工作室网站下载。

本书由赵艳华、温利、佟春明编著。其中第 1 章由杨丽丽、佟春明、龚丽农、温利编写, 其余章节由赵艳华编写, 全书由赵艳华统稿。参加本书编写的还有管殿柱、宋一兵、王献红、李文秋、谈世哲、管玥、姜安宝、赵景波和周鹏。

感谢您选择了本书, 希望我们的努力对您的工作和学习有所帮助, 也希望您把对本书的意见和建议告诉我们。

编著者

目 录

第 1 章 可编程逻辑器件与 EDA 技术概述	1
1.1 EDA 技术概述	1
1.1.1 EDA 技术的起源及发展	1
1.1.2 EDA 技术的主要内容	4
1.1.3 大规模可编程逻辑器件	4
1.1.4 硬件描述语言 (HDL)	6
1.1.5 软件开发工具	6
1.1.6 实验开发系统	9
1.1.7 EDA 技术的发展趋势	10
1.2 EDA 设计流程	11
1.2.1 EDA 设计过程	11
1.2.2 EDA 设计具体实现方法	14
1.3 CPLD/FPGA 基本原理	17
1.3.1 可编程逻辑器件的发展	17
1.3.2 可编程逻辑器件的分类	18
1.3.3 FPGA 的基本结构	19
1.3.4 CPLD 的基本结构	20
1.3.5 CPLD 和 FPGA 的比较	20
1.3.6 CPLD 基本原理	21
1.3.7 FPGA 结构原理	23
1.4 CPLD/FPGA 器件的应用选择	25
思考与练习	27
第 2 章 VHDL 语言基础	28
2.1 VHDL 语言简介	28
2.2 VHDL 基本程序结构	29
2.2.1 实体 (Entity)	30
2.2.2 结构体 (Architecture)	30
2.2.3 库 (Library) 和程序包 (Package)	30
2.2.4 配置 (Configuration)	30
2.3 VHDL 语言描述方式	31
2.3.1 行为 (Behavior) 描述方式	32
2.3.2 数据流 (Dataflow) 描述方式	33
2.3.3 结构 (Structure) 描述方式	33
2.4 VHDL 程序描述	33
2.4.1 实体声明 (Entity Declaration)	34

2.4.2	结构体声明与描述 (Architecture Declaration and Description)	35
2.4.3	程序包 (Package)	36
2.4.4	配置 (Configuration)	38
2.4.5	库 (Library)	39
2.5	VHDL 数据对象 (Objects)	39
2.5.1	信号声明与使用 (Signal)	40
2.5.2	常量声明与使用 (Constants)	40
2.5.3	变量声明与使用 (Variables)	41
2.5.4	数据对象的比较	42
	思考与练习	43
第3章	VHDL 语法要素	44
3.1	VHDL 基本数据类型	44
3.1.1	标量数据类型	44
3.1.2	复合数据类型	47
3.1.3	数据类型转换	48
3.2	VHDL 文字规则	51
3.2.1	数值表达	51
3.2.2	标识符	52
3.3	VHDL 操作符	53
3.3.1	操作符的类型	53
3.3.2	操作符的优先级	54
	思考与练习	55
第4章	VHDL 语句	57
4.1	VHDL 并行语句 (Concurrent Statements)	57
4.1.1	并行信号赋值语句	57
4.1.2	进程语句	61
4.1.3	块语句 (Block Statement)	63
4.1.4	元件例化语句 (Component Instantiation)	66
4.1.5	过程调用语句 (Procedure Call)	70
4.1.6	生成语句 (Generate Statement)	72
4.2	VHDL 顺序语句 (Sequential Statements)	76
4.2.1	信号赋值和变量赋值语句 (Signal and Variable Assignments)	76
4.2.2	IF 语句 (IF Statement)	77
4.2.3	CASE 语句 (CASE Statement)	79
4.2.4	NULL 语句	80
4.2.5	LOOP 语句	80
4.2.6	NEXT 语句	83
4.2.7	EXIT 语句	83
4.2.8	WAIT 语句	84



4.2.9 RETURN 语句	85
4.3 其他语句结构	86
4.3.1 子程序 (Subprogram)	86
4.3.2 函数 (Function)	86
4.3.3 过程 (Procedure)	92
4.3.4 子程序重载 (Subprogram Overload)	94
4.3.5 预定义属性 (Predefined Attributes)	95
思考与练习	99
第5章 基本逻辑电路设计实例	103
5.1 多路选择器设计	103
5.1.1 4选1多路选择器	103
5.1.2 数据分配器	105
5.2 译码/编码器设计	106
5.2.1 3-8译码器	106
5.2.2 7段数码管译码器	108
5.2.3 优先编码器	109
5.2.4 二-十进制码制转换	110
5.3 计数器设计	112
5.3.1 基本计数器	112
5.3.2 可逆计数器	114
5.3.3 可预置计数器	115
5.4 微分电路设计	116
5.4.1 延迟电路	116
5.4.2 微分电路	117
5.4.3 同步计数器	119
5.5 分频电路设计	120
5.5.1 整数分频电路	120
5.5.2 2.5分频电路	121
5.6 键盘去抖动电路设计	123
5.7 计数与显示电路设计	125
思考与练习	129
第6章 状态机设计	131
6.1 状态机的设计与实现	131
6.1.1 状态机的优点	131
6.1.2 VHDL 语言中的状态机设计种类	131
6.1.3 状态机的结构	132
6.2 状态机设计举例	133
6.2.1 交通灯控制器的逻辑实现	133
6.2.2 单进程 Moore 状态机	135

6.2.3 Mealy 型有限状态机	137
6.3 状态机综合设计实例	139
思考与练习	142
第7章 Quartus II 软件操作指南	143
7.1 Quartus II 9.0 简介	143
7.2 Quartus II 设计流程	144
7.2.1 建立新工程	144
7.2.2 设计输入	148
7.2.3 分析与综合	151
7.2.4 适配 (Fitter)	153
7.2.5 全程编译 (Start Simulation)	153
7.2.6 时序仿真	153
7.2.7 电路观察器 (RTL Viewer)	160
7.2.8 打开原有工程	160
7.3 引脚分配与下载	160
7.3.1 引脚分配	160
7.3.2 编程与配置	162
7.4 Project Navigator (工程导航) 与工程管理	165
7.4.1 【Hierarchy】标签页	166
7.4.2 【Files】标签页	168
7.4.3 工程文件管理	169
思考与练习	172
第8章 Quartus II 应用技巧	173
8.1 文本编辑器	173
8.1.1 【Files】菜单	173
8.1.2 文本编辑器编辑工具	174
8.2 原理图编辑器 (Block Diagram/Schematic File)	178
8.2.1 原理图编辑工具栏	179
8.2.2 添加原理图符号	180
8.2.3 导线绘制与命名	182
8.2.4 视图切换	185
8.3 波形编辑器	187
8.3.1 波形编辑器界面	187
8.3.2 波形编辑工具栏	189
8.3.3 仿真设置	192
8.4 用原理图输入法进行设计	194
8.5 资源分配编辑器 (Assignments Editor)	201
8.5.1 用户界面和主要功能	201
8.5.2 引脚规划器	203



8.6 工程设置	205
思考与练习	211
第9章 Quartus II 综合应用示例	212
9.1 LPM 定制与应用	212
9.2 简易正弦信号发生器设计	216
9.2.1 正弦信号发生器的构成	217
9.2.2 定制波形信息初始化文件	217
9.2.3 定制 LPM_ROM	219
9.2.4 原理图方式设计顶层实体	224
9.2.5 用例化语句完成顶层实体设计	225
9.3 SignalTap II 嵌入式逻辑分析仪的应用	226
9.3.1 SignalTap II 文件的建立	226
9.3.2 逻辑分析仪的使用操作	228
思考与练习	233
第10章 综合设计项目	234
10.1 数字钟设计	234
10.1.1 计时模块	234
10.1.2 显示控制	237
10.2 密码锁设计	241
10.3 电子乐曲播放器设计	246
10.3.1 音阶分频系数	247
10.3.2 音阶频率的产生	248
10.3.3 音长与乐谱的确定	250
10.3.4 乐谱播放控制	253
10.3.5 顶层设计	253
10.3.6 实现乐曲演奏的另一种方法	255
10.4 简易电子琴设计	264
10.5 简易信号发生器	267
10.6 投球游戏设计	274
10.6.1 游戏设计实现	275
10.6.2 自定义程序包	275
10.6.3 游戏控制器设计	276
10.6.4 游戏顶层设计	280
10.7 DS18B20 温度采集控制器	284
思考与练习	294
第11章 EDA/SOC 实验系统使用说明	295
11.1 教学实验系统原理与使用介绍	295
11.1.1 B 类插座实验模块可接插的 10 针双插座	296
11.1.2 主系统其他接口说明	297

11.2 主控板模块介绍	297
11.3 实验系统各模块简介	299
11.3.1 4×4 16 键键盘	299
11.3.2 4×4+8 个单脉冲综合键盘模块	300
11.3.3 1 个 7 段数码显示模块和 4 个动态扫描显示模块	300
11.3.4 7 位数码管串行静态显示模块	301
11.3.5 32 位输出显示 HEX 模块	302
11.3.6 32 位输入显示 HEX 模块	303
11.3.7 24 位综合键盘模块	303
11.3.8 交通灯显示模块	303
11.3.9 动态扫描数码显示模块	303
11.3.10 点阵式 128×64 液晶显示模块	304
11.3.11 字符式 20×4 液晶显示模块	304
11.3.12 800×480 数字 TFT 彩屏液晶显示模块	304
11.3.13 双通道 DAC 和 ADC 标准模块	305
11.3.14 高速 A/D 和双通道 D/A 模块	306
11.3.15 高速 12 位 SPI 串行双 ADC 模块	307
11.3.16 8 位+16 位高分辨率 ADC 模块	308
11.3.17 SPI 串行接口高速 ADC+DAC 模块	308
11.3.18 SD+PS2+RS232+VGA 显示接口模块	308
11.3.19 电动机接口模块	309
11.3.20 语音+数码动态扫描模块	309
附录	310
附录 A VHDL 语句格式速查	310
附录 B Verilog HDL 语句格式速查	322
参考文献	329

第 1 章 可编程逻辑器件与 EDA 技术概述



可编程逻辑器件 (Programmable Logic Device) 是现代电子元件的一个重要门类, 以其为应用目标的 EDA 技术快速发展, 在电子产品设计领域占用重要位置。EDA 技术在通信、国防、航天、医学、工业自动化、计算机应用、仪器仪表等领域的电子系统设计工作中日益普及, 高新技术项目的开发也更加依赖于 EDA 技术的应用。即便是普通的电子产品开发, EDA 技术也发挥了越来越重要的作用, 常轻松突破一些原来的技术瓶颈, 大大缩短产品的开发周期、提高产品的性价比。EDA 技术已经成为电子设计领域中极其重要的组成部分。

1.1 EDA 技术概述

电子设计自动化 (Electronic Design Automation, EDA) 技术是 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 技术在硬件实现方面融合了大规模集成电路制造技术、IC 版图设计、专用集成电路 (Application Specific Integrated Circuit, ASIC) 测试和封装、现场可编程门阵列 (Field Programmable Gate Array, FPGA) / 复杂可编程逻辑器件 (Complex Programmable Logic Device, CPLD) 编程下载和自动测试等技术。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、数字系统建模和优化技术及长线技术理论等; 以计算机为工具, 在 EDA 软件平台上, 根据硬件描述语言 HDL 完成的设计文件, 能自动地完成用软件方式描述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、布局布线、逻辑仿真, 直至完成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。设计者的工作仅限于利用软件的方式来完成对系统硬件功能的描述, 在 EDA 工具的帮助下, 应用相应的 CPLD/FPGA 器件, 就可以得到最后的设计结果。尽管目标系统是硬件, 但整个设计和修改过程如同完成软件设计一样方便和高效。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。

1.1.1 EDA 技术的起源及发展

微电子技术的进步主要表现在大规模集成电路加工技术即半导体工艺技术的发展上, 使得表征半导体工艺水平的线宽已经达到了 60nm, 并还在不断地缩小, 而在硅片单位面积上集成了更多的晶体管。集成电路设计正在不断地向超大规模、极低功耗和超高速的方向发展, ASIC 的设计成本不断降低。

现代电子设计技术的核心已日趋转向基于计算机的技术。EDA 技术就是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, 自动地完成逻辑编译、化简、分割、综合、布局布线以及逻辑优化和仿真测试, 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, 这是电子设计技术的一个巨大进步。

正因为 EDA 技术丰富的内容以及与电子技术各学科领域的相关性, 其发展的历程同大规模集成电路设计技术、计算机辅助工程、可编程逻辑器件, 以及电子设计技术和工艺的发展是同步的。就过去 30 多年电子技术的发展历程, 可将 EDA 技术的发展大致分为以下 4 个阶段。

1) 第一阶段 20 世纪 70 年代的计算机辅助设计 CAD 阶段。早期的电子系统硬件设计采用的是分立元器件, 后来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。此时的硬件设计大量选用中小规模标准集成电路, 人们将这些元器件焊接到电路板上, 制作成初级的电子系统, 对电子系统的调试是在组装好的印制电路板 (Printed Circuit Board, PCB) 上进行的。由于传统的手工布图方法无法满足产品复杂性的要求, 工作效率也不高。此时, 可编程逻辑技术及其器件问世, 计算机作为一种运算工具在科研领域得到广泛应用。CAD 的概念已见雏形, 这一阶段人们开始利用计算机取代手工劳动, 辅助进行集成电路版图编辑、PCB 布局布线等工作。

2) 第二阶段 20 世纪 80 年代的计算机辅助工程设计 CAE 阶段。这段时期, 随着电子工艺的发展, 集成上万只的微处理器、集成几十万直至上百万单元的随机存储器和只读存储器相继出现。同时, 支持定制单元电路设计的硅编程、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等一系列微结构和微电子学研究成果都为电子系统的设计提供了新思路。

与此同时, 集成电路设计进入了互补场效应管 (CMOS) 时代, 复杂可编程逻辑器件进入商业应用, 相应的辅助设计软件投入使用。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。EDA 工具以逻辑模拟、定时分析、故障仿真、自动布局和布线为核心, 重点解决电路设计未完成前的功能检测等问题。有了这些工具的支持, 设计师能在产品制作之前预知产品的功能和性能, 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。

在 20 世纪 80 年代末, 出现了 FPGA, 使得 CAE 和 CAD 技术的应用更为广泛, 它们在 PCB 设计方面的原理图输入、自动布局布线及 PCB 分析, 以及逻辑设计、逻辑仿真等方面担任了重要的角色。特别是各种硬件描述语言的出现、应用和标准化方面的重大进步, 为电子设计自动化必须解决的电路建模、标准文档及仿真测试奠定了基础。

20 世纪 80 年代出现的具有自动综合能力的 CAE 工具替代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。尤其到 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等; CAE 阶段的 EDA 工具不仅为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便利。但具体化的元器件图形制约了优化设计, 大部分从原理图出发的 EDA 工具仍然不能适应复杂的电子系统设计要求。

3) 第三阶段 即 20 世纪 90 年代电子系统设计自动化 EDA 阶段。在此阶段, 随着硬件描述语言的标准化进一步确立, 计算机辅助工程、辅助分析和辅助设计在电子技术领域获得



更加广泛的应用。与此同时, 电子技术在通信、计算机及家电产品生产中的市场需求和技术需求, 极大地推动了全新的电子设计自动化技术的应用和发展, 特别是集成电路设计工艺步入了超深亚微米阶段, 百万门以上的大规模可编程逻辑器件的陆续面世, 以及基于计算机技术的面向用户的低成本大规模 ASIC 设计技术的应用, 促进了 EDA 技术的形成。更为重要的是各 EDA 公司致力于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 工具软件的研究, 更有效地将 EDA 技术推向成熟和实用。

设计师逐步从使用硬件转向了设计硬件, 从单个电子产品开发转向系统级电子产品开发, 即片上系统设计 (System On a Chip, SOC)。EDA 工具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。此时的 EDA 工具不仅具有电子系统设计的能力, 同时还能够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 以及 Verilog-HDL 等; 同时还具有各种工艺的元件库。

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, 设计者通过设计芯片实现电子系统功能。该阶段的 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, 设计师可以在较短的时间内掌握 EDA 工具, 通过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百万门 ASIC 和集成系统的设计与验证。

4) 第四阶段 即进入 21 世纪 EDA 进一步发展阶段。进入 21 世纪后, EDA 技术得到了更大的发展, 突出表现在以下几个方面。

- ⑤ 在 FPGA 上实现数字信号处理 (DSP) 应用成为可能, 用纯数字逻辑进行 DSP 模块的设计, 使得高速 DSP 实现成为现实, 并有力地推动了软件无线电技术的实用化和发展。基于 FPGA 的 DSP 技术, 为高速数字信号处理算法提供了实现途径。
- ⑤ 嵌入式处理器软核的成熟, 使得 SOPC (System On a Programmable Chip) 步入大规模应用阶段, 在一片 FPGA 上实现一个完备的数字处理系统成为可能。
- ⑤ 在仿真和设计两方面支持标准硬件描述语言的功能强大的 EDA 软件不断推出。
- ⑤ 电子技术领域全方位融入 EDA 技术, 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变化, 如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、数字信号处理和图像处理的全硬件实现方案的普遍接受以及软硬件技术的进一步融合等。
- ⑤ EDA 使得电子领域各学科的界限更加模糊, 更加互为包容, 如模拟与数字、软件与硬件、系统与元器件、ASIC 与 FPGA、行为与结构等。
- ⑤ 基于 EDA 并用于 ASIC 设计的标准单元已涵盖了大规模电子系统及复杂 IP 核模块。
- ⑤ 软硬知识产权 (Intellectual Property, IP) 核在电子行业的产业领域广泛应用。
- ⑤ SOC 高效低成本设计技术的成熟。
- ⑤ 系统级、行为验证级硬件描述语言 (如 System C) 的出现, 使复杂电子系统的设计和验证趋于简单。

1.1.2 EDA 技术的主要内容

EDA 技术涉及面很广，内容丰富，从教学和实用的角度看，主要应掌握如下四个方面的内容：

- ☺ 大规模可编程逻辑器件；
- ☺ 硬件描述语言；
- ☺ 软件开发工具；
- ☺ 实验开发系统。

其中，大规模可编程逻辑器件是利用 EDA 技术进行电子系统设计的载体；硬件描述语言是利用 EDA 技术进行电子系统设计的主要表达手段；软件开发工具是利用 EDA 技术进行电子系统设计的智能化的自动设计工具；实验开发系统则是利用 EDA 技术进行电子系统设计的下载工具及硬件验证工具。

1.1.3 大规模可编程逻辑器件

可编程逻辑器件 (Programmable Logic Device, 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。FPGA 和 CPLD 的功能基本相同，只是实现原理略有不同，所以有时可以忽略这两者的区别，统称为可编程逻辑器件或 CPLD/FPGA。PLD 是电子设计领域中最具活力和发展前途的一项技术，PLD 能完成任何数字器件的功能。PLD 如同一张白纸或是一堆积木，工程师可以通过传统的原理图输入法，或是硬件描述语言自由地设计一个数字系统，通过软件仿真，我们可以事先验证设计的正确性。在 PCB 完成以后，还可以利用 PLD 的在线修改能力，随时修改设计而不必改动硬件电路。使用 PLD 来开发数字电路，可以大大缩短设计时间，减少 PCB 面积，提高系统的可靠性。PLD 的这些优点使得 PLD 技术在 20 世纪 90 年代以后得到飞速的发展，同时也大大推动了 EDA 软件和硬件描述语言 (HDL) 的进步。

目前 FPGA 和 CPLD 器件的应用已经十分广泛，随着 EDA 技术的发展将成为电子设计领域的重要角色。国内市场份额较大的 CPLD/FPGA 公司，主要有 Xilinx、Altera、Lattice 等。

美国 Xilinx 公司 1984 年首先推出了世界上第一片现场可编程门阵列 FPGA (Field Programmable Gate Array)。之后，从 XC3000、XC4000 到 Spartan - II、Spartan - IIE 和 Virtex - II、Virtex - II Pro，Xilinx 公司不断推陈出新，一直保持着 FPGA 领域的全球领先地位。

Xilinx 公司的主流 PLD 产品包括：XC9500 Flash 工艺 PLD，常见型号有 XC9536、XC9572、XC95144 等，型号后两位表示宏单元数量；CoolRunner - II 系列 1.8V 低功耗 PLD 产品。

2015 年 12 月英特尔公司斥资 167 亿美元收购了 Altera 公司。Altera 公司成为其可编程解决方案事业部。Altera 公司的 CPLD 器件为 MAX II 系列。MAX II 是新一代 PLD 器件，0.18 μm flash 工艺，2004 年底推出，采用 FPGA 结构，配置芯片集成在内部，和普通 PLD 一样上电即可工作。其容量比上一代大大增加，内部集成一片 8kbit 串行 EEPROM，增加很多功能。MAX II 采用 2.5V 或者 3.3V 内核电压，MAX II G 系列采用 1.8V 内核电压。

Altera 公司的 FPGA 器件大致分三个系列，一是低端的 CYCLONE，二是高端的 STRATIX，三是介于二者之间可以方便 ASIC 化的 ARRIVA 系列。

Altera 公司的主流 FPGA 分为两大类，一种侧重低成本应用，容量中等，性能可以满足



一般的逻辑设计要求,如 Cyclone 至 Cyclone V;还有一种侧重于高性能应用,容量大,性能能满足各类高端应用,如 Stratix 至 Stratix V 等,用户可以根据自己实际应用要求进行选择。在性能可以满足的情况下,优先选择低成本器件。

- ⊙ Cyclone (飓风): Altera 公司的中等规模 FPGA,2003 年推出,0.13 μm 工艺,1.5V 内核供电,与 Stratix 结构类似,是一种低成本 FPGA 系列,其配置芯片也改用全新的产品。
- ⊙ Cyclone II: Cyclone 的下一代产品,2005 年开始推出,90nm 工艺,1.2V 内核供电,属于低成本 FPGA,性能和 Cyclone 相当,提供了硬件乘法器单元。
- ⊙ Cyclone III FPGA 系列:2007 年推出,采用台积电 (TSMC) 65nm 低功耗 (LP) 工艺技术制造,以相当于 ASIC 的价格实现了低功耗。
- ⊙ Cyclone IV FPGA 系列:2009 年推出,60nm 工艺,面向对成本敏感的大批量应用,帮助用户满足越来越高的带宽需求,同时降低了成本。
- ⊙ Cyclone V FPGA 系列:2011 年推出,28nm 工艺,实现了业界最低的系统成本和功耗,其性能水平使得该器件系列成为突出大批量应用优势的理想选择。与前几代产品相比,它具有高效的逻辑集成功能,提供集成收发器型号,总功耗降低了 40%,静态功耗降低了 30%。
- ⊙ Stratix: Altera 公司的大规模高端 FPGA,2002 年中期推出,0.13 μm 工艺,1.5V 内核供电。集成硬件乘法器,芯片内部结构比 Altera 公司以前的产品有很大变化。
- ⊙ Stratix II: Stratix 的下一代产品,2004 年中期推出,90nm 工艺,1.2V 内核供电,大容量高性能 FPGA。
- ⊙ Stratix V: Altera 公司的高端产品,采用 28nm 工艺,提供了 28GB 的收发器件,适合高端的 FPGA 产品开发。

Lattice 公司的 ISP - PLD 器件有 ispLSI1000、ispLSI2000、ispLSI3000、ispLSI6000 系列等,集成了多达 25 000 个 PLD 等效门。近几年,随着集成电路制造技术的飞速发展,这些公司不断地推出集成度更高、性能更好的产品系列。

CPLD 在结构上主要包括可编程逻辑宏单元、可编程输入/输出单元和可编程内部连线三个部分。FPGA 在结构上主要分为三个部分,即可编程逻辑单元、可编程输入/输出单元和可编程连线。

与 ASIC 设计相比,CPLD/FPGA 最明显的优势是大大降低了设计成本,缩短了设计周期,投资风险小,市场适应能力强且硬件升级的余地大。一个产品开发项目,是使用 FPGA 还是 CPLD,主要看开发项目本身的需要。如果是规模普通,并且产量不是很大的产品项目,通常使用 CPLD 比较好;如果是大规模的逻辑设计、ASIC 设计,或者单片系统设计,则多采用 FPGA。另外,FPGA 掉电后将丢失原有的逻辑信息,所以在使用中需要为 FPGA 芯片配置一个专用的 ROM。

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。

高集成度、高速度和高可靠性是 CPLD/FPGA 最显著的特点。芯片的时钟延迟可小至纳秒 (ns) 级。这样的特性结合其并行的工作方式,在超高速应用领域和实时测控方面有着

非常广阔的应用前景。CPLD/FPGA 的高可靠性还表现在几乎可将整个系统下载于同一芯片中，实现片上系统，大大缩小体积，易于设计的屏蔽和管理。

基于 CPLD/FPGA 的高集成度特点，可以利用先进的 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，同时设计过程又几乎与所有元器件的硬件结构无关，因此开发设计成功的各类逻辑功能软件有很好的兼容性和可移植性。几乎可以用于任何型号和规模的 CPLD/FPGA 器件中，从而使得产品设计效率大幅度提高，可以在很短的时间内完成较复杂的系统设计。美国的 IT 公司认为，一个 ASIC 设计 80% 的功能可以用 IP 核 (Core) 等现成逻辑合成。未来大系统的 CPLD/FPGA 设计仅仅是各类再应用逻辑与 IP 核的拼装，设计周期将进一步缩短。

1.1.4 硬件描述语言 (HDL)

硬件描述语言 (HDL) 是相对于一般的计算机软件语言，如 C、Pascal 而言的。HDL 是用于设计硬件电子系统的计算机语言，它描述电子系统的逻辑功能、电路结构和连接方式。HDL 具有与具体硬件电路和设计平台无关的特性，并且具有良好的电路行为描述和系统描述的能力，并在语言易读性和层次化结构化设计方面，表现出强大的生命力和应用潜力。用 HDL 进行电子系统设计的一个很大的优点是设计者可以专心致力于其功能的实现，而不需要对不影响功能的与工艺有关的因素花费过多的时间和精力。

就 CPLD/FPGA 开发来说，比较常用和流行的 HDL 主要有 VHDL、Verilog HDL、ABEL、AHDL、System Verilog 和 System C 等。其中 VHDL、Verilog 在现在 EDA 设计中使用最多，也拥有几乎所有的主流 EDA 工具的支持。而 System Verilog 和 System C 这两种 HDL 语言还处于完善过程中。目前，VHDL 和 Verilog 作为 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，已成为事实上的通用硬件描述语言。有专家认为，在 21 世纪，VHDL 与 Verilog HDL 语言将承担起大部分的数字系统设计任务。VHDL 起源于美国国防部的 VHSIC，Verilog 起源于集成电路的设计。

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。

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者适配器所接受。因此，VHDL 语言的源程序的综合通常要从行为级到 RTL 级再到门电路级的转化，VHDL 几乎不能直接控制门电路的生成，对综合器的性能要求较高；而 Verilog 语言源程序的综合过程要稍简单，即经过 RTL 级到门电路级的转化，易于控制电路的资源。

随着系统级 FPGA 以及系统芯片的出现，软硬件协调设计和系统设计变得越来越重要。传统意义上的硬件设计越来越倾向于与系统设计和软件设计结合。硬件描述语言为适应新的情况，迅速发展，出现了很多新的硬件描述语言，像 Superlog、SystemC、Cynlib C++ 等。

1.1.5 软件开发工具

1) 主流厂家的 EDA 软件开发工具 这类软件一般由 CPLD/FPGA 芯片厂家提供，基本都可以完成所有的设计输入 (原理图或 HDL)、仿真、综合、布线和下载等工作。目前主流



厂家提供的比较常用的一些 EDA 软件工具有 Altera 公司的 Quartus II、Lattice 的 ispLEVER、Xilinx 的 ISE/ISE - Web PACK Series 等。这些软件的基本功能相同，主要的差别在于面向的目标器件不同，同时性能也各有优劣。

Altera 公司提供的系列软件包括以下几种。

- ⊙ MAX + Plus II：普遍认为 Maxplus II 曾经是最优秀的 PLD 开发平台之一，适合开发早期的中小规模 PLD/FPGA 由 Quartus II 替代，不再推荐使用。
- ⊙ Quartus II：Altera 新一代 FPGA/PLD 开发软件，适合新器件和大规模 FPGA 的开发，已经取代 Maxplus II。
- ⊙ SOPC Builder：配合 Quartus II，可以完成集成 CPU 的 FPGA 芯片的开发工作。
- ⊙ DSP Builder：Quartus II 与 Matlab 的接口，利用 IP 核在 Matlab 中快速完成数字信号处理的仿真和最终 FPGA 实现。

Quartus II 是 Altera 公司的综合性 PLD/FPGA 开发软件，原理图、VHDL、VerilogHDL 以及 AHDL (Altera Hardware 支持 Description Language) 等多种设计输入形式，内嵌自有的综合器以及仿真器，可以完成从设计输入到硬件配置的完整 PLD 设计流程。

Quartus II 工具软件属于第四代 PLD 开发平台。该平台支持一个工作组环境下的设计要求，其中包括支持基于 Internet 的协作设计。Quartus II 平台与 Cadence、Exemplar Logic、Mentor Graphics、Synopsys 和 Synplcity 等 EDA 供应商的开发工具相兼容。改进了软件的 Logic Lock 模块设计功能，增添了 Fast Fit 编译选项，推进了网络编辑性能，而且提升了调试能力。该软件完全支持 VHDL、Verilog 的设计流程，内部嵌有 VHDL、Verilog 逻辑综合器。支持第三方的综合工具，如 Leonardo Spectrum、Synplify Pro、FPGA Compiler II 等。这些综合工具具有更好的综合效果，所以通常建议使用这些工具进行 VHDL/Verilog 源程序的综合。Quartus II 可以直接调用这些第三方工具。同样，它还具备仿真功能，同时支持第三方的仿真工具，如 Modelsim。此外，它还为 Altera DSP 开发包进行系统模型设计提供了集成综合环境，与 MATLAB 和 DSP Builder 结合可以进行基于 FPGA 的 DSP 系统开发，是 DSP 硬件系统实现的关键 EDA 工具。Quartus II 还可以与 SOPC Builder 结合，实现 SOPC 系统开发。Quartus II (3.0 和更高版本) 设计软件是业界唯一提供 FPGA 和固定功能 HardCopy 器件统一设计流程的设计工具。工程师使用同样的低价位工具对 Stratix FPGA 进行功能验证和原型设计，又可以设计 Hard Copy Stratix 器件用于批量成品。系统设计者现在能够用 Quartus II 软件评估 HardCopy Stratix 器件的性能和功耗，相应地进行最大吞吐量设计。

Quartus II 5.0 以上版本支持双核 CPU 的嵌入。公司每出一个新版本都会缩短其编译速度。CPU 内核可以用 Altera 提供的 32 位处理器软核 NIOS II 实现。

Quartus II 8.0 版本采用高级布局布线算法，支持多处理器，使编译时间平均缩短 20%；渐进式编译支持，编译时间进一步减少了 70%。利用 Quartus II 软件特性能够迅速地达到时序逼近，满足功耗预算要求。

Quartus II 设计套装的最新版本已发展到 15.1，软件的其他特性包括以下内容。

- ⊙ DSP Builder 12.0 新的数字信号处理 (DSP) 支持：通过系统控制台，与 MATLAB 的 DDR 存储器进行通信，并具有新的浮点功能，提高了设计效能以及 DSP 效率。
- ⊙ 经过改进的视频和图像处理 (VIP) 套装以及视频接口 IP：通过具有边缘自适应算法的 Scaler II MegaCore 功能以及新的 Avalon - Streaming (Avalon - ST) 视频监控和跟踪