



工业和信息化部普通高等教育“十二五”规划教材立项项目



21 世纪高等院校电气工程与自动化规划教材

21 century institutions of higher learning materials of Electrical Engineering and Automation Planning

**E** DA Technology

# EDA 技术 实用教程

朱娜 张金保 王志强 李建利 编著



人民邮电出版社  
POSTS & TELECOM PRESS



工业和信息化普通高等教育“十二五”规划教材立项项目



21 世纪高等院校电气工程与自动化规划教材

21 century institutions of higher learning materials of Electrical Engineering and Automation Planning

**E** DA Technology

# EDA 技术 实用教程

朱娜 张金保 王志强 李建利 编著

人民邮电出版社

北京

## 图书在版编目 (C I P) 数据

EDA技术实用教程 / 朱娜等编著. — 北京 : 人民邮电出版社, 2012.7  
21世纪高等院校电气工程与自动化规划教材  
ISBN 978-7-115-27969-9

I. ①E… II. ①朱… 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087468号

## 内 容 提 要

随着半导体技术和计算机技术的飞速发展, 集成电路的设计方法发生了深刻的变化。硬件描述语言应运而生, 它的出现标志电路设计领域的一次重大的变革。

本书侧重实战, 内容丰富, 经典新颖实例兼而有之。全书共 8 章。第 1 章是 EDA 技术的概述和 FPGA/CPLD 相关产品的概述; 第 2 章分别介绍 ISE、Quartus II 和 Actel Libero IDE 三种开发环境下 FPGA 工程的创建和调试方法; 第 3 章主要介绍 VHDL 的基础知识, 目的是使初学者对 VHDL 产生系统的认识, 有一定开发经验的读者可以跳过这部分; 第 4 章主要介绍常用的 VHDL 程序结构, 目的是使读者掌握扎实的基本功; 第 5 章主要介绍 VHDL 语言规范化标准和常见的 Warning 分析; 第 6 章介绍 Nios II 软核 SOPC 系统, 以具体实例讲解 Quartus II 开发环境下软核的创建和配置方法; 第 7 章介绍 Quartus II 开发环境下各种 IP 核的配置方法, 合理使用 IP 核可以简化用户的工程设计; 第 8 章主要介绍 Quartus II 9.0 软件功能及常用辅助设计工具, 这些工具可以使用户更高效、更快速地完成其工程设计。书中列举的大量实例都经过精心设计, 包含了自顶向下的设计思想, 模块化和层次化的设计方式, 全部实例都经过软件仿真验证或硬件实际测试。

本书的特点是注重实用、讲述清楚、由浅入深, 书中的实例具有很高的参考价值和实用价值, 能够使读者掌握较多的实战技能和经验。它既可作为高等院校电气、自动化、计算机、通信、电子类专业的研究生、本科生的教材或参考书, 也可供广大 ASIC 设计人员和电子电路设计人员阅读参考。

21 世纪高等院校电气工程与自动化规划教材

### EDA 技术实用教程

- 
- ◆ 编 著 朱 娜 张金保 王志强 李建利  
责任编辑 刘 博
  - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号  
邮编 100061 电子邮件 315@ptpress.com.cn  
网址 <http://www.ptpress.com.cn>  
大厂聚鑫印刷有限责任公司印刷
  - ◆ 开本: 787×1092 1/16  
印张: 20.25 2012 年 7 月第 1 版  
字数: 509 千字 2012 年 7 月河北第 1 次印刷

---

ISBN 978-7-115-27969-9

定价: 39.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223  
反盗版热线: (010)67171154

EDA 技术是在电子 CAD 技术基础上发展起来的计算机软件系统，以计算机为工作平台，用硬件描述语言 HDL 完成设计文件，融合应用电子技术、计算机技术、信息处理及智能化技术的最新成果，进行电子产品自动设计。利用 EDA 工具，可以从概念、算法、协议等开始设计电子系统，可将电子产品从电路设计、性能分析到设计出 IC 版图或 PCB 版图的整个过程在计算机上自动处理完成。EDA 技术设计的标准化语言，日益强大的逻辑设计仿真测试技术已使其成为现代电子设计技术的核心。EDA 软件工具的一个重要分支 PLD，是一种由用户根据需要而自行构造逻辑功能的数字集成电路。目前主要有 CPLD 和 FPGA 两大类型，最有代表性的 PLD 厂家为 Altera、Xilinx 和 Lattice 公司。VHDL 语言和硬件描述语言 Verilog HDL，是 ASIC 设计和 PLD 设计的主要输入工具，在 ASIC 设计方面平分秋色。

EDA 技术已经渗透到各行各业，在机械、电子、通信、航空航天、化工、矿产、生物、医学、军事等各个领域都有应用。几乎所有理工科（特别是电子信息）类的高校都开设了 EDA 课程。主要是让学生了解 EDA 的基本概念和基本原理，掌握用 HDL 语言编写规范，掌握逻辑综合的理论和算法，使用 EDA 工具进行电子电路课程的实验验证并从事简单系统的设计，为今后工作打下基础。科研方面主要利用电路仿真工具（multiSIM 或 PSPICE）进行电路设计与仿真，利用虚拟仪器进行产品测试，将 CPLD/FPGA 器件实际应用到仪器设备中，从事 PCB 设计和 ASIC 设计等。产品设计与制造方面，包括计算机仿真，产品开发中的 EDA 工具应用，系统级模拟及测试环境的仿真，生产流水线的 EDA 技术应用，产品测试等各个环节。ASIC 和 PLD 设计正向超高速、高密度、低功耗、低电压方面发展。EDA 技术发展迅猛，应用广泛，已涉及各行各业。

当前工科院校相关专业开设 EDA 课程，多注重实用设计，面向高校及工程技术人员，但对 VHDL 语言规范讲解不是很细。在实际应用中，对 EDA 技术的全面掌握，系统高可靠性及可移植性，要求设计人员不仅熟悉设计流程，语言规范可靠，且对不同系统设计能举一反三。这显然说明教学内容与当前实际应用需求之间存在差距，不利于实用型人才培养。本书正是为解决上述问题，同时也为相关教学改革提出新的思路而编写的。

本书主要以 Altera 公司 Cyclone 系列 FPGA 芯片为硬件平台，VHDL 语言为输入工具，基于学校培养模式，兼顾市场需求，着重语言基础及创新性培养的同时，结合具体应用对象，采用规范化语言，清晰易懂的设计流程，达到举一反三的教学效果。本书在相关章节



介绍了输入语言规范化标准和常见警告消除方法，提供诸多高质量典型例程以及练习，使读者掌握相应知识点。读者能够通过完整的实例，快速、有效地掌握规范化 VHDL 语言及 FPGA 开发流程；通过同步实验掌握各章节的重点难点，真正对相关知识做到融会贯通。本书配有电子教案，便于教师教学和相关课程的补充与完善。

本书首先介绍 ISE、Quartus II 和 Actel Libero IDE 三种开发环境下 FPGA 工程的创建和调试方法；然后详细介绍 VHDL 的语言基础和 VHDL 语言规范化标准；最后以 Quartus II 开发环境为基础，重点介绍 NIOS 软核系统设计、IP 核设计以及 Quartus II 工程设计的辅助设计工具。

本书可作为高等学校电子工程、通信、工业自动化、计算机应用技术、电子对抗、仪器仪表、数字信号或图像处理等学科专业与相关的实验指导课的授课教材或主要参考书，也可供从事数字电路硬件设计的工程师阅读和参考，还可作为 EDA 技术短训班的教材。

本书由朱娜、张金保和王志强编著，其中，第 1 章到第 3 章由朱娜编写，第 5 章、第 7 章和第 8 章由张金保编写，第 4 章和第 6 章由王志强编写，全书由朱娜统稿。参与本书编写工作的还有姜雪松、王素贞、渠丰沛、刘艳梅、姜雪峰、渠莉娜、杜平、尹斯星等。本书在编写过程中参考了很多专家和学者的著作以及 Xilinx 公司的数据手册，在此表示深深的谢意！

由于 EDA 设计技术和可编程逻辑器件发展迅速，各种相关新技术不断涌现，同时限于编者的实际开发经验和理论水平，书中难免存在一些不足之处或者错误，恳请相关专家和读者批评指正。

编者  
2012 年 5 月

# 目 录

第1章 绪论.....1	2.4 Actel Libero IDE v9.1 软件概述.....63
1.1 EDA 技术概论.....1	2.4.1 软件特点及支持的器件.....63
1.1.1 EDA 技术及发展.....2	2.4.2 软件用户界面及功能简介.....65
1.1.2 EDA 技术应用.....4	2.4.3 Actel Libero IDE v9.1 软件 设计实例.....68
1.2 PLD 及 FPGA/CPLD 产品概述.....5	小结.....72
1.2.1 PLD 技术及发展.....6	习题.....73
1.2.2 PLD 技术分类.....6	第3章 VHDL 基础.....74
1.2.3 PLD 产品可编程原理.....7	3.1 VHDL 的基本元素.....74
1.2.4 CPLD 结构与可编程原理.....21	3.1.1 标识符.....74
1.2.5 FPGA 结构与可编程原理.....29	3.1.2 数据对象.....76
1.2.6 FPGA/CPLD 器件配置器件.....32	3.1.3 数据类型.....80
1.3 EDA 设计基础(数字系统).....33	3.1.4 运算操作符.....85
小结.....36	3.2 VHDL 的语句基础.....90
习题.....36	3.2.1 进程语句.....90
第2章 FPGA/CPLD 系统设计.....37	3.2.2 信号赋值语句.....94
2.1 FPGA/CPLD 设计流程及 常用开发工具.....37	3.2.3 顺序描述语句.....98
2.1.1 FPGA/CPLD 设计流程.....37	3.2.4 并行描述语句.....115
2.1.2 FPGA/CPLD 常用开发工具.....38	3.3 VHDL 的属性描述.....124
2.2 Quartus II 9.0 软件概述.....38	3.3.1 值类属性.....125
2.2.1 Quartus II 9.0 软件特点及 支持的器件.....39	3.3.2 函数类属性.....128
2.2.2 Quartus II 9.0 软件用户界面 及功能简介.....40	3.3.3 信号类属性.....131
2.2.3 Quartus II 9.0 软件设计 流程.....46	3.3.4 数据类型类属性.....133
2.2.4 Quartus II 9.0 软件设计 实例.....47	3.3.5 数据范围类属性.....134
2.3 ISE10.1 软件概述.....54	小结.....135
2.3.1 ISE10.1 软件特点及支持的 器件.....55	习题.....135
2.3.2 ISE10.1 软件用户界面及 功能简介.....55	第4章 VHDL 程序结构.....136
2.3.3 ISE10.1 软件设计实例.....60	4.1 VHDL 基本建模结构.....136
	4.1.1 行为描述.....136
	4.1.2 数据流描述.....137
	4.1.3 结构描述.....137
	4.2 VHDL 实体.....139
	4.2.1 实体语句.....139
	4.2.2 参数传递说明语句.....139

4.2.3	参数传递映射语句	141	5.2.12	I/O	190
4.2.4	端口说明语句	141	5.2.13	异步设计	191
4.2.5	实体说明部分	142	5.3	优化设计	192
4.3	VHDL 结构体	142	5.3.1	标准单元实现	193
4.3.1	结构体名	143	5.3.2	复杂逻辑运算单元共享	193
4.3.2	结构体说明部分	143	5.3.3	中间信号	193
4.3.3	并行处理语句	143	5.3.4	针对目标 FPGA 的优化	194
4.4	子程序	144	5.3.5	综合工具设置优化	194
4.4.1	过程	144	5.4	常见 Warning 分析	195
4.4.2	函数	149	小结		196
4.5	VHDL 库和程序包	154	习题		196
4.5.1	库	154	<b>第 6 章 NIOS 软核 SOPC 系统设计</b>		197
4.5.2	程序包	155	6.1	Nios II 软核 SOPC 系统	197
4.6	元件和配置	156	6.1.1	SOPC 技术简介	197
4.6.1	默认连接和默认配置	156	6.1.2	Nios II 软核 SOPC 系统 结构及开发环境	198
4.6.2	元件配置	159	6.2	Nios II 指令系统	199
4.6.3	块的配置	161	6.2.1	寄存器结构	199
4.6.4	结构体配置	164	6.2.2	寻址方式	201
小结		164	6.2.3	Nios II 指令集	201
习题		165	6.2.4	Nios II 定制指令	206
<b>第 5 章 VHDL 语言规范化标准及 常见 Warning 分析</b>		166	6.3	SOPC 硬件系统开发	211
5.1	编码设计	166	6.3.1	硬件开发流程及环境介绍	212
5.1.1	结构化设计	166	6.3.2	创建 Quartus II 工程	212
5.1.2	代码描述	167	6.3.3	生成 Nios II 系统	214
5.1.3	内部命名规则	167	6.3.4	集成 Nios II 系统到 Quartus II 工程	218
5.1.4	其他	168	6.3.5	Quartus II 工程编译及下载	219
5.2	可综合代码描述规则	168	6.4	SOPC 软件系统开发	219
5.2.1	概述	168	6.4.1	软件开发流程及环境介绍	219
5.2.2	复位	169	6.4.2	HAL 系统库	220
5.2.3	时钟	172	6.4.3	Nios II IDE 用户应用 程序建立	220
5.2.4	接口异步信号处理	174	6.5	Nios II 外围设备	223
5.2.5	寄存器和锁存器	175	6.5.1	SDRAM 控制器	223
5.2.6	有限状态机	177	6.5.2	CFI 控制器	225
5.2.7	使用 Case 语句的多路复用	181	6.5.3	EPCS 设备控制器	226
5.2.8	编码器	181	6.5.4	PIO 控制器	227
5.2.9	译码器	185	6.5.5	定时器控制器	227
5.2.10	计数器	188			
5.2.11	运算	188			

6.5.6 DMA 控制器 .....	229	习题 .....	255
6.5.7 SPI 核 .....	230	<b>第 8 章 Quartus II 9.0 软件功能及常用</b>	
6.5.8 UART 核 .....	230	<b>辅助设计工具</b> .....	256
6.5.9 JTAGUART 核 .....	231	8.1 I/O 分配验证 .....	256
6.5.10 系统 ID 核 .....	232	8.1.1 I/O 分配验证简介 .....	257
6.5.11 PLL 核 .....	232	8.1.2 I/O 分配验证运行 .....	260
6.5.12 mutex 核 .....	233	8.2 功率分析 .....	262
<b>6.6 SOPC 系统深入设计——用户</b>		8.2.1 Excel-based 功率计算器 .....	263
定制外设 .....	234	8.2.2 Simulation-based 功率估算 .....	266
6.6.1 元件开发流程 .....	234	8.3 原理图观察工具 .....	266
6.6.2 硬件设计 .....	235	8.3.1 RTL 阅读器 .....	267
6.6.3 软件设计 .....	236	8.3.2 技术映射查看器 .....	273
6.6.4 验证设计 .....	236	8.4 SignalProbe 及 SignalTap II	
6.6.5 共享元件 .....	237	逻辑分析器 .....	274
小结 .....	237	8.4.1 SignalProbe .....	274
习题 .....	237	8.4.2 SignalTap II 逻辑分析器 .....	277
<b>第 7 章 Altera 的 IP 核工具</b> .....	238	8.5 时序收敛平面布局规划器和	
7.1 基本概念 .....	238	器件规划图 .....	283
7.1.1 IP 核的概念 .....	238	8.5.1 概述 .....	284
7.1.2 Altera 可提供的 IP 核 .....	239	8.5.2 设计流程 .....	285
7.2 Altera 基本宏功能设计实现 .....	241	8.6 Quartus II 9.0 支持的第三方	
7.2.1 基本宏功能定制 .....	241	工具 .....	293
7.2.2 基本宏功能实现 .....	244	8.6.1 NativeLink 与 WYSIWYG .....	293
7.2.3 设计实例 .....	247	8.6.2 Modelsim 仿真工具 .....	295
7.3 Altera IP 核设计实现 .....	250	8.6.3 Synplify/Sinplify Pro 综合	
7.3.1 IP 核定制 .....	251	工具 .....	310
7.3.2 IP 核实现 .....	253	小结 .....	315
7.3.3 设计实例 .....	254	习题 .....	315
小结 .....	255	<b>参考文献</b> .....	316



EDA (Electronic Design Automation) 技术是现代电子产品设计技术中一门综合性的新技术, 代表了电子设计技术和应用技术的发展方向。本书主要介绍面向 FPGA/CPLD 的 EDA 技术应用。本章主要对 EDA 技术作概括性介绍, 主要内容如下:

- (1) EDA 技术的概念及发展过程。
- (2) EDA 技术的应用。
- (3) FPGA/CPLD 产品的概述。
- (4) EDA 的设计基础。

## 1.1 EDA 技术概论

【问题引出】什么叫 EDA 技术?

EDA 技术是指以大规模可编程逻辑器件为设计载体, 由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。

为了满足千差万别的系统用户提出的设计要求, 最好的办法是由用户自己设计芯片, 让他们把想设计的电路直接设计在自己的专用芯片上。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工作由设计工具来完成, 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、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。

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)。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、Verilog 或 AHDL），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。未来的 EDA 技术将向广度和深度两个方向发展，EDA 将会超越电子设计的范畴进入其他领域。随着基于 EDA 的 SOC（单片系统）设计技术的发展，软、硬核功能库的建立，以及基于 VHDL 所谓自顶向下设计理念的确立，未来的电子系统的设计与规划将不再是电子工程师们的专利。有专家认为，21 世纪将是 EDA 技术快速发展的时期，并且 EDA 技术将是对 21 世纪产生重大影响的十大技术之一。

### 1.1.1 EDA 技术及发展

**【展开讲解】EDA 技术的发展历史及发展趋势。**

自从进入 20 世纪 90 年代以后，EDA 技术的发展和普及给电子系统设计带来了革命性的变化，传统的“固定功能模块+连线”的设计方法正在逐步退出历史舞台，而基于芯片的设计方法正在成为电子系统设计的主流。在基于芯片的设计方法中，EDA 技术扮演了越来越重要的角色，逐渐成为设计人员不可或缺的重要设计手段。如今无论是逻辑芯片设计还是电子系统设计，所有的设计工作都需要在计算机上借助相应的 EDA 工具进行，否则将会很难完成相应的设计任务。

目前，可编程逻辑器件和 EDA 技术的结合为电子系统的设计带来了极大的方便，它们已经逐渐成为设计人员进行硬件设计的最强有力工具。现在设计人员只要拥有一台计算机、一套相应的 EDA 开发工具和可编程逻辑器件，基本上就可以实现电子系统的设计操作。可以毫不夸张地说，任何的电子系统设计都离不开可编程逻辑器件和 EDA 开发工具。现在，很多电子系统设计相关领域的各大公司和科研单位都采用可编程逻辑器件和 EDA 工具的结合来进行相应的设计，这样可以大大缩短系统的设计周期，以适应当今品种多、批量小的电子市场的需求，提高产品的竞争能力。

下面将对 EDA 设计技术的相关知识进行简单介绍。

从 EDA 设计技术的发展历史来看，它大致经历了 3 个重要阶段，即第一代 EDA 工具的产生和发展、第二代 EDA 工具的产生和发展以及第三代 EDA 工具的产生和发展。每一代 EDA 工具都具有其优缺点，后一代 EDA 工具都是对前一代 EDA 工具的不断改进和功能扩展，从而不断满足各个历史时期设计人员的开发需要。

#### (1) 第一代 EDA 工具

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经无法满足设计精度和效率的要求，因此工程师们就开始进行二维平面图形的计算机辅助设计，以摆脱复杂机械的版图设计工作。

20 世纪 70 年代，第一代 EDA 工具产生了，它的典型代表是风靡一时的 TANGO 软件。在当时的情况下，EDA 工具供应商只有很少的几家，开发技术十分不成熟，EDA 工具的功能和自动化程度较低，而且产品也几乎全部面向 LSI 或 PCB 的设计，应用领域比较单一。一般来说，第一代 EDA 工具也称为 CAD（Computer Aided Design，计算机辅助设计）。

## (2) 第二代 EDA 工具

随着科学技术的不断进步和发展,许多公司如 Mentor 公司、Daisy Systems 公司、Logic Systems 公司进入了 EDA 工具的市场,它们开始为设计开发人员提供电路图逻辑工具和逻辑模拟工具的 EDA 软件,这个时期的 EDA 工具以数字电路分析工具为代表,主要功能是用来解决电路设计没有完成之前的功能检验问题。

20 世纪 80 年代,第二代 EDA 工具产生了,它主要以计算机仿真和自动布局布线技术为核心,同时产生了 CAM (Computer Aided Manufacturing, 计算机辅助制造)、CAT (Computer Aided Test, 计算机辅助测试)、CAE (Computer Aided Engineering, 计算机辅助工程) 等新概念。第二代 EDA 工具的应用软件主要有数字电路分析、模拟电路分析、印制电路板、现场可编程门阵列的布局布线等,它们以软件工具为核心,即针对产品开发分为设计分析、生产测试等多个独立的软件包,每个软件只能完成其中的一项工作,通过顺序循环完成设计的全过程。

第二代 EDA 工具的最大缺点就是不能进行系统级的仿真和综合,如果在产品发展的后期才发现设计错误,进行修改无疑是十分困难且浪费大量的人力。另外,由于软件开发商的不统一,一个工具的输出作为另外几个工具的输入需要进行界面处理,影响了 EDA 工具的设计速度。

## (3) 第三代 EDA 工具

20 世纪 90 年代后期,各大 EDA 厂商开始推出以高级语言描述、系统级仿真和综合技术为核心技术的第三代 EDA 工具,目前设计人员使用的 EDA 工具都属于第三代 EDA 工具的范畴之内。第三代 EDA 工具以逻辑综合、硬件行为仿真、参数分析和测试为重点,提供了门类齐全和满足系统设计需要的全部开发工具。例如,描述设计意图的设计输入工具、具有逻辑综合和设计优化能力的设计工具以及验证设计和评估性能的仿真工具。

第三代 EDA 工具的主要特征是自动化程度大大提高,设计人员只需要在前期进行较少的设置便可以由计算机完成整个设计,人工干预大大减少,设计效率大大提高。第三代 EDA 系统主要以并行设计工程的方式和系统级目标设计方法为支持。系统设计的核心是可编程逻辑器件的设计。由于可编程逻辑器件自身的可重复编写的特性,使电子设计的灵活性和工作效率大大提高。

随着科学技术的不断发展,EDA 设计技术的发展趋势主要体现在以下几个方面。

### (1) EDA 工具的 PC 平台化

目前,可编程逻辑器件和 EDA 技术的结合为电子系统的设计带来了极大的方便,它们已经成为设计人员进行系统设计的主要工具。在过去相当长的一段时间内,EDA 工具软件价格十分昂贵,它的操作平台是工作站和 UNIX 操作系统,硬件环境要求高,因此大大阻碍了 EDA 工具的迅速普及。

最近十年内,经过 EDA 工具厂商和科技研发人员的共同努力和推广,EDA 工具的 PC 平台化进展速度十分显著,如 Xilinx 公司的 Foundation 和 ISE、Altera 公司的 MAX+plus II 和 Quartus II 都是可以在 Windows 或者 Windows NT 操作系统中运行的 EDA 工具。这些基于 PC 平台的 EDA 工具包含有逻辑设计、仿真、综合、优化等工具,而且价格比较便宜,目前已经得到了十分广泛的应用。

可以看出,随着 PC 性能的不不断提高,基于 PC 平台的 EDA 工具的软件功能将会更加完善和齐全。因此,EDA 工具的 PC 平台化是 EDA 工具迅速普及的重要前提,同样它也是 EDA

设计技术发展的必然趋势。

### (2) EDA 设计技术朝着 ESDA 和 CE 方向发展

对于现有的各种 EDA 工具来说,各种 EDA 工具通常是用来进行某一方面的系统设计,如 Protel 工具主要是用来进行印制电路板的设计操作。随着科学技术的发展以及缩短电子系统设计周期的要求,设计人员往往希望各种不同功能的 EDA 工具能够在统一的数据库或者管理框架下进行工作,因此提出了 ESDA 和 CE 的概念。

ESDA (Electronic System Design Automation) 即电子系统设计自动化,它强调建立从系统到电路的统一描述语言,同时考虑仿真、综合与测试,把定时、驱动能力、电磁兼容特性、机械特性、热特性等约束条件加入到设计综合中,然后进行统一的设计描述和优化操作,从而提高系统设计的一次成功率。ESDA 要求系统级设计人员改变优先考虑具体实现的传统思路,而是集中精力进行系统的总体设计、综合方案比较和优化设计。可见,这将会是一种全新的设计思路。

CE (Concurrent Engineering) 即并行设计工程,它要求 EDA 工具从管理层次上把与系统设计有关的工具、任务、时间、工艺等进行合理安排,设计人员使用统一的集成化设计环境,各个设计小组能够共享与设计相关的数据库和其他资源,这样可以同步地进行系统的设计工作。可以看出,CE 改变了在系统设计中过分依赖专业分工和设计人员专业知识的传统设计方法。

### (3) EDA 工具应该具有编译选择能力

对于 EDA 工具开发厂商来说,除了简单加快软件的编译速度外,EDA 设计工具还应该能够减少编译时间,而不需要考虑其编译处理的能力如何。在一个具体的设计过程中,最耗费时间的应该是布局和布线过程,如果能够减少布局和布线过程的时间,那么将会大大提高系统设计的效率。随着技术的不断发展,设计人员希望能够实现一种编译选择的方案,即 EDA 工具应该具有只对上次编译后发生变化的那部分设计进行布局和布线操作的能力,即增量编译能力。可以看出,具有增量编译能力的 EDA 工具将是未来 EDA 设计技术的一个发展方向,它将会大大提高设计效率,从而缩短产品开发周期,进而提高产品的市场竞争力。

## 1.1.2 EDA 技术应用

### 【展开讲解】EDA 技术的主要组成方面。

EDA 技术涉及面广,内容丰富,主要应从如下 3 个方面掌握 EDA 技术:大规模可编程逻辑器件、硬件描述语言和软件开发工具。其中,大规模可编程逻辑器件是 EDA 技术的主要目标器件,硬件描述语言是 EDA 技术的主要描述和表达方式,软件开发工具是 EDA 技术的开发平台。下面分别对 3 部分进行概要的介绍。

#### 1. 大规模可编程逻辑器件

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。主要有现场可编程门阵列 (Field Programmable Gate Array, FPGA) 和复杂可编程逻辑器件 (Complex Programmable Logic Device, CPLD)。FPGA 和 CPLD 器件的应用十分广泛,它们随着 EDA 技术的发展而成为了电子设计领域的重要角色。

FPGA/CPLD 最明显的特点是集成度高、高速度和高可靠性,其时钟延时可小至纳秒级。其并行工作方式使得在超高速应用领域和实时测控方面有着很广阔的应用前景。如果用户设计合理,将不会存在类似于 MCU 的复位不可靠和 PC 可能跑飞等问题,而且几乎可将整个系

统下载于同一芯片中，真正实现片上系统，大大缩小了体积，易于管理和屏蔽，具有很高的可靠性。

FPGA/CPLD 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，设计开发成功的各类逻辑功能模块有很好的兼容性和可移植性，几乎可用于任何型号和规模的 FPGA/CPLD 中，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。

国际上生产 FPGA/CPLD 的主流公司主要有 Xilinx、Altera、Actel 和 Lattice 公司。

(1) Xilinx 公司的 FPGA 器件有：Spartan FPGA 系列、Virtex FPGA 系列、CoolRunner、XC9500XL 等。

(2) Altera 公司的 FPGA 器件有：Classic、MAX、FLEX、APEX、ACEX、APEX II、Cyclone1/2/3/4、MAX II、Stratix1/2/3/4/6 系列等。

(3) Actel 是反熔丝 PLD 技术的领导者，它们公司的器件主要有：

低功耗 Flash 型：LGOO 系列、ProASIC3 系列；

混合信号：Fusion 系列；

耐辐射器件：RTAX-S 系列、RTSX-SU 系列；

反熔丝器件：Axcelerator、SX-A、eX、MX 系列。

(4) Lattice 公司的 ISP—PLD 器件有：ispLS1 系列、MachXO 系列、MACH4000 系列、LatticeSC/M FPGA 系列、LatticeECP3 FPGA 系列。

## 2. 硬件描述语言

硬件描述语言 (HDL) 是 EDA 技术的主要组成部分，是 EDA 各种描述方法中最能体现优越性的一种描述方法。所谓硬件描述语言，就是用于描述设计系统的逻辑功能，实现该功能的算法、电路结构、约束条件等。常用的硬件描述语言有 VHDL、Verilog、System Verilog 和 System C，其中 System Verilog 和 System C 还在不断完善。VHDL 作为 IEEE 的工业标准硬件描述语言，在电子工程领域已成为通用的硬件描述语言。Verilog 支持的 EDA 工具较多，适用于 RTL 级和门电路级的描述，其综合过程较 VHDL 稍简单，但其在高级描述方面不如 VHDL。

## 3. 软件开发工具

目前，在国内比较流行的 EDA 的软件工具有 Altera 公司的 MAX+plus II 和 Quartus II、Xilinx 公司的 ISE、Actel 公司的 Libero。这 3 个软件的基本功能相同，主要差别在于面向的目标器件不一样。它们界面友好，使用便捷，均可以语言文本、波形、EDIF 等格式的文件作为设计输入，并支持这些文件的任意混合设计；具有门级仿真器，可以进行功能仿真和时序仿真，能够产生精确的仿真结果；在适配之后生成供时序仿真用的 EDIF、VHDL 和 Verilog 这 3 种不同格式的网表文件。

本书主要是以 Altera 公司的 Quartus II 9.0 为开发平台，以可编程逻辑器件为目标器件，以 VHDL 为硬件描述语言的 EDA 技术学习。

## 1.2 PLD 及 FPGA/CPLD 产品概述

【问题引出】PLD 技术是怎样发展的？主要有哪几类？其各自的结构和工作原理是什么？

可编程逻辑器件 (Programmable Logic Devices, PLD) 是 20 世纪 70 年代发展起来的一

种新的集成器件，PLD 是大规模集成电路技术发展的产物，是一种半定制的集成电路，结合 EDA 技术可以快速、方便地构建数字系统。

本节主要介绍 PLD 技术和几类常用的大规模可编程逻辑器件的结构和工作原理，并对 FPGA/CPLD 器件的配置器件作进一步介绍。

在 20 世纪 80 年代以后，PLD 随着集成电路技术和计算机技术的发展而迅速发展。自问世以来，PLD 经历了从 PROM、PLA、PAL、GAL 到 FPGA 等高密度 PLD 的发展过程。在此期间，PLD 的集成度、速度不断提高，功能不断增强，结构趋于更合理，使用变得更加灵活方便。PLD 的出现，打破了由中小规模通用型集成电路和大规模专用集成电路垄断的局面。与中小规模通用型集成电路相比，用 PLD 实现的数字系统，有集成度高、速度快、功耗小、可靠性高等优点。与大规模专用集成电路相比，用 PLD 实现的数字系统，有研制周期短、前期投资少、无风险、修改逻辑设计方便、小批量生产成本低等优势。可以预见，在不久的将来，PLD 将在集成电路市场占统治地位。

### 1.2.1 PLD 技术及发展

**【展开讲解】** PLD 技术的发展历程。

最早的可编程逻辑器件出现在 20 世纪 70 年代初，主要是可编程只读存储器（PROM）和可编程逻辑阵列（PLA）。20 世纪 70 年代末，AMD 公司对 PLA 进行了改进，推出了可编程阵列逻辑（Programmable Array Logic, PAL）器件。20 世纪 80 年代初期，美国 Lattice 公司推出了一种电可擦写的比 PAL 使用更灵活的器件，称为通用阵列逻辑（Generic Array Logic, GAL）。随着技术的进步，生产工艺的不断改进，器件规模不断扩大，逻辑功能不断增强，各种可编程逻辑器件逐渐产生，如 PROM、EPROM、EEPROM 等。现在一般把超过某一集成度的 PLD 器件都称为 CPLD。在 20 世纪 80 年代中期，美国 Xilinx 公司首先提出现场可编程概念并生产出了世界上第一片现场可编程门阵列（FPGA）器件。FPGA 器件采用逻辑单元阵列结构和静态随机存取存储器工艺，设计灵活，集成度高，可无限次反复编程。进入 20 世纪 90 年代，可编程逻辑集成电路技术进入飞速发展时期，器件的可用逻辑门数超过了百万门，并出现了内嵌复杂功能模块（如 RAM、CPU 核、DSP 核等）的片上系统（System On a Programmable Chip, SOPC）。

### 1.2.2 PLD 技术分类

**【展开讲解】** PLD 技术的分类方法。

可编程逻辑器件的种类很多，几乎每个大的可编程逻辑器件供应商都能提供具有自身特点的 PLD 器件。由于历史原因，PLD 的分类方法较多也不统一，下面简单介绍 4 种。

#### 1. 从结构的集成度分类

从结构的集成度上一般可将 PLD 分为简单 PLD 和复杂 PLD，或分为低密度 PLD 和高密度 PLD (HDPLD)。通常，当 PLD 中的等效门数超过 500 门时认为是高密度 PLD，反之是低密度 PLD。传统的 PAL 和 GAL 是典型的低密度 PLD，其余如 EPLD、FPGA 等则称为 HDPLD 或 CPLD。

#### 2. 从结构上分类

从结构上可将 PLD 分为乘积项结构器件和查找表结构器件。

(1) 乘积项结构器件：其基本结构为“与—或阵列”的器件，大部分简单的 PLD 和 CPLD



都属于此类器件。

(2) 查找表结构器件：由简单的查找表组成可编辑门，再构成阵列形式。大多数 FPGA 属于此类器件。

### 3. 从可编程特性上分类

从可编程特性上可将 PLD 分为一次可编程和重复可编程两类。一次可编程的典型产品是 PROM、PAL 和熔丝型 FPGA，其他大多是重复可编程的。其中，用紫外线擦除的产品的编程次数一般在几十次，采用电擦除方式的产品的编程次数稍多些，采用 E<sup>2</sup>CMOS 工艺的产品，擦写次数可达上千次，而采用 SRAM 结构，则可实现无限次编程。

### 4. 从可编程器件的编程元件上分类

最早的 PLD 器件（如 PAL）大多采用的是 TTL 工艺，但后来的 PLD 器件（如 GAL、EPLD、FPGA）都采用 MOS 工艺（如 NMOS、CMOS、E<sup>2</sup>CIOS 等）。目前，一般有下列 6 种编程元件。

(1) 熔丝型 (Fuse)：编程过程就是根据设计的熔丝图文件来烧断对应的熔丝进行编程，早期的 PROM 采用此种编程方式。

(2) 反熔丝型 (Anti-Fuse)：在编程处通过击穿漏层使得两点之间获得导通，Actel 公司的某些 FPGA 器件采用此种编程方式。

(3) EPROM 型：需要有石英窗口，紫外线擦除，可多次编程。

(4) EEPROM 型：电可擦写编程器件，现有的大部分 CPLD 和 GAL 器件采用此类结构。

(5) SRAM 型：查找表结构器件，大部分 FPGA 器件都是采用此种编程工艺。这种编程方式在速度和编程要求上要优于前 4 种，但 SRAM 型器件的编程信息放在 RAM 中，断电丢失，重新上电后需要专用器件对它进行再次配置。

(6) Flash 型：可以实现多次可编程，同时做到掉电后不需要重新配置。

## 1.2.3 PLD 产品可编程原理

**【展开讲解】** PLD 产品的种类及其各类的可编程原理。

简单 PLD 是早期出现的可编程逻辑器件，它们的逻辑规模都比较小，结构上是由简单的“与—或”门阵列和输入/输出单元组成，常见的 PLD 有 PROM、PLA、PAL 和 GAL，下面分别进行介绍。

### 1. 可编程只读存储器 (PROM)

半导体只读存储器是一种能够存储大量二进制信息的半导体器件，它是电子计算机和其他许多电子设备中常用的一种器件。通常，半导体存储器的种类很多，一般可以分为只读存储器 (Read Only Memory, ROM) 和随机存储器 (Random Access Memory, RAM) 两大类。这里，只读存储器在正常情况下只能从中读取数据，但是不能快速地随时修改或重新写入数据；随机存储器在正常工作状态下可以随时向存储单元里写入数据或读出数据。

从具体的逻辑结构上来看，只读存储器是一种可编程逻辑器件，它的电路结构框图如图 1-1 所示。可以看出，只读存储器包括地址译码器、存储矩阵和输出缓冲器 3 个部分。可见，只读存储器的逻辑电路比较简单，利用它来实现组合逻辑函数，尤其是多输出函数是十分方便的。

在只读存储器的结构框图中，地址译码器的作用是将输入的地址代码译成相应的控制信号，从而利用这个控制信号从存储矩阵中把指定的单元选出，然后把其中的数据送到输出缓

冲器上。如果地址译码器具有  $n$  条地址输入线，那么译码器将会有  $2^n$  条输出线，这样的译码输出线通常也称之为字线，它连接到存储矩阵的相应存储单元。

存储矩阵是由许多存储单元构成的，而每个存储单元又是由若干位基本存储单元构成的，它们是只读存储器的核心部分。通常，基本存储单元既可以采用二极管构成，也可以采用双极型三极管和 CMOS 管构成，其中每个基本存储单元可以存储一位二进制信息。每一个存储单元可以存放若干位二进制信息，这样的若干位二进制信息称为一个字，其位数称为字长。由于每一个存储单元或者一组存储单元对应于一个地址代码，因此可以推算，一个具有  $n$  条地址输入线、 $m$  位字长的只读存储器的存储容量为  $2^n \times m$  位。

输出缓冲器是由三态门构成的，它的作用是用来驱动只读存储器的具体数据输出，这里输出缓冲器的输出线称之为位线。每当只读存储器输入一组地址信号，如果相应的译码器输出有效，这时便选中一条字线同时把这个存储单元存放的内容读出到它的输出线上。可以看出，输出缓冲器的作用体现在两个方面：一是用来提高只读存储器的带载能力；二是实现对输出数据的三态控制，目的是便于与总线进行相应的连接操作。

通过上面的分析可以得出，只读存储器实际上是一种可编程逻辑器件，它的地址译码器在结构上来说就是一个与阵列，而存储矩阵实际上就是一个或阵列。对于  $n$  位输入的二进制译码器，它通过  $2^n$  个与门产生所需的译码输出，每个输出对应于一个乘积项。

综上所述，只读存储器的与阵列是固定的，只要它的地址输入数相同，它们的与阵列都是相同的。但是每种不同的只读存储器的或阵列是不同的，它们通常是根据所存储的数据信息来决定的，或者说是根据要求的逻辑关系来决定的，或阵列正是通过编程来实现不同的逻辑关系的。因此，与阵列固定、或阵列可编程是只读存储器中逻辑阵列的特点。

在数字电路中，只读存储器又可以分为掩模只读存储器、可编程只读存储器、可擦除的可编程只读存储器等几种不同的类型。

掩模只读存储器中的数据是由生产厂商在生产时写入的，数据信息只能读出而不能进行改写。由于掩模只读存储器只能在生产厂商处进行加工，成本相对较高，因此它只适用于大量生产的已经定型的产品，这样产量越大，每片 ROM 的相对成本就越低。可编程只读存储器中的数据可以由用户根据自己的具体设计需要进行写入，但是一旦写入以后同样不可进行重写操作。可擦除的可编程只读存储器中的数据不但可以由用户根据自己的具体设计需要进行写入，而且还可以进行重写操作，因此它具有很大的灵活性。

在开发数字系统的过程中，设计人员常常希望能够根据自己的意愿来迅速得到存储有设计所需内容的只读存储器，这种特殊的 ROM 编程可以通过可编程只读存储器来完成。一般来说，PROM 的总体电路结构与普通的 ROM 完全相同，它也是由地址译码器、存储矩阵和输出缓冲器 3 个部分构成的。不同之处在于 PROM 在出厂时，其存储矩阵的发射极全部采用熔丝与位线进行了连接。这样，用户在使用 PROM 时只需要设法将与位线相连的熔丝烧断，即可以完成相应数据的写入操作。

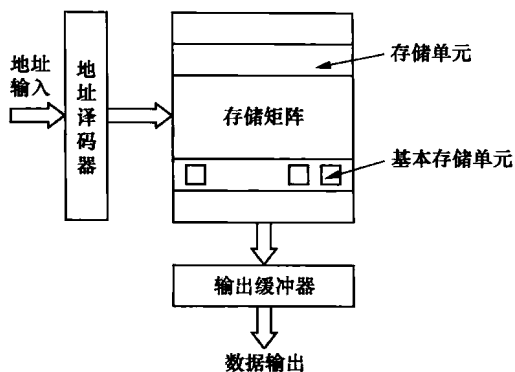


图 1-1 只读存储器的电路结构框图

可以看出, PROM 与普通 ROM 相比具有一定的灵活性, 它可以根据设计的需要来自行写入相应的存储数据。但是, 由于熔丝烧断后不能进行再次连接, 所以 PROM 的内容一经写入以后就不可能进行再次修改了, 它只能进行一次写操作。可见, 这种简单的 PROM 不能够满足数字系统设计过程中需要重复修改只读存储器内容的需要, 因此就需要一种可以进行重复写入操作的可编程只读存储器。

可擦除的可编程只读存储器 (EPROM) 的出现解决了上面的问题, 因此它在那些需要经常修改 ROM 内容的场合中获得了广泛应用。早期最常使用的 EPROM 是采用紫外线照射来进行擦除操作的, 因此这种 EPROM 被称作 UVEPROM (Ultra Violet EPROM)。如果用户想要擦除这种 EPROM 中的写入数据, 那么需要采用紫外线来对 EPROM 照射一定的时间, 目的是使浮栅上的电子获得能量以使它穿过绝缘层而回到衬底。通常, UVEPROM 器件上具有一个用来接收紫外线照射的石英窗口, 写入完成后石英窗口将会关闭以免擦除器件中的有用信息。

另外一种应用较为广泛的可擦除的可编程只读存储器是电可擦除的可编程只读存储器 (Electrically EPROM, EEPROM), 它通常也可以称作  $E^2$ PROM。对于这种 EPROM 来说, 它的擦除操作不需要采用紫外线照射, 而只需要在漏极和源极之间加上一定的擦除电压即可。更为灵活的是, 这种 EPROM 可以有选择地进行逐字擦除或者逐字改写, 同时进行擦除操作时不需要将器件从电路中取出。

在实际的应用过程中, 设计人员或者用户常常将可编程只读存储器 (PROM) 和可擦除的可编程只读存储器 (EPROM) 统称为可编程的只读存储器, 即 PROM。

## 2. 可编程逻辑阵列 (PLA)

在前面的介绍中, 我们知道只读存储器的输入地址和存储矩阵的信息之间存在着一一对应的关系, 即给定一个地址就可以选中一个存储单元。可以看出, 为了实现这一点, 只读存储器的与阵列必须是一个能够产生  $2^n$  个输出的地址译码器。但在实际应用的过程中, 不管具体设计的逻辑函数需要使用多少个最小项, 这  $2^n$  个最小项都是始终存在的。从芯片资源利用的角度来看, 没有使用的部分最小项所占用的芯片资源是可以节省下来的。

为了提高芯片资源的利用情况, 用户往往希望与阵列不需要产生所有的最小项, 它只需要产生具体设计的逻辑函数所需要的乘积项即可, PLA 即可实现上面的功能。从具体的结构上来看, PLA 就是把 ROM 中的译码器改成乘积项发生器的一种可编程逻辑器件。PLA 的结构特点是与阵列可编程, 或阵列也可以编程。根据 PLA 的结构特点可以判断: PLA 产生的乘积项的具体数目一般总是小于  $2^n$  个; 另外, PLA 中的每一个乘积项是根据设计的需要来进行输入信号组合的, 它不一定是全部  $n$  个输入信号的组合。

在数字电路的设计中, PLA 根据使用情况可以分为两大类。

(1) 一类是完全按照用户的要求, 设计一片专用的 PLA 芯片或者把 PLA 作为某种大规模集成电路的一部分。由于输入和输出数目已由设计要求规定, 所以减少所需乘积项数目成为优化的主要目标。

(2) 另外一类是现场可编程逻辑阵列 (Field PLA, FPLA), 它的最大特点是由用户自己编程写入与或阵列的内容。与第一类 PLA 相比, 这才是一种真正的可编程逻辑器件。目前, 许多芯片制造厂商都提供 FPLA 芯片, 这种芯片的输入数目、输出数目和乘积项数目都是确定的, 使用时只要不超过这些限制就可以。

这里将重点介绍 FPLA。简单的 FPLA 是由可编程的与阵列、可编程的或阵列和输出缓