



“十二五”普通高等教育本科国家级规划教材  
普通高等教育“十一五”国家级规划教材  
普通高等教育电子设计系列规划教材



# EDA技术与VHDL设计 (第2版)

◎ 徐志军 王金明 尹廷辉 等编著



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>

“十二五”普通高等教育本科国家级规划教材  
普通高等教育“十一五”国家级规划教材  
普通高等教育电子设计系列规划教材

# EDA 技术与 VHDL 设计

## (第2版)

徐志军 王金明  
尹廷辉 徐光辉 苏 勇



电子工业出版社  
Publishing House of Electronics Industry  
北京·BEIJING

## 内 容 简 介

本书根据电子信息类课程教学和实验要求,以提高学生的实践动手能力和工程设计能力为目的,对 EDA 技术和 VHDL 设计的相关知识进行系统和完整的介绍。全书共 10 章,主要内容包括:EDA 技术概述、可编程逻辑器件基础、典型 FPGA/CPLD 的结构与配置、原理图与宏功能模块设计、VHDL 设计输入方式、VHDL 结构与要素、VHDL 基本语句与基本设计、VHDL 设计进阶、数字接口实例及分析、通信算法实例及分析等。本书内容新颖,技术先进,由浅入深,既有关于 EDA 技术、大规模可编程逻辑器件和 VHDL 硬件描述语言的系统介绍,又有丰富的设计应用实例。本书提供配套电子课件、程序代码和习题参考答案。

本书可作为高等学校电子、通信、雷达、计算机应用、工业自动化、仪器仪表、信号与信息处理等学科本科生或研究生的 EDA 技术或数字系统设计课程的教材和实验指导书,也可作为相关科研人员的技术参考书。

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

EDA 技术与 VHDL 设计 / 徐志军等编著. —2 版. —北京:电子工业出版社,2015.2

普通高等教育电子设计系列规划教材

ISBN 978-7-121-25178-8

I. ①E… II. ①徐… III. ①电子电路—电路设计—计算机辅助设计—高等学校—教材 ②VHDL 语言—程序设计—高等学校—教材 IV. ①TN702 ②TP312

中国版本图书馆 CIP 数据核字(2014)第 298387 号

策划编辑:王羽佳

责任编辑:王羽佳 文字编辑:王晓庆

印 刷:三河市华成印务有限公司

装 订:三河市华成印务有限公司

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:787×1092 1/16 印张:20.75 字数:600 千字

版 次:2009 年 1 月第 1 版

2015 年 2 月第 2 版

印 次:2015 年 2 月第 1 次印刷

印 数:3000 册 定价:45.00 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。

质量投诉请发邮件至 [zltz@phei.com.cn](mailto:zltz@phei.com.cn), 盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

服务热线:(010)88258888。

# 前 言

EDA (Electronic Design Automation, 电子设计自动化) 技术是 20 世纪 90 年代以来迅速发展起来的电子设计新技术, 它以可编程逻辑器件为载体, 以计算机为工作平台, 以 EDA 软件工具为开发环境, 以硬件描述语言 (HDL) 为电子系统的功能描述方式, 以电子系统设计为目标, 在教学、科研, 以及大学生电子设计竞赛等应用场合中起着越来越重要的作用。

EDA 技术目前成为电子类本科生必须掌握的专业基础知识与基本技能, 国内许多高校的相关学科已将 EDA 技术作为一门重要的专业基础课程。随着教学改革的深入, 对 EDA 课程教学的要求也在不断提高, 为与 EDA 技术的发展相适应, 必须对教学内容进行更新和优化。

我们认为在 EDA 教学中应注意如下几点。

首先, 要明确最基本的教学内容, 并突出重点。EDA 技术教学的目的是使学生掌握一种通过软件的方法来高效地完成硬件设计的设计技术, 应以培养学生的创新思维和设计思想为主, 同时使学生掌握基本的设计工具和设计方法。

其次, 要改进教学方法。EDA 教学应主要以引导性教学为主, 合理安排理论教学和实验教学学时比例, 使学生能够理论联系实际, 提高实践动手能力和工程设计能力。

再次, 要注重教学实效。EDA 课程具有很强的实践性, 针对性强的实验应该是教学的重要环节, 应格外重视 EDA 实验的质量。

基于以上的认识, 我们安排了本书的章节, 本书是以可编程逻辑器件、EDA 设计工具、VHDL 硬件描述语言三方面内容为主线展开的, 贯穿其中的则是现代数字设计的新思想、新方法。

本书是在普通高等教育“十一五”国家级规划教材《EDA 技术与 VHDL 设计》基础上编写的, 全书共 10 章, 主要内容涵盖了 EDA 技术的硬件资源、软件操作和设计应用。

第 1 章对 EDA 技术做了综述, 介绍了 EDA 技术的发展、EDA 设计流程及 EDA 技术涉及的领域。第 2 章介绍可编程逻辑器件的基本概念、结构组成和工作原理, 可编程逻辑器件的编程工艺及测试技术等。第 3 章具体介绍 Altera 公司典型的 FPGA/CPLD 器件的结构与配置。第 4 章介绍使用集成工具 Quartus II 软件进行设计开发的过程, 并介绍宏功能模块的设计与应用。第 5 章介绍基于 VHDL 的设计过程及 VHDL 综合工具的使用方法。第 6 章介绍 VHDL 的语法、结构与要素。第 7 章介绍 VHDL 的语句及常用组合电路、时序电路的 VHDL 设计。第 8 章结合具体实例介绍用 VHDL 进行设计的方法。第 9 章是用 VHDL 进行数字接口开发的实例。第 10 章是数字通信用常用算法与模块的设计实例。

为了方便使读者能够较系统、完整地学习 EDA 技术, 掌握 EDA 设计基本技能, 本书从教学的角度出发, 尽量将有关 EDA 技术的内容编入书中, 并力求内容精炼, 语言通俗易懂。读者也可以根据实际需要, 节选学习书中的部分内容, 然后再通过相关 EDA 技术书籍的学习, 达到掌握 EDA 技术的目的。

本书的教学可安排 32~40 学时, 其中第 1 章占 2 学时, 第 2 章占 4 学时, 第 3 章占 4 学时, 第 4 章占 4 学时, 第 5 章占 4 学时, 第 6 章占 4 学时, 第 7 章占 4~6 学时, 第 8 章占 2~4 学时, 第 9 章占 2~4 学时, 第 10 章占 2~4 学时。建议安排 8~16 学时的实验, 第一个实验可安排 EDA 工具软件的使用方法。此外, 各学校也可根据自己的教学计划适当调整学时安排。本书提供配套电子课件、程序代码和习题参考答案等教学资源, 请登录华信教育资源网 (<http://www.hxedu.com.cn>) 注册下载。

本书由徐志军教授主编，并编写第 1、2 章，第 3、4、5 章由王金明编写，第 6、7 章由尹廷辉编写，徐光辉编写了第 8 章，苏勇编写了第 9、10 章，全书由徐志军统稿。南京航空航天大学的王成华教授审阅了全书，并提出了修改意见和建议，杭州电子科技大学的潘松老师也给予了支持和帮助，我们在此表示衷心的感谢！

本书是几位老师在多年 EDA 教学经验的基础上精心编写而成的，虽经很大努力，但由于作者水平所限，加之时间仓促，书中错误与疏漏之处在所难免，真诚地希望同行和广大读者批评指正。

作 者

2015 年 2 月

# 目 录

|                                    |    |                                             |    |
|------------------------------------|----|---------------------------------------------|----|
| 第 1 章 EDA 技术概述 .....               | 1  | 2.2.6 GAL .....                             | 37 |
| 1.1 EDA 技术及其发展历程 .....             | 1  | 2.3 CPLD/FPGA 的结构特点 .....                   | 38 |
| 1.2 EDA 技术的特征和优势 .....             | 3  | 2.3.1 Lattice 公司的 CPLD/FPGA .....           | 39 |
| 1.2.1 EDA 技术的基本特征 .....            | 4  | 2.3.2 Xilinx 公司的 CPLD/FPGA .....            | 41 |
| 1.2.2 EDA 技术的优势 .....              | 6  | 2.3.3 Altera 和 Actel 公司的<br>CPLD/FPGA ..... | 43 |
| 1.3 EDA 设计的目标和流程 .....             | 7  | 2.3.4 CPLD 和 FPGA 的异同 .....                 | 44 |
| 1.3.1 EDA 技术的实现目标 .....            | 8  | 2.4 可编程逻辑器件的基本资源 .....                      | 45 |
| 1.3.2 EDA 设计流程 .....               | 8  | 2.4.1 功能单元 .....                            | 45 |
| 1.3.3 数字集成电路的设计 .....              | 9  | 2.4.2 输入-输出焊盘 .....                         | 46 |
| 1.3.4 模拟集成电路的设计 .....              | 10 | 2.4.3 布线资源 .....                            | 47 |
| 1.4 EDA 技术与 ASIC 设计 .....          | 11 | 2.4.4 片内 RAM .....                          | 48 |
| 1.4.1 ASIC 的特点与分类 .....            | 11 | 2.5 可编程逻辑器件的编程工艺 .....                      | 49 |
| 1.4.2 ASIC 的设计方法 .....             | 12 | 2.5.1 熔丝型开关 .....                           | 50 |
| 1.4.3 SoC 设计 .....                 | 15 | 2.5.2 反熔丝型开关 .....                          | 50 |
| 1.5 硬件描述语言 .....                   | 18 | 2.5.3 浮栅编程器件 .....                          | 51 |
| 1.5.1 VHDL .....                   | 18 | 2.5.4 基于 SRAM 的编程器件 .....                   | 53 |
| 1.5.2 Verilog HDL .....            | 19 | 2.6 可编程逻辑器件的设计与开发 .....                     | 54 |
| 1.5.3 AHDL .....                   | 19 | 2.6.1 CPLD/FPGA 设计流程 .....                  | 54 |
| 1.5.4 Verilog HDL 和 VHDL 的比较 ..... | 20 | 2.6.2 CPLD/FPGA 开发工具 .....                  | 56 |
| 1.6 EDA 设计工具 .....                 | 20 | 2.6.3 CPLD/FPGA 的应用选择 .....                 | 58 |
| 1.6.1 EDA 设计工具分类 .....             | 21 | 2.7 可编程逻辑器件的测试技术 .....                      | 61 |
| 1.6.2 EDA 公司与工具介绍 .....            | 22 | 2.7.1 边界扫描测试原理 .....                        | 61 |
| 习题 1 .....                         | 25 | 2.7.2 IEEE 1149.1 标准 .....                  | 62 |
| 第 2 章 可编程逻辑器件基础 .....              | 26 | 2.7.3 边界扫描策略及相关工具 .....                     | 65 |
| 2.1 概述 .....                       | 26 | 习题 2 .....                                  | 66 |
| 2.1.1 可编程逻辑器件发展历程 .....            | 26 | 第 3 章 典型 FPGA/CPLD 的结构与配置 .....             | 67 |
| 2.1.2 可编程逻辑器件分类 .....              | 27 | 3.1 Stratix 高端 FPGA 系列 .....                | 67 |
| 2.1.3 可编程逻辑器件的优势 .....             | 30 | 3.1.1 Stratix 器件 .....                      | 67 |
| 2.1.4 可编程逻辑器件的发展趋势 .....           | 30 | 3.1.2 Stratix II 器件 .....                   | 70 |
| 2.2 PLD 器件的基本结构 .....              | 32 | 3.2 Cyclone 低成本 FPGA 系列 .....               | 73 |
| 2.2.1 基本结构 .....                   | 32 | 3.2.1 Cyclone 器件 .....                      | 73 |
| 2.2.2 电路符号 .....                   | 33 | 3.2.2 Cyclone II 器件 .....                   | 77 |
| 2.2.3 PROM .....                   | 34 | 3.3 典型 CPLD 器件 .....                        | 83 |
| 2.2.4 PLA .....                    | 35 | 3.3.1 MAX II 器件 .....                       | 83 |
| 2.2.5 PAL .....                    | 36 |                                             |    |

|       |                                   |     |       |                |     |
|-------|-----------------------------------|-----|-------|----------------|-----|
| 3.3.2 | MAX 7000 器件                       | 84  | 6.1.3 | 实体描述举例         | 148 |
| 3.4   | FPGA/CPLD 的配置                     | 86  | 6.2   | 结构体            | 149 |
| 3.4.1 | CPLD 器件的配置                        | 87  | 6.2.1 | 结构体的命名         | 149 |
| 3.4.2 | FPGA 器件的配置                        | 88  | 6.2.2 | 结构体信号定义语句      | 150 |
| 习题 3  |                                   | 92  | 6.2.3 | 结构体功能描述语句      | 150 |
| 第 4 章 | 原理图与宏功能模块设计                       | 93  | 6.2.4 | 结构体描述方法        | 150 |
| 4.1   | Quartus II 原理图设计                  | 93  | 6.3   | VHDL 库         | 152 |
| 4.1.1 | 半加器原理图输入                          | 93  | 6.3.1 | 库的种类           | 152 |
| 4.1.2 | 半加器编译                             | 96  | 6.3.2 | 库的用法           | 153 |
| 4.1.3 | 半加器仿真                             | 98  | 6.4   | VHDL 程序包       | 155 |
| 4.1.4 | 全加器设计与仿真                          | 100 | 6.4.1 | 程序包组成和格式       | 155 |
| 4.2   | Quartus II 的优化设置                  | 101 | 6.4.2 | VHDL 标准程序包     | 156 |
| 4.2.1 | Settings 设置                       | 101 | 6.5   | 配置             | 157 |
| 4.2.2 | 分析与综合设置                           | 103 | 6.5.1 | 默认配置           | 157 |
| 4.2.3 | 优化布局布线                            | 103 | 6.5.2 | 结构体的配置         | 159 |
| 4.2.4 | 使用设计助手检查设计可靠性                     | 109 | 6.6   | VHDL 文字规则      | 161 |
| 4.3   | Quartus II 的时序分析                  | 111 | 6.6.1 | 标识符            | 161 |
| 4.3.1 | 时序设置与分析                           | 111 | 6.6.2 | 数字             | 162 |
| 4.3.2 | 时序逼近                              | 114 | 6.6.3 | 字符串            | 162 |
| 4.4   | 宏功能模块设计                           | 116 | 6.7   | VHDL 数据类型      | 163 |
| 4.4.1 | Megafunctions 库                   | 116 | 6.7.1 | 预定义数据类型        | 163 |
| 4.4.2 | Maxplus2 库                        | 125 | 6.7.2 | 自定义数据类型        | 165 |
| 4.4.3 | Primitives 库                      | 126 | 6.7.3 | 用户自定义的子类型      | 167 |
| 习题 4  |                                   | 128 | 6.7.4 | 数据类型的转换        | 167 |
| 第 5 章 | VHDL 设计输入方式                       | 131 | 6.8   | VHDL 操作符       | 169 |
| 5.1   | Quartus II 的 VHDL 输入设计            | 131 | 6.8.1 | 逻辑操作符          | 169 |
| 5.1.1 | 创建工程文件                            | 132 | 6.8.2 | 关系操作符          | 169 |
| 5.1.2 | 编译                                | 133 | 6.8.3 | 算术操作符          | 170 |
| 5.1.3 | 仿真                                | 134 | 6.8.4 | 并置操作符          | 171 |
| 5.2   | Synplify Pro 的 VHDL 输入设计          | 136 | 6.8.5 | 操作符重载          | 171 |
| 5.2.1 | 用 Synplify Pro 综合的过程              | 137 | 6.9   | 数据对象           | 172 |
| 5.2.2 | Synplify Pro 与 Quartus II 的<br>接口 | 141 | 6.9.1 | 常量             | 172 |
| 5.3   | Synplify 的 VHDL 输入设计              | 141 | 6.9.2 | 变量             | 173 |
| 习题 5  |                                   | 144 | 6.9.3 | 信号             | 174 |
| 第 6 章 | VHDL 结构与要素                        | 145 | 6.9.4 | 文件             | 174 |
| 6.1   | 实体                                | 145 | 习题 6  |                | 175 |
| 6.1.1 | 类属参数说明                            | 146 | 第 7 章 | VHDL 基本语句与基本设计 | 176 |
| 6.1.2 | 端口说明                              | 147 | 7.1   | 顺序语句           | 176 |
|       |                                   |     | 7.1.1 | 赋值语句           | 176 |
|       |                                   |     | 7.1.2 | IF 语句          | 176 |

|       |                      |     |        |                           |     |
|-------|----------------------|-----|--------|---------------------------|-----|
| 7.1.3 | CASE 语句              | 179 | 9.2.2  | 同步数字复接分接接口设计<br>实例        | 248 |
| 7.1.4 | LOOP 语句              | 180 | 9.3    | I <sup>2</sup> C 接口设计     | 254 |
| 7.1.5 | NEXT 语句              | 182 | 9.3.1  | I <sup>2</sup> C 总线工作原理   | 255 |
| 7.1.6 | EXIT 语句              | 182 | 9.3.2  | I <sup>2</sup> C 总线接口设计实例 | 257 |
| 7.1.7 | WAIT 语句              | 183 | 9.4    | GMSK 基带调制接口设计             | 263 |
| 7.1.8 | 子程序调用语句              | 184 | 9.4.1  | GMSK 调制基本原理               | 263 |
| 7.2   | 并行语句                 | 186 | 9.4.2  | GMSK 调制实现的基本方法            | 265 |
| 7.2.1 | 并行信号赋值语句             | 186 | 9.4.3  | GMSK 基带调制接口的实现<br>代码      | 266 |
| 7.2.2 | 进程语句                 | 189 | 习题 9   |                           | 284 |
| 7.2.3 | 并行过程调用语句             | 190 | 第 10 章 | 通信算法实例及分析                 | 285 |
| 7.2.4 | 元器件例化语句              | 191 | 10.1   | 伪随机序列的产生、检测设计             | 285 |
| 7.2.5 | 生成语句                 | 193 | 10.1.1 | m 序列的产生                   | 285 |
| 7.3   | VHDL 组合逻辑电路设计        | 196 | 10.1.2 | m 序列的性质                   | 286 |
| 7.4   | VHDL 时序逻辑电路设计        | 202 | 10.1.3 | m 序列发生器的 VHDL 设计          | 286 |
| 7.4.1 | 触发器                  | 202 | 10.1.4 | m 序列检测电路的 VHDL<br>设计      | 288 |
| 7.4.2 | 寄存器                  | 204 | 10.2   | 比特同步设计                    | 292 |
| 7.4.3 | 计数器                  | 205 | 10.2.1 | 锁相功能的自同步法原理               | 292 |
| 7.4.4 | 分频器                  | 206 | 10.2.2 | 锁相比特同步的 EDA 实现<br>方法      | 294 |
| 习题 7  |                      | 209 | 10.3   | 基带差分编码设计                  | 303 |
| 第 8 章 | VHDL 设计进阶            | 210 | 10.3.1 | PSK 调制和差分编码原理             | 303 |
| 8.1   | VHDL 行为描述方式          | 210 | 10.3.2 | PSK 差分编码设计                | 306 |
| 8.2   | VHDL 结构化描述方式         | 212 | 10.4   | FIR 滤波器设计                 | 312 |
| 8.3   | VHDL RTL 描述方式        | 215 | 10.4.1 | FIR 滤波器简介                 | 312 |
| 8.4   | 有限状态机 (FSM) 设计       | 216 | 10.4.2 | 使用 MATLAB 设计 FIR<br>滤波器   | 314 |
| 8.4.1 | Moore 和 Mealy 状态机的选择 | 216 | 10.4.3 | FIR 滤波器的 FPGA 普通<br>设计    | 315 |
| 8.4.2 | 有限状态机的描述方式           | 218 | 10.4.4 | FIR 滤波器的并行 FPGA 优化<br>设计  | 317 |
| 8.4.3 | 有限状态机的同步和复位          | 226 | 习题 10  |                           | 323 |
| 8.4.4 | 改进的 Moore 型有限状态机     | 232 | 参考文献   |                           | 324 |
| 8.4.5 | 小结                   | 237 |        |                           |     |
| 习题 8  |                      | 238 |        |                           |     |
| 第 9 章 | 数字接口实例及分析            | 240 |        |                           |     |
| 9.1   | ST-BUS 总线接口设计        | 240 |        |                           |     |
| 9.1.1 | ST-BUS 总线时序关系        | 240 |        |                           |     |
| 9.1.2 | ST-BUS 总线接口实例        | 242 |        |                           |     |
| 9.2   | 数字复接分接接口技术及设计        | 246 |        |                           |     |
| 9.2.1 | 数字复接分接接口技术原理         | 247 |        |                           |     |



# 第1章 EDA 技术概述

**本章概要：**本章主要介绍 EDA 技术的发展、EDA 技术的实现目标、EDA 设计流程和设计工具、EDA 技术涉及的领域。

- 知识要点：**
- (1) EDA 技术的特征；
  - (2) EDA 设计的目标和流程；
  - (3) “自顶向下”的设计方法；
  - (4) EDA 技术与 ASIC 设计；
  - (5) EDA 设计工具。

**教学安排：**本章教学安排 2 学时。通过本章的学习，读者可了解 EDA 的基本概念，熟悉 EDA 设计的流程，了解 EDA 设计工具的主要功能。

## 1.1 EDA 技术及其发展历程

信息社会的发展离不开集成电路，当前集成电路正朝着速度快、容量大、体积小、功耗低的方向发展，实现这种进步的主要原因就是生产制造技术和电子设计技术的发展。前者以微细加工技术为代表，目前已进展到纳米阶段，可以在几平方厘米的芯片上集成数亿个晶体管；后者的核心就是 EDA (Electronic Design Automation) 技术，目前已经渗透到电子产品设计的各个环节。

EDA 是电子设计自动化的英文缩写，是随着集成电路和计算机技术飞速发展应运而生的一种快速、有效、高级的电子设计自动化技术。EDA 工具融合了应用电子技术、计算机技术和智能化技术的最新成果，主要进行三方面的辅助设计工作：集成电路 (IC) 设计、电子电路设计及印制电路板 (PCB) 设计。在数字设计领域，EDA 技术就是依靠功能强大的电子计算机和 EDA 软件工具，对以硬件描述语言 HDL (Hardware Description Language) 形式给出的系统设计文件自动地进行逻辑编译、化简、分割、综合、优化和仿真，直至下载到可编程逻辑器件 CPLD/FPGA 或专用集成电路芯片中，实现既定的电路功能。EDA 技术使电路设计者的工作仅限于利用硬件描述语言和 EDA 软件平台来完成对系统硬件功能的实现，极大地提高了设计效率，缩短了设计周期，节约了设计成本。

EDA 技术的发展历程同大规模集成电路技术、计算机技术、可编程逻辑器件，以及电子设计技术和工艺技术的发展是同步的。回顾 60 多年来电子技术的发展历程，可以将电子设计自动化技术大致分为三个发展阶段，如图 1.1 所示。

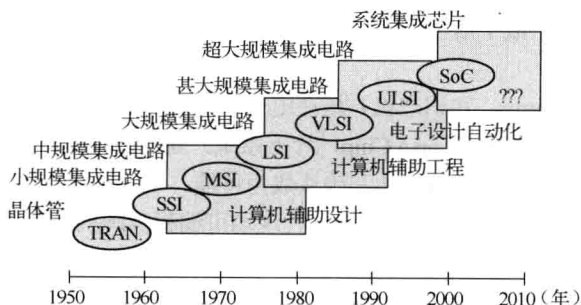


图 1.1 EDA 技术的不同发展阶段

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。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(PCB)布局布线和 IC 版图编辑等方面取代了设计人员烦琐的手工计算和操作,大大提高了电子系统和集成电路设计的效率和可靠性,从而产生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、集成电路的布局布线、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模拟功能。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, Computer Aided Engineering)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速率已经达到 10Gb/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。例如,数字声广播(DAB)与音响系统、高清晰度电视(HDTV)、多媒体信息处理与传播、光通信等电子系统,它们对集成电路和专用集成电路(ASIC)的容量、速度、频带等都提出了更高的要求,这种高难度的 IC 要在短时间内正确地设计成功,必须将 EDA 技术提高到一个更高的水平。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电路的版图,并对版图的面积及电路延时特性进行优化处理。

进入 21 世纪以后,EDA 技术得到了更大的发展,开始步入了一个崭新的时期,突出地表现在以下几个方面。

(1) 电子技术各个领域全方位融入 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——软件无线电技术崛起、模拟电路硬件描述语言的表达和设计标准化、在系统可编程模拟器件出现、数字信号处理和图像处理的全硬件实现方案推出、软硬件技术进一步融合等。

(2) IP (Intellectual Property, 知识产权)核在电子产业的产业领域、技术领域和设计领域得到了广泛应用,基于 IP 核的 SoC (System on a Chip, 片上系统)高效低成本设计技术趋向成熟,使电子设计成果以自主知识产权的方式得以明确表达和确认成为可能。

(3) 在 FPGA (Field Programmable Gate Array, 现场可编程门阵列)上实现 DSP (数字信号处理)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。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。

(4) 嵌入式微处理器软核的出现,更大规模 FPGA/CPLD 器件的不断推出,使得 SOPC (System

On a Programmable Chip, 可编程片上系统)步入了大规模应用阶段,在一片 FPGA 芯片中实现一个完备的数字信号处理系统成为可能。

(5) 在仿真和设计两方面支持标准硬件描述语言的 EDA 软件不断推出,系统级、行为验证级硬件描述语言的出现(如 System C)使得复杂电子系统的设计和验证趋于简单。

(6) EDA 技术使得电子领域各学科的界限更加模糊、更加相互包容和渗透,如模拟与数字、软件与硬件、系统与器件、ASIC 与 FPGA、行为与结构等的基于 EDA 工具的 ASIC 设计标准单元已涵盖大规模电子系统及 IP 核模块。

EDA 技术为现代电子学理论和设计理念的表达与实现提供了可能性。在硬件实现方面,EDA 技术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/CPLD 编程下载技术、自动测试技术等;在工程实现方面,EDA 技术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)技术及多种计算机语言的设计概念;而在现代电子学方面,EDA 技术则容纳了更多的内容,如电路基础理论、数字信号处理技术、数字系统建模、优化设计技术等。因此,现代 EDA 技术已经不是某一学科的分支或某种新的技能技术,而应该是一门综合性学科。它融合多学科于一体,又渗透于各学科之中,打破了软件与硬件间的壁垒,使计算机的软件技术与硬件实现、设计效率和产品性能合二为一,代表了现代电子设计技术和应用技术的发展方向。随着科学技术的进步和市场需求的不断增长,EDA 技术将呈现出以下发展趋势。

一是 EDA 开发工具将进一步得到完善。EDA 开发工具将朝着功能强大、简单易学、使用方便的方向发展,主要体现在 EDA 工具的 PC 平台化、灵活多样的设计输入工具、更为有效的仿真工具、更为理想的综合工具等几个方面。

二是 EDA 技术将促使 ASIC 和 FPGA 逐步走向融合。随着系统开发对 EDA 技术的目标器件各种性能指标要求的提高,ASIC 和 FPGA 将更大程度地相互融合。这是因为,虽然标准逻辑 ASIC 芯片尺寸小、功能强大、耗电省,但却设计复杂,并且有批量生产要求;可编程逻辑器件的开发费用低廉,能在现场进行编程,但却体积大、功能有限,而且功耗较大。因此,FPGA 和 ASIC 正在走到一起,两者之间正在诞生一种“杂交”产品,互相融合,取长补短,以满足成本和上市速度的要求。目前传统 ASIC 设计和 FPGA 之间的界限正变得模糊。系统级芯片不仅集成 RAM 和微处理器,也集成 FPGA。整个 EDA 和 IC 设计工业都在朝这个方向发展,这并非是 FPGA 与 ASIC 制造商竞争的产物,而对于用户来说,意味着有了更多的选择。

三是 EDA 技术的应用领域日益广泛。现代电子系统的设计将呈现以下特点:用软件的方式设计硬件,设计过程中可用有关软件进行各种仿真,系统现场可编程和在线升级,整个系统可集成在一个芯片上。这些特点使得 EDA 技术将广泛应用于科研和新产品的开发工作中。此外,传统机电设备的电气控制系统,如果利用 EDA 技术进行重新设计或技术改造,不但可以缩短设计周期,降低设计成本,而且还将提高产品和设备的性能及可靠性,缩小产品体积,提高产品的技术含量和附加值。

## 1.2 EDA 技术的特征和优势

在现代电子设计领域,EDA 技术已经成为电子系统设计的重要手段。无论是设计数字系统还是集成电路芯片,其设计作业的复杂程度都在不断增加,仅仅依靠手工进行设计已经不能满足要求,所有的设计工作都需要在计算机上借助 EDA 软件工具进行。在 EDA 软件的支持下,设计者只需完成对系统功能的描述,就可以由计算机软件进行处理,得到设计结果,修改设计如同修改软件一样方便。利用 EDA 设计工具,设计者可以预知设计结果,减少设计的盲目性,极大地提高了设计的效率。

### 1.2.1 EDA 技术的基本特征

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,具有开放式的设计环境,具有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 (Hardware Description Language) 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,硬件描述语言更适合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下而上地构成电路、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通用芯片,整机系统用户采用这些芯片组成电子系统的 Bottom-up (自底向上) 设计方法改变为一种新的 Top-down (自顶向下) 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 ASIC 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。图 1.2 所示为电子系统的两种不同的设计步骤。

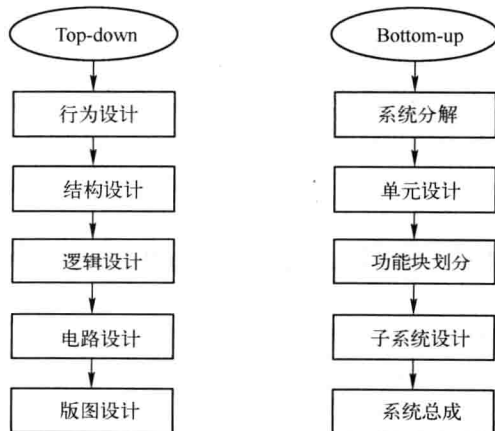


图 1.2 “自顶向下”与“自底向上”设计

“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的具体电路、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“采用什么电路、元器件和工艺去实现该设计”这类具体问题。

在进行“自顶向下”的设计时，首先从系统级设计入手，在顶层进行功能方框图的划分和结构设计；在方框图一级进行仿真、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，然后再进行子系统的设计，最后完成系统的总成设计。“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。

逻辑综合的特点是将高层次的系统行为设计自动翻译成门级逻辑的电路描述，做到了设计与工艺的相互独立。逻辑综合的作用是根据一个系统的逻辑功能与性能的要求，在一个包含众多结构、功能和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的最佳（至少是较佳的）实现方案。

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：主要是进行逻辑化简与优化，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 系统只要建立了一个符合标准的开放式框架结构，就可以接纳其他厂商的 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,从而进一步提高了 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 (Library),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,版图生成时需要适应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 输入所需的库则更为庞大和齐全,几乎包括了上述所有库的内容。各种模型库的规模和功能是衡量 EDA 工具优劣的一个重要标识。

### 1.2.2 EDA 技术的优势

传统的数字系统设计一般是采用搭“积木块”的手工设计方式,即由元器件搭成电路板,由电路板搭成电子系统。数字系统最初的“积木块”是标准的集成电路,如 74/54 系列 (TTL)、4000/4500 系列 (CMOS) 芯片和一些固定功能的大规模集成电路。在设计数字电路时,一般先按照数字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逻辑真值表和状态转换真值表,用卡诺图进行手工逻辑化简和状态化简,写出布尔表达式,画出相应的逻辑线路图,再据此选择合适的器件,并按照器件推荐的电路设计电路板,最后进行实测与调试。

手工设计方法有很多缺点:如对于复杂电路的设计,调试十分困难;对设计过程中出现的错误,查找和修改十分不便;设计过程中产生大量文档,不易管理;只有在设计出样机或生产出芯片后才能进行实测;等等。

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) 采用硬件描述语言,便于复杂系统的设计

从电子设计方法学来看,EDA 技术的最大优势是能够将所有的设计环节纳入统一的自顶向下的设计方案中。用 HDL 对数字电子系统进行结构描述、功能描述和行为描述,从而可以在电子设计的各个阶段和各个层进行计算机模拟验证,保证了设计过程的正确性,降低了设计成本,缩短了设计周期。此外,某些硬件描述语言 (如 VHDL) 也是文档型的语言,可以极大地简化设计文档的管理。

#### (2) 强大的系统建模和电路仿真功能

EDA 技术中最为瞩目的功能是日益强大的仿真测试技术。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进行各种不同层次的性能测试和逻辑仿真,在实际系统完成后,还能对系统上的目标器件进行边界扫描测试,极大地提高了大规模电子系统的设计自动化程度。

#### (3) 具有自主的知识产权

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部件,都掩盖不了一个无情的事实,即该系统对于设计者来说,没有任何的知识产权可言。因为系统中的关键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,而且这种情况有时是致命的 (如系统中某些关键器件失去供货来源、应用于军事电子装备中的关键器件等)。基于 EDA 技术的设计则不同,由于用 HDL 表达的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 FPGA 器件实现,也可以直接以 ASIC 来实现,设计者拥有完全的自主权,再也不用受制于人。



#### (4) 开发技术的标准化和规范化

传统的电子设计方法至今没有任何标准规范加以约束，设计效率低，系统性能差，开发成本高，市场竞争能力弱。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；EDA 开发工具是规范化的，它支持任何标准化的设计语言；EDA 技术的设计成果是通用性的，IP 核具有规范的接口协议；良好的可移植性与可测试性，为系统开发提供了可靠的保证。

#### (5) 全方位地利用计算机的自动设计、仿真和测试技术

EDA 不但在整个设计流程上充分利用了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的测试。对于传统的设计方法，如单片机仿真器的使用，仅仅只能在最后完成的系统上进行局部的软件仿真调试，在整个设计的中间过程中则是无能为力的。至于硬件系统测试，由于现在的许多系统主板不但层数多，而且许多器件都是 BGA (Ball-Grid Array) 封装，所有引脚都在芯片的底面，焊接后普通的仪器仪表无法接触到所需要的信号点，因此无法测试。

#### (6) 对设计者的硬件知识和硬件经验要求低

传统的电子设计对于电子工程师的要求似乎很高：在电子技术理论和设计实际方面必须是行家；不但应该是软件高手，同时还应该是经验丰富的硬件设计能工巧匠；必须熟悉针对不同单片机或 DSP 器件开发系统的使用方法和性能，还必须知道许多器件的封装形式和电气特性，知道不同的在线测试仪表的使用方法和性能指标；要熟练掌握大量的与设计理论和优化技术毫无关系的技能技巧，不得不事无巨细，事必躬亲。

所有这一切显然不符合现代电子技术的发展需求，首先不符合快速换代的产品市场要求，不符合需求巨大的人才市场的要求。EDA 技术的标准化和 HDL 设计语言与设计平台对具体硬件的无关性，使设计者能更大程度地将自己的才能和创造力集中在设计项目性能的提高和成本的降低上，而将更具体的硬件实现工作交给专业部门来完成。显然，高技术人才比经验性人才的培养效率要高得多。

### 1.3 EDA 设计的目标和流程

EDA 技术的范畴应包括电子工程师进行产品开发的全过程，以及电子产品生产的全过程中期望由计算机提供的各种辅助工作。从一个角度来看，EDA 技术可粗略分为系统级、电路级和物理实现级三个层次的辅助设计过程；从另一个角度来看，EDA 技术应包括电子电路设计的各个领域，即从低频电路到高频电路、从线性电路到非线性电路、从模拟电路到数字电路、从分立电路到集成电路的全部设计过程。EDA 技术的范畴如图 1.3 所示。

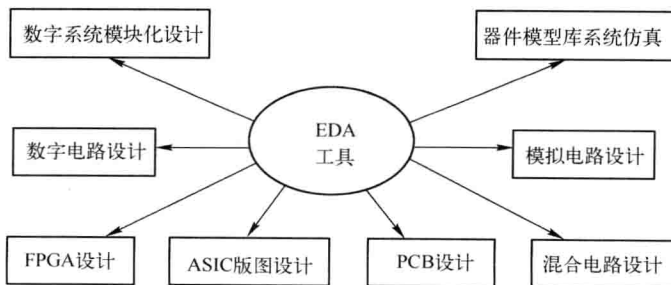


图 1.3 EDA 技术的范畴

### 1.3.1 EDA 技术的实现目标

一般来说,利用 EDA 技术进行电子系统设计,归纳起来主要有以下 4 个应用领域:印制电路板(PCB)设计、集成电路(IC 或 ASIC)设计、可编程逻辑器件(FPGA/CPLD)设计和混合电路设计。

印制电路板设计是 EDA 技术最初的实现目标。电子系统大多采用印制电路板的结构。在系统实现过程中,印制电路板的设计、装配和测试占据了很大的工作量。印制电路板设计是一个电子系统进行技术实现的重要环节,也是一个很具有工艺性、技巧性的工作。利用 EDA 工具来进行印制电路板的布局布线设计和验证分析是早期 EDA 技术最基本的应用。

集成电路是指通过一系列特定的加工工艺,将晶体管、二极管等有源器件和电阻、电容等无源器件,按照一定的电路互连,“制作”(集成)在一块半导体单晶薄片上,经过封装而形成的具有特定功能的完整电路。集成电路一般要通过“掩膜”来制作,按照实现的工艺,又分为全定制或半定制的集成电路。集成电路设计包括逻辑(或功能)设计、电路设计、版图设计和工艺设计多个环节。随着大规模和超大规模集成电路规模的出现,传统的手工设计方法遇到的困难越来越多,为了保证设计的正确性和可靠性,必须采用先进的 EDA 软件工具来进行集成电路的逻辑设计、电路设计和版图设计。集成电路设计是 EDA 技术的最终实现目标,也是推动 EDA 技术推广和发展的一个重要源泉。

可编程逻辑器件 PLD (Programmable Logic Device) 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,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,开发成本低,上市时间短,工作可靠性高。可编程逻辑器件目前主要有两大类型:复杂可编程逻辑器件 CPLD (Complex PLD) 和 FPGA (Field Programmable Gate Array)。它们的基本设计方法是借助于 EDA 软件,用原理图、状态机、布尔表达式、硬件描述语言等方法,生成相应的目标文件,最后用编程器或下载电缆,由目标器件实现。可编程逻辑器件的开发与应用是 EDA 技术将电子系统设计与硬件实现进行有机融合的一个重要体现。

随着集成电路复杂程度的不断提高,各种不同学科技术、不同模式、不同层次的混合设计方法已被认为是 EDA 技术所必须支持的方法。不同学科的混合设计方法 (Mixed-discipline) 主要指电子技术与非电学科技术的混合设计方法;不同模式的混合方法 (Mixed-mode) 主要指模拟电路与数字电路的混合,模拟电路与 DSP 技术的混合,电路级与器件级的混合等;不同层次的混合方法 (Multi-level) 主要指逻辑设计中行为级、寄存器级、门级和开关级的混合设计方法。目前在各种应用领域,如数字电路、模拟电路、DSP 专用集成电路、多芯片模块 (MCM, Multi-Chip Module) 及印制电路系统的设计中都需要采用各种混合设计方法。

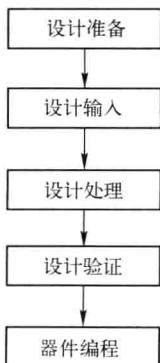


图 1.4 典型的 EDA 设计流程

### 1.3.2 EDA 设计流程

利用 EDA 技术进行电路设计的大部分工作是在 EDA 软件平台上进行的。一个典型的 EDA 设计流程主要包括设计准备、设计输入、设计处理、设计验证和器件编程等 5 个基本步骤,如图 1.4 所示。

设计输入有多种方式,包括采用硬件描述语言(如 AHDL、VHDL 和 Verilog HDL 等)进行设计的文本输入方式、图形输入方式和波形输入方式,或者采用文本、图形两者混合的设计输入方式,也可以采用“自顶向下”的层次结构设计方法,将多个输入文件合并成一个设计文件等。

设计处理是 EDA 设计中的核心环节。在设计处理阶段,编译软件将对设计输入文件进行逻辑化简、综合和优化,并适当地用一片或多片器件自动地进行适配,最后产



生编程用的编程文件。设计处理主要包括设计编译和检查、逻辑优化和综合、适配和分割、布局和布线、生成编程数据文件等过程。

设计验证过程包括功能仿真和时序仿真，这两项工作是在设计处理过程中同时进行的。功能仿真是在设计输入完成以后，选择具体器件进行编译之前的逻辑功能验证，因此又称为前仿真。此时的仿真没有延时信息或只有由系统添加的微小标准延时，这对于初步的功能检测非常方便。仿真前，先要利用波形编辑器或硬件描述语言等建立波形文件或测试向量（即将所关心的输入信号组合成序列），仿真结果将会生成报告文件和输出信号波形，从中可以观察到各个节点信号变化，若发现错误，则返回设计输入方式，修改逻辑设计。时序仿真是在选择了具体器件并完成布局、布线之后进行的时序关系仿真，因此又称为后仿真或时延仿真。由于不同器件的内部延时不一样，不同的布局、布线方案也会给延时造成不同的影响，因此在设计处理以后，对系统的各个模块进行时序仿真、分析其时序关系、估计设计的性能，以及检查和消除竞争冒险等都是非常必要的。

器件编程是将设计处理中产生的编程数据文件通过软件放到具体的可编程逻辑器件中去。对于 CPLD 器件来说，就是将熔丝图 JED 文件下载（Down）到 CPLD 器件中去；对 FPGA 器件来说，就是将位流数据文件配置到 FPGA 器件中去。

设计验证可以在 EDA 硬件开发平台上进行。EDA 硬件开发平台的核心是一片可编程逻辑器件 FPGA 或 CPLD，再附加一些输入/输出设备，如按键、数码显示器、指示灯、扬声器等，还提供时序电路需要的脉冲信号源。将设计电路编程下载到 FPGA 或 CPLD 中后，根据 EDA 硬件开发平台的操作模式要求，进行相应的输入操作，然后检查输出结果，验证设计电路。

### 1.3.3 数字集成电路的设计

数字集成电路的 EDA 设计流程如图 1.5 所示。首先由系统描述开始。在这个阶段要对用户的需求、市场前景及互补产品进行充分的调研与分析，对设计模式和制造工艺的选择进行认证，最终目标是用工程化语言将待设计 IC 的技术指标、功能、外形尺寸、芯片面积、工作速度与功耗等描述出来，形成这一步的文档。

下一步是功能设计。这一阶段的工作是根据用户提出的系统指标要求，将该系统划分成若干子系统，在行为级上将 IC 的功能及其各组成子系统的功能关系正确而完整地描述出来。

然后进行逻辑设计。这一阶段的主要任务是得到一个实现系统功能的逻辑结构（通常用逻辑图、HDL 文本或布尔表达式来表示），并对它进行逻辑模拟，验证其正确性。

接着进行电路设计。这一阶段的主要任务是将逻辑图中的各个逻辑部件细化到由一些基本门电路互连的结构，最后转变成由晶体管互连构成电子电路。电路设计中要考虑电路的速度和功耗，要注意所使用的元器件性能。

最后进行物理设计，包括版图设计与版图验证两方面的任务。版图设计是将电路的表示转化为几何表示（制造芯片所用的掩膜版图）。版图验证是保证集成电路版图设计正确性和可靠性的重要手段。版图的设计应符合与制造工艺有关的设计要求。版图验证的内容包括几何设计规则检查（DRC）、电学规则检查（ERC）、版图与电路原理图一致性检查（LVS）以及版图的电参数提取（LPE）及后模拟。

需要指出的是，在逻辑设计完成后要进行逻辑模拟，在电路设计完成后要进行电路模拟，在版图设计的全过程中及完成以后均需进行版图验证，以保证所设计的版图满足制造工艺要求和符合系统的设计规范。当不满足要求时，在后模拟与版图设计之间将会发生一个多次迭代的过程。逻辑模拟十分重要，因为任何逻辑设计上的错误如果一直到芯片做出来后才能发现（有时往往很难分清是否是逻辑设计上的错误），就要花费很多时间和费用去纠正，而且每纠正一次都需要重新进行电路设计、版图设计和工艺流水。