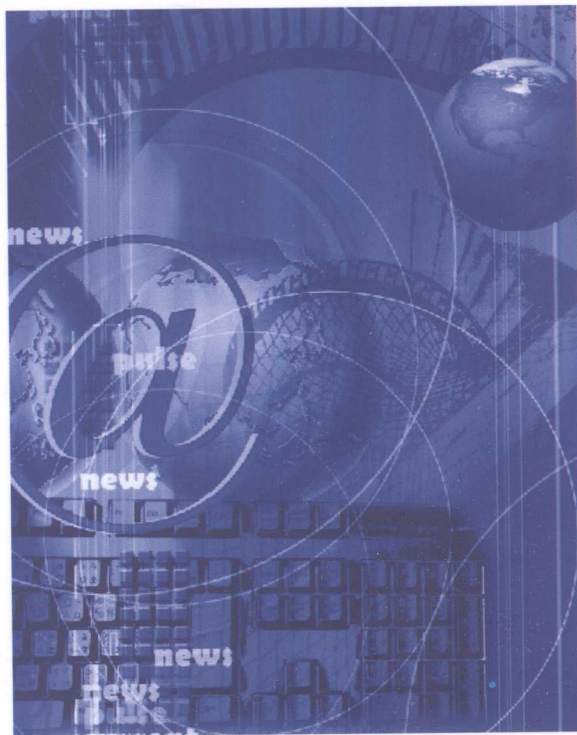


EDA技术与VHDL 程序开发教程

- ◆ EDA开发技术基础
- ◆ Quartus II集成开发环境
- ◆ VHDL语言基础
- ◆ VHDL语言的基本要素
- ◆ VHDL语言的顺序描述语句
- ◆ VHDL语言的并行描述语句
- ◆ 使用VHDL语言来设计组合逻辑电路
- ◆ 使用VHDL语言来设计时序电路
- ◆ 使用VHDL语言来设计存储器
- ◆ 使用VHDL语言来设计状态机
- ◆ VHDL语言的层次化设计
- ◆ VHDL语言的仿真、综合和优化
- ◆ VHDL语言的属性描述语句
- ◆ VHDL语言的数字系统设计应用实例



高等学校计算机应用规划教材

EDA 技术与 VHDL 程序开发教程

王千文 编著

清华大学出版社

TN702.2

11

内 容 简 介

EDA(Electronic Design Automation)工程是现代电子信息工程领域中发展迅速的一门新技术。它是以计算机为工作平台,以 EDA 软件工具为开发环境,以硬件描述语言为主要表达方式,以大规模可编程器件为设计载体,以 ASIC、SoC、FPGA 芯片为目标器件,以电子系统设计为应用方向的电子产品自动化设计过程。

本书介绍了 EDA 技术和硬件描述语言 VHDL 的基础知识,通过对工程实例的系统分析、程序设计和仿真,深入细致地讨论了其在数字系统设计中的广泛应用,内容包括 EDA 和 VHDL 语言基础、集成开发环境 Quartus II、VHDL 程序结构、VHDL 语言要素、VHDL 顺序语句和并行语句、基本电路的 VHDL 实现方法,还介绍了 VHDL 语言的综合、优化和层次设计并且给出了多个综合应用实例。

本书由浅入深地介绍了使用 VHDL 语言进行 EDA 设计的整个流程,并且给出了大量的应用实例,适合具有一定数字电路基础、希望学习集成电路设计的读者,可作为高等院校电子类相关专业的教材,也可作为 VHDL 语言设计的参考手册。

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 技术与 VHDL 程序开发教程 / 王千文 编著. —北京:清华大学出版社,2014

(高等学校计算机应用规划教材)

ISBN 978-7-302-35797-1

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

中国版本图书馆 CIP 数据核字(2014)第 060886 号



责任编辑:王 定

装帧设计:孔祥峰

责任校对:曹 阳

责任印制:沈 露

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 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

课 件 下 载: <http://www.tup.com.cn>, 010-62794504

印 刷 者:三河市君旺印务有限公司

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:23.25

字 数:537 千字

版 次:2014 年 9 月第 1 版

印 次:2014 年 9 月第 1 次印刷

印 数:1~3000

定 价:38.00 元

产品编号:044809-01

前言

行业背景

电子设计自动化(Electronic Design Automation, EDA)技术是指包括电路系统设计、系统仿真、设计综合、PCB 版图设计和制版的一整套自动化流程。随着计算机、集成电路和电子设计技术的高速发展,EDA 技术已经渗透百姓生活的各个角落,日益成为电子信息类产品的支柱产业;而硬件描述语言则是集成电路设计的核心技术,在常用的硬件描述语言中,VHDL 因其功能强大、严谨等特点占据了绝大部分市场。

关于本书

本书介绍了 EDA 技术和硬件描述语言 VHDL 的基础知识,通过对工程实例的系统分析、程序设计和仿真,深入细致地讲述了其在数字系统设计中的广泛应用。

本书首先介绍了 EDA 和 VHDL 语言基础知识,然后介绍了最常用的 VHDL 语言集成开发环境 Quartus II,接着基于 Quartus II 介绍了 VHDL 语言的语法知识,包括 VHDL 程序结构(如实体、结构体、块语句、库、程序包等)、VHDL 语言要素(如文字规则、数据变量、数据类型、操作符等)、VHDL 顺序语句和并行语句(如赋值语句、流程控制语句、子程序调用语句、进程语句、并行信号赋值语句等),还了基本电路(如组合逻辑里的加法器、译码器等,时序逻辑里的触发器、寄存器等,以及 Mealy 和 Moore 型状态机)的 VHDL 实现方法,最后介绍了 VHDL 语言的综合、优化和层次设计,并且给出了几个综合应用实例。

本书各个章节说明如下:

- 第 1 章主要介绍 EDA 技术和 VHDL 语言的基础知识,详细介绍了使用 VHDL 语言进行 EDA 设计的流程。
- 第 2 章主要介绍了 VHDL 语言的集成开发环境 Quartus II 的使用方法。
- 第 3 章主要介绍了 VHDL 语言程序的结构,包括描述风格、结构体、实体等。
- 第 4 章主要介绍了 VHDL 语言的基本要素,包括标识符的命名规范、数据对象的使用和数据类型等。
- 第 5 章主要介绍了 VHDL 语言的顺序描述语句,包括赋值语句、代入语句、流程控制语句、WAIT 语句等。
- 第 6 章主要介绍了 VHDL 语言的并行描述语句,包括并发信号代入语句、进程语句、过程语句等。
- 第 7 章主要介绍了如何使用 VHDL 语言设计组合逻辑电路,包括基础门电路、编

码器、译码器、数据选择器等。

- 第 8 章主要介绍了如何使用 VHDL 语言设计时序电路，包括触发器、寄存器、计数器等。
- 第 9 章主要介绍了如何使用 VHDL 语言设计存储器，包括 ROM、RAM、FIFO 等。
- 第 10 章主要介绍了如何使用 VHDL 语言设计状态机，包括米勒型状态机、莫尔型状态机以及状态机的实际应用。
- 第 11 章主要介绍了 VHDL 语言的层次化设计方法，包括库、程序包的使用等。
- 第 12 章主要介绍了如何对 VHDL 语言的代码进行仿真、综合和优化。
- 第 13 章主要介绍了 VHDL 语言的属性描述语句，包括数值类属性描述语句、函数类属性描述语句、信号类属性描述语句等。
- 第 14 章给出了四个 VHDL 语言的综合应用实例，包括交通灯控制器、出租车计费器、自动售货机和通用异步发送器。

本书特色

本书实例典型，内容丰富，有很强的针对性，不仅详细介绍了 VHDL 语言的基本用法，作为高层次综合语言的众多特点，还介绍了数字电路中最常见的组合逻辑电路、时序电路设计、存储器设计和状态机设计方法，并且给出了多个典型代表实例。

作者介绍

本书由王千文组织编写。其中，王千文主编第 1~7 章，程雅青主编第 8、9 章，王永皎主编第 10~12 章，马建红主编第 13、14 章，此外，参与本书编写的还有陈小亮、张国栋、张国华、李华、王林、李志国、陈晨、冯慧、徐红、吴文林、周建国、张建、刘海涛、张琴、高梅、吴晓、朱维、陈浩、汪梅、姚琳、何武和许小荣等同志，在此，编者对他们表示衷心的感谢。

由于时间仓促、程序和图表较多，受学识水平所限，书中不足之处在所难免，请广大读者给予批评指正。

编 者

2014 年 5 月

目

第 1 章 EDA 开发技术基础	1
1.1 EDA 技术基础	1
1.1.1 EDA 技术简介	1
1.1.2 EDA 技术的发展	2
1.2 EDA 技术涉及的内容	3
1.2.1 可编程逻辑器件	3
1.2.2 硬件描述语言	4
1.2.3 集成开发环境	5
1.2.4 实验开发系统	5
1.3 常用的硬件描述语言	6
1.3.1 AHDL 语言	6
1.3.2 VerilogHDL 语言	6
1.3.3 VHDL 语言	6
1.4 EDA 的集成开发环境	8
1.4.1 Quartus II 集成开发环境	8
1.4.2 ISE 和 ModelSim 集成开发环境	9
1.4.3 ispLEVER 集成开发环境	10
1.5 使用 EDA 技术进行数字系统开发	11
1.5.1 数字系统的设计模型	11
1.5.2 数字系统的设计方法	11
1.5.3 使用 VHDL 语言进行数字系统的开发流程	12
1.6 课后习题	15
1.6.1 填空题	15
1.6.2 简答题	15
第 2 章 Quartus II 集成开发环境	16
2.1 Quartus II 应用基础	16
2.1.1 使用图形用户界面	17
2.1.2 使用 EDA 工具设计界面	18

录

2.1.3 使用命令行可执行界面	20
2.1.4 Quartus II 集成开发环境的应用流程	21
2.1.5 Quartus II 的常用窗口	22
2.2 在 Quartus II 集成开发环境中进行 VHDL 语言开发	23
2.2.1 编辑设计文件	24
2.2.2 创建工程	25
2.2.3 设置工程属性	27
2.2.4 编译项目	28
2.2.5 仿真	29
2.2.6 观察 RTL 电路	33
2.2.7 其他	34
2.3 课后习题	34
2.3.1 填空题	34
2.3.2 简答题	35
2.3.3 实验题	35
第 3 章 VHDL 语言基础	37
3.1 VHDL 语言程序的结构	37
3.2 VHDL 语言的实体	39
3.2.1 实体说明	39
3.2.2 实体的类属说明	39
3.2.3 实体的端口说明	40
3.3 VHDL 语言的结构体	41
3.3.1 结构体的说明语句	42
3.3.2 结构体的信号定义	42
3.3.3 结构体的功能描述语句	42
3.4 VHDL 语言的描述风格	43
3.4.1 行为描述	44
3.4.2 数据流描述	45
3.4.3 结构描述	45

3.4.4 混合描述	47	4.7.4 VHDL 语言的逻辑 操作符	75
3.5 课后习题	48	4.7.5 VHDL 语言的重载 操作符	76
3.5.1 填空题	48	4.8 课后习题	76
3.5.2 简答题	48	4.8.1 填空题	76
3.5.3 实验题	49	4.8.2 简答题	77
第 4 章 VHDL 语言的基本要素	50	4.8.3 实验题	77
4.1 VHDL 语言文字的命名规范	50	第 5 章 VHDL 语言的顺序描述语句	79
4.1.1 数值型文字	50	5.1 赋值语句	79
4.1.2 字符串型文字	51	5.2 代入语句	80
4.2 VHDL 语言的标识符	51	5.3 赋值语句和代入语句进阶	81
4.2.1 短标识符	51	5.3.1 赋值语句和代入 语句的区别	81
4.2.2 扩展标识符	52	5.3.2 赋值语句和代入语句的 赋值对象	82
4.3 VHDL 语言的下标和 下标段名	52	5.4 流程控制语句	84
4.3.1 下标	52	5.4.1 IF 语句	84
4.3.2 下标段名	53	5.4.2 CASE 语句	86
4.4 VHDL 语言的注释	53	5.4.3 LOOP 语句	88
4.5 VHDL 语言的数据对象	54	5.4.4 NEXT 语句	89
4.5.1 常量	54	5.4.5 EXIT 语句	90
4.5.2 变量	55	5.5 WAIT 语句	91
4.5.3 信号	56	5.5.1 WAIT 语句的无限等待	91
4.5.4 文件	57	5.5.2 WAIT ON 语句	92
4.6 VHDL 语言的数据类型	58	5.5.3 WAIT UNTIL 语句	92
4.6.1 VHDL 语言的数据类型 按性质分类	58	5.5.4 WAIT FOR 语句	93
4.6.2 VHDL 语言的数据类型 按数据来源分类	58	5.5.5 多条件 WAIT 语句	94
4.6.3 VHDL 语言的数据 类型简介	59	5.6 空操作语句	94
4.6.4 VHDL 语言的数据 类型转换	67	5.7 报告语句	94
4.7 VHDL 语言的操作符	70	5.8 断言语句	95
4.7.1 VHDL 语言的操作符 种类和优先级	70	5.9 课后习题	96
4.7.2 VHDL 语言的算术操作符	72	5.9.1 填空题	96
4.7.3 VHDL 语言的关系操作符	74	5.9.2 简答题	96
		5.9.3 实验题	97

第 6 章 VHDL 语言的并行描述语句	98	7.1.2 逻辑函数的表示方法	126
6.1 并发信号代入语句	98	7.1.3 逻辑代数的定理和规则	127
6.1.1 并发信号代入语句基础	98	7.1.4 组合逻辑电路设计过程	129
6.1.2 条件信号代入语句	99	7.1.5 组合逻辑电路中的 竞争与冒险	130
6.1.3 选择信号代入语句	100	7.1.6 组合逻辑电路的设计 模式和方法	132
6.2 进程语句	101	7.2 组合逻辑的基本电路设计	132
6.3 元件调用语句	103	7.2.1 与门设计	133
6.4 端口映射语句	105	7.2.2 非门设计	137
6.4.1 端口位置映射	105	7.2.3 或门设计	139
6.4.2 端口名称映射	106	7.2.4 与非门设计	143
6.5 类属参数传递语句	106	7.2.5 异或门设计	146
6.6 块语句	107	7.2.6 三态门设计	149
6.6.1 普通块语句	107	7.3 译码器设计	153
6.6.2 卫式块语句	109	7.3.1 变量译码器设计	153
6.7 过程语句	110	7.3.2 码制变化译码器设计	156
6.7.1 子程序和包集合	110	7.3.3 显示译码器设计	161
6.7.2 过程语句的定义与使用	111	7.4 编码器设计	165
6.7.3 并发过程调用语句	113	7.4.1 编码器基础	165
6.8 函数语句	114	7.4.2 74 系列优先编码器	165
6.9 返回语句	115	7.4.3 VHDL 描述的优先 编码器	165
6.10 生成语句	116	7.5 数据选择器设计	167
6.10.1 FOR-GENERATE 结构的生成语句	116	7.5.1 数据选择器基础	167
6.10.2 IF-GENERATE 结构的 生成语句	117	7.5.2 74 系列数据选择器	168
6.11 延时语句	119	7.5.3 数据选择器的 VHDL 语言实现	169
6.11.1 惯性延时语句	119	7.6 比较器设计	171
6.11.2 传输延时语句	120	7.6.1 比较器基础	171
6.12 课后习题	120	7.6.2 74 系列比较器	172
6.12.1 填空题	120	7.6.3 VHDL 描述的比较器	173
6.12.2 简答题	121	7.7 加法器设计	175
6.12.3 实验题	121	7.7.1 加法器基础	175
第 7 章 使用 VHDL 语言设计		7.7.2 74 系列加法器	176
组合逻辑电路	123	7.7.3 加法器的 VHDL 语言实现	177
7.1 组合逻辑电路设计基础	123		
7.1.1 逻辑代数基础	123		

7.8	减法器设计	180
7.8.1	减法器基础	180
7.8.2	减法器的 VHDL 语言实现	181
7.9	乘法器设计	182
7.9.1	乘法器的工作原理	182
7.9.2	乘法器的 VHDL 语言实现	183
7.10	奇偶检测电路设计	184
7.10.1	奇偶检测电路基础	184
7.10.2	74 系列奇偶检测电路	185
7.10.3	VHDL 描述的奇偶检测电路	185
7.11	其他逻辑电路设计	187
7.11.1	双向总线设计	187
7.11.2	补码器	189
7.12	课后习题	190
7.12.1	填空题	190
7.12.2	简答题	190
7.12.3	实验题	190
第 8 章	使用 VHDL 语言设计时序电路	192
8.1	时序电路基础	192
8.2	时序电路的时钟信号	193
8.2.1	VHDL 语言的时钟边沿描述	193
8.2.2	使用时钟作为敏感信号	194
8.3	时序电路的复位信号	196
8.3.1	时序电路的同步复位方法	196
8.3.2	时序电路的异步复位方法	197
8.4	基础时序电路(触发器)设计	198
8.4.1	RS 触发器设计	199
8.4.2	JK 触发器设计	205
8.4.3	D 触发器设计	209

8.4.4	T 触发器设计	212
8.5	寄存器设计	214
8.5.1	寄存器基础	214
8.5.2	74 系列寄存器	214
8.5.3	VHDL 语言描述的寄存器	214
8.6	移位寄存器设计	214
8.6.1	移位寄存器基础	214
8.6.2	74 系列移位寄存器	215
8.6.3	VHDL 语言描述的移位寄存器	216
8.7	计数器设计	221
8.7.1	计数器基础	221
8.7.2	74 系列计数器	221
8.7.3	VHDL 语言描述的计数器	222
8.8	课后习题	225
8.8.1	填空题	225
8.8.2	简答题	226
8.8.3	实验题	226
第 9 章	使用 VHDL 语言设计存储器	228
9.1	只读存储器设计	228
9.1.1	只读存储器基础	228
9.1.2	只读存储器的 VHDL 语言实现	228
9.2	随机存储器设计	232
9.2.1	随机存储器基础	232
9.2.2	随机存储器的 VHDL 语言实现	233
9.3	先进先出存储器设计	236
9.3.1	先进先出存储器基础	236
9.3.2	先进先出存储器的 VHDL 语言实现	236
9.4	课后习题	238
9.4.1	填空题	238

9.4.2 简答题	238	11.5 VHDL 语言的配置	275
9.4.3 实验题	238	11.5.1 配置的标准调用格式 和结构	276
第 10 章 使用 VHDL 语言 设计状态机	239	11.5.2 结构体中的元件配置	279
10.1 状态机基础	239	11.5.3 端口映射和实体 映射的配置	280
10.1.1 状态机的分类	239	11.5.4 块的配置	280
10.1.2 状态机的 VHDL 语言 描述方法	240	11.5.5 使用配置来传递类 属性	280
10.1.3 状态机的状态编码	241	11.6 课后习题	282
10.1.4 状态机的非法状态	242	11.6.1 填空题	282
10.2 状态机的 VHDL 语言实现	243	11.6.2 简答题	282
10.2.1 带变量输出的状态机	243	11.6.3 实验题	282
10.2.2 带同步复位的状态机	245	第 12 章 VHDL 语言的仿真、 综合和优化	283
10.2.3 米勒型状态机	247	12.1 VHDL 语言的仿真	283
10.2.4 莫尔型状态机	249	12.1.1 使用 VHDL 语言生成 仿真输入信号	284
10.2.5 带莫尔/米勒型输出的 状态机	251	12.1.2 VHDL 语言仿真程序 的规范	289
10.3 状态机的实际应用	253	12.1.3 VHDL 语言仿真的 δ 延时	292
10.3.1 状态机应用实例 ——序列信号发生器	253	12.2 VHDL 语言的综合	293
10.3.2 状态机应用实例 ——循环彩灯控制器	255	12.2.1 VHDL 语言综合的 分类	293
10.4 课后习题	258	12.2.2 VHDL 语言的综合 约束条件	294
10.4.1 填空题	258	12.2.3 VHDL 语言的综合 工艺库	295
10.4.2 简答题	258	12.2.4 门级映射	296
10.4.3 实验题	258	12.3 VHDL 语言的设计优化	296
第 11 章 VHDL 语言的层次化设计	260	12.3.1 VHDL 语言的设计 优化基础	296
11.1 层次化结构设计基础	260	12.3.2 VHDL 语言的模块 复用	297
11.2 自顶向下的设计方法	262		
11.3 VHDL 语言的库	263		
11.3.1 库的声明	263		
11.3.2 VHDL 语言的常用库	264		
11.4 VHDL 语言的程序包	265		
11.4.1 程序包的定义和使用	266		
11.4.2 VHDL 语言的常用 程序包	267		

12.3.3	VHDL 语言的流水线设计	300
12.4	课后习题	301
12.4.1	填空题	301
12.4.2	简答题	302
12.4.3	实验题	302
第 13 章 VHDL 语言的属性		
	描述语句	304
13.1	属性预定义	304
13.2	数值类属性描述语句	304
13.2.1	一般数据的数值属性描述语句	305
13.2.2	数组的数值属性描述语句	306
13.2.3	块的数值属性描述语句	307
13.3	函数类属性描述语句	309
13.3.1	数据类型的属性函数	309
13.3.2	数组的属性函数	312
13.3.3	信号的属性函数	313
13.4	信号类属性描述语句	317
13.4.1	属性'DELAYED 语句	317
13.4.2	属性'STABLE 语句	319
13.4.3	属性'QUIET 语句	320
13.4.4	属性'TRANSACTION 语句	322
13.5	数据类型类属性描述语句	323
13.6	数据区间类属性描述语句	324
13.7	用户自定义属性语句	325
13.8	课后习题	327
13.8.1	填空题	327
13.8.2	简答题	327
13.8.3	实验题	327

第 14 章 VHDL 语言的数字系统	
	设计应用实例
14.1	交通灯控制器
14.1.1	交通灯控制器的需求分析
14.1.2	交通灯控制器的系统设计
14.1.3	交通灯控制器的 VHDL 语言实现
14.1.4	交通灯控制器的仿真
14.2	出租车计费器
14.2.1	出租车计费器的需求分析
14.2.2	出租车计费器的系统设计
14.2.3	出租车计费器的 VHDL 语言实现
14.2.4	出租车计费器的仿真
14.3	自动售货机
14.3.1	自动售货机的需求分析
14.3.2	自动售货机的系统设计
14.3.3	自动售货机的 VHDL 语言实现
14.3.4	自动售货机的仿真
14.4	通用异步发送器
14.4.1	通用异步发送器的需求分析
14.4.2	通用异步发送器的系统设计
14.4.3	通用异步发送器的 VHDL 语言实现
14.4.4	通用异步发送器的仿真

第1章 EDA开发技术基础

在 EDA 开发中,用户以计算机为工具,在对应的集成开发环境中,用硬件描述语言完成设计文件,然后由计算机自动地完成逻辑编译、综合、优化、布局、布线和仿真等工作,并且对于特定目标芯片进行适配编译、逻辑映射和编程下载。本章将介绍 EDA 开发技术的基础知识,包括 EDA 的发展历史、可编程逻辑语言的分类和特点、可编程逻辑器件的种类和开发方法,以及使用 EDA 进行设计的流程等。

1.1 EDA 技术基础

EDA 是电子设计自动化(Electronic Design Automation)的缩写,是在 20 世纪 60 年代中期从计算机辅助设计(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。

1.1.1 EDA 技术简介

EDA 技术是现代电子信息工程领域中一门发展迅速的新技术,是以计算机为工作平台,以 EDA 软件工具为开发环境,以硬件描述语言为主要表达方式,以大规模可编程器件为设计载体,以 ASIC、SoC、FPGA 芯片为目标器件,以电子系统设计为应用方向的电子产品自动化设计过程。

EDA 技术有广义和狭义之分。

从广义来说,EDA 技术包括半导体工艺设计自动化、可编程器件设计自动化、电子系统设计自动化、印制电路板设计自动化、仿真与测试故障诊断自动化等,如 PSPICE、EWB、MATLAB 等计算机辅助分析(CAA)技术和 Protel、OrCAD 等印刷制版计算机辅助设计等。

从狭义来说,EDA 技术就是电子设计自动化,即通过相关的开发软件,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、化简、分割、综合、优化,以及布局布线、逻辑仿真等工作,最终完成对于特定目标芯片的适配编译、逻辑映射、编程下载,从而形成集成电子系统,不包含电子生产自动化。

EDA 技术在电子系统设计中具有以下特点,所以得到了广泛应用:

- (1) 用软件的方式设计硬件。
- (2) 用软件方式设计的系统到硬件系统的转换是由相关软件自动完成的。
- (3) 在设计过程中可以用软件进行各种仿真以便进行验证。
- (4) 现代 EDA 工具具有高层综合和优化功能,能够在系统级进行综合、优化和仿真,从而缩短设计周期,提高工作效率。

- (5) 系统可以现场编程, 在线升级。
- (6) 整个系统易集成、体积小、功耗低且可靠性高。
- (7) 带有嵌入 IP 核的 ASIC 设计, 提供软硬件协同设计。
- (8) 提供开放和标准化的操作环境, 容易实现资源共享和设计移植。
- (9) 支持并行设计, 适合团队协作, 分工设计。

1.1.2 EDA 技术的发展

早在 20 世纪 60 年代中期, 人们就开始着眼于开发出各种计算机辅助设计工具来帮助设计人员进行集成电路和电子系统的设计, 集成电路技术的发展不断地对 EDA 技术提出新的要求, 并促进了 EDA 技术的发展。在过去的三十多年间, 计算机技术迅猛发展, 也给 EDA 行业带来了巨大的变化, 进入 20 世纪 90 年代后, 电子系统已经从电路板级系统集成发展成为包括 ASIC、FPGA 和嵌入式系统的多种模式, EDA 产业已经成为电子信息类产品的支柱产业。EDA 的蓬勃发展离不开设计方法学的进步, 回顾过去几十年电子技术的发展历程, 可大致将 EDA 技术的发展分为以下几个阶段:

(1) 20 世纪 70 年代, 是 EDA 技术发展初期, 我们称之为计算机辅助设计(Computer Aided Design, CAD)阶段。随着集成电路的出现和应用, 硬件设计开始大量选用中小规模的标准集成电路, 这也使得传统的手工布线方法很难满足产品复杂性和工作效率的要求。CAD 的概念已见雏形, 人们开始利用计算机替代产品设计过程中的高度重复性的复杂劳动, 如利用二维图形编辑与分析工具, 辅助进行集成电路版图编辑、PCB 布局布线等工作。最具代表性的产品当属美国 ACCEL 公司的 Tango 布线软件。

(2) 20 世纪 80 年代, 随着集成电路设计进入 CMOS 时代, EDA 技术也进入到了计算机辅助工程(Computer Assist Engineering, CAE)阶段。PAL、GAL 和 FPGA 等一系列复杂可编程逻辑器件都为电子系统的设计提供新的平台。较之 20 世纪 70 年代的自动布局布线的 CAD 工具能够替代设计中绘图的重复劳动而言, 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, 它在 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、布尔方程综合和简化等方面都担任了重要角色。

(3) 20 世纪 90 年代, 以在设计前期将设计师从事的许多高层次设计交由工具来完成为目的, EDA 技术开始从以单个电子产品开发为对象转向针对系统级电子产品的设计。EDA 工具以系统级设计为核心, 包括了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及系统决策与文件生成等一系列完整的功能。随着硬件描述语言标准的进一步确立, 此时的 EDA 工具还具有高级抽象的设计构思手段, 各 EDA 公司也致力于推出兼容各种硬件方案和支持标准硬件描述语言的 EDA 软件的研究。

(4) 21 世纪以来, EDA 技术得到了更大的发展。高速 DSP、嵌入式处理器软核的成熟令 EDA 软件的功能日益强大。电子领域各学科全方位融入 EDA 技术, 除了成熟的数字技术外, 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等, 使得 EDA 工具不论在广度上还是深度上

都取得了长足的发展,现在已经出现了大量的可以直接设计处理器的技术,如 Altera 公司的 SoC FPGA 技术,这些技术可以更好地应用于各种嵌入式系统中。

1.2 EDA 技术涉及的内容

作为一门发展迅速、有着广阔应用前景的新技术,EDA 技术涉及面广,内容丰富,其主要涉及如下四个方面的内容:

- (1) 可编程逻辑器件(PLD):也就是应用 EDA 设计完成的电子系统的载体,包括 CPLD、FPGA 等。
- (2) 硬件描述语言(HDL):用于描述系统的结构和功能,是 EDA 设计的主要表达手段,包括 VHDL、AHDL、VerilogHDL 等。
- (3) 集成开发环境:是 EDA 设计的开发环境,常见的有 Quartus、Foundation Series 等。
- (4) 实验开发系统:实现可编程器件下载和验证的工具,包括编程器、开发板等。

1.2.1 可编程逻辑器件

可编程逻辑器件(Programmable Logic Device, PLD)是一种通过用户编程来实现某种逻辑功能的新型逻辑器件。经过近三十年的发展,可编程逻辑器件已经从最初简单的 PLA、PAL、GAL 发展到目前应用最为广泛的 CPLD(Complex Programmable Logic Device, 复杂的可编程逻辑器件)、FPGA(Field Programmable Gate Array, 现场可编程门阵列)。

FPGA 是一种基于查找表(Look Up Table, LUT)的可编程逻辑器件,在结构上主要分为可编程逻辑单元、可编程输入/输出单元和可编程连线三个部分。FPGA 内部阵列块之间采用分段式进行互连,结构比较灵活,但是延时不可预测;比较适合于触发器多的逻辑相对简单的数据型系统。FPGA 保存逻辑功能的物理结构多为 SRAM 型,即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 ROM,将设计好的逻辑信息烧录到此配置芯片中。系统上电时, FPGA 就能自动从配置芯片中读取逻辑信息。

CPLD 是一种基于乘积项的可编程逻辑器件,主要由可编程逻辑宏单元、可编程输入/输出单元和可编程内部连线组成。CPLD 内部采用固定长度的线进行各逻辑块的互连,因此,与 FPGA 相比,引脚到引脚的延迟时间几乎是固定的,与逻辑设计无关,这使得设计调试比较简单,逻辑设计中的毛刺现象比较容易处理,性价比较高。CPLD 具有很宽的输入结构,较适合逻辑复杂、输入变量多、对触发器的需求量相对较少的逻辑型系统。另外, CPLD 结构大多为 EEPROM 或 Flash ROM 形式,具有编程后即可固定下载的逻辑功能,掉电不丢失。

相比 PLA、GAL 而言,高集成度、高速度和高可靠性是 CPLD、FPGA 最显著的特点。高集成度方面, CPLD、FPGA 几乎可以将整个系统下载于同一芯片中,实现所谓片上系统,大大缩小了产品的体积,使之易于管理和屏蔽。在高可靠性领域,只要设计得当, CPLD、FPGA 完全不存在类似于 MCU 的复位不可靠和 PC 可能跑飞的问题。CPLD、FPGA 还可

将时钟延时缩短至毫秒级,大大提升了产品的系统性能。由于开发工具的通用性、设计语言的标准化以及设计过程与目标芯片的硬件结构的独立性,设计成功的各种逻辑功能块软件可以很好地兼容和移植,从而使得产品设计效率得到了大幅提高。

目前,国际上生产 CPLD、FPGA 并在我国占有较大市场份额的公司主要是 Xilinx、Lattice 和 Altera 三家。Xilinx 的 FPGA 产品有 XC2000、XC3000、XC4000、XC5000、EXC4000XLA 和 XC5200 系列等。Lattice 公司的 ISP-PLD 器件有 ispLSI1000、ispLSI2000、ispLSI3000 和 ispLSI6000 系列等。Altera 公司的 CPLD 器件现在主要是 MAX 系列,包括 MAX7000S、MAX3000A、MAXIIIZ、MAX II 和 MAX V 五大子系列,具有成本低、创新的查找表技术、非易失性的特点,支持 1.8~5V 的多种工作电压,最多可以达到 1700 个典型等价宏单元。Altera 公司还提供了 Cyclone、Arria、Stratix 等系列的 FPGA 芯片,其内建了高性能硬件宏功能,如 PLL、存储器模块、DSP 模块等,能更好地为复杂高速的控制系统或者数据处理系统服务。此外,Altera 公司还提供了基于 ARM 体系的 SoC FPGA 芯片,在 FPGA 架构中集成了基于 ARM 的硬核处理器系统,包括 CPU、外设和存储器接口,实现了硬件的性能和低功耗性以及可编程逻辑的灵活性。

1.2.2 硬件描述语言

硬件描述语言(Hardware Description Language, HDL)是一种对于数字电路和系统进行性能描述和模拟的语言,即利用高级语言来描述硬件电路的功能、信号连接关系和各器件间的时序关系。其设计理念是将硬件设计软件化,即采用软件的方式来描述硬件电路。数字电路和数字系统设计者利用这种语言来描述自己的设计思想,然后利用电子设计自动化工具进行仿真、综合,最后利用专用集成电路或可编程逻辑器件来实现其设计功能。

传统的数字电路设计方法中,要完成整个产品设计之后才能进行产品的测试和设计的验证,因此一旦设计出现问题,需要在设计流程的最后才能发现并进行修改,直到设计的产品完全符合设计要求。而采用硬件描述语言,可以在数字系统的设计阶段对系统性能进行描述和模拟仿真,以便在设计初期发现和修改绝大多数错误,缩短了硬件开发的时间,减少了硬件设计的成本,因此这种被称为高层次设计的方法目前已经得到了广泛的应用。

硬件描述语言是 EDA 技术的重要组成部分,常见的 HDL 主要有 VHDL、Verilog HDL、ABEL、AHDL、System Verilog 和 SystemC。学术界自 20 世纪 70 年代已经开始使用 HDL,发展至今已有四十多年的历史。但是最初由于各个 EDA 公司均开发支持自己公司产品的硬件描述语言,导致硬件描述语言品种很多,互相之间不能通用,语言本身的性能也不够完善,影响了这种设计工具的推广。直到 20 世纪 80 年代,标准化的硬件描述语言开始被研究和应用,VHDL 和 Verilog HDL 两种硬件描述语言先后成为 IEEE 的标准,采用硬件语言描述的设计方法才得到了广泛的应用。

目前,VHDL 和 Verilog HDL 几乎得到了所有主流 EDA 工具的支持。VHDL 作为 IEEE 的工业标准硬件描述语言,在电子工程领域,已成为事实上的通用硬件描述语言。Verilog HDL 适用于 RTL 级和门电路级的描述,其综合过程较 VHDL 稍简单,但其在高级语言描述方面不及 VHDL。

与传统的原理图设计方法相比较,硬件描述语言能更有效地表示硬件电路的特性,具有更多优点。

(1) 用硬件描述语言设计电路能够获得非常抽象的描述,设计与具体的硬件电路无关。对设计者而言,在高抽象层次进行系统设计,可以将精力主要集中在系统级问题上,而不必关心低层次结构设计或制造工艺等的细节问题,这样可以节省开发时间和成本。例如,设计者不用选择通用数字芯片就能写出电路的算法级描述,不用选择特定的制造工艺就能写出电路的寄存器传输级(Register Transfer Level)描述,逻辑综合工具能自动将算法级描述转换成 RTL 描述或将 RTL 描述转换成任何一种制造工艺。如果出现了新器件、新工艺,也不用重新设计电路,只需将电路的算法级描述或 RTL 描述输入到逻辑综合工具中,就能产生新的门级网表。逻辑综合工具将针对新的器件、新工艺自动进行结构综合、电路面积和时序的优化等。

(2) 用硬件描述语言描述电路设计,在设计的前期就可以利用仿真工具完成电路功能级的验证和基于某种指定的可编程逻辑器件的时序验证。由于设计者工作在 RTL 级,可以不断修改和优化 RTL 描述,直到设计的系统能够满足所需要的功能和指标,这样能够发现并改进设计中绝大部分的错误,在设计后期的门级网表及物理版图中出现功能性错误的概率非常小,可以非常有效地缩短产品的开发周期。

(3) 用硬件描述语言设计电路,类似于计算机编程。辅以注释的硬件描述语言程序,可以非常方便地进行数字电路和数字系统的开发和调试。

(4) 用硬件描述语言设计电路,使电路具有很好的易读性,便于理解。在大型的复杂电路系统设计中,采用门级原理图进行的设计几乎是不可理解的。

功能强大的逻辑综合工具把硬件描述语言推到了数字电路设计的最前沿,设计者不再需要手工放置通用数字芯片的办法来“搭建”数字电路。由于使用 HDL,设计者能够将更多精力投入到系统上,更好地从功能、行为和算法上表述自己的设计,并加上详细的注解,便于设计的移植和再开发。

1.2.3 集成开发环境

EDA 技术的综合应用设计离不开 EDA 软件的支持。目前比较流行的、主流厂家的 EDA 软件工具有 Altera 公司的 MAX+PLUS II、Quartus II, Lattice 公司的 ispDesignEXPERT, Xilinx 公司的 Foundation Series、ISE/ISE-WebPACK Series。业界最流行的第三方 EDA 工具中,逻辑综合性能最好的是 Synplify, 仿真功能最强大的是 ModelSim。

在 1.4 小节中将比较详细地对目前常用的 EDA 集成开发环境进行介绍。

1.2.4 实验开发系统

除了 EDA 开发工具软件,EDA 实验中还需要用到其他一些外围资源,以共同完成完整的 EDA 设计开发。

如计算机辅助分析软件 Matlab,用于复杂系统的系统及仿真、DSP 的系统建模与有关数据的计算和分析等;印刷电路板(PCB)设计软件 Protel、OrCAD 等,用于设计系统的外

围部件或外围部件的 PCB 图。

又如开发所需的各类基本信号发生模块, 如时钟模块、脉冲和高电平零; CPLD/FPGA 输出信息显示模块, 包括数码显示、发光管显示和声响显示等。

此外, 不可或缺的还有目标芯片的适配座以及上面的 CPLD/FPGA 目标芯片和编程下载电路等。

1.3 常用的硬件描述语言

硬件描述语言(HDL)是EDA技术的核心内容, 在实际应用中最常见的HDL语言有AHDL、VerilogHDL和VHDL三种。

1.3.1 AHDL 语言

AHDL(Altera HDL)语言是 Altera 公司发明的 HDL 语言, 其优点是易学易用, 学过高级语言的人可以在很短的时间内掌握; 其缺点是移植性不好, 通常只用于 Altera 公司自己的开发系统。

1.3.2 VerilogHDL 语言

VerilogHDL 语言是在 C 语言的基础上发展起来的一种硬件描述语言, 它是由 GDA(Gateway Design Automation)公司的 Phil Moorby 在 1983 年末首创的, 最初只设计了一个仿真与验证工具, 之后又陆续开发了相关的故障模拟与时序分析工具。1985 年, Moorby 推出它的第三个商用仿真器 Verilog-XL, 获得了巨大的成功, 从而使得 Verilog HDL 迅速得到推广应用。1989 年, Cadence 公司收购了 GDA 公司, 使得 Verilog HDL 成为了该公司的独家专利。1990 年, Cadence 公司公开发布了 Verilog HDL, 并成立 LVI 组织以促进 Verilog HDL 成为 IEEE 标准, 即 IEEE 1364—1995。

Verilog HDL 的最大特点是和 C 语言有很多共通的地方, 包括语法、书写习惯和风格等, 如果用户有 C 语言的编程经验, 可以在一个较短的时间内很快地学习和掌握它。与之相比, VHDL 的学习要困难一些, 但是 Verilog HDL 较自由的语法, 也容易对不熟悉的用户带来困扰。

1.3.3 VHDL 语言

VHDL 是 Very High Speed Integrated Circuit HDL(超高速集成电路硬件描述语言)的缩写, 是在 ADA 语言的基础上发展起来的硬件描述语言。VHDL 是美国国防部在 20 世纪 70 年代末提出的 VHSIC(Very High Speed Integrated Circuit, 超高速集成电路)计划的产物, VHSIC 计划的目标是使工业界可以开发比以前更为复杂的集成电路, 加速美国微电子行业的发展。但是由于美国国防部电子系统项目由众多公司承包, 各公司使用自己的语言, 使得设计的可移植性和再开发性很差, 信息交换和设计维护很困难。因此美国国防部委托