

# VHDL

## 实用教程

潘松 王国栋 编著  
李广军 审校

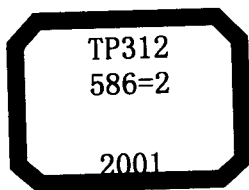
EDA 技术丛书



电子科技大学出版社



WOLFGANG



EDA 技术丛书

# VHDL 实用教程

(修订版)

潘 松 王国栋 编著

李广军 审校

电子科技大学出版社

## 内 容 简 介

本书比较系统地介绍了 VHDL 的基本语言现象和实用技术。全书以实用和可操作为基点, 简洁而又不失完整地介绍了 VHDL 基于 EDA 技术的理论与实践方面的知识。

其中包括 VHDL 语句语法基础知识(第 1 章至第 7 章)、仿真(第 8 章)、逻辑综合与编程技术(第 9 章)、有限状态机及其设计(第 10 章)、基于 FPGA 的数字滤波器设计(第 11 章)、多种常用的支持 VHDL 的 EDA 软件使用方法(第 12 章)、VHDL 数字系统设计实践(第 13 章)和大学生电子设计竞赛题的 VHDL 应用介绍(第 14 章)。全书列举了大量 VHDL 设计示例, 其中大部分经第 12 章介绍的 VHDL 综合器编译通过, 第 13 章的程序绝大部分都通过了附录介绍的 EDA 实验系统上的硬件测试, 可直接使用。书中还附有大量程序设计和实验/实践方面的习题。

本书可作为高等院校的电子工程、通信、工业自动化、计算机应用技术、电子对抗、仪器仪表、数字信号处理、图像处理等学科领域和专业的高年级本科生或研究生的 VHDL 或 EDA 技术课程的教材及实验指导, 也可作为相关专业技术人员自学参考书。

# VHDL实用教程

(修订版)

潘 松 王国栋 编著  
李广军 审校

---

出 版: 电子科技大学出版社(成都建设北路二段四号, 邮编: 610054)

责任编辑: 张 琴

发 行: 新华书店

印 刷: 成都蜀通印务有限责任公司

开 本: 787×1092 1/16 印张 23.75 字数 575 千字

版 次: 2001 年 7 月第二版

印 次: 2004 年 1 月第五次印刷

书 号: ISBN 7-81065-290-7/TP·172

印 数: 12001—15000 册

定 价: 25.00 元

---

# 前 言

随着 VLSI 的发展,电子产品市场运作节奏的进一步加快,涉及诸多领域(如计算机应用、通信、智能仪表、医用设备、军事、民用电器等领域)的现代电子设计技术已迈入一个全新的阶段,其特点是:(1)电子器件及其技术的发展将更多地趋向于支持 EDA (Electronic Design Automation);(2)借助于硬件描述语言,硬件设计与软件设计技术得到了有机的融合;(3)就电子设计的技术、开发过程和目标器件的形式与结构来说,将从过去的“群雄并起”的局面向规范化、标准化发展;(4)应用系统的设计已从单纯的 ASIC 设计走向了系统设计和单片系统 SOC (System On a Chip) 设计。一些专家预言,未来的电子技术将是 EDA 的时代。为了适应这一时代,国外各大 VLSI 厂商纷纷推出各种系列的大规模和超大规模 FPGA 和 CPLD 产品。其产品性能提高之快,品种之多让人应接不暇。Xilinx 公司和 Altera 公司推出了多种高性能的 FPGA/CPLD 芯片,FPGA 器件的规模已进入了千万门的行列。作为世界最早发明 GAL 等可编程逻辑器件的 Lattice 公司,在原来已成熟的 PLD/CPLD 产品的基础上,再次首创了可编程逻辑器件革命性的编程下载方式,即在系统可编程下载方式,并相继推出了多种系列各具特色的 ISP 下载方式的 CPLD 器件,以及大规模在系统可编程模拟器件。在最近几年中,可编程逻辑器件的开发生产和销售规模以惊人的速度增长,统计资料表明,其平均年增长率高达 23%。与此相适应,Cadence、Data I/O、Mentor Graphics、OrCAD、Synopsys 和 Viewlogic 等世界各大 EDA 公司亦相继推出各类高性能的 EDA 工具软件。在现代电子设计技术高速推进浪潮的多层因素促进下,CAD、CAM、CAT 和 CAE 技术发生了进一步融合与升华,形成了更为强大的 EDA 和 ESDA (Electronic System Design Automation) 技术,从而成为当代电子设计技术发展的总趋势。

面对现代电子技术的迅猛发展,高新技术日新月异的变化以及人才市场、产品市场的迫切需求,我国许多高校迅速地作出了积极的反应,在不长的时间内,在相关的专业教学与学科领域卓有成效地完成了具有重要意义的教学改革及学科建设。例如,适用于各种教学层次的 EDA 实验室的建立;EDA、VHDL 及大规模可编程逻辑器件相关课程的设置;两年一度的全国大学生电子设计竞赛也已使用了 FPGA、CPLD 及相应的 EDA 开发系统;同时对革新传统的数字电路课程的教学内容和实验方式作了许多大胆的尝试,从而使得诸如电子信息、通信工程、计算机应用、工业自动化等专业的毕业生的实际电子工程设计能力、新技术应用能力以及高新技术市场的适应能力都有了明显的提高。

VHDL 作为 IEEE 标准的硬件描述语言和 EDA 的重要组成部分,经过十几年的发展、应用和完善,以其强大的系统描述能力、规范的程序设计结构、灵活的语句表达风格和多层次的仿真测试手段,受到业界的普遍认同和广泛的接受,从数十种国际流行的硬件描述语言中脱颖而出,成为现代 EDA 领域的首选硬件设计计算机语言,而且目前流行的 EDA 工具软件全部支持 VHDL。除了作为电子系统设计的主选硬件描述语言外,VHDL 在 EDA 领域的仿真测试、学术交流、电子设计的存档、程序模块的移植、ASIC 设计源程序的交

付、IP 核 (Intelligence Property Core) 的应用等方面担任着不可或缺的角色, 因此不可避免地将成为了我国高等教育中电子信息类专业知识结构的重要组成部分。

在新世纪中, 电子技术的发展将更加迅猛, 电子设计的自动化程度将更高, 电子产品的上市节奏将更快, 传统的电子设计技术、工具和器件将在更大的程度上被 EDA 所取代, EDA 技术和 VHDL 势必成为广大电子信息工程类专业领域工程技术人员的必修课。

与一般的高级语言相比, VHDL 的学习具有更强的实践性, 它的学习和应用所涉及的内容和工具比较多, 类似传统软件编程语言的语法语句和编程练习的学习已不足以掌握 VHDL。有鉴于此, 本教程从实际的应用出发, 以实用和可操作为基点, 以初步掌握 EDA 技术和培养基于 VHDL 的开发能力为目标, 始终围绕一个主题: 学以致用。

实用性是本教程的特点, 主要表现在: (1) 适当略去有关行为仿真语句的内容, 主要考虑到这些内容不能参与综合和在硬件电路中实现。而在实用中, VHDL 的仿真大都采用功能仿真、时序仿真和硬件仿真。(2) 以可综合的 VHDL 程序设计介绍为基点, 将软件程序与对应的硬件电路结构紧密联系在一起, 力图提高读者实现既定硬件电路的 VHDL 软件设计驾驭能力, 在教程中尽可能给出对应程序的综合后的电路原理图。(3) 全书从不同角度介绍了 VHDL 的最直接的实用技术。本教程的另一特点是可操作性: (1) 教程中的程序几乎全部经 VHDL 综合器通过, 且大部分经硬件测试, 并可直接在实验或电子设计中使用, 特别是第 11 章至第 14 章中的实践与实验项目, 许多是我校毕业生的课余作品和毕业设计项目, 这些同学有蔡邦忠、金荣伟、傅剑斌、姜寒冰和李永成等。(2) 在第 12 章安排了 3 种目前最流行的基于 PC 的 VHDL 设计 EDA 软件的使用介绍, 而且采用的是“向导”式介绍方法, 即从一个 VHDL 设计实例开始, 通过各个处理项目, 从编辑、编译、仿真、布局布线和适配, 直至配置/下载和硬件测试, 向读者完整地展示了该软件的各项主要功能使用的全过程, 比较适合于 EDA 工具使用的速成式自学。(3) 附录 1 和附录 2 为 VHDL 学习的最后一个阶段, 为 VHDL 的实验和硬件仿真/调试提供了有用的信息。

我们期望本教程能有助于读者在 EDA 的教学与实验方面、在学生的电子设计和电子工程实践能力的提高方面、在高新技术领域的产品开发与推广以及相关学科领域的技术拓展方面收到良好的效果。

不可否认, 本书的出发点是一回事, 实际编写水平可能又是另外一回事。良好的愿望未必就是现实。我们真诚地欢迎, 并期待读者能对书中的错误给予指正, 让我们共同进步!

与作者的联系方式如下:

E-mail: span88@mail.hz.zj.cn

电话: 0571-5525171 / 5972935

地址: 杭州文一路 65 号, 杭州电子工业学院电子信息学院 (310012)

责任编辑联系方式:

E-mail: Hjwang@uestc.edu.cn, 电话: (028) 3203189, 3251067-8999

地址: 成都电子科技大学出版社 (610054)

编者

2001 年 3 月于杭州电子工业学院

## 目 录

<b>第 1 章 绪论</b> .....	1
§ 1.1 关于 EDA.....	1
§ 1.2 关于 VHDL.....	3
§ 1.3 关于自顶向下的系统设计方法.....	5
§ 1.4 关于应用 VHDL 的 EDA 过程.....	6
§ 1.5 关于在系统编程技术.....	9
§ 1.6 关于 FPGA/CPLD 的优势.....	10
§ 1.7 关于 VHDL 的学习.....	10
<b>第 2 章 VHDL 入门</b> .....	12
§ 2.1 用 VHDL 设计多路选择器和锁存器.....	12
§ 2.2 用 VHDL 设计全加器.....	15
<b>第 3 章 VHDL 程序结构</b> .....	19
§ 3.1 实体 (ENTITY).....	19
§ 3.2 结构体 (ARCHITECTURE).....	26
§ 3.3 块语句结构 (BLOCK).....	29
§ 3.4 进程 (PROCESS).....	32
§ 3.5 子程序 (SUBPROGRAM).....	35
3.5.1 函数 (FUNCTION).....	36
3.5.2 重载函数 (OVERLOADED FUNCTION).....	39
3.5.3 过程 (PROCEDURE).....	43
3.5.4 重载过程 (OVERLOADED PROCEDURE).....	45
§ 3.6 库 (LIBRARY).....	46
§ 3.7 程序包 (PACKAGE).....	49
§ 3.8 配置 (CONFIGURATION).....	52
习题.....	54
<b>第 4 章 VHDL 语言要素</b> .....	56
§ 4.1 VHDL 文字规则.....	56
§ 4.2 VHDL 数据对象.....	59
4.2.1 变量 (VARIABLE).....	60
4.2.2 信号 (SIGNAL).....	61
4.2.3 常数 (CONSTANT).....	64
§ 4.3 VHDL 数据类型.....	65
4.3.1 VHDL 的预定义数据类型.....	66
4.3.2 IEEE 预定义标准逻辑位与矢量.....	69
4.3.3 其它预定义标准数据类型.....	71

4.3.4	用户自定义数据类型方式.....	72
4.3.5	枚举类型.....	74
4.3.6	整数类型和实数类型.....	75
4.3.7	数组类型.....	75
4.3.8	记录类型.....	77
4.3.9	数据类型转换.....	79
§ 4.4	VHDL 操作符.....	83
4.4.1	操作符种类.....	83
4.4.2	逻辑操作符.....	84
4.4.3	关系操作符.....	86
4.4.4	算术操作符.....	88
4.4.5	重载操作符.....	95
	习题.....	95
<b>第 5 章</b>	<b>VHDL 顺序语句</b> .....	<b>97</b>
§ 5.1	赋值语句.....	97
5.1.1	信号和变量赋值.....	98
5.1.2	赋值目标.....	99
§ 5.2	流程控制语句.....	101
5.2.1	IF 语句.....	101
5.2.2	CASE 语句.....	104
5.2.3	LOOP 语句.....	108
5.2.4	NEXT 语句.....	112
5.2.5	EXIT 语句.....	113
§ 5.3	WAIT 语句.....	114
§ 5.4	子程序调用语句.....	118
§ 5.5	返回语句(RETURN).....	121
§ 5.6	空操作语句(NULL).....	122
§ 5.7	其它语句和说明.....	123
5.7.1	属性(ATTRIBUTE) 描述与定义语句.....	123
5.7.2	文本文件操作(TEXTIO).....	129
5.7.3	ASSERT 语句.....	130
5.7.4	REPORT 语句.....	131
5.7.5	决断函数.....	131
	习题.....	132
<b>第 6 章</b>	<b>VHDL 并行语句</b> .....	<b>134</b>
§ 6.1	进程语句.....	135
§ 6.2	块语句.....	140
§ 6.3	并行信号赋值语句.....	141



6.3.1 简单信号赋值语句.....	141
6.3.2 条件信号赋值语句.....	142
6.3.3 选择信号赋值语句.....	142
§ 6.4 并行过程调用语句.....	144
§ 6.5 元件例化语句.....	146
§ 6.6 类属映射语句.....	148
§ 6.7 生成语句.....	149
习题.....	154
<b>第 7 章 VHDL 的描述风格</b> .....	<b>157</b>
§ 7.1 行为描述.....	157
§ 7.2 数据流描述.....	159
§ 7.3 结构描述.....	160
习题.....	162
<b>第 8 章 仿真</b> .....	<b>163</b>
§ 8.1 VHDL 仿真.....	163
§ 8.2 延时模型.....	168
8.2.1 固有延时.....	168
8.2.2 传输延时.....	169
§ 8.3 仿真 $\delta$ .....	169
§ 8.4 仿真激励信号的产生.....	169
§ 8.5 VHDL 测试基准.....	172
§ 8.6 VHDL 系统级仿真.....	175
习题.....	176
<b>第 9 章 综合</b> .....	<b>177</b>
§ 9.1 VHDL 综合.....	177
§ 9.2 有关可综合性的考虑.....	180
§ 9.3 寄存器引入方法.....	181
9.3.1 容易发生的错误.....	181
9.3.2 常规寄存器的引入.....	186
9.3.3 具有时钟门控结构寄存器的引入.....	189
9.3.4 同步置位 / 复位功能的引入.....	190
9.3.5 异步置位 / 复位功能的引入.....	191
§ 9.4 引入寄存器的有关技巧.....	193
§ 9.5 三态门引入方法.....	197
§ 9.6 资源共享.....	202
习题.....	204
<b>第 10 章 有限状态机 FSM</b> .....	<b>205</b>
§ 10.1 一般状态机设计.....	206

§ 10.2	状态机的状态编码.....	218
§ 10.3	状态机剩余状态处理.....	220
	习题.....	222
<b>第 11 章</b>	<b>数字滤波器设计</b> .....	<b>223</b>
§ 11.1	基于 FPGA 的数字滤波器优势.....	223
§ 11.2	FIR 数字滤波器设计.....	225
11.2.1	FIR 滤波器结构与原理概要.....	225
11.2.2	FIR 滤波器设计方案确定.....	228
11.2.3	FIR 滤波器主系统设计.....	231
11.2.4	FIR 滤波器附加功能实现.....	235
§ 11.3	IIR 数字滤波器设计.....	237
11.3.1	IIR 滤波器设计方案.....	237
11.3.2	IIR 滤波器的实现.....	240
	习题.....	243
<b>第 12 章</b>	<b>VHDL 设计平台使用向导</b> .....	<b>244</b>
§ 12.1	ispVHDL 使用向导.....	244
12.1.1	ispLSI 系列介绍.....	245
12.1.2	ispVHDL 设计套件介绍.....	245
12.1.3	ispVHDL 设计向导.....	246
§ 12.2	Altera MAX+plus II VHDL 使用向导.....	255
§ 12.3	MAX+plus II 与 Synplify 接口.....	264
§ 12.4	Xilinx Foundation VHDL 使用向导.....	265
12.4.1	Foundation 设计流程.....	266
12.4.2	VHDL 输入方式设计向导.....	267
	习题.....	275
<b>第 13 章</b>	<b>VHDL 设计实践与实验</b> .....	<b>276</b>
§ 13.1	8 位并行预置加法计数器设计.....	276
	实验习题.....	277
§ 13.2	位宽可预置中断处理器.....	279
	实验习题.....	280
§ 13.3	静态随机存储器 (SRAM) 设计.....	280
	实验习题.....	281
§ 13.4	堆栈设计.....	282
	实验习题.....	283
§ 13.5	8 位硬件加法器设计.....	283
	实验习题.....	285
§ 13.6	8 位硬件乘法器设计.....	285
	实验习题.....	290

§ 13.7 乒乓球游戏电路设计.....	290
实验习题.....	296
§ 13.8 序列检测器设计.....	296
实验习题.....	297
§ 13.9 正负脉宽数控调制信号发生器设计.....	297
实验习题.....	299
§ 13.10 “梁祝”乐曲演奏电路设计.....	300
实验习题.....	305
§ 13.11 RS232 通信方式控制电子琴.....	305
实验习题.....	309
§ 13.12 数字频率计设计.....	309
实验习题.....	313
§ 13.13 PC 机、单片机、FPGA 双向通信.....	314
实验习题.....	316
§ 13.14 VGA 显示器彩条信号发生器设计.....	316
实验习题.....	319
§ 13.15 A/D 采样控制器设计.....	320
实验习题.....	324
§ 13.16 D/A 接口电路与波形发生器设计.....	324
实验习题.....	326
§ 13.17 MCS-51 单片机与 CPLD 接口逻辑设计.....	326
13.17.1 总线方式.....	326
13.17.2 独立方式.....	327
实验习题.....	329
§ 13.18 PS/2 键盘接口逻辑设计.....	330
§ 13.19 7 段 LED 译码显示电路设计.....	331
实验习题.....	333
<b>第 14 章 电子设计竞赛实例介绍.....</b>	<b>334</b>
§ 14.1 多功能等精度频率计.....	334
14.1.1 测频原理.....	334
14.1.2 测频专用模块工作原理和设计.....	335
14.1.3 频率计功能模块的 VHDL 描述.....	337
14.1.4 测频主系统实现.....	341
14.1.5 专用模块测试控制信号说明.....	342
§ 14.2 电子设计竞赛开发板.....	342
实战习题.....	345
<b>附录 1 GW48 型 EDA 实验开发系统使用介绍.....</b>	<b>345</b>
<b>附录 2 一些 FPGA 和 CPLD 芯片引脚图.....</b>	<b>362</b>

# 第1章 绪论

电子设计的必由之路是数字化已成为共识。在数字化的道路上,我国电子设计技术的发展经历了,并将继续经历许多重大的变革与飞跃。从应用 SSI 通用数字电路芯片构成电路系统,到广泛地应用 MCU (微控制器或单片机),在电子系统设计上发生了具有里程碑意义的飞跃。这一飞跃不但克服了纯 SSI 数字电路系统许多不可逾越的困难,同时也为电子设计技术的应用开拓了更广阔的前景。它使得电子系统的智能化水平在广度和深度上产生了质的飞跃。MCU 的广泛应用并没有抛弃 SSI 的应用,而是为它们在电子系统中找到了更合理的地位。随着社会经济发展的延伸,各类新型电子产品的开发为我们提出了许多全新的课题和更高的要求。FPGA/CPLD (现场可编程逻辑器件/复杂可编程逻辑器件)在 EDA 基础上的广泛应用,从某种意义上说,新的电子系统运转的物理机制又将回到原来的纯数字电路结构,但这是一种更高层次的循环,应是一次否定之否定的运动,它在更高层次上容纳了过去数字技术的优秀部分,对 MCU 系统将是一种扬弃,但在电子设计的技术操作和系统构成的整体上却发生了质的飞跃。如果说 MCU 在逻辑的实现上是无限的话,那么高速发展的 FPGA/CPLD 不但包括了 MCU 这一特点,并兼有串、并行工作方式和高速、高可靠性以及宽口径适用性等诸多方面的特点。不仅如此,随着 EDA 技术的发展和 FPGA/CPLD 在深亚微米领域的进军,它们与 MCU、MPU、DSP、A/D、D/A、RAM 和 ROM 等独立器件间的物理与功能界限正日趋模糊。特别是软/硬 IP 芯核产业的迅猛发展,嵌入式通用与标准 CPLD 和 FPGA 器件的出现,片上系统已成为可能。以大规模集成电路为物质基础的 EDA 技术终于打破了软硬件之间最后的屏障,使软硬件工程师们有了共同的语言。

## § 1.1 关于 EDA

在电子设计技术领域,可编程逻辑器件的广泛应用,为数字系统的设计带来了极大的灵活性。由于该器件可以通过软件编程而对其硬件的结构和工作方式进行重构,使得硬件的设计可以如同软件设计那样方便快捷。这一切极大地改变了传统的数字系统设计方法、设计过程乃至设计观念。在传统的数字系统设计中,用户能够通过编程方式改变器件逻辑功能只有两种途径,即微处理器的软件编程(如单片机)和特定器件的控制字配置(如 8255)。在传统的设计概念中,器件引脚功能的硬件方式的任意确定是不可能的。而对于系统构成的设计过程只能对器件功能和电路板图分别进行设计和确定,通过设计电路板来规划系统功能。在此期间,大量的时间和精力花在元件选配和系统结构的可行性定位上。但若采用可编程逻辑器件,便可利用计算机软件的方式对目标器件进行设计,而以硬件的

形式实现既定的系统功能。在设计过程中,设计者可根据需要,随时改变器件的内部逻辑功能和管脚的信号方式。借助于大规模集成的可编程逻辑器件和高效的设计软件,用户不仅可通过直接对芯片结构的设计实现多种数字逻辑系统功能,而且由于管脚定义的灵活性,大大减轻了电路图设计和电路板设计的工作量和难度;同时,这种基于可编程逻辑器件芯片的设计大大减少了系统芯片的数量,缩小了系统的体积,提高了系统的可靠性。

纵观可编程逻辑器件(PLD)的发展史,它在结构原理、集成规模、下载方式、逻辑设计手段等方面的每一次进步都为现代电子设计技术的革命与发展提供了不可或缺的强大动力。不难理解,如果失去了可编程逻辑器件,电子设计自动化将成为无源之水。

随着可编程逻辑器件自身功能的不断完善和计算机辅助设计技术的提高,在现代电子系统设计领域中的 EDA 便应运而生了。传统的数字电路设计模式、卡诺图的逻辑化简手段、冗杂难懂的布尔方程表达方式、小规模 TTL 芯片的堆砌技术在迅速崛起的 EDA 面前已成为历史的风景。

EDA 是电子设计自动化(Electronic Design Automation)的缩写,是 20 世纪 90 年代初,从 CAD(计算机辅助设计)、CAM(计算机辅助制造)、CAT(计算机辅助测试)和 CAE(计算机辅助工程)的概念发展而来的。EDA 技术就是以计算机为工具,在 EDA 软件平台上,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地完成逻辑编译、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、逻辑映射和编程下载等工作。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来完成对系统硬件功能的描述,在 EDA 工具的帮助下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。

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就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所谓边界扫描测试。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。

另一方面,高速发展的 CPLD/FPGA 器件又为 EDA 技术的不断进步奠定了坚实的物质基础。CPLD/FPGA 器件更广泛的应用及厂商间的竞争,使得普通的设计人员获得廉价的器件和 EDA 软件成为可能。

现代的 EDA 工具软件已突破了早期仅能进行 PCB 版图设计,或类似某些仅限于电路功能模拟的、纯软件范围的局限,以最终实现可靠的硬件系统为目标,配备了系统设计自动化的全部工具。如配置了各种常用的硬件描述语言平台 VHDL、Verilog HDL、ABEL-HDL 等;配置了多种能兼用和混合使用的逻辑描述输入工具,如硬件描述语言文本输入法(其中包括布尔方程描述方式、原理图描述方式、状态图描述方式等)以及原理图输入法、波形输入法等;同时还配置了高性能的逻辑综合、优化和仿真模拟工具。

所有这一切都为今天的电子设计工程技术人员提供了强有力的工具。在过去令人难以置信的事,今天已成为平常之事,一台计算机、一套 EDA 软件和一片 CPLD 或 FPGA 芯片,就能在家中完成大规模集成电路和数字系统的设计。

未来的 EDA 将会超越电子设计的范畴进入其它领域,随着基于 EDA 的 SOC(单片系统)设计技术的发展,软硬功能核库的建立,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。

## § 1.2 关于 VHDL

VHDL 的英文全名是 Very-High-Speed Integrated Circuit Hardware Description Language, 诞生于 1982 年。1987 年底, VHDL 被 IEEE (The Institute of Electrical and Electronics Engineers) 和美国国防部确认为标准硬件描述语言。自 IEEE 公布了 VHDL 的标准版本 (IEEE-1076) 之后, 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和 VHDL 接口。此后 VHDL 在电子设计领域得到了广泛的接受, 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 VHDL 的内容, 公布了新版本的 VHDL, 即 IEEE 标准的 1076-1993 版本。现在, VHDL 和 Verilog 作为 IEEE 的工业标准硬件描述语言, 又得到众多 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, 在新的世纪中, VHDL 与 Verilog 语言将承担起几乎全部的数字系统设计任务。

### 1. VHDL 的特点

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十分类似于一般的计算机高级语言。VHDL 的程序结构特点是将一项工程设计, 或称设计实体 (可以是一个元件、一个电路模块或一个系统) 分成外部 (或称可视部分, 即端口) 和内部 (或称不可视部分), 即设计实体的内部功能和算法完成部分。在对一个设计实体定义了外部界面后, 一旦其内部开发完成后, 其它的设计就可以直接调用这个实体。这种将设计实体分成内外部分的概念是 VHDL 系统设计的基本点。应用 VHDL 进行工程设计的优点是多方面的, 具体如下:

- 与其它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。就目前流行的 EDA 工具和 VHDL 综合器而言, 将基于抽象的行为描述风格的 VHDL 程序综合成为具体的 FPGA 和 CPLD 等目标器件的网表文件已不成问题, 只是在综合与优化效率上略有差异。

- 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, 其丰富的仿真语句和库函数, 使得在任何大系统的设计早期 (即尚未完成), 就能用于查验设计系统的功能可行性, 随时可对设计进行仿真模拟。即在远离门级的高层次上进行模拟, 使设计者对整个工程设计的结构和功能的可行性作出决策。

- VHDL 语句的行为描述能力和程序结构决定了它具有支持大规模设计的分解和已有设计的再利用功能, 符合市场所需求的, 大规模系统高效、高速的完成必须由多人甚至多个开发组共同并行工作才能实现的特点。VHDL 中设计实体的概念、程序包的概念、设计库的概念为设计的分解和并行工作提供了有力的支持。

- 对于用 VHDL 完成的一个确定的设计, 可以利用 EDA 工具进行逻辑综合和优化, 并

自动地把 VHDL 描述设计转变成门级网表。这种方式突破了门级设计的瓶颈,极大地减少了电路设计的时间和可能发生的错误,降低了开发成本。应用 EDA 工具的逻辑优化功能,可以自动地把一个综合后的设计变成一个更高效、更高速的电路系统。反过来,设计者还可以容易地从综合和优化后的电路获得设计信息,反回去更新修改 VHDL 设计描述,使之更为完善。

- VHDL 对设计的描述具有相对独立性,设计者可以不懂硬件的结构,也不必管最终设计实现的目标器件是什么,而进行独立的设计。正因为 VHDL 的硬件描述与具体的工艺技术和硬件结构无关,VHDL 设计程序的硬件实现目标器件有广阔的选择范围,其中包括各系列的 CPLD、FPGA 及各种门阵列实现目标。

- 由于 VHDL 具有类属描述语句和子程序调用等功能,对于已完成的设计,在不改变源程序的条件下,只需改变端口类属参量或函数,就能轻易地改变设计的规模和结构。

## 2. VHDL 与 Verilog、ABEL 语言的比较

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。具备行为级描述能力的硬件描述语言是以自顶向下方式设计系统级电子线路的基本保证。而 VHDL 语言的特点决定了它更适于行为级(也包括 RTL 级)的描述,难怪有人将它称为行为描述语言。Verilog 属于 RTL 级硬件描述语言,通常只适于 RTL 级和更低层次的门电路级的描述。由于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适配器所接受,因此 VHDL 语言源程序的综合通常要经过行为级→RTL 级→门电路级的转化,而 Verilog 语言源程序的综合过程要稍简单,即经过 RTL 级→门电路级的转化。与 Verilog 相比,VHDL 语言是一种高级描述语言,适用于电路高级建模,比较适合于 FPGA/CPLD 目标器件的设计,或间接方式的 ASIC 设计。随着 VHDL 综合器的进步,综合的效率和效果将越来越好。Verilog 语言则是一种较低级的描述语言,更适用于描述门级电路,易于控制电路资源,因此更适用于直接的大规模集成电路或 ASIC 设计。显然,VHDL 和 Verilog 主要的区别在于逻辑表达的描述级别。VHDL 虽然也可以直接描述门电路,但这方面的能力却不如 Verilog 语言;反之,Verilog 在高级描述方面不如 VHDL。Verilog 语言的描述风格接近于电路原理图,从某种意义上说,它是电路原理图的高级文本表示方式。VHDL 语言适于描述电路的行为,然后由综合器根据功能(行为)要求来生成符合要求的电路网络。

由于 VHDL 和 Verilog 各有所长,市场占有率也相差不多。VHDL 描述语言层次较高,不易控制底层电路,因而对 VHDL 综合器的综合性能要求较高。但是当设计者积累一定经验后会发现,每种综合器一般将一定描述风格的语言综合成确定的电路,只要熟悉基本单元电路的描述风格,综合后的电路还是易于控制的。VHDL 入门相对稍难,但在熟悉以后,设计效率明显高于 Verilog,生成的电路性能也与 Verilog 的不相上下。在 VHDL 设计中,综合器完成的工作量是巨大的,设计者所做的工作就相对减少了;而在 Verilog 设计中,工作量通常比较大,因为设计者需要搞清楚具体电路结构的细节。

目前,大多数高档 EDA 软件都支持 VHDL 和 Verilog 混合设计,因而在工程应用中,有些电路模块可以用 VHDL 设计,其它的电路模块则可以用 Verilog 设计,各取所长,已成为目前 EDA 应用技术发展的一个重要趋势。

ABEL 语言与 Verilog 语言属同一种描述级别 (ABEL 与许多其它的 HDL 在语句格式和用法上具有相似性), 但 ABEL 语言的特性和受支持的程度远远不如 Verilog。Verilog 是从集成电路设计中发展而来, 语言较为成熟, 支持的 EDA 工具很多。而 ABEL 语言是从可编程逻辑器件 (PLD) 的设计中发展而来, ABEL-HDL 是一种支持各种不同输入方式的 HDL, 其输入方式, 即电路系统设计的表达方式, 包括布尔方程、高级语言方程、状态图和真值表。ABEL-HDL 被广泛用于各种可编程逻辑器件的逻辑功能设计, 由于其语言描述的独立性, 因而适用于各种不同规模的可编程器的设计。如 DOS 版的 ABEL3.0 软件可对包括 GAL 器件进行全方位的逻辑描述和设计, 而在诸如 Lattice 的 ispEXPERT、DATAIO 的 Synario、Vantis 的 Design-Direct、Xilinx 的 FOUNDATION 和 WEBPACK 等 EDA 软件中, ABEL-HDL 同样可用于更大规模的 FPGA/CPLD 器件功能设计。ABEL-HDL 还能对所设计的逻辑系统进行功能仿真。ABEL-HDL 的设计也能通过标准格式设计转换文件转换成其他设计环境, 如 VHDL、Verilog-HDL 等。与 VHDL、Verilog-HDL 等硬件描述语言相比, ABEL-HDL 具有适用面宽 (DOS, Windows 版, 及大、中小规模 PLD 设计)、使用灵活、格式简洁、编译要求宽松等优点。虽然有不少 EDA 软件支持 ABEL-HDL, 但提供 ABEL-HDL 综合器的 EDA 公司仅 DATAIO 一家。描述风格一般只用门电路级描述方式。但从 Internet 上获知, ABEL 已经开始了国际标准化的努力。

### § 1.3 关于自顶向下的系统设计方法

传统的电路设计方法都是自底向上的, 即首先确定可用的元器件, 然后根据这些器件进行逻辑设计, 完成各模块后进行连接, 最后形成系统。基于 EDA 技术的所谓自顶向下的 (TOP-TO-DOWN) 设计方法正好相反, 其步骤首先是从整体上对系统设计作详细的规划, 然后完成电路系统功能行为方面的设计, 一般是采用完全独立于具体器件物理结构的硬件描述语言, 如 VHDL, 从系统的基本功能或行为级上对设计的产品进行描述和定义, 进行多层次的仿真评估, 在确保设计的可行性与正确性的前提下, 完成功能确认, 即以 VHDL 描述的系统行为模型的目标就是确保使之具有可模拟性和正确的功能行为。这一切都可以在综合之前完成。其测试仿真方法也同样可以用 VHDL 设计出相应的测试基准, 在行为机上为电路模型产生激励信号, 以检测系统响应的正确性。在行为级仿真测试中, 还能根据目标系统的需要, 利用各种现成的 VHDL 仿真模型, 如 MCU、RAM、ROM、其它的 FPGA 和 ASIC 器件等仿真模型, 对所设计的系统在整个运行上作测试。当一切通过后, 再利用 EDA 工具的逻辑综合功能, 把功能描述转换成一具体目标芯片的网表文件。随着设计流程的下行, 设计项目的详细程度逐渐增加, 输出给该器件厂商的布局布线适配器, 进行逻辑映射及布局布线, 再利用产生的仿真文件进行包括功能和时序的验证, 以确保实际系统的性能。即系统的结构构成方式与系统的行为或算法方式相混的描述 (称为混合层次描述), 由于 VHDL 具有这种描述能力, 设计者就可以在抽象度相当高的层次上描述系统的基本结构, 自顶向下设计方法的优越性表现在: (1) 由于顶层的功能描述可以完全独立于目标器件的结构, 在设计的最初阶段, 设计人员可不受芯片结构的约束, 集中精力对产品进行最适应市场需



求的设计,从而避免了传统设计方法中的再设计风险,缩短了产品的上市周期。(2)设计成果的再利用得到保证。就这方面而言,单片机系统的设计成果难以得到再利用。现代的电子应用系统以及电子产品的开发与生产正向模块化发展,或者说向软硬核组合的方向发展。对于以往成功的设计成果稍作修改、组合就能投入再利用,从而产生全新的或派生的设计模块,同时还可以以一种 IP 核的方式进行存档。(3)由于采用的是结构化开发手段,一旦主系统基本功能结构得到确认,即可实现多人多任务的并行工作方式,使系统的设计规模和效率大幅度提高。(4)在选择实现系统的目标器件的类型、规模、硬件结构等方面具有更大的自由度。

如上所述,基于现代 EDA 技术的自顶向下设计方法有两个重要的阶段,即行为仿真测试阶段和面向实现的综合阶段,在前一阶段里,在整个系统设计的行为级仿真评估中,大量使用现成的,以硬件描述语言表达的器件模型和测试模型;而在最终实现硬件系统的综合过程的阶段中,也同样大量使用现成的,以硬件描述语言表达的功能模块,即 IP Core。

由此我们不难发现,作为优秀的行为级硬件描述语言 VHDL 无疑成了整个设计过程的主角,从顶层目标系统的构建、行为级仿真测试系统的表达、现成的通用测试模型的程序、测试基准的设计,直到可综合系统的行为描述、参与综合的 IP 核的表达、综合后产生的用于时序仿真的文件格式,乃至输出的网表文件,几乎全部可以用 VHDL 来担任。

## § 1.4 关于应用 VHDL 的 EDA 过程

为了使读者对 VHDL 的应用和工程设计的一般过程有个轮廓性的了解,本节将对此作扼要介绍。详细过程可参阅以后各章,特别是第 12、13 章。

由于 VHDL 的适用面比较广,且 EDA 技术的发展日新月异,因此,如图 1-1 所示的 VHDL 工程设计流程图只表示目前比较常用的设计流程,并非最一般的设计流程。本流程图旨在配合本教程的内容,对 VHDL 的应用作一些说明。

图 1-1 所示的整个过程是通常的工程设计的基本流程,也是 VHDL 学习实践所必须完成的过程。VHDL 最初的目的就是用于大规模及超大规模集成电路的设计,作为一种标准硬件描述语言,在工程应用上需要 EDA 工具的支持。一般地,VHDL 的设计文件(程序)需依靠 EDA 软件转换为实际可用的电路网表,最后生成用于 IC 生产的版图,或者由适配软件用此网表对 FPGA / CPLD 进行布线。图 1-1 所示的过程主要是针对目标器件为 FPGA 和 CPLD 的 VHDL 设计。

如图 1-1 所示,一项工程的设计(包括 VHDL 程序的设计和验证)首先需利用 EDA 工具的文本编辑器或图形编辑器将它用文本方式(VHDL 程序方式)或图形方式(流程图方式和状态图方式)表达出来。这两种表达方式必须首先通过 EDA 工具进行排错编译,变成 VHDL 文件格式,为进一步的逻辑综合作准备。在此,对于不少 EDA 软件来说,最初的设计究竟采用哪一种输入形式(文本形式或流程图/状态图形式)是可选的。如果采