

VHDL

语言程序设计及应用 (第2版)

VHDL YUYAN CHENGXU SHEJI JI YINGYONG

姜立东 等编著



北京邮电大学出版社
www.buptpress.com

TP312
694=2D

VHDL 语言程序设计及应用 (第 2 版)

姜立东 等编著

北方工业大学图书馆



00569567

北京邮电大学出版社

·北京·

内 容 提 要

VHDL 语言是国际标准化硬件描述语言,在电子系统自动化设计中已经十分流行,而且成为主要的硬件描述工具。如今在电子系统设计领域中,它已成为广大技术人员必须掌握的一种语言。

本书共分 14 章。第 1~8 章主要介绍 VHDL 语言的基础知识,目的是为初学者打下一个良好的基础;第 9~11 章主要介绍目前硬件电路设计中常用电路的 VHDL 语言程序设计,这部分重点介绍了组合逻辑电路、时序逻辑电路和有限状态机的 VHDL 语言程序设计;第 12~13 章主要介绍了两个大型复杂数字系统的 VHDL 语言程序设计,目的是使读者掌握自顶向下的设计方法;第 14 章主要介绍了一种常用的 EDA 开发工具——XILINX ISE5.1i 开发系统——的基本操作。本书附有一张配套光盘,光盘中包含有本书中的所有 VHDL 语言程序、VHDL 语言标准程序包以及用于教学参考的幻灯片,可满足读者的不同需要。

本书的特点是全面系统、易读易懂、由浅入深,能够使读者逐步掌握 VHDL 语言。本书既可作为高等学校计算机和电子工程专业的研究生、本科生的教材和教学参考书,也可作为广大电子电路设计工程师、ASIC 设计人员和系统设计人员的参考书。

图书在版编目(CIP)数据

VHDL 语言程序设计及应用/姜立东等编著. —2 版. 北京:北京邮电大学出版社,2004

ISBN 7-5635-0897-X

I.V... II.姜... III.硬件描述语言,VHDL—程序设计 IV.TP312

中国版本图书馆 CIP 数据核字(2004)第 044566 号

书 名: VHDL 语言程序设计及应用(第 2 版)
编 著: 姜立东 等
责任编辑: 李欣一
出 版 者: 北京邮电大学出版社(北京市海淀区西土城路 10 号)
邮 编: 100876 电 话: 62282185 62283578
网 址: <http://www.buptpress.com>
经 销: 各地新华书店
印 刷: 北京通州皇家印刷厂
印 数: 1—3 000 册
开 本: 787 mm × 1 092 mm 1/16 印张: 22 字数: 545 千字
版 次: 2004 年 6 月第 1 版 2004 年 6 月第 1 次印刷
书 号: ISBN 7-5635-0897-X/TP·115
定 价: 35.00 元

如有印装质量问题请与北京邮电大学出版社发行部联系

前 言

随着信息技术的不断发展,电子技术面临着严峻的挑战,专用集成电路(ASIC)的设计正面临着设计复杂度不断提高而设计周期不断缩短的矛盾。为了解决这一矛盾,要求设计人员必须采用新的设计方法,使用高层次的设计工具。其中重要的就是采用电子设计自动化(EDA)的方法,这种方法使得几乎所有硬件电路的设计过程都可以通过计算机来完成。这种方法的好处是大大缩短专用集成电路的设计周期,使得生产厂商的产品能够迅速面市,大大提高产品的竞争能力。

在硬件电子电路设计领域中,设计自动化工具已经被广大电子工程师所接受,它必将取代人工设计方法,成为主要的设计手段。电子设计自动化的重要特征就是要求采用形式化语言来描述硬件电子电路的功能,即使用硬件描述语言来描述硬件电子电路。目前,作为硬件描述语言之一的 VHDL 已经成为各家 EDA 工具和集成电路厂商所普遍认同和共同推广的标准化硬件描述语言。掌握 VHDL 语言,学会用 VHDL 语言设计硬件电子电路,是每个电子工程师所必须掌握的一项基本技能。

VHDL 语言是一种非常适用于可编程逻辑器件的硬件语言,并且正在得到普及。VHDL 设计技术是引起电子系统设计方式发生突破性变化的技术。VHDL(Very High Speed Integrated Circuit(VHSIC)Hardware Description Language)作为 IEEE 标准所规范的硬件描述语言,随着各种 EDA 工具和集成电路厂商的普遍认同和推广,目前正在全球范围内先进国家的电子系统设计领域获得广泛应用。

如今在国内电子产业界,无论 ASIC 设计人员还是系统设计人员,都需要学习 VHDL 来提高他们的工作效率;多所高等学校也已先后开设了电子电路设计自动化的课程,要求某些专业的本科生、研究生必须掌握 VHDL 语言。

随着 VHDL 语言的广泛应用,无论是电子设计工程师,还是高等院校的学生都迫切需要一本除了介绍 VHDL 基本概念和基本语法外,还能够从实际出发、着重介绍各种不同电路结构的描述方法的参考书。可以看出,广大读者迫切需要一本这样的书:能够较详细地介绍 VHDL 的基本概念和基本语法;能够介绍目前在硬件电子电路设计中常用的电路结构的 VHDL 描述;能够举一些实例,来描述用 VHDL 语言设计硬件电子电路的流程和在设计过程中所用到的设计技巧等等。作者编写此书的目的就是从以上方面来满足广大读者的需要,以使读者能够全面掌握 VHDL 语言。

本书共分 14 章。第 1~8 章主要介绍 VHDL 语言的基础知识,目的是为初学者打下一个良好的基础;这部分重点介绍了 VHDL 语言的基本常识、VHDL 语言程序的元素和结构、VHDL 语言结构体的子结构、VHDL 语言的顺序描述语句和并行描述语句、VHDL 语言的预定义属性和 VHDL 语言的配置;第 9~11 章主要介绍目前硬件电路设计中常用电路的 VHDL 语言程序设计,这部分重点介绍了组合逻辑电路、时序逻辑电路和有限状态机的 VHDL 语言程序设计;第 12~13 章主要介绍了两个大型复杂数字系统的 VHDL 语言程序设计,目的是使读者掌握自顶向下的设计方法;第 14 章主要介绍了一种常用的 EDA 开发工具——

XILINX ISE5.1i 开发系统——的基本操作,目的是使读者掌握一种 EDA 开发工具的使用。

作者在本书中列举的大量小例子许多都取自于实际硬件电路的设计,并且经过了微机上的 XILINX ISE5.1i 软件和 SUN SPARC 工作站上的 Synopsys 软件的编译验证。本书附有一张配套光盘,光盘中包含了本书中的所有 VHDL 语言程序、VHDL 语言标准程序包以及用于教学参考的幻灯片,可满足读者的不同需要。

该书的特点是全面系统、易读易懂、由浅入深,能够使读者循序渐进地逐步掌握 VHDL 语言。本书既可作为高等学校计算机和电子工程专业的研究生和本科生的教材和教学参考书,也可作为广大电子电路设计工程师、ASIC 设计人员和系统设计人员的参考书。

作者在编写本书的过程中参考了许多关于 VHDL 语言的最新专著及文献,同时本书也包含着作者在使用 VHDL 语言设计硬件电路过程中的经验总结。作者在此向所有对本书提出宝贵建议的学者表示衷心地感谢!

作者有幸通过本书的编写与读者交流使用 VHDL 语言进行硬件电路设计的心得体会。但是鉴于作者的水平,书中一定存在错误和不足之处,请读者不吝批评指正。

作 者

2004 年 5 月

目 录

第 1 章 VHDL 语言概述

1.1 硬件描述语言(HDL)	1
1.1.1 HDL 语言的产生	1
1.1.2 HDL 语言的种类	2
1.1.3 VHDL 语言的发展历史	3
1.2 VHDL 语言的特点	3
1.2.1 VHDL 语言的优点	3
1.2.2 VHDL 语言的缺点	4
1.3 VHDL 语言的开发流程	5
1.3.1 VHDL 语言的开发流程图	5
1.3.2 VHDL 语言的开发步骤	5
本章小结	7
习题 1	7

第 2 章 VHDL 语言程序的元素

2.1 VHDL 语言的对象	8
2.1.1 常量	8
2.1.2 信号	9
2.1.3 变量	10
2.1.4 文件	10
2.1.5 信号和变量的区别	11
2.2 VHDL 语言的数据类型	13
2.2.1 标准定义的数据类型	13
2.2.2 用户定义的数据类型	15
2.2.3 VHDL 语言程序中常用的数据类型	22
2.3 VHDL 语言的运算符	23
2.3.1 逻辑运算符	23
2.3.2 算术运算符	24

2.3.3	关系运算符	25
2.3.4	并置运算符	26
2.3.5	各个运算符的优先级	27
2.4	VHDL 语言的标识符	27
2.4.1	短标识符	27
2.4.2	扩展标识符	28
2.5	VHDL 语言的词法单元	28
2.5.1	注释	28
2.5.2	数字	29
2.5.3	字符和字符串	29
2.5.4	位串	29
	本章小结	30
	习题 2	30

第 3 章 VHDL 语言程序的结构

3.1	VHDL 语言程序的结构组成	32
3.1.1	库	32
3.1.2	程序包	33
3.1.3	实体说明	33
3.1.4	结构体	34
3.1.5	配置	34
3.2	实体说明	36
3.2.1	类属参数说明	36
3.2.2	端口说明	38
3.2.3	实体说明	39
3.3	结构体	39
3.3.1	结构体的书写格式	39
3.3.2	结构体的 3 种描述方式	41
3.4	库	46
3.4.1	库的概念及使用	46
3.4.2	常见的库	47
3.5	程序包	49
3.5.1	程序包的书写结构	49
3.5.2	常见的程序包	50
	本章小结	51
	习题 3	52

第 4 章 VHDL 语言结构体的子结构

4.1 块语句	53
4.1.1 块语句的书写结构	53
4.1.2 块语句的保护	54
4.1.3 块语句的一个完整实例	55
4.1.4 块语句的嵌套	57
4.2 进程语句	58
4.2.1 进程语句的书写结构	59
4.2.2 进程语句的启动	60
4.2.3 进程语句的同步	60
4.3 子程序 1——过程	62
4.3.1 过程的书写结构	63
4.3.2 过程在不同位置的定义	65
4.3.3 过程调用	66
4.4 子程序 2——函数	67
4.4.1 函数的书写结构	67
4.4.2 函数在不同位置的定义	69
4.4.3 函数的调用	70
4.5 VHDL 语言中的重载	72
4.5.1 重载的概念	72
4.5.2 子程序重载的两种方式	72
4.5.3 运算符重载	77
本章小结	79
习题 4	80

第 5 章 VHDL 语言的顺序描述语句

5.1 信号赋值语句和变量赋值语句	81
5.2 WAIT 语句	83
5.2.1 WAIT ON 语句	83
5.2.2 WAIT UNTIL 语句	84
5.2.3 WAIT FOR 语句	85
5.2.4 超时等待的处理	87
5.3 IF 语句	89
5.3.1 IF 语句的书写结构	89
5.3.2 IF_THEN 语句	89
5.3.3 IF_THEN_ELSE 语句	90

5.3.4 IF_THEN_ELSIF_ELSE 语句	91
5.4 CASE 语句	92
5.5 LOOP 语句	94
5.5.1 FOR LOOP 语句	94
5.5.2 WHILE LOOP 语句	96
5.6 跳出循环的语句	97
5.7 RETURN 语句	98
5.8 NULL 语句	99
5.9 顺序断言语句	100
5.10 REPORT 语句	101
本章小结	102
习题 5	103

第 6 章 VHDL 语言的并行描述语句

6.1 进程语句	104
6.2 并行信号赋值语句	107
6.2.1 并发信号赋值语句	107
6.2.2 条件信号赋值语句	109
6.2.3 选择信号赋值语句	111
6.3 并行断言语句	112
6.4 参数传递语句	113
6.5 元件例化语句	115
6.6 生成语句	118
6.6.1 FOR_GENERATE 语句	118
6.6.2 IF_GENERATE 语句	120
本章小结	122
习题 6	123

第 7 章 VHDL 语言的预定义属性

7.1 值类属性	124
7.1.1 常用数据类型的值类属性	124
7.1.2 数组的值类属性	125
7.1.3 块的值类属性	126
7.2 函数类属性	127
7.2.1 数据类型属性函数	128
7.2.2 数组属性函数	128
7.2.3 信号属性函数	130

7.3 信号类属性	132
7.3.1 属性'delayed[(t)]	132
7.3.2 属性'stable[(t)]	134
7.4 数据类型类属性	136
7.5 数据范围类属性	137
本章小结	138
习题 7	139

第 8 章 VHDL 语言的配置

8.1 默认连接和默认配置	140
8.1.1 默认连接	140
8.1.2 默认配置	141
8.2 元件配置	143
8.2.1 元件的低层配置	144
8.2.2 元件的实体-结构体对配置	145
8.3 块的配置	146
8.3.1 块的低层配置	146
8.3.2 块的实体-结构体对配置	148
8.4 结构体的配置	149
本章小结	150
习题 8	150

第 9 章 组合电路的 VHDL 语言程序设计

9.1 基本门电路	152
9.1.1 二输入与门电路	152
9.1.2 其他简单门电路	154
9.1.3 多输入简单门电路	155
9.1.4 三态门电路	157
9.1.5 总线缓冲器	158
9.2 编码器	160
9.2.1 普通编码器	160
9.2.2 优先编码器	161
9.3 译码器	164
9.3.1 变量译码器	164
9.3.2 码制变换译码器	165
9.4 选择器	167
9.5 运算器	168

9.5.1 通用加法器	168
9.5.2 比较器	170
本章小结	171
习题 9	172

第 10 章 时序电路的 VHDL 语言程序设计

10.1 触发器	173
10.1.1 D 触发器	173
10.1.2 JK 触发器	179
10.1.3 T 触发器	180
10.1.4 锁存器	181
10.2 寄存器和移位寄存器	183
10.2.1 寄存器	183
10.2.2 串入/串出移位寄存器	185
10.2.3 串入/并出移位寄存器	186
10.2.4 循环移位寄存器	188
10.3 计数器	190
10.3.1 计数器简介	190
10.3.2 二进制同步计数器	190
10.3.3 同步可逆计数器	192
10.3.4 异步计数器	194
本章小结	196
习题 10	196

第 11 章 有限状态机

11.1 有限状态机的基本概念	197
11.2 有限状态机的基本描述	199
11.2.1 有限状态机的描述方式	199
11.2.2 一个 Moore 型有限状态机的设计实例	200
11.3 有限状态机的同步和复位	210
11.3.1 输出信号的同步	210
11.3.2 有限状态机的同步复位	212
11.3.3 有限状态机的异步复位	215
11.4 改进的 Moore 型有限状态机	217
11.4.1 状态作为输出信号	217
11.4.2 并行输出寄存器的译码输出	220
本章小结	223

习题 11	224
-------------	-----

第 12 章 应用实例——数字密码引爆器的设计

12.1 数字密码引爆器的具体功能	225
12.2 顶层模块的 VHDL 语言程序	226
12.2.1 顶层实体说明的 VHDL 语言程序	227
12.2.2 顶层结构体的设计及其 VHDL 语言程序	228
12.3 底层模块的分析及其 VHDL 语言程序	235
12.3.1 输入消抖同步电路	235
12.3.2 编码电路	237
12.3.3 比较电路	239
12.3.4 预置密码电路	239
12.3.5 计数器选择电路	240
12.3.6 控制指示灯电路	241
12.3.7 分频电路	242
12.3.8 七段显示译码电路	245
12.3.9 其他简单电路	247
12.4 控制器模块的分析及其 VHDL 语言程序	248
本章小结	253
习题 12	254

第 13 章 应用实例——微处理器的设计

第 14 章 XILINX ISE 5.1i 的使用入门

14.1 XILINX ISE5.1i 简介	287
14.2 XILINX ISE5.1i 的入门	288
14.2.1 运行 ISE5.1i 开发系统	288
14.2.2 创建新的工程	289
14.2.3 HDL 方式的源代码输入	289
14.2.4 HDL 方式的源代码综合	293
14.2.5 源代码的 Implement Design 和下载文件生成	303
14.3 ISE5.1i 的原理图设计输入	304
14.3.1 建立一个原理图输入文件	304
14.3.2 例化计数器模块	305
14.3.3 原理图中的连线操作	306
14.3.4 添加网络名	307

14.3.5 添加输入/输出管脚标记	309
本章小结	310
习题 14	311
附录 1 保留字	312
附录 2 标准程序包	313
附录 3 一些有用的网址	336
参考文献	337

第 1 章 VHDL 语言概述

本章教学目的:

- 了解硬件描述语言的历史;
- 熟悉 VHDL 语言的特点;
- 掌握 VHDL 语言的开发流程图和开发步骤。

1.1 硬件描述语言(HDL)

1.1.1 HDL 语言的产生

在科学技术高速发展的今天,集成电路的设计正朝着速度快、性能高、容量大、体积小和功耗微的方向发展,这种发展必将导致集成电路的设计规模日益增大,复杂程度日益提高。在这种情况下,沿用了几十年的传统硬件电路设计方法已不能满足需要,它已经远远落后于当今科学技术的发展。

熟悉电路设计的工程人员知道,传统的硬件电路设计方法主要是采用电路原理图的设计方法。这种设计方法通常是根据硬件电路设计的需要,合理划分功能模块;接下来将对各个功能模块进行详细设计并绘出相应的电路原理图,然后进行模块电路的调试;最后将各个功能模块连接起来进行整体调试,从而完成整个硬件电路的设计。一般来说,这种设计方法主要存在以下缺点:

(1) 工程人员往往采用布尔方程或者门级描述的方法来设计硬件电路,这种方法对于小规模电路非常有效;但是对于大规模复杂的电路来说,采用布尔方程或者门级描述的方法往往需要花费大量的人力和时间。

(2) 在这种设计方法中,电路模拟和调试通常是在设计后期进行的。一旦前面的设计考虑不周到,硬件电路将会存在较大的缺陷,这时要修改就只能重新进行设计,可见这样会大大增加硬件电路的设计周期。

(3) 采用这种设计方法的主要文件是电路原理图。对于小规模电路来说,一般只需要几十张至几百张电路原理图即可;但是对于大规模复杂的电路来说,往往需要几千张甚至几万张电路原理图。如此繁多的电路原理图,必将给硬件电路设计的归档、阅读、修改和使用带来极大的不便。

正是由于上面的原因,广大设计工程人员开始寻找一种新的硬件电路设计方法来代替传统的硬件电路设计方法,目的是为了克服上述缺点,从而满足大规模复杂电路设计的需要。在这种背景下,硬件描述语言(HDL)应运而生了,它的出现为硬件电路的设计带来了一次重大的变革。

顾名思义,硬件描述语言就是指对硬件电路进行行为描述、寄存器传输描述或者结构化描述的一种新兴语言。硬件描述语言的最大优势在于可以借鉴高级程序语言设计的特性来对硬件电路的行为和功能结构进行高度抽象化的描述。同时,硬件描述语言也可以对硬件电路的设计进行不同层次、不同领域的模拟验证和综合优化等处理,从而实现硬件电路设计的高度自动化。采用硬件描述语言来设计硬件电路可以大大节省人力和物力、缩短研制周期,适用于小批量产品开发,也适用于大批量产品的样品研制,因而得到了广泛的应用。

采用硬件描述语言来设计硬件电路的优点主要体现在以下几个方面:

(1) 采用自顶向下的设计方法,可以在设计的早期发现硬件电路设计中的错误,提高电路设计的成功率,缩短开发周期。

(2) 硬件描述语言可以对硬件电路的行为和功能结构进行多层次的抽象描述,可以从数学模型直到门级电路。因而,采用硬件描述语言来设计硬件电路可以提高设计人员的工作效率,降低硬件电路设计的难度。

(3) 硬件描述语言可以使硬件电路的设计更加方便、快捷;而可编程逻辑器件可以将电路设计的前期风险降至最低。硬件描述语言和可编程逻辑器件结合在一起可以大大提高硬件电路设计的效率,使电路产品快速上市。

(4) 硬件电路设计的主要文件是硬件描述语言编写的程序,这必将给硬件电路设计的归档、阅读、修改和使用带来极大的方便。

1.1.2 HDL 语言的种类

现在,国内外硬件描述语言的种类十分繁多,各大 EDA 公司、科研单位甚至高等学校都有自己的硬件描述语言。在这些硬件描述语言中,有些硬件描述语言已经成为 IEEE 标准,但是大多数的硬件描述语言还只是一种工业标准。目前在我国广泛应用的硬件描述语言主要有:ABEL 语言、AHDL 语言、Verilog HDL 语言和 VHDL 语言,其中 Verilog HDL 语言和 VHDL 语言最为流行。

(1) ABEL 语言

ABEL 语言是由美国 DATAIO 公司开发的一种高级可编程逻辑设计语言。由于这种硬件描述语言支持可编程逻辑器件来实现电路的逻辑设计,描述方式和设计方法较为灵活,因此曾经十分流行。近年来,随着其他硬件描述语言的迅速兴起,ABEL 语言已经渐渐淡出了历史舞台。但是,由于 ABEL 语言对于计算机和烧片器等硬件系统适应性强、硬件支持要求较低,因而仍有一些单位或者个人在使用这种硬件描述语言。

(2) AHDL 语言

AHDL 语言是由美国 ALTERA 公司开发的一种高级硬件描述语言。ALTERA 公司是一家享誉全球的半导体器件公司,该公司的 CPLD 器件在世界市场上占有主导地位。同时,ALTERA 公司也是 EDA 工具开发商,它的 EDA 工具 MAX + plus II 和 Quartus 深受广大电路设计人员的欢迎。20 世纪 90 年代,运行在 MAX + plus II 环境下的 AHDL 语言由于其自身的 C 语言设计风格以及 ALTERA 公司的大力推广,已经得到了众多用户群体的使用。但是,AHDL 语言的移植性较差,一般只能在 ALTERA 公司的开发系统上使用,这大大限制了这种语言的使用范围。

(3) Verilog HDL 语言

Verilog HDL 语言最早是由 GDA (Gateway Design Automatic) 公司的设计师 Phil Moorby 在 1983 年开发出来的。在 1984 ~ 1985 年间, Phil Moorby 成功设计了 Verilog-XL 仿真器并于 1986 年提出了快速门级仿真的 XL 算法, 这使得 Verilog HDL 语言变得更加丰富和完善。1989 年, CADENCE 公司收购了 GDA 公司, Verilog HDL 语言从此成为 CADENCE 公司 EDA 设计环境中的硬件描述语言。1990 年, CADENCE 公司公开发表了 Verilog HDL 语言, 并且成立 LVI 组织以促使 Verilog HDL 语言成为 IEEE 标准, 即 IEEE Standard 1364-1995。由于 Verilog HDL 语言是从 C 语言发展而来的, 因而它的应用也十分广泛。

1.1.3 VHDL 语言的发展历史

自从硬件描述语言产生以后, 众多 EDA 公司和科研单位纷纷研制开发了适应自身 EDA 开发工具的硬件描述语言。这些硬件描述语言具有很大的差异, 并且只能在本公司的 EDA 开发工具上使用, 这大大限制了硬件描述语言的使用。因此电路设计人员需要一种强大的面向设计的多层次、多领域并得到广大 EDA 厂商认同的标准化硬件描述语言。

美国国防部在 20 世纪 70 年代末和 80 年代初提出了 VHSIC (Very High Speed Integrated Circuit) 计划, VHSIC 计划的目标是为下一代集成电路的生产、实现阶段性的工艺极限以及完成 10 万门级以上的设计, 建立一项新的描述方法。

1981 年, 美国国防部提出了一种新的硬件描述语言——超高速集成电路硬件描述语言 (VHSIC Hardware Description Language), 简称 VHDL 语言。这个语言只是一个使电路文本化的一种标准, 目的是使文本描述的电路设计能够为其他人所理解, 同时也可以作为一种模型语言并能采用软件进行模拟。

1986 年, IEEE 致力于 VHDL 语言的标准化工作, 并成立了一个 VHDL 语言标准化小组。经过多次反复的修改与扩充, 直到 1987 年 12 月, VHDL 语言才被接纳为 IEEE 1076 标准。1988 年, Milstd454 规定所有为美国国防部设计的 ASIC 产品必须采用 VHDL 语言来进行描述。1993 年, IEEE 1076 标准被修订, 更新为新的 VHDL 语言标准 IEEE 1164。1996 年, IEEE 1076.3 成为 VHDL 语言的综合标准。

VHDL 语言成为标准以后, 很快在世界各地得到了广泛应用, 这为电子设计自动化 (EDA) 的普及和推广奠定了坚实的基础。1995 年中国国家技术监督局制定的《CAD 通用技术规范》推荐 VHDL 语言作为我国电子设计自动化硬件描述语言的国家标准。

1.2 VHDL 语言的特点

1.2.1 VHDL 语言的优点

VHDL 语言能够成为标准化的硬件描述语言并获得广泛应用, 它自身必然具有很多其他硬件描述语言所不具备的优点。归纳起来, VHDL 语言主要具有以下优点:

(1) VHDL 语言功能强大, 设计方式多样

VHDL 语言具有强大的语言结构, 只需采用简单明确的 VHDL 语言程序就可以描述十

分复杂的硬件电路。同时,它还具有多层次的电路设计描述功能。此外,VHDL 语言能够同时支持同步电路、异步电路和随机电路的设计实现,这是其他硬件描述语言所不能比拟的。

VHDL 语言设计方法灵活多样,既支持自顶向下的设计方式,也支持自底向上的设计方法;既支持模块化设计方法,也支持层次化设计方法。

(2) VHDL 语言具有强大的硬件描述能力

VHDL 语言具有多层次的电路设计描述功能,既可描述系统级电路,也可以描述门级电路;描述方式既可以采用行为描述、寄存器传输描述或者结构描述,也可以采用三者的混合描述方式。同时,VHDL 语言也支持惯性延迟和传输延迟,这样可以准确地建立硬件电路的模型。

VHDL 语言的强大描述能力还体现在它具有丰富的数据类型。VHDL 语言既支持标准定义的数据类型,也支持用户定义的数据类型,这样便会给硬件描述带来较大的自由度。

(3) VHDL 语言具有很强的移植能力

VHDL 语言很强的移植能力主要体现在:对于同一个硬件电路的 VHDL 语言描述,它可以从一个模拟器移植到另一个模拟器上、从一个综合器移植到另一个综合器上或者从一个工作平台移植到另一个工作平台上去执行。

(4) VHDL 语言的设计描述与器件无关

采用 VHDL 语言描述硬件电路时,设计人员并不需要首先考虑选择进行设计的器件。这样做的好处是可以使设计人员集中精力进行电路设计的优化,而不需要考虑其他的问题。当硬件电路的设计描述完成以后,VHDL 语言允许采用多种不同的器件结构来实现。

(5) VHDL 语言程序易于共享和复用

VHDL 语言采用基于库(library)的设计方法。在设计过程中,设计人员可以建立各种可再次利用的模块,一个大规模的硬件电路的设计不可能从门级电路开始一步步地进行设计,而是一些模块的累加。这些模块可以预先设计或者使用以前设计中的存档模块,将这些模块存放在库中,就可以在以后的设计中进行复用。

由于 VHDL 语言是一种描述、模拟、综合、优化和布线的标准硬件描述语言,因此它可以使设计成果在设计人员之间方便地进行交流和共享,从而减小硬件电路设计的工作量,缩短开发周期。

1.2.2 VHDL 语言的缺点

虽然 VHDL 语言具有很多其他硬件描述语言所不具备的优点,但是它也并不是一种完全理想的硬件描述语言。同样,它也具有一些自身的缺点,或者说 VHDL 语言还有一些需要不断完善的地方。总的来说,VHDL 语言的缺点体现在以下 3 个方面:

(1) VHDL 语言有时不能准确地描述硬件电路

虽然 VHDL 语言具有强大的硬件描述能力,但是这并不代表它能描述所有的硬件电路。事实上,硬件电路的工作方式是多种多样的,VHDL 语言有时候要想准确地描述出硬件电路的工作方式将是十分困难的。例如,对于数字电路中一些以并行方式工作的组合电路,设计人员常常采用 VHDL 语言的顺序语句来进行描述,因为这时采用并行方式描述是十分困难的,甚至是不可行的。

(2) VHDL 语言的系统级抽象描述能力较差

虽然 VHDL 语言可以对硬件电路的系统级进行描述,但是抽象性不能太强,否则相应的