

if (clk' event) then

所有范例程序的源代码可到

www.khp.com.cn 下载

cclk <= '1';

else

cclk <= '0';

end if;

end if;

VHDL

Design

VHDL 电路设计

D I A N L U S H E J I

end process;

process(cclk, count3, ctemp)

begin

雷伏容 编著

count3 <= count3 + 1;

if(count3 = "000") then

ctemp <= "00000001";

elsif(count3 = "001") then

ctemp <= "00000010";

elsif(count3 = "010") then

ctemp <= "00000100";

elsif(count3 = "011") then

ctemp <= "00001000";

elsif(count3 = "100") then

ctemp <= "00010000";

elsif(count3 = "101") then

ctemp <= "00100000";

elsif(count3 = "110") then

ctemp <= "01000000";

elsif(count3 = "111") then

ctemp <= "10000000";

else

ctemp <= "00000000";

end if;

- 按“基础→电路部件设计→集成电路设计”的顺序,通过174个专业案例,全方位讲解VHDL语言及其在电路设计中的各种应用。
- 以“技术→电路设计实例→应用场合、应用技巧点评”的形式介绍各项技术,实用性强。
- 提供上百个问答题、设计题,强化概念理解,增强设计能力。



清华大学出版社

TP312

2222

2006

VHDL 电路设计

雷伏容 编著



清华大学出版社

北京

内 容 提 要

本书根据作者多年高校EDA教学、研发经验,借助近200个专业案例,全方位介绍VHDL语言及其在电路设计中的各种应用。

本书内容按“基础—电路部件设计—集成电路设计”的顺序进行讲解,基础部分(第1~4章),涉及EDA技术概况、数字逻辑电路设计基础和传统设计方法、VHDL语言的基本知识和程序结构;电路部件设计部分(第5~7章),用72个电路设计案例介绍使用VHDL语言设计组合逻辑电路、时序逻辑电路和有限状态机;集成电路设计部分(第8~10章),第8章借助14个例子介绍用VHDL语言进行复杂数字系统设计的思想、手段、流程和方法,第9章介绍仿真、综合与优化,第10章讲述如何用当前最流行的EDA工具软件Quartus II进行电路设计。各章采用先介绍技术,再用相关电路设计实例演示其应用,并在最后指出其应用场合,实用性强。

本书可作为高等院校电子、通信、计算机、自动化、仪器仪表等专业开设EDA课程的教材,也可作为电子设计工程师的实用参考书。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

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。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

VHDL 电路设计/雷伏容编著. —北京:清华大学出版社, 2006

ISBN 7-302-14226-2

I. V… II. 雷… III. ①硬件描述语言, VHDL—程序设计

②数字电路—电路设计 IV. ①TP312②TN79

中国版本图书馆CIP数据核字(2006)第142423号

出 版 者: 清华大学出版社

<http://www.tup.com.cn>

社总机: 010-62770175

地 址: 北京清华大学学研大厦

邮 编: 100084

客户服务: 010-82896445

组稿编辑: 科海

文稿编辑: 潘秀燕

封面设计: 林 陶

版式设计: 科海

印 刷 者: 北京市耀华印刷有限公司

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 25.5 字数: 620千字

版 次: 2006年12月第1版 2006年12月第1次印刷

书 号: ISBN 7-302-14226-2/TN·374

印 数: 1~4000

定 价: 39.80元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010) 82896445

前 言

随着计算机技术、电子技术的迅速发展以及集成电路生产工艺的不断提高,电子产品的更新换代日新月异。其中数字电路系统设计的电路规模更大,集成度更高,而且要求设计周期短,灵活性强,面对不同的用户,能方便修改和迅速升级。传统的手工设计和分立元件设计显然是不能适应这些要求的,数字电路的设计方法从手工、分立、小型化、纯硬件的设计方法逐渐变成运用计算机、大规模地、软硬件协同设计的大规模专用集成电路设计。也就是说,以计算机为主要开发工具,在EDA(电子设计自动化)开发软件平台上,对用硬件描述语言(Hardware Description Language, HDL)描述的数字电路系统,自动地完成编译、逻辑化简、逻辑综合和布局布线,以及进行优化和仿真测试。因此,EDA技术成为现代电子设计工程师必须掌握的技术。采用EDA技术进行数字集成电路的设计开发,可以极大地提高电子产品设计的正确性、缩短产品的开发周期以及增加产品修改的灵活性。

EDA技术的关键在于采用硬件描述语言来描述硬件电路。硬件描述语言从20世纪80年代初发展到现在已经有近30年的历史了,最初各家EDA公司都有自己的硬件描述语言,这些众多的硬件描述语言对于不同的开发工具和不同厂商生产的芯片互相不兼容,因此设计的可移植性、可重用性差,相互之间交流困难,所以许多EDA公司、芯片生产厂商以及数字电路设计者均希望对此有一个统一的标准。在此情况下,最早由美国国防部为了超高速集成电路计划而开发的硬件描述语言VHDL(Very High Speed Integrated Circuit HDL)在1987年被IEEE标准化,并在1993年进行了进一步的修订。VHDL语言是被IEEE标准化的两种语言(VHDL和Verilog HDL)之一,得到了业界广泛的应用,目前所有EDA厂商的软件开发工具和可编程逻辑器件都支持它。

VHDL语言具有以下突出的优点:

(1) IEEE标准化的语言,所有EDA厂商的软件开发工具和可编程逻辑芯片均支持它,也即采用VHDL语言开发的数字电路,与开发平台以及硬件实现芯片无关,可移植性、可重用性好。

(2) VHDL语言能够在系统级、行为级、寄存器传输级、门级等各个层次对数字电路进行描述,并可以在不同层次进行不同级别的仿真,能极大地保证设计的正确性和设计指标的实现。

本书作为一本帮助初、中级用户,从数字电路设计向大规模集成电路开发过渡的教材,首先介绍了数字逻辑电路的一些基本概念和传统设计方法,然后介绍了VHDL语言的基本知识和程序结构,并通过众多的实例详细讲述了采用VHDL语言进行基本数字电路开发的过程,最后介绍了采用VHDL语言进行数字系统开发的设计流程、设计思想和一些设计技巧。

本书从实用角度出发,考虑到读者的知识接受顺序,根据作者多年在高校进行EDA课程教学的经验,参考众多国内外优秀教材编写而成。全书共分10章和4个附录,第1章是对EDA技术及VHDL语言的概述;第2章是介绍数字逻辑电路的一些基本概念和基本数字电路传统设计方法;第3~4章主要讲述了VHDL语言的程序结构、基本文字规则和语法规则;第5~7章是基本组合逻辑电路、时序逻辑电路和有限状态机的VHDL语言描述;第8章是介绍采用VHDL语言进行复杂数字系统开发的设计思想、设计手段、设计流程和设计实例;第9章介绍了在采用VHDL语言进行数字逻辑电路开发时,在前端设计完成后还需要进行的几个重要步骤:仿真、综合、优化和设计实现的概念与方法;第10章介绍了目前应用非常广泛的,可编程逻辑器件的集成开发平台——Altera公司的Quartus II集成开发软件的使用方法;附录A为VHDL保留字;附录B为常用逻辑符号对照表;附录C为常用VHDL开发软件;附录D为习题答案。

本书主要面向广大EDA工程师以及大中专院校的师生,同时本书还具有较高的工程实用性,也可作为高级用户的使用参考手册。

本书由北京化工大学信息学院的雷伏容老师执笔编写,在编写过程中引用并参考了国内外众多教材、文献以及网络资料,在此表示衷心感谢。另外,在本书的编写过程中,得到了北京化工大学信息学院的领导、老师和同学们的大力帮助和支持,得到了众多同事和朋友们的帮助,在此一并表示衷心感谢。此外,赵昌剑、李辉、王从明、刘湘、宋婷婷、张清欢、李大宇、欧阳志强、马天一、潘媛、崔凌、于陈华、李欣和葛爱琳等同志在整理材料方面给予了编者很大的帮助,在此,编者对他们表示衷心的感谢。

由于编写时间仓促,作者水平有限,错误和不当之处在所难免,恳请各位读者批评指正。最后,感谢您选择了本书,也请您把对本书的意见和建议告诉我们,电子邮箱sckj0001@163.com。

为方便读者学习,特将书中所有实例的程序代码摘录出来放于科海网站上,读者可登录科海网站(www.khp.com.cn)进行下载。

编 者

2006年11月

目 录

第1章 概述	1
1.1 硬件描述语言介绍	1
1.2 VHDL语言的发展与特点	2
1.3 VHDL语言的开发流程	4
1.4 可编程逻辑器件简介	5
1.5 学习小结	7
1.6 思考与练习	7
第2章 数字逻辑基础	8
2.1 逻辑代数基础	8
2.1.1 逻辑变量和逻辑函数	8
2.1.2 基本逻辑运算	8
2.1.3 逻辑函数的表示方法	11
2.1.4 逻辑代数的定理和规则	12
2.2 组合逻辑电路	13
2.2.1 组合逻辑电路的分析	14
2.2.2 组合逻辑电路的综合	15
2.2.3 组合逻辑电路中的竞争与冒险	16
2.3 时序逻辑电路	18
2.3.1 触发器	18
2.3.2 时序逻辑电路的分析	22
2.3.3 时序逻辑电路的设计	26
2.4 学习小结	28
2.5 思考与练习	29
第3章 VHDL程序结构与要素	31
3.1 VHDL程序基本结构	31
3.1.1 实体	32
3.1.2 结构体	35
3.1.3 描述风格	36
3.2 VHDL文字规则	40
3.2.1 数值型文字	40
3.2.2 标识符	42
3.3 VHDL基本数据对象与数据类型	43
3.3.1 VHDL数据对象	43
3.3.2 VHDL数据类型	47

3.4 VHDL表达式与运算符	57
3.5 学习小结	63
3.6 思考与练习	63
第4章 VHDL基本描述语句	64
4.1 顺序描述语句	64
4.1.1 WAIT语句	65
4.1.2 代入语句	68
4.1.3 赋值语句	69
4.1.4 IF语句	71
4.1.5 CASE语句	75
4.1.6 FOR LOOP语句	78
4.1.7 WHILE LOOP语句	79
4.1.8 NEXT语句	80
4.1.9 EXIT语句	82
4.1.10 RETURN语句	83
4.1.11 REPORT语句	83
4.1.12 NULL语句	84
4.1.13 ASSERT语句	85
4.2 并行描述语句	86
4.2.1 并发信号代入语句 (Concurrent Signal Assignment)	87
4.2.2 进程语句 (PROCESS)	92
4.2.3 元件调用语句 (COMPONENT)	94
4.2.4 端口映射语句 (PORT MAP)	96
4.2.5 类属参数传递语句 (GENERIC)	98
4.2.6 块语句 (BLOCK)	100
4.2.7 过程语句 (PROCEDURE)	105
4.2.8 函数 (FUNCTION)	108
4.2.9 生成语句 (GENERATE)	111
4.2.10 延时语句	115
4.3 属性描述语句	117
4.3.1 属性预定义	117
4.3.2 数值类属性描述语句	117
4.3.3 函数类属性描述语句	124
4.3.4 信号类属性描述语句	132
4.3.5 数据类型类属性描述语句	139
4.3.6 数据区间类属性描述语句	140
4.3.7 用户自定义属性语句	141
4.4 学习小结	142

4.5 思考与练习	144
第5章 组合逻辑电路的VHDL语言描述	145
5.1 基本逻辑门电路	145
5.1.1 2输入与非门电路	145
5.1.2 2输入或非门电路	148
5.1.3 反相器电路	149
5.1.4 2输入异或门电路	150
5.1.5 2输入同或门电路	151
5.1.6 多输入门电路	152
5.2 选择器电路	155
5.2.1 2选1选择器	156
5.2.2 16选1选择器	157
5.3 编码器与译码器电路	159
5.3.1 编码器	159
5.3.2 译码器	161
5.4 三态门及总线缓冲器电路	165
5.4.1 三态门	165
5.4.2 总线缓冲器	167
5.5 加法器电路	170
5.6 求补器电路	177
5.7 乘法器电路	178
5.8 数值比较器电路	182
5.9 移位器电路	185
5.10 学习小结	186
5.11 思考与练习	187
第6章 时序逻辑电路的VHDL语言描述	189
6.1 时序电路的时钟信号	189
6.1.1 时钟边沿的描述	189
6.1.2 时钟作为敏感信号	191
6.2 时序电路的复位信号	193
6.2.1 时序电路的同步复位方法	193
6.2.2 时序电路的异步复位方法	194
6.3 锁存器电路	195
6.3.1 电平锁存器	195
6.3.2 同步锁存器	196
6.3.3 异步锁存器	197
6.3.4 定制锁存器	198
6.4 触发器电路	200

6.4.1 D触发器	200
6.4.2 T触发器	205
6.4.3 JK触发器	208
6.5 寄存器电路	210
6.5.1 10位通用寄存器	211
6.5.2 串入-串出移位寄存器	211
6.5.3 串入-并出移位寄存器	214
6.5.4 循环移位寄存器	216
6.5.5 双向移位寄存器	220
6.5.6 串/并入-串出移位寄存器	221
6.6 计数器电路	225
6.6.1 同步计数器	225
6.6.2 异步计数器	230
6.6.3 自定义计数器	231
6.7 学习小结	234
6.8 思考与练习	235
第7章 有限状态机的VHDL语言描述	236
7.1 有限状态机概述	236
7.2 有限状态机的VHDL语言描述	237
7.3 Moore型状态机的设计	247
7.4 Mealy型状态机的设计	250
7.5 有限状态机的状态编码	252
7.5.1 符号化的状态编码	252
7.5.2 直接状态编码	252
7.6 非法状态的处理	258
7.7 学习小结	261
7.8 思考与练习	261
第8章 采用VHDL的数字系统设计	262
8.1 层次化结构设计	262
8.2 自顶向下的设计方法	263
8.3 库	265
8.4 程序包	267
8.5 配置	275
8.6 系统设计实例	280
8.6.1 存储器设计	281
8.6.2 4位微处理器设计	287
8.7 学习小结	296
8.8 思考与练习	296

第9章 仿真、综合与优化	297
9.1 仿真	297
9.1.1 仿真输入信号的生成	298
9.1.2 仿真的 δ 延时	305
9.2 综合	306
9.2.1 约束条件	307
9.2.2 工艺库	308
9.2.3 门级映射	309
9.3 设计优化	309
9.3.1 模块复用	310
9.3.2 流水线设计	313
9.4 设计实现	315
9.5 学习小结	317
9.6 思考与练习	317
第10章 在Quartus II中进行电路设计	318
10.1 Quartus II软件的安装	318
10.1.1 配置要求	318
10.1.2 安装步骤	319
10.1.3 授权方法	324
10.1.4 设计特点	326
10.2 Quartus II软件的设计流程	328
10.2.1 图形用户界面设计流程	328
10.2.2 EDA工具设计流程	330
10.2.3 命令行可执行文件	331
10.2.4 Quartus II软件的设计流程	332
10.2.5 常用窗口介绍	333
10.3 设计项目输入	334
10.3.1 创建工程项目文件	335
10.3.2 创建图形编辑文件	336
10.3.3 创建文本编辑文件	346
10.3.4 创建存储器编辑文件	348
10.3.5 创建波形编辑文件	350
10.4 设计项目编译与仿真	353
10.4.1 Quartus II软件的编译器选项设置	353
10.4.2 编译设计项目	359
10.4.3 仿真设计项目	361
10.5 设计项目时序分析	364
10.5.1 时序分析的基本参数介绍	364

10.5.2 时序分析选项设置	364
10.5.3 进行时序分析	366
10.5.4 查看时序分析结果	367
10.6 设计项目的下载编程	367
10.6.1 器件编程介绍	368
10.6.2 进行器件编程	368
10.7 学习小结	370
10.8 思考与练习	370
附录A VHDL保留字	371
附录B 常用逻辑符号对照表	372
附录C 常用VHDL开发软件	372
附录D 习题答案	373
参考文献	393

本书案例程序目录

例3-1: 2选1多路选择器程序	31
例3-2: 有类属说明的2输入与非门的实体描述	33
例3-3: n输入与非门的实体描述	33
例3-4: 半加器的完整VHDL描述	36
例3-5: 2选1多路选择器行为描述程序	37
例3-6: 2选1多路选择器数据流描述程序	37
例3-7: 2选1多路选择器结构描述程序	38
例3-8: 半加器的混合描述程序	39
例3-9: 打开一个字符文件, 读出文件中的内容并关闭文件	51
例4-1: WAIT语句示例程序	65
例4-2: WAIT FOR语句示例程序	65
例4-3: WAIT ON语句示例程序 (2选1选择器)	66
例4-4: WAIT ON语句和PROCESS语句中所使用的敏感信号列表的对比	67
例4-5: WAIT UNTIL语句示例程序	67
例4-6: 多条件WAIT语句的示例程序	68
例4-7: 信号代入语句示例程序	68
例4-8: 变量赋值语句示例程序	69
例4-9: 变量赋值和信号量代入的对比示例程序	70
例4-10: 采用单IF语句描述D触发器的示例程序	71
例4-11: 采用二选择IF语句描述2选1选择器的示例程序	72
例4-12: 采用多选择IF语句描述4选1选择电路的示例程序	73
例4-13: 采用IF语句嵌套结构的带复位端的4选1选择器的示例程序	74
例4-14: 采用CASE语句来描述4选1选择器的示例程序	76
例4-15: 采用CASE语句设计3-8译码器的示例程序	77
例4-16: 采用FOR LOOP语句实现32位奇偶校验电路的示例程序	78
例4-17: WHILE LOOP语句格式的示例程序	79
例4-18: 采用WHILE LOOP语句实现32位奇偶校验电路的示例程序	80
例4-19: 采用外部信号控制的32位奇偶校验电路的示例程序	81
例4-20: NEXT语句嵌在两个循环中的示例程序	81
例4-21: 采用EXIT语句的示例程序	82
例4-22: 采用RETURN语句的示例程序	83
例4-23: 采用REPORT语句的示例程序	84
例4-24: 使用NULL语句的示例程序	84
例4-25: 采用ASSERT语句的示例程序	85
例4-26: 采用条件信号代入语句描述异或门的示例程序	89
例4-27: 采用条件信号代入语句描述3-8译码器的示例程序	89
例4-28: 采用条件信号代入语句来描述4选1选择器的示例程序	90
例4-29: 采用选择信号代入语句描述异或门的示例程序	91
例4-30: 采用选择信号代入语句来描述4选1选择器的示例程序	91
例4-31: 利用进程语句设计半加器的示例程序	93
例4-32: 利用在一个构造体中有两个进程语句通信的示例程序	93

例4-33: 利用COMPONENT语句的2选1选择器的示例程序	95
例4-34: 利用GENERIC语句定义二输入或非门的上升沿和下降沿参数的示例程序	98
例4-35: 利用GENERIC语句的示例程序	99
例4-36: 利用BLOCK语句设计2选1选择器的示例程序	100
例4-37: 利用BLOCK语句和PROCESS语句嵌套设计半加器的示例程序	101
例4-38: 利用BLOCK语句的CPU芯片设计的示例程序	102
例4-39: 利用卫式BLOCK语句设计D触发器的示例程序	104
例4-40: 利用PROCEDURE语句结构把位矢量转换成整数的示例程序	106
例4-41: 在主程序中调用过程bitvector_to_integer的示例程序	107
例4-42: 利用FUNCTION语句结构把位矢量转换成整数的示例程序	109
例4-43: 利用FUNCTION语句结构实现取最大值的示例程序	110
例4-44: 调用函数getmax()实现最大值检出的示例程序	110
例4-45: 用FOR-GENERATE语句设计4位移位寄存器的示例程序	112
例4-46: 用一般的方法设计4位移位寄存器的示例程序	113
例4-47: 利用IF-GENERATE语句设计n位移位寄存器的示例程序	114
例4-48: 获取数值型枚举类型的数值属性的示例程序	118
例4-49: 用DOWNT0来排列数据时, 获取该数据的数值属性的示例程序	118
例4-50: 利用数据的数值属性检测数据总线宽度的示例程序	119
例4-51: 获取字符型枚举类型的数值属性的示例程序	120
例4-52: 利用'LENGTH属性来获取数组的范围长度的示例程序	120
例4-53: 利用'LENGTH属性来获取枚举类型的多维数组范围长度的示例程序	121
例4-54: 说明块的数值属性'Structure和'BEHAVIOR的示例程序	123
例4-55: 利用数据类型的属性函数解析欧姆定律的示例程序	125
例4-56: 说明'SUCC、'PRED、'LEFTOF、'RIGHTOF属性使用方法的示例程序	127
例4-57: 利用数组的属性函数的示例程序	128
例4-58: 属性函数'EVENT和'LAST_VALUE的使用方法的示例程序	129
例4-59: 利用信号的属性函数'LAST_EVENT检查建立时间的示例程序	131
例4-60: 利用属性'DELAYED (time) 描述三输入或门的示例程序	133
例4-61: 利用信号类属性'DELAYED检查保持时间的示例程序	134
例4-62: 利用属性'STABLE (time) 的示例程序	136
例4-63: 利用属性'QUIET (time) 描述具有优先级的中断机制的示例程序	136
例4-64: 利用属性'BASE的示例程序	139
例4-65: 使用数据区间类属性描述语句——属性'RANGE和属性'REVERSE_RANGE的示例程序	140
例4-66: 使用用户自定义属性语句ATTRIBUTE的示例程序	142
例5-1: 一般2输入与非门电路设计的示例程序1	146
例5-2: 一般2输入与非门电路设计的示例程序2	146
例5-3: 调用集电极开路的2输入与非门电路的示例程序	147
例5-4: 2输入或非门电路设计的示例程序1	148
例5-5: 2输入或非门电路设计的示例程序2	148
例5-6: 反相器电路设计的行为描述方式的示例程序	149
例5-7: 反相器电路设计的RTL描述方式的示例程序	149
例5-8: 2输入异或门电路设计的行为描述方式的示例程序	150
例5-9: 2输入异或门电路设计的结构描述方式的示例程序	150
例5-10: 2输入同或门电路设计的行为描述方式的示例程序	151
例5-11: 2输入同或门电路设计的结构描述方式的示例程序	152

例5-12: 3输入或门电路设计的行为描述方式的示例程序	153
例5-13: 3输入或门电路设计的结构描述方式的示例程序	153
例5-14: 4输入与非门电路设计的行为描述方式的示例程序	154
例5-15: 4输入与非门电路设计的结构描述方式的示例程序	154
例5-16: 采用IF-ELSE语句实现2选1选择器的示例程序	156
例5-17: 采用CASE语句实现2选1选择器的示例程序	157
例5-18: 采用IF-ELSIF-ELSE语句实现16选1选择器的示例程序	158
例5-19: 采用IF-ELSE语句实现BCD输出的10线-4线优先级编码器电路的示例程序	160
例5-20: 4线-16线译码器电路的示例程序	162
例5-21: BCD码输入的4线-10线译码器电路的示例程序	164
例5-22: 多种方法实现三态门电路的示例程序	166
例5-23: 两种方法实现74244形式的单向总线缓冲器的示例程序	168
例5-24: 实现74245形式的双向总线缓冲器电路的VHDL示例程序	169
例5-25: 半加器电路的VHDL示例程序	171
例5-26: 直接根据真值表设计的全加器电路的VHDL示例程序	172
例5-27: 由两个半加器元件和一个或门构成的全加器电路的VHDL示例程序	173
例5-28: 采用行波进位的4位加法器电路的VHDL示例程序	173
例5-29: 基于兆函数LPM_ADD_SUB模块生成的定制加/减法电路的VHDL示例程序	176
例5-30: 采用行为描述方法的16位求补器电路的示例程序	177
例5-31: 4位×4位原码移位的乘法器电路的示例程序	179
例5-32: 基于兆函数LPM_MULT模块生成的定制8位×8位无符号乘法器电路的VHDL示例程序	180
例5-33: 4位输入数据的一般数值比较器的VHDL示例程序	182
例5-34: 基于兆函数LPM_COMPARE模块生成的定制8位与8位无符号数据比较器电路的VHDL 示例程序	184
例5-35: 对8位输入数据进行移位的VHDL示例程序	185
例6-1: 显式表示时钟敏感信号(上升沿)的示例程序	191
例6-2: 隐式表示时钟敏感信号(下降沿)的示例程序	192
例6-3: VHDL程序中描述时序电路同步复位的示例程序	193
例6-4: VHDL程序中描述时序电路异步复位的示例程序	194
例6-5: 单输入电平锁存器的VHDL模型的示例程序	195
例6-6: 多输入电平锁存器的VHDL模型的示例程序	196
例6-7: 同步锁存器的VHDL模型的示例程序	197
例6-8: 异步锁存器的VHDL模型的示例程序	197
例6-9: 基于兆函数LPM_LATCH模块生成的定制的带有异步置位端和异步清零端的8位锁存器 电路的VHDL示例程序	199
例6-10: 上升沿触发的D触发器的示例程序	201
例6-11: 异步复位的D触发器的示例程序	201
例6-12: 异步复位/置位的D触发器的示例程序	202
例6-13: 同步复位的D触发器的示例程序	203
例6-14: 基于兆函数LPM_DFF模块生成的定制的带有异步置位端(aset)和异步清零端(aclr)的 16位D触发器电路的VHDL示例程序	204
例6-15: 下降沿触发的T触发器的示例程序	206
例6-16: 基于兆函数LPM_TFF模块生成的定制的带有同步置位端(sset)、同步清零端(sclr)和 同步加载端(sload)的24位T触发器电路的VHDL示例程序	208
例6-17: 带有异步置位/复位端的上升沿触发的JK触发器的示例程序	210

例6-18: 用VHDL语言描述的10位通用寄存器的示例程序	211
例6-19: 用VHDL语言描述的8位串入-串出移位寄存器的示例程序	212
例6-20: 直接采用信号代入的方法实现串入-串出移位寄存器的示例程序	213
例6-21: 用VHDL语言描述的8位串入-并出移位寄存器的示例程序	215
例6-22: 用VHDL语言描述的8位循环左移位寄存器的示例程序	217
例6-23: 调用包集合CYPAC中的循环左移过程的8位循环左移位寄存器示例程序	218
例6-24: 用VHDL语言描述的8位双向移位寄存器的示例程序	221
例6-25: 用VHDL语言描述的8位串/并入-串出移位寄存器74166的示例程序	223
例6-26: 用VHDL语言描述的带使能端的十二进制计数器的示例程序	226
例6-27: 利用两个4位二进制计数器构成一个六十进制计数器的示例程序	227
例6-28: 用VHDL语言描述的8位二进制加/减计数器的示例程序	229
例6-29: 用VHDL语言描述的4位行波型异步计数器电路的示例程序	230
例6-30: 基于兆函数LPM_COUNTER模块生成的自定义的带有时钟使能端 (clk_en)、计数使能端 (cnt_en)、异步置位端 (aset)、异步加载端 (aload) 和异步清零端 (aclr) 的12位模30加/减 (updown) 计数器电路的VHDL示例程序	232
例7-1: 交通灯控制系统的VHDL有限状态机实现	240
例7-2: 交通灯控制器采用单进程描述的有限状态机实现	243
例7-3: 步进电机控制器的有限状态机实现	248
例7-4: Mealy型有限状态机	250
例7-5: 顺序编码的有限状态机	253
例7-6: 采用状态位直接输出编码的交通灯控制系统	255
例7-7: 一位热码编码的有限状态机	257
例8-1: 全加器的设计	276
例8-2: 采用元件配置说明方式实现的全加器的设计	277
例8-3: 64×8bit 的ROM初始化 (采用在程序包中直接定义的方法)	281
例8-4: 64×8bit 的ROM初始化 (采用读TEXTIO的方式)	282
例8-5: 64×8bit 的ROM设计 (采用在程序包中直接定义的方法)	282
例8-6: 64×8bit 的ROM设计 (采用读TEXTIO的方式)	283
例8-7: 32×4bit 的RAM设计	284
例8-8: 16×4bit的FIFO设计	285
例8-9: ALU输入选择模块设计	290
例8-10: ALU算术逻辑运算模块设计	290
例8-11: 存储器模块设计	292
例8-12: 寄存器模块设计	293
例8-13: 输出选择模块设计	293
例8-14: 4位微处理器系统的顶层描述	294
例9-1: 全加器仿真程序 (采用程序直接生成全加器仿真输入信号)	299
例9-2: 全加器仿真程序 (利用TEXTIO文本文件的方式来生成仿真输入信号)	302
例9-3: 全加器仿真程序 (不采用观察仿真输出波形, 而使用完整的测试平台程序)	304
例9-4: 仿真程序	305
例9-5: 加法器示例程序	310
例9-6: 采用了面积优化的加法器示例程序	311
例10-1: 由Quartus II软件生成的电路模块testblock下层的VHDL语言文件testblock.vhd的示例程序	340
例10-2: 由Quartus II软件的文本编辑器的“插入模板”对话框生成的实体声明的示例程序 (采用VHDL语言编写程序)	347

第1章 概述

随着电子技术的飞速发展,各种电子设备及数字系统的复杂度、集成度越来越高,现代电子产品性能进一步提高,产品更新换代的节奏越来越快,要求产品开发周期短、开发成本低、保密性和可修改以及可扩展性好,因此对集成电路的设计方法不断提出了新的要求。目前,传统的手工设计不能满足微电子技术飞速发展对集成电路设计的要求,已经被先进的电子设计自动化技术(EDA技术)所代替。所谓EDA(Electronic Design Automation)技术,是依赖目前功能已十分强大的计算机为工具,代替人完成数字系统的设计、逻辑综合、布局布线和仿真工作的技术。

现代的EDA设计工具,主要有两种设计输入方式:图形输入方式和硬件描述语言输入。对于图形输入的方法,由于对电子设计工程师在硬件方面的要求较高,而且具有设计的可移植性差等缺点,因此在复杂的数字系统设计中,一般很少采用。而采用硬件描述语言进行设计,由于其具有与硬件和设计平台无关、强大的对电路系统及其行为的建模及描述能力、易读性、可移植性好以及层次化与结构化设计性能等诸多优点,目前在数字系统设计中得到广泛应用。因此熟悉和掌握硬件描述语言以及这些设计理念,已成为集成电路和数字系统设计工程师必备的素质。本章将向读者简单介绍硬件描述语言的产生、特点以及开发流程等。

1.1 硬件描述语言介绍

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、信号连接关系以及各器件间的时序关系。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。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。

传统的数字电路设计方法,需要在整个产品设计完成之后,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,可以在设计初期发现和修改绝大多数错误,这样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。

硬件描述语言自20世纪70年代在学术界开始使用,发展至今,已有30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。

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

(1) 用硬件描述语言(HDL)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。比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, RTL)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,设计者不用重新设计电路,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(HDL)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,他们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(HDL)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(HDL)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(HDL)推到了数字电路设计的最前沿,设计者不再需要使用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 VHDL语言的发展与特点

硬件描述语言(HDL)有很多种,但是最流行和通用的只有VHDL和Verilog HDL两种。它们作为IEEE标准化的硬件描述语言,具有许多优点:能够抽象地描述电路结构和行为、支持逻辑设计层次与领域的描述、硬件描述与实现工艺无关、易于理解和可移植性好等。

但是VHDL相对于Verilog HDL而言,在语法上更严谨一些。虽然这样也使它失去了