

高等学校“十二五”规划教材

电子信息与通信工程系列

Verilog HDL 数字系统设计

Digital Design using Verilog HDL

王建民 田晓华 江晓林 编著

Verilog HDL 数字系统设计

Digital Design using Verilog HDL

王建民 田晓华 江晓林 编著

内容简介

本书通过大量完整的实例介绍基于 Verilog HDL 进行数字系统设计的基本原理、概念和方法。全书重点关注基于 HDL 的寄存器传输级(Register Transfer Level, RTL)数字系统设计,主要包括数字电路基础回顾、组合逻辑电路设计、规则时序逻辑电路、有限状态机以及 FSMD 的设计。本书所有代码兼容 Verilog HDL IEEE 1364—2001标准。

尽管本书简单地回顾了数字电路的基本知识,但是如果读者能够掌握数字电路的基本原理和设计方法对于阅读本书将是十分有意义的。本书适合高年级的本科生、研究生以及从事数字电路设计的工程人员使用。

图书在版编目(CIP)数据

Verilog HDL 数字系统设计/王建民,田晓华,江晓林
编著. —哈尔滨:哈尔滨工业大学出版社,2011.4
ISBN 978 - 7 - 5603 - 3169 - 0

I. ①V… II. ①王… ②田… ③江… III. ①硬件描述语言, Verilog HDL - 程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2011)第 012889 号

责任编辑 许雅莹
封面设计 刘长友
出版发行 哈尔滨工业大学出版社
社 址 哈尔滨市南岗区复华四道街 10 号 邮编 150006
传 真 0451 - 86414749
网 址 <http://hitpress.hit.edu.cn>
印 刷 哈尔滨工业大学印刷厂
开 本 787mm×1092mm 1/16 印张 23 字数 560 千字
版 次 2011 年 6 月第 1 版 2011 年 6 月第 1 次印刷
书 号 ISBN 978 - 7 - 5603 - 3169 - 0
印 数 1 ~ 2 000 册
定 价 44.00 元

(如因印装质量问题影响阅读,我社负责调换)

高等学校“十二五”规划教材

电子信息与通信工程系列

编 委 会

主 任 吴 群

编 委 (按姓氏笔画排序)

于晓洋	王艳春	史庆军	齐怀琴	刘 梅
孙道礼	邹 斌	何 鹏	杨明极	周 成
宗成阁	孟维晓	胡 文	姜成志	姚仲敏
赵志杰	赵金宪	童子权	冀振元	魏凯丰

总 序

电子信息与通信工程是当今世界发展最快的领域,该技术领域的新概念、新理论、新技术不断涌现,其知识更新速度也是令人吃惊。这就使得从事电子信息与通信工程技术的科技人员要不断学习,把握前沿动态,吸收最新知识。近年来,各高校通过教学改革,在引导学生将最新知识应用于社会实践,解决实际问题,培养学生实践动手能力、探索性学习能力和创新思维能力等方面取得了可喜成就。

为了培养国家和社会急需的电子信息与通信工程领域的高级科技人才,配合高等院校电子信息与通信工程专业的教学改革和教材建设,哈尔滨工业大学出版社组织哈尔滨工业大学、哈尔滨理工大学、齐齐哈尔大学、佳木斯大学、黑龙江科技学院等多所高校,通过共同研讨和合作,相互取长补短、发挥各自的优势和特色,联合编写了这套面向普通高等院校“电子信息与通信工程系列”教材。

本系列教材的编写目标:结合新的专业规范,融合先进的教学思想、方法和手段,体现科学性、先进性和实用性,强调对学生实践能力的培养,以适应新世纪对通信、电子人才培养的需求。

本系列教材编写要求:专业基础课教材概念清晰、理论准确、深度合理、内容精练、并注意与专业课教学的衔接;专业课教材覆盖面广、深度适中,体现相关领域的新发展和新成果,注重理论联系实际。

本系列教材的编委会阵容强大,编者都是在教学工作第一线的骨干教师。他们具有多年丰富的教学和科研经历,掌握最新的理论知识,具有丰富的实践经验,是一支高水平的教材编写队伍。

本系列教材理论性与工程实践性紧密结合,旨在引导读者将电子信息与通信工程的理论、技术与应用有机结合,适合于高等学校电子、信息、通信和自动控制等专业的教材选取。我深信:这套教材的出版,对于推动电子信息与通信工程领域的教学改革、提高人才培养质量必将起到积极的推动作用,并以其内容的先进性、实用性和系统性为特色而获得成功。

吴 群

哈尔滨工业大学教授

2010年4月

前 言

PREFACE

本书特色

随着微电子以及计算机技术的深入发展,传统的模拟电子电路的应用已经越来越少,数字电路(系统)逐渐显示出越来越多的优势。数字信息更容易传输、存储和处理;数字系统具有更强的抗干扰性;数字器件价格也更加低廉,因此,数字系统设计已经成为高等学校电子信息类专业学生必须掌握的基础能力。

目前,国内高等学校数字电路课程主要分为两个层次,一个层次是数字电路基础课程(几乎所有的电子信息类专业都会开设此类课程),这类课程的教材基本是 20 世纪 80 年代左右编写的,其内容除数字系统设计的基本概念外,主要介绍基于中小规模集成电路(如 74 系列)的数字系统的分析和设计。另一个层次是只在某些相关专业开设的数字系统设计课程,这一层次的课程并没有形成统一地授课内容,不同层次的高校和专业讲授的内容存在很大差异。但是,随着近二十年可编程逻辑器件(Programmable Logic Devices, PLDs)技术的快速发展,基于硬件描述语言(Hardware Description Language, HDL)的数字系统设计已经不再是集成电路设计的专利,越来越多的工程师采用 HDL 进行数字系统的设计,并将其应用到自己的产品中。采用硬件描述语言进行数字系统设计已经成为数字系统设计的主流。因此,越来越多的高等学校在电子信息类专业开设基于硬件描述语言或者可编程逻辑器件的高级数字系统设计课程。但是此类课程的教材建设相对滞后,国内关于 HDL 的著作绝大多数侧重于语法本身的介绍而对于数字系统(电路)设计原理及方法则涉及较少。

本书主要介绍 Verilog HDL 数字系统设计的基本原理和方法。内容主要涉及:基于 Verilog HDL 的数字系统设计的基本概念、原理和方法;基于 Verilog HDL 的组合逻辑、规则时序逻辑电路、有限状态机设计、带数据通道的有限状态机的设计。本书的主要特色如下:

- ① 全书以讲述数字系统的概念、原理和方法为主;
- ② 通过大量完整实例讲解数字系统设计的基本概念和设计方法;
- ③ 全书包括难易程度不同的各种类型的设计实例 146 个,所有的设计实例均给出完整的 Verilog HDL 代码。

读者对象

本书适合高年级的本科生、研究生以及从事数字电路设计的工程人员使用。尽管本书简单地回顾了数字电路的基本知识,但是如果读者能够掌握数字电路的基本原理和设计方法对于阅读本书将是十分有意义的。

组织结构

全书共分3个部分,第1部分(第1~4章)主要介绍数字电路的基础知识以及 Verilog HDL 的基本语法,是全书的基础。

第2部分(第5~10章)介绍基于 Verilog HDL 的数字电路设计方法。其中第5、6章介绍基本组合逻辑电路,规则时序逻辑电路的设计方法;第7~10章介绍较复杂的同步数字系统设计方法,包括同步有限状态机的设计、带数据通道的有限状态机结构的数字设计以及时序分析的基本原理等内容。

第3部分(第11章)给出了一个完整的 SPI 主机接口模块的设计实例作为从基本设计理论和方法到工程实践的过渡。

全书篇幅较大,有利于授课教师灵活选材,也为学生自学提供了较好的条件。推荐以下授课方案供参考:

第1种方案:1→2→3→4→5→6→8

适合于电气信息类专业以 Verilog HDL 硬件描述语言为授课目标的本科生或者研究生课程,建议上课学时在40学时;学习过数字电路基础的学生可以跳过第2~3章。

第2种方案:1→2→3→4→5→6→7→8→9→10

适合于以 Verilog HDL 为工具进行数字系统设计为授课目标的本科生或者研究生课程,建议学时在48~56学时。

本书包含大量的设计实例,教师可以根据课时以及专业特点选择适当难度的实例作为授课内容。本书配套课件和部分习题答案可以通过 email 向作者索取或者通过出版社网站下载。

本书由王建民、田晓华、江晓林编著,其中第5、6、8、9、10章由王建民编写,第1、7、11章由田晓华编写,第2、3、4章由江晓林编写,全书由王建民统稿。

本书编写过程参考了国内外同行的大量文献,研究生马宁为本书的代码录入和整理做了大量工作,作者借此机会,向他们表示衷心感谢。

限于水平和能力有限,书中疏漏之处在所难免,敬请读者批评指正。

来信请寄作者邮箱:wjmverilog@163.com。

作者

2011年1月

目 录

CONTENTS

第 1 章 数字系统设计概述	1
1.1 引言	1
1.2 ASIC 和 FPGA	2
1.3 数字设计的层次	3
1.4 硬件描述语言	5
1.5 典型设计流程	7
本章小结	8
习题与思考题 1	8
第 2 章 组合逻辑电路设计回顾	9
2.1 数字电路的基本概念	9
2.2 布尔代数和逻辑门	10
2.3 逻辑函数的化简	14
2.4 组合逻辑电路的设计方法	18
2.5 若干常用组合逻辑电路	19
本章小结	28
习题与思考题 2	28
第 3 章 时序逻辑设计回顾	29
3.1 时序逻辑电路	29
3.2 基本存储元件	30
3.3 时序逻辑电路的分析	35
3.4 时序逻辑电路的设计	38
3.5 若干常用的时序逻辑电路	41
本章小结	47
习题与思考题 3	47
第 4 章 Verilog 硬件描述语言	49
4.1 引言	49
4.2 第 1 个 Verilog HDL 实例	49
4.3 基本词法规定	50
4.4 数据类型	52

4.5 程序框架	55
4.6 结构级描述	57
4.7 门级描述	60
4.8 Testbench	64
本章小结	66
习题与思考题 4	66
第 5 章 组合逻辑电路	68
5.1 引言	68
5.2 连续赋值语句	68
5.3 Verilog HDL 操作符	70
5.4 组合逻辑 always 块	73
5.5 If 语句	76
5.6 case 语句	83
5.7 条件语句的综合	86
5.8 可重用设计	89
5.9 组合逻辑电路设计实例	93
5.10 高效的 HDL 描述	109
5.11 组合逻辑电路设计要点	129
本章小结	133
习题与思考题 5	133
第 6 章 基本时序逻辑电路	135
6.1 引言	135
6.2 时序逻辑电路	135
6.3 同步时序逻辑电路	138
6.4 基于原语的时序电路设计	140
6.5 基本存储元件的 Verilog HDL 实现	142
6.6 设计实例	146
6.7 时序逻辑电路的 Testbench	159
6.8 时序逻辑电路设计要点	162
本章小结	170
习题与思考题 6	171
第 7 章 同步时序逻辑电路的时序分析	172
7.1 引言	172
7.2 Verilog HDL 的抽象层次	172
7.3 同步时序电路的时序分析方法	174
7.4 组合逻辑的传播延迟	177
7.5 时序逻辑电路的传播延迟	179
7.6 提高电路的最高工作频率	184
7.7 提高电路的建立时间和保持时间	187

本章小结	188
习题与思考题 7	188
第 8 章 有限状态机	190
8.1 引言	190
8.2 有限状态机	190
8.3 米利状态机和摩尔状态机	191
8.4 状态转换图和算法状态机图	196
8.5 有限状态机的性能和时序	201
8.6 状态赋值	203
8.7 FSM 的 Verilog HDL 实现	208
8.8 输出缓冲器	222
8.9 设计实例	229
本章小结	237
习题与思考题 8	237
第 9 章 数据通道(FSMD)	239
9.1 引言	239
9.2 寄存器传输级设计	240
9.3 FSMD 设计原理	242
9.4 FSMD 设计方法和步骤	245
9.5 流水线设计	259
9.6 FSMD 设计实例	273
本章小结	279
习题与思考题 9	279
第 10 章 FSMD 设计实践	281
10.1 引言	281
10.2 定点数的表示及饱和算术运算	281
10.3 混合方程	284
10.4 混合方程的直接实现	286
10.5 输入寄存器和输出寄存器	290
10.6 流水线设计和流水线执行单元	291
10.7 资源共享数据通道的设计	295
10.8 带有握手信号的数据通道	299
10.9 具有输入总线的数据通道	302
10.10 递归计算、初始化和计算	307
10.11 复杂数据通道的设计方法	311
10.12 寄存器的 Schedule	314
10.13 数据流图的等价变形	320
本章小结	321
习题与思考题 10	321

第 11 章 SPI 主机接口设计	322
11.1 引言	322
11.2 SPI 总线标准	322
11.3 SPI 主机功能描述	324
11.4 微控制器接口模块	327
11.5 SPI 主机接口模块	337
本章小结	356
习题与思考题 11	356
参考文献	357

第 1 章

数字系统设计概述

1.1 引言

对数字信号进行算术运算和逻辑运算的电路称为数字电路(Digital Circuit),或数字系统(Digital System);由于其具有逻辑运算和处理功能,所以又称为数字逻辑电路(Digital Logic Circuit)。数字设计(Digital Design)^①的目标是构建具有一定具体功能的实际物理电路,根据其实现平台的不同,其最终的实现可能是 ASIC(Application-Specific Integrated Circuit, ASIC)芯片,也可能是复用于大型数字设计项目中的 IP(Intellectual Property, IP)软核,也可能是基于某型号 FPGA(Field Programmable Gate Array)器件的应用系统。

过去 40 年,数字系统经历了巨大改进和提高,单个芯片中包含的晶体管的数目以指数规律增长。今天,在一块普通的芯片内可能包含成百上千,甚至上亿个晶体管。然而,随着芯片体积变得越来越小,速度变得越来越快,成本不断降低,功能越来越强大,许多电子系统、控制系统、通信系统甚至某些机械系统都被“数字化”,都会使用数字器件存储、处理以及传输信息。

现实的需求促使数字系统的结构和功能变得越来越复杂,传统的基于中小规模集成电路(74 系列)的数字系统设计已经无法适应数字系统在功能、体积以及成本等方面的需要,同时也促使数字系统的设计方法发生了巨大改变。传统的基于原理图的设计方法已经无法适应现代数字系统设计的需求。因此,从 20 世纪 70 年代末期开始,在各大 EDA(Electronic Design Automatism)公司和大学以及研究机构的共同努力下,出现了多种类型的硬件描述语言(Hardware Description Language, HDL),用来描述数字系统的结构和功能。在众多的 HDL 中,VHDL(Very High Speed Integrated Circuit HDL)和 Verilog HDL 凭借自身的优势,最终成为业界进行数字电路设计的标准硬件描述语言。两种语言都有着大量的使用者,其中 Verilog HDL 在美国、日本以及我国大陆和台湾地区比较受欢迎,VHDL 在欧洲更为普及。设计者采用 HDL 从更高的抽象层次对数字系统进行建模,使用 EDA 软件获得实际电路结构并对其功能和时序进行仿真和验证已经成为现代数字系统的最佳方式。这种设计方式的最大优势是使设计者不必过分关注电路实现的具体细节,而将主要精力集中到电路功能设计上。

这里需要强调的是,虽然各种数字系统设计工具发展迅速,可以自动执行某些设计任务(比如综合(synthesis)),但就目前的 EDA 软件而言,一般只能执行有限的转换^②和局部优化,还无法将一个风格不良的 HDL 描述自动转换成高效的电路实现,最终实现还是要依靠设计者的智慧和经验。

^① 有时也被称为逻辑设计(Logic Design),指数字电路或者数字系统的设计。

^② 当前,综合软件一般能够将 RTL 级描述有效转换为电路网表,对更高抽象层次的描述会存在一定问题。

1.2 ASIC 和 FPGA

数字设计有多种不同的实现方式,从简单的现场可编程逻辑器件到全定制集成电路都可以作为其实现平台。一般需要根据具体应用的特点,选择合适的实现方式。

在一块芯片上制造多个晶体管,并完成一定功能,称为集成电路(Integrated Circuit, IC)。现代大规模集成电路在一块芯片上可能包含上亿个晶体管。根据 IC 逻辑功能的定制(customization)方式不同,集成电路可以分为专用集成电路和可编程逻辑器件两类。专用集成电路的功能事先确定,终端用户在使用过程中不能改变其功能。可编程逻辑器件功能由终端用户根据应用要求自行设计,并可在“现场”改变。

在选择具体的实现方式时,主要考虑逻辑功能的定制方式。在某些实现方式中,器件的每一层^①都事先确定,因此这类器件只适用于实现通用逻辑功能。与之相反,某些逻辑功能的定制“现场(in the field)”实现,在应用现场将编程文件下载(配置)到器件内部,使其实现需要的逻辑功能。逻辑的定制涉及掩模板(mask)的设计以及光刻等复杂加工过程,成本昂贵,一般只能在集成电路工厂(foundry or a fab)完成。因此逻辑功能是否需要在 fab 定制是器件的重要特征,本书称逻辑功能需要在 fab 中加工的器件为 ASIC。根据逻辑电路实现方式属于 ASIC 还是非 ASIC,本节介绍以下几种数字设计的实现方式:

(1) 全定制 ASIC

全定制 ASIC(Full - custom ASIC)中,电路的所有特征都针对一个具体应用。电路设计以及加工的所有方面,甚至包括晶体管的布局都需要根据电路的性能要求而专门设计。也就是说,全定制 ASIC 是针对具体应用充分优化的,因此具有最佳的性能。但是,晶体管级电路设计异常复杂,一般只适合于小规模电路的设计,采用这种方法设计大规模数字系统是不现实的。因此,全定制 ASIC 设计方法主要应用于中小规模的通用逻辑器件的设计。另外,全定制 ASIC 也用于设计某些 1 位的逻辑电路,比如 1 位加法器和 1 位存储器。通过级联这些 1 位的逻辑单元可以构造 N 位宽的逻辑单元。

(2) 基于标准单元的 ASIC

基于标准单元的 ASIC(Standard - cell ASIC)使用预先定义的标准逻辑单元(Standard cell)构造逻辑电路。标准逻辑单元是事先定义的,而且经过充分的验证和测试。由于使用标准逻辑单元设计电路从而简化了整个电路的设计过程。器件制造商通常提供标准单元库作为电路设计的基本单元。标准单元库通常包含基本逻辑门、简单的组合逻辑元件(比如与门、或门、非门、2 选 1 数据选择器以及 1 位全加器等)和基本的存储元件(比如 D 锁存器和触发器)。某些标准单元库可能包含更复杂的逻辑单元,比如加法器、筒式移位寄存器以及 RAM 等。

标准单元 ASIC 设计的基础是标准逻辑单元,设计时根据电路功能不同选择合适的标准单元以及连接模式。标准单元电路的设计是事先完成的,但是整个 ASIC 电路的布局需要根据具体应用进行设计和优化。因此,标准单元 ASIC 的加工与全定制 ASIC 相同,需要在 fab 中进行。

(3) 复杂的现场可编程逻辑器件

最典型的非 ASIC 器件称为复杂的现场可编程逻辑器件(Complex field - Programmable Logic

^① ASIC 和 FPGA 器件的加工都是分层进行的。

Device, CPLD)。CPLD 由通用的逻辑单元阵列(Generic logic cell)和互联结构(interconnect structure)组成。通用逻辑单元和互联结构事先已经确定,但两者都是现场可编程的。器件的编程通过半导体“熔丝(fuses)”或者“开关(switches)”技术实现。具体应用中,逻辑功能的实现通过采用特定的编程文件对器件进行配置实现。通常情况下,主机和可编程逻辑器件之间需要连接1个下载器,用于将主机上生成的编程文件下载到可编程逻辑器件。因为逻辑功能的定制是“现场”实现的,因此这类器件被称为现场可编程逻辑器件。

CPLD 中的基本逻辑单元也称为宏单元(macrocell)。通常,宏单元由一维或者二维的与或阵列加触发器构成,其中与或阵列实现组合逻辑,触发器用于实现记忆功能。现场可编程逻辑器件内部互联结构也是事先确定的。为了降低宏单元之间连接的复杂性,通常使用逻辑功能更加强大的宏单元。根据宏单元逻辑功能的复杂程度不同,复杂的现场可编程逻辑器件分为两大类:复杂的可编程逻辑器件(Complex Programmable Logic Device, CPLD)和现场可编程门阵列(Field Programmable Gate Array, FPGA)。

CPLD 的宏单元的逻辑功能更为复杂,通常由 D 触发器和 PAL^① 组成。CPLD 器件的内部互联结构一般比较集中。FPGA 内部的标准逻辑单元的规模较小,典型情况下,由 D 触发器和查找表(LookUp Table, LUT)以及多个数据选择器实现。FPGA 的内部互联结构则分布到器件各处,而且更加灵活。因为内部互联结构分布到器件的各处,因此,FPGA 适合大规模、高复杂性的数字系统设计问题。

(4)简单的现场可编程逻辑器件

简单的现场可编程逻辑器件(Simple field - Programmable Logic Device, SPLD)的内部结构更为简单,有些情况下这类器件也称为可编程逻辑器件(Programmable Logic Device, PLD)。为了与 CPLD 和 FPGA 区别,本书称这类器件为 SPLD。SPLD 通常由两级逻辑构成,即1级与平面和1级或平面。两级与或平面或者只有与平面可以编程,以积之和形式实现特定的逻辑功能。这类器件包含许多种类,包括 PROM(Programmable Read Only Memory)、PAL(Programmable Array Logic)、PLA(Programmable Logic Array)等。

与以上介绍的 FPGA 和 CPLD 不同,SPLD 器件不包含通用的互联结构,因此 SPLD 能够实现的逻辑功能有限,目前这类器件已经很少使用。

随着 FPGA 技术不断成熟,FPGA 器件的性能也不断提高。目前 FPGA 技术已经能够和 ASIC 技术相媲美,FPGA 器件不但可以直接用于实现应用系统,而且可以作为 ASIC 的验证平台,因此采用 FPGA 实现数字系统已经成为数字系统设计的重要方向。本书的重点在于介绍数字逻辑的设计,所介绍的内容与数字逻辑的实现方式无关。学习本书的内容一般不需要理解具体实现方式,但是具备相关的知识对读者更好理解本书的内容大有裨益。

1.3 数字设计的层次

现代数字设计异常复杂,一个完整的数字系统设计往往被划分为不同的设计层次。设计人员通常只在某一个具体的层次上从事设计工作,而不能兼顾整个完整的设计过程。掌握其他设计层次的知识,从整体上把握自己从事的设计工作在整个设计流程中的地位和作用,是成

^① PAL(Programmable Array Logic)由两级可编程的与或阵列组成的一种数字器件。

为优秀的数字设计人员的必要基础。

数字设计的最底层是器件物理以及 IC 工艺设计。这一层次的设计人员主要从事基础理论、新材料、新器件以及新工艺的研究,目的是提高 IC 的工作速度、集成度。这一层次的进展对整个数字系统设计和应用的影响可能是革命性的。过去二十年,IC 技术的发展可以用摩尔定律描述:一块 IC 芯片上可以集成的晶体管的数目,每年都会翻一番。当然,最近几年,IC 集成度的增长速度已经有所下降,约每 18 个月会翻一番。IC 技术的这些进展都得益于基础理论的发展以及新工艺进步。本书不涉及器件物理以及 IC 制造工艺方面的内容。

(1) 晶体管级设计

晶体管级设计的基本单元是晶体管,设计者采用晶体管作为设计起点,设计不同功能的逻辑电路。例如,图 1.1 给出了一个由 CMOS 传输门和晶体管构成的数据选择器。注意:这里只是给出简介,关于电路的具体实现可以参考本书第 2 章或者其他数字电路基础教材。通常而言,逻辑功能复杂的数字系统,一般需要上百万甚至上千万的晶体管,以晶体管为起点进行设计并不现实。因此,晶体管级设计一般只用于设计基本的逻辑单元(逻辑门以及基本存储元件,比如触发器),通常情况下,IC 制造厂商会在晶体管级设计自己的标准单元库(stand cell library)。

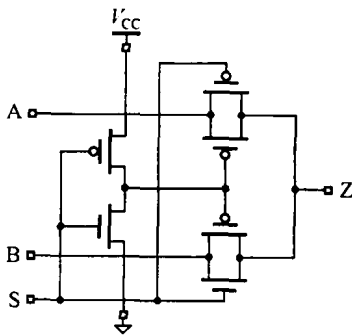


图 1.1 数据选择器的晶体管级原理图

(2) 门级设计

门级设计采用基本门电路作为基本单元进行数字系统的设计。设计者首先对设计问题进行抽象,采用布尔代数表示。还是以上面介绍的数据选择器为例,介绍门级设计基本过程。2 选 1 数据选择器的逻辑真值表如表 1.1 所示。通常情况下,采用真值表表示输入和输出之间的逻辑关系,并对依据真值表获得的逻辑表达式进行化简。对于表 1.1 所示的 2 选 1 数据选择器,其化简结果为

$$Z = \bar{S} \cdot A + S \cdot B$$

其门级实现方式如图 1.2 所示。

与晶体管级设计相比,门级设计的起点已经有了很大的提高,对于某些设计任务而言,其设计难度显著下降,正因如此,门级设计一度成为数字设计的主要方式,设计人员采用商业化的中小规模的集成电路(74 系列的逻辑门)设计各种不同的逻辑电路。但是随着设计任务变得越来越复杂,门级设计正在被基于硬件描述语言的数字设计方法所取代。

表 1.1 数据选择器的真值表

S A B	Z
000	0
001	0
010	1
011	1
100	0
101	1
110	0
111	1

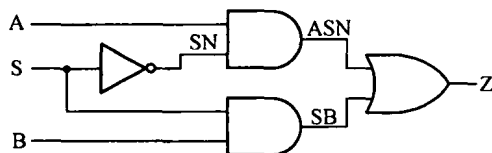


图 1.2 数据选择器的门级实现

采用硬件描述语言进行数字设计使设计者将更多精力集中于系统功能的设计,不必花更多精力在具体的实现方式上。采用 HDL 也可以在晶体管级、门级进行数字系统的设计,但更多情况下,会在更高的抽象层次(比如寄存器传输级)上进行设计,这也是采用 HDL 进行数字设计的优势所在。本书重点关注基于 HDL 的寄存器传输级设计,寄存器传输级设计的最基本单元是中等规模的电路单元,比如加法器、数据选择器等。

1.4 硬件描述语言

目前,采用 HDL 进行数字系统设计已经成为设计方法的主流。数字系统设计流程的各个关键步骤都会使用 HDL。首先基于 HDL 建立电路模型,其次采用综合软件将 RTL 级或者更高抽象级别的 HDL 模型综合成实际的物理电路,以及之后电路的功能验证、时序分析以及故障覆盖率的测试过程都会采用 HDL。

1.4.1 基于 HDL 的数字系统设计

与传统的基于原理图设计方式相比,基于 HDL 的数字系统设计的最大好处在于设计者可以将主要精力集中于电路功能的实现上,而不必关心如何采用晶体管实现具体电路。因此,大大提高了设计效率,这在如此激烈的市场竞争中是有优势的。其次,HDL 描述是基于文本的描述方式,这对于以团队协作为基础的大规模现代电路设计具有重要意义,基于文本的描述方式便于设计的管理、维护以及存档;而且,由于 HDL 是业界普遍接受的标准描述方法,方便移植以及交流。第三,采用 HDL 进行数字系统设计,与具体的实现工艺无关,大大提高了数字系统设计的效率。最后,基于 HDL 进行数字系统设计,设计者在可以设计的早期对电路的功能进行验证,并能在设计的初期发现和排除设计中的大多数错误。

1.4.2 Verilog HDL 和 VHDL

1. Verilog HDL

Verilog HDL 是一种业界普遍采用的硬件描述语言,用于从算法级、门级到开关级等多种抽象设计层次的数字系统建模。

Verilog HDL 从 C 语言中继承了多种操作符和语法结构,尽管二者有着本质上的区别,但其语法规则与 C 语言非常相似。考虑到绝大多数的数字设计工程师都熟悉 C 语言,因此

Verilog HDL 语言的入门相比较 VHDL 语言更为简单。

Verilog HDL 语言不仅定义了语法结构,而且对每个语法结构都定义了清晰的模拟、仿真语义。因此,用 Verilog HDL 描述的数字系统模型能够使用 Verilog HDL 仿真器进行验证。Verilog HDL 提供了扩展的建模能力,其中许多扩展对于初学者都很难理解;但是,Verilog HDL 语言的核心子集非常易于学习和使用,这对大多数建模应用来说已经足够。当然,完整的硬件描述语言足以对从最复杂的芯片到完整的电子系统进行描述。

Verilog HDL 语言最初是 1983 年由 Gateway Design Automation 公司(后来被 Cadence Design Systems 公司收购)为其仿真器产品开发的硬件建模语言,其设计初衷只是用于该公司开发的仿真器产品,是一种专用语言。Verilog HDL 作为一种便于使用且实用的语言逐渐为众多设计者所接受。Verilog HDL 于 1990 年被推向公众领域,目前由 OVI(Open Verilog International)组织负责升级和维护。1995 年,Verilog HDL 正式成为 IEEE 标准,称为 IEEE Std 1364—1995。IEEE Std 1364—1995 标准存在一些问题,2001 年,IEEE 发布了 IEEE Std 1364—2001 标准,对 Std 1364—1995 标准中存在的问题进行了改进并增加了一些新的特性^①。目前绝大多数的主流 EDA 软件都支持 Std 1364—2001 标准。2005 年,IEEE 发布了 IEEE Std 1364—2005 标准,对原标准出现的一些问题做了更正并加入了一些新的语言特征,同时增加了一个独立标准 Verilog-AMS,尝试对模拟电路以及混合信号电路的支持。

2. VHDL

VHDL 是另外一种成为 IEEE 标准的硬件描述语言,最初由美国国防部(Department of Defense)组织开发,旨在提高设计的可靠性和缩减开发周期。起初 VHDL 是一种小范围使用的设计语言,1987 年底,VHDL 成为 IEEE 标准。自 IEEE 公布了 VHDL 的标准版本 IEEE Std 1076—1987(简称 87 版)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支持,并逐步取代了原有的非标准硬件描述语言。1993 年,IEEE 对 VHDL 标准进行了修订,从更高的抽象层次和系统描述能力上扩展 VHDL 的描述能力,公布了新版本的 VHDL 标准,即 IEEE Std 1076—1993 版本(简称 93 版)。

目前,VHDL 和 Verilog HDL 作为 IEEE 的标准硬件描述语言,得到众多 EDA 公司的支持,已成为事实上的通用硬件描述语言。有专家认为,VHDL 与 Verilog 语言将承担起大部分的数字系统设计任务。

需要指出,采用 Verilog HDL 还是 VHDL 作为设计语言并不重要^②,本书选择 Verilog HDL 只是考虑其入门简单、容易上手的特点。作为一个成熟的数字设计工程师,Verilog HDL 和 VHDL 都应该熟悉。最低的要求应该是能够读懂一种,熟练掌握另一种进行设计。HDL 只是数字系统的设计工具,虽然对工具的掌握对于高效进行数字系统设计至关重要,但是更为重要的是对于数字设计的基本原理和理论的学习,只有深入掌握了数字系统设计的基本原理和理论,才能设计出符合实际需求的数字系统,这样的前提下学习设计工具才是有意义的。

^① 本书的代码完全符合 IEEE Std 1364—2001 标准。

^② 在国外的某些数字设计论坛中,讨论采用 Verilog HDL 还是 VHDL 作为数字设计的入门语言是被禁止的。