



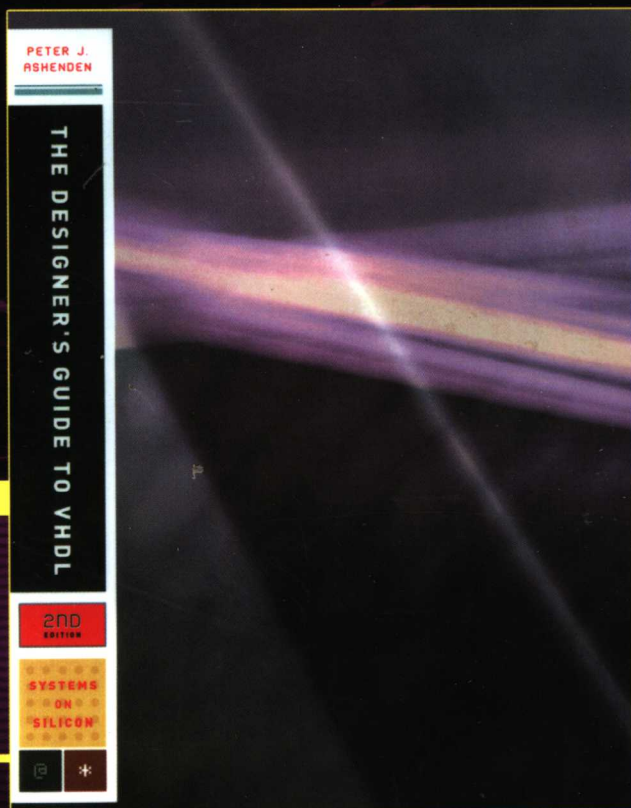
VHDL 设计指南

(原书第2版)

The Designer's Guide to VHDL

(Second Edition)

(美) Peter J. Ashenden 著
葛红 黄河 吴继明 译



机械工业出版社
China Machine Press



电子与电气工程丛书

VHDL 设计指南

(原书第2版)

The Designer's Guide to VHDL (Second Edition)

(美) Peter J. Ashenden 著

葛红 黄河 吴继明 译



机械工业出版社
China Machine Press

本书系统地介绍标准VHDL语言及其使用方法,包括系统建模工具在数字系统设计中的应用方法。本书从简单的模型问题入手,然后逐步深入,引向更复杂的实例研究。书中包括四个完整的实例研究,将帮助读者综合掌握VHDL技术。章后安排了精心设计的难易程度不同的练习,以及部分答案,有利于读者掌握书中的概念,方便教师进行教学参考。

本书适合作为高等院校通信、电子、计算机专业相关课程的教材或参考书,也适合希望熟练掌握VHDL技巧的技术人员参考。

Peter J. Ashenden: The Designer's Guide to VHDL, Second Edition (ISBN 1-55860-674-2).

Copyright © 2002 by Elsevier Science (USA).

Translation Copyright © 2005 by China Machine Press.

All rights reserved.

本书中文简体字版由美国Elsevier Science公司授权机械工业出版社独家出版。未经出版者书面许可,不得以任何方式复制或抄袭本书内容。

版权所有,侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号: 图字: 01-2003-6715

图书在版编目(CIP)数据

VHDL设计指南(原书第2版)/(美)阿森顿(Ashenden, P. J.)著;葛红等译.
—北京:机械工业出版社,2005.6

(电子与电气工程丛书)

书名原文: The Designer's Guide to VHDL, Second Edition

ISBN 7-111-16216-1

I. V… II. ①阿… ②葛… 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(2005)第016337号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑: 吴怡 许萍

北京瑞德印刷有限公司印刷·新华书店北京发行所发行

2005年6月第1版第1次印刷

787mm×1092mm 1/16·36.75印张

印数: 0 001-4 000册

定价: 69.00元

凡购本书,如有倒页、脱页、缺页,由本社发行部调换

本社购书热线:(010) 68326294

译者序

近年来,随着计算机技术和半导体技术的发展,电子数字系统的应用变得非常普遍。同时,数字系统本身的复杂程度也在不断提高。随着市场竞争的加剧,对电子产品的价格、性能和推出市场的时间的要求也越来越苛刻。传统的硬件电路设计方法已大大落后于当今技术的发展。采用硬件描述语言的设计方法已成为当今以及未来电子设计自动化(EDA)解决方案的核心,特别是对于深亚微米复杂数字系统的设计,硬件描述语言具有独特的作用。

目前,广泛使用的国际标准硬件描述语言有VHDL (Very High Speed Integrated Circuit Hardware Description Language) 和Verilog HDL。使用VHDL硬件描述语言,可以用仿真器的方式测试所设计的电路的功能是否正确,直接完成以前需要建造好样机才能进行的电子电路特性的测量和调试。从而能够大大地减少设计成本和缩短设计时间。

本书全面、系统地介绍了国际标准的硬件描述语言VHDL以及VHDL在现代数字电子系统设计中的应用,对VHDL和基于VHDL的数字电子电路设计中的有关问题进行了深入细致地讲解,并结合理论分析了大量实例,使本书兼具知识性和实用性。本书按照一种渐变发展的方式组织内容,深入浅出。本书还结合VHDL语言所提供的各种建模工具,详细介绍了这些工具在数字系统设计中的应用。既能使读者对有关VHDL的知识有一个全面了解,同时又能较好地掌握VHDL语言在不同设计层次中的应用方法。

与一般VHDL的书籍不同,本书并不要求读者具有编程语言的经验,对第一次接触VHDL的初学者来说也是一本理想的教材。而对于那些需要熟练使用VHDL的设计人员而言,本书也可以作为一个很好的指南和手册。本书还配备了大量针对性很强的习题。

本书非常适合作为大专院校电子工程和计算机类高年级本科生和研究生学习VHDL语言的教科书和参考书,也可以为广大从事数字电路设计的工程技术人员提供相关的技术参考。

参加本书翻译工作的人员有:黄河(序、第1版序、前言、目录、和第1、2、3、4、6、7、8章),葛红(第17、18、19、20、21章和附录A、B、C)和吴继明(第5、9、10、11、12、13、14、15、16章)。

限于译者水平,本书的翻译难免会存在不少错误,恳请广大读者批评指正。

黄河

华南师范大学计算机学院

序

Paul Menchini

Menchini & Associates, Durham, North Carolina

自从本书第1版出版以来的五年时间，对VHDL原创的设计者们的压力越来越大。电子产品越来越普遍，但对电子产品的价格、性能和推出市场的时间的要求也越来越苛刻。在医疗和航天系统这些对准确性和可靠性要求极为高的领域，对数字电子控制的依赖越来越大。新型的更大规模的现场可编程器件已经开辟了一个全新的应用领域，这些应用也带来了新的需要学习VHDL的设计师群体。与此同时，随着产品的各种新的和改进版本以及同类竞争产品越来越快地进入市场，各种设计的商业寿命急剧缩短。

为了应对这些新的变化，VHDL已经做了许多方面的改进。尽管基本语言框架保持相对的稳定，只是对VHDL-93的共享变量进行了重写以及对一些错误进行了纠正等，新的VHDL-2001并没有重大更新，但是在相关的领域已经进行了大量的工作。

VHDL在基本语言的框架上开发了大量新的集合包和标准：numeric_bit、numeric_std、math_real、math_complex和vital。新开发的VHDL-AMS，一个VHDL的扩展集，使得VHDL可以应用于模拟和混合信号建模的领域。一级标准的综合子集已经做了规定和定义，扩充这些定义以增加更多功能的各种工作正在进行之中。这些新增加的内容使得VHDL在满足硅系统的建模要求上具有独特的能力。

在本书第2版中，Ashenden博士在第1版的基础上化了大量的精力来处理这些变化（VHDL-AMS除外）。他详尽描述了自从第1版发行以来，IEEE组织采用的新的集合包，2000年采纳的共享变量的新的实现方法，以及VHDL-2001中新的变化（现在正在批准阶段）。附录中有关综合部分的内容已经按照IEEE标准1076.6（标准综合子集描述）进行了完全的重写。

另外，DLX处理器的实例研究部分根据当前设计实践进行了重新修订。这样，可以更加容易地得到处理器模型的综合版本。

本书为VHDL的教材设立了一个全新标准。将它放在手边作为参考书肯定是非常有价值的。

第1版序

Paul Menchini

Research Triangle Park, North Carolina

电子数字系统的复杂性随着时间呈指数形式增加。事实上，产品寿命周期的不断降低和产品可靠性要求的不断提高，迫使电子产品的设计人员需要极大地提高产品质量和产出率。

VHDL的开发就是应顺了这种趋势。借助于软件工程领域的复杂性管理和错误检测技术，VHDL可以消除各种无关的细节，采用与工艺无关的描述，及早发现错误，增强了从门级到系统级模型之间的可移植性和共用性。

为此，电子设计自动化（EDA）工具已经在设计人员的工作中占据了从未有过的份额。现在，利用单个的表示媒介工具就可以完成从规格说明到细节数字设计的全部设计过程。

VHDL起初只是美国政府国防部开发的标准硬件描述语言（HDL），在经过两次IEEE标准化进程后，VHDL已经变成了世界上通用的最好的两个硬件描述语言之一。随着VHDL在细节定时模型、综合指示和模拟能力等方面的进一步发展，VHDL将继续为下一世纪电子产品的先进设计提供所需要的表述工具。得益于VHDL精确定义的各种设计工具会不断出现，提供更加强大的仿真性能，综合能力，甚至会增加一些全新的能力，比如说模型功能等效性的形式验证等。

由于VHDL使用了许多在常规硬件描述中所没有的概念，因此第一次接触VHDL的设计人员在应用这些概念时，需要一个可靠的指南来说明这些特征。现在有关VHDL的书籍都太强调读者具有编程语言的经验，而本书对那些非程序员来说是学习VHDL的理想教材。

本书详细探究了VHDL的最新版本VHDL-93（IEEE Std. 1076-1993）。假设在此之前你没有任何VHDL的知识，Ashenden教授会指引你游历VHDL王国，首先处理简单的模型问题，然后逐步深入，引向更复杂的研究。因为VHDL包含大量的特性，书中的四个完整的工作实例研究将给读者以很大的帮助。这些实例研究包含了VHDL的各种不同特性和功能，并且根据不同复杂程度的数字系统的建模来说明这些特性和功能如何作用。

附录总结了VHDL在设计综合中的应用，以及建筑在VHDL框架上的其他标准句法，给出了与VHDL最初的IEEE标准（IEEE Std. 1076-1987）的差异列表，这对于有经验的VHDL用户也是很有帮助的。

欢迎进入VHDL世界。

前言

VHDL是一种描述数字电子系统的语言，它源自美国政府的甚高速集成电路（VHSIC）计划。随着计划的进行，人们发现迫切地需要一种标准语言来描述集成电路（IC）的结构和功能。因此，VHSIC硬件描述语言（VHDL）就应运而生。在国际电气与电子工程师（IEEE）协会的支持下，VHDL得到了进一步的发展，并在1987年以“标准VHDL语言参考手册”的形式，被正式颁布为IEEE标准1076。VHDL语言的第一个标准版本通常被称为VHDL-87。

像所有其他的IEEE标准一样，VHDL标准每五年审阅更新一次。负责VHDL标准的IEEE工作小组分析研究来自各个VHDL-87用户的意见和建议，并于1992年提出了修订后的版本。这个版本在1993年被采纳，人们称为VHDL-93。第二轮对标准的修订始于1998年，并在2001年完成。本书描述的就是该语言现在的版本VHDL-2001。

VHDL语言是为满足设计过程中的各种需要而设计的。第一，它可以对系统的结构进行描述，即一个系统怎样分解成为子系统，这些子系统之间是怎样相互连接的。第二，它可以使用熟悉的编程语言的形式来对系统的功能做详尽的说明。第三，作为结果，它可以先于实际制造前，对系统的设计进行仿真。这样一来，设计人员可以很快地比较各种替代方案，可以在毫不延时和不需要花费硬件样机费用情况下，直接测试方案的正确性。第四，它可以实现从一个较抽象的系统说明综合成一个设计的细节结构。这样，设计人员可以专注于一些更全局更战略性的设计决策，缩短产品推向市场的时间。

本书详细介绍了VHDL语言所提供的建模工具及其在数字系统设计中的应用方法。本书的宗旨并不是讲授数字设计，因为数字设计本身是一个需要好几本教科书才能覆盖其各个方面的大课题。本书假定读者已经掌握了数字设计的基本概念，例如，学过工程专业的有关数字设计的初级课程。当然，了解计算机编程和计算机组织结构的基本知识对学习本书的内容也是非常有帮助的。本书适合作为数字设计或计算机设计中的入门或中级课程。对于那些需要熟练掌握VHDL技巧的工程师而言，本书也可以作为一本有用的参考书。

建模仿真和综合是大规模系统设计方法中非常重要的部分。VHDL可以在一个抽象层次的范围内表示模型，包括从门级向上到算法级和结构级的多个层次，它将在未来的一段时间内，在基于硅系统的设计方面扮演重要的角色。

本书的结构

本书是按照从头至尾顺序阅读的方式组织的。这种途径提供了一种渐变的发展过程，每章都会涉及前面章节介绍的概念。每章都引进一些相关的概念或语言工具，并举若干例子来进行阐述。书中共有4个实例研究章，将前面的相关内容通过综合工作实例的形式结合到一起。

第1章介绍硬件描述语言的思想，并勾画了其使用的种种理由和由此带来的好处。接着介绍了VHDL的一些基本概念，奠定了后续章节中各个例证的基础。接下来的三章主要讲述VHDL中与传统编程语言最为相似的一些基本内容，这些内容涉及采用算法术语描述系统的

行为。第2章解释了语言的基本类型体系并引进了标量数据类型。第3章描述了顺序控制结构，而第4章则介绍了表示数据元素集合的复合数据结构。

第5章详细介绍了VHDL中建模硬件的主要工具和功能设施，包括设计中对各种基本行为单元建模的工具，和连接各基本单元的信号，以及设计的层次结构。如果将前面各章节介绍的各种功能设施组合在一起的话，可以完成许多建模任务。因此，第6章是第一个综合性的实例研究，设计了一个乘法累加器电路。

接下来的章节组进一步扩充了VHDL语言的基本特性，使得设计人员在处理大系统时更加得心应手。第7章介绍过程和函数，可以将设计中的某些行为内容包装成一些独立的功能段落。第8章介绍了包的概念，利用包将设计的相关部分组合在一起，生成一些可以被多个设计使用的模块。同时，还介绍了一系列VHDL设计中常用的IEEE标准包。第9章是有关别名的内容，主要针对大规模模型中有可能出现的大量命名的管理问题。这个章节组的最后一章是第二个实例研究，设计了二进制算术操作的集合包。

第三个章节组则主要讨论VHDL中的高级建模功能。第11章处理决断信号这一重要课题，第12章描述了类属常数，是一种对设计中的结构和行为进行参数化处理的方法。虽然这部分内容是许多真实模型的基础，但在本书中还是安排在较后的章节中。经验已经证实，没有坚实的硬件描述语言知识的基础，要理解这些概念是比较困难的。第13章主要讲元件的实例化和配置问题。这些功能特性对于大型真实模型也是十分重要的，但较难理解。因此，本书在前面的章节中，通过直接例化的方式进行结构建模，而把元件实例化和配置这些更普遍的情况留到了这一章来讲解。第14章介绍了一些生成的常规结构。第15章则是第三个实例研究，描述了一个CPU的寄存器传输层模型。

第四个章节组主要讨论系统级建模中使用的语言工具。第16章详细处理了有关保护信号和块的内容。第17章介绍了访问类型标记（或指针）的表示方法，并利用它们建立链接数据结构。这就涉及了有关抽象数据类型的讨论，处理与链接数据结构相关的复杂性问题。第18章讨论使用文件输入和输出的语言工具，包括二进制文件和文本文件。第19章则是第四个实例研究，建立一个队列网络的计算机系统模型。

最后两章将剩下的一些语言工具归并，第20章描述了属性机制，对设计加入一些附加信息的注解。第21章介绍其他一些前面未涉及的高级课题。

虽然通篇地阅读本书，可以对VHDL语言有一个完整的了解，但是也可以按不同顺序有选择地阅读。下面针对不同读者，给出了一些阅读和教学的建议。

- 对于VHDL数字系统建模的初级课程：建议学习第1章到第5章，第7、8、11和12章，再加上第6章的实例研究。如果时间允许，可以增加第9章和第10章的内容。
- 对于一个较高级的课程：建议再增加第13章和第14章的内容，如果时间许可，也应把第15章的实例研究包括在内。
- 如果读者已经熟悉常规的编程语言：可以将第2章到第4章作为一个复习。
- 如果读者已有一些VHDL的入门级的背景：将第1章至第4章作为一个复习。

每章结尾都设计了一定数量的练习题，可以帮助读者增强对内容的理解。每道练习题使用一个方括号标明该练习题所对应的本章的特定主题的章节数目，同时，加入了一个粗略的难度等级，具体含义如下：

- ① 问答题，测验基本概念。
- ② 基本建模练习——10分钟至半小时的工作。

③ 高级建模练习——1小时到2小时的作业。

④ 建模项目——需要半天或更多时间。

以上第一类型的练习题的答案列于附录G。其他类型的练习涉及到开发VHDL模型。建议读者利用VHDL仿真器执行自己书写的模型，以验证这些建模练习的正确性。对于那些只是比较纸上的模型和答案来说，这是一种更为有效的练习方法。

贯穿于本书内容的一个基本思想就是，使用一个硬件描述语言建模本质上是一个软件设计练习。这就意味着应该应用好的软件工程惯例，因此本书的处理方法是直接从软件工程的经验中提取出来的。本书中呈现的大量提示和技巧都来自于小规模 and 大规模软件工程，真诚地希望这些经验对读者的学习有用处。

第2版的一些改动

本书的第1版出版于1995年，正值VHDL-93得到越来越普遍的接受。此版的更新反映了VHDL-2001的一些变化。语言标准中的许多改变纠正了前面标准中存在的一些模糊性问题，这些问题容易造成不同供应商的VHDL工具之间的不兼容性。同样，在增强语言的应用性方面也做了一些改进。书的正文和例子都随之进行了修订以适应这种语言上的改动。而且，自从本书的第1版出版之后，又新公布了一系列的与VHDL相关的标准并取得了广泛的认同。第8章扩充了一个包括IEEE标准综合和数学包的描述，附录A完整地重写了IEEE综合协同性标准。根据这些发展，第15章中DLX实例研究进行了较大的修订。附录D也进行了重大修订和扩充，以反映在当前EDA（Electronic Design Automation，电子设计自动化）领域使用的大量的VHDL相关标准。

求助和信息资源

尽管本书试图全面地反映VHDL的内容，但无疑还有许多问题是本书没有涉及和不能回答的。为此，读者需要寻找其他资料。这里有一个常常被忽视的，但十分有价值的经验和忠告，

在工作地点或在用户小组中，向你身边的同事求助。用户小组通常会举行定期的正式或非正式的会议，其中包括一个提问和答疑时间。许多用户小组还在网络上通过电子邮件和在线讨论的方式解决问题。在全世界有许多VHDL用户小组，其中很多小组是在VHDL国际组织（www.vhdl.org）的支持下运作。该国际组织针面向用户的部分是VHDL国际组织论坛（www.vhdl.org/viuf/）。该组织支持与VHDL相关的会议、设计比赛和其他活动。

VHDL国际组织是多个支持EDA行业工作团体网络服务器（www.eda.org）的组织之一。这个服务器连接到多个VHDL标准工作小组和用户小组的网页和知识库。

对于访问Usenet电子新闻网的读者来说，你会发现新闻组comp.lang.vhdl是一个非常有用的资源。这个新闻小组的资源包括通知、样板模型、问题和答案，以及有用的软件等。小组的参加人员包括VHDL小组的用户以及那些积极参加语言标准工作小组和从事VHDL工具开发的人员。对该小组来说，“经常询问的问题”（FAQ）文件是一个对有关书籍、产品和其他信息的有用的指针资源。这些文件存档于www.eda.org。

一个值得一提的资源是IEEE 1076标准，IEEE标准VHDL语言参考手册（IEEE Standard VHDL Language Reference Manual），有时候它被称为“VHDL的圣经”。它是关于VHDL权威

的信息资料。然而，因为它是一种定义式的文件，而不是教材，所以它是以一种复杂条文的形式书写的。这使得它非常难回答那些在书写VHDL模型时产生的问题。因此，只有当你对VHDL比较熟悉后，才建议使用它，可以从IEEE的网站standards.ieee.org直接订购。

本书含有大量VHDL模型的例子，它们都可以用作求解问题的资源。这些例子和实例研究的VHDL源代码，还有其他相关信息，都可以在线访问以下网址：

www.mkp.com/vhdl2e

尽管我已经非常仔细地避免在例子代码中出现错误，但是难免会存在一些错误。我将非常愿意听到读者对书中和源代码中的错误的反馈，这样我可以从在线代码和未来本书的重印中纠正这些错误。勘误表和一般的评论可以通过以下电子邮件发给我：

peter@ashenden.com.au

致谢

本书的写作源泉可以追溯到1990年，当时我在澳大利亚阿德雷德大学为我所教授的计算机结构课程制作了一套简短的笔记稿，VHDL使用手册（Cookbook）。在当时，有关VHDL的书籍很少，所以我把自己的小册子放到了网上供大家浏览。这份小册子可以浏览的消息很快在世界范围内传播，几天时间内，我的电子邮件信箱就都被来信挤爆了。在10年后，编写本书的时间，还不时收到有关那本使用手册的消息。许多读者都催促我编写一本完整的教材版本的书籍。借助于大家的鼓励，我开始尝试写作，终于出版了本书的第1版。我非常感谢世界各地的许许多多的工程师、教师和学生，他们给了我那样的推动力，并让本书的第1版是那样的成功。我希望这个第2版可以继续满足读者们对全面了解VHDL的需求。

在本书第1版中，我有机会对许多帮助我写作这本书的人表示了感谢。他们包括我在阿德雷德大学的同事；研究合作伙伴Phil Wilsey和他在美国辛辛那提大学的学生；Morgan Kaufmann出版社的工作人员，特别是Bruce Spatz 和Cheri Palmer；本书第1版手稿的评审人，即得克萨斯大学的Poras Balsara，Menchini的Paul Menchini及其合作伙伴，GTE实验室和Lowell大学的David Pitts，辛辛那提大学的Philip Wisley；David Bishop在综合部分贡献了资料，以及Model Technology公司使用他们的V系统分析器和仿真器来测试例证模型。我还要感谢那些对本书第1版作出有价值贡献的所有人和组织。

从那以后，我很幸运地继续与许多人一道工作，才有了今天的第2版。我向我的合作研究者们和EDA标准领域的同事们表达我诚挚的谢意，感谢他们的努力形成了一个活生生的智力环境，让VHDL在这里扮演了重要的角色。太多的人需要感谢，在这里难以一一列出他们的姓名。但是，我必须单独致谢Phil Wilsey的大力支持和一起共饮美妙咖啡的畅谈，感谢Perry Alexander在出外旅行期间让我占有他的私人办公室来编辑我的手稿。我继续发展着我和Morgan Kaufmann出版社的工作人员的良好的合作关系，特别是高级编辑Denise Penrose。本书得以高质量的出版归于他们的辛勤努力。

我把本书的第1版献给我的妻子Katrina。正如我在第1版前言中所说，过去我总以为作者们将所著的书籍献给他们伴侣是有些做作，但是Katrina的理解、鼓励和支持告诉我并非如此。我在此仍旧要向我的妻子表示深深的谢意，感谢她持续不断的支持。并且，我仍旧非常荣幸地将这本书第2版献给她。

目 录

译者序

序

第1版序

前言

第1章 基本概念1

1.1 数字系统建模1

1.2 建模的域和层次3

1.3 建模语言6

1.4 VHDL的建模概念6

1.4.1 行为元素7

1.4.2 结构元素8

1.4.3 混合结构和行为模型9

1.4.4 测试台10

1.4.5 分析、加工和执行11

1.5 学习一种新语言: 词汇元素和句法13

1.5.1 词汇元素14

1.5.2 句法描述18

练习20

第2章 标量数据类型和操作23

2.1 常数和变量23

2.1.1 常数和变量声明23

2.1.2 变量赋值24

2.2 标量类型25

2.2.1 类型声明25

2.2.2 整数类型26

2.2.3 浮点类型28

2.2.4 物理类型29

2.2.5 枚举类型31

2.3 类型分类36

2.3.1 子类型37

2.3.2 类型限定38

2.3.3 类型转换38

2.4 标量类型的属性39

2.5 表达式和操作符41

练习43

第3章 顺序语句45

3.1 if 语句45

3.2 case 语句47

3.3 null语句52

3.4 loop 语句52

3.4.1 exit 语句53

3.4.2 next 语句56

3.4.3 while 循环57

3.4.4 for 循环58

3.4.5 loop 语句小结60

3.5 断言和报告语句61

练习65

第4章 复合数据类型和操作67

4.1 数组67

4.1.1 多维数组69

4.1.2 数组聚集70

4.1.3 数组属性72

4.2 非约束数组类型74

4.2.1 字符串75

4.2.2 位矢量75

4.2.3 标准逻辑数组75

4.2.4 字符串和位串文字75

4.2.5 非约束数组端口76

4.3 数组操作和引用77

4.3.1 数组片78

4.3.2 数组类型转换79

4.4 记录80

练习83

第5章 基本建模结构85

5.1 实体声明85

5.2 构造体	87	7.4.1 功能建模	171
5.2.1 并发语句	87	7.4.2 纯函数和非纯函数	171
5.2.2 信号声明	88	7.4.3 now函数	172
5.3 行为描述	89	7.5 重载	173
5.3.1 信号赋值	89	7.6 声明的可见性	176
5.3.2 信号属性	91	练习	179
5.3.3 wait语句	93	第8章 包和use子句	183
5.3.4 delta 延迟	96	8.1 包声明	183
5.3.5 传输和惯性延迟机制	98	8.1.1 包声明中的子程序	186
5.3.6 进程语句	102	8.1.2 包声明中的常数	187
5.3.7 并发信号赋值语句	103	8.2 包体	188
5.3.8 并发断言语句	108	8.3 use 子句	190
5.3.9 实体与被动进程	109	8.4 预定义包标准	192
5.4 结构化描述	110	8.5 IEEE标准包	193
5.5 设计加工	117	8.5.1 std_logic_1164多值逻辑系统	193
5.5.1 分析	117	8.5.2 标准VHDL综合包	194
5.5.2 设计库、库子句和use子句	118	8.5.3 标准VHDL数学包	198
5.5.3 加工	120	练习	202
5.5.4 执行	122	第9章 别名	205
练习	123	9.1 数据对象的别名	205
第6章 实例研究：一个流水线乘法累加器	131	9.2 非数据项的别名	208
6.1 算法概要	131	练习	210
6.2 行为模型	133	第10章 实例研究：一个位矢量算术包	213
6.3 寄存器传输级模型	140	10.1 包接口	213
6.3.1 寄存器传输级模型中的功能模块	141	10.2 包体	216
6.3.2 寄存器传输级的构造体	147	10.3 一个使用算术包的ALU	223
6.3.3 测试寄存器传输级模型	150	练习	224
练习	151	第11章 决断信号	227
第7章 子程序	153	11.1 基本决断信号	227
7.1 过程	153	11.1.1 复合决断子类型	231
7.2 过程参数	158	11.1.2 决断子类型小结	234
7.2.1 信号参数	161	11.2 IEEE std_logic_1164决断子类型	234
7.2.2 默认值	163	11.3 决断信号和端口	236
7.2.3 非约束数组参数	164	11.3.1 决断端口	237
7.2.4 过程参数小结	166	11.3.2 驱动值属性	238
7.3 并发过程调用语句	167	11.4 决断信号参数	239
7.4 函数	169	练习	240

第12章 类属常数	245	15.4.1 算术逻辑单元	327
12.1 参数化行为	245	15.4.2 寄存器	330
12.2 参数化结构	248	15.4.3 寄存器文件	333
练习	249	15.4.4 多路复用器	334
第13章 元件和配置	251	15.4.5 扩展器	335
13.1 元件	251	15.4.6 构造体	336
13.1.1 元件声明	251	15.4.7 控制器	340
13.1.2 元件实例生成	252	15.4.8 配置声明	354
13.1.3 组装元件	253	15.5 测试寄存器传输级模型	355
13.2 配置元件实例	255	练习	359
13.2.1 基本配置声明	255	第16章 保护和块	361
13.2.2 配置多层次的结构	256	16.1 保护信号和断开	361
13.2.3 配置实体的直接实例生成	259	16.1.1 驱动属性	365
13.2.4 配置中的类属映射和 端口映射	260	16.1.2 保护端口	365
13.2.5 延缓的元件绑定	264	16.1.3 保护信号参数	366
13.3 配置说明	266	16.2 块和保护信号赋值	368
练习	271	16.2.1 显式保护信号	370
第14章 生成语句	275	16.2.2 断开说明	372
14.1 生成迭代结构	275	16.3 使用块的结构化模型	373
14.2 条件生成结构	279	16.3.1 块中的类属和端口	375
14.3 生成语句的配置	284	16.3.2 配置带有块的设计	376
练习	288	练习	378
第15章 实例研究: DLX计算机 系统	293	第17章 访问类型和抽象数据类型	381
15.1 DLX CPU综述	293	17.1 访问类型	381
15.1.1 DLX寄存器	293	17.1.1 访问类型声明和分配器	381
15.1.2 DLX指令集	294	17.1.2 访问值的赋值语句和等式	383
15.1.3 DLX 外部接口	297	17.1.3 记录和数组的访问类型	384
15.2 行为模型	298	17.2 链接数据结构	386
15.2.1 DLX类型包	299	17.3 使用程序包的抽象数据类型	390
15.2.2 DLX 实体声明	300	练习	400
15.2.3 DLX 指令集包	301	第18章 文件及其输入和输出	403
15.2.4 DLX 行为构造体	307	18.1 文件	403
15.3 测试行为模型	319	18.1.1 文件声明	403
15.3.1 测试台时钟发生器	319	18.1.2 读文件	404
15.3.2 测试台存储器	320	18.1.3 写文件	407
15.3.3 测试台构造体和配置	324	18.1.4 子程序中的文件声明	409
15.4 寄存器传输级模型	326	18.1.5 直接打开和关闭操作	410
		18.1.6 子程序中的文件参数	412
		18.1.7 文件的可移植性	414

18.2 程序包textio	414	20.1.4 已命名项目的属性	460
18.2.1 textio读操作	416	20.2 用户定义的属性	466
18.2.2 textio写操作	423	20.2.1 属性声明	467
18.2.3 用户自定义类型的读写	425	20.2.2 属性说明	467
练习	426	20.2.3 外来属性	475
第19章 实例研究: 队列网络	429	20.3 组	476
19.1 队列网络概念	429	练习	478
19.2 队列网络模块	430	第21章 其他主题	481
19.2.1 随机数发生器	430	21.1 缓冲和连接端口	481
19.2.2 令牌和arc类型的程序包	433	21.2 关联列表中的转换函数	482
19.2.3 令牌源模块	435	21.3 延迟进程	487
19.2.4 令牌池模块	438	21.4 共享变量	489
19.2.5 队列模块	439	练习	497
19.2.6 令牌服务器模块	444	附录A 综合	499
19.2.7 分叉模块	447	附录B 预定义程序包标准	513
19.2.8 结合模块	449	附录C IEEE标准程序包	517
19.3 磁盘系统的队列网络	452	附录D 相关标准	527
练习	457	附录E VHDL句法	537
第20章 属性和组	459	附录F VHDL-87、VHDL-93和VHDL-2001 之间的区别	549
20.1 预定义属性	459	附录G 练习答案	553
20.1.1 标量类型的属性	459	参考文献	571
20.1.2 数组类型和对象的属性	460		
20.1.3 信号属性	460		

第1章 基本概念

本章将介绍什么是数字系统建模，以及为什么建模和仿真是设计过程中十分重要的一个环节。读者将会看到怎样使用硬件描述语言（VHDL）来为数字系统设计建模，同时还将引入与VHDL语言有关的一些基本概念。学习完本章，读者将了解VHDL语言的基本词汇和句法结构，为进一步学习VHDL语言的各种特性打好基础。

1.1 数字系统建模

要讨论数字系统设计建模，首先需要了解什么是数字系统。对此，不同的工程师们有不同的定义，这与他们不同的工作背景和工作领域有关。有些人可能会把一个大规模集成电路（VLSI）作为一个独立的数字系统，而其他一些人则认为数字系统应该更大，包括一个完整的计算机和各种外围控制设备以及接口等。

本书的目标是将任何涉及到处理和储存信息的数字电路都当作一个数字系统。这样一来，我们考虑的不单是系统的整体部分，而且还要包括构成系统的各个不同部件。因此，本书有关数字系统的内容覆盖了从低层门电路构成的元件到顶层的功能模块元素这样一个广泛领域。

如果要在这样一个层次上讨论数字系统，我们必须意识到将要面临的数字系统的复杂性。要完全靠人力去理解和处理这些复杂的系统是不可能的，我们需要寻找处理这些复杂问题的方法。依靠这些技术和方法，我们就可以有一定程度的自信去设计满足各种需求的功能部件和数字系统。

应付这些挑战的一个最重要的方法是采用数字设计的系统方法论。如果从一个系统的需求文件着手，我们可以先设计出一个满足这些需求的抽象结构。然后把这个抽象结构分解成一系列的部件，将它们组合在一起实现和抽象系统相同的功能。而其中每一个部件也都可以继续分解到一些最基本的、现成的元件层次，去完成所要求的功能。这种处理方法就是从最基本元素开始分层构建一个数字系统。

这种系统方法论的优点是可以独立地设计每一个子系统。在使用一个子系统时，只需要把它当作一个抽象结构，而无需考虑其组成细节。因此，在设计过程的任一特殊环节上，我们只需关注与目前设计相关的少量信息，而从大量繁重的细节工作中解放出来。

我们可以使用所谓模型（model）这一概念来理解数字系统。一个模型代表了一些和当前设计层次相关的信息，并且模型可以从那些与当前设计无关的细节中抽象出来。然而，采用这种方法也意味着相同的一个系统可能存在几个模型，这是因为在不同的内容中抽取的相关信息是不同的。某种模型可能关注的是表征系统的功能，而另一种模型则可能是表示由子系统组成系统的方式。在接下来的章节中，我们将详细讨论这些细节问题。

采用模型这种思想有许多重要的理由。首先，当人们需要使用一个数字系统时，必须指

定对该数字系统的各种要求。设计工程师们的工作就是去设计满足这些要求的数字系统。为此,工程师们首先必须理解这些要求,并能够在某种程度上撇开这些要求去寻找一些替代的实现方法或者按照某些标准选择最好的实现方式。在这个过程中,往往会产生一个问题,即这些要求的表述常常是不完整和模糊的,这样就造成用户和设计工程师们之间难以对需求文件达成一致的看法。人们可以通过采用一个形式上的模型的方法来沟通对这些要求的看法,以避免上面出现的这个问题。

第二个理由是采用形式上的模型可以与用户沟通对系统功能的理解。设计师们不可能总能预测出一个系统的所有可能的使用方式,因此也不可能列举出系统的所有可能的行为。但是如果设计师提供了一个模型,则用户就可以根据任意给定的输入状态来检验这个系统模型,并确定系统在这种输入环境下的行为状态。因此,一个形式上的模型在对一个系统进行验证和文件归档时是一个非常有价值的工具。

第三个理由是模型化使得设计师们可以利用仿真来测试和验证一个数字设计。如果从一个定义了系统行为的需求模型开始,采用测试输入和记录相关输出的方法能仿真系统的行为。按照设计方法论,接下来就可以把系统分解为子系统,设计有关子系统的电路,每个子系统都有自己的行为模型。同样,可以对这种合成系统进行仿真,在输入相同时,将输出结果与前面的系统仿真结果进行比较。如果这些仿真结果是一致的,那么就on知道合成系统也满足这一测试环境下的各种要求。否则就需要对设计进行某些修正和改进。我们可以继续这样的过程直到其设计层次的最底层,这时候所有的部件都是一些其行为特性为大家所熟悉的真实器件。接下来,当对所设计的产品进行制造时,这些来源于仿真的输入和输出结果可以被用来验证物理线路的功能是否操作正确。设计过程中采用的测试手段和验证过程都假定测试中的输入量已经包含了最终实际电路应用中的各种可能情况。这种测试的覆盖率问题本身就是一个非常复杂的问题,它是数字系统设计研究的一个活跃领域。

建立模型的第四个理由是对一个数字系统设计的正确性提供了形式上的验证。形式验证需要有一个系统需求功能的数学陈述。这种数学陈述可以采用一个形式逻辑系统的表示方法表示出来,比如说一个暂存逻辑电路。形式验证同样需要对建模语言或用来描述一个数字设计的表示方法的意义进行数学定义。验证的过程也包括了逻辑系统推理规则的应用,以证实这个数字设计是否具备了所要求的各种功能。形式验证到目前为止,还没有得到广泛的应用,它也是数字系统设计研究的活跃领域之一。形式验证技术已经在真实的设计项中得到了重要的示范,其未来的前景是十分光明的。

最后一个同样重要的理由是建模可以实现设计线路的自动综合合成。如果可以在形式上指定一个系统的需求功能,那么理论上就可以把这些特性说明转化成能够实现这些功能的电子线路。这种方法的优点是可以大量节省人力成本,设计工程师们可以自由地寻找各种各样的替代选择方案,而不必陷入繁琐的设计细节的困境。同样,建模这种方法也可以使得在一个数字系统设计中导入错误和错误未被检测出来的机会大大减少。如果可以自动地完成从特性说明到电路实现的过程,那么我们也会对设计出来的电路的正确性更加充满信心。

当然所有这些讨论的最终意义是希望在设计过程中可以用最小的成本和最短的时间获得最大限度的可靠性。我们需要保证对数字系统的各种需求能够被清晰详尽地表述出来和得到很好地理解,并确保可以正确地使用各种子系统且使设计结果满足要求。如果不得不在产品制造后再校正设计上的错误,那只会增加额外的成本费用。为了减少错误和为设计过程提供

更好的工具，一些成本上的费用和时间上的延误是可以理解的。

1.2 建模的域和层次

在前面的部分已经提到了由于关注点的不同，一个数字系统可以有不同的模型。可以将这些模型归纳为三个描述域：功能域（function）、结构域（structure）和物理域（geometry）。功能域描述的是系统的操作和实现的功能。在某种意义上，功能域是最抽象的描述域，因为它不涉及到这些功能是怎么实现的。结构域处理的是系统怎样由内部互连的子系统组成的。而物理域表述的是系统在物理空间的布局 and 具体的几何实现。

每一个域还可以被分成不同的抽象层次。在最高层次上，我们仅仅考虑对功能域、结构域和物理域总的宏观的描述，接下来在较低的层次上逐渐将系统的描述细化和具体化。图1-1（由Gajski和Kuhn设计，参见参考文献[6]）利用三个独立的轴表示这三个不同的描述域，而利用与这些轴相交的同心环代表不同的设计层次。

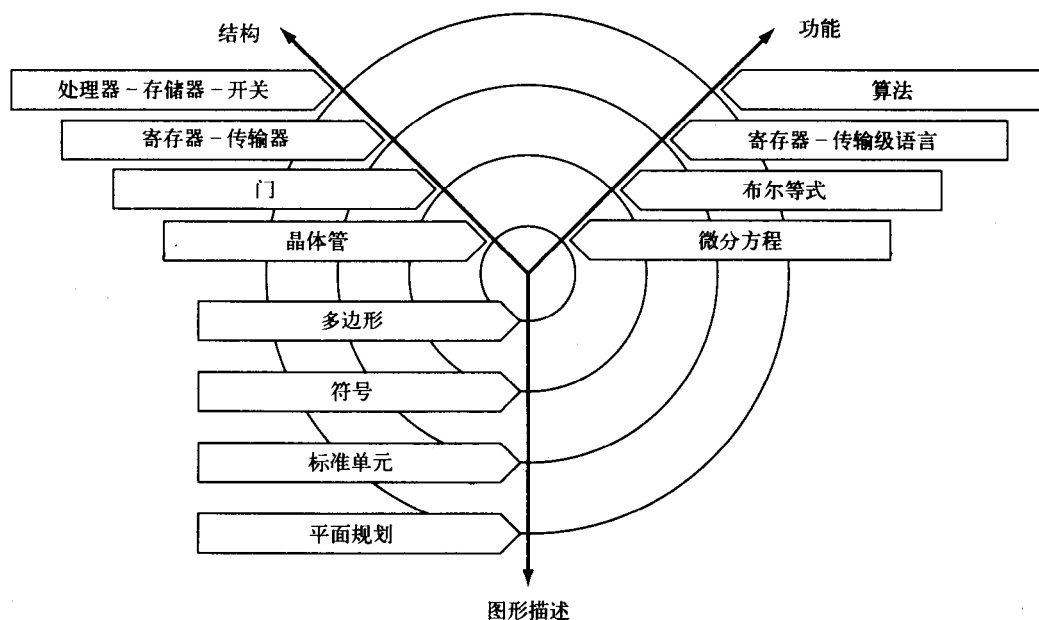


图1-1 描述域和抽象层次。辐射轴表示三个不同的建模描述域，同心环表示抽象层次，越外圈越抽象，越靠近中心越详细

下面来详细讨论这种分类方法，看一看在不同描述域的每个不同层次是怎样来建立模型的。下面举一个例子，考虑一个应用在某些测量仪器中作为控制器的单片微处理器，微处理器具有数据输入连接和某种形式的显示输出。

在最抽象的层次上，整个系统可以按照一种算法来描述，非常类似计算机程序的算法。功能建模的这一层次通常称为行为建模（behavioral modeling），在对系统的功能进行抽象描述时将采用这个术语。图1-2给出了所讨论的仪器控制器的一个可能的算法。这个模型描写了控制器怎样重复扫描每一个输入数据，并将输入数值进行合适的大小缩放后写到输出显示。

在抽象描述的最高层次上，可以将系统结构描述成一些部件的互联，比如处理器、存储