国外电子与通信教材系列

VHDL数字系统设计

Digital System Design with VHDL

英文版

[英] Mark Zwoliński 著

Prentice Hall

電子工業出版社
Publishing House of Electronics Industry
www.phei.com.cn

通信教材系列

VHDL数字系统设计

（英文版）

Digital System Design with VHDL

［英］ Mark Zwoliński 著

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 简 介

本书主要介绍数字系统设计的基本原理。VHDL是一种设计语言，这种设计语言允许设计人员先对基本的数字电路的特性和结构建模，然后再自动实现高级描述的电路结构。全书共分12章，主要讨论了电路设计自动工具的使用、CMOS和可编程逻辑技术，布尔代数的原理和组合逻辑设计。全书通过基本逻辑门模型引入了VHDL，强调了文档化代码的重要性，同时描述了各种建模技术，有限状态机的设计等。本书适合于学习数字电路课程的学生和相关工程技术人员。

版权贸易合同登记号：图字：01-2002-4134

图书在版编目（CIP）数据

VHDL数字系统设计/（英）左林斯基（Zwoliński, M.）著. -北京：电子工业出版社，2002.10
（国外电子与通信教材系列）
书名原文：Digital System Design with VHDL
ISBN 7-5053-8020-6

Ⅰ.V... Ⅱ.左... Ⅲ.硬件描述语言，VHDL-程序设计-英文 Ⅳ.TP312

中国版本图书馆CIP数据核字（2002）第072298号

责任编辑：赵红燕
印 刷 者：北京市增富印刷有限责任公司
出版发行：电子工业出版社 http://www.phei.com.cn
北京市海淀区万寿路173信箱 邮编：100036
经 销：各地新华书店
开 本：787×980 1/16 印张：21.25 字数：490千字
版 次：2002年10月第1版 2002年10月第1次印刷
定 价：35.00元

序

2001年7月间，电子工业出版社的领导同志邀请各高校十几位通信领域方面的老师，商量引进国外教材问题。与会同志对出版社提出的计划十分赞同，大家认为，这对我国通信事业、特别是对高等院校通信学科的教学工作会很有好处。

教材建设是高校教学建设的主要内容之一。编写、出版一本好的教材，意味着开设了一门好的课程，甚至可能预示着一个崭新学科的诞生。20世纪40年代MIT林肯实验室出版的一套28本雷达丛书，对近代电子学科、特别是对雷达技术的推动作用，就是一个很好的例子。

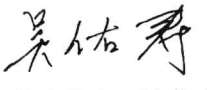
我国领导部门对教材建设一直非常重视。20世纪80年代，在原教委教材编审委员会的领导下，汇集了高等院校几百位富有教学经验的专家，编写、出版了一大批教材；很多院校还根据学校的特点和需要，陆续编写了大量的讲义和参考书。这些教材对高校的教学工作发挥了极好的作用。近年来，随着教学改革不断深入和科学技术的飞速进步，有的教材内容已比较陈旧、落后，难以适应教学的要求，特别是在电子学和通信技术发展神速、可以讲是日新月异的今天，如何适应这种情况，更是一个必须认真考虑的问题。解决这个问题，除了依靠高校的老师和专家撰写新的符合要求的教科书外，引进和出版一些国外优秀电子与通信教材，尤其是有选择地引进一批英文原版教材，是会有好处的。

一年多来，电子工业出版社为此做了很多工作。他们成立了一个“国外电子与通信教材系列”项目组，选派了富有经验的业务骨干负责有关工作，收集了230余种通信教材和参考书的详细资料，调来了100余种原版教材样书，依靠由20余位专家组成的出版委员会，从中精选了40多种，内容丰富，覆盖了电路理论与应用、信号与系统、数字信号处理、微电子、通信系统、电磁场与微波等方面，既可作为通信专业本科生和研究生的教学用书，也可作为有关专业人员的参考材料。此外，这批教材，有的翻译为中文，还有部分教材直接影印出版，以供教师用英语直接授课。希望这些教材的引进和出版对高校通信教学和教材改革能起一定作用。

在这里，我还要感谢参加工作的各位教授、专家、老师与参加翻译、编辑和出版的同志们。各位专家认真负责、严谨细致、不辞辛劳、不怕琐碎和精益求精的态度，充分体现了中国教育工作者和出版工作者的良好美德。

随着我国经济建设的发展和科学技术的不断进步，对高校教学工作会不断提出新的要求和希望。我想，无论如何，要做好引进国外教材的工作，一定要联系我国的实际。教材和学术专著不同，既要注意科学性、学术性，也要重视可读性，要深入浅出，便于读者自学；引进的教材要适应高校教学改革的需要，针对目前一些教材内容较为陈旧的问题，有目的地引进一些先进的和正在发展中的交叉学科的参考书；要与国内出版的教材相配套，安排好出版英文原版教材和翻译教材的比例。我们努力使这套教材能尽量满足上述要求，希望它们能放在学生们的课桌上，发挥一定的作用。

最后，预祝“国外电子与通信教材系列”项目取得成功，为我国电子与通信教学和通信产业的发展培土施肥。也恳切希望读者能对这些书籍的不足之处、特别是翻译中存在的问题，提出意见和建议，以便再版时更正。



中国工程院院士、清华大学教授

“国外电子与通信教材系列”出版委员会主任

出 版 说 明

21世纪初的5至10年是我国国民经济和社会发展的重要时期，也是信息产业快速发展的关键时期。在我国加入WTO后的今天，培养一支适应国际化竞争的一流IT人才队伍是我国高等教育的重要任务之一。信息科学和技术方面人才的优劣与多寡，是我国面对国际竞争时成败的关键因素。

当前，正值我国高等教育特别是信息科学领域的教育调整、变革的重大时期，为使我国教育体制与国际化接轨，有条件的高等院校正在为某些信息学科和技术课程使用国外优秀教材和优秀原版教材，以使我国在计算机教学上尽快赶上国际先进水平。

电子工业出版社秉承多年来引进国外优秀图书的经验，翻译出版了“国外计算机科学教材系列”丛书，这套教材覆盖学科范围广、领域宽、层次多，既有本科专业课程教材，也有研究生课程教材，以适应不同院系、不同专业、不同层次的师生对教材的需求，广大师生可自由选择和自由组合使用。这些教材涉及的学科方向包括网络与通信、操作系统、计算机组织与结构、算法与数据结构、数据库与信息处理、编程语言、图形图像与多媒体、软件工程等。同时，我们也适当引进了一些优秀英文原版教材，本着翻译版本和英文原版并重的原则，对重点图书既提供英文原版又提供相应的翻译版本。

在图书选题上，我们大都选择国外著名出版公司出版的高校教材，如Pearson Education培生教育出版集团、麦格劳－希尔教育出版集团、麻省理工学院出版社、剑桥大学出版社等。撰写教材的许多作者都是蜚声世界的教授、学者，如道格拉斯·科默（Douglas E. Comer）、威廉·斯托林斯（William Stallings）、哈维·戴特尔（Harvey M. Deitel）、尤利斯·布莱克（Uyless Black）等。

为确保教材的选题质量和翻译质量，我们约请了清华大学、北京大学、北京航空航天大学、复旦大学、上海交通大学、南京大学、浙江大学、哈尔滨工业大学、华中科技大学、西安交通大学、国防科学技术大学、解放军理工大学等著名高校的教授和骨干教师参与了本系列教材的选题、翻译和审校工作。他们中既有讲授同类教材的骨干教师、博士，也有积累了几十年教学经验的老教授和博士生导师。

在该系列教材的选题、翻译和编辑加工过程中，为提高教材质量，我们做了大量细致的工作，包括对所选教材进行全面论证；选择编辑时力求达到专业对口；对排版、印制质量进行严格把关。对于英文教材中出现的错误，我们通过与作者联络和网上下载勘误表等方式，逐一进行了修订。

此外，我们还将与国外著名出版公司合作，提供一些教材的教学支持资料，希望能为授课老师提供帮助。今后，我们将继续加强与各高校教师的密切联系，为广大师生引进更多的国外优秀教材和参考书，为我国计算机科学教学体系与国际教学体系的接轨做出努力。

电子工业出版社

教材出版委员会

Preface

About this book

There are probably tens, if not hundreds, of textbooks in print about digital design. There are a large number of books about VHDL. Why then did I add to this abundance by writing yet another textbook on digital design? As the title suggests, this book combines the discipline of digital design with a guide to the use of VHDL. By doing this, I have attempted to create an approach that is distinct from either of the other types of book.

The fact that I am trying to provide a different type of book should not be interpreted to mean that I am necessarily critical of everything that has gone before. There are a number of very good digital design textbooks, some of which are listed in the bibliography. Similarly, there are a number of excellent VHDL books. The majority of VHDL books, however, are aimed at practising engineers. This book is intended as a student textbook. Therefore, some features of VHDL are not described at all in this book. Equally, aspects of digital design are covered that would not be included in a typical VHDL book.

Syllabuses for electrical, electronic and computer engineering degrees vary between countries and between universities or colleges. The material in this book has been developed over a number of years for second and third year undergraduates and for postgraduate students. It is assumed that students will be familiar with the principles of Boolean algebra and with combinational logic design. At the University of Southampton, the first year undergraduate syllabus also includes introductions to synchronous sequential design and to programmable logic. This book therefore builds upon these foundations. It has often been assumed that topics such as VHDL are too specialized for second year teaching and are best left to final year or postgraduate courses. There are several good reasons why VHDL should be introduced earlier into the curriculum. With increasing integrated circuit complexity, there is a need from industry for graduates with knowledge of VHDL and the associated design tools. If left to the final year, there is little or no time for the student to apply such knowledge in project work. Second,

conversations with colleagues from many countries suggest that today's students are opting for computer science or computer engineering courses in preference to electrical or electronic engineering. VHDL offers a means to interest computing-oriented students in hardware design. Finally, simulation and synthesis tools are now mature and available relatively cheaply to educational establishments on PC platforms.

Structure of this book

Chapter 1 introduces the ideas behind this book, namely the use of electronic design automation tools and CMOS and programmable logic technology. We also consider some engineering problems, such as noise margins and fan-out. In Chapter 2, the principles of Boolean algebra and of combinational logic design are reviewed. The important matter of timing and the associated problem of hazards are discussed. Some basic techniques for representing data are discussed.

VHDL is introduced in Chapter 3 through basic logic gate models. The importance of documented code is emphasized. We show how to construct netlists of basic gates and how to model delays through gates. We also discuss parameterized models and constant and unconnected inputs and outputs. The idea of using VHDL to verify VHDL models by using testbenches is introduced. Finally, we briefly introduce the concept of configurations.

In Chapter 4, a variety of modelling techniques are described. Combinational building blocks, buffers, decoders, encoders, multiplexers, adders and parity checkers are modelled using a range of concurrent and sequential VHDL coding constructs. The VHDL models of hardware introduced in this chapter and in Chapters 5, 6 and 7 are, in principle, synthesizable, although discussion of exactly what is supported is deferred until Chapter 9. In addition, the IEEE dependency notation is introduced.

Chapter 5 is probably the most important chapter of the book and discusses what might be considered the cornerstone of digital design: the design of finite state machines. The ASM chart notation is used. The design process from ASM chart to D flip-flops and next state and output logic is described. VHDL models of state machines are introduced.

Chapter 6 introduces various sequential building blocks: latches, flip-flops, registers, counters, memory and a sequential multiplier. The same style as Chapter 4 is used, with IEEE dependency notation and the introduction of VHDL coding constructs.

In Chapter 7 the concepts of the previous three chapters are combined. The ASM chart notation is extended to include coupled state machines and registered outputs, and hence to datapath-controller partitioning. From this, we explain the idea of instructions in hardware terms and go on to model a very basic microprocessor in VHDL. This provides a vehicle to introduce VHDL subroutines and packages.

VHDL remains primarily a modelling language. Chapter 8 describes the operation of a VHDL simulator. The idea of event-driven simulation is first explained and the specific features of a VHDL simulator are then discussed. Although the entire VHDL language can be simulated, some constructs simulate more efficiently than others; therefore techniques for writing models that are more efficient are discussed. File operations are also discussed in this chapter because such functionality is only appropriate to simulation models.

The other, increasingly important, role of VHDL is as a language for describing synthesis models, as discussed in Chapter 9. The dominant type of synthesis tool available today is for RTL synthesis. Such tools can infer the existence of flip-flops and latches from a VHDL model. These constructs are described. Conversely, flip-flops can be created in error if the description is poorly written, and common pitfalls are described. The synthesis process can be controlled by constraints. Because these constraints are outside the language, they are discussed in general terms. Suitable constructs for FPGA synthesis are discussed. Finally, behavioural synthesis, which promises to become an important design technology, is briefly examined.

Chapters 10 and 11 are devoted to the topics of testing and design for test. This area has often been neglected, but is now recognized as being an important part of the design process. In Chapter 10 the idea of fault modelling is introduced. This is followed by test generation methods. The efficacy of a test can be determined by fault simulation. At the time of writing, there are no commercial VHDL-based fault simulators available. The final section of this chapter shows how fault modelling and fault simulation can be performed using a standard VHDL simulator. The VHDL code also introduces constructs such as pointers and shared (global) variables.

In Chapter 11, three important design-for-test principles are described: scan path, built-in self-test (BIST) and boundary scan. This has always been a very dry subject, but a VHDL simulator can be used, for example, to show how a BIST structure can generate different signatures for fault-free and faulty circuits. Boundary scan uses a subset of VHDL to describe the test structures used on a chip, and an example is given.

In the final chapter, we use VHDL as a tool for exploring anomalous behaviour in asynchronous sequential circuits. Although the predominant design style is currently synchronous, it is likely that digital systems will increasingly consist of synchronous circuits communicating asynchronously with each other. We introduce the concept of the fundamental mode and show how to analyze and design asynchronous circuits. We use VHDL simulations to illustrate the problems of hazards, races and setup and hold time violations. We also discuss the problem of metastability.

Three appendices are included. The first appendix lists the various VHDL-related standards and speculates on the future development of VHDL. The second appendix briefly describes the Verilog hardware description language. Verilog is the major alternative to VHDL and it is likely that designers will have to be familiar with both. The third appendix covers shared variables.

At the end of each chapter a number of exercises have been included. These exercises are almost secondary to the implicit instruction in each chapter to simulate and, where appropriate, synthesize each VHDL example. To perform these simulation and synthesis tasks, the reader will have to write his or her own testbenches and constraints files. The examples are available on the World Wide Web at the address given in the next section.

How to use this book

Obviously, this book can be used in a number of different ways, depending on the level of the course. At the University of Southampton, I am using the material as follows.

Second year of MEng/BEng in Electronic Engineering and Computer Engineering
Chapters 1 and 2 are review material, which the students would be expected to read independently. Lectures then cover the material of Chapters 3, 4, 5, 6 and 7. Some of this material can be considered optional, such as Sections 3.8, 6.3 and 6.7. Additionally, constructs such as `with select` could be omitted if time presses. The single-stuck fault model of Section 10.2 and the principles of test pattern generation in Section 10.3, together with the principles of scan design in section 11.2 would also be covered in lectures.

Third year of MEng/BEng in Electronic Engineering and Computer Engineering
Students would be expected to independently re-read Chapters 3 to 7. Lectures would cover Chapters 8, 9, 10, 11 and 12.

In both years, students need to have access to a VHDL simulator and an RTL synthesis tool in order to use the examples in the text. In the second year, a group design exercise involving synthesis to an FPGA would be an excellent supplement to the material. In the third year at Southampton, all students do an individual project. There is no additional formal laboratory work. Some of the individual projects will involve the use of VHDL.

The VHDL code for all the examples in the text is available on the World Wide Web at:
`http://www.ecs.soton.ac.uk/~mz/VHDL`

Mark Zwoliński
Southampton
August 1999

Acknowledgements

The publishers are grateful to the following for permission to reproduce the material:
Figure 1.11 reproduced with the permission of Lattice Semiconductor Corporation.
Whilst every effort has been made to trace the owners of copyright material, in a few cases this has proved impossible and we take this opportunity to offer our apologies to any copyright holders whose rights we may have unwittingly infringed.

目录概览

Contents

Chapter 1

Introduction

In this chapter we will review the design process, with particular emphasis on the design of digital systems using hardware description languages such as VHDL. The technology of CMOS integrated circuits will be briefly revised and programmable logic technologies will be discussed. Finally, the relevant electrical properties of CMOS and programmable logic are reviewed.

1.1 Modern digital design

Electronic circuit design has traditionally fallen into two main areas: analogue and digital. These subjects are usually taught separately, and electronics engineers tend to specialize in one area. Within these two groupings there are further specializations, such as radio frequency analogue design; digital integrated circuit design; and, where the two domains meet, mixed-signal design. In addition, of course, software engineering plays an increasingly important role in embedded systems.

Digital electronics is ever more significant in consumer goods. Cars have sophisticated control systems. Many homes now have personal computers. Products that used to be thought of as analogue, such as radio, television and telephones, are or are becoming digital. Digital compact discs have almost entirely replaced analogue LPs for recorded audio. With these changes, the lifetimes of products have lessened. In a period of less than a year, new models will probably have replaced all the digital electronic products in your local store.

1.1.1 Design automation

To keep pace with this rapid change, electronics products have to be designed extremely quickly. Analogue design is still a specialized (and well-paid) profession. Digital design has become very dependent on computer-aided design (CAD) – also known as design automation (DA) or electronic design automation (EDA). The EDA tools allow two tasks to be performed: *synthesis*, in other words the translation of a specification into an actual implementation of the design; and *simulation*, in which the specification or the detailed implementation can be exercised in order to verify correct operation.

Synthesis and simulation EDA tools require that the design be transferred from the designer's imagination into the tools themselves. This can be done by drawing a diagram of the design using a graphical package. This is known as *schematic capture*. Alternatively, the design can be represented in a textual form, much like a software program. Textual descriptions of digital hardware can be written in a modified programming language, such as C, or in a *hardware description language* (HDL). Over the past thirty years or so, a number of HDLs have been designed. Two HDLs are in common usage today: Verilog and VHDL (VHSIC Hardware Description Language, where VHSIC stands for Very High Speed Integrated Circuit). Standard HDLs are important because they can be used by different CAD tools from different tool vendors. In the days before Verilog and VHDL, every tool had its own HDL, requiring laborious translation between HDLs, for example to verify the output from a synthesis tool with another vendor's simulator.

1.1.2 Logic gates

The basic building blocks of digital circuits are *gates*. A gate is an electronic component with a number of inputs and, generally, a single output. The inputs and the outputs are normally in one of two states: logic 0 or logic 1. These logic values are represented by voltages (for instance, 0 V for logic 0 and 3.3 V for logic 1) or currents. The gate itself performs a logical operation using all of its inputs to generate the output. Ultimately, of course, digital gates are really analogue components, but for simplicity we tend to ignore their analogue nature.

It is possible to buy a single integrated circuit containing, say, four identical gates, as shown in Figure 1.1. (Note that two of the connections are for the positive and negative power supplies to the device. These connections are not normally shown in logic diagrams.) A digital system could be built by connecting hundreds of such devices together – indeed many systems have been designed in that way. Although the individual integrated circuits might cost as little as 10 cents each, the cost of designing the printed circuit board for such a system and the cost of assembling the board are very significant and this design style is no longer cost-effective.

Much more complicated functions are available as mass-produced integrated circuits, ranging from flip-flops through to microprocessors. With increasing complexity comes flexibility – a microprocessor can be programmed to perform a near-infinite variety of tasks. Digital system design therefore consists, in part, of taking standard components and connecting them together. Inevitably, however, some aspect of the functionality will