经 典 原 版 书 库

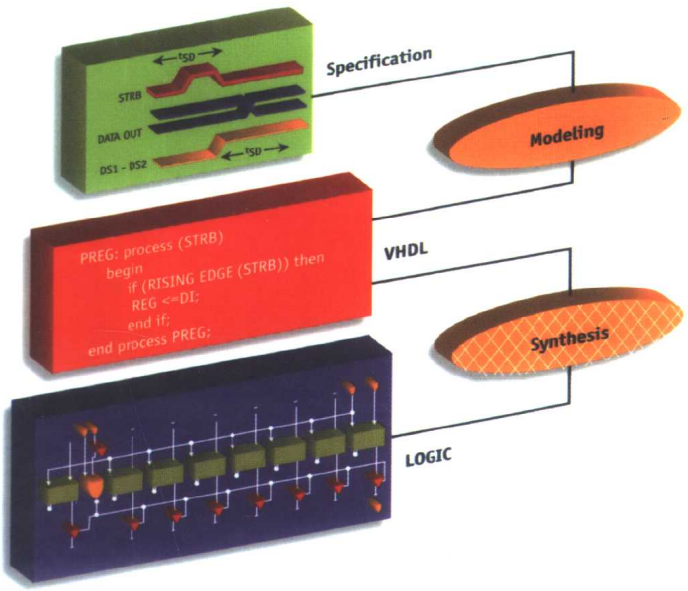
VHDL 设计、表示和综合

（英文版 · 第2版）

VHDL Design

Representation and Synthesis

SECOND EDITION



附赠 CD-ROM

James R. Armstrong F. Gail Gray

（美）James R. Armstrong F. Gail Gray 著

机械工业出版社
China Machine Press

Prentice Hall

VHDL设计、表示和综合（英文版·第2版）

VHDL Design Representation and Synthesis (Second Edition)

全面介绍基于综合的VHDL设计

本书清晰、全面地阐述了当今硬件设计的主流方法——使用硬件描述语言进行综合，和当今主流的综合工具方面的最新知识。两位权威作者从结构化设计的介绍开始，深入讲述了VHDL语言及其关键结构，然后循序渐进地讲解建模过程，使用了不同抽象层次的各种例子，并阐述了用于提高仿真效率和综合工具兼容性的各种技巧。

主要内容

- **设计工具**：编辑器，仿真器，检查器，分析器，优化器和综合器
- **VHDL**：主要结构，句法描述，源文件，数据类型，数据对象，语句和高级特性
- **基本的VHDL建模技术**：传输和时延，并发性，组合逻辑，时序逻辑，系统原语等等
- **将VHDL与设计流程结合**：从算法层次的可执行规范，到门级或单元级的实现PLD，门阵列，FPGA（使用Xilinx工具）和标准单元（使用Synopsys工具）建模

作者简介

James R. Armstrong 博士是弗吉尼亚理工大学（Virginia Tech）电机与计算机工程系的教授，VHDL方面的国际权威。IEEE标准化委员会的创始成员以及IEEE VHDL标准化工作组和设计自动化工作组的主席。

F. Gail Gray 博士是弗吉尼亚理工大学（Virginia Tech）电机与计算机工程系的教授。拥有密歇根大学计算机、信息和控制工程博士学位。

配套光盘含有书中所有VHDL模型，模型测试基准，以及项目序列，等等。

封面设计 陈子平

ISBN 7-111-11676-3

9 787111 116769

网上购书：www.china-pub.com

北京市西城区百万庄南街1号 100037

读者服务热线：(010) 68995259 8006100280 (北京地区)

总编信箱：chiefeditor@hzbook.com

限中国大陆地区销售

ISBN 7-111-11676-3/TP·2814

定价：69.00元（附光盘）

经 典 原 版 书 库

VHDL 设计、表示和综合

（英文版 · 第2版）

VHDL Design Representation and Synthesis

(Second Edition)

（美） James R. Armstrong
F. Gail Gray 著

机械工业出版社
China Machine Press

本书版权登记号：图字：01-2003-1006

图书在版编目（CIP）数据

VHDL设计、表示和综合（英文版·第2版）/（美）阿姆斯特朗（Armstrong, J.R.），（美）格雷（Gray, F.G.）著. —北京：机械工业出版社，2003.3

（经典原版书库）

书名原文：VHDL Design Representation and Synthesis, Second Edition

ISBN 7-111-11676-3

Ⅰ.V… Ⅱ.①阿…②格… Ⅲ.硬件描述语言，VHDL－英文 Ⅳ.TP312

中国版本图书馆CIP数据核字（2003）第009745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：华 章

北京市密云县印刷厂印刷·新华书店北京发行所发行

2003年3月第1版第1次印刷

787mm×1092mm 1/16·42.25印张

印数：0 001-3 000册

定价：69.00元（附光盘）

Preface

The purpose of this book is to integrate hardware description languages into the digital design process at all levels of abstraction. There are two main steps in this process: (1) development of a hardware description language model and (2) synthesis of the model into an ASIC logic circuit or FPGAs. In teaching this process, we use VHDL, the VHSIC Hardware Description Language. VHDL, whose development began in 1983 under DOD sponsorship, was further developed by the IEEE and released as IEEE Standard 1076 in 1987. Further improvements were incorporated since then and the language was re-released as an updated standard in 1993. Since that time, VHDL has evolved into a de facto industry standard for hardware description languages. In the opinion of the authors, it has the most comprehensive set of modeling constructs available in any hardware description language. For these reasons, VHDL was chosen as the base language for this book. We explore the language in an in-depth, unified manner.

Most books currently on the market that treat hardware description languages, particularly VHDL, are either: (1) language texts that cover the VHDL language thoroughly, but do not show how to integrate the language into the digital design process, or (2) logic design books that primarily use VHDL models as simulation tools to validate designs that are produced in the classical manner. This book fully integrates VHDL into the design process starting with a high-level executable model that provides an unambiguous, executable version of the specification, and concluding with a gate-level implementation.

In this book, synthesis is viewed as a multistep process, beginning with an English description which is transformed first into VHDL and then from VHDL into a circuit schematic. We discuss synthesis from two viewpoints: 1) the mappings: emphasis is placed on understanding the relationship between VHDL language constructs and the implied logic circuit. A full chapter is devoted to correct modeling style for synthesis; 2) the tools: we illustrate the synthesis process using two very popular tool sets, the Synopsys Design Analyzer and Compiler (for ASICs) and the Xilinx Foundation Series (for FPGAs). Since ASICs and FPGAs are the targets,

a chapter is devoted to these technologies. The book also contains a chapter illustrating the complete top-design design process from specification to logic synthesis.

This book is written for three main educational purposes: (1) for a second course in logic design for undergraduate students in Electrical Engineering, Computer Engineering, and Computer Science; (2) for a graduate course dealing with hardware description languages and other design aids; and (3) for practicing engineers who wish to learn about design with hardware description languages. Thus the assumed background for the book is (1) a basic course in computer organization and logic design and (2) some knowledge of high-level languages, such as C, C++, or JAVA.

The authors use the text in a course, which is the second course in a logic design sequence. The students are either juniors in Computer Engineering, for whom the course is required, or Electrical Engineering seniors, for whom the course is an elective. In this semester length course we cover Chapters 1, 2, 3, 4, 5, 9, 10, and 11. The emphasis is on developing VHDL models in a conservative algorithmic style that can be synthesized. To support this in the laboratory, we use a PC version of ViewLogic, Inc.'s Workview for VHDL modeling and simulation and schematic capture. Xilinx software and XS40/XTEND boards are used for FPGA synthesis. We also employ System View from Elanix to provide for high-level design of digital filters. Workstation-based Synopsys tools are used for ASIC synthesis. All students in our department have their own PCs, so the use of a PC-based system such as Workview has been effective in being able to serve the large number of students we normally teach in our second digital design course. For this same reason, we use telnet and dc_shell scripts for Synopsys synthesis. Typical assignments include:

1. An introductory assignment to familiarize students with Workview's VHDL modeling, simulation and schematic capture environment.
2. An assignment to develop and simulate a single VHDL behavioral model.
3. An assignment to develop a model of a counter, or some similar circuit. VHDL behavioral models are developed for counter flip-flops and gates, and the schematic capture capability of Workview is used to construct the structural model of the counter.
4. An assignment to translate a system description is first translated into a VHDL behavioral model which is simulated. This is typically a state machine such as an interface protocol, a vending machine, or a traffic light controller.
5. An introductory tutorial to the Xilinx Foundation Series Software.
6. An assignment to implement a small circuit in both Synopsys ASIC logic and FPGAs and compare speed of execution.
7. A fairly complicated FPGA project such as a booth multiplier, calculator, small processor, digital filter, or graphics display. For the filter, the codec on the XSTEND board is used for A/D and D/A comversion. The Xilinx filter code is developed using System View. The graphics display displays a pattern stored in RAM on a VGA monitor.

If used for a graduate course, the entire book can be covered in one semester. In such a course, one can cover the broad range of constructs in the language and examine in detail the language semantics for both simulation and synthesis. In our graduate course at Virginia Tech, we synthesize with Synopsys and validate synthesized models. We study ways to control the synthesis to achieve optimum circuits in a delay or area sense. High-level modeling tools such as

Express VHDL, SPW, and System View are also covered. A comparison is done between VHDL and Verilog.

For this course, the student's laboratory assignments include:

1. An assignment to develop and simulate a single VHDL behavioral model.
2. An assignment to develop a model of a counter or some similar circuit. VHDL behavioral models are developed for counter flip-flops and gates, and then a VHDL structural model is developed for the whole system.
3. An assignment involving complex data types, e.g., using array aggregates and record types to implement a tabular representation of a finite state machine.
4. A system modeling assignment that involves the use of bus resolution and bus protocols. This system employs the IEEE 9 valued logic system. Examples include the URISC processor system in Chapter 6 or a histogram construction system for image processing.
5. An assignment where a model is written, simulated, and synthesized using both VHDL and Verilog and comparison's made
6. A semester project where the students model a system of their choice. One can choose projects, which stretch the language, i.e., involve applications that are not typical, such as modeling parallel processing systems or modeling systems which are not digital.

The book contains hundreds of VHDL models and code fragments. All code has been analyzed, and simulated, and synthesized (where required), using the Synopsys VHDL system. The only exception to this is the VHDL 93 code. In addition, the text contains over 300 homework problems with a wide range of difficulty. Types of problems include short answer questions, simple design problems, complex system design problems involving design, modeling, and simulation, and problems that require a study of a design or design tool issue. Some problems in this latter category would make good thesis projects!

Accompanying the book is a CD-ROM. On the CD are: 1) source files for all VHDL code in the book, 2) a set of projects accompanied by supporting data command files, and 3) packages to support common design paradigms. Problem and project solutions and Power Point lecture slides are available to instructors who adopt our text for classroom use.

Writing a book of this nature is a large undertaking. In doing so we have received the help and assistance of a number of individuals and organizations. We would like to thank:

1. Viewlogic, Synopsys, Inc., and Xilinx, for their support in providing us with the VHDL software to check out the VHDL code in the book and for use in our courses on hardware description languages.
2. Dave Barton of Intermetrics for his review of the manuscript.
3. Our production editor at Prentice Hall, Vincent Janoski.
4. Our book editor at Prentice Hall, Bernard Goodwin, for his enthusiastic support of the project.

We would also like to thank our wives, Marie and Caryl, for their encouragement and support in spite of the long hours we spent in front of our computers and their tolerance of laptops accompanying us on trips and vacations over a period of two years.

Contents

CHAPTER 1

Structured Design Concepts

In this chapter we present basic definitions that relate to the design process. It is necessary to introduce them now so that other concepts can be explained. The reader should study them carefully in order to comprehend material introduced later. It will also be useful to revisit this chapter as one proceeds through the text since the full meaning of the terms will only become clear through use and example.

1.1 THE ABSTRACTION HIERARCHY

In this section we present the abstraction hierarchy employed by digital designers. Abstraction can be expressed in the following two domains:

Structural domain. A domain in which a component is described in terms of an interconnection of more primitive components.

Behavioral domain. A domain in which a component is described by defining its input/output response.

Figure 1.1 shows structural and behavioral descriptions for a logic circuit, which detects two or more consecutive 1's or two or more consecutive 0's on its input X. The structural description is an interconnection of gate and flip-flop primitives. The behavioral description is expressed textually in a hardware description language (HDL).

An abstraction hierarchy can be defined as follows:

Abstraction hierarchy. A set of interrelated representation levels that allow a system to be represented in varying amounts of detail.

Figure 1.2 shows a picture of a typical abstraction hierarchy. For each level i in the hierarchy there exists a transformation to level i+1. The level of detail usually increases monotonically as one moves down in the hierarchy.