Introduction

This self-study guide came about as the result of the popularity of my textbook, *Verilog Designer's Guide.* That book is an intermediate to advanced level reference book about the Verilog Hardware Description Language. The book has a lot of good advice and a large number of well-documented Verilog routines that can be used to create real-world hardware. Shortly after its publication, the Institute of Electrical and Electronic Engineers (IEEE) approached me to create a study guide based on the book. I realized that for those who already knew Verilog, the book was fairly self-explanatory. However, it could be difficult to navigate for those who are just learning. Yet the functions in the book make ideal examples and problems for learning the language. With that in mind I decided to create this introductory study guide using my textbook for illustration and problem assignments. This study guide is based on the Verilog seminar that I give around the world. Over the years I've used the feedback from students to try to make this the best introductory Verilog course available. I hope I've succeeded. If you want to comment, either to congratulate me on the excellent job I've done, to ask a question, to point out a mistake or misconception, or to suggest improvements for the future, or simply to complain, please do so. I welcome all feedback.

Course Objectives

Hardware Description Languages (HDLs) use statements, like programming language statements, in order to define, simulate, synthesize, and layout hardware. One of the main HDLs is Verilog, a widely used and standardized language. Verilog can be used to design anything from the most complex ASIC to the least complex PAL. As ASICs and FPGAs become more complex, HDLs become a necessity for their design.

This course teaches how to use Verilog to design and simulate hardware. It begins by explaining the benefits of HDLs over other design entry methods, including its ability to model different levels of abstraction, its reusability, and documentability. Next, the syntax of the Verilog language is explained in detail. By the end of the course, the student should be able to design and simulate real hardware using Verilog.

Intended Audience

The book is aimed at electrical engineering students and practicing electrical engineers who are not yet familiar with Verilog. It is intended for engineers who wish to cover a lot of ground toward understanding and using Verilog to create real designs. The prerequisite is a good knowledge of digital logic design. Knowledge of programming languages, such as C or C++, and knowledge of other Hardware Description Languages (HDLs) such as VHDL, is beneficial but not mandatory.

How To Use This Guide

This guide is divided into lessons. Each lesson is divided into sections. Each section has an icon to give some idea about it. The sections and their corresponding icons are described below.

Objectives

At the beginning of each lesson in this guide, the *Objectives* section will describe the goals of the particular lesson. This enables you to know the concepts that will be discussed and examined in the section.

Comments

This is where I get to shoot my mouth off. I will give my opinions, anecdotes, and real-world examples relating to the subject. Many students find these tangents to be very interesting, breaking up the otherwise dry material. Other, however, think it's a great waste of time. If you fall into the latter category, feel free to skip these sections. You won't miss any crucial material (you'll simply hurt my feelings). Comments are in grey boxes to separate them from critical subject matter.

Problems

At the end of each lesson in this guide, there will be a number of problems to solve based on the material in that lesson. Often these problems use specific sections of my textbook, *Verilog Designer's Library*, as their basis. These problems are designed not only to test you, but to reinforce the subject. You learn better by applying your knowledge to solve problems. The answers to the problems are in the back of this guide.

Solutions

In this section you will find all of the answers to all of your problems. Too bad life isn't like this guide.

Acknowledgements

I'd like to thank Bernard Goodwin, my editor at Prentice-Hall, who put me in touch with the IEEE in order to create this study guide. Also, Jill Bagley at the IEEE, for her assistance in getting this guide approved. I'd like to thank Ken McElvain, Chief Technical Officer of Synplicity (his title changes so often that it may be something else by the time you read this), a brilliant engineer, for his technical assistance in reviewing examples and Verilog code and generally giving recommendations and advice. Thanks to Dan Hafeman, CTO of Ikos Systems, another brilliant engineer who gave me advice and feedback. Carlo Treves was nice enough to proofread the manuscript while he was on vacation, and I'm grateful to him for his comments. Finally, of course, acknowledgement and thanks go to my wife Carrie for putting up with my bad moods while I'm under pressure to get things done on schedule, which is almost all of the time.

About the Author

Bob Zeidman is the president of The Chalkboard Network (www.chalknet.com), a company that provides high tech training courses on the Internet. Previously, Bob was the president of Zeidman Consulting, a contract R & D firm specializing in digital hardware design and software design. His clients included Apple Computer, Cisco Systems, Ikos Systems, and Texas Instruments. He has been working in the electronics industry since 1983 and is the author of the textbook, *Verilog Designer's Library* from Prentice-Hall, as well as a number of articles on engineering and business. Bob lectures on engineering and business at conferences throughout the world. He holds an MSEE from Stanford University, and a BSEE and a BA in physics from Cornell University.