Preface

The Verilog language is a hardware description language that provides a means of specifying a digital system at a wide range of levels of abstraction. The language supports the early conceptual stages of design with its behavioral level of abstraction, and the later implementation stages with its structural abstractions. The language includes hierarchical constructs, allowing the designer to control a description's complexity.

Verilog was originally designed in the winter of 1983/84 as a proprietary verification/simulation product. Later, several other proprietary analysis tools were developed around the language, including a fault simulator and a timing analyzer. More recently, Verilog has also provided the input specification for logic and behavioral synthesis tools. The Verilog language has been instrumental in providing consistency across these tools. The language was originally standardized as IEEE standard #1364-1995. It has recently been revised and standardized as IEEE standard #1364-2001. This book presents this latest revision of the language, providing material for the beginning student and advanced user of the language.

It is sometimes difficult to separate the language from the simulator tool because the dynamic aspects of the language are defined by the way the simulator works. Further, it is difficult to separate it from a synthesis tool because the semantics of the language become limited by what a synthesis tool allows in its input specification and produces as an implementation. Where possible, we have stayed away from simulator- and synthesis-specific details and concentrated on design specification. But, we have included enough information to be able to write working executable models.
The book takes a tutorial approach to presenting the language. Indeed, we start with a tutorial introduction that presents, via examples, the major features of the language and the prevalent styles of describing systems. We follow this with a detailed presentation on using the language for synthesizing combinational and sequential systems. We then continue with a more complete discussion of the language constructs.

Our approach is to provide a means of learning by observing the examples and doing exercises. Numerous examples are provided to allow the reader to learn (and re-learn!) easily by example. It is strongly recommended that you try the exercises as early as possible with the aid of a Verilog simulator. The examples shown in the book are available in electronic form on the enclosed CD. Also included on the CD is a simulator. The simulator is limited in the size of description it will handle.

The majority of the book assumes a knowledge of introductory logic design and software programming. As such, the book is of use to practicing integrated circuit design engineers, and undergraduate and graduate electrical or computer engineering students. The tutorial introduction is organized in a manner appropriate for use with a course in introductory logic design. A separate appendix, keyed into the tutorial introduction, provides solved exercises that discuss common errors. The book has also been used for courses in introductory and upper level logic and integrated circuit design, computer architecture, and computer-aided design (CAD). It provides complete coverage of the language for design courses, and how a simulator works for CAD courses. For those familiar with the language, we provide a preface that covers most of the new additions to the 2001 language standard.

The book is organized into eleven chapters and eight appendices. The first part of the book contains a tutorial introduction to the language which is followed by a chapter on its use for logic synthesis. The second part of the book, Chapters 3 through 6, provide a more rigorous presentation of the language's behavioral, hierarchical, and logic level modeling constructs. The third part of the book, Chapters 7 through 11, covers the more specialized topics of cycle-accurate modeling, timing and event driven simulation, user-defined primitives, and switch level modeling. Chapter 11 suggests two major Verilog projects for use in a university course. One appendix provides tutorial discussion for beginning students. The others are reserved for the dryer topics typically found in a language manual; read those at your own risk.

Have fun designing great systems...

always,
Donald E. Thomas
Philip R. Moorby
March 2002
The Verilog® Hardware Description Language
Thomas, D.; Moorby, P.
2002, XXII, 386 p. With online files/update., Softcover
ISBN: 978-0-387-84930-0