



A VHDL Primer

Jayaram Bhasker

*American Telephone and Telegraph Company
Bell Laboratories Division*

ROCKWELL INTERNATIONAL, INC.
INFORMATION CENTER (NB)
4311 JAMBOREE ROAD (501-345)
NEWPORT BEACH, CA. 92660



P T R Prentice Hall
Englewood Cliffs, New Jersey 07632

Dedicated to
my parents, Nagamma and Appiah Jayaram

Contents

Preface	vi
CHAPTER 1. Introduction	1
1.1. What is VHDL?	1
1.2. History	1
1.3. Capabilities	1
1.4. Hardware Abstraction	2
CHAPTER 2. A Tutorial	4
2.1. Basic Terminology	4
2.2. Entity Declaration	5
2.3. Architecture Body	6
2.3.1. Structural Style of Modeling	6
2.3.2. Dataflow Style of Modeling	7
2.3.3. Behavioral Style of Modeling	8
2.3.4. Mixed Style of Modeling	9
2.4. Configuration Declaration	10
2.5. Package Declaration	11
2.6. Package Body	12
2.7. Model Analysis	12
2.8. Simulation	13
CHAPTER 3. Basic Language Elements	14
3.1. Identifiers	14
3.2. Data Objects	14
3.3. Data Types	16
3.3.1. Subtypes	16
3.3.2. Scalar Types	16
3.3.3. Composite Types	19
3.3.4. Access Types	22
3.3.5. Incomplete Types	23
3.3.6. File Types	24
3.4. Operators	25
3.4.1. Logical Operators	26
3.4.2. Relational Operators	26
3.4.3. Adding Operators	26
3.4.4. Multiplying Operators	26
3.4.5. Miscellaneous Operators	27
CHAPTER 4. Behavioral Modeling	28
4.1. Entity Declaration	28
4.2. Architecture Body	28
4.3. Process Statement	29
4.4. Variable Assignment Statement	30
4.5. Signal Assignment Statement	30
4.6. Wait Statement	31
4.7. If Statement	32
4.8. Case Statement	33
4.9. Null Statement	34
4.10. Loop Statement	34

4.11. Exit Statement	35
4.12. Next Statement	36
4.13. Assertion Statement	37
4.14. More on Signal Assignment Statement	38
4.14.1. Inertial Delay Model	38
4.14.2. Transport Delay Model	38
4.14.3. Creating Signal Waveforms	39
4.14.4. Signal Drivers	39
4.15. Other Sequential Statements	42
4.16. Multiple Processes	42
CHAPTER 5. Dataflow Modeling	44
5.1. Concurrent Signal Assignment Statement	44
5.2. Concurrent versus Sequential Signal Assignment	45
5.3. Delta Delay Revisited	46
5.4. Multiple Drivers	47
5.5. Conditional Signal Assignment Statement	49
5.6. Selected Signal Assignment Statement	50
5.7. Block Statement	50
5.8. Concurrent Assertion Statement	53
CHAPTER 6. Structural Modeling	54
6.1. An Example	54
6.2. Component Declaration	54
6.3. Component Instantiation	55
6.4. Other Examples	57
6.5. Resolving Signal Values	59
CHAPTER 7. Generics and Configurations	61
7.1. Generics	61
7.2. Why Configurations?	63
7.3. Configuration Specification	63
7.4. Configuration Declaration	67
7.5. Default Rules	69
CHAPTER 8. Subprograms and Overloading	71
8.1. Subprograms	71
8.1.1. Functions	71
8.1.2. Procedures	72
8.1.3. Declarations	74
8.2. Subprogram Overloading	74
8.3. Operator Overloading	76
CHAPTER 9. Packages and Libraries	78
9.1. Package Declaration	78
9.2. Package Body	79
9.3. Design Libraries	79
9.4. Design File	80
9.5. Order of Analysis	80
9.6. Implicit Visibility	81
9.7. Explicit Visibility	81
9.7.1. Library Clause	82
9.7.2. Use Clause	82

CHAPTER 10. Advanced Features	84
10.1. Entity Statements	84
10.2. Generate Statements	84
10.3. Aliases	87
10.4. Qualified Expressions	88
10.5. Type Conversions	88
10.6. Guarded Signals	88
10.7. Attributes	90
10.7.1. User-Defined Attributes	90
10.7.2. Predefined Attributes	91
10.8. Aggregate Targets	95
10.9. More on Block Statements	96
CHAPTER 11. Model Simulation	98
11.1. Simulation	98
11.2. Writing a Test Bench	100
11.2.1. Waveform Generation	101
11.2.2. Monitoring Behavior	106
CHAPTER 12. Hardware Modeling Examples	107
12.1. Modeling Entity Interface	107
12.2. Modeling Simple Elements	107
12.3. Different Styles of Modeling	110
12.4. Modeling Regular Structures	111
12.5. Modeling Delays	112
12.6. Modeling Conditional Operations	113
12.7. Modeling Synchronous Logic	113
12.8. State Machine Modeling	117
12.9. Interacting State Machines	118
12.10. Modeling Moore FSM	121
12.11. Modeling Mealy FSM	122
12.12. A Simplified Blackjack Program	123
12.13. Hierarchy in Design	124
APPENDIX A. Predefined Environment	128
A.1. Reserved Words	128
A.2. Package STANDARD	129
A.3. Package TEXTIO	130
APPENDIX B. Syntax Reference	132
B.1. Conventions	132
B.2. The Syntax	132
APPENDIX C. A Package Example	144
C.1. The Package ATT_PACKAGE	144
Bibliography	150
Index	

Preface

VHDL is a hardware description language that can be used to model a digital system. It contains elements that can be used to describe the behavior or structure of the digital system, with the provision for specifying its timing explicitly. The language provides support for modeling the system hierarchically and also supports top-down and bottom-up design methodologies. The system and its subsystems can be described at any level of abstraction ranging from the architecture level to gate level. Precise simulation semantics are associated with all the language constructs, and therefore, models written in this language can be verified using a VHDL simulator.

The aim of this book is to introduce the VHDL language to the reader at the beginner's level. No prior knowledge of the language is required. The reader is, however, assumed to have some knowledge of a high-level programming language, like C or Pascal, and a basic understanding of hardware design. This text is intended for both software and hardware designers interested in learning VHDL with no specific emphasis being placed on either discipline.

VHDL is a large and verbose language with many complex constructs that have complex semantic meanings and is difficult to understand initially (VHDL is often quoted to be an acronym for Very Hard Description Language). However, it is possible to quickly understand a subset of VHDL which is both simple and easy to use. The emphasis of this text is on presenting this set of simple and commonly used features of the language so that the reader can start writing models in VHDL. These features are powerful enough to be able to model designs of large degrees of complexity.

This book is not intended to replace the IEEE Standard VHDL Language Reference Manual, the official language guide, but to complement it by explaining the complex constructs of the language using an example-based approach. Emphasis is placed on providing illustrative examples that explain the different formulations of the language constructs and their semantics. The complete syntax of language constructs is often not described, instead, the most common usage of these constructs are presented. Syntax for constructs is written in a self-explanatory fashion rather than through the use of formal terminology (the Backus-Naur Form) that is used in the Language Reference Manual. This text does not cover the entire language but concentrates on the most useful aspects.

Book organization

Chapter 1 gives a brief history of the development of the VHDL language and presents its major capabilities. Chapter 2 provides a quick tutorial to demonstrate the primary modeling features. The following chapters expand on the concepts presented in this tutorial. Chapter 3 describes the basic elements of the language such as types, subtypes, objects, and literals. It also presents the predefined operators of the language.

Chapter 4 presents the behavior style of modeling. It introduces the different sequential statements that are available and explains how they may be used to model the sequential behavior of a design. This modeling style is very similar in semantics to that of any high-level programming language. Chapter 5 describes the dataflow style of modeling. It describes concurrent signal assignment statements, and block statements, and provides examples to show how the concurrent behavior of a design may be modeled using these statements.

Chapter 6 presents the structural style of modeling. In this modeling style, a design is expressed as a set of interconnected components, possibly in a hierarchy. Component instantiation statements are explained in detail in this chapter. Chapter 7 explains the notion of an entity-architecture pair and describes how component instances can be bound to designs residing in different libraries. This chapter also explains how to pass static information into a design using generics.

Chapter 8 describes subprograms. A subprogram is a function or a procedure. The powerful concept of subprogram and operator overloading is also introduced. Chapter 9 describes packages and the design library environment as defined by the language. It also explains how items stored in one library may be accessed by a design residing in another library. Advanced features of the language such as entity statements, aliases, guarded signals, and attributes are described in Chap. 10.

Chapter 11 describes a methodology for simulating VHDL models and describes techniques for writing test benches. Examples for generating various types of clocks and waveforms and their application to the design under test are presented. Chapter 12 contains a comprehensive set of hardware modeling examples. These include among others, examples of modeling combinational logic, synchronous logic, and finite-state machines.

In all the VHDL descriptions that appear in this book, reserved words are in **boldface**. A complete list of reserved words also appears in Appendix A. Most of the language constructs are explained using easy-to-understand words, rather than through the use of formal terminology adopted in the Language Reference Manual. Also, some constructs are described only in part to explain specific features. The complete language grammar is provided in Appendix B. Appendix C contains a complete description of a package that is referred to in Chaps. 11 and 12.

In all the language constructs that appear in this book, names in *italics* indicate information to be supplied by the model writer. For example,

```
entity entity-name is  
    [ port ( list-of-interface-ports ) ]...
```

Entity, is, and port are reserved words while *entity-name* and *list-of-interface-ports* represent information to be provided by the model writer. The square brackets, [...], indicate optional items. Occasionally, ellipsis (...) are used in VHDL source to indicate code that is not relevant to that discussion. All examples that are described in this book have been validated using a native VHDL system.

Throughout this text, we shall refer to the circuit, system, design, or whatever it is that we are trying to model as the *entity*.

Tills book was prepared using the FrameMaker workstation publishing software. (FrameMaker is a registered trademark of France Technology Corporation.)

Book usage

The first time reader of this book is strongly urged to read the tutorial presented in Chap. 2. The remaining chapters are organized such that they use information from the previous chapters. However, if the tutorial is well understood, it is possible to go directly to the chapter of interest. A complete chapter has been devoted to examples on hardware modeling; this can be used as a future reference. A list of suggested readings and books on the language and the complete language syntax is provided at the end of the book. For further clarifications on the syntax and semantics of the language constructs, the reader can refer to the IEEE Standard VHDL Language Reference Manual (IEEE Std 1076-1987), published by the IEEE.

Acknowledgments

I gratefully acknowledge the efforts of several of my colleagues in making this book possible. I am especially thankful to Dinesh Bettadapur, Ray Voith, Joel Schoen, Sindhu Xirasagar, Gary Imken, Paul Harper, Oz Levia, Jeff Jones, and Gum Rao. Their constructive criticism and timely review on earlier versions of the text have resulted in several improvements in the book. A special thanks to Gary Imken for being patient enough in answering a number of questions on the idiosyncrasies of VHDL.

I would also like to thank my wife, Geetha, for reviewing the first version of this text and for providing the motivational guidance during the entire preparation of this book.

J. Bhasker
October, 1991