PREFACE

Now that you know a hardware description language (HDL), where do you go from here? As a VHDL trainer and consultant, I experienced that many engineers understand the HDL from a software viewpoint, but do not know how to approach design problems. There is a fallacy that HDL is the panacea to all design issues, and that synthesis tools will perform the magic of translating the HDL into hardware. The reality is that synthesis is just a tool to help in the implementation of what is described. It is necessary that the authors of the HDL code understand the hardware architecture and implications of what is described. A proper HDL description of an incorrect or improper architecture does not necessarily yield a correct or optimum design. The following quotation, from a contributor in *comp.lang.vhdl* newsgroup, expresses this point in an interesting manner: *HDL (e.g., VHDL or Verilog), like any other EE design tool, assumes that the user has a working knowledge of electronics and digital design. I know many who are poor designers before learning HDL and remain poor designers after taking an HDL class.*

Real Chip Design and Verification Using Verilog and VHDL addresses the practical and real aspects of logic design, processes, and verification. It incorporates a collection of FPGA and ASIC design practices, and uses Verilog and VHDL as a tool for expression of the desired architectures. This book is not intended to teach either HDL, as there are several books specifically geared toward teaching the languages. However, it provides various architectural design primitives, applications, and verification techniques, along with design methodologies and common practices.

Logic design is an art that is learned, and often relearned by designers. There are several common classes of design problems and several common classes of hardware architectures including synchronization logic, counters, controllers, arithmetic elements, and storage elements. This book addresses those classes of designs with practical examples to expose the reader to variations in styles and approaches. The architectural issues, design decomposition, and HDL code in both VHDL and Verilog are discussed and demonstrated. Transaction-based testbenches with error injection methodologies demonstrate, by example, design verification techniques. Models used for this verification task include a counter and an EDAC (error detection and correction) logic with a RAM.

This book is intended as a training book in conjunction with an HDL class as a means to demonstrate the transition of design requirements into an HDL design. Specifically, it demonstrates by example the following:

1. Styles of hardware architectures.
2. Logic design architectural decomposition process and the translation of the architectures
   into HDL.
3. HDL coding styles.
4. Verification techniques with HDL using transaction-based methodologies.

Cadence *NC-Sim* simulator and *HAL* analysis and lint checking tools were used because of their levels of efficiency, accuracy, and maturity. Cadence represents a vendor that is a leader in the EDA industry. Synplicity *Synplify Pro*® FPGA synthesis tool was also extensively used because Synplicity is recognized as a vendor of advanced, efficient, and easy to use synthesis tools targeted for FPGAs, and now ASICs. Even though these specific tools were used, almost all of the information is tool independent.

Chapter 1 provides an overview of the architectural decomposition process, and presents the classes of hardware designs. **Chapter 2** presents fundamental architectural elements used in the construction of designs. These include flip-flops, latches, synchronous edge detector, application of both edges of the clock, registers, counter styles *(e.g., Binary, One-Hot, Gray, LFSR, Johnson)*, memories including ROM, RAM, FIFO, and Error Detection and Correction (EDAC) logic. A trigonometric function defined in C, but implemented in HDL as a ROM, is also demonstrated. This chapter also addresses the importance of understanding the cell primitives and FPGA architecture including the clocking features of ASICs and FPGAs. Topics on clocking schemes and phase lock loops are discussed. **Chapter 3** addresses the synchronous/asynchronous aspects of the real world, and methods to resolve those issues. Metastability is explained, MTBF calculations are defined, and solutions in the handling of metastability are presented. The design of an asynchronous FIFO is demonstrated. The topic of crossing clock domains is also presented. **Chapter 4** addresses the verification issue and presents through two examples the transaction-based verification methodology. The topic of forcing design errors is also demonstrated in those examples, including the verification of a loadable counter and an EDAC model for a thirty-two-bit wide memory. **Chapter 5** focuses on control machines and uses a very simple CPU design to demonstrate implementation methodologies with FSM and microprogrammed solutions. **Chapter 6** addresses arithmetic intensive machines. It explains the application of SIGNED and UNSIGNED types in HDL. Verilog 1995/2001 type issues are demonstrated. **Chapter 7** explains and demonstrates mixed mode simulations and synthesis. **Chapter 8** presents a discussion on minimizing design errors and addresses miscellaneous design issues. **Chapter 9** compares Verilog to VHDL to enable users of one discipline to understand the language differences and nuances of the other discipline. It also provides Verilog coding style guidelines for VHDL and Verilog users.

All HDL code described in the book is on a companion CD. All code was verified and simulated with *NC-Sim version v03.30.1*[1]. All synthesizable code was synthesized with *Synplify Pro® version 6.2.4*[2]. The CD also includes application notes and files of practical use that were collected over a period of several years. EMACS editor for Windows, along with VHDL and Verilog modes is on CD. The CD includes data sheets and additional information on Synplicity's product line, and excellent Cadence's Verilog reference and HDL simulation documentation.

This book is intended for:

1. **Engineers.** Book provides classes of architectural examples and decomposition into HDLs. Engineers are better at copying and improving upon what is done, than from starting from scratch. This book will provide a head start in these processes.
2. **Trainers.** This book provides the focus of an advanced hardware design class using HDLs. Emphasis is on architecture, processes, methodologies, and style.
3. **College students**. Book demonstrates the hardware architectural processes.

A list of Verilog books that are often recommended includes:

Verilog HDL : A Guide to Digital Design and Synthesis, Samir Palnitkar, 396 pages, Prentice Hall 1996, ISBN: 0134516753

Verilog HDL Synthesis, A Practical Primer, J. Bhasker, 236 pages, Star Galaxy Publishing, ISBN 0-9650391-5-3

A Verilog HDL Primer, Second Edition, J. Bhasker, 1999, Star Galaxy Publishing, ISBN 0-9650391-7-X.

The Verilog Hardware Description Language, Fourth Edition, Thomas, D . E . / Moorby, Philip R , 354 Pages, Kluwer Academic Publishers 1998, 354 Pages

For information about the new features of Verilog, I recommend the book *VERILOG 2001, A Guide to the New Features of the Verilog Hardware Description Language*, Stuart Sutherland, 2002, Kluwer Academic Publishers, ISBN 0-7923-7568-8

1 Cadence Design Systems, Inc. http://www.cadence.com/
NC-Sim is available on several platforms including Win98, Win2000, WinNT4.0, Linux, Unix, HPPA.
For a guided tour of the Cadence VHDL and Verilog Desktop simulator please go to following page.
http://www.orcad.com/product/simulation/hdlsim/

2 Synplicity http://www.synplicity.com

For VHDL, I second Janick Bergeron's recommendation[3] for the book *VHDL Coding Styles and Methodologies, 2nd Edition*, 1999, Kluwer Academic Publishers, ISBN 0-7923-8474-1.

Another highly recommended book for VHDL is *The Designer's Guide to VHDL, 2nd Edition*, Peter J. Ashenden, 740 pages, Morgan Kaufmann Publishers, ISBN 1558606912

For verification, the book *Writing testbenches: Functional Verification of HDL Models*, Janick Bergeron[4] Kluwer Academic Publishers, ISBN 0-7923-7766-4 is recognized as a standard.

[3] *Writing testbenches, Functional verification of HDL models*, Janick Bergeron, Kluwer Academic Publishers 2000

[4] http://janick.bergeron.com/