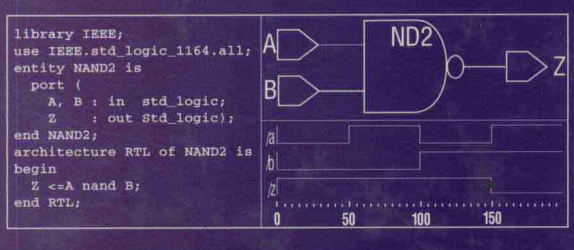
Digital Systems Design with VHDL and Synthesis

An Integrated Approach

K.C. Chang



IEEE COMPUTER SOCIETY

Library of Congress Cataloging-in-Publication Data

Chang, K. C. (Kou-Chuan), 1957 –
Digital systems design with VHDL and synthesis : an integrated approach / K.C. Chang.
p. cm.
ISBN 0-7695-0023-4
1. Electronic digital computers – Circuits – Design and construction – Data processing. 2. VHDL (Hardware description language). 3. System design – Data processing. I. Title.
TK7888.4.C435 1999
621.39 ' 2 — dc21

99-24750
CIP

IEEE Computer Society Press Order Number BP00023
Library of Congress Number 99-24750
ISBN 0-7695-0023-4

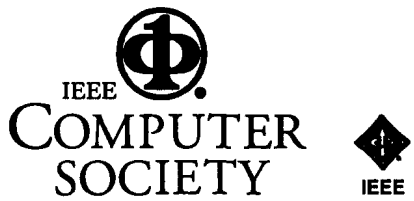
Additional copies may be ordered from:

IEEE Computer Society Press
Customer Service Center
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1314
Tel: +1-714-821-8380
Fax: +1-714-821-4641
cs.books@computer.org

IEEE Service Center
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
Tel: +1-732-981-0060
Fax: +1-732-981-9667
mis.custserv@computer.org

IEEE Computer Society
Watanabe Building
1-4-2 Minami-Aoyama
Minato-ku, Tokyo 107-0062
JAPAN
Tel: +81-3-3408-3118
Fax: +81-3-3408-3553
tokyo.ofc@computer.org

Publisher: Matt Loeb
Manager of Production, CS Press: Deborah Plummer
Advertising/Promotions: Tom Fink
Production Editor: Denise Hurst
Printed in the United States of America



Dedicated to

my parents,

Chia-Shia and Jin-Swei,

my wife,

Tsai-Wei,

and my children,

Alan, Steven, and Jocelyn

Preface

The advance of very large-scale integration (VLSI) process technologies has made "system on a chip" a feasible goal for even the most complex system. The market has demanded shorter design cycles than ever before, while design complexity continues to increase. Traditional schematic capture design approaches are no longer sufficient to be competitive in the industry. New design processes and methodologies are required to design much more complex systems within a shorter design cycle. Furthermore, the gap between the design concept and implementation must be minimized. The goal of this book is to introduce an integrated approach to digital design principles, process, and implementation. This is accomplished by presenting the digital design concepts and principles, VHDL coding, VHDL simulation, synthesis commands, and strategies together.

It is the author's belief that it is better to learn digital design concepts and principles together with the high-level design language, verification, and synthesis. Overlooking key digital design concepts will result in inefficient designs that don't meet performance requirements. Without the high level design language, the design concepts cannot be captured fast enough. Lacking the verification techniques results in a design with errors. Not knowing the synthesis strategy and methodology does not allow the design to be synthesized with the desired results. Therefore, an integrated approach to design concepts, principles, VHDL coding, verification, and synthesis is crucial. To accomplish this in a book is challenging since there are so many potential combinations of design concepts with techniques for VHDL coding, verification, and synthesis. The author has approached this challenge by focusing on the ultimate end products of the design cycle: the implementation of a digital design. VHDL code conventions, synthesis methodologies, and verification techniques are presented as tools to support the final design implementation. By taking this approach, readers will be taught to apply and adapt techniques for VHDL coding, verification, and synthesis to various situations.

To minimize the gap between design concepts and implementation, design concepts are introduced first. VHDL code is presented and explained line by line to capture the logic behind the design concepts. The VHDL code is then verified using VHDL test benches and simulation tools. Simulation waveforms are shown and explained to verify the correctness of the design. The same VHDL code is synthesized. Synthesis commands and strategies are presented and discussed. Synthesized schematics and results are analyzed for area and timing. Variations on the design techniques and common mistakes are also addressed.

This book is the result of the author's practical experience as both a designer and an instructor. Many of the design techniques and design considerations illustrated throughout the chapters are examples of viable designs. The author's teaching experi-

ence has led to a step-by-step presentation that addresses common mistakes and hard-to-understand concepts in a way that makes learning easier.

Practical design concepts and examples are presented with VHDL code, simulation waveforms, and synthesized schematics so that readers can better understand their correspondence and relationships. Unique features of the book include the following:

1. More than 207 complete examples of 14,099 lines of VHDL code are developed. Every line of VHDL code is analyzed, simulated, and synthesized with simulation waveforms and schematics shown and explained. VHDL codes, simulation waveforms, and schematics are shown together, allowing readers to grasp the concepts in the context of the entire process.
2. Every line of VHDL code has an associated line number for easy reference and discussion. The Courier font is used for the VHDL code portion because each character of this font occupies the same width, which allows them to line up vertically. This is close to how an American Standard Code for Information Interchange (ASCII) file appears when readers type in their VHDL code with any text editor.
3. The VHDL code examples are carefully designed to illustrate various VHDL constructs, features, practical design considerations, and techniques for the particular design. The examples are complete so that readers can assimilate overall ideas more easily.
4. Challenging exercises are provided at the end of each chapter so that readers can put into practice the ideas and information offered in the chapter.
5. A complete Finite Impulse Response filter (700K+ transistors) ASIC design project from concept, VHDL coding, verification, synthesis, layout, to final timing verification is provided to demonstrate the entire semi-custom standard cell design process.
6. A complete microcontroller (AM2910) design project from concept, VHDL coding, verification, synthesis, to release to a gate array vendor for layout, to final postlayout timing verification with test vector generation is used to demonstrate the entire gate array application-specific integrated circuit (ASIC) design process.
7. A complete error correction and detection circuit (TI SN54ALS616) design is taken from concept, VHDL coding, verification, synthesis, to a field programmable gate array (FPGA) implementation.
8. Synthesis commands and strategies are presented and discussed for various design situations.
9. Basic digital design circuits of counters, shifters, adders, multipliers, dividers, and floating-point arithmetic are discussed with complete VHDL examples, verification, and synthesis.
10. Test benches VHDL coding and techniques are used and demonstrated along with the design examples.
11. Preferred practices for the effective management of designs are discussed.
12. VHDL design partitioning techniques are addressed and discussed with examples.
13. An entire chapter (Chapter 8) is dedicated to clock- and reset-related circuits. Clock skew, short path, setup and hold timing, and synchronization between clock domains are discussed with examples.

14. Commonly used custom blocks such as first in, first out (FIFO), dual port random access memory (RAM), and dynamic RAM models are discussed with examples.
15. All figures in the book are available through the Internet to assist instructors.
16. VHDL code examples are available through the Internet.

0.1 ORGANIZATION OF THE BOOK

This book is divided into 15 chapters and one appendix. Appendix A is a VHDL package example that has been referenced in many examples in the chapters. The details of each chapter are as follows:

Chapter 1 describes design process and flow.

Chapter 2 describes how to write VHDL to model basic digital circuit primitives or gates. Flip-flop, latch, and three-state buffer inferences are illustrated with examples. VHDL synthesis rules are presented to provide readers with an understanding of what circuits will be synthesized from the VHDL.

Chapter 3 presents the VHDL simulation and synthesis design environments. Synopsys simulation and synthesis design environments and design tools are introduced. Mentor QuickVHDL simulation environment is also discussed. A typical design process for a block is discussed to improve the debugging process.

Chapter 4 presents VHDL modeling, synthesis, and verification of several basic combination circuits such as selector, encoder, code converter, equality checker, and comparators. Each circuit is presented with different VHDL models and their differences and trade-offs are discussed.

Chapter 5 concentrates on several binary arithmetic circuits. Half adder, full adder, ripple adder, carry look ahead adder, countone, leading zero, and barrel shifter are presented with VHDL modeling, synthesis, and test bench.

Chapter 6 discusses sequential circuits such as counters, shift registers, parallel to serial converter, and serial to parallel converter with VHDL modeling, synthesis, and test bench.

Chapter 7 presents a framework to organize registers in the design. Registers are categorized and discussed. Partition, synthesis, and verification strategies of registers are discussed. VHDL modeling, synthesis, and verification techniques are illustrated.

Chapter 8 is dedicated to clock- and reset-related circuits. The synchronization between different clock domains is discussed. The clock tree generation, clock delay, and clock skew are presented and discussed with timing diagrams and VHDL code. The issues of gated clock and clock divider are also introduced.

Chapter 9 presents examples of dual-port RAM, synchronous and asynchronous FIFO, and dynamic RAM VHDL models. These blocks are commonly used as custom drop-in macros. They are used to interact with the rest of the design so that the complete design can be verified.

Chapter 10 illustrates the complete semicustom ASIC design process of a finite impulse response ASIC design through the steps of design description, VHDL coding, functional verification, synthesis, layout, and back-annotated timing verification.

Chapter 11 discusses the concept of a microprogram controller. The design of a AMD AM2910 is presented through the gate array design process from VHDL coding to postlayout back-annotated timing verification. The test vector generation is also illustrated.

Chapter 12 discusses the principles of error and correcting Hamming codes. An actual TI EDAC integrated circuit is used as an example to design from VHDL code to all steps of FPGA design process.

Chapter 13 presents the concepts of binary fixed-point multiplication algorithms such as Booth-Wallace multiplier. The VHDL coding, synthesis, and verification are presented.

Chapter 14 discusses the concepts of binary fixed-point division algorithms. VHDL coding, synthesis, and verification are presented.

Chapter 15 discusses the floating-point number representation. Floating-point addition and multiplication algorithms are discussed and implemented in VHDL. They are verified and synthesized.

Appendix A lists a package that is used and referenced by many examples.

0.2 AUTHOR'S NOTE

This book assumes that readers have the basic knowledge of VHDL syntax, modeling, synthesis concepts, and Boolean algebra. To acquire some background in VHDL, readers are referred to my previous book (*Digital Design and Modeling with VHDL and Synthesis*, IEEE Computer Society Press, 1997) which concentrates on the complete VHDL language, syntax and coding techniques. The recommended VHDL background can be obtained from Chapters 2, 3, 4, 5, 6, 7, and 8 of that book. This book concentrates on the digital design from a basic half adder to a complete ASIC design, using VHDL coding, simulation, and synthesis techniques. The interfaces to FPGA and ASIC layout tools are also addressed.

Synopsys and Mentor Modeltech (QuickVHDL, which is similar to Modelsim) VHDL simulators, Exemplar FPGA synthesis, and Synopsys ASIC synthesis tools are used in this book. These tools are widely used and available throughout university and industry settings.

The book is intended for both practicing engineers and as a college text. Chapters are divided according to specific designs, from simple to complex. Techniques for VHDL coding, test bench, verification, synthesis, design management, and actual

design examples are good references for practicing engineers. The book also provides a robust introduction to digital design. It presents the basic design concepts of a half adder, full adder, ripple adder, carry look ahead adder, counter, shift register, multiplier, divider, floating-point arithmetic unit, error correction and detection unit, microcontroller, FIFO, dual port RAM, dynamic RAM, to a 700 thousands transistor finite impulse response filter ASIC design. These design concepts and disciplines are reinforced by actual VHDL code, verification strategies, simulation commands and waveforms, and synthesis commands and strategies. VHDL code is available for reference and practice. Previous experiences with VHDL, as suggested earlier, would be helpful. However, an understanding of the code conventions can also be acquired from this book. Most students will soon become practicing engineers. This book not only gives you a block diagram, but also, more importantly, shows you how each design concept can be captured, implemented, and realized in real life. An old Chinese saying is: "do not let your eyes and head reach higher than your hands," which is to say that it is better to be able to do it with your hands, rather than just think and see something that is out of your grasp.

I wrote this book to help the reader cope with both the design revolution that is taking place, as well as the increased demand for design complexity and a shortened design cycle.

0.3 ACKNOWLEDGMENTS

This book would have been impossible without the following people:

Ron Fernandes, who edited the entire first manuscript.

My wife, Tsai-Wei, and my children, Alan, Steven, and Jocelyn, who were patient and sacrificed many of our weekends and dinners together. I owe my children so many smiles and bedtime stories because I often got home late and tired.

My brothers, sisters, and many friends who encouraged and supported me.

Ramagopal Madamala and Tsipi Landen at Chip Express provided support in running through the place and route tools for the design in Chapter 11.

The Institute of Electrical and Electronics Engineers (IEEE) Computer Society Press editorial team, Mathew Loeb, Cheryl Baltes, and Denise Hurst contributed many valuable comments. Additionally, Joseph Pomerance did a skillful and thorough editing of the manuscript.

The author deeply appreciates all of these people.

Contents

Chapter 1

Introduction

Welcome to the book!

The advance of very large-scale integration (VLSI) process technologies has made "system on a chip" feasible for very complex systems. This is partly responsible for the market demand for much shorter design cycles even though design complexity has continued to increase.

There are many design and VHDL code examples, simulation waveforms, and synthesized schematics shown in this book to illustrate their correspondence. VHDL code can be downloaded from the Internet for exercises. All examples have been verified with VHDL simulation and synthesis tools to ensure high fidelity of the VHDL code. All VHDL codes are listed line by line for reference and discussion in the text. Each VHDL example is complete, no fragments so that the complete picture is clear to you. Design techniques of VHDL coding, verification, and synthesis that implement the design concepts and principles are illustrated with actual examples throughout the book.

In the next section, an integrated design process and methodology is introduced to serve as our base line design flow and approach.

1.1 INTEGRATED DESIGN PROCESS AND METHODOLOGY

Figure 1-1 shows a flow chart that describes an integrated VHDL design process and methodology. The design is first described in VHDL. VHDL coding techniques will be applied to model the design efficiently and effectively. A preliminary synthesis for each VHDL code is done to check for obvious design errors such as unintended latches and insufficient sensitivity list before the design is simulated. This also ensures that all VHDL code can be synthesized. This process is discussed and illustrated in Chapter 2 as the VHDL design process for a block. A test bench is developed to facilitate the design verification. Test bench examples and verification techniques are discussed throughout the book for various designs. A more detailed synthesis can be started (in parallel with the test bench development) to calibrate design and timing constraints. The design architecture may require a change, based on the timing result. The complete design can be verified with the test bench. There will be iterations

among the VHDL coding, synthesis, test bench, and simulation. After the design is free of design errors, the complete and detailed synthesis is done. A netlist can be generated for the layout tool to do the layout. The layout may be performed with Field Programmable Gate Array (FPGA), gate array, or standard cell layout tools. The layout tool can generate another VHDL netlist with the timing delay file in Standard Delay Format (SDF), which can be used for simulation with the test bench. Using the simulation results, test vectors can be generated for the foundry to test the fabricated ASIC design. A FPGA design does not need the test vector. The layout can be performed in-house or by a vendor. For the ASIC design, the layout tool can generate a GDSII file to describe the masks for fabrication.

The synthesis tool can also generate a VHDL netlist for postsynthesis and prelayout simulation. This is not common. The postlayout simulation is preferred since the timing generated from the layout tools is much closer to the actual hardware. Note also that the test bench should be used for both functional simulation and postlayout timing simulation without the need to change.

In this book, many examples are used to illustrate all the steps of the process described in Figure 1-1.

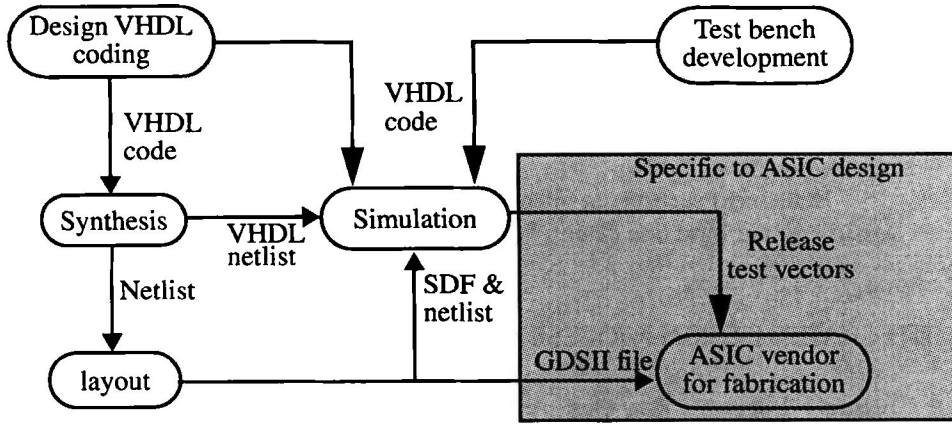


Figure 1-1 VHDL design process and methodology.

1.2 BOOK OVERVIEW

Chapter 2 describes how to write VHDL to model basic digital circuit primitives or gates. Flip-flops, latches, and three-state buffers inference is illustrated with examples. VHDL synthesis rules are presented to provide guidelines what circuits will be synthesized by synthesis tools.

Chapter 3 presents the VHDL simulation and synthesis design environments. Synopsys simulation and synthesis design environments and design tools are introduced. Mentor QuickVHDL simulation environment is also discussed. A typical design process for a block is discussed to improve the debugging process.

Chapter 4 presents VHDL modeling, synthesis, and verification of several basic combination circuits such as selector, encoder, code converter, equality checker, and comparators. Each circuit is presented with different VHDL models so that their differences and trade-offs are discussed.

Chapter 5 concentrates on several binary arithmetic circuits. Half adder, full adder, ripple adder, carry look ahead adder, count one, leading zero, and barrel shifter are presented with VHDL modeling, synthesis, and test bench.

Chapter 6 discusses sequential circuits such as counters, shift registers, parallel to serial converter, and serial to parallel converter with VHDL modeling, synthesis, and test bench.

Chapter 7 presents a framework to organize registers in the design. Registers are categorized and discussed. Partition, synthesis, and verification strategies of registers are discussed. VHDL modeling, synthesis, and verification are illustrated.

Chapter 8 is dedicated to clock- and reset-related circuits. The synchronization between different clock domains is discussed. Clock tree generation, clock delay, and clock skew are presented and discussed with timing diagrams and VHDL code. The issues of gated clock and clock divider are also introduced.

Chapter 9 presents examples of dual-port RAM, synchronous and asynchronous FIFO, and dynamic RAM VHDL models. These blocks are commonly used as custom drop-in macros. They are used to interact with the rest of the design so that the complete design can be verified.

Chapter 10 illustrates the complete semicustom ASIC design process of a Finite Impulse Response ASIC design through the steps of design description, VHDL coding, functional verification, synthesis, layout, back-annotated timing verification.

Chapter 11 discusses the concept of a microprogram controller. The design of a AMD AM2910 is presented through the gate array design process from VHDL coding to postlayout back-annotated timing verification. Test vector generation is also illustrated.

Chapter 12 discusses the principles of error and correcting Hamming codes. An actual TI EDAC integrated circuit is used as an example to design from VHDL code to all steps of FPGA design process.

Chapter 13 presents the concepts of binary fixed-point multiplication algorithms such as Booth-Wallace multiplier. The VHDL coding, synthesis, and verification are presented.

Chapter 14 discusses the concepts of binary fixed-point division algorithms. VHDL coding, synthesis, and verification are presented.

Chapter 15 discusses the floating-point number representation. Floating-point addition and multiplication algorithms are discussed and implemented with VHDL codes. They are verified and synthesized.