Designing Digital Computer Systems with Verilog

This unique book serves both as an introduction to computer architecture and as a guide to using a hardware description language (HDL) to design, model and simulate real digital systems. The book starts with an introduction to Verilog: the HDL chosen for the book since it is widely used in industry and straightforward to learn. Next, the instruction set architecture (ISA) for the simple VeSPA (Very Small Processor Architecture) processor is defined; this processor has been simulated and thoroughly tested at the University of Minnesota by the authors. The VeSPA ISA is used throughout the remainder of the book to demonstrate how behavioral and structural models can be developed and intermingled in Verilog. Although Verilog is used throughout, the lessons learned will be equally applicable to other HDLs. Written for senior and graduate students, this book is also an ideal introduction to Verilog for practicing engineers. A companion website is available with the Verilog source code for all of the examples in the text, Verilog source code for the VeSPA processor, and additional software to assist in using the VeSPA simulations. See www.cambridge.org/052182866X for details.

DAVID LILJA is a professor of electrical and computer engineering, and a fellow of the Minnesota Supercomputing Institute, at the University of Minnesota in Minneapolis. He also serves as a member of the graduate faculties in computer science and scientific computation, and was the founding director of graduate studies for computer engineering. He has served on the program committees of numerous conferences and as associate editor for IEEE Transactions on Computers. David is a senior member of the IEEE and a member of the ACM.

SACHIN SAPATNEKAR is the Robert and Marjorie Henle Professor in the Department of Electrical and Computer Engineering at the University of Minnesota, and serves on the graduate faculty in computer science and engineering. He has served as associate editor for several IEEE journals, and has been a distinguished visitor for the IEEE Computer Society, and a distinguished lecturer for the IEEE Circuits and Systems Society. He is a recipient of the NSF Career Award and the SRC Technical Excellence Award. He is a fellow of the IEEE and a member of the ACM.

Designing Digital Computer Systems with Verilog

David J. Lilja and Sachin S. Sapatnekar
Department of Electrical and Computer Engineering
University of Minnesota
Minneapolis

CAMBRIDGE UNIVERSITY PRESS

PUBLISHED BY THE PRESS SYNDICATE OF THE UNIVERSITY OF CAMBRIDGE
The Pitt Building, Trumpington Street, Cambridge, United Kingdom

CAMBRIDGE UNIVERSITY PRESS
The Edinburgh Building, Cambridge CB2 2RU, UK
40 West 20th Street, New York, NY 10011–4211, USA
477 Williamstown Road, Port Melbourne, VIC 3207, Australia
Ruiz de Alarcón 13, 28014 Madrid, Spain
Dock House, The Waterfront, Cape Town 8001, South Africa

http://www.cambridge.org

First published 2005

Printed in the United Kingdom at the University Press, Cambridge

Typeface Times 10.5/14pt. *System* LaTeX 2ε

A catalog record for this book is available from the British Library

Library of Congress Cataloging in Publication data

Lilja, David J.
Designing digital computing systems with Verilog/David J.Lilja and Sachin S. Sapatnekar.
p. cm.
Includes bibliographical references and index.
ISBN 0 521 82866 X (alk. paper)
1. Verilog (Computer hardware description language) 2. Electronic digital computers – Design and Construction. I. Sapatnekar, Sachin S., 1967 – II. Title.

TK7885.7.L55 2005
621.39'2 – dc22 2004054515

ISBN 0 521 82866 X hardback

Contents

Preface

To the engineer, all matter in the universe can be placed into one of two categories: (1) things that need to be fixed, and (2) things that will need to be fixed after you've had a few minutes to play with them.

Quoted by Scott Adams in The Dilbert Principle.

Hardware description languages (HDLs) are used extensively in industry to design digital systems ranging from microprocessors to components within consumer appliances, such as cellular telephones. These languages allow engineers to quickly and precisely specify and document their designs in a high-level language that has strong similarities to conventional programming languages, such as Java, C, and C++. Automatic tools exist to simulate the design using this high-level description and, ultimately, to translate the design from the hardware description language into an actual silicon chip.

There are two primary hardware description languages in use by industry today, Verilog and VHDL. Both languages solve the same basic problem, but in slightly different ways. There are strong adherents to both languages and arguments about which is better often seem to have the feel of a religious war. The Institute of Electrical and Electronic Engineers (IEEE) made Verilog an international standard in 1995 and it continues to be refined and enhanced.

Numerous books have been written to describe both languages. These books range from the IEEE standards document, to tutorial texts that introduce the language to novices, to texts for advanced users that describe the many subtle aspects of the languages. However, these books tend to treat the languages as just another programming language to be learned. They tend to miss the forest for the trees since they fail to view the broader goals of design as they focus on the details of describing specific circuits rather than considering the larger picture that encompasses architectural choices and their impact on circuit design, and vice versa.

The philosophy behind this book is to view the language as only a tool in the overall design process used to produce the final product, which is the piece of silicon that is a part of a larger system. This book will teach the reader how to design a complex digital system using the Verilog HDL as the vehicle for modeling and simulating the design. We have chosen to focus on Verilog due to its economic

importance and due to its popularity and widespread use among the designers of processors and other digital systems within the computing industry. Our philosophy is to teach the components of the language that are necessary for the design of our example processor simultaneously with the teaching of the overall design process.

Organization

This book begins with a brief introduction to the hierarchical design process that is used throughout the book to control the complexity involved in designing a large-scale digital system. Chapter 2 then introduces the Verilog language using several relatively small examples. The instruction set architecture (ISA) for the simple VeSPA processor is defined in Chapter 3 along with a running commentary explaining the reasoning behind the various trade-offs that must be made during the design of any system. This ISA is used throughout the remainder of the book to demonstrate how a behavioral model of the processor can be developed (Chapter 4) and then refined into a detailed pipelined implementation (Chapter 7). Along the way, we describe how an assembler is constructed to translate assembly language programs into machine language (Chapter 5) and the general concepts involved in pipelining a processor design (Chapter 6). The book concludes with a discussion of several techniques that are used to verify the correctness of the final design (Chapter 8). Appendix A provides a concise summary of the VeSPA instruction set, while Appendix B presents the details of the VeSPA assembler.

At the end of some chapters, we provide suggestions for further reading for those who are interested in pursuing a particular topic in more detail. There is also a companion web site for this book (www.cambridge.org/052182866X) with the Verilog source code for all of the examples used in the book, plus some additional tools that can be used with the VeSPA processor models, such as the assembler.

Suggestions for using this text

This book was written primarily with the following three uses in mind:

1. **As a supplemental text for an undergraduate course in computer architecture.** Popular existing computer architecture textbooks tend to base their discussions around simple processors designed specifically for the needs of the given textbook. What these books often fail to do, however, is to tie the overall design process into the hardware description language methodology that engineers actually use in industry. This book provides a useful supplement to a computer architecture course to show how a hardware description language is actually used to design

a processor from start to finish. For example, a course instructor could assign students the task of modifying the VeSPA processor presented in this book as a series of homework assignments throughout the term.

2. **As the primary text for a computer design laboratory course.** This book would be useful as the text for a computer design laboratory course. This type of course may be offered independently, or as a formal component of a computer architecture course, as suggested above.
3. **As a self-teaching guide for graduate engineers.** The book has a very tutorial flavor. This approach makes it useful for practicing engineers who want a self-study text to update their skills in the area of digital systems design. It would also be appropriate for someone who knows VHDL, for instance, but needs to learn Verilog.

Acknowledgements

The development of any book requires the efforts of more than just the authors. And this book is no exception.

We would particularly like to acknowledge the contributions of Saurabh Dighe to the design of the pipelined implementation of VeSPA presented in Chapter 7. Saurabh also provided substantial help in generating the figures used in this chapter.

We also would like to thank Jerry Sobelman for his careful reading of an early draft of the book, and his courage in using this draft while teaching a course on computer organization and design. He pointed out numerous errors and confusing explanations, which we hope we have corrected satisfactorily.

The numerous students who muddled through our very early attempts at teaching our computer organization course using some rather raw versions of the VeSPA design and this text deserve both our apologies and sincere thanks. Their suggestions and feedback helped us improve both the processor design and our writing.

Finally, we thank the anonymous reviewers of our original proposal to develop this book for their insightful comments and specific suggestions for focusing some of the explanations and discussion.

While all of these people made substantial contributions to the outcome of this book, any errors that remain are our responsibility.