THE VERILOG[®] HARDWARE DESCRIPTION LANGUAGE

THE VERILOG[®] HARDWARE DESCRIPTION LANGUAGE

by

Donald E. Thomas
Carnegie Mellon University

and

Philip R. Moorby
Cadence Design Systems, Inc.

SPRINGER SCIENCE+BUSINESS MEDIA, LLC

Library of Congress Cataloging-in-Publication Data

Thomas, D.E. (Donald E.), 1951-

The Verilog hardware description language / by Donald E. Thomas and Phillip R. Moorby.

p. cm.

Includes index.

ISBN 978-1-4613-6784-0 ISBN 978-1-4615-3992-6 (eBook)

DOI 10.1007/978-1-4615-3992-6

1. Verilog (Computer hardware description language) I. Moorby,
Phillip R., 1953- . II. Title.

TK7885.7.T48 1991

621.39'2--dc20

90-48370

CIP

Verilog® is a registered trade mark of Cadence Design Systems, Inc.

Copyright 1991 by Springer Science+Business Media New York

Originally published by Kluwer Academic Publishers in 1991

Softcover reprint of the hardcover 1st edition 1991

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, mechanical, photo-copying, recording, or otherwise, without the prior written permission of the publisher, Kluwer Academic Springer Science+Business Media, LLC.

Printed on acid-free paper.

*To Sandie and Teresa,
and John and Holland,
and Jill.*

Table of Contents

Preface.....	xiii
---------------------	-------------

Acknowledgements	xv
-------------------------------	-----------

1. Verilog -- A Tutorial Introduction	1
--	----------

1.1	Describing Digital Systems	1
1.2	Getting Started.....	2
1.2.1	A Structural Description	2
1.2.2	Simulating the NAND Latch	4
1.3	Module Hierarchy	7
1.3.1	The Counter.....	8
1.3.2	Components of the Counter	10
1.3.3	A Clock for the System.....	11
1.3.4	Tying the Whole Circuit Together.....	12
1.4	Behavioral Modeling.....	14
1.4.1	A Behavioral Model of the m16 Counter	14
1.4.2	Mixing Structure and Behavior.....	16
1.4.3	Assignment Statements.....	21
1.4.4	Mixing Behavioral and Structural Descriptions.....	22
1.5	Summary	23
1.6	Exercises.....	23

2. Behavioral Modeling Constructs	25
--	-----------

2.1	Process Model	25
2.2	If-Then-Else.....	26
2.2.1	Where Does The ELSE Belong?	31

2.2.2	The Conditional Operator.....	32
2.3	Loops.....	33
2.3.1	Four Basic Loop Statements.....	33
2.3.2	Exiting Loops on Exceptional Conditions.....	36
2.4	Multi-way branching.....	37
2.4.1	If-Else-If.....	37
2.4.2	Case.....	38
2.4.3	Comparison of Case and If-Else-If.....	40
2.4.4	CaseZ and CaseX.....	41
2.5	Functions and Tasks.....	42
2.5.1	Tasks.....	43
2.5.2	Functions.....	45
2.5.3	A Structural View.....	47
2.6	Summary.....	49
2.7	Exercises.....	49
3.	Concurrent Process Statements	51
3.1	Concurrent Processes.....	51
3.2	Events.....	53
3.2.1	Event Control Statement.....	53
3.2.2	Named Events.....	55
3.2.3	An Example of Hierarchical Names.....	58
3.3	The Wait Statement.....	58
3.3.1	A Complete Producer-Consumer Handshake	60
3.3.2	Comparison of the Wait and While Statements.....	64
3.3.3	Comparison of Wait and Event Control Statements	65
3.4	Disabling Named Blocks.....	65
3.5	Quasi-continuous assignment.....	67
3.6	Sequential and Parallel Blocks.....	69
3.7	Exercises.....	70
4.	Logic Level Modeling	73
4.1	Introduction.....	73
4.2	Logic Gates and Nets	74
4.2.1	Modeling Using Primitive Logic Gates.....	74
4.2.2	Four-Level Logic Values.....	77
4.2.3	Nets.....	78
4.2.4	Module Port Specifications.....	81
4.3	Continuous Assignment	82
4.3.1	Behavioral Modeling of Combinational Circuits.....	83
4.3.2	Net and Continuous Assign Declarations	85
4.4	Parameterized Definitions.....	87
4.5	Logic Delay Modeling.....	89

4.5.1	A Gate Level Modeling Example.....	89
4.5.2	Gate and Net Delays.....	91
4.5.3	Minimum, Typical, and Maximum Delays.....	93
4.6	Delay Paths Across a Module.....	94
4.7	Summary	96
4.8	Exercises.....	96
5.	Defining Gate Level Primitives.....	99
5.1	Combinational Primitives.....	99
5.1.1	Basic Features of User-Defined Primitives	99
5.1.2	Describing Combinational Logic Circuits.....	101
5.2	Level- and Edge-Sensitive Sequential Primitives.....	103
5.2.1	Level-Sensitive Primitives.....	104
5.2.2	Edge-Sensitive Primitives.....	104
5.3	Shorthand Notation	105
5.4	Mixed Level- and Edge-Sensitive Primitives.....	107
5.5	Summary	110
5.6	Exercises.....	110
6.	Switch Level Modeling	113
6.1	A Dynamic MOS Shift Register Example.....	113
6.2	Switch Level Modeling	118
6.2.1	Strength Modeling.....	118
6.2.2	Strength Definitions.....	121
6.2.3	An Example Using Strengths.....	123
6.2.4	Resistive MOS Gates.....	124
6.3	Ambiguous Strengths.....	126
6.3.1	Illustrations of Ambiguous Strengths	127
6.3.2	The Underlying Calculations.....	128
6.4	Summary	133
6.5	Exercises.....	133
7.	Two Large Examples	135
7.1	The miniSim Example.....	135
7.1.1	Overview.....	135
7.1.2	The miniSim Source.....	136
7.1.3	Simulation Results	144
7.2	The 8251A Example.....	146
7.2.1	Overview.....	146
7.2.2	The 8251A Source.....	146
7.3	Exercises.....	168
	Appendix A. Lexical Conventions.....	169

A.1	White Space and Comments	169
A.2	Operators.....	169
A.3	Numbers	170
A.4	Strings.....	171
A.5	Identifiers, System Names, and Keywords.....	172

Appendix B. Verilog Operators.....175

B.1	Table of Operators.....	175
B.2	Operator Precedence.....	179
B.3	Operator Truth Tables.....	180
B.3.1	Bitwise AND.....	180
B.3.2	Bitwise OR.....	181
B.3.3	Bitwise XOR.....	181
B.3.4	Bitwise XNOR.....	181
B.4	Expression Bit Lengths.....	182

Appendix C. Verilog Gate Types.....183

C.1	Logic Gates.....	183
C.2	BUF and NOT Gates.....	185
C.3	BUFIF and NOTIF Gates	185
C.4	MOS Gates	187
C.5	Bidirectional Gates.....	188
C.6	CMOS Gates.....	188
C.7	Pullup and Pulldown Gates.....	188

Appendix D. Registers, Memories, Integers, and Time..189

D.1	Registers.....	189
D.2	Memories	190
D.3	Integers and Times.....	190

Appendix E. System Tasks and Functions193

E.1	Display and Write Tasks.....	193
E.2	Continuous Monitoring	194
E.3	Strobed Monitoring	195
E.4	File Output	195
E.5	Simulation Time.....	196
E.6	Stop and Finish.....	196
E.7	Random.....	197

Appendix F. Formal Syntax Definition.....199

F.1	Source Text.....	200
F.2	Declarations.....	203

F.3	Primitive Instances	205
F.4	Module Instantiations	206
F.5	Behavioral Statements	207
F.6	Specify Section.....	209
F.7	Expressions.....	212
F.8	General.....	214
Index		217

Preface

The Verilog language is a hardware description language which 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 level of abstraction. The language provides hierarchical constructs, allowing the designer to control the complexity of a description.

Verilog was originally designed in the winter of 1983/84 as a proprietary verification/simulation product. Since then, several other proprietary analysis tools have been developed around the language, including a fault simulator and a timing analyzer; the language being instrumental in providing consistency across these tools. Now, the language is openly available for any tool to read and write. This book introduces 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. Where possible, we have stayed away from simulator-specific details and concentrated on design specification, but have included enough information to be able to have working executable models.

The book takes a tutorial approach to presenting the language. Indeed, we start with a tutorial introduction which presents, via examples, the major features of the language. We then continue with a more complete discussion of the language constructs. Numerous examples are provided to allow the reader to easily learn (and re-

learn!) by example. Finally, in the appendix we provide a formal description of the language. Overall, our approach is to provide a means of learning by observing the examples, and doing the exercises.

We have provided a set of exercises to stimulate thought while reading the book. It is strongly recommended that you try the exercises as early as possible with the aid of the Verilog simulator. Or, if you have your own designs, or some from a data book, try them out too. The examples shown in the book are available in electronic form from Cadence Design Systems, Inc. by sending a request to receive the examples to the e-mail address "thomasmoorbybook@cadence.com".

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 provides enough information for students in an introductory logic design course to make simple use of logic simulation as part of their laboratory experience. The rest of the book could then be used in upper level logic design and architecture courses.

The book is organized into seven chapters and six appendices. We start with the tutorial introduction to the language in chapter 1. Chapters 2 and 3 present the language's behavioral modeling constructs. Chapters 4 and 5 then present logic level modeling, and chapter 6 presents the more advanced topic of switch level modeling. Finally, chapter 7 shows sizable examples of the use of the language. The appendices are reserved for the dryer topics typically found in a language manual. Read them at your own risk.

D. E. Thomas
P. R. Moorby

Acknowledgements

The authors would like to acknowledge the many people in Cadence Design Systems, Inc. and the customers of the Verilog-based products who have contributed to the continuing development of the Verilog language. In particular, the authors would like to thank Leigh Brady for her help in reviewing the manuscript.