MODELING, SYNTHESIS, AND RAPID PROTOTYPING WITH THE VERILOG™ HDL

Michael D. Ciletti
University of Colorado, Colorado Springs

"THE SOFTWARE PREVIOUSLY AVAILABLE WITH THIS TEXT IS NOW AVAILABLE FOR PURCHASE SEPARATELY UNDER ISBN 0131858394."

Prentice Hall
Upper Saddle River, New Jersey 07458
CONTENTS

PREFACE, xix

Chapter 1
INTRODUCTION TO ELECTRONIC DESIGN AUTOMATION, 1

1.1 ELECTRONIC DESIGN AUTOMATION, 1
1.1.1 Design Flow, 2
1.1.2 Design Entry, 6
1.1.3 Schematic-Based Design Entry, 6
1.1.4 Design Entry Based on a Hardware Description Language, 8
1.1.5 Hints for Model Development, 12

1.2 A BRIEF HISTORY OF HDLs, 13

1.3 THE ROLE AND REQUIREMENTS OF HARDWARE DESCRIPTION LANGUAGES IN EDA, 14

1.4 BENEFITS OF USING HDLs IN EDA, 16

1.5 SUMMARY, 18

REFERENCES, 19

PROBLEMS, 20

Chapter 2
HARDWARE MODELING WITH THE VERILOG HDL, 22

2.1 HARDWARE ENCAPSULATION: THE VERILOG MODULE, 23
2.1.1 Module Ports, 26
2.1.2 Module Implementation, 26
Chapter 2
HARDWARE MODELING: VERILOG PRIMITIVES, 28
DESCRIPTIVE STYLES, 32
- Explicit Structural Description, 33
- Implicit Structural Description—Continuous Assignments, 34
- Multiple Instantiations and Assignments, 36
STRUCTURAL CONNECTIONS, 37
- Module Port Connections, 37
- Primitive Terminal Connections, 39
- Empty Port Connections, 39
BEHAVIORAL DESCRIPTIONS IN VERILOG, 40
- RTL/Data Flow Descriptions, 40
- Algorithm-Based Descriptions, 41
HIERARCHICAL DESCRIPTIONS OF HARDWARE, 44
STRUCTURED (TOP-DOWN) DESIGN METHODOLOGY, 45
ARRAYS OF INSTANCES, 48
USING VERILOG FOR SYNTHESIS, 51
LANGUAGE CONVENTIONS, 56
REPRESENTATION OF NUMBERS, 56
SUMMARY, 58
REFERENCES, 58
PROBLEMS, 58

Chapter 3
EVENT-DRIVEN SIMULATION AND TESTBENCHES, 63
SIMULATION WITH VERILOG, 63
- Event-Driven Simulation, 64
- Simulation Data Structures, 65
- Effect of Propagation Delay, 66
- Inertial Delay and Event De-scheduling, 67
DESIGN UNIT TESTBENCH, 70
SUMMARY, 78
PROBLEMS, 78
Chapter 4
LOGIC SYSTEM, DATA TYPES, AND OPERATORS
FOR MODELING IN VERILOG HDL, 81

4.1 VARIABLES, 81
4.2 LOGIC VALUE SET, 82
4.3 DATA TYPES, 83
  4.3.1 Net Data Types, 84
  4.3.2 Initial Value of a Net, 88
  4.3.3 Wired Logic, 89
  4.3.4 Undeclared Nets—Default Net Type, 90
  4.3.5 Register Data Types, 91
  4.3.6 Initial Value of a Register Variable, 92
  4.3.7 Undeclared Register Variables, 92
  4.3.8 Addressing Register Variables, 92
  4.3.9 Passing Variables Through Ports, 92
  4.3.10 Two-Dimensional Arrays (Memories), 93
  4.3.11 Data Type: integer, 95
  4.3.12 Data Type: real, 95
  4.3.13 Data Type: time, 95
  4.3.14 Data Type: realtime, 96
  4.3.15 Scope of a Variable, 96
  4.3.16 Variable References and Hierarchical
    De-referencing, 96

4.4 STRINGS, 98
4.5 CONSTANTS, 98
  4.5.1 Direct Substitution of Parameters, 99
  4.5.2 Indirect Substitution of Parameters, 100

4.6 OPERATORS, 100
  4.6.1 Arithmetic Operators, 101
  4.6.2 Bitwise Operators, 102
  4.6.3 Reduction Operators, 104
  4.6.4 Logical Operators, 104
  4.6.5 Relational Operators, 105
  4.6.6 Shift Operators, 106
  4.6.7 Conditional Operator, 107
  4.6.8 Concatenation Operator, 108

4.7 EXPRESSIONS AND OPERANDS, 109
Chapter 7
BEHAVIORAL DESCRIPTIONS IN VERILOG HDL, 159

7.1 VERILOG BEHAVIORS, 162
7.2 BEHAVIORAL STATEMENTS, 163
7.3 PROCEDURAL ASSIGNMENT, 166
7.4 PROCEDURAL CONTINUOUS ASSIGNMENT, 167
  7.4.1 assign ... deassign Procedural Continuous Assignment, 168
  7.4.2 force ... release Procedural Continuous Assignment, 171
7.5 PROCEDURAL TIMING CONTROLS AND SYNCHRONIZATION, 172
  7.5.1 Delay Control Operator (#), 173
  7.5.2 Event Control Operator (@), 175
  7.5.3 Event or, 178
  7.5.4 Named Events, 180
  7.5.5 The wait Construct, 183
7.6 INTRA-ASSIGNMENT DELAY—BLOCKED ASSIGNMENTS, 184
7.7 NON-BLOCKING ASSIGNMENT, 185
7.8 INTRA-ASSIGNMENT DELAY: NON-BLOCKING ASSIGNMENT, 187
7.9 SIMULATION OF SIMULTANEOUS PROCEDURAL ASSIGNMENTS, 193
7.10 REPEATED INTRA-ASSIGNMENT DELAY, 196
7.11 INDETERMINATE ASSIGNMENTS AND AMBIGUITY, 198
7.12 CONSTRUCTS FOR ACTIVITY FLOW CONTROL, 202
7.12.1 Activity Flow Control: Conditional Operator (? ... :), 203
7.12.2 Activity Flow Control: The case Statement (case, casex, casez), 203
7.12.3 Activity Flow Control: Conditional Statement (if ... else), 206
7.12.4 Activity Flow Control: Loops, 208
7.12.4.1 The repeat Loop, 208
7.12.4.2 The for Loop, 209
7.12.4.3 The while Loop, 216
7.12.4.4 The forever Loop, 218
7.12.4.5 Comparison of Loops, 219
7.12.4.6 Comparison of “always” and “forever”, 219
7.12.5 The disable Statement, 219
7.12.6 Parallel Activity Flow: The fork ... join Statement, 220
7.12.7 Race Conditions and the fork ... join Statement, 222

7.13 TASKS AND FUNCTIONS, 223
7.13.1 Tasks, 223
7.13.2 Rules for Tasks, 225
7.13.3 Functions, 225

7.14 SUMMARY OF DELAY CONSTRUCTS IN-VERILOG, 229

7.15 SYSTEM TASKS FOR TIMING CHECKS, 230
7.15.1 Setup and Hold Conditions ($setup, $hold, $setuphold), 231
7.15.2 Signal Period ($period), 233
7.15.3 Minimum Pulse Width ($width), 234
7.15.4 Signal Skew ($skew), 234
7.15.5 Recovery Time ($recovery), 235
7.15.6 No Signal Change ($nochange), 235
7.15.7 Edge Semantics for Timing Checks, 235
7.15.8 Conditioned Events for Timing Checks, 236

7.16 VARIABLE SCOPE REVISITED, 237

7.17 MODULE CONTENTS, 237

7.18 BEHAVIORAL MODELS OF FINITE STATE MACHINES, 238
7.18.1 Explicit Finite State Machines, 239
7.18.2 Implicit Finite State Machines, 255
7.18.3 Handshaking, 260

7.19 SUMMARY, 263
Chapter 8
SYNTHESIS OF COMBINATIONAL LOGIC, 281

8.1 HDL-BASED SYNTHESIS, 281
8.1.1 Logic Synthesis, 284
8.1.2 Register Transfer Level (RTL) Synthesis, 292
8.1.3 Finite State Machine Synthesis, 292
8.1.4 Behavioral Synthesis, 292

8.2 TECHNOLOGY-INDEPENDENT DESIGN, 293

8.3 BENEFITS OF SYNTHESIS, 295

8.4 SYNTHESIS METHODOLOGY, 296

8.5 VENDOR SUPPORT, 296
8.5.1 Commonly-Supported Verilog Constructs, 297
8.5.2 Unsupported and Ignored Constructs, 297

8.6 STYLES FOR SYNTHESIS OF COMBINATIONAL LOGIC, 298
8.6.1 Combinational Synthesis from a Netlist of Primitives, 299
8.6.2 Combinational Synthesis from UDPs, 303
8.6.3 Combinational Synthesis from Continuous Assignments, 304
8.6.4 Combinational Synthesis from a Cyclic Behavior, 305
8.6.5 Combinational Synthesis from a Function or Task, 309
8.6.6 Combinational Synthesis from Interconnected Modules, 310
8.6.7 Constructs to Avoid in Combinational Synthesis, 310
8.6.8 Simulation Efficiency and Procedural Continuous Assignments, 311
8.6.9 Synthesis of Control Logic for Multiplexed Datapaths, 312
8.6.10 Unexpected and Unwanted Latches, 317
8.6.11 Synthesis of Priority Structures, 323
8.6.12 Treatment of Default Conditions, 324

8.7 TECHNOLOGY MAPPING AND SHARED RESOURCES, 330
8.8 THREE-STATE BUFFERS, 333
   8.8.1 Buses, 334
   8.8.2 Bi-directional Bus Drivers, 334
   8.8.3 Bus Loading, 335
8.9 THREE-STATE OUTPUTS AND DON'T-CARES, 336
8.10 SUMMARY, 338
REFERENCES, 338
PROBLEMS, 339

Chapter 9
SYNTHESIS OF SEQUENTIAL LOGIC, 345
  9.1 SYNTHESIS OF SEQUENTIAL UDPs, 345
  9.2 SYNTHESIS OF LATCHES, 348
  9.3 SYNTHESIS OF EDGE-TRIGGERED FLIP-FLOPS, 354
  9.4 REGISTERED COMBINATIONAL LOGIC, 358
  9.5 SHIFT REGISTERS AND COUNTERS, 361
  9.6 SYNTHESIS OF FINITE STATE MACHINES, 378
     9.6.1 Modeling the Combinational Logic of an FSM, 378
     9.6.2 Synthesis of Explicit State Machines, 385
     9.6.3 Synthesis of Implicit Finite State Machines, 386
     9.6.4 Pitfalls in Modeling State Machines, 392
     9.6.5 Some Rules for Implicit FSMs, 404
     9.6.6 Comparison of Explicit and Implicit FSMs, 409
     9.6.7 State Encoding and One-Hots, 409
  9.7 RESETS, 412
  9.8 SYNTHESIS OF GATED CLOCKS, 414
  9.9 DESIGN PARTITIONS AND HIERARCHICAL STRUCTURES, 415
  9.10 SUMMARY, 417
  9.10 PROBLEMS, 417

Chapter 10
SYNTHESIS OF LANGUAGE CONSTRUCTS, 425
  10.1 SYNTHESIS OF NETS, 425
  10.2 SYNTHESIS OF REGISTER VARIABLES, 426
     10.2.1 Synthesis of Integers, 427
10.2.2 Synthesis of real, time and realtime Variables., 427
10.2.3 Synthesis of Memories (Arrays), 427
10.2.4 Synthesis of Strings, 429

10.3 RESTRICTIONS ON SYNTHESIS OF "X" AND "Z", 429

10.4 SYNTHESIS OF EXPRESSIONS AND OPERATORS, 429
10.4.1 Synthesis of Arithmetic Operators, 429
10.4.2 Synthesis of Non-Arithmetic Operators, 431
10.4.3 Synthesis of Shift Operators, 431
10.4.4 Synthesis of Relational and Identity Operators, 434
10.4.5 Synthesis of Reduction, Bitwise, and Logical Operators, 437
10.4.6 Conditional Operator, 444
10.4.7 Synthesis of the Concatenation Operator, 445
10.4.8 Grouping of Operators, 445

10.5 SYNTHESIS OF ASSIGNMENTS, 446
10.5.1 Synthesis of Continuous Assignments, 446
10.5.2 Synthesis of Procedural Assignments, 446
10.5.3 Expression Substitution in Procedural Assignments, 447
10.5.4 Synthesis of Non-Blocking Assignments, 450
10.5.5 More Pitfalls in Modeling State Machines, 455

10.6 SYNTHESIS OF case AND CONDITIONAL (if...) STATEMENTS, 457

10.7 SYNTHESIS OF RESETS, 461

10.8 TIMING CONTROLS IN SYNTHESIS, 463
10.8.1 Delay Controls, 463
10.8.2 Event Controls, 463
10.8.3 Multiple Event Controls, 464
10.8.4 Synthesis of the wait Statement, 465
10.8.5 Synthesis of Named Events, 465

10.9 SYNTHESIS OF MULTI-CYCLE OPERATIONS, 467

10.10 SYNTHESIS OF LOOPS, 469
10.10.1 Static Loops Without Internal Timing Controls, 470
10.10.2 Static Loops With Internal Timing Controls, 475
10.10.3 Non-Static Loops Without Internal Timing Controls., 476
10.10.4 Non-Static Loops With Internal Timing Controls, 478
10.10.5 State Machine Replacements for Loops, 482
### Chapter 11
**SWITCH-LEVEL MODELS IN VERILOG**, 495

11.1 MOS TRANSISTOR TECHNOLOGY, 495
11.2 SWITCH-LEVEL MODELS OF MOS TRANSISTORS, 498
11.3 SWITCH-LEVEL MODELS OF STATIC CMOS CIRCUITS, 498
11.4 ALTERNATIVE LOADS AND PULL GATES, 500
11.5 CMOS TRANSMISSION GATES, 501
11.6 BI-DIRECTIONAL GATES (SWITCHES), 508
11.7 SIGNAL STRENGTHS, 509
  11.7.1 Strength of a "Driven" Net, 511
  11.7.2 Supply Nets, 512
  11.7.3 Charge Storage Nets, 512
11.8 AMBIGUOUS SIGNALS, 512
11.9 STRENGTH REDUCTION BY PRIMITIVES, 513
  11.9.1 Transistor Switch and Bi-Directional Switches, 513
  11.9.2 Resistive MOS Devices, 514
11.10 COMBINATION AND RESOLUTION OF SIGNAL STRENGTHS, 516
  11.10.1 Signal Contention: Known Strength and Known Value, 516
  11.10.2 Combination of Ambiguous Strength and Known Value, 522
11.11 SIGNAL STRENGTHS AND WIRED LOGIC, 524
Chapter 12
DESIGN EXAMPLES IN VERILOG, 534

12.1 FIFO—BUFFERS FOR DATA ACQUISITION, 534
12.2 FIFO APPLICATION: TEMPERATURE MONITOR SYSTEM, 548
12.3 UART, 567
  12.3.1 UART—Transmitter, 570
  12.3.2 UART—Receiver, 580
12.4 BIT-SLICE MICROCONTROLLER, 594
12.5 SUMMARY, 608
REFERENCES, 608
PROBLEMS, 608

Chapter 13
RAPID PROTOTYPING WITH XILINX FPGAs, 610

13.1 INTRODUCTION TO FPGAs, 610
13.2 ROLE OF FPGAS IN THE ASIC MARKET, 612
13.3 FPGA TECHNOLOGIES, 614
13.4 THE XILINX XC3000 FPGA FAMILY, 620
  13.4.1 The XC3000 Configurable Logic Block (CLB), 620
  13.4.2 XC3000 Interconnect Resources, 622
  13.4.3 XC3000 Switch Box Functional Configuration and Connectivity, 623
  13.4.4 XC3000 I/O Block (IOB), 624
  13.4.5 XC3000 Performance, 626
13.5 THE XC4000 FPGA FAMILY, 627
  13.5.1 XC4000 Configurable Logic Block (CLB), 627
  13.5.2 XC4000 Interconnect Resources, 628
  13.5.3 XC4000 I/O Block (IOB), 631
  13.5.4 Enhancements in the XC4000E and XC4000X Series, 633
13.6 RAPID PROTOTYPING WITH VERILOG AND FPGAs, 633

13.7 DESIGN EXERCISES, 634
13.7.1 Microcontroller, 635
13.7.2 Electronic Roulette Wheel, 635
13.7.3 Electronic Dice Game, 641

13.8 SUMMARY, 641
REFERENCES, 641
PROBLEMS, 642

Appendix A
PREDEFINED PRIMITIVES, 647
A.1 MULTI-INPUT COMBINATIONAL LOGIC GATES, 648
A.2 MULTI-OUTPUT COMBINATIONAL GATES, 650
A.3 THREE-STATE GATES, 650
A.4 MOS TRANSISTOR SWITCHES, 652
A.5 MOS PULL-UP/PULL-DOWN GATES, 656
A.6 MOS BI-DIRECTIONAL SWITCHES, 657

Appendix B
VERILOG KEYWORDS, 659

Appendix C
VERILOG OPERATORS AND PRECEDENCE, 660

Appendix D
BACKUS–NAUR (BNF) FORMAL SYNTAX NOTATION, 662

Appendix E
SYSTEM TASKS AND FUNCTIONS, 663
E.1 DISPLAY TASKS, 665
E.2 FILE I/O TASKS, 670
E.3 TIME-SCALE TASKS, 673
Contents

E.4 SIMULATION CONTROL TASKS, 674
E.5 PLA MODELING TASKS, 675
E.6 STOCHASTIC ANALYSIS TASKS, 678
E.7 SIMULATION-TIME TASKS, 680
E.8 CONVERSION TASKS, 681
E.9 TIMING-CHECK TASKS, 681
E.10 PROBABILITY DISTRIBUTIONS, 682
E.11 VALUE-CHANGE DUMP (VCD) FILE TASKS, 684
E.12 ADDITIONAL (NONSTANDARD) TASKS, 684

Appendix F
VERILOG LANGUAGE FORMAL SYNTAX, 689
F.1 Source Text, 689
F.2 Declarations, 690
F.3 Primitive Instances, 692
F.4 Module Instantiation, 693
F.5 UDP Declaration and Instantiation, 693
F.6 Behavioral Statements, 694
F.7 Specify Section, 696
F.8 Expressions, 699
F.9 General, 701

Appendix G
PROGRAMMING LANGUAGE INTERFACE (PLI), 702

Appendix H
COMPILER DIRECTIVES, 704
H.1 `celldefine and `endcelldefine, 704
H.2 `defaultnettype, 704
H.3 `define and `undef, 705
H.4 `ifdef, `else, `endif, 705
H.5 `include, 706
H.6 `resetall, 706
H.7 `timescale, 706
H.8 `nounconnected_drive and `unconnected_drive, 707

Appendix I
FLIP-FLOP AND LATCH TYPES, 708
INDEX, 710
XILINIX STUDENT EDITION, 725