Advanced Digital Design with the Verilog HDL

Michael D. Ciletti

Department of Electrical and Computer Engineering
University of Colorado at Colorado Springs

Pearson Education, Inc.
Upper Saddle River, New Jersey 07458
Contents

Preface xvii
Simplify, Clarify, and Verify xviii

1 Introduction to Digital Design Methodology 1
1.1 Design Methodology—An Introduction 2
  1.1.1 Design Specification 4
  1.1.2 Design Partition 4
  1.1.3 Design Entry 4
  1.1.4 Simulation and Functional Verification 5
  1.1.5 Design Integration and Verification 6
  1.1.6 Presynthesis Sign-Off 6
  1.1.7 Gate-Level Synthesis and Technology Mapping 6
  1.1.8 Postsynthesis Design Validation 7
  1.1.9 Postsynthesis Timing Analysis 8
  1.1.10 Test Generation and Fault Simulation 8
  1.1.11 Placement and Routing 8
  1.1.12 Physical and Electrical Design Rules 9
  1.1.13 Parasitic Extraction 9
  1.1.14 Design Sign-Off 9
1.2 IC Technology Options 9
1.3 Overview 11
References 11
# Contents

## 2 Review of Combinational Logic Design  13

2.1 Combinational Logic and Boolean Algebra  13
  2.1.1 ASIC Library Cells  13
  2.1.2 Boolean Algebra  16
  2.1.3 DeMorgan's Laws  18

2.2 Theorems for Boolean Algebraic Minimization  18

2.3 Representation of Combinational Logic  21
  2.3.1 Sum of Products Representation  23
  2.3.2 Product-of-Sums Representation  26

2.4 Simplification of Boolean Expressions  27
  2.4.1 Simplification with Exclusive-Or  36
  2.4.2 Karnaugh Maps (SOP Form)  36
  2.4.3 Karnaugh Maps (POS Form)  39
  2.4.4 Karnaugh Maps and Don't-Cares  40
  2.4.5 Extended Karnaugh Maps  41

2.5 Glitches and Hazards  42
  2.5.1 Elimination of Static Hazards (SOP Form)  44
  2.5.2 Summary: Elimination of Static Hazards in Two-Level Circuits  48
  2.5.3 Static Hazards in Multilevel Circuits  49
  2.5.4 Summary: Elimination of Hazards in Multilevel Circuits  52
  2.5.5 Dynamic Hazards  52

2.6 Building Blocks for Logic Design  55
  2.6.1 NAND–NOR Structures  55
  2.6.2 Multiplexers  60
  2.6.3 Demultiplexers  61
  2.6.4 Encoders  62
  2.6.5 Priority Encoder  63
  2.6.6 Decoder  64
  2.6.7 Priority Decoder  66

References  67

Problems  67

## 3 Fundamentals of Sequential Logic Design  69

3.1 Storage Elements  69
  3.1.1 Latches  70
  3.1.2 Transparent latches  71

3.2 Flip-Flops  71
  3.2.1 D-Type Flip-Flop  71
  3.2.2 Master–Slave Flip-Flop  73
  3.2.3 J-K Flip-Flop  75
  3.2.4 T Flip-Flop  75
Introduction to Logic Design with Verilog  103

4.1  Structural Models of Combinational Logic  104
   4.1.1  Verilog Primitives and Design Encapsulation  104
   4.1.2  Verilog Structural Models  107
   4.1.3  Module Ports  107
   4.1.4  Some Language Rules  108
   4.1.5  Top-Down Design and Nested Modules  108
   4.1.6  Design Hierarchy and Source-Code Organization  111
   4.1.7  Vectors in Verilog  113
   4.1.8  Structural Connectivity  114

4.2  Logic Simulation, Design Verification, and Test Methodology  119
   4.2.1  Four-Valued Logic and Signal Resolution in Verilog  119
   4.2.2  Test Methodology  120
   4.2.3  Signal Generators for Testbenches  123
   4.2.4  Event-Driven Simulation  125
   4.2.5  Testbench Template  125
   4.2.6  Sized Numbers  126

4.3  Propagation Delay  126
   4.3.1  Inertial Delay  129
   4.3.2  Transport Delay  131

4.4  Truth Table Models of Combinational and Sequential Logic with
     Verilog  132

References  140
Problems  140

Logic Design with Behavioral Models of Combinational
and Sequential Logic  143

5.1  Behavioral Modeling  143
5.2  A Brief Look at Data Types for Behavioral Modeling  145
5.3  Boolean-Equation-Based Behavioral Models of
     Combinational Logic  145
5.4 Propagation Delay and Continuous Assignments 148
5.5 Latches and Level-Sensitive Circuits in Verilog 150
5.6 Cyclic Behavioral Models of Flip-Flops and Latches 153
5.7 Cyclic Behavior and Edge Detection 154
5.8 A Comparison of Styles for Behavioral Modeling 156
  5.8.1 Continuous-Assignment Models 156
  5.8.2 Dataflow/RTL Models 158
  5.8.3 Algorithm-Based Models 162
  5.8.4 Port Names: A Matter of Style 164
  5.8.5 Simulation with Behavioral Models 164
5.9 Behavioral Models of Multiplexers, Encoders, and Decoders 165
5.10 Dataflow Models of a Linear-Feedback Shift Register 174
5.11 Modeling Digital Machines with Repetitive Algorithms 176
  5.11.1 Intellectual Property Reuse and Parameterized Models 181
  5.11.2 Clock Generators 183
5.12 Machines with Multicycle Operations 185
5.13 Design Documentation with Functions and Tasks: Legacy or Lunacy? 186
  5.13.1 Tasks 187
  5.13.2 Functions 189
5.14 Algorithmic State Machine Charts for Behavioral Modeling 190
5.15 ASMD Charts 194
5.16 Behavioral Models of Counters, Shift Registers, and Register Files 196
  5.16.1 Counters 197
  5.16.2 Shift Registers 203
  5.16.3 Register Files and Arrays of Registers (Memories) 207
5.17 Switch Debounce, Metastability, and Synchronizers for Asynchronous Signals 210
5.18 Design Example: Keypad Scanner and Encoder 216
References 224
Problems 225

6 Synthesis of Combinational and Sequential Logic 233
6.1 Introduction to Synthesis 234
  6.1.1 Logic Synthesis 235
  6.1.2 RTL Synthesis 243
  6.1.3 High-Level Synthesis 244
6.2 Synthesis of Combinational Logic 245
  6.2.1 Synthesis of Priority Structures 250
  6.2.2 Exploiting Logical Don’t-Care Conditions 251
  6.2.3 ASIC Cells and Resource Sharing 256
6.3 Synthesis of Sequential Logic with Latches 258
  6.3.1 Accidental Synthesis of Latches 260
  6.3.2 Intentional Synthesis of Latches 264
6.4 Synthesis of Three-State Devices and Bus Interfaces 268
6.5 Synthesis of Sequential Logic with Flip-Flops 271
6.6 Synthesis of Sequential Logic with Flip-Flops 275
   6.6.1 Synthesis of a BCD-to-Excess-3 Code Converter 275
   6.6.2 Synthesis of a Mealy-Type NRZ-to-Manchester Line Code
       Converter 280
   6.6.3 Synthesis of a Moore-Type NRZ-to-Manchester Line Code
       Converter 282
   6.6.4 Synthesis of a Sequence Recognizer 283
6.7 Registered Logic 292
6.8 State Encoding 299
6.9 Synthesis of Implicit State Machines, Registers, and Counters 301
   6.9.1 Implicit State Machines 301
   6.9.2 Synthesis of Counters 302
   6.9.3 Synthesis of Registers 304
6.10 Resets 309
6.11 Synthesis of Gated Clocks and Clock Enables 313
6.12 Anticipating the Results of Synthesis 314
   6.12.1 Synthesis of Data Types 314
   6.12.2 Operator Grouping 314
   6.12.3 Expression Substitution 316
6.13 Synthesis of Loops 319
   6.13.1 Static Loops without Embedded Timing Controls 319
   6.13.2 Static Loops with Embedded Timing Controls 322
   6.13.3 Nonstatic Loops without Embedded Timing Controls 326
   6.13.4 Nonstatic Loops with Embedded Timing Controls 328
   6.13.5 State-Machine Replacements for Unsynchronizable Loops 331
6.14 Design Traps to Avoid 338
6.15 Divide and Conquer: Partitioning a Design 338
References 339
Problems 339

7 Design and Synthesis of Datapath Controllers 347
7.1 Partitioned Sequential Machines 347
7.2 Design Example: Binary Counter 349
7.3 Design and Synthesis of a RISC Stored-Program Machine 355
   7.3.1 RISC SPM: Processor 357
   7.3.2 RISC SPM: ALU 357
   7.3.3 RISC SPM: Controller 357
   7.3.4 RISC SPM: Instruction Set 358
   7.3.5 RISC SPM: Controller Design 360
   7.3.6 RISC SPM: Program Execution 375
7.4 Design Example: UART 378
   7.4.1 UART Operation 379
8 Programmable Logic and Storage Devices 415

8.1 Programmable Logic Devices 417
8.2 Storage Devices 417
  8.2.1 Read-Only Memory (ROM) 418
  8.2.2 Programmable ROM (PROM) 421
  8.2.3 Erasable ROMs 422
  8.2.4 ROM-Based Implementation of Combinational Logic 423
  8.2.5 Verilog System Tasks for ROMs 424
  8.2.6 Comparison of ROMs 426
  8.2.7 ROM-Based State Machines 426
  8.2.8 Flash Memory 430
  8.2.9 Static Random Access Memory (SRAM) 432
  8.2.10 Ferroelectric Nonvolatile Memory 454
8.3 Programmable Logic Array (PLA) 456
  8.3.1 PLA Minimization 459
  8.3.2 PLA Modeling 461
8.4 Programmable Array Logic (PAL) 465
8.5 Programmability of PLDs 467
8.6 Complex PLDs (CPLDs) 467
8.7 Altera MAX 7000 CPLD 468
  8.7.1 Shareable Expander 471
  8.7.2 Parallel Expander 472
  8.7.3 I/O Control Block 473
  8.7.4 Timing Considerations 473
  8.7.5 Device Resources 473
  8.7.6 Other Altera Device Families 474
8.8 XILINX XC9500 CPLDs 474
8.9 Field-Programmable Gate Arrays 476
  8.9.1 The Role of FPGAs in the ASIC Market 478
  8.9.2 FPGA Technologies 479
8.10 Altera Flex 8000 FPGAs 480
8.11 Altera Flex 10 FPGAs 481
8.12 Altera Apex FPGAs 486
8.13 Altera Chip Programmability 488
8.14 XILINX XC4000 Series FPGA 488
  8.14.1 Basic Architecture 488
  8.14.2 XC4000 Configurable Logic Block 488
9 Algorithms and Architectures for Digital Processors 547

9.1 Algorithms, Nested-Loop Programs, and Data Flow Graphs 548

9.2 Design Example: Halftone Pixel Image Converter 551
  9.2.1 Baseline Design for a Halftone Pixel Image Converter 554
  9.2.2 NLP-Based Architectures for the Halftone Pixel Image Converter 558
  9.2.3 Concurrent ASMD-Based Architecture for a Halftone Pixel Image Converter 570
  9.2.4 Halftone Pixel Image Converter: Design Tradeoffs 583
  9.2.5 Architectures for Dataflow Graphs with Feedback 584

9.3 Digital Filters and Signal Processors 591
  9.3.1 Finite-Duration Impulse Response (FIR) Filter 594
  9.3.2 Digital Filter Design Process 595
  9.3.3 Infinite-Duration Impulse Response (IIR) Filter 600

9.4 Building Blocks for Signal Processors 603
  9.4.1 Integrators (Accumulators) 604
  9.4.2 Differentiators 608
  9.4.3 Decimation and Interpolation Filters 608

9.5 Pipelined Architectures 614
  9.5.1 Design Example: Pipelined Adder 617
  9.5.2 Design Example: Pipelined FIR Filter 622

9.6 Circular Buffers 622

9.7 FIFOs and Synchronization across Clock Domains 628

References 642

Problems 642
## 10 Architectures for Arithmetic Processors 651

10.1 Number Representation 651
   10.1.1 Signed Magnitude Representation of Negative Integers 652
   10.1.2 Ones Complement Representation of Negative Integers 653
   10.1.3 Twos Complement Representation of Positive and Negative Integers 654
   10.1.4 Representation of Fractions 656

10.2 Functional Units for Addition and Subtraction 656
   10.2.1 Ripple-Carry Adder 656
   10.2.2 Carry Look-Ahead Adder 656
   10.2.3 Overflow and Underflow 662

10.3 Functional Units for Multiplication 663
   10.3.1 Combinational (Parallel) Binary Multiplier 663
   10.3.2 Sequential Binary Multiplier 667
   10.3.3 Sequential Multiplier Design: Hierarchical Decomposition 668
   10.3.4 STG-Based Controller Design 669
   10.3.5 Efficient STG-Based Sequential Binary Multiplier 676
   10.3.6 ASMD-Based Sequential Binary Multiplier 682
   10.3.7 Efficient ASM-Based Sequential Multiplier 686
   10.3.8 Summary of ASMD-Based Datapath Controller Design 691
   10.3.9 Reduced-Register Sequential Multiplier 693
   10.3.10 Implicit-State-Machine Binary Multiplier 698
   10.3.11 Booth's-Algorithm Sequential Multiplier 711
   10.3.12 Bit-Pair Encoding 721

10.4 Multiplication of Signed Binary Numbers 728
   10.4.1 Product of Signed Numbers: Negative Multiplicand, Positive Multiplier 729
   10.4.2 Product of Signed Numbers: Positive Multiplicand, Negative Multiplier 729
   10.4.3 Product of Signed Numbers: Negative Multiplicand, Negative Multiplier 730

10.5 Multiplication of Fractions 731
   10.5.1 Signed Fractions: Positive Multiplicand, Positive Multiplier 732
   10.5.2 Signed Fractions: Negative Multiplicand, Positive Multiplier 733
   10.5.3 Signed Fractions: Positive Multiplicand, Negative Multiplier 733
   10.5.4 Signed Fractions: Negative Multiplicand, Negative Multiplier 734

10.6 Functional Units for Division 735
   10.6.1 Division of Unsigned Binary Numbers 735
   10.6.2 Efficient Division of Unsigned Binary Numbers 742
   10.6.3 Reduced-Register Sequential Divider 750
   10.6.4 Division of Signed (2s Complement) Binary Numbers 757

References 757
Problems 757
11 Postsynthesis Design Tasks 765

11.1 Postsynthesis Design Validation 765

11.2 Postsynthesis Timing Verification 768

11.2.1 Static Timing Analysis 770

11.2.2 Timing Specifications 773

11.2.3 Factors That Affect Timing 775

11.3 Elimination of ASIC Timing Violations 779

11.4 False Paths 783

11.5 Dynamically Sensitized Paths 785

11.6 System Tasks for Timing Verification 787

11.6.1 Timing Check: Setup Condition 787

11.6.2 Timing Check: Hold Condition 788

11.6.3 Timing Check: Setup and Hold Conditions 789

11.6.4 Timing Check: Pulselength Constraint 790

11.6.5 Timing Constraint: Signal Skew Constraint 791

11.6.6 Timing Check: Clock Period 791

11.6.7 Timing Check: Recovery Time 792

11.7 Fault Simulation and Testing 794

11.7.1 Circuit Defects and Faults 795

11.7.2 Fault Detection and Testing 798

11.7.3 D-Notation 800

11.7.4 Automatic Test-Pattern Generation for Combinational Circuits 804

11.7.5 Fault Coverage and Defect Levels 805

11.7.6 Test Generation for Sequential Circuits 806

11.8 Fault Simulation 811

11.8.1 Fault Collapsing 811

11.8.2 Serial Fault Simulation 812

11.8.3 Parallel Fault Simulation 812

11.8.4 Concurrent Fault Simulation 812

11.8.5 Probabilistic Fault Simulation 813

11.9 Fault Simulation with Verifault-XL 813

11.9.1 Tasks for Fault Simulation 813

11.9.2 Fault Collapsing and Classification with Verifault-XL 814

11.9.3 Structural and Behavioral Fault Propagation 816

11.9.4 Testbench for Fault Simulation with Verifault-XL 817

11.9.5 Fault Descriptors 819

11.10 JTAG Ports and Design for Testability 821

11.10.1 Boundary Scan and JTAG Ports 823

11.10.2 JTAG Modes of Operation 825

11.10.3 JTAG Registers 826

11.10.4 JTAG Instructions 828
A Verilog Primitives  883
A.1 Multiinput Combinational Logic Gates  883
A.2 Multioutput Combinational Gates  885
A.3 Three-State Gates  886
A.4 MOS Transistor Switches  889
A.5 MOS Pull-Up/Pull-Down Gates  892
A.6 MOS Bidirectional Switches  892

B Verilog Keywords  895

C Verilog Data Types  897
C.1 Nets  897
C.2 Register Variables  898
C.3 Constants  902
C.4 Referencing Arrays of Nets or Regs  903

D Verilog Operators  905
D.1 Arithmetic Operators  905
D.2 Bitwise Operators  907
D.3 Reduction Operators  908
D.4 Logical Operators  909
D.5 Relational Operators  910
D.6 Shift Operators  910
D.7 Conditional Operator  910
D.8 Concatenation Operator  911
D.9 Expressions and Operands  912
D.10 Operator Precedence  912

E Backus–Naur Formal Syntax Notation  915

F Verilog Language Formal Syntax  917
F.1 Source Text  917
F.2 Declarations  918
F.3 Primitive Instances  920