

CONTENTS

Chapter 1

DESIGN CONCEPTS 1

- 1.1 Digital Hardware 2
 - 1.1.1 Standard Chips 4
 - 1.1.2 Programmable Logic Devices 4
 - 1.1.3 Custom-Designed Chips 5
- 1.2 The Design Process 6
- 1.3 Design of Digital Hardware 8
 - 1.3.1 Basic Design Loop 8
 - 1.3.2 Design of a Digital Hardware Unit 9
- 1.4 Logic Circuit Design in This Book 11
- 1.5 Theory and Practice 14
 - References 15

Chapter 2

INTRODUCTION TO LOGIC CIRCUITS 17

- 2.1 Variables and Functions 18
- 2.2 Inversion 21
- 2.3 Truth Tables 22
- 2.4 Logic Gates and Networks 23
 - 2.4.1 Analysis of a Logic Network 24
- 2.5 Boolean Algebra 27
 - 2.5.1 The Venn Diagram 30
 - 2.5.2 Notation and Terminology 34
 - 2.5.3 Precedence of Operations 34
- 2.6 Synthesis Using AND, OR, and NOT Gates 35
 - 2.6.1 Sum-of-Products and Product-of-Sums Forms 37
- 2.7 NAND and NOR Logic Networks 41
- 2.8 Design Examples 44
 - 2.8.1 Three-Way Light Control 44
 - 2.8.2 Multiplexer Circuit 45
- 2.9 Introduction to CAD Tools 48
 - 2.9.1 Design Entry 48
 - 2.9.2 Synthesis 51
 - 2.9.3 Functional Simulation 52
 - 2.9.4 Summary 52

- 2.10 Introduction to Verilog 54
 - 2.10.1 Structural Specification of Logic Circuits 55
 - 2.10.2 Behavioral Specification of Logic Circuits 58
 - 2.10.3 How *Not* to Write Verilog Code 60
- 2.11 Concluding Remarks 60
 - Problems 61
 - References 65

Chapter 3

IMPLEMENTATION TECHNOLOGY 67

- 3.1 Transistor Switches 69
- 3.2 NMOS Logic Gates 71
- 3.3 CMOS Logic Gates 74
 - 3.3.1 Speed of Logic Gate Circuits 81
- 3.4 Negative Logic System 82
- 3.5 Standard Chips 83
 - 3.5.1 7400-Series Standard Chips 83
- 3.6 Programmable Logic Devices 87
 - 3.6.1 Programmable Logic Array (PLA) 87
 - 3.6.2 Programmable Array Logic (PAL) 90
 - 3.6.3 Programming of PLAs and PALs 92
 - 3.6.4 Complex Programmable Logic Devices (CPLDs) 94
 - 3.6.5 Field-Programmable Gate Arrays 98
 - 3.6.6 Using CAD Tools to Implement Circuits in CPLDs and FPGAs 102
- 3.7 Custom Chips, Standard Cells, and Gate Arrays 103
- 3.8 Practical Aspects 106
 - 3.8.1 MOSFET Fabrication and Behavior 106
 - 3.8.2 MOSFET On-Resistance 110
 - 3.8.3 Voltage Levels in Logic Gates 111
 - 3.8.4 Noise Margin 113
 - 3.8.5 Dynamic Operation of Logic Gates 114
 - 3.8.6 Power Dissipation in Logic Gates 117

- 3.8.7 Passing 1s and 0s Through Transistor Switches 118
- 3.8.8 Fan-in and Fan-out in Logic Gates 120
- 3.9 Transmission Gates 126
 - 3.9.1 Exclusive-OR Gates 127
 - 3.9.2 Multiplexer Circuit 128
- 3.10 Implementation Details for SPLDs, CPLDs, and FPGAs 129
 - 3.10.1 Implementation in FPGAs 135
- 3.11 Concluding Remarks 137
 - Problems 138
 - References 147

Chapter 4
OPTIMIZED IMPLEMENTATION OF LOGIC FUNCTIONS 149

- 4.1 Karnaugh Map 150
- 4.2 Strategy for Minimization 158
 - 4.2.1 Terminology 159
 - 4.2.2 Minimization Procedure 160
- 4.3 Minimization of Product-of-Sums Forms 164
- 4.4 Incompletely Specified Functions 166
- 4.5 Multiple-Output Circuits 167
- 4.6 Multilevel Synthesis 171
 - 4.6.1 Factoring 172
 - 4.6.2 Functional Decomposition 175
 - 4.6.3 Multilevel NAND and NOR Circuits 181
- 4.7 Analysis of Multilevel Circuits 184
- 4.8 Cubical Representation 189
 - 4.8.1 Cubes and Hypercubes 189
- 4.9 A Tabular Method for Minimization 193
 - 4.9.1 Generation of Prime Implicants 193
 - 4.9.2 Determination of a Minimum Cover 195
 - 4.9.3 Summary of the Tabular Method 200
- 4.10 A Cubical Technique for Minimization 201
 - 4.10.1 Determination of Essential Prime Implicants 204
 - 4.10.2 Complete Procedure for Finding a Minimal Cover 206
- 4.11 Practical Considerations 208
- 4.12 CAD Tools 209
 - 4.12.1 Logic Synthesis and Optimization 210
 - 4.12.2 Physical Design 211
 - 4.12.3 Timing Simulation 213

- 4.12.4 Summary of Design Flow 213
- 4.12.5 Examples of Circuits Synthesized from Verilog Code 216
- 4.13 Concluding Remarks 220
 - Problems 221
 - References 226

Chapter 5
NUMBER REPRESENTATION AND ARITHMETIC CIRCUITS 229

- 5.1 Positional Number Representation 230
 - 5.1.1 Unsigned Integers 230
 - 5.1.2 Conversion Between Decimal and Binary Systems 231
 - 5.1.3 Octal and Hexadecimal Representations 232
- 5.2 Addition of Unsigned Numbers 234
 - 5.2.1 Decomposed Full-Adder 238
 - 5.2.2 Ripple-Carry Adder 239
 - 5.2.3 Design Example 240
- 5.3 Signed Numbers 240
 - 5.3.1 Negative Numbers 240
 - 5.3.2 Addition and Subtraction 244
 - 5.3.3 Adder and Subtractor Unit 248
 - 5.3.4 Radix-Complement Schemes 249
 - 5.3.5 Arithmetic Overflow 253
 - 5.3.6 Performance Issues 254
- 5.4 Fast Adders 255
 - 5.4.1 Carry-Lookahead Adder 255
- 5.5 Design of Arithmetic Circuits Using CAD Tools 262
 - 5.5.1 Design of Arithmetic Circuits Using Schematic Capture 262
 - 5.5.2 Design of Arithmetic Circuits Using Verilog 265
 - 5.5.3 Using Vecteded Signals 268
 - 5.5.4 Using a Generic Specification 269
 - 5.5.5 Nets and Variables in Verilog 270
 - 5.5.6 Arithmetic Assignment Statements 271
 - 5.5.7 Representation of Numbers in Verilog Code 275
- 5.6 Multiplication 277
 - 5.6.1 Array Multiplier for Unsigned Numbers 279
 - 5.6.2 Multiplication of Signed Numbers 279

5.7 Other Number Representations 282
 5.7.1 Fixed-Point Numbers 282
 5.7.2 Floating-Point Numbers 282
 5.7.3 Binary-Coded-Decimal Representation 284
 5.8 ASCII Character Code 289
 Problems 291
 References 295

Chapter 6
COMBINATIONAL-CIRCUIT BUILDING BLOCKS 297

6.1 Multiplexers 298
 6.1.1 Synthesis of Logic Functions Using Multiplexers 303
 6.1.2 Multiplexer Synthesis Using Shannon's Expansion 304
 6.2 Decoders 311
 6.2.1 Demultiplexers 314
 6.3 Encoders 316
 6.3.1 Binary Encoders 316
 6.3.2 Priority Encoders 317
 6.4 Code Converters 318
 6.5 Arithmetic Comparison Circuits 320
 6.6 Verilog for Combinational Circuits 320
 6.6.1 The Conditional Operator 321
 6.6.2 The If-Else Statement 323
 6.6.3 The Case Statement 326
 6.6.4 The For Loop 331
 6.6.5 Verilog Operators 333
 6.6.6 The Generate Construct 338
 6.6.7 Tasks and Functions 339
 6.7 Concluding Remarks 343
 Problems 343
 References 347

Chapter 7
FLIP-FLOPS, REGISTERS, COUNTERS, AND A SIMPLE PROCESSOR 349

7.1 Basic Latch 351
 7.2 Gated SR Latch 353
 7.2.1 Gated SR Latch with NAND Gates 355
 7.3 Gated D Latch 356
 7.3.1 Effects of Propagation Delays 358

7.4 Master-Slave and Edge-Triggered D Flip-Flops 359
 7.4.1 Master-Slave D Flip-Flop 359
 7.4.2 Edge-Triggered D Flip-Flop 360
 7.4.3 D Flip-Flops with Clear and Preset 362
 7.5 T Flip-Flop 364
 7.5.1 Configurable Flip-Flops 367
 7.6 JK Flip-Flop 367
 7.7 Summary of Terminology 368
 7.8 Registers 368
 7.8.1 Shift Register 369
 7.8.2 Parallel-Access Shift Register 370
 7.9 Counters 371
 7.9.1 Asynchronous Counters 371
 7.9.2 Synchronous Counters 374
 7.9.3 Counters with Parallel Load 378
 7.10 Reset Synchronization 378
 7.11 Other Types of Counters 382
 7.11.1 BCD Counter 382
 7.11.2 Ring Counter 383
 7.11.3 Johnson Counter 384
 7.11.4 Remarks on Counter Design 385
 7.12 Using Storage Elements with CAD Tools 385
 7.12.1 Including Storage Elements in Schematics 385
 7.12.2 Using Verilog Constructs for Storage Elements 388
 7.12.3 Blocking and Non-blocking Assignments 390
 7.12.4 Non-blocking Assignments for Combinational Circuits 394
 7.12.5 Flip-Flops with Clear Capability 395
 7.13 Using Registers and Counters with CAD Tools 396
 7.13.1 Including Registers and Counters in Schematics 396
 7.13.2 Using Library Modules in Verilog Code 399
 7.13.3 Using Verilog Constructs for Registers and Counters 400
 7.14 Design Examples 405
 7.14.1 Bus Structure 405
 7.14.2 Simple Processor 417
 7.14.3 Reaction Timer 429
 7.14.4 Register Transfer Level (RTL) Code 433
 7.15 Concluding Remarks 434

Problems 434
References 442

Chapter 8
SYNCHRONOUS SEQUENTIAL
CIRCUITS 445

8.1 Basic Design Steps 447
 8.1.1 State Diagram 447
 8.1.2 State Table 449
 8.1.3 State Assignment 449
 8.1.4 Choice of Flip-Flops and Derivation of Next-State and Output Expressions 451
 8.1.5 Timing Diagram 453
 8.1.6 Summary of Design Steps 454
 8.2 State-Assignment Problem 458
 8.2.1 One-Hot Encoding 460
 8.3 Mealy State Model 462
 8.4 Design of Finite State Machines Using CAD Tools 467
 8.4.1 Verilog Code for Moore-Type FSMs 467
 8.4.2 Synthesis of Verilog Code 468
 8.4.3 Simulating and Testing the Circuit 470
 8.4.4 Alternative Styles of Verilog Code 471
 8.4.5 Summary of Design Steps When Using CAD Tools 473
 8.4.6 Specifying the State Assignment in Verilog Code 474
 8.4.7 Specification of Mealy FSMs Using Verilog 475
 8.5 Serial Adder Example 477
 8.5.1 Mealy-Type FSM for Serial Adder 477
 8.5.2 Moore-Type FSM for Serial Adder 479
 8.5.3 Verilog Code for the Serial Adder 480
 8.6 State Minimization 486
 8.6.1 Partitioning Minimization Procedure 486
 8.6.2 Incompletely Specified FSMs 493
 8.7 Design of a Counter Using the Sequential Circuit Approach 495
 8.7.1 State Diagram and State Table for a Modulo-8 Counter 495
 8.7.2 State Assignment 496
 8.7.3 Implementation Using D-Type Flip-Flops 497
 8.7.4 Implementation Using JK-Type Flip-Flops 498
 8.7.5 Example—A Different Counter 502

8.8 FSM as an Arbiter Circuit 505
 8.8.1 Implementation of the Arbiter Circuit 508
 8.8.2 Minimizing the Output Delays for an FSM 511
 8.8.3 Summary 511
 8.9 Analysis of Synchronous Sequential Circuits 512
 8.10 Algorithmic State Machine (ASM) Charts 516
 8.11 Formal Model for Sequential Circuits 519
 8.12 Concluding Remarks 521
 Problems 521
 References 525

Chapter 9
ASYNCHRONOUS SEQUENTIAL
CIRCUITS 527

9.1 Asynchronous Behavior 528
 9.2 Analysis of Asynchronous Circuits 531
 9.3 Synthesis of Asynchronous Circuits 540
 9.4 State Reduction 553
 9.5 State Assignment 568
 9.5.1 Transition Diagram 571
 9.5.2 Exploiting Unspecified Next-State Entries 574
 9.5.3 State Assignment Using Additional State Variables 578
 9.5.4 One-Hot State Assignment 582
 9.6 Hazards 584
 9.6.1 Static Hazards 585
 9.6.2 Dynamic Hazards 590
 9.6.3 Significance of Hazards 592
 9.7 A Complete Design Example 592
 9.7.1 The Vending-Machine Controller 592
 9.8 Concluding Remarks 599
 Problems 599
 References 604

Chapter 10
DIGITAL SYSTEM DESIGN 605

10.1 Building Block Circuits 606
 10.1.1 Flip-Flops and Registers with Enable Inputs 606
 10.1.2 Shift Registers with Enable Inputs 607

10.1.3	Static Random Access Memory (SRAM)	609	A.3	Signals in Verilog Code	699
10.1.4	SRAM Blocks in PLDs	611	A.4	Identifier Names	699
10.2	Design Examples	612	A.5	Signal Values, Numbers, and Parameters	700
10.2.1	A Bit-Counting Circuit	612	A.6	Net and Variable Types	701
10.2.2	ASM-Chart-Implied Timing Information	613	A.6.1	Nets	701
10.2.3	Shift-and-Add Multiplier	618	A.6.2	Variables	702
10.2.4	Divider	623	A.6.3	Memories	703
10.2.5	Arithmetic Mean	631	A.7	Operators	703
10.2.6	Sort Operation	641	A.8	Verilog Module	705
10.3	Clock Synchronization	653	A.9	Gate Instantiations	706
10.3.1	Clock Skew	653	A.10	Concurrent Statements	708
10.3.2	Flip-Flop Timing Parameters	655	A.10.1	Continuous Assignments	709
10.3.3	Asynchronous Inputs to Flip-Flops	656	A.10.2	Using Parameters	710
10.3.4	Switch Debouncing	657	A.11	Procedural Statements	711
10.4	Concluding Remarks	659	A.11.1	Always and Initial Blocks	711
	Problems	659	A.11.2	The If-Else Statement	713
	References	663	A.11.3	Statement Ordering	714
			A.11.4	The Case Statement	715
			A.11.5	Casex and Casez Statements	717
			A.11.6	Loop Statements	717
			A.11.7	Blocking versus Non-blocking Assignments for Combinational Circuits	721
			A.12	Using Subcircuits	721
			A.12.1	Subcircuit Parameters	723
			A.12.2	Verilog 2001 Generate Capability	725
			A.13	Functions and Tasks	726
			A.14	Sequential Circuits	730
			A.14.1	A Gated D Latch	730
			A.14.2	D Flip-Flop	730
			A.14.3	Flip-Flops with Reset	731
			A.14.4	Instantiating a Flip-Flop from a Library	732
			A.14.5	Registers	733
			A.14.6	Shift Registers	734
			A.14.7	Counters	735
			A.14.8	An Example of a Sequential Circuit	736
			A.14.9	Moore-Type Finite State Machines	737
			A.14.10	Mealy-Type Finite State Machines	739
			A.15	Guidelines for Writing Verilog Code	742
			A.16	MAX+PlusII Verilog Support	745
			A.16.1	Limitations in MAX+PlusII	745
			A.17	Concluding Remarks	746
				References	746
Chapter 11					
TESTING OF LOGIC CIRCUITS 665					
11.1	Fault Model	666			
11.1.1	Stuck-at Model	666			
11.1.2	Single and Multiple Faults	667			
11.1.3	CMOS Circuits	667			
11.2	Complexity of a Test Set	667			
11.3	Path Sensitizing	669			
11.3.1	Detection of a Specific Fault	671			
11.4	Circuits with Tree Structure	673			
11.5	Random Tests	674			
11.6	Testing of Sequential Circuits	677			
11.6.1	Design for Testability	677			
11.7	Built-in Self-Test	681			
11.7.1	Built-in Logic Block Observer	685			
11.7.2	Signature Analysis	687			
11.7.3	Boundary Scan	688			
11.8	Printed Circuit Boards	688			
11.8.1	Testing of PCBs	690			
11.8.2	Instrumentation	691			
11.9	Concluding Remarks	692			
	Problems	692			
	References	695			
Appendix A					
VERILOG REFERENCE 697					
A.1	Documentation in Verilog Code	699			
A.2	White Space	699			

xx CONTENTS

Appendix B

TUTORIAL 1 747

- B.1 Introduction 748
 - B.1.1 Getting Started 748
- B.2 Design Entry Using Schematic Capture 751
 - B.2.1 Specifying the Project Name 752
 - B.2.2 Using the Graphic Editor 752
 - B.2.3 Synthesizing a Circuit from the Schematic 758
 - B.2.4 Performing Functional Simulation 759
 - B.2.5 Using the Message Processor to Locate and Fix Errors 763
- B.3 Design Entry Using Verilog 765
 - B.3.1 Specifying the Project Name 765
 - B.3.2 Using the Text Editor 765
 - B.3.3 Synthesizing a Circuit from the Verilog Code 767
 - B.3.4 Performing Functional Simulation 767
 - B.3.5 Using the Message Processor to Debug Verilog Code 768
- B.4 Design Entry Using Truth Tables 768
 - B.4.1 Using the Waveform Editor 769
 - B.4.2 Create the Timing Diagram 769
 - B.4.3 Synthesizing a Circuit from the Waveforms 770
- B.5 Mixing Design-Entry Methods 772
 - B.5.1 Creating a Schematic that Includes a Truth Table 772
 - B.5.2 Synthesizing and Simulating a Circuit from the Schematic 774
 - B.5.3 Using the Hierarchy Display 775
 - B.5.4 Concluding Remarks 775

Appendix C

TUTORIAL 2 777

- C.1 Implementing a Circuit in a MAX 7000 CPLD 778
 - C.1.1 Using the Compiler 779
 - C.1.2 Selecting a Chip 780
 - C.1.3 Viewing the Logic Synthesis Options 781
 - C.1.4 Examining the Implemented Circuit 782
 - C.1.5 Running the Timing Simulator 783
 - C.1.6 Using the Floorplan Editor 784
- C.2 Implementing a Circuit in a FLEX 10K FPGA 787

- C.3 Downloading a Circuit into a Device 789
- C.4 Making Pin Assignments 790
 - C.4.1 Assigning Signals to Pins in the Floorplan Editor 792
 - C.4.2 Making Pin Assignments Permanent 794
- C.5 Concluding Remarks 795

Appendix D

TUTORIAL 3 797

- D.1 Design Using Verilog Code 798
 - D.1.1 The Ripple-Carry Adder Code 798
 - D.1.2 Using the Timing Analyzer Module 801
- D.2 Using an LPM Module 802
- D.3 Design of a Sequential Circuit 806
 - D.3.1 Using the Graphic Editor 806
 - D.3.2 Synthesizing a Circuit and Using the Timing Simulator 812
 - D.3.3 Using the Timing Analyzer 813
 - D.3.4 Using Verilog Code 814
- D.4 Design of a Finite State Machine 815
 - D.4.1 Implementation in a CPLD 815
 - D.4.2 Implementation in an FPGA 816
- D.5 Concluding Remarks 819

Appendix E

COMMERCIAL DEVICES 821

- E.1 Simple PLDs 822
 - E.1.1 The 22V10 PAL Device 822
- E.2 Complex PLDs 824
 - E.2.1 Altera MAX 7000 825
- E.3 Field-Programmable Gate Arrays 826
 - E.3.1 Altera FLEX 10K 827
 - E.3.2 Xilinx XC4000 830
 - E.3.3 Altera APEX 20K 831
 - E.3.4 Altera Stratix 832
 - E.3.5 Xilinx Virtex 834
- E.4 Transistor-Transistor Logic 835
 - E.4.1 TTL Circuit Families 836
- References 837

INDEX 838