

# 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 Structure of a Computer 9
  - 1.3.3 Design of a Digital Hardware Unit 12
- 1.4 Logic Circuit Design in This Book 16
- 1.5 Theory and Practice 16
  - References 17

## Chapter 2

### INTRODUCTION TO LOGIC CIRCUITS 19

- 2.1 Variables and Functions 20
- 2.2 Inversion 23
- 2.3 Truth Tables 24
- 2.4 Logic Gates and Networks 25
  - 2.4.1 Analysis of a Logic Network 27
- 2.5 Boolean Algebra 29
  - 2.5.1 The Venn Diagram 33
  - 2.5.2 Notation and Terminology 35
  - 2.5.3 Precedence of Operations 37
- 2.6 Synthesis Using AND, OR, and NOT Gates 37
  - 2.6.1 Sum-of-Products and Product-of-Sums Forms 39
- 2.7 NAND and NOR Logic Networks 45
- 2.8 Design Examples 50
  - 2.8.1 Three-Way Light Control 50
  - 2.8.2 Multiplexer Circuit 51
- 2.9 Introduction to CAD Tools 54
  - 2.9.1 Design Entry 54
  - 2.9.2 Synthesis 56
  - 2.9.3 Functional Simulation 56

- 2.9.4 Physical Design 57
- 2.9.5 Timing Simulation 57
- 2.9.6 Chip Configuration 57

- 2.10 Introduction to Verilog 59
  - 2.10.1 Structural Specification of Logic Circuits 60
  - 2.10.2 Behavioral Specification of Logic Circuits 62
  - 2.10.3 How *Not* to Write Verilog Code 65
- 2.11 Concluding Remarks 66
- 2.12 Examples of Solved Problems 66
  - Problems 70
  - References 74

## Chapter 3

### IMPLEMENTATION TECHNOLOGY 77

- 3.1 Transistor Switches 79
- 3.2 NMOS Logic Gates 82
- 3.3 CMOS Logic Gates 85
  - 3.3.1 Speed of Logic Gate Circuits 91
- 3.4 Negative Logic System 91
- 3.5 Standard Chips 95
  - 3.5.1 7400-Series Standard Chips 95
- 3.6 Programmable Logic Devices 98
  - 3.6.1 Programmable Logic Array (PLA) 98
  - 3.6.2 Programmable Array Logic (PAL) 101
  - 3.6.3 Programming of PLAs and PALs 103
  - 3.6.4 Complex Programmable Logic Devices (CPLDs) 105
  - 3.6.5 Field-Programmable Gate Arrays 109
  - 3.6.6 Using CAD Tools to Implement Circuits in CPLDs and FPGAs 114
  - 3.6.7 Applications of CPLDs and FPGAs 114
- 3.7 Custom Chips, Standard Cells, and Gate Arrays 114
- 3.8 Practical Aspects 118
  - 3.8.1 MOSFET Fabrication and Behavior 118
  - 3.8.2 MOSFET On-Resistance 121
  - 3.8.3 Voltage Levels in Logic Gates 122
  - 3.8.4 Noise Margin 123

3.8.5	Dynamic Operation of Logic Gates	125
3.8.6	Power Dissipation in Logic Gates	128
3.8.7	Passing 1s and 0s Through Transistor Switches	130
3.8.8	Fan-in and Fan-out in Logic Gates	132
3.9	Transmission Gates	138
3.9.1	Exclusive-OR Gates	139
3.9.2	Multiplexer Circuit	140
3.10	Implementation Details for SPLDs, CPLDs, and FPGAs	140
3.10.1	Implementation in FPGAs	146
3.11	Concluding Remarks	149
3.12	Examples of Solved Problems	149
	Problems	157
	References	166

**Chapter 4**  
**OPTIMIZED IMPLEMENTATION OF LOGIC FUNCTIONS 167**

4.1	Karnaugh Map	168
4.2	Strategy for Minimization	176
4.2.1	Terminology	177
4.2.2	Minimization Procedure	178
4.3	Minimization of Product-of-Sums Forms	181
4.4	Incompletely Specified Functions	183
4.5	Multiple-Output Circuits	185
4.6	Multilevel Synthesis	189
4.6.1	Factoring	190
4.6.2	Functional Decomposition	193
4.6.3	Multilevel NAND and NOR Circuits	199
4.7	Analysis of Multilevel Circuits	202
4.8	Cubical Representation	207
4.8.1	Cubes and Hypercubes	207
4.9	A Tabular Method for Minimization	211
4.9.1	Generation of Prime Implicants	211
4.9.2	Determination of a Minimum Cover	212
4.9.3	Summary of the Tabular Method	217
4.10	A Cubical Technique for Minimization	219
4.10.1	Determination of Essential Prime Implicants	222
4.10.2	Complete Procedure for Finding a Minimal Cover	224
4.11	Practical Considerations	226
4.12	Examples of Circuits Synthesized from Verilog Code	227

4.13	Concluding Remarks	230
4.14	Examples of Solved Problems	230
	Problems	239
	References	243

**Chapter 5**  
**NUMBER REPRESENTATION AND ARITHMETIC CIRCUITS 247**

5.1	Positional Number Representation	248
5.1.1	Unsigned Integers	248
5.1.2	Conversion between Decimal and Binary Systems	249
5.1.3	Octal and Hexadecimal Representations	250
5.2	Addition of Unsigned Numbers	252
5.2.1	Decomposed Full-Adder	256
5.2.2	Ripple-Carry Adder	256
5.2.3	Design Example	257
5.3	Signed Numbers	259
5.3.1	Negative Numbers	260
5.3.2	Addition and Subtraction	262
5.3.3	Adder and Subtractor Unit	265
5.3.4	Radix-Complement Schemes	266
5.3.5	Arithmetic Overflow	271
5.3.6	Performance Issues	272
5.4	Fast Adders	272
5.4.1	Carry-Lookahead Adder	273
5.5	Design of Arithmetic Circuits Using CAD Tools	280
5.5.1	Design of Arithmetic Circuits Using Schematic Capture	280
5.5.2	Design of Arithmetic Circuits Using Verilog	282
5.5.3	Using Vectored Signals	285
5.5.4	Using a Generic Specification	286
5.5.5	Nets and Variables in Verilog	288
5.5.6	Arithmetic Assignment Statements	289
5.5.7	Representation of Numbers in Verilog Code	293
5.6	Multiplication	295
5.6.1	Array Multiplier for Unsigned Numbers	296
5.6.2	Multiplication of Signed Numbers	297
5.7	Other Number Representations	300
5.7.1	Fixed-Point Numbers	300
5.7.2	Floating-Point Numbers	300

5.7.3	Binary-Coded Decimal Representation	302	7.4	Master-Slave and Edge-Triggered D Flip-Flops	393
5.8	ASCII Character Code	306	7.4.1	Master-Slave D Flip-Flop	393
5.9	Examples of Solved Problems	309	7.4.2	Edge-Triggered D Flip-Flop	393
	Problems	315	7.4.3	D Flip-Flops with Clear and Preset	397
	References	319	7.4.4	Flip-Flop Timing Parameters	398
<b>Chapter 6</b>			7.5	T Flip-Flop	400
<b>COMBINATIONAL-CIRCUIT BUILDING BLOCKS 321</b>			7.5.1	Configurable Flip-Flops	402
6.1	Multiplexers	322	7.6	JK Flip-Flop	402
6.1.1	Synthesis of Logic Functions Using Multiplexers	327	7.7	Summary of Terminology	403
6.1.2	Multiplexer Synthesis Using Shannon's Expansion	328	7.8	Registers	404
6.2	Decoders	335	7.8.1	Shift Register	404
6.2.1	Demultiplexers	339	7.8.2	Parallel-Access Shift Register	405
6.3	Encoders	341	7.9	Counters	405
6.3.1	Binary Encoders	341	7.9.1	Asynchronous Counters	406
6.3.2	Priority Encoders	342	7.9.2	Synchronous Counters	409
6.4	Code Converters	343	7.9.3	Counters with Parallel Load	413
6.5	Arithmetic Comparison Circuits	344	7.10	Reset Synchronization	415
6.6	Verilog for Combinational Circuits	345	7.11	Other Types of Counters	417
6.6.1	The Conditional Operator	345	7.11.1	BCD Counter	417
6.6.2	The If-Else Statement	348	7.11.2	Ring Counter	418
6.6.3	The Case Statement	350	7.11.3	Johnson Counter	420
6.6.4	The For Loop	357	7.11.4	Remarks on Counter Design	420
6.6.5	Verilog Operators	358	7.12	Using Storage Elements with CAD Tools	421
6.6.6	The Generate Construct	364	7.12.1	Including Storage Elements in Schematics	421
6.6.7	Tasks and Functions	365	7.12.2	Using Verilog Constructs for Storage Elements	422
6.7	Concluding Remarks	367	7.12.3	Blocking and Non-Blocking Assignments	425
6.8	Examples of Solved Problems	368	7.12.4	Non-Blocking Assignments for Combinational Circuits	430
	Problems	375	7.12.5	Flip-Flops with Clear Capability	430
	References	380	7.13	Using Registers and Counters with CAD Tools	432
<b>Chapter 7</b>			7.13.1	Including Registers and Counters in Schematics	432
<b>FLIP-FLOPS, REGISTERS, COUNTERS, AND A SIMPLE PROCESSOR 383</b>			7.13.2	Using Library Modules in Verilog Code	434
7.1	Basic Latch	385	7.13.3	Using Verilog Constructs for Registers and Counters	435
7.2	Gated SR Latch	387	7.14	Design Examples	441
7.2.1	Gated SR Latch with NAND Gates	389	7.14.1	Bus Structure	441
7.3	Gated D Latch	390	7.14.2	Simple Processor	452
7.3.1	Effects of Propagation Delays	392	7.14.3	Reaction Timer	463
			7.14.4	Register Transfer Level (RTL) Code	469
			7.15	Timing Analysis of Flip-Flop Circuits	470

- 7.16 Concluding Remarks 472
- 7.17 Examples of Solved Problems 473
  - Problems 477
  - References 485

**Chapter 8**  
**SYNCHRONOUS SEQUENTIAL**  
**CIRCUITS 487**

- 8.1 Basic Design Steps 489
  - 8.1.1 State Diagram 489
  - 8.1.2 State Table 491
  - 8.1.3 State Assignment 491
  - 8.1.4 Choice of Flip-Flops and Derivation of Next-State and Output Expressions 493
  - 8.1.5 Timing Diagram 494
  - 8.1.6 Summary of Design Steps 496
- 8.2 State-Assignment Problem 499
  - 8.2.1 One-Hot Encoding 503
- 8.3 Mealy State Model 505
- 8.4 Design of Finite State Machines Using CAD Tools 509
  - 8.4.1 Verilog Code for Moore-Type FSMs 510
  - 8.4.2 Synthesis of Verilog Code 511
  - 8.4.3 Simulating and Testing the Circuit 514
  - 8.4.4 Alternative Styles of Verilog Code 514
  - 8.4.5 Summary of Design Steps When Using CAD Tools 516
  - 8.4.6 Specifying the State Assignment in Verilog Code 517
  - 8.4.7 Specification of Mealy FSMs Using Verilog 518
- 8.5 Serial Adder Example 520
  - 8.5.1 Mealy-Type FSM for Serial Adder 520
  - 8.5.2 Moore-Type FSM for Serial Adder 523
  - 8.5.3 Verilog Code for the Serial Adder 525
- 8.6 State Minimization 529
  - 8.6.1 Partitioning Minimization Procedure 529
  - 8.6.2 Incompletely Specified FSMs 536
- 8.7 Design of a Counter Using the Sequential Circuit Approach 538
  - 8.7.1 State Diagram and State Table for a Modulo-8 Counter 539
  - 8.7.2 State Assignment 540
  - 8.7.3 Implementation Using D-Type Flip-Flops 540

- 8.7.4 Implementation Using JK-Type Flip-Flops 543
- 8.7.5 Example—A Different Counter 544
- 8.8 FSM as an Arbiter Circuit 546
  - 8.8.1 Implementation of the Arbiter Circuit 552
  - 8.8.2 Minimizing the Output Delays for an FSM 554
  - 8.8.3 Summary 555
- 8.9 Analysis of Synchronous Sequential Circuits 555
- 8.10 Algorithmic State Machine (ASM) Charts 559
- 8.11 Formal Model for Sequential Circuits 563
- 8.12 Concluding Remarks 565
- 8.13 Examples of Solved Problems 565
  - Problems 573
  - References 578

**Chapter 9**  
**ASYNCHRONOUS SEQUENTIAL**  
**CIRCUITS 581**

- 9.1 Asynchronous Behavior 582
- 9.2 Analysis of Asynchronous Circuits 586
- 9.3 Synthesis of Asynchronous Circuits 594
- 9.4 State Reduction 607
- 9.5 State Assignment 622
  - 9.5.1 Transition Diagram 625
  - 9.5.2 Exploiting Unspecified Next-State Entries 628
  - 9.5.3 State Assignment Using Additional State Variables 632
  - 9.5.4 One-Hot State Assignment 637
- 9.6 Hazards 638
  - 9.6.1 Static Hazards 639
  - 9.6.2 Dynamic Hazards 643
  - 9.6.3 Significance of Hazards 644
- 9.7 A Complete Design Example 646
  - 9.7.1 The Vending-Machine Controller 646
- 9.8 Concluding Remarks 651
- 9.9 Examples of Solved Problems 653
  - Problems 661
  - References 665

**Chapter 10**  
**DIGITAL SYSTEM DESIGN 667**

- 10.1 Building Block Circuits 668
  - 10.1.1 Flip-Flops and Registers with Enable Inputs 668
  - 10.1.2 Shift Registers with Enable Inputs 670
  - 10.1.3 Static Random Access Memory (SRAM) 670
  - 10.1.4 SRAM Blocks in PLDs 673
- 10.2 Design Examples 674
  - 10.2.1 A Bit-Counting Circuit 674
  - 10.2.2 ASM Chart Implied Timing Information 676
  - 10.2.3 Shift-and-Add Multiplier 679
  - 10.2.4 Divider 687
  - 10.2.5 Arithmetic Mean 698
  - 10.2.6 Sort Operation 703
- 10.3 Clock Synchronization 712
  - 10.3.1 Clock Skew 714
  - 10.3.2 Flip-Flop Timing Parameters 715
  - 10.3.3 Asynchronous Inputs to Flip-Flops 718
  - 10.3.4 Switch Debouncing 719
- 10.4 Concluding Remarks 719
  - Problems 721
  - References 725

**Chapter 11**  
**TESTING OF LOGIC CIRCUITS 727**

- 11.1 Fault Model 728
  - 11.1.1 Stuck-at Model 728
  - 11.1.2 Single and Multiple Faults 729
  - 11.1.3 CMOS Circuits 729
- 11.2 Complexity of a Test Set 729
- 11.3 Path Sensitizing 731
  - 11.3.1 Detection of a Specific Fault 733
- 11.4 Circuits with Tree Structure 735
- 11.5 Random Tests 736
- 11.6 Testing of Sequential Circuits 739
  - 11.6.1 Design for Testability 739
- 11.7 Built-in Self-Test 743
  - 11.7.1 Built-in Logic Block Observer 747
  - 11.7.2 Signature Analysis 749
  - 11.7.3 Boundary Scan 750
- 11.8 Printed Circuit Boards 750
  - 11.8.1 Testing of PCBs 752
  - 11.8.2 Instrumentation 753

- 11.9 Concluding Remarks 754
  - Problems 754
  - References 757

**Chapter 12**  
**COMPUTER AIDED DESIGN TOOLS 759**

- 12.1 Synthesis 760
  - 12.1.1 Netlist Generation 760
  - 12.1.2 Gate Optimization 760
  - 12.1.3 Technology Mapping 762
- 12.2 Physical Design 768
  - 12.2.1 Placement 768
  - 12.2.2 Routing 770
  - 12.2.3 Static Timing Analysis 771
- 12.3 Concluding Remarks 772
  - References 772

**Appendix A**  
**VERILOG REFERENCE 775**

- A.1 Documentation in Verilog Code 776
- A.2 White Space 776
- A.3 Signals in Verilog Code 776
- A.4 Identifier Names 777
- A.5 Signal Values, Numbers, and Parameters 777
  - A.5.1 Parameters 778
- A.6 Net and Variable Types 778
  - A.6.1 Nets 778
  - A.6.2 Variables 779
  - A.6.3 Memories 780
- A.7 Operators 780
- A.8 Verilog Module 782
- A.9 Gate Instantiations 783
- A.10 Concurrent Statements 785
  - A.10.1 Continuous Assignments 786
  - A.10.2 Using Parameters 787
- A.11 Procedural Statements 788
  - A.11.1 Always and Initial Blocks 788
  - A.11.2 The If-Else Statement 790
  - A.11.3 Statement Ordering 791
  - A.11.4 The Case Statement 792
  - A.11.5 Casez and Casex Statements 793
  - A.11.6 Loop Statements 795

A.11.7	Blocking versus Non-blocking Assignments for Combinational Circuits	798
A.12	Using Subcircuits	798
A.12.1	Subcircuit Parameters	800
A.12.2	The Generate Capability	802
A.13	Functions and Tasks	803
A.14	Sequential Circuits	807
A.14.1	A Gated D Latch	807
A.14.2	D Flip-Flop	807
A.14.3	Flip-Flops with Reset	808
A.14.4	Instantiating a Flip-Flop from a Library	809
A.14.5	Registers	809
A.14.6	Shift Registers	810
A.14.7	Counters	812
A.14.8	An Example of a Sequential Circuit	812
A.14.9	Moore-Type Finite State Machines	814
A.14.10	Mealy-Type Finite State Machines	815
A.15	Guidelines for Writing Verilog Code	816
A.16	Quartus II Verilog Support	822
A.17	Concluding Remarks	822
	References	822

**Appendix B**  
**COMMERCIAL DEVICES 823**

B.1	Simple PLDs	823
B.1.1	The 22V10 PAL Device	823
B.2	Complex PLDs	825
B.2.1	Altera MAX 7000	826
B.3	Field-Programmable Gate Arrays	828
B.3.1	Altera FLEX 10K	828
B.3.2	Xilinx XC4000	832
B.3.3	Altera APEX 20K	833
B.3.4	Altera Stratix	834
B.3.5	Altera Cyclone, Cyclone II, and Cyclone III	835
B.3.6	Altera Stratix II and Stratix III	835
B.3.7	Xilinx Virtex	836
B.3.8	Xilinx Virtex-II, Virtex-II Pro, Virtex-4, and Virtex-5	838
B.3.9	Xilinx Spartan-3	838
B.4	Transistor-Transistor Logic	838
B.4.1	TTL Circuit Families	839
	References	840

**ANSWERS 843**

**INDEX 857**