

Contents

- 1 Introduction 1**
 - 1.1 Evolution of Logic Design 1
 - 1.2 System and Logic Design Abstractions 3
 - 1.3 Integrated Circuit Design and Methodologies 4
 - 1.3.1 RTL Design 4
 - 1.3.2 Functional Verification 5
 - 1.3.3 Synthesis 5
 - 1.3.4 Physical Design 5
 - 1.4 Verilog HDL 5
 - 1.5 Verilog Design Description 7
 - 1.5.1 Structural Design 7
 - 1.5.2 Behavior Design 9
 - 1.5.3 Synthesizable RTL Design 10
 - 1.6 Key Verilog Terminologies 10
 - 1.6.1 Verilog Arithmetic Operators 11
 - 1.6.2 Verilog Logical Operators 11
 - 1.6.3 Verilog Equality and Inequality Operators 11
 - 1.6.4 Verilog Sign Operators 13
 - 1.6.5 Verilog Bitwise Operators 16
 - 1.6.6 Verilog Relational Operators 18
 - 1.6.7 Verilog Concatenation and Replication Operators 18
 - 1.6.8 Verilog Reduction Operators 19
 - 1.6.9 Verilog Shift Operators 20
 - 1.7 Summary 26
- 2 Combinational Logic Design (Part I). 27**
 - 2.1 Introduction to Combinational Logic 27
 - 2.2 Logic Gates and Synthesizable RTL 28
 - 2.2.1 NOT or Invert Logic 28
 - 2.2.2 Two-Input OR Logic 28

2.2.3	Two-Input NOR Logic	28
2.2.4	Two-Input AND Logic	32
2.2.5	Two-Input NAND Logic	33
2.2.6	Two-Input XOR Logic	34
2.2.7	Two-Input XNOR Logic	34
2.2.8	Tri-state Logic	37
2.3	Arithmetic Circuits	38
2.3.1	Adder	38
2.3.2	Subtractor	41
2.3.3	Multi-bit Adders and Subtractors	44
2.3.4	Comparators and Parity Detectors	46
2.3.5	Code Converters	49
2.4	Summary	51
3	Combinational Logic Design (Part II)	53
3.1	Multiplexers	53
3.1.1	Multiplexer as Universal Logic	54
3.2	Decoders	63
3.2.1	1 Line to 2 Decoder Using “case”	63
3.2.2	1 Line to 2 Decoder with Enable Using “case”	63
3.2.3	2 Line to 4 Decoder with Enable Using “case”	63
3.2.4	2 Line to 4 Decoder with Active Low Enable Using ‘case’	67
3.2.5	4 Line to 16 Decoder Using 2:4 Decoder	68
3.3	Encoders	75
3.3.1	Priority Encoders	77
3.4	Summary	78
4	Combinational Design Guidelines	79
4.1	Use of Blocking Assignments and Event Queue	80
4.2	Incomplete Sensitivity List	81
4.3	Continuous Versus Procedural Assignments	82
4.4	Combinational Loops in Design	85
4.5	Unintentional Latches in the Design	88
4.6	Use of Blocking Assignments	89
4.7	Use of If-Else Versus Case Statements	91
4.8	MUX Nested or Priority Structure	92
4.9	Decoder 2:4	92
4.10	Encoder 4:2	93
4.11	Missing ‘Default’ Clause in Case	93
4.12	If-Else with Else Missing	95
4.13	Logical Equality Versus Case Equality	97
4.13.1	Logical Equality and Logical Inequality Operators	97
4.13.2	Case Equality and Case Inequality Operators	97

4.14	Arithmetic Resource Sharing	98
4.14.1	With Resource Sharing	98
4.15	Multiple Driver Assignments	102
4.16	Summary	102
5	Sequential Logic Design	103
5.1	Sequential Logic	103
5.1.1	Positive Level Sensitive D-Latch	104
5.1.2	Negative Level Sensitive D Latch.	106
5.2	Flip-Flop.	107
5.2.1	Positive Edge Triggered D Flip-Flop.	108
5.2.2	Negative Edge Triggered D Flip-Flop	108
5.3	Synchronous and Asynchronous Reset	109
5.3.1	D Flip-Flop Asynchronous Reset	109
5.3.2	D Flip-Flop Synchronous Reset	111
5.3.3	Flip-Flop with Load Enable Asynchronous Reset	112
5.3.4	Flip-Flop with Synchronous Load and Synchronous Reset	112
5.4	Synchronous Counters	114
5.4.1	Three Bit Up Counter	114
5.4.2	Three-Bit Down Counter	118
5.4.3	Three-Bit Up-Down Counter	120
5.4.4	Gray Counters	121
5.4.5	Gray and Binary Counter.	123
5.4.6	Ring Counters	125
5.4.7	Johnson Counters	127
5.4.8	Parameterized Counter.	130
5.5	Shift Register.	130
5.5.1	Right and Left Shift	130
5.5.2	Parallel Input and Parallel Output (PIPO) Shift Register.	132
5.6	Timing and Performance Evaluation	138
5.7	Asynchronous Counter Design	138
5.7.1	Ripple Counters	140
5.8	Memory Modules and Design	140
5.9	Summary	144
6	Sequential Design Guidelines	145
6.1	Use of Blocking Assignments	146
6.1.1	Blocking Assignments and Multiple “Always” Blocks.	146
6.1.2	Blocking Assignments in the Same “Always” Block	146
6.1.3	Example Blocking Assignment.	149

6.2	Nonblocking Assignments	150
6.2.1	Example Nonblocking Assignment	150
6.2.2	Ordering on Non-blocking Assignments	153
6.3	Latch Versus Flip-Flop	154
6.3.1	D Flip-Flop	154
6.3.2	Latch	154
6.4	Use of Synchronous Versus Asynchronous Reset	156
6.4.1	Asynchronous Reset D Flip-Flop	157
6.4.2	Synchronous Reset D Flip_Flop	157
6.5	Use of If-Else Versus Case Statements	157
6.6	Internally Generated Clocks	157
6.7	Gated Clocks	161
6.8	Use of Pipelining in Design	161
6.8.1	Design Without Pipelining	162
6.8.2	Design with Pipelining	163
6.9	Guidelines for Modeling Synchronous Designs	163
6.10	Multiple Clocks in the Same Module	163
6.11	Multi Phase Clocks in the Design	165
6.12	Guidelines for Modeling Asynchronous Designs	169
6.13	Summary	169
7	Complex Designs Using Verilog RTL	171
7.1	ALU Design	172
7.1.1	Logical Unit Design	172
7.1.2	Arithmetic Unit	179
7.1.3	Arithmetic and Logical Unit	181
7.2	Functions and Tasks	184
7.2.1	Counting 1's from the Given String	185
7.2.2	Module to Count 1's using Functions	185
7.3	Parity Generators and Detectors	187
7.3.1	Parity Generator	187
7.3.2	Add_Sub_Parity Checker	189
7.4	Barrel Shifters	192
7.5	Summary	195
8	Finite State Machines	197
8.1	Moore Versus Mealy Machines	198
8.1.1	Level to Pulse Converter	200
8.2	FSM Encoding Styles	205
8.2.1	Binary Encoding	206
8.2.2	Gray Encoding	208
8.3	One-Hot Encoding	210
8.4	Sequence Detectors Using FSMs	212

8.4.1	Sequence Detector Using Mealy Machine Two Always Blocks	212
8.4.2	Sequence Detector Using Mealy Machine for ‘101’ Sequence	215
8.5	Improving the Design Performance for FSMs	215
8.6	Summary	217
9	Simulation Concepts and PLD-Based Designs	219
9.1	Key Simulation Concepts	219
9.1.1	Simulation for Blocking and Nonblocking Assignments	220
9.1.2	Blocking Assignments with Inter-assignment Delays.	222
9.1.3	Blocking Assignments with Intra-assignment Delays.	224
9.1.4	Nonblocking Assignments with Inter-assignment Delays.	224
9.1.5	Nonblocking Assignments with Intra-assignment Delays.	226
9.2	Simulation Using Verilog	226
9.3	Introduction to PLD	230
9.4	FPGA as Programmable ASIC	233
9.4.1	SRAM Based FPGA	233
9.4.2	Flash-Based FPGA	233
9.4.3	Antifuse FPGAS.	234
9.4.4	FPGA Building Blocks	235
9.5	FPGA Design Flow	236
9.5.1	Design Entry	237
9.5.2	Design Simulation and Synthesis	238
9.5.3	Design Implementation	238
9.5.4	Device Programming.	238
9.6	Logic Realization Using FPGA	239
9.6.1	Configurable Logic Block	239
9.6.2	Input–Output Block (IOB).	240
9.6.3	Block RAM.	241
9.6.4	Digital Clock Manager (DCM) Block	242
9.6.5	Multiplier Block	242
9.7	Design Guidelines for FPGA-Based Designs	243
9.7.1	Verilog Coding Guidelines.	243
9.7.2	FSM Guidelines	244
9.7.3	Combinational Design and Combinational Loops	245
9.7.4	Grouping the Terms	245
9.7.5	Assignments	245
9.7.6	Simulation and Synthesis Mismatch	245
9.7.7	Post-synthesis Verification	246

9.7.8	Guidelines for Area Optimization	246
9.7.9	Guidelines for Clock	247
9.7.10	Synchronous Versus Asynchronous Designs.	248
9.7.11	Guidelines for Use of Reset	249
9.7.12	Guidelines for CDC	250
9.7.13	Guidelines for Low Power Design	251
9.7.14	Guidelines for Use of Vendor-Specific IP Blocks	251
9.8	Summary	252
	References	253
10	ASIC RTL Synthesis	255
10.1	What Is ASIC?	256
10.1.1	Full-Custom ASIC	256
10.1.2	Standard Cell ASIC	256
10.1.3	Gate Array ASIC	257
10.2	ASIC Design Flow	257
10.2.1	Design Specification	257
10.2.2	RTL Design and Verification	259
10.2.3	ASIC Synthesis	259
10.2.4	Physical Design and Implementation	260
10.3	ASIC Synthesis Using Design Compiler	261
10.4	ASIC Synthesis Guidelines	263
10.5	Constraining Design Using Synopsys DC	264
10.5.1	Reading the Design.	264
10.5.2	Checking of the Design.	265
10.5.3	Clock Definitions	265
10.5.4	Skew Definition	266
10.5.5	Defining Input and Output Delay	267
10.5.6	Defining Minimum (Min) and Maximum (Max) Delay	267
10.5.7	Design Synthesis	267
10.5.8	Saving the Design.	267
10.6	Synthesis Optimization Techniques.	268
10.6.1	Resource Allocation	269
10.6.2	Common Factors and Sub-expressions Use for Optimization.	270
10.6.3	Moving the Piece of Code	271
10.6.4	Constant Folding	272
10.6.5	Dead Zone Elimination	273
10.6.6	Use of Parentheses	273
10.6.7	Partitioning and Structuring the Design	274
10.7	Summary	274
	Reference	275

11	Static Timing Analysis	277
11.1	Setup Time	278
11.2	Hold Time	279
11.3	Clock to Q Delay	280
11.3.1	Frequency Calculations	280
11.4	Skew in Design	282
11.5	Timing Paths in Design	284
11.5.1	Input-to-Register Path	284
11.5.2	Register-to-Output Path	284
11.5.3	Register-to-Register Path	285
11.5.4	Input-to-Output Path	286
11.6	Timing Goals for the Design	286
11.7	Min-Max Analysis for ASIC Design	287
11.8	Fixing Design Violations	289
11.8.1	Changes at the Architecture Level	289
11.8.2	Changes at Microarchitecture Level	290
11.8.3	Optimization During Synthesis	290
11.9	Fixing Setup Violations in the Design	291
11.9.1	Logic Duplication	291
11.9.2	Encoding Methods	292
11.9.3	Late Arrival Signals	293
11.9.4	Register Balancing	293
11.10	Hold Violation Fix	294
11.11	Timing Exceptions in the Design	295
11.11.1	Asynchronous and False Paths	295
11.11.2	Multicycle Paths	296
11.12	Pipelining and Performance Improvement	297
11.13	Summary	298
	Reference	298
12	Constraining ASIC Design	299
12.1	Introduction to Design Constraints	300
12.2	Compilation Strategy	304
12.2.1	Top-Down Compilation	304
12.2.2	Bottom-Up Compilation	305
12.3	Area Minimization Techniques	306
12.3.1	Avoid Use of Combinational Logic as Individual Block	306
12.3.2	Avoid Use of Glue Logic Between Two Modules	307
12.3.3	Use of set_max_area Attribute	308
12.3.4	Area Report	308
12.4	Timing Optimization and Performance Improvement	309
12.4.1	Design Compilation with 'map_effort high'	309
12.4.2	Logical Flattening	309

12.4.3	Use of group_path Command.	310
12.4.4	Submodule Characterizing	311
12.4.5	Register Balancing	313
12.4.6	FSM Optimization	314
12.4.7	Fixing Hold Violations	315
12.4.8	Report Command	315
12.5	Constraint Validation	318
12.6	Commands for the DRC, Power, and Optimization.	318
12.7	Summary	319
	References	320
13	Multiple Clock Domain Design.	321
13.1	What Is Multiple Clock Domain?	322
13.2	What Is Clock Domain Crossing (CDC)	322
13.3	Level Synchronizers	327
13.4	Pulse Synchronizers	330
13.5	MUX Synchronizer	331
13.6	Challenges in the Design of Synchronizers	331
13.7	Data Path Synchronizers	338
	13.7.1 Handshaking Mechanism.	338
	13.7.2 FIFO Synchronizer	340
	13.7.3 Gray Encoding.	341
13.8	Design Guidelines for the Multiple Clock Domain Designs.	343
13.9	FIFO Depth Calculations.	347
13.10	Case Study	351
13.11	Summary	358
14	Low Power Design.	359
14.1	Introduction to Low Power Design	359
14.2	Power Dissipation in CMOS Inverter	360
14.3	Switching and Leakage Power Reduction Techniques	363
	14.3.1 Clock Gating and Clock Tree Optimizations	364
	14.3.2 Operand Isolations	364
	14.3.3 Multiple V_{th}	364
	14.3.4 Multiple Supply Voltages (MSV)	364
	14.3.5 Dynamic Voltage and Frequency Scaling (DVFS).	365
	14.3.6 Power Gating (Power Shut-Off)	365
	14.3.7 Isolation Logic	365
	14.3.8 State Retention.	366
14.4	Low Power Design Techniques at the RTL Level.	366

14.5	Low Power Design Architecture and UPF Case Study	370
14.5.1	Isolation Cells	371
14.5.2	Retention Cells.	372
14.5.3	Level Shifters.	374
14.5.4	Power Sequencing and Scheduling	374
14.6	Summary	380
	References	380
15	System on Chip (SOC) Design	381
15.1	What is System on Chip (SOC)?	382
15.2	SOC Architecture.	382
15.3	SOC Design Flow	383
15.3.1	IP Design and Reuse.	383
15.3.2	SOC Design Considerations.	385
15.3.3	Hardware Software Codesign	386
15.3.4	Interface Timings	386
15.3.5	EDA Tool and License Requirements	387
15.3.6	Developing the Required Prototyping Platform.	387
15.3.7	Developing the Test Plan.	388
15.3.8	Developing the Verification Environment.	388
15.3.9	Prototyping Using FPGAs	388
15.3.10	ASIC Porting	389
15.4	SOC Design Challenges	390
15.5	Case Study	392
15.6	SOC Design Blocks	392
15.6.1	Microprocessors or Microcontrollers	392
15.6.2	Counters and Timers.	393
15.6.3	General Purpose IO Block	395
15.6.4	Universal Asynchronous Receiver and Transmitter (UART).	396
15.6.5	Bus Arbitration Logic	397
15.7	Summary	398
Appendix I: Synthesizable and Non-Synthesizable Verilog Constructs		399
Appendix II: Xilinx Spartan Devices		401
Appendix III: Design For Testability		405
Index		409

<http://www.springer.com/978-81-322-2789-2>

Digital Logic Design Using Verilog

Coding and RTL Synthesis

Taraate, V.

2016, XXIII, 416 p. 267 illus., 226 illus. in color.,

Hardcover

ISBN: 978-81-322-2789-2