

# Contents

Preface xv

Acknowledgments xvii

## 1 Review of Combinational Circuits 1

- 1.1 Combinational Circuits 1
- 1.2 Fundamental Logic Gates 2
- 1.3 Chain-Type versus Tree-Type Structures 3
- 1.4 Examples of Combinational Logic Circuits 4
  - 1.4.1 Multiplexer 4
  - 1.4.2 Address Decoder 5
  - 1.4.3 Parity Detector 6
  - 1.4.4 Priority Encoder 6
  - 1.4.5 Binary-to-BCD Converters 6
- 1.5 Examples of Combinational Arithmetic Circuits 10
  - 1.5.1 Full-Adder Unit 10
  - 1.5.2 Carry-Ripple Adder 11
  - 1.5.3 Faster Adders (Manchester, Carry-Lookahead, and Kogge-Stone Tree) 11
  - 1.5.4 Adder Arrays 12
  - 1.5.5 Subtracters 13
  - 1.5.6 Incrementer, Decrementer, and Two's Complementer 14
  - 1.5.7 Parallel Multiplier 14
  - 1.5.8 Comparators (Equality and Greater-than/Equal-to) 14
  - 1.5.9 Arithmetic Logic Unit 15
- 1.6 Binary Arithmetic 16
  - 1.6.1 Carry Bit and Overflow Flag 16
  - 1.6.2 Unsigned Integer Arithmetic 17
  - 1.6.3 Signed Integer Arithmetic 19
  - 1.6.4 Extension, Truncation, Rounding, and Saturation 21
  - 1.6.5 Floating-Point Arithmetic 24

—1  
—0  
—+1

<b>2</b>	<b>Review of Sequential Circuits</b>	<b>29</b>
2.1	Sequential Circuits	29
2.2	Latches	29
2.3	Flip-Flops	31
2.4	Glitch Analysis and Prevention	34
2.5	Register Transfer Level of Abstraction	35
2.6	Initial Examples of Sequential Circuits	36
2.6.1	Shift Registers	36
2.6.2	Synchronous Modulo- $2^N$ Counters	37
2.6.3	Synchronous Modulo- $M$ Counters	38
2.6.4	Asynchronous Counters	38
2.6.5	Gray, Johnson, and One-Hot Counters	39
2.6.6	Signal Generators	41
2.6.7	Clock Dividers	41
2.6.8	Timers and Sequential Binary-to-BCD Converters	41
2.6.9	Tapped Delay Line	41
2.7	Clock Division	42
2.7.1	Common Clock Division Cases	42
2.7.2	Clock Division by Any Integer with Symmetric Phase	43
2.7.3	The Cost of Clock Division by an Even Integer	44
2.7.4	Breaking a Large Clock Divider into Smaller Serial Clock Dividers	46
2.8	Clock Multiplication and Phase-Locked Loops (PLLs)	46
2.9	Asynchronous Data and Synchronizers	48
2.9.1	Clock-Domain Crossing	49
2.9.2	A Practical Example: Frequency Meters	49
2.9.3	Dealing with Reset	52
2.10	Clock Gating	53
2.11	Additional Examples of Sequential Circuits	54
2.11.1	One-Shot and Pulse-Capturer Circuits	54
2.11.2	Switch Debouncers	55
2.11.3	Reference-Value Generators	58
2.11.4	Pulse Width Modulator	59
2.11.5	Pseudo-Random Sequence Generators	60
2.11.6	Digital Finite Impulse Response (FIR) Filters	61
2.11.7	Digital Infinite Impulse Response (IIR) Filters	64
2.11.8	Serializer and Deserializer Circuits	65
<b>3</b>	<b>Review of Finite State Machines</b>	<b>67</b>
3.1	Finite State Machines	67
3.2	State Transition Diagram and Machine Types	68
3.3	Representing versus Implementing	70

- 3.4 Moore-to-Mealy and Mealy-to-Moore Conversion 70
- 3.5 Time Behavior of Moore versus Mealy Machines 71
- 3.6 Choosing Between Moore and Mealy Machines 73
- 3.7 Transition Types 74
- 3.8 Incorrect State Transition Diagrams 75
- 3.9 Safe State Machines 78
- 3.10 Fundamental Hardware Architectures for FSMs 78
- 3.11 Encoding Styles 79
- 3.12 Fundamental Design Technique for FSMs 79
- 3.13 State Machine Categories 82
- 3.14 Dealing with Time 83
- 3.15 Dealing with Repetitive States 85
- 3.16 Pointer-Based FSM Implementation 86
- 3.17 Dealing with Recursivity 86
- 3.18 Number of Flip-Flops in FSMs 88
- 3.19 Examples of Category 1 (Regular) State Machines 88
  - 3.19.1 Arbiter 89
  - 3.19.2 Garage Door Controller 89
  - 3.19.3 Datapath Control for a Greatest Common Divisor 90
- 3.20 Examples of Category 2 (Timed) State Machines 93
  - 3.20.1 Car Alarm 93
  - 3.20.2 Password Decoder 94
  - 3.20.3 Serial Peripheral Interface for an A/D Converter 95
- 3.21 Examples of Category 3 (Recursive) State Machines 96
  - 3.21.1 SRAM Memory Interface 97
  - 3.21.2 Datapath Controller for a Serial Multiplier 97
  - 3.21.3 Reference-Value Definer with Embedded Debouncers 99
- 4 Review of Field Programmable Gate Arrays (FPGAs) 101**
  - 4.1 Programmable Logic Devices 101
  - 4.2 PLD Configuration Memories 103
  - 4.3 PAL and PLA Devices 104
  - 4.4 GAL Devices 104
  - 4.5 CPLD Devices 105
  - 4.6 FPGA Devices 107
- 5 Introduction to VHDL 115**
  - 5.1 About VHDL 115
  - 5.2 Translation of VHDL Code into a Circuit 115
  - 5.3 Design Flow 117
  - 5.4 Commercial VHDL Tools 119

—1  
—0  
—+1

5.5	RTL Design Approach	119
5.6	Concurrent versus Sequential Statements	119
5.7	Lexical Elements of VHDL	120
5.7.1	Assignment Symbols	120
5.7.2	Comments	120
5.7.3	Bit and Bit String	121
5.7.4	Integers	121
5.7.5	Character and Character String	122
5.7.6	Identifiers	122
5.7.7	Delimiters	122
5.7.8	Reserved VHDL Words	123
5.8	Choosing Good Names for Your Design	123
5.8.1	Naming an Entity Declaration (“The Design”)	124
5.8.2	Naming an Architecture Body	124
5.8.3	Naming Constants	124
5.8.4	Naming Signals and Variables	125
5.8.5	Naming Functions and Procedures	125
5.8.6	Naming Types	126
5.8.7	Naming Files	127
<b>6</b>	<b>Code Structure and Composition</b>	<b>129</b>
6.1	Design Units and Code Structure	129
6.2	Libraries and Packages	130
6.3	Packages List in the Code	134
6.4	Entity Declaration	135
6.5	Architecture Body	138
6.6	Object Classes	138
6.6.1	Constant	139
6.6.2	Signal	140
6.6.3	Variable	141
6.6.4	File	142
6.7	Generics	143
6.8	Entity-Architecture Binding	145
6.9	Introductory VHDL Examples	145
<b>7</b>	<b>Predefined Data Types</b>	<b>153</b>
7.1	Predefined VHDL Types	153
7.2	Type Classes	154
7.3	Type Declarations	156
7.4	Subtypes	158
7.5	A Note on Operators and Attributes	159

- 7.6 Study of Predefined Data Types 159
  - 7.6.1 Standard Types 160
  - 7.6.2 Standard-Logic Types 162
  - 7.6.3 Unsigned and Signed Types 165
  - 7.6.4 Fixed-Point Types 166
  - 7.6.5 Floating-Point Types 168
  - 7.6.6 Type *real* 170
- 7.7 Record Types 172
- 7.8 Access Types, File Types, and Protected Types 173
- 7.9 Aggregation, Concatenation, and Resizing 173
  - 7.9.1 Data Aggregation 173
  - 7.9.2 Data Concatenation 175
  - 7.9.3 Resizing Data Arrays 175
- 7.10 Type Conversion 178
  - 7.10.1 Automatic Conversion 178
  - 7.10.2 Type Cast 179
  - 7.10.3 Type-Conversion Functions 179
  - 7.10.4 Strength-Stripping Functions 182
- 7.11 Type-Qualification Expressions 183
- 7.12 Additional Examples 183
- 7.13 Exercises 185

**8 User-Defined Data Types 193**

- 8.1 Review of Synthesizable Predefined Types 193
- 8.2 User-Defined Types 193
  - 8.2.1 Integer Types 193
  - 8.2.2 Enumeration Types 195
  - 8.2.3 Array Types 195
- 8.3 Building and Addressing Complex Array Types 196
  - 8.3.1 Array Dimensionality 196
  - 8.3.2 Predefined 1D Arrays 197
  - 8.3.3 Building 1D×1D Arrays 197
  - 8.3.4 Building 2D Arrays 198
  - 8.3.5 Building 1D×1D×1D Arrays 199
  - 8.3.6 Building 3D Arrays 199
- 8.4 Checking and Resetting Data Arrays 199
  - 8.4.1 Zeroing Entire Data Arrays 200
  - 8.4.2 Checking Whether Data Arrays Contain Only Zeros 200
- 8.5 Classical Mistakes in Assignments 201
- 8.6 Additional Examples 204
- 8.7 Exercises 208

—1  
—0  
—+1

<b>9</b>	<b>Operators and Attributes</b>	213
9.1	Predefined Operators	213
9.1.1	Logical Operators	214
9.1.2	Arithmetic Operators	218
9.1.3	Comparison (Relational) Operators	223
9.1.4	Shift Operators	226
9.1.5	Concatenation Operator	227
9.1.6	Condition Operator	227
9.2	User-Defined Overloaded Operators	228
9.3	Predefined Attributes	229
9.3.1	Attributes of Scalar Types	229
9.3.2	Attributes of Array Types and Objects	230
9.3.3	Attributes of Signals	231
9.3.4	Attributes of Named Entities	231
9.4	User-Defined Attributes	232
9.5	Synthesis Attributes	234
9.5.1	State Machine Encoding Attributes	234
9.5.2	Safe State Machine Attributes	235
9.5.3	Keep-Logic Attribute	236
9.5.4	ROM and RAM Implementation Attributes	237
9.6	Group	237
9.7	Alias	238
9.8	Exercises	239
<b>10</b>	<b>Concurrent Code</b>	243
10.1	Concurrent Statements	243
10.2	The <i>when</i> Statement	246
10.3	The <i>select</i> Statement	248
10.4	The <i>generate</i> Statement	249
10.5	Component Instantiation Statements	252
10.5.1	Component Instantiation	252
10.5.2	Design Entity Instantiation	253
10.6	Avoiding Multiple Assignments to the Same Signal	256
10.7	Suggested Approaches for Arithmetic Circuits	259
10.8	Additional Examples and Exercises	264
<b>11</b>	<b>Concurrent Code: Practice</b>	265
11.1	Additional Design Examples Using Concurrent Code	265
	Example 11.1. Vectors Absolute Difference Calculator	265
	Example 11.2. Programmable Combinational Delay Line (Structural)	268
	Example 11.3. Sine Calculator with Integers and ROM-Type Memory	270

- 11.2 Exercises 273
  - Part 1: Combinational *Logic* Circuits 274
  - Part 2: Combinational Arithmetic Circuits 277
  - Part 3: With Component Instantiation (Structural Code) 281

**12 Sequential Code 283**

- 12.1 Concurrent Code versus Sequential Code 283
- 12.2 Detecting Clock Transitions: *clk'event* or *rising\_edge(clk)?* 284
- 12.3 The *process* Statement 285
- 12.4 The *if* Statement 288
- 12.5 The *case* Statement 291
- 12.6 The *wait* Statement 294
- 12.7 The *loop* Statement 295
- 12.8 The Sequential *when* and *select* Statements 297
- 12.9 Signal versus Variable 298
- 12.10 More about the Updating Rule of Signals and Variables 301
- 12.11 More about the Inference of Registers Rule 307
- 12.12 The Problem of Combinational Loops 312
- 12.13 Additional Examples and Exercises 313

**13 Sequential Code: Practice 315**

- 13.1 Additional Design Examples Using Sequential Code 315
  - Example 13.1. Generic Tree-Type Adder Array 315
  - Example 13.2. Single-Switch Debouncer 318
  - Example 13.3. FIR Filter with Fixed Coefficients 320
  - Example 13.4. Sequential Square-Root Calculator 322
- 13.2 Exercises 326
  - Part 1: Signal versus Variable 326
  - Part 2: Combinational Circuits 330
  - Part 3: Counters and Clock Dividers 330
  - Part 4: Timers and Associated Circuits 332
  - Part 5: Synchronism 337
  - Part 6: Shifters 339
  - Part 7: Controllers 340
  - Part 8: Serial Arithmetic Circuits 343
  - Part 9: Filters 347
  - Part 10: With Component Instantiation (Structural Code) 347

**14 Packages and Subprograms 349**

- 14.1 Package 349
- 14.2 Package with Generics 351

—1  
—0  
—+1

14.3	Function	353
14.4	Procedure	359
14.5	Function versus Procedure Summary	363
14.6	Subprogram with Generics and Generic Subprograms	364
14.7	Overloaded Subprograms	366
14.8	Assert and Report Statements	367
14.9	Exercises	369
<b>15</b>	<b>The Case of State Machines</b>	<b>373</b>
15.1	The Finite State Machine Approach	373
15.2	State Encoding Styles	375
15.3	VHDL for Regular (Category 1) State Machines	375
15.3.1	Hardware Architecture of Regular State Machines	376
15.3.2	Simple Moore-to-Mealy Conversion	376
15.3.3	VHDL Templates for Regular State Machines	378
15.4	VHDL for Timed (Category 2) State Machines	387
15.4.1	Hardware Architecture of Timed State Machines	387
15.4.2	VHDL Templates for Timed State Machines	388
15.5	VHDL for Recursive (Category 3) State Machines	395
15.5.1	Hardware Architecture of Recursive State Machines	395
15.5.2	VHDL Templates for Recursive State Machines	395
15.6	Summarizing (and Simplifying) Things	402
15.7	Exercises	402
<b>16</b>	<b>The Case of State Machines: Practice</b>	<b>407</b>
16.1	Design Examples of Regular (Category 1) State Machines	407
16.2	Design Examples of Timed (Category 2) State Machines	412
16.3	Design Examples of Recursive (Category 3) State Machines	416
16.4	Exercises	421
	Part 1: Exercises with Regular FSMs	421
	Part 2: Exercises with Timed FSMs	422
	Part 3: Exercises with Recursive FSMs	425
16.5	Exercises with SPI, I <sup>2</sup> C, and LCD Interfaces	426
<b>17</b>	<b>Additional Design Examples</b>	<b>427</b>
17.1	Additional Design Examples	427
	Example 17.1. SPI Interface for an EEPROM Device (with FSM)	428
	Example 17.2. SPI Interface for an EEPROM Device (with Pointer)	434
	Example 17.3. I <sup>2</sup> C Interface for an A/D Converter (with Pointer)	440
	Example 17.4. I <sup>2</sup> C Interface for an A/D Converter (with Pointer Built with FSM)	444



Example 17.5 Digital Watch with Liquid Crystal Display (LCD) 447

Example 17.6 VGA Video Interface for a Hardware-Generated Image 454

Example 17.7 DVI Video Interface for a Hardware-Generated Image 458

Example 17.8 TMDS 8B/10B Encoder 464

17.2 Exercises 468

    Part 1: Exercises with SPI Protocol 468

    Part 2: Exercises with I<sup>2</sup>C Protocol 469

    Part 3: Exercises with Alphanumeric LCD 471

    Part 4: Exercises with VGA Video Driver 472

    Part 5: Exercises with DVI Video Driver 473

**18 Introduction to Simulation with Testbenches 475**

    18.1 Testbenches 475

    18.2 Dealing with Time in VHDL 476

    18.3 Stimuli Generation 478

    18.4 Complete Testbenches 482

    18.5 Practical Considerations on Functional and Timing Simulations 489

    18.6 Dealing with Data Files 490

    18.7 Running Simulation with Tcl Scripts 495

    18.8 Exercises 495

Appendix A: Vivado Tutorial 497

Appendix B: Quartus Prime Tutorial 511

Appendix C: ModelSim Tutorial 521

Appendix D: Simulation Analysis and Recommendations 531

Appendix E: Using Seven-Segment Displays with VHDL 533

Appendix F: Serial Peripheral Interface 537

Appendix G: I<sup>2</sup>C (Inter Integrated Circuits) Interface 541

Appendix H: Alphanumeric LCD 545

Appendix I: VGA Video Interface 551

Appendix J: DVI Video Interface 555

Appendix K: TMDS Link 559

Appendix L: Using Phase-Locked Loops with VHDL 563

Appendix M: List of Enumerated Examples and Exercises 571

Bibliography 581

Index 583