3 Design Challenges  53
  3.1 Complexity  53
  3.2 Changes  55
  3.3 Dependencies  56
  3.4 Duplicate Information  57
  3.5 Technical Debt  58
  3.6 Your Wetware  60
    3.6.1 Recognition versus Recall  61

4 Design Principles  63
  4.1 Modularity  63
    4.1.1 What Is a Module?  63
    4.1.2 Key Benefits of Modularity  64
    4.1.3 Effectively Decomposing a System into Modules  65
  4.2 Abstraction  69
    4.2.1 Levels of Abstraction  69
    4.2.2 Key Benefits of Abstraction  71
    4.2.3 Using Abstraction Effectively  72
  4.3 Hierarchy  74
    4.3.1 Key Benefits of Hierarchy  75
  4.4 Loose Coupling  78
    4.4.1 Types of Coupling  79
    4.4.2 Decoupling  80
  4.5 Strong Cohesion  81
  4.6 The Single Responsibility Principle  83
  4.7 Orthogonality  84
    4.7.1 Guidelines for Creating Orthogonal Systems  85
  4.8 Single Source of Truth (The DRY Principle)  86

Part II Basic Elements of VHDL  89

5 Analysis, Elaboration, and Execution  91
  5.1 VHDL in the System Design Flow  91
  5.2 The Design Processing Flow  93
  5.3 Overview of the Analysis and Elaboration Process  96
  5.4 Detailed View of the Analysis Phase  98
    5.4.1 Lexical Analysis  98
    5.4.2 Lexical Elements  99
    5.4.3 Syntax Analysis  101
    5.4.4 Semantic Analysis  103
  5.5 Detailed View of the Elaboration Phase  104
5.6 The Execution Phase 107
  5.6.1 Discrete-Event Simulation 107
  5.6.2 Drivers 108
  5.6.3 Initialization Phase 111
  5.6.4 The Simulation Cycle 111

6 VHDL Design Units 113
  6.1 Design Units 113
    6.1.1 Entity Declaration 114
    6.1.2 Architecture Body 118
    6.1.3 Package Declaration and Package Body 119
  6.2 Design Libraries and Library Units 121
  6.3 Guidelines and Recommendations for VHDL Design Units 123

7 Statements, Declarations, and Expressions 129
  7.1 Statements 129
  7.2 Declarations 132
    7.2.1 Scope 132
    7.2.2 Visibility 133
    7.2.3 Overloading 134
    7.2.4 Scope, Visibility, and Overloading 136
    7.2.5 Use Clauses 136
    7.2.6 Live Time and Span 139
  7.3 Expressions 140
  7.4 Guidelines and Recommendations for Statements, Declarations, and Expressions 142

8 Operators, Operands, and Attributes 147
  8.1 Operators 147
    8.1.1 Condition Operator (??) 149
    8.1.2 Logical Operators 150
    8.1.3 Relational Operators 152
    8.1.4 Shift Operators 154
    8.1.5 Adding Operators 154
    8.1.6 Operator Overloading 156
  8.2 Operands 158
    8.2.1 Literals 159
    8.2.2 Aggregates 161
    8.2.3 Names 165
    8.2.4 Aliases 168
    8.2.5 Other Kinds of Operands 171
8.3 Attributes 172
8.4 Guidelines and Recommendations for Operators, Operands, and Attributes 180

Part III Statements 185

9 Concurrent Statements 187
9.1 Introduction 187
9.2 The process Statement 188
  9.2.1 The Ins and Outs of VHDL Processes 189
  9.2.2 The Sensitivity List 192
  9.2.3 Equivalence between Processes and Other Concurrent Statements 198
  9.2.4 Guidelines and Recommendations for Writing Processes 201
9.3 The Concurrent Procedure Call Statement 203
9.4 The Component Instantiation Statement 205
9.5 The generate Statement 209
  9.5.1 General Remarks about generate Statements 216
9.6 Other Concurrent Statements 217
  9.6.1 The block Statement 217
  9.6.2 The Concurrent Signal Assignment Statement 220
  9.6.3 The Concurrent Assertion Statement 221

10 Sequential Statements 223
10.1 Control Structures 223
  10.1.1 The Problems with Deep Nesting 224
10.2 The if Statement 224
  10.2.1 Guidelines for the if Statement 225
10.3 The case Statement 230
  10.3.1 Guidelines for the case Statement 232
10.4 Loops 234
  10.4.1 Kinds of Loops 234
  10.4.2 Auxiliary Loop Control Statements 236
  10.4.3 Loops in Hardware 238
  10.4.4 Guidelines for Using Loops 242
10.5 The wait Statement 246
  10.5.1 Synthesizable wait Statements 250
10.6 The assert Statement 251
  10.6.1 When to Use an Assertion 251
  10.6.2 When Not to Use an Assertion 253
  10.6.3 Severity Levels 253
10.7 The null Statement 253
19.5 Initialization and Reset  516
19.6 Pragmas  518
   19.6.1 Synthesis Attributes  518
   19.6.2 Metacomments  521
20  Testbenches  523
20.1 Introduction  523
   20.1.1 Test or Verification?  525
   20.1.2 Functional Verification  525
   20.1.3 What Should We Test?  525
   20.1.4 Self-Checking Testbenches  527
   20.1.5 Basic Parts of a Testbench  527
   20.1.6 What is the Output of a Testbench?  528
   20.1.7 Test-Driven Development  528
20.2 Example #1: Testbench for a Combinational Circuit  530
20.3 Example #2: Testbench for a Clocked Circuit  541
20.4 Example #3: Testbench for an FSM  547
20.5 Example #4: Testbenches with File I/O  555
20.6 Random Stimulus and Functional Coverage  559
   20.6.1 Functional Coverage  559
   20.6.2 Directed Testing  559
   20.6.3 Random-Based Verification  559
   20.6.4 Intelligent Coverage  559
   20.6.5 Coverage Bins  560
   20.6.6 VHDL and Random-Based Verification  560
20.7 Example #5: Testbench with Random Stimulus and Functional Coverage  561
20.8 Example #6: FSM Transition Coverage  566
20.9 Conclusion  572

Notes  575
Bibliography  583
Index  585