Table of Contents

Foreword xiii

Preface to the Third Edition xv

Acknowledgements xvii

1 Introduction 1

1.1 Goals of This Manual 2
1.1.1 Assumptions 3
1.1.2 Definitions 3
1.1.3 Virtual Socket Interface Alliance 4
1.2 Design for Reuse: The Challenge 4
1.2.1 Design for Use 5
1.2.2 Design for Reuse 5
1.3 The Emerging Business Model for Reuse 6

2 The System-on-Chip Design Process 9

2.1 A Canonical SoC Design 9
2.2 System Design Flow 11
2.2.1 Waterfall vs. Spiral 11
2.2.2 Top-Down vs. Bottom-Up 15
2.2.3 Construct by Correction 15
2.2.4 Summary 16

2.3 The Specification Problem 17
2.3.1 Specification Requirements 17
2.3.2 Types of Specifications 18
2.4 The System Design Process 19

3 System-Level Design Issues: Rules and Tools 23

3.1 The Standard Model 23
3.1.1 Soft IP vs. Hard IP 25
3.1.2 The Role of Full-Custom Design in Reuse 27
3.2 Design for Timing Closure: Logic Design Issues 28
3.2.1 Interfaces and Timing Closure 28
3.2.2 Synchronous vs. Asynchronous Design Style 33
3.2.3 Clocking 35
3.2.4 Reset 36
3.2.5 Timing Exceptions and Multicycle Paths 37
3.3 Design for Timing Closure: Physical Design Issues 38
3.3.1 Floorplanning 38
3.3.2 Synthesis Strategy and Timing Budgets 39
3.3.3 Hard Macros 39
3.3.4 Clock Distribution 40
3.4 Design for Verification: Verification Strategy 40
3.5 System Interconnect and On-Chip Buses 42
3.5.1 Basic Interface Issues 43
3.5.2 Tristate vs. Mux Buses 47
3.5.3 Synchronous Design of Buses 47
3.5.4 Summary 47
3.5.5 IP-to-IP Interfaces 48
3.6 Design for Bring-Up and Debug: On-Chip Debug Structures 51
3.7 Design for Low Power 52
3.7.1 Lowering the Supply Voltage 53
3.7.2 Reducing Capacitance and Switching Activity 54
3.7.3 Sizing and Other Synthesis Techniques 56
3.7.4 Summary 57
3.8 Design for Test: Manufacturing Test Strategies 57
3.8.1 System-Level Test Issues 57
3.8.2 Memory Test 58
3.8.3 Microprocessor Test 58
3.8.4 Other Macros 59
3.8.5 Logic BIST 59

3.9 Prerequisites for Reuse 60
3.9.1 Libraries 60
3.9.2 Physical Design Rules 61

4 The Macro Design Process 63

4.1 Overview of IP Design 63
4.1.1 Characteristics of Good IP 64
4.1.2 Implementation and Verification IP 65
4.1.3 Overview of Design Process 67
4.2 Key Features 68
4.3 Planning and Specification 69
4.3.1 Functional Specification 69
4.3.2 Verification Specification 71
4.3.3 Packaging Specification 71
4.3.4 Development Plan 71
4.3.5 High-Level Models as Executable Specifications 72
4.4 Macro Design and Verification 73
4.4.1 Summary 77
4.5 Soft Macro Productization 78
4.5.1 Productization Process 78
4.5.2 Activities and Tools 78

5 RTL Coding Guidelines 81

5.1 Overview of the Coding Guidelines 81
5.2 Basic Coding Practices 82
5.2.1 General Naming Conventions 82
5.2.2 Naming Conventions for VITAL Support 84
5.2.3 State Variable Names 85
5.2.4 Include Informational Headers in Source Files 85
5.2.5 Use Comments 87
5.2.6 Keep Commands on Separate Lines 87
5.2.7 Line Length 87
5.2.8 Indentation 88
5.2.9 Do Not Use HDL Reserved Words 89
5.2.10 Port Ordering 89
5.2.11 Port Maps and Generic Maps 92
5.2.12 VHDL Entity, Architecture, and Configuration Sections 93
5.2.13 Use Functions 93
5.2.14 Use Loops and Arrays 94
5.2.15 Use Meaningful Labels 96

5.3 Coding for Portability 97
5.3.1 Use Only IEEE Standard Types (VHDL) 97
5.3.2 Do Not Use Hard-Coded Numeric Values 98
5.3.3 Packages (VHDL) 98
5.3.4 Constant Definition Files (Verilog) 98
5.3.5 Avoid Embedding Synthesis Commands 99
5.3.6 Use Technology-Independent Libraries 99
5.3.7 Coding For Translation 100
5.4 Guidelines for Clocks and Resets 101
5.4.1 Avoid Mixed Clock Edges 102
5.4.2 Avoid Clock Buffers 103
5.4.3 Avoid Gated Clocks 103
5.4.4 Avoid Internally Generated Clocks 104
5.4.5 Gated Clocks and Low-Power Designs 105
5.4.6 Avoid Internally Generated Resets 106
5.4.7 Reset Logic Function 107
5.4.8 Single-Bit Synchronizers 108
5.4.9 Multiple-Bit Synchronizers 108
5.5 Coding for Synthesis 108
5.5.1 Infer Registers 109
5.5.2 Avoid Latches 110
5.5.3 If you must use a latch 113
5.5.4 Avoid Combinational Feedback 113
5.5.5 Specify Complete Sensitivity Lists 114
5.5.6 Blocking and Nonblocking Assignments (Verilog) 117
5.5.7 Signal vs. Variable Assignments (VHDL) 119
5.5.8 Case Statements vs. if-then-else Statements 120
5.5.9 Coding Sequential Logic 122
5.5.10 Coding Critical Signals 124
5.5.11 Avoid Delay Times 124
5.5.12 Avoid full_case and parallel_case Pragmas 124
5.6 Partitioning for Synthesis 125
5.6.1 Register All Outputs 125
5.6.2 Locate Related Combinational Logic in a Single Module 126
5.6.3 Separate Modules That Have Different Design Goals 127
5.6.4 Asynchronous Logic 128
5.6.5 Arithmetic Operators: Merging Resources 128
5.6.6 Partitioning for Synthesis Runtime 130
5.6.7 Avoid Timing Exceptions 130
5.6.8 Eliminate Glue Logic at the Top Level 133
5.6.9 Chip-Level Partitioning 134

5.7 Designing with Memories 135
5.8 Code Profiling 136

6 Macro Synthesis Guidelines 137

6.1 Overview of the Synthesis Problem 137
6.2 Macro Synthesis Strategy 138
6.2.1 Macro Timing Constraints 139
6.2.2 Subblock Timing Constraints 139
6.2.3 Synthesis in the Design Process 140
6.2.4 Subblock Synthesis Process 141
6.2.5 Macro Synthesis Process 141
6.2.6 Wire Load Models 142
6.2.7 Preserve Clock and Reset Networks 142
6.2.8 Code Checking Before Synthesis 143
6.2.9 Code Checking After Synthesis 143
6.3 Physical Synthesis 144
6.3.1 Classical Synthesis 144
6.3.2 Physical Synthesis 145
6.3.3 Physical Synthesis Deliverables 145
6.4 RAM and Datapath Generators 145
6.4.1 Memory Design 146
6.4.2 RAM Generator Flow 147
6.4.3 Datapath Design 148
6.5 Coding Guidelines for Synthesis Scripts 150

7 Macro Verification Guidelines 153

7.1 Overview of Macro Verification 153
7.1.1 Verification Plan 154
7.1.2 Verification Strategy 155
7.2 Inspection as Verification 159
7.3 Adversarial Testing 160
7.4 Testbench Design 161
7.4.1 Transaction-Based Verification 161
7.4.2 Component-Based Verification 163
7.4.3 Automated Response Checking 165
7.4.4 Verification Suite Design 166
7.5 Design of Verification Components 169
7.5.1 Bus Functional Models 169
7.5.2 Monitors 171

7.5.3 Device Models 171
7.5.4 Verification Component Usage 172
7.6 Getting to 100% 172
7.6.1 Functional and Code Coverage 172
7.6.2 Prototyping 172
7.6.3 Limited Production 173
7.6.4 Property Checking 173
7.6.5 Code Coverage Analysis 174
7.7 Timing Verification 177

8 Developing Hard Macros 179

8.1 Overview 179
8.1.1 Why and When to Use Hard Macros 180
8.1.2 Design Process for Hard vs. Soft Macros 181
8.2 Design Issues for Hard Macros 181
8.2.1 Full-Custom Design 181
8.2.2 Interface Design 182
8.2.3 Design For Test 183
8.2.4 Clock 184
8.2.5 Aspect Ratio 185
8.2.6 Porosity 186
8.2.7 Pin Placement and Layout 187
8.2.8 Power Distribution 187
8.2.9 Antenna Checking 188
8.3 The Hard Macro Design Process 190
8.4 Productization of Hard Macros 190
8.4.1 Physical Design 190
8.4.2 Verification 193
8.5 Model Development for Hard Macros 194
8.5.1 Functional Models 194
8.5.2 Timing Models 199
8.5.3 Power Models 200
8.5.4 Test Models 201
8.5.5 Physical Models 204
8.6 Porting Hard Macros 204

9 Macro Deployment: Packaging for Reuse 207

9.1 Delivering the Complete Product 207
9.1.1 Soft Macro Deliverables 208

9.1.2 Hard Macro Deliverables 210
9.1.3 Software 212
9.1.4 The Design Archive 213
9.2 Contents of the User Guide 214

10 System Integration with Reusable Macros 217

10.1 Integration Overview 217
10.2 Integrating Macros into an SoC Design 218
10.2.1 Problems in Integrating IP 218
10.2.2 Strategies for Managing Interfacing Issues 219
10.2.3 Interfacing Hard Macros to the Rest of the Design 220
10.3 Selecting IP 221
10.3.1 Hard Macro Selection 221
10.3.2 Soft Macro Selection 221
10.3.3 Soft Macro Installation 222
10.3.4 Soft Macro Configuration 223
10.3.5 Synthesis of Soft Macros 223
10.4 Integrating Memories 223
10.5 Physical Design 224
10.5.1 Design Planning and Synthesis 226
10.5.2 Physical Placement 230
10.5.3 Timing Closure 234
10.5.4 Verifying the Physical Design 237
10.5.5 Summary 238

11 System-Level Verification Issues 239

11.1 The Importance of Verification 239
11.2 The Verification Strategy 240
11.3 Interface Verification 241
11.3.1 Transaction Verification 241
11.3.2 Data or Behavioral Verification 242
11.3.3 Standardized Interfaces 244
11.4 Functional Verification 244
11.5 Random Testing 247
11.6 Application-Based Verification 249
11.6.1 Software-Driven Application Testbench 250
11.6.2 Rapid Prototyping for Testing 251

11.7 Gate-Level Verification 253
11.7.1 Sign-Off Simulation 253
11.7.2 Formal Verification 254
11.7.3 Gate-Level Simulation with Full Timing 255
11.8 Specialized Hardware for System Verification 256
11.8.1 Accelerated Verification Overview 258
11.8.2 RTL Acceleration 259
11.8.3 Software Driven Verification 260
11.8.4 Traditional In-Circuit Verification 260
11.8.5 Design Guidelines for Emulation 261
11.8.6 Testbenches for Emulation 261

12 Data and Project Management 265

12.1 Data Management 265
12.1.1 Revision Control Systems 265
12.1.2 Bug Tracking 267
12.1.3 Regression Testing 267
12.1.4 Managing Multiple Sites 267
12.1.5 Archiving 268
12.2 Project Management 269
12.2.1 Development Process 269
12.2.2 Functional Specification 269
12.2.3 Project Plan 270

13 Implementing Reuse-Based SoC Designs 271

13.1 Alcatel 272
13.2 Atmel 274
13.3 Infineon Technologies 276
13.4 LSI Logic 278
13.5 Philips Semiconductor 280
13.6 STMicroelectronics 282
13.7 Conclusion 284

Bibliography 285

Index 287