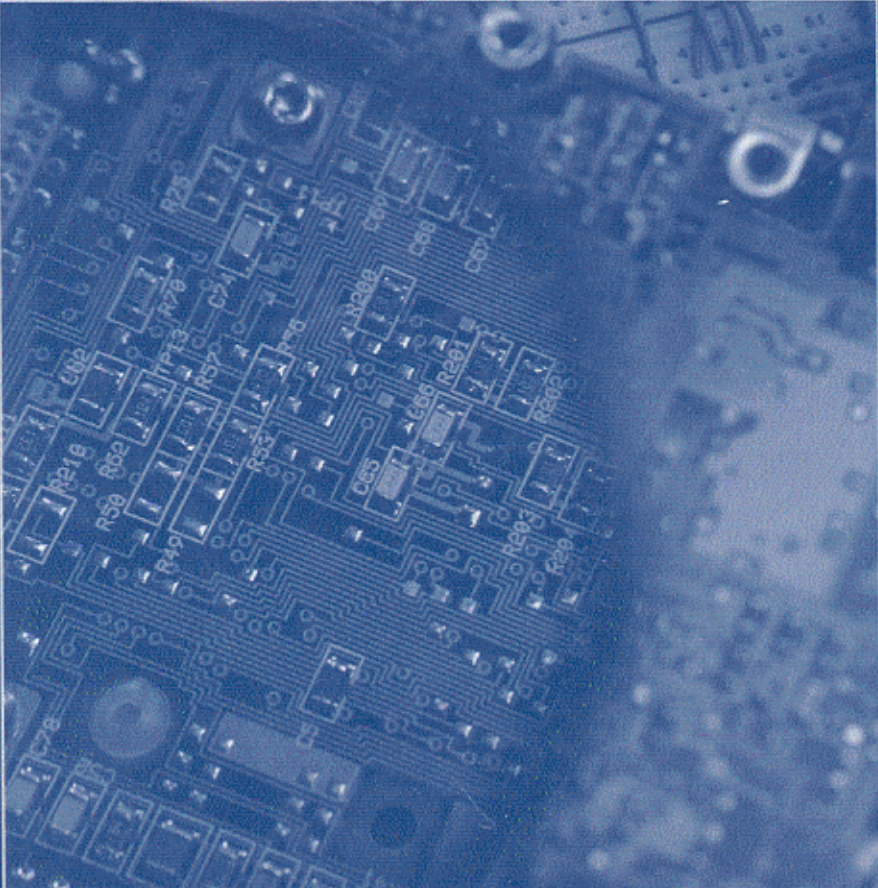


Hardware Design Verification

Simulation and Formal Method-Based Approaches



William K. Lam

Prentice Hall Modern Semiconductor Design Series

Contents

Preface	xvii
Acknowledgments	xxiii
About the Author	xxv
Chapter 1 An Invitation to Design Verification	1
1.1 What Is Design Verification?	2
1.2 The Basic Verification Principle	4
1.3 Verification Methodology	8
1.3.1 Simulation-Based Verification	8
1.3.2 Formal Method-Based Verification	12
1.4 Simulation-Based Verification versus Formal Verification	15
1.5 Limitations of Formal Verification	17
1.6 A Quick Overview of Verilog Scheduling and Execution Semantics	17
1.6.1 Concurrent Processes	17
1.6.2 Nondeterminism	19
1.6.3 Scheduling Semantics	22
1.7 Summary	23
Chapter 2 Coding for Verification	25
2.1 Functional Correctness	27
2.1.1 Syntactical Checks	27
2.1.2 Structural Checks	36
2.2 Timing Correctness	39
2.2.1 Race Problem	39
2.2.2 Clock Gating	41
2.2.3 Time Zero and Zero Time Glitches	42
2.2.4 Domain-Crossing Glitches	43

2.3	Simulation Performance	44
2.3.1	Higher Level of Abstraction.....	44
2.3.2	Simulator Recognizable Components	46
2.3.3	Vector versus Scalar	48
2.3.4	Minimization of the Interface to Other Simulation Systems	51
2.3.5	Low-Level/Component-Level Optimization.....	51
2.3.6	Code Profiling	52
2.4	Portability and Maintainability.....	52
2.4.1	Project Code Layout	52
2.4.2	Centralized Resource	54
2.4.3	RTL Design File Format	55
2.5	"Synthesizability," "Debugability," and General Tool Compatibility.....	56
2.5.1	Synthesizability.....	57
2.5.2	Debugability	57
2.6	Cycle-Based Simulation	59
2.7	Hardware Simulation/Emulation	62
2.8	Two-State and Four-State Simulation.....	64
2.9	Design and Use of a Linter	66
2.10	Summary	67
2.11	Problems	67
Chapter 3	Simulator Architectures and Operations	73
3.1	The Compilers	74
3.2	The Simulators.....	79
3.2.1	Event-Driven Simulators	80
3.2.2	Cycle-Based Simulators	88
3.2.3	Hybrid Simulators.....	100
3.2.4	Hardware Simulators and Emulators.....	106
3.3	Simulator Taxonomy and Comparison.....	108
3.3.1	Two-State and Four-State Simulators.....	108
3.3.2	Zero- versus Unit-Delay Simulators.....	109
3.3.3	Event-Driven versus Cycle-Based Simulators	110
3.3.4	Interpreted versus Compiled Simulators.....	111
3.3.5	Hardware Simulators	111
3.4	Simulator Operations and Applications	112
3.4.1	The Basic Simulation File Structure.....	113
3.4.2	Performance and Debugging.....	114
3.4.3	Timing Verification	116
3.4.4	Design Profiling	122
3.4.5	Two State and Four State	123

3.4.6	Cosimulation with Encapsulated Models	125
3.5	Incremental Compilation	126
3.6	Summary	129
3.7	Problems	130
Chapter 4	Test Bench Organization and Design	137
4.1	Anatomy of a Test Bench and a Test Environment	137
4.2	Initialization Mechanism	142
4.2.1	RTL Initialization	143
4.2.2	PLI Initialization	145
4.2.3	Initialization at Time Zero	147
4.3	Clock Generation and Synchronization	148
4.3.1	Explicit and Toggle Methods	148
4.3.2	Absolute Transition Delay	149
4.3.3	Time Zero Clock Transition	150
4.3.4	Time Unit and Resolution	150
4.3.5	Clock Multiplier and Divider	151
4.3.6	Clock Independence and Jitter	152
4.3.7	Clock Synchronization and Delta Delay	153
4.3.8	Clock Generator Organization	154
4.4	Stimulus Generation	155
4.4.1	Asynchronous Stimuli Application	159
4.4.2	Instruction Code or Programmed Stimuli	161
4.5	Response Assessment	162
4.5.1	Design State Dumping	162
4.5.2	Golden Response	167
4.5.3	Checking Temporal Specifications	178
4.6	Verification Utility	183
4.6.1	Error Injector	184
4.6.2	Error Warning and Alert Mechanism	185
4.6.3	Memory Loading and Dumping Mechanism	186
4.6.4	Sparse Memory and Content Addressable Memory (CAM)	190
4.6.5	Assertion Routines	194
4.7	Test Bench-to-Design Interface	195
4.8	Common Practical Techniques and Methodologies	196
4.8.1	Verification Environment Configuration	196
4.8.2	Bus Functional Model	199
4.9	Summary	204
4.10	Problems	204

Chapter 5	Test Scenarios, Assertions, and Coverage	211
5.1	Hierarchical Verification	214
5.1.1	System Level	215
5.1.2	Unit Level	216
5.1.3	Module Level	216
5.2	Test Plan	217
5.2.1	Extract Functionality from Architectural Specifications	217
5.2.2	Prioritize Functionality	222
5.2.3	Create Test Cases	223
5.2.4	Track Progress	223
5.3	Pseudorandom Test Generator	227
5.3.1	User Interface	228
5.3.2	Register and Memory Allocation	229
5.3.3	Program Construction	230
5.3.4	Self-checking Mechanism	230
5.4	Assertions	232
5.4.1	Defining What to Assert	232
5.4.2	Assertion Components	233
5.4.3	Writing Assertions	235
5.4.4	Built-in Commercial Assertions	247
5.5	SystemVerilog Assertions	248
5.5.1	Immediate Assertions	248
5.5.2	Concurrent Assertions	249
5.6	Verification Coverage	259
5.6.1	Code Coverage	261
5.6.2	Parameter Coverage	270
5.6.3	Functional Coverage	273
5.6.4	Item Coverage and Cross-Coverage	278
5.7	Summary	279
5.8	Problems	280
Chapter 6	Debugging Process and Verification Cycle	287
6.1	Failure Capture, Scope Reduction, and Bug Tracking	288
6.1.1	Failure Capture	288
6.1.2	Scope Reduction	289
6.1.3	Error Tracking System	295
6.2	Simulation Data Dumping	297
6.2.1	Spatial Neighborhood	297
6.2.2	Temporal Window	298

6.3	Isolation of Underlying Causes	300
6.3.1	Reference Value, Propagation, and Bifurcation	300
6.3.2	Forward and Backward Debugging	301
6.3.3	Tracing Diagram	302
6.3.4	Time Framing	304
6.3.5	Load, Driver, and Cone Tracing	306
6.3.6	Memory and Array Tracing	308
6.3.7	Zero Time Loop Constructs	310
6.3.8	The Four Basic Views of Design	311
6.3.9	Typical Debugger Functionality	312
6.4	Design Update and Maintenance: Revision Control	315
6.5	Regression, Release Mechanism, and Tape-out Criteria	318
6.6	Summary	321
6.7	Problems.	322
Chapter 7	Formal Verification Preliminaries.	331
7.1	Sets and Operations	332
7.2	Relation, Partition, Partially Ordered Set, and Lattice	334
7.3	Boolean Functions and Representations	342
7.3.1	Symmetric Boolean Functions	346
7.3.2	Incompletely Specified Boolean Functions	349
7.3.3	Characteristic Functions.	351
7.4	Boolean Functional Operators	353
7.5	Finite-State Automata and Languages	359
7.5.1	Product Automata and Machines	363
7.5.2	State Equivalence and Machine Minimization	365
7.5.3	Finite-State Machine Equivalence	370
7.5.4	Graph Algorithms	371
7.5.5	Depth-First Search	373
7.5.6	Breadth-First Search	378
7.6	Summary	380
7.7	Problems.	381
Chapter 8	Decision Diagrams, Equivalence Checking, and Symbolic Simulation	387
8.1	Binary Decision Diagrams	388
8.1.1	Operations on BDDs	395
8.1.2	Variable Ordering	404

8.2	Decision Diagram Variants	412
8.2.1	Shared BDDs (SBDDs)	412
8.2.2	Edge-Attributed BDDs	412
8.2.3	Zero-Suppressed BDDs (ZBDDs)	414
8.2.4	Ordered Functional Decision Diagrams (OFDDs)	417
8.2.5	Pseudo Boolean Functions and Decision Diagrams	420
8.2.6	Binary Moment Diagram (BMD).....	423
8.3	Decision Diagram-Based Equivalence Checking.....	424
8.3.1	Node Mapping and Constraining.....	425
8.4	Boolean Satisfiability.....	430
8.4.1	Resolvent Algorithm	431
8.4.2	Search-Based Algorithm.....	434
8.4.3	Implication Graph and Learning	438
8.5	Symbolic Simulation	442
8.5.1	Symbolic Verification	446
8.5.2	Input Constraining	447
8.5.3	Symbolic Simulation Using Characteristic Functions.....	450
8.5.4	Parameterization	453
8.6	Summary	457
8.7	Problems	458
Chapter 9	Model Checking and Symbolic Computation.....	465
9.1	Properties, Specifications, and Logic	466
9.1.1	Sequential Specification Using Automata	467
9.1.2	Temporal Structure and Computation Tree.....	468
9.1.3	Propositional Temporal Logic: LTL, CTL*, and CTL	475
9.1.4	Fairness Constraint	482
9.1.5	Relative Expressiveness of CTL*, CTL, and LTL.....	483
9.1.6	SystemVerilog Assertion Language	484
9.2	Property Checking	484
9.2.1	Property Specification Using an Automaton.....	485
9.2.2	Language Containment	487
9.2.3	Checking CTL Formulas	489
9.2.4	Checking with Fairness Constraints.....	493
9.3	Symbolic Computation and Model Checking	494
9.3.1	Symbolic Finite-State Machine Representation and State Traversal	495
9.3.2	Counterexample Generation	501
9.3.3	Equivalence Checking	502
9.3.4	Language Containment and Fairness Constraints	507

9.4	Symbolic CTL Model Checking	513
9.4.1	Fix-point Computation	513
9.4.2	CTL Checking with Fairness Constraints	520
9.5	Computational Improvements	524
9.5.1	Early Quantification	525
9.5.2	Generalized Cofactor	527
9.6	Using Model-Checking Tools	529
9.7	Summary	531
9.8	Problems	531
	Bibliography	539
	Index	561