Contents

Preface xi

Contributing Authors xiii

Introduction xix
Rolf Drechsler
1 Formal Verification xix
2 Challenges xxi
3 Contributions to this Book xxiii

1
What Sat-Solvers can and cannot do 1
Eugene Goldberg
1 Introduction 1
2 Hard Equivalence Checking CNF formulas 3
2.1 Introduction 3
2.2 Common Specification of Boolean Circuits 5
2.3 Equivalence Checking as SAT 11
2.4 Class $M(p)$ and general resolution 12
2.5 Computation of existentially implied functions 13
2.6 Equivalence Checking in General Resolution 14
2.7 Equivalence Checking of Circuits with Unknown CS 20
2.8 A Procedure of Equivalence Checking for Circuits with a Known CS 22
2.9 Experimental Results 23
2.10 Conclusions 26
3 Stable Sets of Points 26
3.1 Introduction 26
3.2 Stable Set of Points 28
3.3 SSP as a reachable set of points 31
3.4 Testing Satisfiability of CNF Formulas by SSP Construction 32
3.5 Testing Satisfiability of Symmetric CNF Formulas by SSP Construction 35
3.6 SSPs with Excluded Directions 39
3.7 Conclusions 42

2
Advancements in mixed BDD and SAT techniques 45
Gianpiero Cabodi and *Stefano Quer*
1 Introduction 45
2 Background 47
2.1 SAT Solvers 47
2.2 Binary Decision Diagrams 48
2.2.1 Zero-Suppressed Binary Decision Diagrams 49
2.2.2 Boolean Expression Diagrams 50
2.3 Model Checking and Equivalence Checking 52
3 Comparing SAT and BDD Approaches: Are they different? 54
3.1 Theoretical Considerations 54
3.2 Experimental Benchmarking 55
3.2.1 Bug Hunting in an Industrial Setting 56
3.2.2 Modifying BDD-based Techniques to Perform BMC 56
3.2.3 Conclusions 58
3.3 Working on Affinities: Variable Order 58
3.3.1 Affinities on circuit-width correlation 59
3.3.2 Recursion tree and Variable Order 59
3.3.3 A Common Static Variable Order Heuristic 60
3.3.4 Conclusions 60
4 Decision Diagrams as a Slave Engine in general SAT: Clause Compression by Means of ZBDDs 61
4.1 ZBDDs for Symbolic Davis-Putnam Resolution 61
4.2 ZBDDs for Symbolic DLL 62
4.3 ZBDDs for Breadth-First SAT 62
4.4 Conclusions 62
5 Decision Diagram Preprocessing and Circuit-Based SAT 62
5.1 BED Preprocessing 63
5.2 Circuit-Based SAT 64
5.2.1 BDD Sweeping and SAT 64
5.2.2 SAT on BEDs 66
5.3 Preprocessing by Approximate Reachability 67
6 Using SAT in Symbolic Reachability Analysis 68
6.0.1 BDDs at SAT leaves 69
6.0.2 SAT-Based Symbolic Image and Pre-image 70
7 Conclusions, Remarks and Future Works 71

3
Equivalence Checking of Arithmetic Circuits 77
Dominik Stoffel, *Evgeny Karibaev*, *Irina Kufareva* and *Wolfgang Kunz*
1 Introduction 78
2 Verification Using Functional Properties 81
3 Bit-Level Decision Diagrams 85
4 Word-Level Decision Diagrams 88
4.1 Pseudo-Boolean functions and decompositions 89
4.2 *BMDs 92
4.3 Equivalence Checking Using *BMDs 94
4.4 Experiments with *BMD synthesis 97
5 Arithmetic Bit-Level Verification 105
5.1 Verification at the Arithmetic Bit Level 108
5.2 Extracting the Half Adder Network 112

5.3 Verification Framework 115
5.4 Experimental Results 115
6 Conclusion 118
7 Future Perspectives 119

4
Application of Property Checking and Underlying Techniques 125
Raik Brinkmann, *Peer Johannsen* and *Klaus Winkelmann*
1 Circuit Verification Environment: User's View 126
1.1 Tool Environment 126
1.2 The gateprop Property Checker 127
2 Circuit Verification Environment: Underlying Techniques 129
2.1 From Property to Satisfiability 129
2.2 Preserving Structure during Problem Construction 131
2.3 The Experimental Platform RtProp 132
3 Exploiting Symmetries 133
3.1 Symmetry in Property Checking Problems 133
3.2 Finding Symmetrical Value Vectors 136
3.3 Practical Results 140
4 Automated Data Path Scaling to Speed Up Property Checking 142
4.1 Bitvector Satisfiability Problems 143
4.2 Formal Abstraction Techniques 145
4.3 Speeding Up Hardware Verification by One-To-One Abstraction 146
4.4 Data Path Scaling of Circuit Designs 147
5 Property Checking Use Cases 152
5.1 Application Example: Reverse Engineering 155
5.1.1 Functionality 155
5.1.2 Task 155
5.1.3 Examples for a property 156
5.1.4 Results 157
5.2 Application Example: Complete Block-Level ASIC Verification 158
5.2.1 Verification Challenge and Approach 158
5.2.2 Verifying the Control Path 159
5.2.3 Data Path Results 160
5.2.4 Overall Result 160
5.3 Productivity Statistics 161
6 Summary 162
6.1 Achievements 162
6.2 Challenges and Perspectives 163

5
Assertion-Based Verification 167
Claudionor Nunes Coelho Jr. and *Harry D. Foster*
1 Introduction 167
1.1 Specifying properties 169
1.2 Observability and controllability 171
1.3 Formal property checking framework 172
2 Assertion Specification 177
2.1 Temporal logic 177
2.2 Property Specification Language (PSL) 179

2.2.1 Boolean layer 180
2.2.2 Temporal layer 180
2.2.3 Verification layer 182
3 Assertion libraries 183
4 Assertion simulation 184
5 Assertions and formal verification 186
5.1 Handling complexity 186
5.2 Formal property checking role 190
6 Assertions and synthesis 191
6.1 On-line validation 191
6.2 Synthesizable assertions 192
6.3 Routing scheme for assertion libraries 194
6.4 Assertion processors 195
6.5 Impact of Assertions in Real Circuits 197
7 PCI property specification example 197
7.1 PCI overview 198
7.2 PCI master reset requirement 199
7.3 PCI burst order encoding requirement 199
7.4 PCI basic read transaction 200
8 Summary 202

6
Formal Verification for Nonlinear Analog Systems 205
Walter Hartong, *Ralf Klausen* and *Lars Hedrich*
1 Introduction 206
2 System Description 206
2.1 Analog Circuit Classes 208
2.2 State Space Description 208
2.2.1 Index 209
2.2.2 Solving a DAE System 209
2.2.3 Linearized System Description 211
3 Equivalence Checking 211
3.1 Basic Concepts 212
3.1.1 Nonlinear Mapping of State Space Descriptions 212
3.2 Equivalence Checking Algorithm 213
3.2.1 Sampling the State Space 213
3.2.2 Consistent Sample Point 215
3.3 Linear Transformation Theory 217
3.3.1 System Transformation to a Kronecker Canonical Form 217
3.3.2 DAE System Transformation into the Virtual State Space 219
3.3.3 Error Calculation 222
3.4 Experimental Results 222
3.4.1 Schmitt Trigger Example 222
3.4.2 Bandpass Example 225
4 Model Checking 227
4.1 Model Checking Language 227
4.2 Analog Model Checking Algorithm 230
4.2.1 Transition Systems 230
4.2.2 Discrete Time Steps 231
4.2.3 State Space Subdivision 232
4.2.4 Transition Relation 234

4.2.5 Border Problems 236
4.2.6 Input Value Model 237
4.2.7 Optimizations 239
4.3 Experimental Results 239
4.3.1 Schmitt Trigger Example 239
4.3.2 Tunnel Diode Oscillator Example 240
5 Summary 242
6 Acknowledgement 242
Appendix: Mathematical Symbols 243
Index 247