

Contents

1	Introduction	1
1.1	Setting the Context	1
1.1.1	Faults and the Design Cycle	2
1.1.2	Avoiding Design Errors in Hardware Designs	3
1.2	Circuit Design	4
1.3	Fighting Design Errors	8
1.4	Verification versus Validation	10
1.5	Hardware Verification	10
1.5.1	Verification versus Simulation	11
1.5.2	Formal Specifications	12
1.5.3	Formal Implementation Description	15
1.5.4	Correctness Relation and Proof	18
1.6	The Success of Formal Hardware Verification	21
1.6.1	Prerequisites for the Success of Hardware Verification	22
1.6.2	Properties of Successful Hardware Verification Approaches	23
1.7	Limitations of Formal Hardware Verification	25
1.8	The Pragmatic Approach – Recipes for Verifying Circuits	26
1.9	Summary	26
1.10	Structure of the Book	28
1.11	Literature	28
2	Boolean Functions	31
2.1	Motivation	31
2.1.1	Hardware Verification Tasks	31
2.2	Representations for Boolean Functions	32
2.2.1	Function Tables	33
2.2.2	Propositional Logic	34
2.2.3	Binary Decision Diagrams	37

2.3	Modeling Hardware Behavior	48
2.3.1	Functional Circuit Representation	48
2.3.2	Relational Circuit Representation	49
2.3.3	Characteristic Functions	51
2.4	Specification, Proof Goals and Proof	52
2.4.1	Implicit Hardware Verification	52
2.4.2	Explicit Hardware Verification	53
2.4.3	Model-Based Proof Approaches	55
2.5	Further Developments and Tools	55
2.5.1	Extensions and Variants of Binary Decision Diagrams	55
2.5.2	Variable Ordering Heuristics	71
2.6	Technical Details	76
2.6.1	Classes of Boolean Functions	77
2.7	Summary	80
3	Approaches Based on Finite State Machines	83
3.1	Motivation	83
3.2	Formal Basics	84
3.2.1	Automata for Finite Sequences	85
3.2.2	Automata for Infinite Sequences	89
3.2.3	Image and Pre-Image of a Function	90
3.3	Modeling Hardware Behavior	93
3.4	Specification, Proof Goal and Proof	94
3.4.1	Symbolic State Machine Traversal	95
3.5	Further Developments	100
3.5.1	Structural Approaches to Circuit Equivalence	102
3.5.2	Relational FSM Representation	103
3.5.3	Functional FSM Representation	114
3.5.4	State Space Traversal Variants	119
3.5.5	ROBDD Variable Ordering for FSM Equivalence Checking	132
3.5.6	Verification of Sequential Circuits without Reset Lines	133
3.5.7	Verification based Test Generation for Fabrication Faults	147
3.6	Summary	148
4	Propositional Temporal Logics	151
4.1	Motivation	151
4.2	Formal Basics	153
4.2.1	Temporal Structures	153
4.2.2	The Propositional Temporal Logics CTL*, CTL and LTL	156
4.2.3	Comparing CTL, LTL and CTL*	164
4.2.4	Proof Algorithms	167
4.2.5	Comparing Proof Complexity	180

4.3	Modeling Hardware Behavior	183
4.3.1	Describing Implementations with Temporal Structures	184
4.3.2	Describing Implementations by Temporal Logic Formulas	188
4.4	Specification, Proof Goal and Proof	188
4.4.1	Creating Temporal Logic Specifications.	189
4.5	Further Developments	192
4.5.1	Increasing Efficiency.	192
4.5.2	Specifications	192
4.6	Technical Details	194
4.6.1	Proof Algorithm	194
4.7	Summary	204
5	Higher-Order Logics	207
5.1	Motivation	207
5.2	Formal Basics	209
5.2.1	Formal Systems.	209
5.2.2	First Order Logic.	210
5.2.3	Higher-Order Logic.	213
5.3	Modeling Hardware Behavior	225
5.3.1	Representing Modules by Predicates	225
5.3.2	Modeling Structures	227
5.4	Specification and Proof	227
5.5	Performing Proofs	228
5.5.1	Methodology for Establishing Circuit Correctness.	229
5.5.2	Abstraction Mechanisms.	235
5.5.3	Verification of Generic Circuits	242
5.6	Technical Details	245
5.6.1	Some More Theory	245
5.6.2	Modeling Hardware Behavior.	247
5.6.3	Formalizing Abstraction Mechanisms	249
5.7	Summary	253
Appendix A	Mathematical Basics	255
Appendix B	Axioms and Rules for CTL*	267
Appendix C	Axioms and Rules for Higher Order Logic	271
References.	277
Index.	291