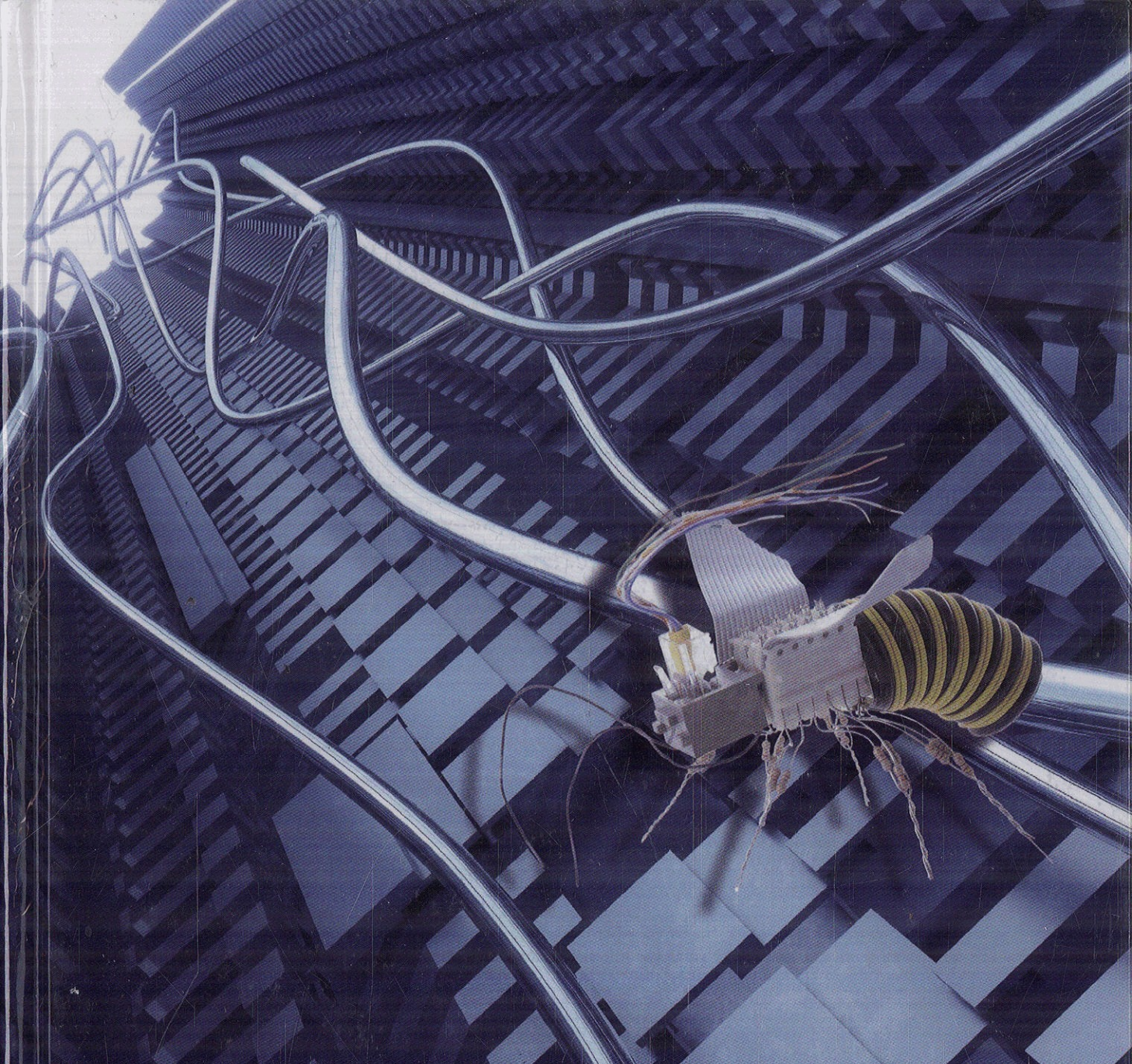




EMBEDDED SYSTEMS AND SOFTWARE VALIDATION

ABHIK ROYCHOUDHURY



MK[®]
MORGAN KAUFMANN

Contents

	Acknowledgments	ix
	Preface	xi
CHAPTER 1	Introduction	1
CHAPTER 2	Model Validation	7
2.1	Platform versus System Behavior	8
2.2	Criteria for Design Model	10
2.3	Informal Requirements: A Case Study	12
2.3.1	The Requirements Document	13
2.3.2	Simplification of the Informal Requirements	14
2.4	Common Modeling Notations	16
2.4.1	Finite-State Machines	16
2.4.2	Communicating FSMs	20
2.4.3	Message Sequence Chart-Based Models	27
2.5	Remarks about Modeling Notations	37
2.6	Model Simulations	39
2.6.1	FSM Simulations	41
2.6.2	Simulating MSC-Based System Models	46
2.7	Model-Based Testing	50
2.8	Model Checking	58
2.8.1	Property Specification	58
2.8.2	Checking Procedure	73
2.9	The SPIN Validation Tool	82
2.10	The SMV Validation Tool	86
2.11	Case Study: Air-Traffic Controller	89
2.12	References	91
2.13	Exercises	93
CHAPTER 3	Communication Validation	95
3.1	Common Incompatibilities	98
3.1.1	Sending/Receiving Signals in Different Order	99
3.1.2	Handling a Different Signal Alphabet	100
3.1.3	Mismatch in Data Format	102
3.1.4	Mismatch in Data Rates	105
3.2	Converter Synthesis	106
3.2.1	Representing Native Protocols and Converters	106
3.2.2	Basic Ideas for Converter Synthesis	108
3.2.3	Various Strategies for Protocol Conversion	115

	3.2.4 Avoiding No-Progress Cycles.....	116
	3.2.5 Speculative Transmission to Avoid Deadlocks.....	118
3.3	Changing a Working Design	121
3.4	References	122
3.5	Exercises	123
CHAPTER 4	Performance Validation	125
4.1	The Conventional Abstraction of Time.....	126
4.2	Predicting Execution Time of a Program.....	131
	4.2.1 WCET Calculation	133
	4.2.2 Modeling of Microarchitecture	145
4.3	Interference within a Processing Element.....	154
	4.3.1 Interrupts from Environment	155
	4.3.2 Contention and Preemption	157
	4.3.3 Sharing a Processor Cache	161
4.4	System-Level Communication Analysis	165
4.5	Designing Systems with Predictable Timing.....	169
	4.5.1 Scratchpad Memories	169
	4.5.2 Time-Triggered Communication	174
4.6	Emerging Applications	176
4.7	References	177
4.8	Exercises	177
CHAPTER 5	Functionality Validation	181
5.1	Dynamic or Trace-Based Checking	184
	5.1.1 Dynamic Slicing	187
	5.1.2 Fault Localization	196
	5.1.3 Directed Testing Methods	203
5.2	Formal Verification	207
	5.2.1 Predicate Abstraction.....	211
	5.2.2 Software Checking via Predicate Abstraction.....	218
	5.2.3 Combining Formal Verification with Testing	225
5.3	References	229
5.4	Exercises	230
	Bibliography	233
	Index	241