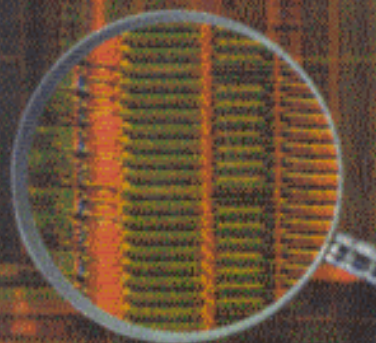




System-on-a-chip Verification

Methodology and Techniques

Prakash Rashinkar
Peter Paterson
Leena Singh

Kluwer Academic Publishers

Contents

<i>Authors</i>	xv
<i>Acknowledgements</i>	xvii
<i>Foreword</i>	xix

CHAPTER 1	<i>Introduction</i>	1
	1.1 Technology Challenges	2
	1.1.1 Timing Closure	2
	1.1.2 Capacity	3
	1.1.3 Physical Properties	3
	1.1.4 Design Productivity Gap	4
	1.1.5 Time-to-Market Trends	5
	1.1.6 SOC Technology	6
	1.2 Verification Technology Options	7
	1.2.1 Simulation Technologies	7
	1.2.1.1 Event-based Simulators	8
	1.2.1.2 Cycle-based Simulators	8
	1.2.1.3 Transaction-based Verification	9
	1.2.1.4 Code Coverage	9
	1.2.1.5 HW/SW Co-verification	9
	1.2.1.6 Emulation Systems	10
	1.2.1.7 Rapid Prototyping Systems	10
	1.2.1.8 Hardware Accelerators	11
	1.2.1.9 AMS Simulation	11
	1.2.2 Static Technologies	11
	1.2.2.1 Lint Checking	11
	1.2.2.2 Static Timing Verification	12
	1.2.3 Formal Technologies	12
	1.2.3.1 Theorem Proving Technique	12
	1.2.3.2 Formal Model Checking	12
	1.2.3.3 Formal Equivalence Checking	13
	1.2.4 Physical Verification and Analysis	13
	1.2.5 Comparing Verification Options	13
	1.2.5.1 Which Is the Fastest Option	14
	1.3 Verification Methodology	16
	1.3.1 System-Level Verification	16
	1.3.2 SOC Hardware RTL Verification	17

1.3.3 SOC Software Verification	18
1.3.4 Netlist Verification	18
1.3.5 Physical Verification	18
1.3.6 Device Test	18
1.4 Testbench Creation	19
1.4.1 Testbench in HDL	20
1.4.2 Testbench in PLI	20
1.4.3 Waveform-based	20
1.4.4 Transaction-based	20
1.4.5 Specification-based	21
1.5 Testbench Migration	21
1.5.1 Testbench Migration from Functional to RTL	22
1.5.2 Testbench Migration from RTL to Netlist	22
1.6 Verification Languages	23
1.7 Verification IP Reuse	23
1.8 Verification Approaches	24
1.8.1 Top-Down Design and Verification Approach	24
1.8.2 Bottom-Up Verification Approach	26
1.8.3 Platform-based Verification Approach	28
1.8.4 System Interface-driven Verification Approach ...	29
1.9 Verification and Device Test	29
1.9.1 Device Test Challenges	30
1.9.2 Test Strategies	30
1.10 Verification Plans	31
1.10.1 Project Functional Overview	31
1.10.2 Verification Approach	31
1.10.3 Abstraction Levels	32
1.10.4 Verification Technologies	32
1.10.5 Abstraction Level for Intent Verification	32
1.10.6 Test Application Approach	32
1.10.7 Results Checking	33
1.10.8 Functional Verification Flow	33
1.10.9 Test Definitions	33
1.10.10 Testbench Requirements	33
1.10.11 Models	33
1.10.12 Testbench Elements	34
1.10.13 Verification Metrics	34
1.10.14 Regression Testing	35
1.10.15 Issue Tracking and Management	35
1.10.16 Resource Plan	35
1.10.17 Project Schedule	35

1.11 Bluetooth SOC: A Reference Design	35
1.11.1 Bluetooth Device Elements	37
1.11.2 Bluetooth Network	37
1.11.3 Bluetooth SOC	39
1.11.3.1 Design Blocks	39
1.11.3.2 SOC Operation	41
 CHAPTER 2 <i>System-Level Verification</i>	45
2.1 System Design	46
2.1.1 Functional/Behavioral Design	47
2.1.2 Architecture Mapping	47
2.2 System Verification	48
2.2.1 Functional Verification	49
2.2.2 Performance Verification	50
2.2.3 System-Level Testbench	50
2.2.4 Creating a System-Level Testbench	50
2.2.5 System Testbench Metrics	51
2.2.6 Applying the System-Level Testbench	52
2.2.6.1 Emulation	53
2.2.6.2 Hardware Acceleration	54
2.2.6.3 Hardware Modeling	54
2.2.6.4 Mixed-Level Simulation	55
2.2.6.5 Design Partitioning	56
2.2.7 System Testbench Migration	56
2.2.7.1 Migrating a Testbench to Different Abstraction Levels	57
2.2.7.2 Migrating a System Testbench to Different Environments	58
2.3 Bluetooth SOC	58
 CHAPTER 3 <i>Block-Level Verification</i>	67
3.1 IP Blocks	68
3.2 Block Level Verification	69
3.3 Block Details of the Bluetooth SOC	70
3.3.1 Arbiter	71
3.3.2 Arbiter Testbench	77
3.3.2.1 Verilog Testbench	77
3.3.2.2 PLI Testbench	78
3.3.2.3 Waveform-based Testbench	79
3.3.2.4 Testbenches with Timing	84
3.3.3 Decoder	85

3.3.4 ASB Master	86
3.3.5 ASB Slave	86
3.3.6 ASB/APB Bridge	86
3.4 Lint Checking	88
3.5 Formal Model Checking	89
3.5.1 When to Use Model Checking	90
3.5.2 Limitations of Model Checking	90
3.5.3 Model Checking Methodology	90
3.5.3.1 Model Checking Properties	91
3.5.3.2 Model Checking Constraints	93
3.5.4 Performing Model Checking	94
3.5.4.1 Constraint Definitions	96
3.5.4.2 State Variable Definitions	97
3.5.4.3 Model Checking the Arbiter	97
3.5.4.4 Model Checking the ASB/APB Bridge	99
3.5.4.5 Model Checking the Decoder	102
3.6 Functional Verification/Simulation	104
3.6.1 Black-Box Verification Approach	105
3.6.2 White-Box Verification Approach	105
3.6.3 Gray-Box Verification Approach	106
3.6.4 Simulation	106
3.7 Protocol Checking	107
3.7.1 Memory/Register Access Signals	108
3.7.2 Protocol Checking Examples	108
3.7.2.1 Processor Routines	109
3.7.2.2 ASB Master	113
3.7.2.3 Decoder	113
3.8 Directed Random Testing	115
3.8.1 Random Vectors Generation in Verilog and C ...	116
3.9 Code Coverage Analysis	118
3.9.1 Types of Coverage	118
3.9.1.1 Statement Coverage	118
3.9.1.2 Toggle Coverage	119
3.9.1.3 State Machine Coverage	119
3.9.1.4 Visited State Coverage	119
3.9.1.5 Triggering Coverage	120
3.9.1.6 Branch Coverage	120
3.9.1.7 Expression Coverage	120
3.9.1.8 Path Coverage	121
3.9.1.9 Signal Coverage	121
3.9.2 Performing Code Coverage Analysis	121

CHAPTER 4	<i>Analog/Mixed Signal Simulation</i>	129
4.1	Mixed-Signal Simulation	130
4.2	Design Abstraction Levels	131
4.3	Simulation Environment	133
4.3.1	Selecting a Simulation Environment	134
4.3.2	Limitations of the Current Environments	135
4.4	Using SPICE	135
4.5	Simulation Methodology	136
4.6	Bluetooth SOC Digital-to-Analog Converter	138
4.6.1	Testbench for the DAC	143
4.6.2	Creating the Netlist	146
4.6.3	Simulation	149
4.6.4	Response	149
4.7	Chip-Level Verification with an AMS Block	150
 CHAPTER 5	 <i>Simulation</i>	 153
5.1	Functional Simulation	154
5.2	Testbench Wrappers	156
5.2.1	Block Details of the Bluetooth SOC	159
5.2.2	Test Vector Translation	159
5.2.3	Stimulus Generation	160
5.2.3.1	write_burst Routine	161
5.2.3.2	read_burst Routine	163
5.2.4	Stimulus Capture	165
5.2.5	Results Checking	166
5.2.6	Testbench Wrapper for a Slave	167
5.3	Event-based Simulation	176
5.3.1	Types of EBS Tools	177
5.3.2	EBS Environment	177
5.3.3	Selecting an EBS Solution	178
5.3.4	EBS Methodology	179
5.4	Cycle-based Simulation	180
5.4.1	When to Use CBS	180
5.4.2	CBS Environment	181
5.4.3	Selecting a CBS Solution	182
5.4.4	Limitations of CBS	182
5.4.5	CBS Methodology	183
5.4.6	Comparing EBS and CBS	184
5.5	Simulating the ASB/APB Bridge	184
5.5.1	ASB/APB Block	184
5.5.2	Design RTL Code	185

5.5.3 Testbench for EBS	193
5.5.4 Running a Simulation	194
5.6 Mixed-Event/Cycle-based Simulation	196
5.7 Transaction-based Verification	197
5.7.1 Elements of TBV	197
5.7.2 TBV Environment	199
5.7.3 Creating a Testbench	199
5.7.3.1 Creating Transactions in Verilog	201
5.7.3.2 Creating Transactions in C++	203
5.7.4 Transaction Analysis	205
5.7.5 Function Coverage in TBV	206
5.7.6 TBV Methodology	207
5.7.7 Bluetooth SOC	208
5.7.7.1 Creating a TVM for ARM7TDMI	209
5.7.7.2 Creating a Test	220
5.7.7.3 Compilation and Simulation	222
5.8 Simulation Acceleration	223
5.8.1 Emulation	223
5.8.1.1 Array Processor-based Emulation	224
5.8.1.2 FPGA-based Emulation	225
5.8.2 When to Use Emulation	226
5.8.3 Emulation Environment	227
5.8.4 Selecting an Emulation Solution	228
5.8.5 Limitations of Emulation	229
5.8.6 Emulation Methodology	230
5.8.7 Rapid Prototyping Systems	231
5.8.8 Hardware Accelerators	231
5.8.9 Design Partitioning	231
CHAPTER 6 <i>Hardware/Software Co-verification</i>	235
6.1 HW/SW Co-verification Environment	236
6.2 Emulation	237
6.3 Soft or Virtual Prototypes	238
6.3.1 Limitations	241
6.3.2 Methodology for Creating a Soft Prototype	241
6.3.3 Soft Prototype for the Bluetooth SOC	243
6.3.3.1 Adding Peripheral Models	243
6.3.3.2 Writing the Application Code	254
6.4 Co-verification	259
6.4.1 Co-verification Environment	260
6.4.2 Selecting a Co-verification Environment	261

6.4.2.1 Limitations of Co-verification Tools	262
6.4.3 Co-verification Methodology	262
6.4.4 UART Co-verification	263
6.4.4.1 UART Design Description	264
6.4.4.2 Transmitter	265
6.4.4.3 Receiver	265
6.4.4.4 UART Register Definitions	265
6.4.4.5 UART RTL Module Port	266
6.4.4.6 Verilog Shell for UART VHDL Design	272
6.4.4.7 Hardware Testbench for the UART	274
6.4.4.8 Software Code for Testing the UART	275
6.4.4.9 Software Header Files	276
6.4.4.10 Software Routines for the UART Test	277
6.4.4.11 Running the Simulation	283
6.4.4.12 Data Transfer from the UART to Flash Memory	283
6.4.4.13 Data Transfer from Flash Memory to the UART	285
6.5 Rapid Prototype Systems	286
6.5.1 Limitations of RPS	287
6.5.2 Reconfigurable RPS	287
6.5.2.1 Selecting a Reconfigurable RPS	288
6.5.2.2 Methodology of Reconfigurable RPS	289
6.5.3 Application-specific RPS	291
6.5.3.1 Commercially-available BSPs	291
6.5.3.2 Application-specific RPS for the Bluetooth SOC	292
6.5.3.3 Application-specific RPS Methodology	293
6.5.3.4 Limitations of Application-specific RPS	295
6.6 Comparing HW/SW Verification Methods	295
6.7 FPGA-based Design	296
6.7.1 Guidelines for FPGA-based Designs	296
6.7.2 FPGA-based Design Methodology	297
6.8 Developing Printed Circuit Boards	299
6.9 Software Testing	301
6.9.1 Software Development Lifecycle	301
6.9.1.1 Waterfall Lifecycle Model	301
6.9.1.2 V Lifecycle Model	302
6.9.1.3 Prototyping or Iterative Lifecycle Model	303
6.9.1.4 Software Maintenance	304
6.9.2 Guidelines for Software Development	304

6.9.3	Software Testing Best Practices	305
6.9.4	Debugging Tools	306
6.9.4.1	Software-based Tools	306
6.9.4.2	Hardware-based Tools	307
6.9.4.3	Debugging Techniques for SOC	308
6.9.5	Debugging Interrupts	309
CHAPTER 7	<i>Static Netlist Verification</i>	317
7.1	Netlist Verification	318
7.2	Bluetooth SOC Arbiter	320
7.3	Equivalence Checking	320
7.3.1	Selecting an EC Solution	322
7.3.1.1	Limitations of EC	323
7.3.2	EC Methodology	323
7.3.3	RTL to RTL Verification	325
7.3.4	RTL to Gate-Level Netlist Verification	325
7.3.5	Gate-Level Netlist to Gate-Level Netlist Verification	328
7.3.6	Debugging	329
7.3.6.1	Debugging Guidelines	330
7.3.7	Performing Equivalence Checking on an Arbiter	332
7.4	Static Timing Verification	334
7.4.1	Selecting an STV Solution	335
7.4.2	STV Methodology	336
7.4.3	Performing STV on an Arbiter	338
7.4.3.1	Running Timing Analysis on the Arbiter Block	339
CHAPTER 8	<i>Physical Verification and Design Sign-off</i>	347
8.1	Design Checks	348
8.2	Physical Effects Analysis	348
8.2.1	Parasitic Extraction	348
8.2.2	Inductance Effects	349
8.2.3	Signal Integrity	349
8.2.3.1	Crosstalk	350
8.2.3.2	IR Drop	351
8.2.4	Electromigration Effects	352
8.2.5	Subwavelength Challenges	352
8.2.5.1	Phase Shift Mask	353
8.2.5.2	Optical Proximity Correction	353
8.2.5.3	Verification after PSM and OPC	353

8.2.6 Process Antenna Effect	353
8.3 Design Sign-off	354
<i>Glossary</i>	361
<i>Index</i>	371