



SYSTEMS-ON-CHIP
NO SILICON

AHMED
AMINE
JERRY
WAYNE
WOLF

MULTIPROCESSOR SYSTEMS-ON-CHIPS



Multiprocessor Systems-on-Chips

Edited by

Ahmed Amine Jerraya

TIMA Laboratory

Wayne Wolf

Princeton University



AMSTERDAM • BOSTON • HEIDELBERG • LONDON
NEW YORK • OXFORD • PARIS • SAN DIEGO
SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO

Morgan Kaufmann Publishers is an imprint of Elsevier



MORGAN KAUFMANN PUBLISHERS

The Morgan Kaufmann Series in Systems on Silicon

Series Editors: Peter Ashenden, Ashenden Designs Pty. Ltd. and Adelaide University, and Wayne Wolf, Princeton University

The rapid growth of silicon technology and the demands of applications are increasingly forcing electronics designers to take a systems-oriented approach to design. This has led to new challenges in design methodology, design automation, manufacture and test. The main challenges are to enhance designer productivity and to achieve correctness on the first pass. *The Morgan Kaufmann Series in Systems on Silicon* presents high quality, peer-reviewed books authored by leading experts in the field who are uniquely qualified to address these issues.

The Designer's Guide to VHDL, Second Edition
Peter J. Ashenden

The System Designer's Guide to VHDL-AMS
Peter J. Ashenden, Gregory D. Peterson, and Darrell A. Teegarden

Readings in Hardware/Software Co-Design
Edited by Giovanni De Micheli, Rolf Ernst, and Wayne Wolf

Modeling Embedded Systems and SoCs
Axel Jantsch

ASIC and FPGA Verification: A Guide to Component Modeling
Richard Munden

Multiprocessor Systems-on-Chips
Edited by Ahmed Amine Jerraya and Wayne Wolf

Forthcoming Titles

Functional Verification
Bruce Wile, John Goss, and Wolfgang Roesner

Rosetta User's Guide: Model-Based Systems Design
Perry Alexander, Peter J. Ashenden, and David L. Barton

Rosetta Developer's Guide: Semantics for Systems Design
Perry Alexander, Peter J. Ashenden, and David L. Barton

Multiprocessor Systems-on-Chips

About the Editors

Ahmed Amine Jerraya (ahmed.jerraya@imag.fr) is Research Director with CNRS and is currently managing research dealing with Multiprocessor System-on-Chips at TIMA Laboratory, France. He received a degree in engineering from the University of Tunis in 1980 and the D.E.A., "Docteur Ingénieur", and the "Docteur d'Etat" degrees from the University of Grenoble in 1981, 1983, and 1989 respectively, all in computer sciences. From April 1990 to March 1991, he was a Member of the Scientific Staff at Nortel in Canada, working on linking system design tools and hardware design environments. He served as General Chair for the Conference DATE in 2001 and published more than 200 papers in International Conferences and Journals. He received the Best Paper Award at the 1994 ED&TC for his work on Hardware/Software Co-simulation.

Wayne Wolf (wolf@princeton.edu) is Professor of Electrical Engineering at Princeton University. Before joining Princeton, he was with AT&T Bell Laboratories, Murray Hill, New Jersey. He received the B.S., M.S., and Ph.D. degrees in Electrical Engineering from Stanford University in 1980, 1981, and 1984, respectively. His research interests include embedded computing, VLSI systems, and multimedia information systems. He is the author of *Computers as Components* and *Modern VLSI Design* (for which he won the ASEE/CSE and HP Frederick E. Terman Award). Wolf has been elected to Phi Beta Kappa and Tau Beta Pi. He is a Fellow of the IEEE and ACM and a member of the SPIE and ASEE.

Preface

This book had its origins in the Multiprocessor System-on-Chip (MPSoC) Workshop, which has been held every summer since 2001. We started the workshop to bring together a broad range of people who need to be involved in SoC design. MPSoCs are much more complex than ASICs and so require traditionally separate disciplines to converge. Computer architecture, real-time operating systems, embedded software, computer-aided design, and circuit design are all fields that must contribute to successful MPSoC designs.

An outstanding cross-section of experts from these disciplines attended the workshop and gave tutorial lectures that introduced their fields to the broader audience. We started to talk about how to make use of all the information that was presented at the workshop and make it available to a broader audience. After some debate we decided to organize a book of contributed chapters.

The contents of the book evolved somewhat as we developed it. Some of the workshop speakers didn't have enough room in their schedules for the considerable effort it takes to write a chapter. As a result, we ended up asking several other groups of people to write chapters. This allowed us to bring more perspectives into the discussion of MPSoC design.

Thanks to a great deal of effort, the book is now a reality. We believe that it will be useful to both professionals who want to know more about multiprocessor SoCs and students who are studying the subject. We hope that this collection of material will be informative and long-lasting. We'd like to thank the IEEE Circuits and Systems Society and the European Design and Automation Association for their sponsorship of the MPSoC Workshop over the years. We would like to thank all the participants—speakers and attendees—of the MPSoC workshops. We would

like to thank all of the authors for their work in creating these chapters. We would like to thank the book's reviewers: Nikil Dutt, University of California, Irvine; Axel Jantsch, Royal Institute of Technology (Stockholm, Sweden); Dan Phillips, Rochester Institute of Technology; Miodrag Potkonjak, UCLA; Gerald E. Sobelman, University of Minnesota; and James M. Ziobro, Rochester Institute of Technology. And we would like to thank the staff at Morgan Kaufman/Elsevier for their tireless efforts to make this book a reality.

Ahmed Amine Jerraya, Grenoble, France

Wayne Wolf, Princeton, New Jersey

Contents

About the Editors	vi
Preface	xix

1	The What, Why, and How of MPSoCs	1
	<i>Ahmed Amine Jerraya and Wayne Wolf</i>	
1.1	Introduction	1
1.2	What are MPSoCs	1
1.3	Why MPSoCs?	5
1.4	Challenges	10
1.5	Design Methodologies	11
1.6	Hardware Architectures	13
1.7	Software	14
	1.7.1 Programmer's Viewpoint	14
	1.7.2 Software architecture and design reuse viewpoint	15
	1.7.3 Optimization Viewpoint	16
1.8	The Rest of the Book	18

PART I HARDWARE **19**

2 Techniques for Designing Energy-Aware MPSoCs **21**

Mary Jane Irwin, Luca Benini, N. Vijaykrishnan, and Mahmut Kandemir

- 2.1 **Introduction** 21
- 2.2 **Energy-Aware Processor Design** 23
 - 2.2.1 **Reducing Active Energy** 24
 - 2.2.2 **Reducing Standby Energy** 26
- 2.3 **Energy-Aware Memory System Design** 27
 - 2.3.1 **Reducing Active Energy** 28
 - 2.3.2 **Reducing Standby Energy** 28
 - 2.3.3 **Influence of Cache Architecture on Energy Consumption** 29
 - 2.3.4 **Reducing Snoop Energy** 33
- 2.4 **Energy-Aware On-Chip Communication System Design** 34
 - 2.4.1 **Bus Encoding for Low Power** 34
 - 2.4.2 **Low Swing Signaling** 39
 - 2.4.3 **Energy Considerations in Advanced Interconnects** 41
- 2.5 **Energy-Aware Software** 44
- 2.6 **Conclusions** 46

3 Networks on Chips: A New Paradigm for Component-Based MPSoC Design **49**

Luca Benini and Giovanni De Micheli

- 3.1 **Introduction** 49
 - 3.1.1 **Technology Trends** 49
 - 3.1.2 **Nondeterminism in SoC Abstraction Models** 50
 - 3.1.3 **A New Design Approach to SoCs** 51

- 3.2 Signal Transmission on Chip 52
 - 3.2.1 Global Wiring and Signaling 53
 - 3.2.2 Signal Integrity 55
- 3.3 Micronetwork Architecture and Control 57
 - 3.3.1 Interconnection Network Architectures 58
 - 3.3.2 Micronetwork Control 63
- 3.4 Software Layers 73
 - 3.4.1 Programming Model 73
 - 3.4.2 Middleware Architecture 75
 - 3.4.3 Software Development Tools 78
- 3.5 Conclusions 80

4 Architecture of Embedded Microprocessors

81

Eric Rotenberg and Aravindh Anantaraman

- 4.1 Introduction 81
- 4.2 Embedded Versus High-Performance Processors:
A Common Foundation 82
- 4.3 Pipelining Techniques 85
 - 4.3.1 Bypasses 86
 - 4.3.2 Branch Prediction 87
 - 4.3.3 Caches 90
 - 4.3.4 Dynamic Scheduling 91
 - 4.3.5 Deeper Pipelining, Multiple-Instruction Issue,
and Hardware Multithreading 93
- 4.4 Survey of General-purpose 32-bit Embedded
Microprocessors 96
 - 4.4.1 ARM 98
 - 4.4.2 High-end Embedded MPUs 103
 - 4.4.3 Ubicom IP3023: Deterministic High Performance via
Multithreading 105
- 4.5 Virtual Simple Architecture (VISA): Integrating Non-Determinism
Without Undermining Safety 108
- 4.6 Conclusions 110

5 Performance and Flexibility for Multiple-Processor SoC Design 113

Chris Rowen

- 5.1 Introduction 113
- 5.2 The Limitations of Traditional ASIC Design 118
 - 5.2.1 The Impact of SoC Integration 120
 - 5.2.2 The Limitations of General-Purpose Processors 120
 - 5.2.3 DSP as Application-Specific Processor 122
- 5.3 Extensible Processors as an Alternative to RTL 122
 - 5.3.1 The Origins of Configurable Processors 123
 - 5.3.2 Configurable, Extensible Processors 123
 - 5.3.3 Configurable and Extensible Processor Features 130
 - 5.3.4 Extending a Processor 132
 - 5.3.5 Exploiting Extensibility 134
 - 5.3.6 The Impact of Extensibility on Performance 136
 - 5.3.7 Extensibility and Energy Efficiency 142
- 5.4 Toward Multiple-Processor SoCs 142
 - 5.4.1 Modeling Systems with Multiple Processors 144
 - 5.4.2 Developing an XTMP Model 144
- 5.5 Processors and Disruptive Technology 147
- 5.6 Conclusions 149

6 MPSoC Performance Modeling and Analysis 153

Rolf Ernst

- 6.1 Introduction 153
 - 6.1.1 Complex Heterogeneous Architectures 153
 - 6.1.2 Design Challenges 156
 - 6.1.3 State of the Practice 157
 - 6.1.4 Chapter Objectives 159
 - 6.1.5 Structuring Performance Analysis 160

6.2	Architecture Component Performance Modeling and Analysis	161
6.2.1	Processing Element Modeling and Analysis	161
6.2.2	Formal Processing Element Analysis	163
6.2.3	Communication Element Modeling and Analysis	165
6.2.4	Formal Communication Element Analysis	166
6.2.5	Memory Element Modeling and Analysis	167
6.2.6	Architecture Component Modeling and Analysis: Summary	168
6.3	Process Execution Modeling	168
6.3.1	Activation Modeling	168
6.3.2	Software Architecture	170
6.3.3	Process Execution Modeling: Summary	170
6.4	Modeling Shared Resources	171
6.4.1	Resource Sharing Principle and Impact	171
6.4.2	Static Execution Order Scheduling	172
6.4.3	Time-Driven Scheduling	173
6.4.4	Priority-Driven Scheduling	176
6.4.5	Resource Sharing-Summary	178
6.5	Global Performance Analysis	179
6.6	Conclusions	185

7 Design of Communication Architectures for High-Performance and Energy-Efficient Systems-on-Chips 187

Sujit Dey, Kanishka Lahiri, and Anand Raghunathan

7.1	Introduction	187
7.2	On-Chip Communication Architectures	189
7.2.1	Terminology	190
7.2.2	Communication Architecture Topologies	190
7.2.3	On-Chip Communication Protocols	192
7.2.4	Communication Interfaces	194
7.3	System-Level Analysis for Designing Communication Architectures	194
7.3.1	Trace-Based Analysis of Communication Architectures	196

7.4	Design Space Exploration for Customizing Communication Architectures	203
7.4.1	Communication Architecture Templates	203
7.4.2	Communication Architecture Template Customization	204
7.5	Adaptive Communication Architectures	210
7.5.1	Communication Architecture Tuners	210
7.6	Communication Architectures for Energy/Battery-Efficient Systems	216
7.6.1	Minimizing Energy Consumed by the Communication Architecture	216
7.6.2	Improving System Battery Efficiency Through Communication Architecture Design	218
7.7	Conclusions	222

8 Design Space Exploration of On-Chip Networks: A Case Study 223

Bishnupriya Bhattacharya, Luciano Lavagno, and Laura Vanzago

8.1	Introduction	223
8.2	Background	225
8.2.1	Function/Architecture Co-Design Methodology	225
8.2.2	Performance Modeling with Architecture Services	227
8.2.3	Mechanics of Architecture Services	229
8.2.4	Architecture Topology Binds Services	230
8.2.5	Communication Patterns	231
8.3	Modeling of Dataflow Networks	233
8.4	Case Study: Hiperlan/2 Application	235
8.4.1	Modeling the Hiperlan/2 Physical Layer	236
8.5	The Architectural Platform	238
8.5.1	Architectural Modeling	240
8.5.2	Mapping and Communication Refinement	241
8.6	Results	243
8.6.1	Communication Refinement	245
8.6.2	FPGA Alternatives	246
8.7	Conclusions	248

PART II SOFTWARE 249

9 Memory Systems and Compiler Support for MPSoC Architectures 251*Mahmut Kandemir and Nikil Dutt*

- 9.1 Introduction and Motivation 251
- 9.2 Memory Architectures 252
 - 9.2.1 Types of Architectures 254
 - 9.2.2 Customization of Memory Architectures 261
 - 9.2.3 Reconfigurability and Challenges 267
- 9.3 Compiler Support 269
 - 9.3.1 Problems 269
 - 9.3.2 Solutions 271
- 9.4 Conclusions 281

10 A SystemC-Based Abstract Real-Time Operating System Model for Multiprocessor System-on-Chips 283*Jan Madsen, Kashif Virk, and Mercury Jair Gonzalez*

- 10.1 Introduction 283
- 10.2 Basic Concepts and Terminology 286
 - 10.2.1 Platform Architecture 286
 - 10.2.2 Tasks 286
 - 10.2.3 Basics of Scheduling 288
- 10.3 Basic System Model 290
- 10.4 Uniprocessor Systems 292
 - 10.4.1 Link Model 292
 - 10.4.2 Task Model 293
 - 10.4.3 Scheduler Model 296
 - 10.4.4 Synchronization Model 300
 - 10.4.5 Resource Allocation Model 302

- 10.5 Multiprocessor Systems 303
 - 10.5.1 Multiprocessing Anomalies 306
 - 10.5.2 Interprocessor Communication 308
 - 10.5.3 Multiprocessor Example 309
- 10.6 Summary 311

11 Cost-Efficient Mapping of Dynamic Concurrent Tasks in Embedded Real-Time Multimedia Systems 313

Peng Yang, Paul Marchal, Chun Wong, Stefaan Himpe, Francky Catthoor, Patrick David, Johan Vounckx, and Rudy Lauwereins

- 11.1 Introduction 313
- 11.2 Platform Based Design 314
- 11.3 Related Work 315
- 11.4 Target Platform Architecture and Model 319
- 11.5 Task Concurrency Management 320
 - 11.5.1 Global TCM Methodology 321
 - 11.5.2 Two-Phase Scheduling Stage 321
 - 11.5.3 Scenarios to Characterize Data-Dependent TFs 324
 - 11.5.4 Platform Simulation Environment 326
- 11.6 3D Rendering QoS Application 327
- 11.7 Experimental Results 329
 - 11.7.1 Gray-Box Model 329
 - 11.7.2 Scenario Selection 329
 - 11.7.3 Reference Cases for Comparison 331
 - 11.7.4 Discussion of All Results 332
- 11.8 Conclusions 335

12 ILP-Based Resource-Aware Compilation 337

Jens Palsberg and Mayur Naik

- 12.1 Introduction 337
- 12.2 Examples 339

12.2.1	Instruction Scheduling	341
12.2.2	Energy Efficiency	343
12.2.3	Code-Size Minimization	345
12.2.4	Register Allocation	348
12.3	Open Problems	350
12.3.1	Combination	351
12.3.2	Correctness	352
12.3.3	Relationships with Other Approaches	353
12.4	Conclusions	354

PART III METHODOLOGY AND APPLICATIONS **355**

13 Component-Based Design for Multiprocessor Systems-on-Chip **357**

Wander O. Cesário and Ahmed A. Jerraya

13.1	From ASIC to System and Network on Chip	357
13.1.1	Applications for MPSoC	358
13.2	Basics for MPSoC Design	359
13.2.1	MPSoC Software Architectures	361
13.2.2	MPSoC Design Methods	362
13.2.3	Component Interface Abstraction	364
13.2.4	Component-Based Approach	365
13.3	Design Models for Component Abstraction	367
13.3.1	Conceptual Design Flow	368
13.3.2	Virtual Architecture Model	368
13.3.3	Target Architecture Model	370
13.3.4	The Hardware/Software Wrapper Concept	370
13.4	Component-Based Design Environment	371
13.4.1	Hardware Generation	371
13.4.2	Memory Wrapper Generation	375

13.4.3	Software Wrapper Generation	378
13.4.4	Simulation Model Generation	382
13.5	Component-Based Design of a VDSL Application	385
13.5.1	The VDSL Modem Architecture Specification	385
13.5.2	Virtual Architecture Specification	387
13.5.3	Resulting MPSoC Architecture	388
13.5.4	Evaluation	391
13.6	Conclusions	392

14 MPSoCs for Video

395

*Santanu Dutta, Jens Rennert, Teihan Lv, Jiang Xu, Shengqi Yang,
and Wayne Wolf*

14.1	Introduction	395
14.2	Multimedia Algorithms	396
14.2.1	Compression	396
14.2.2	Recognition	401
14.3	Architectural Approaches to Video Processing	402
14.4	Optimal CPU Configurations and Interconnections	406
14.4.1	Monolithic CPUs	406
14.4.2	Reconfigurable CPUs	407
14.4.3	Networked CPUs	408
14.4.4	Smart Interconnects	409
14.4.5	Software Support	409
14.5	The Challenges of SoC Integration and IP Reuse	410
14.6	The Panacea/Promise of Platform-Based Design	413
14.7	The Ever Critical Communication Bus Structures	416
14.7.1	PNX-8500 Structure	417
14.8	Design for Testability	421
14.9	Application-Driven Architecture Design	423
14.9.1	Application Characterization	423
14.9.2	Architectural Characterization	424
14.10	Conclusions	429