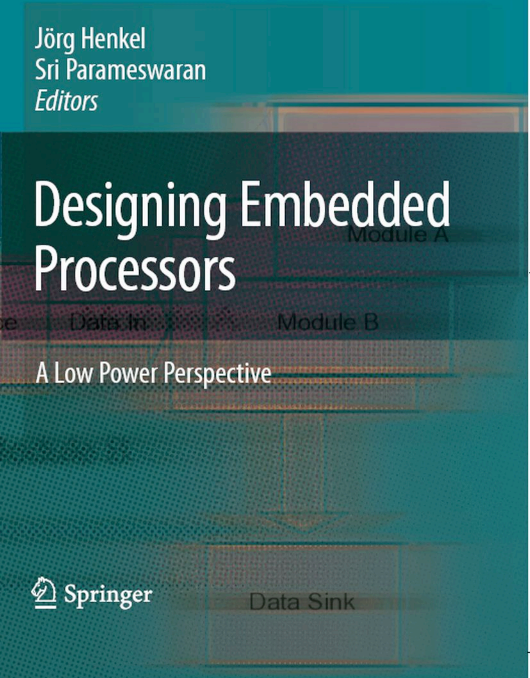
Jörg Henkel
Sri Parameswaran
Editors

Designing Embedded Processors

A Low Power Perspective



Springer

DESIGNING EMBEDDED PROCESSORS

Designing Embedded Processors

A Low Power Perspective

Edited by

J. HENKEL
University of Karlsruhe,
Germany

and

S. PARAMESWARAN
University of South Wales,
NSW, Australia

Springer

A C.I.P. Catalogue record for this book is available from the Library of Congress.

ISBN 978-1-4020-5868-4 (HB)
ISBN 978-1-4020-5869-1 (e-book)

Published by Springer,
P.O. Box 17, 3300 AA Dordrecht, The Netherlands.

www.springer.com

Printed on acid-free paper

All Rights Reserved
© 2007 Springer
No part of this work may be reproduced, stored in a retrieval system, or transmitted
in any form or by any means, electronic, mechanical, photocopying, microfilming, recording
or otherwise, without written permission from the Publisher, with the exception
of any material supplied specifically for the purpose of being entered
and executed on a computer system, for exclusive use by the purchaser of the work.

Contents

Foreword: Embedded Processors – what is next? xv
Jörg Henkel and Sri Parameswaran
1. Philosophy of This Book xvi
2. Contents xvi

Part I Application Specific Embedded Processors

1
Application-Specific Embedded Processors 3
Jörg Henkel, Sri Parameswaran, and Newton Cheung
1. Introduction and Motivation 4
   1.1 Some Trends in Designing SOCs 6
   1.2 Extensible Processor Platforms as a Possible Solution 7
   1.3 Open Issues and Key Techniques 10
   1.4 SOC Design Distinction 11
2. Challenges in Embedded Extensible Processor Design 12
   2.1 Code Segment Identification 12
   2.2 Extensible Instruction Generation 14
   2.3 Architectural Customization Selection 17
   2.4 Summary 19

References 19

2
Low-Power Design with NISC Technology 25
Bita Gorjiara, Mehrdad Reshadi, and Daniel Gajski
1. Introduction 25
2. Overview of NISC Technology 27
3. NISC Compared to Other Approaches 29
   3.1 NISC vs. ASIP 29
   3.2 NISC vs. VLIW 30
   3.3 NISC vs. Microcoded Architectures 30
   3.4 NISC vs. HLS 31
4. Overview of the Compilation Algorithm 32
5. Power Optimizations in NISC 37
   5.1 Reducing the Switching Capacitance 37
   5.2 Reducing Number of Cycles 39
6. Experiments 39
   6.1 The Effect of Pipeline Structure on Power and Performance 39
   6.2 Custom Datapath Design for DCT 42

7. Conclusion 49
References 49

3
Synthesis of Instruction Sets for High-Performance and Energy-Efficient ASIP 51
Jong-Eun Lee, Kiyoung Choi, and Nikil D. Dutt
1. Introduction 52
2. Related Work 53
3. Synthesizing Instruction Sets 54
4. Optimizing for Energy-Efficiency 55
4.1 ASIP Energy Model 56
4.2 EDP Change due to IS Customization 57
4.3 Modifying the Selection Algorithm 58
5. Experiments 58
5.1 Experimental Setup 59
5.2 Improvement through IS Synthesis 60
5.3 Effects of Bitwidth Variation 63
6. Conclusion 63
References 64

4
A Framework for Extensible Processor Based MPSoC Design 65
Fei Sun, Srivaths Ravi, Anand Raghunathan, and Niraj K. Jha
1. Introduction 66
2. Overview of MPSoC Synthesis 67
3. Custom Processor Synthesis 69
3.1 Motivation 70
3.2 Extensible Processor Synthesis Methodology 71
3.3 Template Generation 72
3.4 Experimental Results 75
4. Hybrid Custom Instruction and Co-Processor Synthesis 77
4.1 Methodology Overview 77
4.2 Multiobjective Evolutionary Algorithm 79
4.3 Experimental Results 81
5. Heterogeneous Multiprocessor Synthesis 84
5.1 Basic Synthesis Methodology 84
5.2 Enhanced Synthesis Methodology 87
5.3 Experimental Results 88
6. Related Work 92
7. Conclusions 93
References 93

5
Design and Run Time Code Compression for Embedded Systems 97
Sri Parameswaran, Jörg Henkel, Andhi Janapsatya, Talal Bonny, and Aleksandar Ignjatovic
1. Introduction 98
1.1 Design Time Compression 98
1.2 Run Time Code Compression 100

2. Related Work 100
3. Design Time – Cache Trace Compression and Cache Size Prediction 103
3.1 Methodology 103
3.2 Compression Algorithm 103
3.3 Cache Simulation Algorithm 109
3.4 Experimental Setup and Results 111
4. Run Time – Code Compression 114
4.1 Code Compression Technique 114
4.2 Hardware Implementation 119
4.3 Experiments and Results 121
4.4 Conclusion 124
References 125

Part II Embedded Memories

6
Power Optimisation Strategies Targeting the Memory Subsystem 131
Preeti Ranjan Panda
1. Introduction 131
2. Power-Efficient Memory Architectures 133
2.1 Partitioned Memory and Caches 133
2.2 Augmenting with Additional Buffers/Caches 134
2.3 Reducing Tag Comparison Power 137
2.4 Reducing Cache Leakage Power 139
2.5 Other Cache Ideas 140
3. Compiler Optimisations Targeting Memory Power 141
3.1 Data Layout 142
3.2 Instruction Layout 142
3.3 Scratch Pad Memory Utilisation 143
3.4 Memory Bank Utilisation 144
4. Application Specific Memory Customisation 145
5. Other Techniques: Dynamic Voltage Scaling, Compression, Encoding, etc. 146
5.1 Dynamic Voltage Scaling 147
5.2 Power Management in DRAM 147
5.3 Encoding 148
5.4 Compression 148
References 150

7
Layer Assignment Techniques for Low Energy in Multi-Layered Memory Organizations 157
Erik Brockmeyer, Bart Durinck, Henk Corporaal, and Francky Catthoor
1. Introduction 158
2. Basic Problem Definition 160
2.1 Data Reuse Analysis 161
2.2 Memory Hierarchy Assignment 162
2.3 Power, Area and Time Trade-off 165
2.4 Overall Methodology for MHLA 166

3. Data Reuse Analysis 167
3.1 Copy Candidates 168
3.2 Block Transfers 168
3.3 Non-Carried Copy Candidates 170
3.4 Branches in Reuse Tree 172
3.5 Write Accesses in the Reuse Tree 173
4. High-Level Estimations 173
4.1 Energy Estimation 173
4.2 Size Estimation 173
4.3 Time Estimation 175
5. Exploration Methodology for **MHLA** Search Space 182
5.1 Steering Heuristic 182
5.2 Incremental Assignment 182
6. Case Studies 183
6.1 QSDPCM 183
6.2 DAB Wireless Receiver 185
6.3 Execution Time Measurements 186
7. Related Work 186
8. Conclusion and Future Work 188
References 188

8
Memory Bank Locality and Its Usage in Reducing Energy Consumption 191
Mahmut Kandemir
1. Introduction and Motivation 191
2. Banked Memory Architecture and Low-Power Operating Modes 193
3. Affine Mappings of Arrays to Banks 195
4. Constraints for Bank Locality 197
5. Loop Transformations for Bank Locality 198
6. Implementing Array Decompositions 202
7. Global Optimizations 203
7.1 Single Nest, Multiple Arrays 203
7.2 Multiple Nest, Single Array 204
7.3 Multiple Nests, Multiple Arrays 206
7.4 Discussion 207
8. Folding Functions 207
9. Experiments 208
9.1 Benchmark Codes and Experimental Framework 208
9.2 Results 210
10. Concluding Remarks and Future Work 214
References 215

Part III Dynamic Voltage and Frequency Scaling

9
Fundamentals of Power-Aware Scheduling 219
Xiaobo Sharon Hu and Gang Quan
1. Introduction 219
2. Power and Performance Tradeoff in CMOS Circuits 220

3. Basics in Real-Time Scheduling 222
4. Impacts of Power-Aware Scheduling 223
5. Further Reading 224
5.1 Application Characteristics 224
5.2 Scheduling Decisions 225
5.3 Architectures 226
References 226

10
Static DVFS Scheduling 231
Gang Quan and Xiaobo Sharon Hu
1. Introduction 231
2. EDF Scheduling 232
3. Fixed-Priority Scheduling 233
3.1 Determining the Minimum Constant Speed for Each Job 235
3.2 Determining the Global Voltage Schedule 239
4. Related Work 240
References 241

11
Dynamic DVFS Scheduling 243
Padmanabhan S. Pillai and Kang G. Shin
1. Introduction 243
2. Schedulability Constraints for EDF and RM 244
3. Cycle-Conserving, Real-time DVFS 245
3.1 Cycle-Conserving EDF 247
3.2 Cycle-Conserving RM 249
4. Look-ahead DVFS 252
5. Evaluating Energy Performance of DVFS Algorithms 255
6. Related Readings 257
References 257

12
Voltage Selection for Time-Constrained Multiprocessor Systems 259
Alexandru Andrei, Petru Eles, Zebo Peng, Marcus Schmitz, and Bashir M. Al-Hashimi
1. Introduction 260
2. System and Application Model 261
3. Processor Power and Delay Models 262
4. Optimization of Mapping and Schedule for Voltage Selection 264
4.1 Genetic Task Mapping Algorithm 265
4.2 Genetic Scheduling Algorithm 266
5. Motivational Example 269
5.1 Optimizing the Dynamic and Leakage Energy 269
5.2 Considering the Transition Overheads 270
6. Problem Formulation 272

7. Optimal Continuous Voltage Selection 272
7.1 Continuous Voltage Selection without Overheads (CNOH) 272
7.2 Continuous Voltage Selection with Overheads (COH) 274
8. Optimal Discrete Voltage Selection 274
8.1 Problem Complexity 274
8.2 Discrete Voltage Selection without Overheads (DNOH) 275
8.3 Discrete Voltage Selection with Overheads (DOH) 276
8.4 Discrete Voltage Selection Heuristic 278
9. Experimental Results 279
10. Related Work 280
11. Summary 281
References 282

Part IV Compiler Techniques

13
Compilation Techniques for Power, Energy, and Thermal Management 287
Ulrich Kremer
1. Optimizing Compilers 287
2. Optimization Metrics 289
2.1 Power vs. Energy 289
2.2 Power/Energy vs. Performance 292
2.3 Power/Energy vs. Temperature 295
2.4 Summary 296
3. Future Compiler Research Directions 297
4. Techniques Covered in Subsequent Chapters 298
4.1 Dynamic Voltage and Frequency Scaling 298
4.2 Resource Hibernation 299
4.3 Remote Task Mapping 299
References 300

14
Compiler-Directed Dynamic CPU Frequency and Voltage Scaling 305
Chung-Hsing Hsu and Ulrich Kremer
1. DVFS 305
2. DVFS Scheduling is Challenging 306
3. Our DVFS Algorithm in a Nutshell 307
4. An Illustrating Example 308
5. Design and Implementation Issues 311
5.1 What is a Region 311
5.2 How Many Regions to Slow Down 312
5.3 What Region to Pick 312
5.4 Why Compiler-Directed 313
5.5 Is Profile-Driven Necessary 313
6. Evaluation Strategy 314
6.1 Hardware Platform 314
6.2 Software Platform 314
6.3 Benchmark Choices 315

7. Experimental Results 317
7.1 The Compilation Time 317
7.2 Effectiveness 318
7.3 Different Training Inputs 319
8. Conclusions and Future Work 321
References 322

324

15
Link Idle Period Exploitation for Network Power Management 325
Feihui Li, Guangyu Chen, Mahmut Kandemir, and Mustafa Karakoy
1. Introduction 326
2. Experimental Setup 327
3. Quantification of Last Idle Times 329
4. Network Abstraction and Hardware Support 330
5. Compiler Support 333
5.1 Splitting Loop Nests 334
5.2 Inserting Link Turn-off Instructions 337
5.3 Example 338
5.4 Discussion 340
6. Experimental Results 340
7. Concluding Remarks 343
References 344

16
Remote Task Mapping 347
Zhiyuan Li and Cheng Wang
1. Computation Offloading on Handheld Devices 347
1.1 An Overview 349
1.2 The Execution Model 351
1.3 Message Passing 352
1.4 A Number of Primitives 353
1.5 Code Generation 357
1.6 Task-Assignment Analysis 360
1.7 Experiments 365
1.8 Related Work 369
References 369

Part V Multi-Processors

17
A Power and Energy Perspective on MultiProcessors 373
Grant Martin
1. Introduction 373
1.1 Multicore and Multiprocessor Definitions 374
1.2 Power/Energy Drivers for Multiprocessor and Multicore Architectures 376
1.3 Classifying Multiprocessor Architectures 378

2. A Survey of Multiprocessor Approaches for Low-Power, Low-Energy Design 379
2.1 Basic Techniques 379
2.2 Formal Control of DVFS for CMP 382
2.3 Use of Transactional Memory in Multiprocessor Systems 383
3. Asymmetric Multiprocessing 383
3.1 Multiprocessor Systems of Configurable, Extensible Processors 383
4. Techniques Covered in Subsequent Chapters 385
4.1 Power-Performance Modeling and Design for Heterogeneous Multiprocessors 385
4.2 System-Level Design of Network-on-Chip Architectures 386
5. Conclusion 387
References 388

18
System-Level Design of Network-on-Chip Architectures 391
Karam S. Chatha and Krishnan Srinivasan
1. Introduction 392
1.1 Multi-Processor System-on-Chip (MPSoC) Architectures 392
1.2 Interconnection Woes and Network-on-Chip (NoC) 392
1.3 IP-based Methodology for NoC Design 394
2. NoC Router Architecture and Characterization 395
2.1 NoC Router Architecture 396
2.2 Power and Performance Characterization 398
2.3 Elements of NoC Design Process 399
3. Design and Optimization Techniques for NoC 404
3.1 MILP based Approach 405
3.2 Heuristic Approach 410
3.3 MOCA Phase I: Core to Router Mapping 410
3.4 MOCA Phase II: Route Generation 411
4. Related Research 415
4.1 NoC Router Architectures 415
4.2 NoC Performance and Power Consumption Models 416
4.3 NoC Design and Optimization Techniques 416
5. Conclusion 416
References 417

19
Power-Performance Modeling and Design for Heterogeneous Multiprocessors 423
JoAnn M. Paul and Brett H. Meyer
1. Introduction 423
2. The design space 424
3. MESH 425
3.1 MESH as a Performance Simulator 426
3.2 Energy Modeling in MESH 428
3.3 Power-Performance Design Evaluation in MESH 429
4. Performance Evaluation of SCHMs 430
4.1 Application Performance Classification 432

4.2 Arrival Characteristics and Timing 432
4.3 Observations 434
5. Heterogeneous Performance Balance 435
5.1 Processing Elements 435
5.2 Applications and System Scheduling 438
5.3 Tiles and the Tiled System 439
5.4 Modeling Assumptions 440
5.5 Experiments and Results 441
5.6 Discussion 445
6. Conclusions 445
References 446

Part VI Reconfigurable Computing

20
Basics of Reconfigurable Computing 451
Reiner Hartenstein and TU Kaiserslautern
1. Configurable Cells 459
2. von Neumann vs. Reconfigurable Computing Paradigm 466
3. Future of FPGA (Technologies) 476
4. Coarse-Grained vs. Fine-Grained Reconfigurability 478
5. History of FPGAs 487
References 489

21
Dynamic Reconfiguration 503
Jürgen Becker and Michael Hübner
1. Basis of Reconfiguration and Terminology 504
2. Exploiting Dynamic and Partial Reconfiguration for Power-Reduction 506
References 511

22
Applications, Design Tools and Low Power Issues in FPGA Reconfiguration 513
Adam Donlin
1. Introduction 513
2. Applying Reconfiguration 514
2.1 Applications of FPGA Reconfiguration 516
3. Design Flow for Reconfigurable Systems 522
3.1 Dynamic Reconfiguration Design Flow 523
3.2 Deploying Dynamic Reconfiguration 529
4. Low Power and FPGA Reconfiguration 532
4.1 The FPGA Low Power Landscape 532
4.2 Low Power Use Cases for Dynamic Reconfiguration 535
5. Conclusion 539
Acknowledgments 539
References 540

Index 543

Foreword

Embedded Processors – What is Next?

Jörg Henkel and Sri Parameswaran

These are exciting times for the system level designer/researcher. The world seems to burgeon with embedded systems. Consumers demand superior electronic products more often than ever before. Moore's law continues to be valid 40 years after it was first stated, allowing the adventurous to design with billions of transistors. Demanding managers require products in shorter time than previously. All of this has led to an unending search for superior methods and tools to design embedded systems.

Interminable appetite by consumers for portable embedded systems has continued to churn out chips with ever growing functionality. Soaring non-recurring engineering costs of chips has forced designers towards large scale chips which exhibit computation capability along with communication protocols. These designs are expected to be flexible by being software upgradeable, reduce time to market by being rapidly verifiable, and produced in large volumes to reduce the cost per chip. To truly make such systems ubiquitous, it is necessary to reduce the power consumed by such a system. These often conflicting demands have meant that chips have to exhibit smaller footprint and consume less power. For a long time now, the narrowing feature sizes of chips and continuously reducing supply voltages were sufficient to satisfy the size and power demands. Unfortunately, this trend towards smaller feature sizes and lower supply voltages is slowing due to physical limitations. This has led to looking at system level methods to reduce power in embedded systems.

Unlike circuit level methods to reduce power, system level methods often allow a plethora of techniques to be applied at various levels. Often these techniques are orthogonal to one another, and can be applied simultaneously, assuming that the designer has sufficient time. Some of these techniques are at the architecture level-such as application specific processors, some are run-time techniques-which respond to the workload by switching voltage and frequency, some are at design time-such as compiler techniques which allow lower power consumption of the compiled code.

Time is indeed changing the way we design systems. Reducing design time and the size of a design team are increasingly crucial. Numerous tools and methods are available to educated designer. Many of these are point tools, though several tool vendors work tirelessly towards making these point tools interoperable so that seamless design flows can be created, which are useable by designers, increasing productivity several times. While such design flows from the RTL level down are quite mature, the design tools and flows at the system level are still evolving and will evolve for some time to come. Research in this area is very much alive at this time and will be for the foreseeable future.

This book examines system level design techniques, which allow the automation of system level designs, with a particular emphasis towards low power. We expect researchers, graduate students and system level designers to benefit from this book. The authors of the individual chapters are all well known researchers in their respective fields.

1. Philosophy of This Book

In order to provide a maximum degree of usability for novices and researchers alike, the book is organized in the following way: each of the individual six sub-topics comprises one section that introduces to the whole area. For example, the section Application Specific Embedded Processors starts with the chapter Designing Application Specific Embedded Processors. That chapter gives an introduction to the field in a textbook-like style along with a mentioning of the specific grand challenges followed by a review of the most relevant related work. Thus, the first chapter of a section introduces a novice to the field. Experts in the respective fields may skip that chapter. The subsequent chapters then pick a research topic and present state-of-the-art research approaches that are representative and most challenging from the current perspective. If too specific, more generally interested readers may skip those chapters. In that sense, the book is useful for both, the expert researcher as well as the novice. Also in that sense, this book provides a new (and hopefully useful) concept.

2. Contents

Though there are certainly many more aspects in designing embedded processors with respect to low power, we had to restrict the book and eventually identified six main topics (according to the six sections) namely: I. Application