

Marco Platzner
Jürgen Teich
Norbert Wehn
Editors

Dynamically Reconfigurable Systems

Architectures, Design Methods
and Applications



Springer

Dynamically Reconfigurable Systems

Marco Platzner • Jürgen Teich • Norbert Wehn
Editors

Dynamically Reconfigurable Systems

Architectures, Design Methods
and Applications

 Springer

Editors

Marco Platzner
Department of Computer Science
University of Paderborn
Warburger Str. 100
33098 Paderborn
Germany
platzner@upb.de

Jürgen Teich
Hardware/Software Co-Design
Department of Computer Science
University of Erlangen-Nuremberg
Am Weichselgarten 3
91058 Erlangen
Germany
teich@cs.fau.de

Norbert Wehn
TU Kaiserslautern
Erwin-Schrödinger-Str.
67663 Kaiserslautern
Germany
wehn@eit.uni-kl.de

ISBN 978-90-481-3484-7

e-ISBN 978-90-481-3485-4

DOI 10.1007/978-90-481-3485-4

Springer Dordrecht Heidelberg London New York

Library of Congress Control Number: 2009944064

© Springer Science+Business Media B.V. 2010

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.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Foreword

After six years of exciting and intensive research, one of the world's largest and longest-running programs of research into Reconfigurable Computing is concluding with the publication of the research papers in this volume. The research was launched in 2003 by Deutsche Forschungsgemeinschaft, the German Research Foundation, as "Schwerpunktprogramm (SPP 1148)" under the title "Rekonfigurierbare Rechensysteme". This translates as the Priority Programme in Reconfigurable Computing Systems. For convenience, we will refer to it here as SPP 1148.

Several aspects of SPP 1148 made it noteworthy from its inception. Beneath the umbrella of reconfigurable computing systems, several key topics were identified. These included theoretical aspects, modeling, languages, analysis, design methodologies, computer-aided design (CAD) tools, architectures and applications. From the onset, the scope of the research was deliberately formulated to be as multidisciplinary as possible. A national network of excellence was established with the aim of encouraging participation from as many academics with novel individual contributions as possible while simultaneously emphasizing and promoting interdisciplinary collaboration.

The six-year program was organized as three successive funding periods of two-years each. At each interval, new projects were proposed and existing projects were reviewed. The two largest project sub-themes were design methodologies and tools, and architectures and applications. Feedback from graduate students who participated in SPP 1148 indicates that they felt especially privileged to have been able to conduct their studies within the framework of this program. In particular, they benefited from the highly collaborative environment that was nurtured by the participating institutions and the continuity afforded by such a sustained period of research funding.

Throughout the conduct of SPP 1148, we have had the privilege of interacting and collaborated directly with many of the research teams. We have become familiar with the majority of the research projects albeit at different levels of detail. We have had the opportunity to observe the progress of SPP 1148 as a whole from our complementary positions in academia and industry. There can be no doubt that the most immediate impact of the DFG's (German Research Foundation) foresight

was to establish German academia as the foremost group in the world for advanced research into reconfigurable computing over the last six years.

The legacy of the SPP 1148 in Reconfigurable Computing Systems is that Germany has developed an impressive national capability in precisely those areas that will be of greatest relevance for the next decade. As we reach for 40 nm integrated circuits and beyond, it is clear that fewer companies and fewer architectures will be viable at these ever more advanced process nodes. The successful architectures will be highly concurrent and will combine programmability and reconfigurability capabilities.

This book is unique in many different ways. Not only does it provide a structured compilation for the research resulting from this extensive program, it also gives an excellent overall insight into various strategic directions in which the entire field of reconfigurable technologies and systems may be heading. In addition, in the four separate parts of the book, the coverage of each of the subjects is comprehensive. Aspects relating to architectures, design methodologies, design tools and a large number of applications are all included. No other book currently in print has such an extensive and overall coverage as this. As such, it is equally suitable for supporting graduate level courses and for practical engineers working in the field of reconfigurable hardware and particularly in FPGAs. We are delighted to see the work and experience of such a large group of researchers and engineers being shared with the reconfigurable community at large.

Patrick Lysaght, Xilinx Corporation
Peter Cheung, Imperial College London
August 2009

Preface

While the idea of creating computing systems with flexible hardware dates back to the 1960s, it was the emergence of the SRAM-based field-programmable gate array (FPGA) in the 1980s that boosted *Reconfigurable Computing* as a research and engineering field. Since then, reconfigurable computing has become a vibrant area with an increasingly growing research community and exciting commercial ventures.

Reconfigurable computing devices are able to adapt their hardware to application demands and serve broad and relevant application domains from embedded to high-performance computing, including automotive, aerospace and defense, telecommunication and networking, medical and biometric computing. Especially in the embedded computing domain with its many and often conflicting objectives reconfigurable computing systems offer new trade-offs. Embedded systems are fueled by microelectronics where one of the currently biggest challenges is the trade-off between flexibility and cost. Reconfigurable devices, especially FPGAs, fill this gap since they provide flexibility at both design-time and run-time. Consequently, in the last years we have seen declining ASIC design starts but continuously increasing FPGA design starts. Continuing advances in the miniaturization of microelectronic components have made it possible to integrate systems with multiple processors on a single chip at the size of a fingernail (system-on-chip (SoC)). Often, the production volumes for SoCs are rather low jeopardizing their economic benefits. On the other hand, modern FPGAs are basically complex SoCs integrating embedded processors, signal processing capabilities, multi-gigabit transceivers and a broad portfolio of IP cores. Thus FPGAs are positioned to become a real alternative to ASICs and ASPPs.

This book is the first ever to focus on the emerging field of *Dynamically Reconfigurable Computing Systems*. While programmable logic and design-time configurability are well elaborated and covered in various books, this book presents a unique overview over the state of the art and recent results for dynamic and run-time reconfigurable computing systems. This book targets graduate students and practitioners alike. Over the last years, many educational institutions began to offer courses and seminars on different aspects of reconfigurable computing systems. We recommend this book as reading material for the advanced graduate level and entrance into own

research on dynamically reconfigurable systems. Reconfigurable hardware is not only of utmost importance for large manufacturers and vendors of microelectronic devices and systems, but also a very attractive technology for smaller and medium-sized companies. Hence, this book addresses also researchers and engineers actively working in the field and updates them on the newest developments and trends in runtime reconfigurability.

The book is organized into four parts that present recent efforts and breakthroughs in architectures, design methods and tools, and applications for dynamically reconfigurable computing systems:

Architectures: Three chapters on architectures discuss different dynamically reconfigurable platforms, including multigrained and application-specific architectures as well as an FPGA-based computing system supporting efficient partial reconfiguration.

Design Methods and Tools—Modeling, Evaluation and Compilation: The first part on design methods and tools features four chapters focusing on modeling and evaluation aspects for dynamically reconfigurable hardware, on creating compilers for reconfigurable devices, and on supporting dynamic reconfiguration through object-oriented programming.

Design Methods and Tools—Optimization and Runtime Systems: The second part on design methods and tools comprises six chapters that are devoted to resource allocation in dynamically reconfigurable systems, split into challenging optimization problems that need to be solved during compilation time, e.g., temporal partitioning, and online resource allocation which is provided by a novel breed of reconfigurable hardware runtime systems.

Applications: The last part of the book presents seven chapters with applications of dynamically reconfigurable hardware technology to relevant and demanding domains, including mobile communications, network processors, automotive vision, and geometric algebra.

This book presents the results of a six-years research initiative on dynamically reconfigurable computing systems, initiated and coordinated by Jürgen Teich and funded by the German Research Foundation (DFG) within the Priority Programme (Schwerpunktprogramm) 1148 from 2003 to 2009. To make dynamic reconfigurable computing become a reality this joint research initiative bundled multiple projects and, at times, involved up to 50 researchers working in the topic.

Equivalently, this book summarizes more than 100 person years of research work and more than 20 PhD students have already submitted and defended their theses based on research performed in this initiative. The material presented in this book thus summarizes the golden fruits, major achievements and biggest milestones of this joint research initiative.

We are very grateful to the German Research Foundation for funding and continuously supporting this initiative. We would also like to thank all the researchers contributing to the programme and to this book, the many national and international reviewers as well as the industrial companies that have been steadily supporting our efforts. Additionally, we would like to acknowledge the assistance of Josef Angermeier, Martina Jahn, Enno Lübbers, and Felix Reimann in supporting the editorial

process. Last but not least, we thank Springer for giving us the opportunity to publish our results with them.

We hope you enjoy reading this book!

Marco Platzner, Jürgen Teich, Norbert Wehn

Contents

Part I Architectures

1	Development and Synthesis of Adaptive Multi-grained Reconfigurable Hardware Architecture for Dynamic Function Patterns	3
	Alexander Thomas and Jürgen Becker	
1.1	Introduction	3
1.2	HoneyComb Architecture	5
1.2.1	Architectural Considerations	5
1.2.2	HoneyComb Overview	8
1.2.3	Communication Network and Online Adaptive Routing Technique	9
1.2.4	Datapath Cells (DPHC)	11
1.2.5	Memory Cells (MEMHC)	13
1.2.6	Input/Output Cells (IOHC)	13
1.2.7	Power Saving Techniques	15
1.3	Tool Support	15
1.3.1	HoneyComb Assembler and the Hierarchical Programming Model	16
1.3.2	HoneyComb Language (HCL) and Compiler	17
1.3.3	Debugging Tool—HoneyComb Viewer	18
1.3.4	Super-Configuration Generator, Configuration Editor and Configuration Manager	18
1.3.5	Hierarchy Generator	19
1.3.6	Application and Synthesis Results	19
1.4	Future Work	22
1.5	Conclusion	23
	References	23

2	Reconfigurable Components for Application-Specific Processor Architectures	25
	Tobias G. Noll, Thorsten von Sydow, Bernd Neumann, Jochen Schleifer, Thomas Coenen, and Götz Kappen	
2.1	Introduction	25
2.2	Parameterized eFPGA Target Architecture	27
2.3	Physical Implementation of Application Class Specific eFPGAs . .	32
2.4	Mapping and Configuration	35
2.5	Examples of (Stand Alone) eFPGAs as SoC Building Blocks	37
2.6	Examples of eFPGAs as Coprocessors to Standard RISC Processor Kernels	39
2.6.1	General-Purpose Processors Coupled with eFPGAs	41
2.6.2	ASIPs Coupled with eFPGA-Based Accelerators	41
2.7	Conclusion	46
	References	47
3	Erlangen Slot Machine: An FPGA-Based Dynamically Reconfigurable Computing Platform	51
	Josef Angermeier, Christophe Bobda, Mateusz Majer, and Jürgen Teich	
3.1	Introduction	52
3.2	Drawbacks of Existing Dynamically Reconfigurable Systems	52
3.3	The Erlangen Slot Machine	55
3.3.1	Architecture Overview	55
3.3.2	The BabyBoard	56
3.3.3	The MotherBoard	58
3.4	Inter-module Communication	59
3.4.1	Communication Between Adjacent Modules	60
3.4.2	Communication via Shared Memory	60
3.4.3	Communication via RMB	60
3.4.4	Communication via the Crossbar	61
3.5	Reconfiguration Manager	62
3.5.1	Flexible Plugin Architecture	62
3.5.2	Reconfiguration Scenarios	64
3.5.3	Implementation Results	64
3.6	Case Study: Video and Audio Streaming	65
3.7	Usage of the ESM in Different Fields	67
3.8	Conclusions	69
	References	70

Part II Design Methods and Tools—Modeling, Evaluation and Compilation

4	Models and Algorithms for Hyperreconfigurable Hardware	75
	Sebastian Lange and Martin Middendorf	
4.1	Introduction	75
4.2	Hyperreconfigurable Machines	76

4.3	Example Architectures and Test Cases	81
4.4	The Partition into Hypercontexts Problem	83
4.4.1	Experiments and Results	84
4.5	Diverse Granularity in Multi-level Reconfigurable Systems	86
4.6	Partial Reconfiguration and Hyperreconfiguration	90
4.6.1	Frame Model of Partially Reconfigurable Architectures	90
4.6.2	Results	92
4.7	Conclusions	93
	References	93
5	Evaluation and Design Methods for Processor-Like Reconfigurable Architectures	95
	Sven Eisenhardt, Thomas Schweizer, Julio Oliveira Filho, Tommy Kuhn, and Wolfgang Rosenstiel	
5.1	Introduction	95
5.2	Benefits and Costs of Processor-Like Reconfiguration	97
5.2.1	CRC Model	97
5.2.2	Compiler	99
5.2.3	Evaluation	100
5.3	Specialization/Instruction Set Extension	102
5.3.1	Methodology	104
5.3.2	Study Case: Multipoint FFT for Scalable OFDMA Based Systems	105
5.4	Optimizing Power	108
5.4.1	Optimizing Power by Instruction Set Extensions	108
5.4.2	Optimizing Power by Dual- V_{DD} Architectures	108
5.5	Optimizing External Reconfiguration	111
5.5.1	Multi-Context Configuration Prefetching	112
5.5.2	Speculative Configuration Prefetching	113
5.5.3	Experimental Results	114
5.6	Conclusion	115
	References	115
6	Adaptive Computing Systems and Their Design Tools	117
	Andreas Koch	
6.1	Introduction	117
6.2	Execution Model	118
6.3	ACS Architecture	120
6.3.1	Reconfigurable System-on-Chip Architecture	120
6.3.2	Operating System Integration	121
6.3.3	Evaluation	125
6.4	Hardware/Software Co-compilation Flow	125
6.4.1	Overview	125
6.4.2	Profile-Based Inlining and Partitioning	126
6.4.3	CMDFG Intermediate Representation	127

6.4.4	CoCoMa Controller Model	129
6.5	Infrastructure	131
6.5.1	Parametrized Module Library	131
6.5.2	Physical Design Aspects	131
6.5.3	Reconfiguration Scheduling	133
6.6	Lessons Learned	134
6.7	Future Work	136
6.8	Conclusions	136
	References	137
7	POLYDYN—Object-Oriented Modelling and Synthesis Targeting Dynamically Reconfigurable FPGAs	139
	Andreas Schallenberg, Wolfgang Nebel, Andreas Herrholz, Philipp A. Hartmann, Kim Grüttner, and Frank Oppenheimer	
7.1	Introduction	139
7.2	Related Work	141
7.3	Methodology	142
7.3.1	General Concept	143
7.3.2	Lifetime and Conflict Management	145
7.4	Derived Interface Classes	147
7.5	Modelling Example: Car Audio System	148
7.5.1	Coding Style: From C++ Polymorphism to OSSS+R	149
7.5.2	Simulation	152
7.6	Synthesising OSSS+R	153
7.6.1	From OSSS+R to RT Level	153
7.7	Evaluation	156
7.8	Conclusion and Future Work	157
	References	157
Part III Design Methods and Tools—Optimization and Runtime Systems		
8	Design Methods and Tools for Improved Partial Dynamic Reconfiguration	161
	Markus Rullmann and Renate Merker	
8.1	Introduction	161
8.2	Motivation	163
8.3	Reconfigurable Module Architecture and Partitioning	164
8.4	Reconfiguration State Graph	165
8.5	Module Mapping and Virtual Architecture	166
8.6	High-Level Synthesis of Reconfigurable Modules	168
8.6.1	Resource Type Binding	169
8.6.2	Resource Instance Binding	170
8.6.3	Control Generation	172
8.7	Experiments	173
8.7.1	Experimental Setup	173

8.7.2	Benchmark Results and Discussion	174
8.8	System Design for Efficient Partial Dynamic Reconfiguration	178
	References	180
9	Dynamic Partial Reconfiguration by Means of Algorithmic Skeletons—A Case Study	183
	Norma Montealegre and Franz J. Rammig	
9.1	Introduction	183
9.2	Overview of the Overall System	185
9.3	Library of Algorithmic Skeletons	188
9.3.1	Algorithmic Skeletons for Defining the Structure of the Design	188
9.3.2	Algoskels	189
9.3.3	Algorithmic Skeletons for Partial Reconfigurable Systems	190
9.4	Application Scenario: Channel Vocoder Analyzer	191
9.5	Conclusion	197
	References	197
10	ReCoNodes—Optimization Methods for Module Scheduling and Placement on Reconfigurable Hardware Devices	199
	Ali Ahmadiania, Josef Angermeier, Sándor P. Fekete, Tom Kamphans, Dirk Koch, Mateusz Majer, Nils Schweer, Jürgen Teich, Christopher Tessars, and Jan C. van der Veen	
10.1	Introduction	199
10.2	Offline and Online Aspects of Defragmenting the Module Layout of a Partially Reconfigurable Device	201
10.2.1	Two-dimensional Strip Packing	203
10.2.2	Defragmentation Approach and Computational Results	204
10.2.3	Online Packing	204
10.3	Minimizing Communication Cost for Reconfigurable Slot Modules	206
10.3.1	Mathematical Model	207
10.3.2	Case Study and Results	210
10.4	No-break Dynamic Defragmentation of Reconfigurable Devices	211
10.4.1	Model and Problem Description	212
10.4.2	Problem Complexity and Moderate Densities	213
10.4.3	A Heuristic Method	213
10.5	Scheduling Dynamic Resource Requests	214
10.5.1	An ILP	215
10.5.2	Heuristic Methods	217
	References	219

11	ReCoNets—Design Methodology for Embedded Systems Consisting of Small Networks of Reconfigurable Nodes and Connections	223
	Christian Haubelt, Dirk Koch, Felix Reimann, Thilo Streichert, and Jürgen Teich	
11.1	Introduction	223
11.2	System Model	225
11.3	A Distributed Operating System Architecture for Networked Embedded Systems	226
11.3.1	Self-healing and Self-adaptiveness	227
11.3.2	Hardware/Software Task Migration	234
11.3.3	Hardware/Software Morphing	235
11.3.4	Hardware/Software Checkpointing	237
11.4	Design and Synthesis of ReCoNets	238
11.4.1	Design Space Exploration	238
11.4.2	Dependability Analysis	239
11.5	Demonstrator	240
	References	242
12	Adaptive Runtime System with Intelligent Allocation of Dynamically Reconfigurable Function Model and Optimized Interface Topologies	245
	Lars Braun, Tobias Schwalb, Philipp Graf, Michael Hübner, Michael Ullmann, K.D. Müller-Glaser, and Jürgen Becker	
12.1	Introduction	246
12.2	Partial and Dynamic Reconfiguration	247
12.2.1	One-dimensional Dynamic and Partial Reconfigurable System Architecture	247
12.3	Network on Chip	250
12.3.1	Router Base Modules for the On-line Placement	250
12.3.2	Switch for 2D Mesh Based NoC Approach	252
12.4	On Demand System Adaption	253
12.4.1	Physical on Line Routing of Communication Structures	253
12.4.2	Physical On-line Routing of Parameterizable Filter Modules	254
12.5	System Modelling	255
12.6	Tool Chain	257
12.7	Model Debugging	259
12.7.1	Problem	259
12.7.2	Debugging Flow	259
12.7.3	Interface and Architecture	260
12.8	Test	262
12.9	Conclusions	263
	References	265

13 ReconOS: An Operating System for Dynamically Reconfigurable Hardware	269
Enno Lübbers and Marco Platzner	
13.1 Introduction	269
13.2 Related Work	270
13.3 Programming Model	271
13.3.1 Hardware Threads	273
13.3.2 Thread Creation and Termination	273
13.4 Run-Time System	274
13.4.1 Hardware Architecture	275
13.4.2 Hardware Multitasking	278
13.4.3 Software Architecture	278
13.5 Implementation	280
13.5.1 Target Platforms	280
13.5.2 Prototypes	280
13.5.3 Debugging and Monitoring	283
13.6 Experimental Measurements	284
13.6.1 Application Case Studies	286
13.7 Conclusion and Outlook	288
References	288

Part IV Applications

14 FlexiChaP: A Dynamically Reconfigurable ASIP for Channel Decoding for Future Mobile Systems	293
Matthias Alles, Timo Vogt, Christian Brehm, and Norbert Wehn	
14.1 Introduction	293
14.2 Channel Codes	295
14.2.1 Convolutional Codes	296
14.2.2 Turbo Codes	296
14.2.3 LDPC Codes	297
14.3 Decoder Requirements	299
14.4 ASIP Design Methodologies	299
14.5 Architecture	300
14.5.1 General Considerations	300
14.5.2 Memory Concept	302
14.5.3 Pipeline	303
14.5.4 Dynamically Reconfigurable Channel Code Control	304
14.5.5 Instruction Set	306
14.6 ASIP Validation	308
14.7 Results	310
References	313

15	Dynamically Reconfigurable Systems for Wireless Sensor Networks	315
	Heiko Hinkelmann, Peter Zipf, and Manfred Glesner	
15.1	Introduction	315
15.1.1	Outline	316
15.2	Motivation and Background	316
15.3	Design of a Reconfigurable Function Unit	317
15.3.1	Functional Coverage of the RFU	318
15.3.2	The RFU Data Path	318
15.4	Dynamic Reconfiguration	320
15.4.1	Intra-task Reconfiguration	320
15.4.2	Inter-task Reconfiguration	323
15.5	General System Architecture	324
15.6	Evaluation Results	325
15.6.1	Test Settings	326
15.6.2	Synthesis Results	326
15.6.3	Evaluation of Energy Efficiency	326
15.7	Prototyping of the Sensor Node System	331
15.8	Generalisation of the Results	332
15.9	Conclusion	332
	References	333
16	DynaCORE—Dynamically Reconfigurable Coprocessor for Network Processors	335
	Carsten Albrecht, Jürgen Foag, Roman Koch, Erik Maehle, and Thilo Pionteck	
16.1	Introduction	335
16.2	Network Processors	336
16.2.1	Requirements	337
16.3	System Architecture	338
16.4	Model	339
16.4.1	Principles of Theory	340
16.4.2	Modelling DynaCORE	340
16.4.3	Simulation	341
16.5	Runtime Adaptive Network-on-Chip	343
16.5.1	Architecture	344
16.5.2	Runtime Adaptation	345
16.5.3	Fault Tolerance	345
16.6	Reconfiguration Management	346
16.6.1	Basic Method	347
16.6.2	Optimised Approach	347
16.7	Technical Aspects	350
16.8	Evaluation	351
16.9	Summary	353
	References	353

17 FlexPath NP—Flexible, Dynamically Reconfigurable Processing Paths in Network Processors 355
 Rainer Ohlendorf, Michael Meitinger, Thomas Wild, and Andreas Herkersdorf

17.1 Introduction 355

17.2 FlexPath NP Concept 357

 17.2.1 Application Dependent Path Decision 358

 17.2.2 Load Dependent Path Decision 359

17.3 Formal Analysis 360

17.4 Simulative Exploration 361

 17.4.1 FlexPath NP Architecture Evaluation 362

 17.4.2 Load Balancing in FlexPath NP 363

17.5 FPGA Demonstrator 364

 17.5.1 Architecture 364

 17.5.2 Experiments 367

17.6 Conclusion 372

References 373

18 AutoVision—Reconfigurable Hardware Acceleration for Video-Based Driver Assistance 375
 Christopher Claus and Walter Stechele

18.1 Introduction 375

 18.1.1 State of the Art & Related Work 377

 18.1.2 Typical Scenario & Hardware Accelerators 378

18.2 AutoVision Architecture 380

 18.2.1 The AddressEngine—A Pixel Processing Pipeline 381

18.3 Fast Dynamic Partial Reconfiguration 383

 18.3.1 Motivation for Fast Reconfiguration 383

 18.3.2 Bitstream Modification 384

 18.3.3 Hardware Modification 386

18.4 Results 390

18.5 Performance of the Engines 391

18.6 Conclusion and Outlook 392

References 393

19 Procedures for Securing ECC Implementations Against Differential Power Analysis Using Reconfigurable Architectures 395
 Marc Stöttinger, Felix Madlener, and Sorin A. Huss

19.1 Introduction 395

19.2 Elliptic Curve Cryptography 396

 19.2.1 Elliptic Curve Arithmetic 397

 19.2.2 Finite Field Arithmetic 397

19.3 Side Channel Attacks 401

 19.3.1 Information Leaking of ECC Hardware Architectures ... 404

 19.3.2 General Countermeasure Techniques 405

19.4	Countermeasure Through Reconfiguration	406
19.4.1	Securing ECC on Finite Field Arithmetic Level	407
19.4.2	Securing ECC on Elliptic Curve Arithmetic Level	409
19.5	DPA Experiments on Countermeasures	411
19.5.1	Resistance of the Secured eMSK-Multiplier	412
19.6	Conclusion and Future Work	413
	References	414
20	Reconfigurable Controllers—A Mechatronic Systems Approach	417
	Roland Kasper and Steffen Toscher	
20.1	Introduction	417
20.2	Design Methodology	421
20.2.1	Logical Controller Structure and Partitioning	421
20.2.2	Specification of Reconfigurable Controller Functionalities	423
20.2.3	Distributed Reconfiguration Control and Activation Strategies	424
20.3	Structure and Implementation	425
20.3.1	Structure and Components	426
20.3.2	Implementation and Target Hardware	428
20.3.3	Partial Reconfiguration Solution	429
20.4	Application	430
20.4.1	A Reconfigurable Controller for Piezo-Electric Actuators	430
20.5	Conclusion	434
	References	435
Index	437

Contributors

Ali Ahmadinia Hardware/Software Co-Design, Department of Computer Science, University of Erlangen-Nuremberg, Erlangen, Germany, A.Ahmadinia@ed.ac.uk

Carsten Albrecht Institute of Computer Engineering, University of Lübeck, Ratzeburger Allee 160, 23538 Lübeck, Germany, albrecht@iti.uni-luebeck.de

Matthias Alles Microelectronic Systems Design Research Group, University of Kaiserslautern, 67663 Kaiserslautern, Germany, alles@eit.uni-kl.de

Josef Angermeier Hardware/Software Co-Design, Department of Computer Science, University of Erlangen-Nuremberg, Erlangen, Germany, angermeier@informatik.uni-erlangen.de

Jürgen Becker Institut für Technik der Informationsverarbeitung (ITIV), University of Karlsruhe, Karlsruhe, Germany, Becker@itiv.uni-karlsruhe.de

Christophe Bobda Department of Computer Science, University of Potsdam, Potsdam, Germany, bobda@cs.uni-potsdam.de

Lars Braun Institut für Technik der Informationsverarbeitung (ITIV), University of Karlsruhe, Karlsruhe, Germany, Braun@itiv.uni-karlsruhe.de

Christian Brehm Microelectronic Systems Design Research Group, University of Kaiserslautern, 67663 Kaiserslautern, Germany, brehm@eit.uni-kl.de

Christopher Claus Institute for Integrated Systems, Technische Universität München, Munich, Germany, Christopher.Claus@tum.de

Thomas Coenen Chair of Electrical Engineering and Computer Systems, RWTH Aachen University, Schinkelstr. 2, 52062 Aachen, Germany, coenen@eecs.rwth-aachen.de

Sven Eisenhardt Department of Computer Engineering, Wilhelm-Schickard-Institute for Computer Science, University of Tübingen, Tübingen, Germany, eisenhardt@informatik.uni-tuebingen.de