Embedded Computing for High Performance

Efficient Mapping of Computations Using Customization, Code Transformations and Compilation

João M.P. Cardoso, José Gabriel F. Coutinho, Pedro C. Diniz

MK
MORGAN KAUFMANN

Embedded Computing for High Performance

Efficient Mapping of Computations Using Customization, Code Transformations and Compilation

João M.P. Cardoso, José Gabriel F. Coutinho, Pedro C. Diniz

Best practices for developing high-performance embedded applications

Embedded Computing for High Performance provides a practical guide through a set of real-life examples for developing and mapping applications for high-performance embedded systems. Working with popular hardware including Xilinx's FPGAs and ARM's processors, the book offers a comprehensive description of techniques for mapping computations expressed in programming languages such as C or MATLAB to high-performance embedded architectures consisting of multiple CPUs, GPUs, and reconfigurable hardware (FPGAs). The authors take advantage of a domain-specific language (LARA) to express executable strategies for code instrumentation and for code transformations. In this way, users can decouple original application code from transformed code and enhance productivity and program portability. After reading this book, engineers will understand processes, methodologies, and best practices for the development of applications for high-performance embedded computing systems.

Key Features:

- Focused on maximizing performance while managing energy consumption in embedded systems
- Explains how to retarget code for heterogeneous systems with GPUs and FPGAs
- Demonstrates the use of a domain-specific language that facilitates migrating and retargeting existing applications to modern systems
- Includes downloadable slides, tools, and tutorials

Authors

João M.P. Cardoso, Full Professor, University of Porto, Porto, Portugal
José Gabriel F. Coutinho, Research Associate, Imperial College, London, United Kingdom
Pedro C. Diniz, Research Associate, University of Southern California's Information Sciences Institute (USC/ISI), California, United States

Computers/System Architecture

MK
MORGAN KAUFMANN PUBLISHERS
AN IMPRINT OF ELSEVIER
elsevier.com/books-and-journals

ISBN 978-0-12-804189-5
9 780128 041895

Embedded Computing for High Performance

Cardoso
Coutinho
Diniz

MK

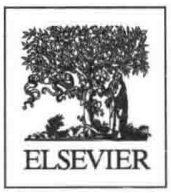
Embedded Computing for High Performance

Efficient Mapping of Computations Using Customization, Code Transformations and Compilation

João M.P. Cardoso

José Gabriel F. Coutinho

Pedro C. Diniz



MK

MORGAN KAUFMANN PUBLISHERS

AN IMPRINT OF ELSEVIER

Morgan Kaufmann is an imprint of Elsevier
50 Hampshire Street, 5th Floor, Cambridge, MA 02139, United States

© 2017 Elsevier Inc. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or any information storage and retrieval system, without permission in writing from the publisher. Details on how to seek permission, further information about the Publisher's permissions policies and our arrangements with organizations such as the Copyright Clearance Center and the Copyright Licensing Agency, can be found at our website: www.elsevier.com/permissions.

This book and the individual contributions contained in it are protected under copyright by the Publisher (other than as may be noted herein).

Notices

Knowledge and best practice in this field are constantly changing. As new research and experience broaden our understanding, changes in research methods, professional practices, or medical treatment may become necessary.

Practitioners and researchers must always rely on their own experience and knowledge in evaluating and using any information, methods, compounds, or experiments described herein. In using such information or methods they should be mindful of their own safety and the safety of others, including parties for whom they have a professional responsibility.

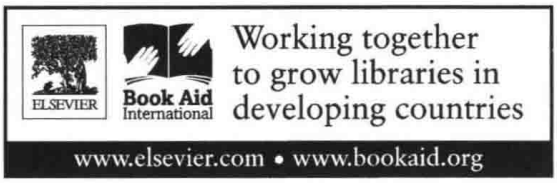
To the fullest extent of the law, neither the Publisher nor the authors, contributors, or editors, assume any liability for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions, or ideas contained in the material herein.

Library of Congress Cataloging-in-Publication Data
A catalog record for this book is available from the Library of Congress

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library

ISBN: 978-0-12-804189-5

For information on all Morgan Kaufmann publications
visit our website at https://www.elsevier.com/books-and-journals



Publisher: Jonathan Simpson
Acquisition Editor: Jonathan Simpson
Editorial Project Manager: Lindsay Lawrence
Production Project Manager: Punithavathy Govindaradjane
Cover Designer: Mark Rogers

Typeset by SPi Global, India

此为试读,需要完整PDF请访问: www.ertongbook.com

Embedded Computing for High Performance

Dedication

We dedicate this book to:

our parents
our families

To Teresa, Rodrigo, Frederico, and Dinis.
To my grandmother Amélia.
To Rafael Nuno, who over the last years has endured so much more than he should have.

About the Authors

João M.P. Cardoso is a full professor at the Department of Informatics Engineering, Faculty of Engineering of the University of Porto, Porto, Portugal and a research member at INESC TEC. Before, he was with the IST/Technical Univ. of Lisbon (UTL) (2006–08), a senior researcher at INESC-ID (2001–09), and with the University of Algarve (1993–2006). In 2001/2002, he worked for PACT XPP Technologies, Inc., Munich, Germany. He received his PhD degree in electrical and computer engineering from IST/Technical University of Lisbon in 2001. He served as a Program Committee member, as General Co-Chair, and as Program Co-Chair in many international conferences. He has (co-)authored over 150 scientific publications on subjects related to compilers, embedded systems, and reconfigurable computing. In addition, he has been involved in several research projects. He is a senior member of IEEE, a member of IEEE Computer Society, and a senior member of ACM. His research interests include compilation techniques, domain-specific languages, reconfigurable computing, application-specific architectures, and high-performance computing with a particular emphasis in embedded computing.

José Gabriel F. Coutinho is an associate researcher working in the Custom Computing Research Group at Imperial College London. He received his M. Eng. degree in computer engineering from Instituto Superior Técnico, Portugal in 1997. In 2000 and 2007 he received his MSc and PhD in computing science from Imperial College London, respectively. Since 2005, he has been involved in United Kingdom and EU research projects, including FP6 hArtes, FP7 REFLECT, FP7 HARNESS, and H2020 EXTRA. In addition, he has published over 50 research papers in peer-referred journals and international conferences and has contributed to four book publications. His research interests include reconfigurable computing, HPC platforms, cloud computing platforms, high-level compilation techniques, programming models, and domain-specific languages.

Pedro C. Diniz received his MS in electrical and computer engineering from the Technical University in Lisbon, Portugal and his PhD in computer science from the University of California, Santa Barbara in 1997. Since 1997 he has been a research associate with the University of Southern California's Information Sciences Institute (USC/ISI) and a research assistant professor of Computer Science at USC in Los Angeles, California. He has participated and/or led various research projects in the area of compilation for high-performance computing, mapping and synthesis for reconfigurable computing architectures, and more recently resilient computing. He has also been heavily involved in the scientific community having participated as part of the technical program committee of over 20 international conferences in the area of high-performance computing, reconfigurable and field-programmable computing.

此为试读,需要完整PDF请访问: www.ertongbook.com

Preface

Over the last decades, computer users have enjoyed the benefits of a seemingly unbounded availability of transistors on a die, with every new microprocessor design exhibiting performance figures that dwarfed previous generations. Computing platforms evolved from a single processor core to general-purpose multicores and specialized cores, such as graphics processing units (GPUs), delivering unprecedented performance thanks to the high degree of parallelism currently available. More recently, energy efficiency has become a major concern, prompting systems to include custom computing engines in the form of field-programmable gate arrays (FPGA) and other forms of reconfigurable computing devices.

All these computing platforms trends are permeating the embedded computing domain, especially in high-performance embedded computing systems. Still, these advanced architectures expose an execution model that is far detached from the traditional sequential programming paradigm that programmers have been accustomed to when developing their extensive code base, and which they rely on when reasoning about program correctness. As a natural consequence of this gap between architectures and high-level programming languages, developers must understand the basic mapping between the application and the target computing architectures to fully exploit their capabilities. To help mitigate the complexity of this mapping and optimization problem, many high-level languages now include language extensions and compiler directives that allow applications to make more effective use of parallel architectures, for instance, to exploit multithreading on multiple cores. Given the stringent requirements of current embedded computing platforms in terms of latency, throughput, power and energy, developers need to further master this mapping process.

This book provides a comprehensive description of the basic mapping techniques and source code transformations for computations expressed in high-level imperative programming languages, such as C or MATLAB, to high-performance embedded architectures consisting of multiple CPUs, GPUs, and reconfigurable hardware (mainly FPGAs). It is therefore meant to help practitioners in the area of electrical, computer engineering, and computer science to effectively map computations to these architectures.

This book also covers existing compilers and their transformations outlining their use in many mapping techniques. These include the classical parallel-oriented transformations for loop constructs, but equally important data-oriented and data-mapping transformations that are key in the context of GPU-based systems. As such, this book is aimed to help computer engineers and computer scientists, as well as electrical engineers, who are faced with the hard task of mapping computations to high-performance embedded computing systems. Given the comprehensive set of source code and retargeting transformations described here, this book can be

effectively used as a textbook for an advanced electrical, computer engineering, and computer science course focused on the development of high-performance embedded systems.

We are very conscious about the difficulty of presenting in a single book, and in a cohesive form, all the topics we consider important about the process of mapping computations to high-performance embedded computing platforms. However, we believe that the topics presented in this book should be mastered by the next generation of developers.

We hope you enjoy reading this book, and that it contributes to increasing your knowledge about developing efficient programs on high-performance embedded platforms, and that it serves as an inspiration to your projects.

João M.P. Cardoso
José Gabriel F. Coutinho
Pedro C. Diniz

Acknowledgments

We would like to acknowledge Walid Najjar, from the University of California Riverside, United States, for reading a previous version of Chapter 2 and for providing important feedback and suggestions for improving it.

We would like to acknowledge all the members of the SPeCS group[1] for their suggestions and discussions, namely, João Bispo, Tiago Carvalho, Pedro Pinto, Luís Reis, and Ricardo Nobre. We are also grateful to all of them for reviewing previous versions of this book's chapters and for their valuable feedback that undoubtedly helped to improve the book.

Students of the PhD Program on Informatics Engineering (ProDEI) of the Faculty of Engineering of the University of Porto (FEUP) have also been a source of helpful feedback regarding some of the contents of this book, as earlier revisions from selected chapters were used as part of the class material for the High-Performance Embedded Computing (CEED) course.

In addition, we would also like to acknowledge the support given by the following companies. Xilinx Inc.[2] (United States) provided, through their University Program, FPGA-based development boards and software licenses including Vivado and Vivado HLS. ARM Ltd.[3] (United Kingdom) provided, through its ARM University Program, a sample ARM Lab-in-a-Box on Efficient Embedded Systems Design and Programming.

João M.P. Cardoso would like to acknowledge the support of the Department of Informatics Engineering of the Faculty of Engineering of the University of Porto, of INESC TEC, and the partial support provided by the following research projects: ANTAREX (H2020 FETHPC-1-2014, ref. 671623), CONTEXTWA (FCT PTDC/EEI-SCR/6945/2014), and TEC4Growth—RL1 SMILES (NORTE-01-0145-FEDER-000020). José Gabriel F. Coutinho would like to acknowledge the support of Wayne Luk, the Department of Computing at Imperial College London, United Kingdom, and the partial support of the EXTRA research project (H2020 FETHPC-1-2014, ref. 671653).

We would like to acknowledge Elsevier for giving us the opportunity to write this book. A warm acknowledgment and appreciation to Lindsay Lawrence, our Elsevier editor, for her belief in this project since the very beginning, as well as her direction which helped us finish this book project.

Last but not least, we would like to thank our families for their support and understanding for the countless hours we had to devote to this book.

[1] SPeCS (Special Purpose Computing Systems, Languages and Tools) Research Group: http://www.fe.up.pt/~specs/.

[2] Xilinx Inc., http://www.xilinx.com.

[3] ARM Ltd., http://www.arm.com/.

Abbreviations

ACPI	advanced configuration and power interface. A standard promoted by Intel, Microsoft, and Toshiba
ADC	analog-to-digital converter
AMD	advanced micro devices
AOP	aspect-oriented programming
API	application programming interface
ARM	advanced RISC machines
ASIP	application-specific instruction-set processor
AST	abstract syntax tree
AVX	advanced vector extensions
BRAM	block RAM
CD	computing device
CDFG	control/data flow graph
CFG	control flow graph
CG	call graph
CGRA	coarse-grained reconfigurable array
CISC	complex instruction set computer
CLB	configurable logic block
CMP	chip multiprocessor
COTS	commercial off-the-shelf
CPA	critical path analysis
CPU	central processing unit
CU	computing unit
DAC	digital-to-analog converter
DAG	directed acyclic graph
DDG	data dependence graph
DDR	double data rate
DFG	data flow graph
DFS	dynamic frequency scaling
DPM	dynamic power management
DRAM	dynamic random-access memory (RAM)
DSE	design space exploration
DSL	domain-specific language
DSP	digital signal processing
DVFS	dynamic voltage and frequency scaling
DVS	dynamic voltage scaling
EDA	electronic design automation
EDP	energy delay product
EEMBC	embedded microprocessor benchmark consortium
FMA	fused multiply-add

FPGA	field-programmable gate array
FPS	frames per second
FSM	finite state machine
GA	genetic algorithm
GCC	GNU compiler collection (originally named GNU C Compiler)
GPGPU	general-purpose graphics processing unit (also known as general-purpose computing on graphics processing unit)
GPIO	general-purpose input/output (IO)
GPU	graphics processing unit
HLS	high-level synthesis
HPC	high-performance computing
HPEC	high-performance embedded computing
HPF	high performance Fortran
ICC	Intel C/C++ compiler
IDE	integrated design environment
ILP	instruction-level parallelism or integer-linear programming
IO	input/output
IOB	input/output block
IR	intermediate representation
ISA	instruction set architecture
LDG	loop dependence graph
LLVM	low level virtual machine
LOC	lines of code
MIC	many integrated core
MPI	message passing interface
MPSoC	multiprocessor SoC (system-on-a-chip)
NFRs	nonfunctional requirements
NUMA	nonuniform memory access
OpenACC	open accelerators
OpenCL	open computing language
OpenMP	open multiprocessing
PC	personal computer
PCI	peripheral component interconnect
PCIe	peripheral component interconnect express
PE	processing element
QoE	quality of experience
QoS	quality of service
QPI	Intel QuickPath interconnect
RAM	random-access memory
RISC	reduced-instruction set computer
ROM	read-only memory
RTOS	real-time operating system
SA	simulated annealing

此为试读,需要完整PDF请访问: www.ertongbook.com