

High-Performance and Time-Predictable Embedded Computing

Luís Miguel Pinho, Eduardo Quiñones,
Marko Bertogna, Andrea Marongiu, Vincent Nélis,
Paolo Gai and Juan Sancho (Editors)



River Publishers

**High-Performance and
Time-Predictable
Embedded Computing**

RIVER PUBLISHERS SERIES IN INFORMATION SCIENCE AND TECHNOLOGY

Series Editors

K. C. CHEN

National Taiwan University

Taipei, Taiwan

and

University of South Florida, USA

SANDEEP SHUKLA

Virginia Tech

USA

and

Indian Institute of Technology Kanpur, India

Indexing: All books published in this series are submitted to the Web of Science Book Citation Index (BkCI), to CrossRef and to Google Scholar.

The “River Publishers Series in Information Science and Technology” covers research which ushers the 21st Century into an Internet and multimedia era. Multimedia means the theory and application of filtering, coding, estimating, analyzing, detecting and recognizing, synthesizing, classifying, recording, and reproducing signals by digital and/or analog devices or techniques, while the scope of “signal” includes audio, video, speech, image, musical, multimedia, data/content, geophysical, sonar/radar, bio/medical, sensation, etc. Networking suggests transportation of such multimedia contents among nodes in communication and/or computer networks, to facilitate the ultimate Internet.

Theory, technologies, protocols and standards, applications/services, practice and implementation of wired/wireless networking are all within the scope of this series. Based on network and communication science, we further extend the scope for 21st Century life through the knowledge in robotics, machine learning, embedded systems, cognitive science, pattern recognition, quantum/biological/molecular computation and information processing, biology, ecology, social science and economics, user behaviors and interface, and applications to health and society advance.

Books published in the series include research monographs, edited volumes, handbooks and textbooks. The books provide professionals, researchers, educators, and advanced students in the field with an invaluable insight into the latest research and developments.

Topics covered in the series include, but are by no means restricted to the following:

- Communication/Computer Networking Technologies and Applications
- Queuing Theory
- Optimization
- Operation Research
- Stochastic Processes
- Information Theory
- Multimedia/Speech/Video Processing
- Computation and Information Processing
- Machine Intelligence
- Cognitive Science and Brian Science
- Embedded Systems
- Computer Architectures
- Reconfigurable Computing
- Cyber Security

For a list of other books in this series, visit www.riverpublishers.com

High-Performance and Time-Predictable Embedded Computing

Editors

Luís Miguel Pinho

CISTER Research Centre, Polytechnic Institute of Porto, Portugal

Eduardo Quiñones

Barcelona Supercomputing Center, Spain

Marko Bertogna

University of Modena and Reggio Emilia, Italy

Andrea Marongiu

Swiss Federal Institute of Technology Zurich, Switzerland

Vincent Nélis

CISTER Research Centre, Polytechnic Institute of Porto, Portugal

Paolo Gai

Evidence Srl, Italy

Juan Sancho

ATOS, Spain



River Publishers

 **Routledge**
Taylor & Francis Group
LONDON AND NEW YORK



Published 2018 by River Publishers
River Publishers
Alsbjergvej 10, 9260 Gistrup, Denmark
www.riverpublishers.com

Distributed exclusively by Routledge
4 Park Square, Milton Park, Abingdon, Oxon OX14 4RN
605 Third Avenue, New York, NY 10017, USA

*High-Performance and Time-Predictable Embedded Computing /*by Luis Miguel Pinho, Eduardo Quinones, Marko Bertogna, Andrea Marongiu, Vincent Nelis, Paolo Gai, Juan Sancho.

© The Editor(s) (if applicable) and The Author(s) 2018. This book is published open access.

Open Access

This book is distributed under the terms of the Creative Commons Attribution-Non-Commercial 4.0 International License, CC-BY-NC 4.0 (<http://creativecommons.org/licenses/by/4.0/>), which permits use, duplication, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, a link is provided to the Creative Commons license and any changes made are indicated. The images or other third party material in this book are included in the work's Creative Commons license, unless indicated otherwise in the credit line; if such material is not included in the work's Creative Commons license and the respective action is not permitted by statutory regulation, users will need to obtain permission from the license holder to duplicate, adapt, or reproduce the material.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper.

Routledge is an imprint of the Taylor & Francis Group, an informa business

ISBN 978-87-93609-69-3 (print)

While every effort is made to provide dependable information, the publisher, authors, and editors cannot be held responsible for any errors or omissions.

Contents

Preface	xiii
List of Contributors	xv
List of Figures	xvii
List of Tables	xxi
List of Abbreviations	xxiii
1 Introduction	1
<i>Luís Miguel Pinho, Eduardo Quiñones, Marko Bertogna, Andrea Marongiu, Vincent Nélis, Paolo Gai and Juan Sancho</i>	
1.1 Introduction	1
1.1.1 The Convergence of High-performance and Embedded Computing Domains	3
1.1.2 Parallelization Challenge	5
1.2 The P-SOCRATES Project	6
1.3 Challenges Addressed in This Book	8
1.3.1 Compiler Analysis of Parallel Programs	8
1.3.2 Predictable Scheduling of Parallel Tasks on Many-core Systems	9
1.3.3 Methodology for Measurement-based Timing Analysis	9
1.3.4 Optimized OpenMP Tasking Runtime System	9
1.3.5 Real-time Operating Systems	10
1.4 The UpScale SDK	10
1.5 Summary	11
References	12

2 Manycore Platforms	15
<i>Andrea Marongiu, Vincent Nélis and Patrick Meumeu Yomsi</i>	
2.1 Introduction	15
2.2 Manycore Architectures	17
2.2.1 Xeon Phi	17
2.2.2 Pezy SC	18
2.2.3 NVIDIA Tegra X1	19
2.2.4 Tilera Tile	21
2.2.5 STMicroelectronics STHORM	22
2.2.6 Epiphany-V	23
2.2.7 TI Keystone II	24
2.2.8 Kalray MPPA-256	25
2.2.8.1 The I/O subsystem	26
2.2.8.2 The Network-on-Chip (NoC)	26
2.2.8.3 The Host-to-IOS communication protocol	28
2.2.8.4 Internal architecture of the compute clusters	28
2.2.8.5 The shared memory	29
2.3 Summary	30
References	31
3 Predictable Parallel Programming with OpenMP	33
<i>Maria A. Serrano, Sara Royuela, Andrea Marongiu and Eduardo Quiñones</i>	
3.1 Introduction	33
3.1.1 Introduction to Parallel Programming Models	34
3.1.1.1 POSIX threads	35
3.1.1.2 OpenCL™	35
3.1.1.3 NVIDIA® CUDA	36
3.1.1.4 Intel® Cilk™ Plus	36
3.1.1.5 Intel® TBB	36
3.1.1.6 OpenMP	37
3.2 The OpenMP Parallel Programming Model	37
3.2.1 Introduction and Evolution of OpenMP	37
3.2.2 Parallel Model of OpenMP	39
3.2.2.1 Execution model	39
3.2.2.2 Acceleration model	40
3.2.2.3 Memory model	41

3.2.3	An OpenMP Example	42
3.3	Timing Properties of OpenMP Tasking Model	43
3.3.1	Sporadic DAG Scheduling Model of Parallel Applications	43
3.3.2	Understanding the OpenMP Tasking Model	44
3.3.3	OpenMP and Timing Predictability	46
3.3.3.1	Extracting the DAG of an OpenMP program	47
3.3.3.2	WCET analysis is applied to <i>tasks</i> and <i>tasks parts</i>	48
3.3.3.3	DAG-based scheduling must not violate the TSCs	49
3.4	Extracting the Timing Information of an OpenMP Program	51
3.4.1	Parallel Structure Stage	52
3.4.1.1	Parallel control flow analysis	53
3.4.1.2	Induction variables analysis	53
3.4.1.3	Reaching definitions and range analysis	53
3.4.1.4	Putting all together: The wave-front example	53
3.4.2	Task Expansion Stage	54
3.4.2.1	Control flow expansion and synchronization predicate resolution	54
3.4.2.2	t_{id} : A unique task instance identifier	56
3.4.2.3	Missing information when deriving the DAG	57
3.4.3	Compiler Complexity	58
3.5	Summary	58
	References	59
4	Mapping, Scheduling, and Schedulability Analysis	63
	<i>Paolo Burgio, Marko Bertogna, Alessandra Melani, Eduardo Quiñones and María A. Serrano</i>	
4.1	Introduction	63
4.2	System Model	64
4.3	Partitioned Scheduler	66
4.3.1	The Optimality of EDF on Preemptive Uniprocessors	66
4.3.2	FP-scheduling Algorithms	67
4.3.3	Limited Preemption Scheduling	68

4.3.4	Limited Preemption Schedulability Analysis	69
4.4	Global Scheduler with Migration Support	70
4.4.1	Migration-based Scheduler	70
4.4.2	Putting All Together	72
4.4.3	Implementation of a Limited Preemption Scheduler	73
4.5	Overall Schedulability Analysis	75
4.5.1	Model Formalization	75
4.5.2	Critical Interference of cp-tasks	78
4.5.3	Response Time Analysis	80
4.5.3.1	Inter-task interference	80
4.5.3.2	Intra-task interference	82
4.5.3.3	Computation of cp-task parameters	84
4.5.4	Non-conditional DAG Tasks	86
4.5.5	Series-Parallel Conditional DAG Tasks	86
4.5.6	Schedulability Condition	86
4.6	Specializing Analysis for Limited Pre-emption Global/ Dynamic Approach	87
4.6.1	Blocking Impact of the Largest NPRs (LP-max)	88
4.6.2	Blocking Impact of the Largest Parallel NPRs (LP-ILP)	88
4.6.2.1	LP worst-case workload of a task executing on c cores	89
4.6.2.2	Overall LP worst-case workload	90
4.6.2.3	Lower-priority interference	91
4.6.3	Computation of Response Time Factors of LP-ILP	92
4.6.3.1	Worst-case workload of τ_i executing on c cores: $\mu_i[c]$	92
4.6.3.2	Overall LP worst-case workload of lp(k) per execution scenario s_l : $\rho_k[s_l]$	94
4.6.4	Complexity	95
4.7	Specializing Analysis for the Partitioned/Static Approach	96
4.7.1	ILP Formulation	96
4.7.1.1	Tied tasks	97
4.7.1.2	Untied tasks	99
4.7.1.3	Complexity	100
4.7.2	Heuristic Approaches	100
4.7.2.1	Tied tasks	101

4.7.2.2	Untied tasks	103
4.7.3	Integrating Interference from Additional RT Tasks	103
4.7.4	Critical Instant	104
4.7.5	Response-time Upper Bound	105
4.8	Scheduling for I/O Cores	107
4.9	Summary	107
	References	109
5	Timing Analysis Methodology	113
<i>Vincent Nélis, Patrick Meumeu Yomsi and Luís Miguel Pinho</i>		
5.1	Introduction	113
5.1.1	Static WCET Analysis Techniques	115
5.1.2	Measurement-based WCET Analysis Techniques	118
5.1.3	Hybrid WCET Techniques	119
5.1.4	Measurement-based Probabilistic Techniques	120
5.2	Our Choice of Methodology for WCET Estimation	121
5.2.1	Why Not Use Static Approaches?	122
5.2.2	Why Use Measurement-based Techniques?	124
5.3	Description of Our Timing Analysis Methodology	127
5.3.1	Intrinsic vs. Extrinsic Execution Times	127
5.3.2	The Concept of Safety Margins	128
5.3.3	Our Proposed Timing Methodology at a Glance	130
5.3.4	Overview of the Application Structure	131
5.3.5	Automatic Insertion and Removal of the Trace-points	133
5.3.5.1	How to insert the trace-points	133
5.3.5.2	How to remove the trace-points	135
5.3.6	Extract the Intrinsic Execution Time: The Isolation Mode	136
5.3.7	Extract the Extrinsic Execution Time: The Contention Mode	137
5.3.8	Extract the Execution Time in Real Situation: The Deployment Mode	141
5.3.9	Derive WCET Estimates	141
5.4	Summary	143
	References	143