Time-Predictable Embedded Software on Multi-Core Platforms: Analysis and Optimization

Sudipta Chattopadhyay
Linköping University

Abhik Roychoudhury
National University of Singapore

Jakob Rosén
Linköping University

Petru Eles
Linköping University

Zebo Peng
Linköping University
 Foundations and Trends® in Electronic Design Automation

Published, sold and distributed by:
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
United States
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

Outside North America:
now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

The preferred citation for this publication is

This Foundations and Trends® issue was typeset in $\LaTeX$ using a class file designed by Neal Parikh. Printed on acid-free paper.

© 2014 S. Chattopadhyay, A. Roychoudhury, J. Rosén, P. Eles, Z. Peng

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, mechanical, photocopying, recording or otherwise, without prior written permission of the publishers.

Photocopying. In the USA: This journal is registered at the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923. Authorization to photocopy items for internal or personal use, or the internal or personal use of specific clients, is granted by now Publishers Inc for users registered with the Copyright Clearance Center (CCC). The ‘services’ for users can be found on the internet at: www.copyright.com

For those organizations that have been granted a photocopy license, a separate system of payment has been arranged. Authorization does not extend to other kinds of copying, such as that for general distribution, for advertising or promotional purposes, for creating new collective works, or for resale. In the rest of the world: Permission to photocopy must be obtained from the copyright owner. Please apply to now Publishers Inc., PO Box 1024, Hanover, MA 02339, USA; Tel. +1 781 871 0245; www.nowpublishers.com; sales@nowpublishers.com

now Publishers Inc. has an exclusive license to publish this material worldwide. Permission to use this content must be obtained from the copyright license holder. Please apply to now Publishers, PO Box 179, 2600 AD Delft, The Netherlands, www.nowpublishers.com; e-mail: sales@nowpublishers.com
Editorial Scope

Topics

Foundations and Trends® in Electronic Design Automation publishes survey and tutorial articles in the following topics:

- System level design
- Behavioral synthesis
- Logic design
- Verification
- Test
- Physical design
- Circuit level design
- Reconfigurable systems
- Analog design
- Embedded software and parallel programming
- Multicore, GPU, FPGA, and heterogeneous systems
- Distributed, networked embedded systems
- Real-time and cyberphysical systems

Information for Librarians

Foundations and Trends® in Electronic Design Automation, 2014, Volume 8, 4 issues. ISSN paper version 1551-3939. ISSN online version 1551-3947. Also available as a combined paper and online subscription.
Time-Predictable Embedded Software on Multi-Core Platforms: Analysis and Optimization

Sudipta Chattopadhyay
Linköping University
sudipta.chattopadhyay@liu.se

Abhik Roychoudhury
National University of Singapore
abhik@comp.nus.edu.sg

Jakob Rosén
Linköping University
jakob.rosen@gmail.com

Petru Eles
Linköping University
petru.eles@liu.se

Zebo Peng
Linköping University
zebo.peng@liu.se
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>1</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>2</td>
</tr>
<tr>
<td>2 WCET analysis and multi-core platforms</td>
<td>6</td>
</tr>
<tr>
<td>2.1 A background on WCET analysis</td>
<td>6</td>
</tr>
<tr>
<td>2.2 Challenges in WCET analysis for multi-core architectures</td>
<td>16</td>
</tr>
<tr>
<td>3 WCET analysis for multi-core platforms</td>
<td>19</td>
</tr>
<tr>
<td>3.1 Modeling shared caches</td>
<td>20</td>
</tr>
<tr>
<td>3.2 Modeling shared buses</td>
<td>38</td>
</tr>
<tr>
<td>3.3 Modeling timing interactions</td>
<td>61</td>
</tr>
<tr>
<td>3.4 Discussion about analysis complexity</td>
<td>86</td>
</tr>
<tr>
<td>3.5 Experimental evaluation</td>
<td>89</td>
</tr>
<tr>
<td>3.6 Data caches and branch target buffers</td>
<td>103</td>
</tr>
<tr>
<td>3.7 A survey of related techniques</td>
<td>105</td>
</tr>
<tr>
<td>4 WCET optimization for multi-core platforms</td>
<td>107</td>
</tr>
<tr>
<td>4.1 Optimization of worst-case response time</td>
<td>107</td>
</tr>
<tr>
<td>4.2 WCRT optimization approach</td>
<td>108</td>
</tr>
<tr>
<td>4.3 Cost function</td>
<td>110</td>
</tr>
<tr>
<td>4.4 Optimization algorithm</td>
<td>112</td>
</tr>
</tbody>
</table>
4.5  Simplified algorithm ................................. 121
4.6  Memory consumption ................................. 122
4.7  Experimental results ................................. 123
4.8  A survey of related techniques...................... 129

5  Time-predictable multi-core architecture 132
  5.1  Resource isolation ................................. 132
  5.2  Usage of software controlled memory .......... 135
  5.3  Extension of instruction set architecture (ISA) 138

6  Discussion and future work 140
  6.1  Summary of recent development .................... 140
  6.2  Limitations imposed by current approaches .... 141
  6.3  Other limitations .................................. 142
  6.4  Analysis pessimism ................................ 143
  6.5  Research challenges in future .................... 144

7  Conclusions 148

Acknowledgements 149

References 150
Abstract

Multi-core architectures have recently gained popularity due to their high-performance and low-power characteristics. Most of the modern desktop systems are now equipped with multi-core processors. Despite the wide-spread adaptation of multi-core processors in desktop systems, using such processors in embedded systems still poses several challenges. Embedded systems are often constrained by several extra-functional aspects, such as time. Therefore, providing guarantees for time-predictable execution is one of the key requirements for embedded system designers. Multi-core processors adversely affect the time-predictability due to the presence of shared resources, such as shared caches and shared buses. In this contribution, we shall first discuss the challenges imposed by multi-core architectures in designing time-predictable embedded systems. Subsequently, we shall describe, in details, a comprehensive solution to guarantee time-predictable execution on multi-core platforms. Besides, we shall also perform a discussion of different techniques to provide an overview of the state-of-the-art solutions in this topic. Through this work, we aim to provide a solid background on recent trends of research towards achieving time-predictability on multi-cores. Besides, we also highlight the limitations of the state-of-the-art and discuss future research opportunities and challenges to accomplish time-predictable execution on multi-core platforms.

DOI: 10.1561/1000000037.
Real-time, embedded systems often need to satisfy several extra-functional constraints, such as timing. In particular, for hard real-time systems, such timing constraints are strictly enforced. Violation of these timing constraints may have serious consequences, potentially costing human lives. Therefore, static timing-analysis of hard real-time systems has emerged to be a critical problem to solve.

In general, a real-time, embedded application is made of several components, usually called tasks. Therefore, timing analysis of embedded software is typically performed in two separate phases: (i) a low-level analysis which derives the worst case execution time (WCET) and best case execution time (BCET) of individual tasks, and (ii) a system-level schedulability analysis which uses the WCET/BCET derived for each task and computes the overall timing characteristics of the application. In this monograph, we shall primarily focus our discussion on low-level WCET analysis.

WCET analysis of an embedded software is typically performed in three stages: (i) a flow-analysis using the control flow graph (CFG) of the program (to determine infeasible paths and loop bounds), (ii) micro-architectural modeling (to determine the worst case execution time of each basic block in the CFG) and (iii) a calculation phase which combines the outcome of
flow-analysis and micro-architectural modeling to derive the worst case execution time (WCET) of the entire program. Micro-architectural modeling systematically considers the timing effects of underlying processor features, such as pipeline, caches, branch prediction and so on. For single-core processors, such a micro-architectural modeling involves the analysis of a single program occupying the processor. However, this criterion no longer holds with multi-core processors. Since their inception, multi-core processors have widely been adopted due to their high-performance and low-power characteristics. Unfortunately, multi-core processors pose some significant challenges in terms of time-predictability. Basically, these challenges arise due to the presence of shared resources, such as shared caches and shared buses [5]. The presence of shared resources makes the WCET analysis significantly more complex than the WCET analysis on single-core processors. In particular, micro-architectural modeling is affected due to the presence of inter-core interferences, such as shared cache conflicts or bus contention. Through this monograph, we primarily aim to highlight the recent advances to address such challenges.

As mentioned in the preceding paragraph, shared resources are the key bottlenecks to build time-predictable embedded software on multi-core platforms. The content of a shared cache is modified by several programs running in parallel on different cores. Therefore, the modeling of inter-core cache conflicts is important to estimate the shared-cache latency accurately. For bus-based systems, shared buses introduce variable access latency to the shared resources (e.g. shared caches and main memory). Such a variable access latency highly depends on the bus contention, which in turn depends on the amount of memory traffic generated by different cores. In this monograph, we shall first describe an approach to model the timing behavior of shared caches [21]. Such a modeling systematically combines abstract interpretation with state-of-the-art program-verification techniques (e.g. model checking and symbolic execution). In particular, such an approach leverages both the scalability offered by abstract interpretation and the accuracy offered by program-verification methods to build a tight modeling of shared caches. We then describe works on analyzing timing behavior for static bus-arbitration policies, such as time division multiple access (TDMA). Even with static bus-arbitration policies, an accurate analysis of shared-bus delay is complex. This
is due to the reason that bus delay highly depends on the context, such as individual loop iterations and procedure calls. In the worst-case, each loop iteration may experience different bus delay. We describe works [69, 9, 22] in this direction whose requirements range from full-fledged loop unrolling to avoiding loop unrolling altogether, depending on the analysis accuracy.

Subsequently, we discuss the development of a full-fledged WCET analysis framework by combining the modeling of shared resources [18]. Such a combination is non-trivial due to the possible presence of timing anomalies [59]. In the presence of timing anomalies, a local worst-case (e.g. a cache miss or maximum bus delay) may not lead to the overall WCET of a program. As a result, it is unsound to model the timing behavior of each micro-architectural component and get the overall timing behavior by a simple composition of individual timing models. This framework systematically models the timing interaction of shared resources with the rest of the micro-architectural features (e.g. pipeline, branch prediction) and it does not assume a timing-anomaly-free architecture. The WCET analysis framework is built on top of Chronos [52], a freely-available, open-source WCET analysis tool. We show the evaluation of this analysis framework via several experiments.

Besides modeling individual micro-architectural features in multi-core processors, predictability of embedded software can also benefit from customized compiler optimizations and time-predictable multi-core hardware. In this direction, we discuss an optimization of bus schedules to improve time-predictability. Specifically, we describe the generation of customized bus schedules that may greatly improve the WCET of a program [69]. Finally, we discuss several designs of time-predictable hardware to reduce the pessimism in the WCET analysis on multi-core platforms.

The main purpose of this monograph is to give the readers a thorough background on time-predictability for multi-core platforms. Therefore, we have also performed a discussion of research activities by several research groups in this area. This discussion provides a comprehensive overview of the state-of-the-art solutions in the respective topic. In particular, our discussion reveals that the area is fast evolving and there is an active interest by real-time research groups on the topic discussed in this monograph. Finally, in the concluding section of this monograph, we have highlighted a set of open challenges in achieving high-performance and time-predictable
embedded software on multi-core platforms. We hope that this monograph will provide a foundation of building time-predictable software on multi-core platforms and it will help the research community to address the existing challenges in this area.
References

References


Full text available at: http://dx.doi.org/10.1561/1000000037


Full text available at: http://dx.doi.org/10.1561/1000000037
References


References


References


References


