Lecture Notes in Computer Science 9999

More information about this series at http://www.springer.com/series/7407

Petro Lutsyk · Jonas Oberhauser · Wolfgang J. Paul

A Pipelined Multi-Core Machine with Operating System Support

Hardware Implementation and Correctness Proof

Springer

Authors
Petro Lutsyk
Saarland University
Saarbrücken, Germany

Jonas Oberhauser
Saarland University
Saarbrücken, Germany

Wolfgang J. Paul
Saarland University
Saarbrücken, Germany

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-030-43242-3 ISBN 978-3-030-43243-0 (eBook)
https://doi.org/10.1007/978-3-030-43243-0

LNCS Sublibrary: SL1 – Theoretical Computer Science and General Issues

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

Building on previous work, we present at the gate level construction and correctness proof of a multi-core machine with pipelined processors and extensive operating system support. More precisely, the machine under consideration has the following features:

- MIPS instruction set architecture (ISA) for application and for system programming
- cache coherent memory system
- store buffers in front of the data caches
- interrupts and exceptions
- memory management units (MMUs)
- pipelined processors: the classical five-stage pipeline is extended by two pipeline stages for address translation
- local interrupt controller (ICs) supporting inter-processor interrupts (IPIs)
- I/O-interrupt controller and a disk

An overall correctness proof for such a machine poses several challenges, which are overcome with the following technical contributions:

- a new pipeline control which allows rollbacks of instructions from multiple pipeline stages while allowing to complete memory accesses that have already started
- a new induction scheme which permits to treat all rollbacks in such a pipeline in a uniformed way
- modeling of ISA by a nondeterministic transition system with guard conditions
- proof technology for exploiting software conditions in such a machine, which only have to hold if the ISA computation abstracted from the hardware computation obeys the guard conditions

February 2020

Petro Lutsyk
Jonas Oberhauser
Wolfgang J. Paul

Contents