From Variability Tolerance to Approximate Computing in Parallel Integrated Architectures and Accelerators

Abbas Rahimi · Luca Benini
Rajesh K. Gupta

From Variability Tolerance to Approximate Computing in Parallel Integrated Architectures and Accelerators

Springer

Abbas Rahimi
Department of Electrical Engineering
and Computer Sciences
University of California Berkeley
Berkeley, CA
USA

Luca Benini
Integrated Systems Laboratory
ETH Zurich
Zürich
Switzerland

Rajesh K. Gupta
Department of Computer Science
and Engineering
University of California, San Diego
La Jolla, CA
USA

ISBN 978-3-319-53767-2 ISBN 978-3-319-53768-9 (eBook)
DOI 10.1007/978-3-319-53768-9

Library of Congress Control Number: 2017932004

Printed on acid-free paper

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

To my wife with everlasting love and gratitude

–Abbas Rahimi

Foreword

There is no question that computing has dramatically changed society, and has furthered humanity in ways that were hard to foresee at its onset. Many factors have contributed to its unfettered success including the adoption of Boolean logic and algorithmic thinking, the invention of the instruction set machine, and the advent of the semiconductor technology. The latter offered us a semi-perfect switch device and effective ways of storing data. All these factors have led to an amazing run of almost 7 decades. Over time, the quest for ever higher performance in the presence of power and energy limitations have forced us to make major changes to how the processors were architected and operated—such as the introduction of concurrency, the adoption of co-processors and accelerators, or the adoption of ever more complex memory hierarchies. However, in essence the fundamentals remained unchanged

For a number of reasons, this model is at the verge of undergoing some major changes and challenges. On the one hand, the scaling model of semiconductors—commonly known as Moore's law—is running out of steam, hence depriving us from a convenient means in improving computational performance, density and efficiency. On the other hand, the nature of computation itself is changing with data rather than algorithm taking primacy. Both these trends force us to reflect on some of the foundational concepts that have driven computation for such a long period. In an abundance of data, statistical distributions become more relevant than deterministic answers. Many perceptual tasks related to human-world interaction fall under the same class. Learning-based programming approaches are gaining rapid interest and influence. Simultaneously the lack of "the perfect switch", as well as the high-variability of nanoscale devices operating under high energy-efficiency (that is, low-voltage) makes deterministic computing an extremely expensive if at all possible undertaking.

All of this has made researchers explore novel computational models that are "approximate" in nature. This means that errors and approximations are becoming acceptable as long as the outcomes have a well-defined statistical behavior. A number of approaches have been identified and are being actively pursued under

different headers such as approximate computing, statistical computing and stochastic processing. In this book, the authors investigate how errors caused by variation (especially those caused by timing) can be exposed to the software layers, and how they can be mitigated using a range of techniques and methods to reduce their impact. The document provides a clear insight of what is possible through pure software intervention.

This book is at the forefront of what is to come at the frontiers in the new age of computation. As such, I heartily recommend it as a great intellectual effort and a superb read.

Jan M. Rabaey

Preface

Variation in performance and power across manufactured parts and their operating conditions is an accepted reality in modern microelectronic manufacturing processes with geometries in nanometer scales. This book views such variations both as a challenge as well an opportunity to rearchitect the hardware/software interface that provides more resilient system architectures. We start with an examination of how variability manifests itself across various levels of microelectronic systems. We examine various mechanisms designers use, and can use, to combat negative effects of variability.

This book attempts a comprehensive look at the entire software/hardware stack and system architecture in order to devise effective strategies to address microelectronic variability. First, we review the key concepts on *timing errors* caused by various variability sources. We use a two-pronged strategy to mitigate such errors by jointly exposing hardware variations to the software and by exploiting flexibility made possible by parallel processing. We consider methods to predict and prevent, detect and correct, and finally conditions under which such errors can be accepted. For each of these methods, our work spans defining and measuring the notion of error tolerance at various levels, from instructions to procedures to parallel programs. These measures essentially capture the likelihood of errors and associated cost of error correction at different levels. The result is a design platform that enables us to combine these methods that enable detection and correction of erroneous results within a defined criterion for acceptable errors using a notion of memoization across the hardware/software interface. Pursuing this strategy, we develop a set of software techniques and microarchitecture optimizations for improving cost and scale of these methods in massively parallel computing units, such as general-purpose graphics processing units (GP-GPUs), clustered many-core architectures, and field-programmable gate array (FPGA) accelerators.

Our results show that parallel architectures and use of parallelism in general provides the best means to combat and exploit variability. Using such programmable parallel accelerator architectures, we show how system designers can

coordinate propagation of error information and its effects along with new techniques for memoization and memristive associative memory. This book naturally leads to use of these techniques into emerging area of approximate computing, and how these can be used in building resilient and efficient computing systems.

Berkeley, USA — Abbas Rahimi
Zürich, Switzerland — Luca Benini
San Diego, USA — Rajesh K. Gupta
January 2017

Contents