Arun Chandrasekharan · Daniel Große
Rolf Drechsler

# Design Automation Techniques for Approximation Circuits

## Verification, Synthesis and Test

Springer

# Design Automation Techniques for Approximation Circuits

Arun Chandrasekharan • Daniel Große
Rolf Drechsler

# Design Automation Techniques for Approximation Circuits

## Verification, Synthesis and Test

Springer

Arun Chandrasekharan
OneSpin Solutions GmbH
Munich, Germany

Daniel Große
University of Bremen and DFKI GmbH
Bremen, Germany

Rolf Drechsler
University of Bremen and DFKI GmbH
Bremen, Germany

ISBN 978-3-319-98964-8 ISBN 978-3-319-98965-5 (eBook)
https://doi.org/10.1007/978-3-319-98965-5

Library of Congress Control Number: 2018952911



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

*To Keerthana,*
*Nanno*
*and*
*Zehra*

# Preface

APPROXIMATE COMPUTING is a novel design paradigm to address the performance and energy efficiency needed for future computing systems. It is based on the observation that many applications compute their results more accurately than needed, wasting precious computational resources. Compounded to this problem, dark silicon and device scaling limits in the hardware design severely undermine the growing demand for computational power. Approximate computing tackles this by deliberately introducing controlled inaccuracies in the hardware and software to improve performance. There is a huge set of applications from multi-media, data analytics, deep learning, etc. that can make a significant difference in performance and energy efficiency using approximate computing. However, despite its potential, this novel computational paradigm is in its infancy. This is due to the lack of efficient design automation techniques that are synergetic to approximate computing. Our book bridges this gap. We explain algorithms and methodologies from automated synthesis to verification and test of an approximate computing system. All the algorithms explained in this book are implemented and thoroughly evaluated on a wide range of benchmarks and use cases. Our methodologies are efficient, scalable, and significantly advance the state of the art of the approximate system design.

# Acknowledgments

First and foremost, we would like to thank the members of the research group for computer architecture at the University of Bremen. We deeply appreciate the continuous support, the inspiring discussions, and the stimulating environment provided. Next, we would like to thank all coauthors of the papers which formed the starting point for this book: Mathias Soeken, Ulrich Kühne, and Stephan Eggersglüß. This book would not have been possible without their academic knowledge and valuable insight. Our sincere thanks also go to Kenneth Schmitz, Saman Fröhlich, and Arighna Deb for numerous discussions and successful collaborations.

Munich, Germany — Arun Chandrasekharan
Bremen, Germany — Daniel Große
Bremen, Germany — Rolf Drechsler
July 2018

# Contents

# List of Algorithms

# List of Figures

# List of Tables

# Chapter 1
# Introduction

APPROXIMATE COMPUTING is an emerging design paradigm to address the performance and energy efficiency needed for the future computing systems. Conventional strategies to improve the hardware performance such as device scaling have already reached its limits. Current device technologies such as 10 nm are already reported to have significant secondary effects such as quantum tunneling. On the energy front, dark silicon and the power density is a serious challenge and limiting factor for several contemporary IC design flows. It is imperative that the current state-of-the-art techniques are inadequate to meet the growing demands of the computational power. Approximate computing can potentially address these challenges. It refers to hardware and software techniques where the implementation is allowed to differ from the specification, but within an acceptable range. The approximate computing paradigm delivers *performance* at the cost of *accuracy*. The key idea is to trade off correct computations against energy or performance. At a first glance, one might think that this approach is not a good idea. But it has become evident that there is a huge set of applications which can tolerate errors. Applications such as multi-media processing and compressing, voice recognition, web search, or deep learning are just a few examples. However, despite its huge potential, approximate computing is not a mainstream technology yet. This is due to the lack of reliable and efficient design automation techniques for the design and implementation of an approximate computing system. This book bridges this gap. Our work addresses the important facets of approximate computing hardware design—from formal verification and error guarantees to synthesis and test of approximation systems. We provide algorithms and methodologies based on classical formal verification, synthesis, and test techniques for an approximate computing IC design flow. Further, towards the end, a novel hardware architecture is presented for cross-layer approximate computing. Based on the contributions, we advance the current state-of-the-art of the approximate hardware design.

Several applications spend a huge amount of energy to guarantee correctness. However, correctness is not always required due to some inherent characteristics



A. Chandrasekharan et al., *Design Automation Techniques for Approximation Circuits*, https://doi.org/10.1007/978-3-319-98965-5_1