Automated Design of Approximate Accelerators

Jorge Castro-Godínez*
Chair for Embedded Systems (CES)
Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany
jorge.castro-godinez@kit.edu

***Abstract*—Approximate computing has emerged as a design paradigm suitable for applications with inherent error resilience. This paradigm aims to reduce the computing costs of exact calculations by lowering the accuracy of their results. In the last decade, many approximate circuits, particularly approximate adders and multipliers, have been reported in the literature. For an ongoing number of such approximate circuits, selecting those that minimize the required resources for designing and generating an approximate accelerator from a high-level specification while satisfying a previously defined accuracy constraint is a joint design space exploration and high-level synthesis challenge. This dissertation proposes automated methods for designing and implementing approximate accelerators built with approximate arithmetic circuits.**

***Index Terms*—Approximate computing, design automation, design tools, error analysis.**

I. Introduction

In the last decade, the need for computing efficiency has motivated the coming forth of new devices, architectures, and design techniques. Approximate Computing (AxC) has emerged as a modern energy-efficient design paradigm for applications that present inherent tolerance to errors. By reducing the accuracy of the results in current applications, such as image processing, computer vision, and machine learning [1], to an acceptable amount, savings in the circuit area, delay, and power consumption can be achieved.

With the emergence of the approximate computing paradigm, different techniques have been proposed at different abstraction layers, ranging from software [2] to hardware [3]. Remarkably, many approximate functional units have been reported in the literature, mainly approximate adders and multipliers. For a plethora of such approximate circuits, and considering their usage as building blocks for the design of approximate accelerators for error- tolerant applications, a challenge arises: selecting those approximate circuits for a given application that minimize the required resources while satisfying a defined accuracy.

This dissertation proposes automated methods for designing and implementing approximate accelerators built with approximate arithmetic circuits. To achieve it, this dissertation addresses the challenges described in the following sections, and it provides subsequent novel contributions.

*The author is also with the Instituto Tecnológico de Costa Rica.

II. Approximate Circuits Generation and Characterization

Many approximate adders and multipliers have been reported in the literature, either by proposing approximate designs from accurate implementations, such as the ripple-carry adder, or by generating them through Approximate Logic Synthesis (ALS) methods. A representative set of these approximate components is required to build approximate accelerators automatically.

In that sense, this dissertation presents two approaches to generate such approximate arithmetic circuits. First, AUGER is introduced, a tool capable of generating Register-Transfer Level (RTL) descriptions for a broad set of approximate adders and multipliers for different data bit-width and accuracy configuration [4]. A Design Space Exploration (DSE) of approximate components can be performed with AUGER to find those Pareto-optimal for a given bit-width, approximation range, and circuit metric. Then, AxLS is presented, a framework for ALS that allows the implementation of state-of-the-art methods, and the proposition of novel ones, to perform structural netlist transformations and to generate approximate arithmetic circuits from accurate ones [3]. Moreover, both tools provide an error characterization, in the form of error distribution and circuit characteristics (area, delay, and power) for each approximate circuit they generate. This information is essential for automating the design of approximate accelerators.

III. Accuracy Estimation

Error estimation is a cornerstone of approximate computing. This is essential for the automated design of approximate accelerators built with approximate arithmetic circuits. Despite the tolerance to errors, approximate accelerators must limit the output error to satisfy quality constraints, whether defined by the application, the developer, or even the end-user. The use of approximate arithmetic circuits in accelerators imposes the challenge of understanding how the error introduced by each approximate circuit behaves and propagates through other exact and approximate computations, and finally, how it accumulates at the output.

This dissertation proposes a novel compiler-driven methodology for estimating the accuracy at the output of an approximate accelerator design through an error propagation methodology [5]. Using as a base a set of analytical rules to model the error propagation for individual calculations, a

model is defined to estimate the propagation of error distributions represented as a probability mass function. These rules consider the propagation of errors through other approximate and accurate arithmetic computations. Since the design of accelerators is carried out in conjunction with the source code of the application, this methodology proposes using the source code of the function to be accelerated as an input. A custom-defined pragma directive is defined to annotate the code and indicate which accurate operations are replaced by approximate ones. A modified compiler is proposed to handle these annotations and append metadata to the code's intermediate representation. This information is later used to statically analyze the code, using error propagation models previously determined, to obtain an error distribution of the output, from which different error metrics can be determined for the accuracy assessment.

IV. High-Level Synthesis Generation

In the design of approximate accelerators, repetitive gate-level simulations and circuit synthesis consume a significant time to explore many, or even all, possible combinations for a given set of approximate arithmetic circuits. On the other hand, current trends for designing accelerators are based on High-Level Synthesis (HLS) tools.

This dissertation presents a novel automated framework for HLS of approximate accelerators using a given library of approximate arithmetic circuits [6]. For this, it considers a set of approximate adders and multipliers previously characterized [4]. To avoid circuit synthesis and gate-level simulations, this dissertation introduces a set of analytical models for estimating the necessary computational resources when using approximate adders and multipliers in approximate designs. These models complement those for accuracy propagation estimation in approximate accelerators. A DSE methodology for error-tolerant applications, in which analytical models are used to estimate resources needed, and the accuracy of approximate accelerators is proposed. This DSE methodology allows finding Pareto-optimal solutions for approximate accelerator designs, minimizing the required resources while meeting accuracy constraints. Furthermore, this DSE methodology is integrated into an HLS tool, LegUp, to generate accelerators from C language descriptions automatically. Experimental results for approximate accelerators generated with this novel framework show that even with a small accuracy degradation on the accuracy, about 30% of energy reduction can be achieved.

V. Error Correction

The use of approximate accelerators must ensure defined accuracy levels. Hence, the errors generated due to approximations must remain within a defined bound required for the application to produce good-enough results. With this consideration, approximate accelerators are designed to meet a given accuracy for an error metric. However, the errors produced by an approximate accelerator depend on the input data. This data may differ from the one used during the design, making the accelerator to have unacceptable errors at run time.

On the other hand, the defined tolerable error threshold can be dynamically changed. In both cases, those undesired errors need to be corrected to achieve the desired accuracy. State-of-the-art approaches address this issue by re-computing those accelerator invocations that produce unacceptable errors at the software level. For this, lightweight, pre-trained predictors are used to estimate if an accelerator invocation will produce an error beyond the defined accuracy threshold for a given set of input data. Nevertheless, software re-computations reduce approximate computing benefits, especially when input data variations are high at run time.

This dissertation proposes a methodology to explore and apply fine-grained, selective error correction in approximate accelerators to balance the costs associated with accuracy control [7]. This methodology reduces the required coarse-grained correction, exact re-computations performed by the host processor, to meet a defined accuracy constraint, finding a balance between software- and hardware-level error correction, while not significantly reducing the benefits obtained due to the approximations, such as the speedup achieved by an approximate accelerator with respect to an accurate one. This methodology is particularly useful for approximate accelerators built with approximate arithmetic circuits, the type of approximate accelerators targeted in this dissertation, achieving up to 20% performance gain compared to reported approaches in the literature.

Acknowledgment

This work was supported by the Instituto Tecnológico de Costa Rica and the STIBET Abschluss-Stipendium of the Deutsche Akademische Austauschdienst (DAAD).

References

[1] J. Castro-Godínez, D. Hernández-Araya, M. Shafique, and J. Henkel, "Approximate Acceleration for CNN-based Applications on IoT Edge Devices," in *2020 IEEE 11th Latin American Symposium on Circuits & Systems (LASCAS)*, Feb. 2020, pp. 1–4.

[2] J. Castro-Godínez, M. Shafique, and J. Henkel, "Towards Quality-Driven Approximate Software Generation for Accurate Hardware: Work-in-Progress," in *2020 International Conference on Compilers, Architecture, and Synthesis for Embedded Systems (CASES)*, Sep. 2020.

[3] J. Castro-Godínez, H. Barrantes-García, M. Shafique, and J. Henkel, "AxLS: An Open-Source Framework for Netlist Transformation Approximate Logic Synthesis," in *3rd Workshop on Open-Source EDA Technology (WOSET)*, Nov. 2020.

[4] D. Hernández-Araya, J. Castro-Godínez, M. Shafique, and J. Henkel, "AUGER: A Tool for Generating Approximate Arithmetic Circuits," in *2020 IEEE 11th Latin American Symposium on Circuits & Systems (LASCAS)*, Feb. 2020, pp. 1–4.

[5] J. Castro-Godínez, S. Esser, M. Shafique, S. Pagani, and J. Henkel, "Compiler-Driven Error Analysis for Designing Approximate Accelerators," in *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Mar. 2018, pp. 1027–1032.

[6] J. Castro-Godínez, J. Mateus-Vargas, M. Shafique, and J. Henkel, "AxHLS: Design Space Exploration and High-Level Synthesis of Approximate Accelerators using Approximate Functional Units and Analytical Models," in *2020 IEEE/ACM 39th International Conference on Computer-Aided Design (ICCAD)*, Nov. 2020.

[7] J. Castro-Godínez, M. Shafique, and J. Henkel, "ECAx: Balancing Error Correction Costs in Approximate Accelerators," *ACM Trans. Embed. Comput. Syst.*, vol. 18, no. 5s, Oct. 2019.