Automated Design of Approximate Accelerators

Jorge Castro-Godínez

**Motivation and Contribution**

With the emergence of the approximate computing paradigm, many approximate functional units have been reported in the literature, mainly approximate adders and multipliers.

**Challenge:**

Given a design for an error-tolerant application and a set of approximate components, which approximate arithmetic circuits should be used to minimize the computational effort (e.g., energy), while satisfying a defined accuracy?

**Contributions**

- **Approximate Arithmetic Circuits [1,2]**
- **HLS Approximate Accelerators [4]**
- **Resource Estimation [4]**
- **Approximate Accelerators**
- **Error Correction [5]**

**High-Level Synthesis Generation**

Repetitive gate-level simulations and circuit synthesis consume a significant time to explore many, or even all, possible designs for an approximate accelerator, given set of approximate arithmetic circuits.

**Goal:** automated framework for HLS of approximate accelerators using a given library of approximate arithmetic circuits and relying on analytical models [4].

DSE methodology finds Pareto-optimal solutions for approximate accelerator designs, minimizing the required resources while meeting accuracy constraints.

**Selective Error Correction**

For approximate accelerators, an accuracy limit must be defined and satisfied [3].

**Goal:** Improve application performance while introducing selective error correction at approximate adder and reducing the required re-computations [5].

**Accuracy Estimation**

The design of approximate accelerators, particularly using approximate arithmetic circuits, requires methods to estimate the accuracy at the output.

- How the errors introduced by an approximate addition propagate up to the output?
- How these and other errors interact with other accurate and approximate computations.
- Models to define how errors propagate are required.

**Goal:** Automate the accuracy evaluation of approximate designs without requiring exhausting simulations [3].

- CEDA: Compiler-driven Error Analysis for Designing Approximate Accelerators
- Software model of accelerator is used.
- DFG representation is extracted from LLVM IR.
- Then analysis is performed.

**Approximate designs for different MED targets.**

Less energy required, with more error tolerance, but meeting accuracy constraint.

---


