Prediction-Based Quality Control for Approximate Accelerators

Presented By: Michael Ringenburg

Divya Mahajan, Amir Yazdanbakhsh, Jongse Park, Bradley Thwaites Hadi Esmaeilzadeh

Alternative Computing Technologies (ACT) Lab
Georgia Institute of Technology
Approximate Acceleration

Core invokes approximate accelerator in lieu of frequently executed and safe-to-approximate function.

Invoking the accelerator at all times leads to fixed degree of error.

Invocations showing high error

Need to intelligently remove the invocations that give high error.
Quality Control for Approximate Acceleration

Lack of microarchitectural mechanisms to control the quality degradation

Core Idea: Predict and avoid accelerator invocations that will lead to large errors
Error Predictor

Error Predictor for quality control sits between core and approximate accelerator

Devised and evaluated two quality control mechanisms: table-based predictor and neural predictor
Factors influencing Error

Accelerator Error = |Output_{accelerator} - Output_{original}|

Output_{accelerator} = f (accelerator inputs, accelerator configuration)

Output_{accelerator} = f (accelerator inputs)

Error predictor only needs the accelerator inputs for quality control
Multi-Input Table-based Predictor

Input FIFO

Output FIFO

Precise

NOT(Precise)

Core

Approximate Accelerator

Error Predictor

Accelerator Inputs

Precise

Input Signature

Multiple Input Signature Register (MISR)
Multi-Input Hashing and Input Signature Generation

Input FIFO


last_element

hash:

Index[7]  Index[6]  Index[1]  Index[0]
A small ensemble of table-based predictors achieve high prediction accuracy.
Neural Predictor

![Diagram showing a neural predictor network with layers labeled Input Layer, Hidden Layer, and Output Layer. Connections between nodes are indicated by arrows, with inputs labeled in0 to in8 and outputs labeled ‘0’ and ‘1’.
Predictor Training

• Programmer provides:
  – Application specific quality requirement
  – Quality Metric
  – Set of representative application inputs

• Algorithm maps final output requirement to a threshold \((\text{th})\) on \textbf{accelerator error}.

\[
\text{Accelerator error} < \text{th} : \text{learn to invoke the accelerator}
\]
\[
\text{Accelerator error} > \text{th} : \text{learn to fall back to the original function}
\]
Prediction Accuracy and Invocation Rate (5% Error)

Table-based (AVG)
78% Accuracy
75% Invocation rate

Neural Predictor (AVG)
82% Accuracy
80% Invocation rate
Table-based

- 2.5x Speedup
- 2.9x Energy Savings

Neural Predictor (AVG)

- 2.6x Speedup
- 3.2x Energy Savings