Axilog: Language Support for Approximate Hardware Design

Amir Yazdanbakhsh
Jongse Park
Kartik Ramkrishnan
Abbas Rahimi

Divya Mahajan
Anandhavel Nagendrakumar
Nishanthi Ravindran
Hadi Esmaeilzadeh

Bradley Thwaites
Sindhuja Sethuraman
Rudra Jariwala
Kia Bazargan

Georgia Institute of Technology
University of Minnesota
UC San Diego

DATE 2015
Approximate computing

Embracing error

• Relax the abstraction of \textit{near-perfect} accuracy in \textit{general-purpose} computing/communication/storage

• Allow errors to happen during computation/communication/storage
  – Improve resource utilization efficiency
    • Energy, bandwidth, capacity, ...
  – Improve performance

• Build \textit{acceptable} systems from \textit{intentionally-made unreliable} software and hardware components

• Avoid \textit{overkill} and \textit{worst-case} design
Avoiding Worst-Case Design

Approximate Computing

Generality

Efficiency

Cost

Precision

Reliability

Determinism

Performance

Cost

Application

Programming Language

Compiler

Architecture

Microarchitecture

Circuit

Physical Device
<table>
<thead>
<tr>
<th>Goals</th>
<th>Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design the <strong>first</strong> HDL for:</td>
<td>Approximate HDL:</td>
</tr>
<tr>
<td>1) Approximate HW <strong>Design</strong></td>
<td>1) <strong>High-level</strong></td>
</tr>
<tr>
<td>2) Approximate HW <strong>Reuse</strong></td>
<td>2) <strong>Automated</strong></td>
</tr>
<tr>
<td>3) Approximate <strong>Synthesis</strong></td>
<td>3) <strong>Backward compatible</strong></td>
</tr>
<tr>
<td></td>
<td>4) <strong>Safety</strong></td>
</tr>
</tbody>
</table>
Safety in Hardware

Controller

Precise

Approximate

Precise

Datapath

Clock
Axilog Annotations

Design Annotations

- relax (relax_local)
- restrict (restrict_global)

Reuse Annotations

- approximate
- critical
- bridge
Design Annotations
Relaxing Accuracy Requirements

module ripple_carry_adder(a, b, c_in, c_out, s)

...  
full_adder f0(a[0], b[0], c_in, w0, s[0])
full_adder f1(a[1], b[1], w0, c_out, s[1])
relax (s);

...
Relaxing Accuracy Requirements

module ripple_carry_adder(a, b, c_in, c_out, s)
...

    full_adder f0(a[0], b[0], c_in, w0, s[0])
    full_adder f1(a[1], b[1], w0, c_out, s[1])

    relax (s);

    ...

b[1]  a[1]
     |     |
     v     v
 c_out   s[1]
     |     |
     v     v
 w0     s[0]
Scoping Approximation (relax_local)

module full_adder
  (a, b, c_in, c_out, s)

relax_local (s);

full_adder f0 (...)
full_adder f1 (...)
relax (s[0]);
Scoping Approximation (relax_local)

module full_adder
  (a, b, c_in, c_out, s)
...
  relax_local (s);
...

... full_adder f0 (…)
full_adder f1(...)
...

relax (s[0]);
...

...
Restricting Approximation
Restricting Approximation
Restricting Approximation

 clk
 rst
 x

Restrict (w1)

Restrict (w2)

Restrict (w3)

relax (y)
Restricting Approximation Globally

\[ \text{module full_adder}(a, b, c_{\text{in}}, c_{\text{out}}, s); \]

\[ \ldots \]

\[ \textbf{approximate} \ \text{output} \ s; \]

\[ \textbf{relax} \ (s); \]

\[ \ldots \]

\[ \textbf{endmodule} \]

\[ \ldots \]

\[ \textbf{restrict}_{\text{global}}(s[31:0]); \]
Restricting Approximation Globally

module full_adder(a, b, c_in, c_out, s);

... approximate output s;

relax (s);

... endmodule

... restrict_global(s[31:0]);
Reuse Annotations
Outputs Carrying Approximate Semantics

sobel_filter


approximate output
r [4:0]
r [7:5]
Critical Inputs

... critical input reset;
... critical input clock;
...
Bridging Approximate Wires to Critical Inputs

\[
\begin{align*}
\text{...}
\text{and } a_1(s, a_0, a_1); \\
\text{relax } (s); \\
\text{bridge } (s); \\
\text{multiplexer } m_0(s, a_0, a_1, \text{out}); \\
\text{...}
\end{align*}
\]
Bridging Approximate Wires to Critical Inputs

... and \(a_1(s, a_0, a_1)\);

relax \((s)\);

bridge \((s)\);

multiplexer \(m_0(s, a_0, a_1, \text{out})\);

...
Baseline Synthesis Flow

Highest frequency with minimum power and area
Relaxability Inference Analysis

1. Circuit under analysis with Axilog annotations
2. Identify the wires which are driving unannotated wires or annotated with restrict within the module under analysis
3. Identify the relaxed outputs of the instantiated submodules
4. Marks any wire that affects a globally restricted wire as precise
5. Safe to approximate gates
Approximate Synthesis Flow
Measurements

**Tools for Synthesis and Energy Analysis**
- Synopsys Design Compiler
- Synopsys Primetime

**Timing Simulation with SDF back annotations**
- Cadence NC-Verilog

**Standard Cell Library**
- TSMC 45-nm multi-$V_t$
- Slowest PVT corner (SS, 0.81V, 0C) for baseline results
<table>
<thead>
<tr>
<th>Benchmark</th>
<th># Lines</th>
<th># Annotations</th>
<th>Design</th>
<th>Reuse</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sobel</td>
<td>143</td>
<td>6</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>Brent-Kung</td>
<td>352</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Kogge-Stone</td>
<td>353</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>K-means</td>
<td>10,985</td>
<td>7</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>Wallace Tree</td>
<td>13,928</td>
<td>5</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>ForwardK</td>
<td>18,282</td>
<td>5</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>InverseK</td>
<td>22,407</td>
<td>8</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>Neural Network</td>
<td>21,053</td>
<td>4</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>

**Benchmarks**

Arithmetic Computation, Signal Processing, Robotics, Machine Learning, Image Processing
Energy Reduction

- Error ≤ 5%
- Error ≤ 10%

Bar chart showing energy reduction for different algorithms.

- Brent-Kung
- FIR
- ForwardK
- InverseK
- K-means
- Kogge-Stone
- Wallace Tree
- Neural
- Sobel
- Geomean
Area Reduction

Error \leq 5\%  \quad \text{Error} \leq 10\%

Area Reduction

Brent-Kung  
FIR  
ForwardK  
InverseK  
K-means  
Kogge-Stone  
Wallace Tree  
Neural  
Sobel  
Geomean

Area Reduction
Output Quality Degradation in Sobel

10% Quality Loss provides 57% energy savings

10% Quality loss is nearly indiscernible to the eye yet provides 57% energy savings
Axilog

First HDL for Approximation

- Design
- Reuse
- Automation
- High-level
- Backward-compatibility
- Safety

<table>
<thead>
<tr>
<th>Energy Savings</th>
<th>Area Reduction</th>
<th>Code Annotations</th>
</tr>
</thead>
<tbody>
<tr>
<td>54%</td>
<td>1.9x</td>
<td>2-12</td>
</tr>
</tbody>
</table>

http://www.act-lab.org/artifacts/axilog