AHA: Agile Hardware-Compiler Co-design and Verification

Priyanka Raina
Assistant Professor of Electrical Engineering
Stanford University
praina@stanford.edu

August 31, 2023
Day 2 Agenda

9:15  Pushing the Limits of Scaling Laws in the Age of Large Language Models  
      Azalia Mirhoseini

10:00 Onyx: A Coarse-Grained Reconfigurable Array for Accelerating Dense and  
       Sparse Tensor Algebra  
       Kalhan Koul

10:20 Break

10:50 Cascade: An Application Pipelining Toolkit for Coarse-Grained  
       Reconfigurable Arrays  
       Jackson Melchert

11:10 Efficiently Synthesizing Lowest Cost Rewrite Rules for Instruction Selection  
      Ross Daly

11:30 Lightweight Online Learning for Bounded Model Checking  
      Andrew Wu

11:50 Lunch
# Day 2 Agenda

<table>
<thead>
<tr>
<th>Time</th>
<th>Session Title</th>
<th>Presenter</th>
</tr>
</thead>
<tbody>
<tr>
<td>13:20</td>
<td>Energy Efficiency and AI Hardware</td>
<td>Bill Dally</td>
</tr>
<tr>
<td>14:05</td>
<td>Sparse PyTorch</td>
<td>Bobby Yan</td>
</tr>
<tr>
<td>14:25</td>
<td>Sparse Recurrences and Linear Solvers</td>
<td>Shiv Sundram</td>
</tr>
<tr>
<td>14:45</td>
<td>ASPEN: Acceleration of Visual Inertial Odometry for Extended Reality on an FPGA</td>
<td>Kathleen Feng</td>
</tr>
<tr>
<td>15:05</td>
<td>Break</td>
<td></td>
</tr>
</tbody>
</table>
## Day 2 Agenda

<table>
<thead>
<tr>
<th>Time</th>
<th>Title</th>
<th>Presenter</th>
</tr>
</thead>
<tbody>
<tr>
<td>15:35</td>
<td>Learnings from Edge AI Chip Designs with Applications in Probabilistic and NLP Inference and Training</td>
<td>Thierry Tambe</td>
</tr>
<tr>
<td>15:55</td>
<td>Interrupt-Driven Viterbi for High-Speed Links</td>
<td>Zachary Myers</td>
</tr>
<tr>
<td>16:15</td>
<td>Avanergy: An Architecture-VLSI Abstraction for Automated Energy-Aware Design-Space Exploration Tools</td>
<td>Christopher Torng</td>
</tr>
<tr>
<td>16:35</td>
<td>Wrap-up Discussion &amp; Closing Thoughts</td>
<td>Priyanka Raina</td>
</tr>
</tbody>
</table>
AHA’s Future Research Directions

- **Domain-Specific Accelerators**
  - Improve performance of formal tools (model checkers, SMT solvers)
  - Bring ML into automated reasoning
  - Bring automated reasoning into design

- **Domain-Specific Compilers**
  - Compiler generator for dynamic languages
  - Exploring new memories (e.g., eDRAM for transient data)
  - Auto-scheduling on accelerators
  - Compilers and programming models for:
    - Sparse ML
    - Linear solvers
    - Recurrences
    - Relational algebra

- **Formal Verification Tools**
  - Chiplet-based design
  - Enabling application designers to make specialized systems

- **Accelerators for sparse ML**
- **Number systems for ML**
- **Auto-scheduling on accelerators**

- **Agile Co-design**