Resources

• Tutorial Related
  – Tutorial Website: http://accelergy.mit.edu/isca20_tutorial.html
  – Tutorial Docker: https://github.com/Accelergy-Project/timeloop-accelergy-tutorial
    • Various exercises and example designs and environment setup for the tools

• Other
  – Infrastructure Docker: https://github.com/Accelergy-Project/accelergy-timeloop-infrastructure
    • Pure environment setup for the tools without exercises and example designs
  – Open Source Tools
    • Accelergy: http://accelergy.mit.edu/
    • Timeloop: https://github.com/NVlabs/timeloop

• Papers:
Data and computation-intensive applications are power hungry

We must quickly evaluate energy efficiency of arbitrary potential designs in the large design space
From Architecture Blueprints to Physical Systems

- How many levels in the memory hierarchy?
- How large are the memories at each level?
- How many PEs are there?
- What are the X and Y dimensions of the PE array?
- …
From Architecture Blueprints to Physical Systems

Global Buffer (GLB)

*processing element

PE*0

buffer

PE2

PE3

Architecture Stage

RTL Model

Physical Layout

Fabricated System

[Chen, ISSCC 2016]
Physical-Level Energy Estimation and Design Exploration

- How many levels in the memory hierarchy?
- How large are the memories at each level?
- How many PEs are there?
- What are the X and Y dimensions of the PE array?
- ...

Slow design space exploration
- Long simulations on gate-level components
- Long turn-around time for each potential design
Physical-Level Energy Estimation and Design Exploration

Building systems with emerging technologies can be even more time-consuming, limiting the amount of design space.
Accelergy Overview

• Accelergy Infrastructure
  – Performs architecture-level estimations to enable rapid design space exploration
  – Supports modeling of diverse architectures with various underlying technologies
  – Improves estimation accuracy by allowing fine-grained classification of components’ runtime behaviors
  – Supports succinct modeling of complicated architectures

• Validation on various accelerator designs
  – 95% accurate on a conventional digital accelerator design
  – Modeling of processing in memory (PIM) based DNN accelerator designs
Architecture-Level Energy Estimation and Design Exploration

Fast design space exploration

- Short simulations on architecture-level components
- Short turn-around time for each potential design
Timeloop requires energy reference tables (ERTs) to evaluate the energy efficiency of a potential mapping.
Existing Accelerator Estimators Lack Flexibility

- **Accelerator-Specific Estimators:**
  - **Aladdin** [Shao, ISCA2014], **fixed-cost** [Yang, Asilomar2017]

**Architecture Description**

- **GLB**
  - tech: 45nm, width: 64, depth: 1024

- **MAC**
  - tech: 45nm, width: 16

- **Buffer**
  - tech: 45nm, width: 16, depth: 256

Description with defined **primitive components** (basic building blocks)

**Energy Estimator**
Existing Accelerator Estimators Lack Flexibility

- **Accelerator-Specific Estimators: Aladdin** [Shao, ISCA2014], **fixed-cost** [Yang, Asilomar2017]

### Architecture Description

- **GLB**
  - tech: 45nm, width: 64, depth: 1024

- **buffer**
  - tech: 45nm, width: 16, depth: 256

- **MAC**
  - tech: 45nm, width: 16

### Energy Estimator

#### Energy Reference Table (ERT)

<table>
<thead>
<tr>
<th>Comp.</th>
<th>Action</th>
<th>Energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLB</td>
<td>access()</td>
<td>100pJ</td>
</tr>
<tr>
<td>buffer</td>
<td>access()</td>
<td>10pJ</td>
</tr>
<tr>
<td>MAC</td>
<td>compute()</td>
<td>5pJ</td>
</tr>
</tbody>
</table>
Existing Accelerator Estimators Lack Flexibility

- Accelerator-Specific Estimators: Aladdin [Shao, ISCA2014], fixed-cost [Yang, Asilomar2017]

### Architecture Description

<table>
<thead>
<tr>
<th>PE</th>
<th>GLB</th>
<th>buffer</th>
<th>MAC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>tech: 45nm</td>
<td>width: 64</td>
<td>depth: 1024</td>
</tr>
<tr>
<td></td>
<td>tech: 45nm, width: 16</td>
<td>depth: 256</td>
<td></td>
</tr>
<tr>
<td></td>
<td>tech: 45nm, width: 16</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Energy Estimator

#### Energy Reference Table (ERT)

<table>
<thead>
<tr>
<th>Comp.</th>
<th>Action</th>
<th>Energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLB</td>
<td>access()</td>
<td>100pJ</td>
</tr>
<tr>
<td>buffer</td>
<td>access()</td>
<td>10pJ</td>
</tr>
<tr>
<td>MAC</td>
<td>compute()</td>
<td>5pJ</td>
</tr>
</tbody>
</table>

### Action Counts

<table>
<thead>
<tr>
<th>Comp.</th>
<th>Action</th>
<th>Counts</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLB</td>
<td>access()</td>
<td>10</td>
</tr>
<tr>
<td>buffer</td>
<td>access()</td>
<td>800</td>
</tr>
<tr>
<td>MAC</td>
<td>compute()</td>
<td>400</td>
</tr>
</tbody>
</table>

*Action Counts* Comes from a performance model (e.g., cycle accurate simulator)
Existing Accelerator Estimators Lack Flexibility

- **Accelerator-Specific Estimators:** Aladdin [Shao, ISCA2014], fixed-cost [Yang, Asilomar2017]

### Architecture Description

- **PE**
  - **GLB**
    - tech: 45nm, width: 64, depth: 1024
  - **MAC**
    - tech: 45nm, width: 16

### Energy Estimator

#### Energy Reference Table (ERT)

<table>
<thead>
<tr>
<th>Comp.</th>
<th>Action</th>
<th>Energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLB</td>
<td>access()</td>
<td>100pJ</td>
</tr>
<tr>
<td>buffer</td>
<td>access()</td>
<td>10pJ</td>
</tr>
<tr>
<td>MAC</td>
<td>compute()</td>
<td>5pJ</td>
</tr>
</tbody>
</table>

### Energy Estimations

<table>
<thead>
<tr>
<th>Name</th>
<th>Energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLB</td>
<td>1000pJ</td>
</tr>
<tr>
<td>buffer</td>
<td>8000pJ</td>
</tr>
<tr>
<td>MAC</td>
<td>2000pJ</td>
</tr>
</tbody>
</table>

### Action Counts

**Comes from a performance model**
(e.g., cycle accurate simulator)
Existing Accelerator Estimators Lack Flexibility

- **Accelerator-Specific Estimators:**
  - **Aladdin** [Shao, ISCA2014], **fixed-cost** [Yang, Asilomar2017]

### Architecture Description

- **GLB’**
  - tech: 65nm, width: 64, depth: 1024
- **MAC’**
  - tech: 65nm, width: 16

### Energy Reference Table (ERT)

<table>
<thead>
<tr>
<th>Comp.</th>
<th>Action</th>
<th>Energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLB’</td>
<td>access()</td>
<td>100pJ</td>
</tr>
<tr>
<td>buffer’</td>
<td>access()</td>
<td>10pJ</td>
</tr>
<tr>
<td>MAC’</td>
<td>compute()</td>
<td>5pJ</td>
</tr>
</tbody>
</table>

### Energy Calculator

Not generalizable to other designs
Accelergy Overview

• Accelergy Infrastructure
  – Performs architecture-level estimations to enable rapid design space exploration
  – **Supports modeling of diverse architectures with various underlying technologies**
  – Improves estimation accuracy by allowing fine-grained classification of components runtime behaviors
  – Supports succinct modeling of complicated architectures

• Validation on various accelerator designs
  – 95% accurate on a conventional digital accelerator design
  – Modeling of processing in memory (PIM) based DNN accelerator designs
Accelergy: Flexibly Model Various Primitive Components

Architecture Description

- **GLB**
  - tech: 45nm
  - width: 64
  - depth: 1024
  - SRAM

- **MAC**
  - tech: 45nm
  - width: 16
  - depth: 256

Each component belongs to a primitive class

Available at [http://accelergy.mit.edu/](http://accelergy.mit.edu/)
Accelergy: Flexibly Model Various Primitive Components

Architecture Description

- GLB
  - tech: 45nm
  - width: 64
  - depth: 1024
- SRAM
- buffer
  - tech: 45nm
  - width: 16
  - depth: 256
- MAC
  - tech: 45nm
  - width: 16

Each component belongs to a primitive class

ERT/ART Generator

Primitive Component Library

Estimation Plug-ins

SRAM class has associated action “access”

Available at http://accelergy.mit.edu/
Accelergy: Flexibly Model Various Primitive Components

Each component belongs to a primitive class

Simple Example Estimation Plug-in

<table>
<thead>
<tr>
<th>class</th>
<th>tech.</th>
<th>width</th>
<th>depth</th>
<th>action</th>
<th>energy (pJ)</th>
<th>area (um²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAC</td>
<td>45nm</td>
<td>16b</td>
<td>N/A</td>
<td>compute</td>
<td>5</td>
<td>0.4</td>
</tr>
<tr>
<td>SRAM</td>
<td>45nm</td>
<td>64b</td>
<td>1024</td>
<td>access</td>
<td>100</td>
<td>20</td>
</tr>
<tr>
<td>SRAM</td>
<td>45nm</td>
<td>16b</td>
<td>256</td>
<td>access</td>
<td>10</td>
<td>2</td>
</tr>
</tbody>
</table>

Available at http://accelergy.mit.edu/
Accelergy: Flexibly Model Various Primitive Components

Each component belongs to a primitive class

Simple Example Estimation Plug-in

<table>
<thead>
<tr>
<th>class</th>
<th>tech.</th>
<th>width</th>
<th>depth</th>
<th>action</th>
<th>energy (pJ)</th>
<th>area (μm²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAC</td>
<td>45nm</td>
<td>16b</td>
<td>N/A</td>
<td>compute</td>
<td>5</td>
<td>0.4</td>
</tr>
<tr>
<td>SRAM</td>
<td>45nm</td>
<td>64b</td>
<td>1024</td>
<td>access</td>
<td>100</td>
<td>20</td>
</tr>
<tr>
<td>SRAM</td>
<td>45nm</td>
<td>16b</td>
<td>256</td>
<td>access</td>
<td>10</td>
<td>2</td>
</tr>
</tbody>
</table>

Available at [http://accelergy.mit.edu/](http://accelergy.mit.edu/)

ERT/ART (in progress)

<table>
<thead>
<tr>
<th>comp.</th>
<th>action</th>
<th>energy</th>
<th>area (μm²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLB</td>
<td>access()</td>
<td>100pJ</td>
<td>20</td>
</tr>
</tbody>
</table>
Accelergy: Flexibly Model Various Primitive Components

**Architecture Description**

Each component belongs to a primitive class

**Simple Example Estimation Plug-in**

<table>
<thead>
<tr>
<th>class</th>
<th>tech.</th>
<th>width</th>
<th>depth</th>
<th>action</th>
<th>energy (pJ)</th>
<th>area (um²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAC</td>
<td>45nm</td>
<td>16b</td>
<td>N/A</td>
<td>compute</td>
<td>5</td>
<td>0.4</td>
</tr>
<tr>
<td>SRAM</td>
<td>45nm</td>
<td>64b</td>
<td>1024</td>
<td>access</td>
<td>100</td>
<td>20</td>
</tr>
<tr>
<td>SRAM</td>
<td>45nm</td>
<td>16b</td>
<td>256</td>
<td>access</td>
<td>10</td>
<td>2</td>
</tr>
</tbody>
</table>

**ERT/ART**

<table>
<thead>
<tr>
<th>comp.</th>
<th>action</th>
<th>energy</th>
<th>area</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLB</td>
<td>access()</td>
<td>100pJ</td>
<td>20um²</td>
</tr>
<tr>
<td>buffer</td>
<td>access()</td>
<td>10pJ</td>
<td>2um²</td>
</tr>
<tr>
<td>MAC</td>
<td>compute()</td>
<td>3pJ</td>
<td>0.3um²</td>
</tr>
</tbody>
</table>

Available at [http://accelergy.mit.edu/](http://accelergy.mit.edu/)
Accelergy: Flexibly Model Various Primitive Components

**Architecture Description**

- **GLB**
  - tech: 45nm
  - width: 64
  - depth: 1024

- **MAC**
  - tech: 45nm
  - width: 16

**Primitive Component Library**

- **buffer**
  - tech: 45nm
  - width: 16
  - depth: 256

- **SRAM**
  - tech: 45nm
  - width: 16
  - depth: 256

**ERT/ART Generator**

- **MAC**
  - compute: 5
  - energy (pJ): 0.4

- **SRAM**
  - access: 100
  - area (μm²): 20

**Energy Calculator**

**Simple Example Estimation Plug-in**

<table>
<thead>
<tr>
<th>class</th>
<th>tech.</th>
<th>width</th>
<th>depth</th>
<th>action</th>
<th>energy (pJ)</th>
<th>area (μm²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAC</td>
<td>45nm</td>
<td>16b</td>
<td>N/A</td>
<td>compute</td>
<td>5</td>
<td>0.4</td>
</tr>
<tr>
<td>SRAM</td>
<td>45nm</td>
<td>64b</td>
<td>1024</td>
<td>access</td>
<td>100</td>
<td>20</td>
</tr>
<tr>
<td>SRAM</td>
<td>45nm</td>
<td>16b</td>
<td>256</td>
<td>access</td>
<td>10</td>
<td>2</td>
</tr>
</tbody>
</table>

Available at [http://accelergy.mit.edu/](http://accelergy.mit.edu/)
Accelergy: Flexibly Model Various Primitive Components

Architecture Description

- GLB
  - tech: 45nm
  - width: 64
  - depth: 1024
- Buffer
  - tech: 45nm, width: 16
  - depth: 256
- SRAM
  - tech: 45nm, width: 16
- MAC
  - tech: 45nm, width: 16
  - depth: 256

Simple Example Estimation Plug-in

<table>
<thead>
<tr>
<th>class</th>
<th>tech.</th>
<th>width</th>
<th>depth</th>
<th>action</th>
<th>energy (pJ)</th>
<th>area (um²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAC</td>
<td>45nm</td>
<td>16b</td>
<td>N/A</td>
<td>compute</td>
<td>5</td>
<td>0.4</td>
</tr>
<tr>
<td>SRAM</td>
<td>45nm</td>
<td>64b</td>
<td>1024</td>
<td>access</td>
<td>100</td>
<td>20</td>
</tr>
<tr>
<td>SRAM</td>
<td>45nm</td>
<td>16b</td>
<td>256</td>
<td>access</td>
<td>10</td>
<td>2</td>
</tr>
</tbody>
</table>

Action Counts

- GLB access(): 10
- Buffer access(): 800
- MAC compute(): 400

Comes from a performance model (e.g., cycle accurate simulator)

Available at http://accelergy.mit.edu/
Accelergy: Flexibly Model Various Primitive Components

Architecture Description

Accelergy

ERT/ART Generator

Primitive Component Library

Energy Calculator

Simple Example Estimation Plug-in

class | tech. | width | depth | action | energy (pJ) | area (um²)
------|-------|-------|-------|--------|-------------|---------
MAC   | 45nm  | 16b   | N/A   | compute| 5           | 0.4     
SRAM  | 45nm  | 64b   | 1024  | access | 100         | 20      
SRAM  | 45nm  | 16b   | 256   | access | 10          | 2       

Action Counts

<table>
<thead>
<tr>
<th>Comp.</th>
<th>Action</th>
<th>Counts</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLB</td>
<td>access()</td>
<td>10</td>
</tr>
<tr>
<td>buffer</td>
<td>access()</td>
<td>800</td>
</tr>
<tr>
<td>MAC</td>
<td>compute()</td>
<td>400</td>
</tr>
</tbody>
</table>

Energy Estimates

<table>
<thead>
<tr>
<th>Name</th>
<th>Energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLB</td>
<td>1000pJ</td>
</tr>
<tr>
<td>buffer</td>
<td>8000pJ</td>
</tr>
<tr>
<td>MAC</td>
<td>2000pJ</td>
</tr>
</tbody>
</table>

Available at [http://accelergy.mit.edu/](http://accelergy.mit.edu/)
Accelrys: Flexibly Model Various Primitive Components

Architecture Description

**GLB**
- Tech: 45nm
- Width: 64
- Depth: 1024

**SRAM**
- Tech: 45nm
- Width: 16

**MAC**
- Tech: 45nm
- Width: 16

**MAC**
- Tech: 65nm
- Width: 16

ERT/ART Generator

Primitive Component Library

Energy Calculator

Action Counts

<table>
<thead>
<tr>
<th>Comp.</th>
<th>Action</th>
<th>Counts</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLB</td>
<td>access()</td>
<td>10</td>
</tr>
<tr>
<td>buffer</td>
<td>access()</td>
<td>800</td>
</tr>
<tr>
<td>MAC</td>
<td>compute()</td>
<td>400</td>
</tr>
</tbody>
</table>

Simple Example Estimation Plug-in

<table>
<thead>
<tr>
<th>Name</th>
<th>Energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLB</td>
<td>1000pJ</td>
</tr>
<tr>
<td>buffer</td>
<td>8000pJ</td>
</tr>
<tr>
<td>MAC</td>
<td>1200pJ</td>
</tr>
</tbody>
</table>

Energy Estimates

<table>
<thead>
<tr>
<th>Class</th>
<th>Tech.</th>
<th>Width</th>
<th>Depth</th>
<th>Action</th>
<th>Energy (pJ)</th>
<th>Area (um²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAC</td>
<td>45nm</td>
<td>16b</td>
<td>N/A</td>
<td>compute</td>
<td>5</td>
<td>0.4</td>
</tr>
<tr>
<td>SRAM</td>
<td>45nm</td>
<td>64b</td>
<td>1024</td>
<td>access</td>
<td>100</td>
<td>20</td>
</tr>
<tr>
<td>SRAM</td>
<td>45nm</td>
<td>16b</td>
<td>256</td>
<td>access</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>MAC</td>
<td>65nm</td>
<td>16b</td>
<td>N/A</td>
<td>compute</td>
<td>3</td>
<td>0.3</td>
</tr>
</tbody>
</table>
Accelergy Overview

• Accelergy Infrastructure
  – Performs architecture-level estimations to enable rapid design space exploration
  – Supports modeling of diverse architectures with various underlying technologies
  – Improves estimation accuracy by allowing fine-grained classification of components runtime behaviors
  – Supports succinct modeling of complicated architectures

• Validation on various accelerator designs
  – 95% accurate on a conventional digital accelerator design
  – Modeling of processing in memory (PIM) based DNN accelerator designs
Plug-ins for Fine-Grain Action Energy Estimation

- External energy/area models that accurately reflect the properties of a macro
  - e.g., multiplier with zero-gating

Energy characterizations of the zero-gated multiplier (normalized to idle)

<table>
<thead>
<tr>
<th>name</th>
<th>tech.</th>
<th>width</th>
<th>action</th>
<th>energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>multiplier</td>
<td>65nm</td>
<td>16b</td>
<td>random multiply</td>
<td>23.0</td>
</tr>
<tr>
<td>multiplier</td>
<td>65nm</td>
<td>16b</td>
<td>reused multiply</td>
<td>16.8</td>
</tr>
<tr>
<td>multiplier</td>
<td>65nm</td>
<td>16b</td>
<td>gated multiply</td>
<td>1.3</td>
</tr>
</tbody>
</table>
Plug-ins for Fine-Grain Action Energy Estimation

- External energy/area models that accurately reflect the properties of a macro
  - e.g., multiplier with zero-gating

Energy characterizations of the zero-gated multiplier (normalized to idle)

With the characterization provided in the plug-in, we can see significant energy savings for sparse workloads.
Plug-ins for Fine-Grain Action Energy Estimation

- External energy/area models that accurately reflect the properties of a macro
  - e.g., multiplier with zero-gating

Energy characterizations of the zero-gated multiplier (normalized to idle)

With the characterization provided in the plug-in, we can see significant energy savings for sparse workloads
Plug-ins for Fine-Grain Action Energy Estimation Plug-ins

- External energy/area models that accurately reflect the properties of a macro
  - e.g., register file with various access types

Energy-Per-Actions of a Register File (normalized to idle)

With the characterization provided in the plug-in, we can see accurate characterization for memories with different access patterns.
Use energy estimation plug-ins to characterize primitive components

- **CACTI Estimation Plug-in**
- **45nm Estimation Plug-in**
- **Proprietary plug-ins**
- **Emerging technology plug-ins**

Traditional open-source plug-ins

Proprietary plug-ins

Emerging technology plug-ins

NVSIM [TCAD 2012]
Accelergy: Flexibly Model Various Primitive Components

**Architecture Description**

- **GLB**
  - tech: 45nm
  - width: 64
  - depth: 1024
- **SRAM**
- **MAC**
  - tech: 45nm
  - width: 16
  - depth: 256

**ERT/ART Generator**

- **GLB**
  - access(): 10pJ, 20μm²
- **buffer**
  - access(): 10pJ, 2μm²
- **MAC**
  - compute(): 5pJ, 0.4μm²

**Primitive Component Library**

- **GLB**
  - tech: 45nm, width: 64
  - depth: 1024
- **buffer**
  - tech: 45nm, width: 16
- **MAC**
  - tech: 45nm, width: 16
  - depth: 256

**Energy Calculator**

- **GLB**
  - 1000pJ
- **buffer**
  - 8000pJ
- **MAC**
  - 2000pJ

**Action Counts**

- **Comp.**
  - **GLB**
    - access(): 10
  - **buffer**
    - access(): 800
  - **MAC**
    - compute(): 400

**Estimation Plug-ins**

- **CACTI**
- **45nm**

**Energy Estimates**

- **Name**
  - **GLB**
    - 1000pJ
  - **buffer**
    - 8000pJ
  - **MAC**
    - 2000pJ
• Practical designs involve many more primitive components
  – Example: smartbuffer – a storage unit with preprogrammed address generators (AGs)
    – Domain-specific applications have predictable storage access patterns, allowing offline access stream generation, e.g., general matrix multiply applications.
  
  o buffer belongs to SRAM class
  o AGs belongs to adder class
Modeling Complicated Designs

• Practical designs involve many more primitive components

Simple Architecture Design

Let’s construct a more practical design!
Modeling Complicated Designs

- Practical designs involve many more primitive components

Practical Architecture Design

Let’s construct a more practical design!
• Practical designs involve many more primitive components
Modeling Complicated Designs

- Architecture description is tedious
- Hard to make modifications
Modeling Complicated Designs

Architecture Description

• Architecture description is tedious
• Hard to make modifications

Action Counts

<table>
<thead>
<tr>
<th>component</th>
<th>action</th>
<th>counts</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLB</td>
<td>access()</td>
<td>10</td>
</tr>
<tr>
<td>AG[0]</td>
<td>add()</td>
<td>7</td>
</tr>
<tr>
<td>AG[1]</td>
<td>add()</td>
<td>3</td>
</tr>
<tr>
<td>PE0.buffer</td>
<td>access()</td>
<td>800</td>
</tr>
<tr>
<td>PE0.AG[0]</td>
<td>add()</td>
<td>680</td>
</tr>
<tr>
<td>PE0.AG[1]</td>
<td>add()</td>
<td>120</td>
</tr>
<tr>
<td>PE0.MAC</td>
<td>compute()</td>
<td>370</td>
</tr>
<tr>
<td>PE0.FIFO</td>
<td>access()</td>
<td>370</td>
</tr>
<tr>
<td>PE1.buffer</td>
<td>access()</td>
<td>830</td>
</tr>
<tr>
<td>PE1.AG[0]</td>
<td>add()</td>
<td>690</td>
</tr>
</tbody>
</table>

Estimation Plug-ins
Existing Architecture-Level Energy Estimators

- Architecture-level energy modeling for general purpose processors
  - Wattch[Brooks, ISCA2000], McPAT[Li, MICRO2009], GPUWattch[Leng, ISCA2013], PowerTrain[Lee, ISLPED2015]

CPU/GPU-Centric Architecture Model

Use a fixed set of **compound components** to represent the architecture

Components that can be decomposed into lower level components
Existing Architecture-Level Energy Estimators

• Architecture-level energy modeling for general purpose processors
  – Wattch[Brooks, ISCA2000], McPAT[Li, MICRO2009], GPUWattch[Leng, ISCA2013], PowerTrain[Lee, ISLPED2015]

The fixed set of compound components is not sufficient to describe various optimizations in the diverse accelerator design space
• Accelergy Infrastructure
  – Performs architecture-level estimations to enable rapid design space exploration
  – Supports modeling of diverse architectures with various underlying technologies
  – Improves estimation accuracy by allowing fine-grained classification of components runtime behaviors
  – Supports succinct modeling of complicated architectures

• Validation on various accelerator designs
  – 95% accurate on a conventional digital accelerator design
  – Modeling of processing in memory (PIM) based DNN accelerator designs
Accelergy: Succinctly Model Arbitrary Architecture

• Allow succinct architecture description with user-defined compound component classes
Accelergy: Succinctly Model Arbitrary Architecture

- Allow succinct architecture description with user-defined compound component classes
- Allow user-defined compound component hardware structure using primitive components
Accelrys: Succinctly Model Arbitrary Architecture

- Allow succinct architecture description with user-defined compound component classes
- Allow user-defined compound component hardware structure using primitive components
- Allow user-defined compound component actions using primitive component actions
Accelergy: Succinctly Model Arbitrary Architecture

- Flexible and succinct architecture representations using user-defined compound components
Accelergy: Succinctly Model Arbitrary Architecture

- Flexible and succinct action counts using compound actions

![Accelergy Diagram]

<table>
<thead>
<tr>
<th>Component</th>
<th>Action</th>
<th>Counts</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLB</td>
<td>read()</td>
<td>10</td>
</tr>
<tr>
<td>PE0.buffer</td>
<td>read()</td>
<td>800</td>
</tr>
<tr>
<td>PE0.MAC</td>
<td>compute()</td>
<td>370</td>
</tr>
<tr>
<td>PE1.buffer</td>
<td>read()</td>
<td>830</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Accelergy High-Level Infrastructure

Architecture Description → ERT/ART Generator → Primitive Component Library → Energy Calculator

Compound Component Description → ERT ART

Estimation Plug-ins
- CACTI Estimation Plug-in
- 45nm Estimation Plug-in
- ... Energy Estimations

Available at http://accelergy.mit.edu/
More details about the syntax for the input and output files will be presented during the hands-on session.

Available at http://accelergy.mit.edu/
Accelergy Overview

• Accelergy Infrastructure
  – Performs architecture-level estimations to enable rapid design space exploration
  – Supports modeling of diverse architectures with various underlying technologies
  – Improves estimation accuracy by allowing fine-grained classification of components runtime behaviors
  – Supports succinct modeling of complicated architectures

• Validation on various accelerator designs
  – 95% accurate on a conventional digital accelerator design
  – Modeling of processing in memory (PIM) based DNN accelerator designs
Energy Validation on Eyeriss [Chen, ISSCC 2016]

- **Experimental Setup:**
  - Workload: Alexnet weights & ImageNet input feature maps
  - Ground Truth: Energy obtained from post-layout simulations

---

**Eyeriss Architecture**

- GLBs
- Shared GLB
- Weights GLB
- IfmapNoC
- PsumWrNoC
- WeightsNoC
- PsumRdNoC

---

PE array 12x14

- PE
- Psum
- MAC

---

Ifmap = input feature map
Psum = partial sum
PE = processing element
*_spad = *_scratchpad
Energy Validation on Eyeriss [Chen, ISSCC 2016]

- Experimental Setup:
  - Workload: Alexnet weights & ImageNet input feature maps
  - Ground Truth: Energy obtained from post-layout simulations

**Zero-gating optimization**

If there is a 0 ifmap data
- Gate on reading the weights data => gated-read
- Gate on computing the MAC => gated-MAC
Energy Validation on Eyeriss [Chen, ISSCC 2016]

• Total energy estimation is 95% accurate of the post-layout energy.

• Estimated relative breakdown of the important units in the design is within 8% of the post-layout energy.

*Total energy might not add up to exact 100.0% due to rounding
PE Array Energy Breakdown

- Comparisons with existing work: Aladdin[Shao, ISCA2014]

Energy Breakdown of PEs across the Array

Energy impact of sparsity is accurately captured with sparsity-aware estimation plug-ins.
Accelergy Overview

• Accelergy Infrastructure
  – Performs architecture-level estimations to enable rapid design space exploration
  – Supports modeling of diverse architectures with various underlying technologies
  – Improves estimation accuracy by allowing fine-grained classification of components runtime behaviors
  – Supports succinct modeling of complicated architectures

• Validation on various accelerator designs
  – 95% accurate on a conventional digital accelerator design
  – Modeling of processing in memory (PIM) based DNN accelerator designs
Accelergy Modeling of PIM Architectures

- Example PIM DNN architectures

- Activation is input voltage ($V_i$)
- Weight is resistor conductance ($G_i$) [= 1/resistance]
- Partial sum is output current ($I_i$)

$$I = I_1 + I_2 = V_1 \times G_1 + V_2 \times G_2$$
Estimation for PIM Accelerators

Architecture Description

Compound Component Description
Estimation for PIM Accelerators

Architecture Description

Example Estimation Plug-in

<table>
<thead>
<tr>
<th>class</th>
<th>tech.</th>
<th>width</th>
<th>action</th>
<th>energy (pJ)</th>
<th>area ((\mu m^2))</th>
</tr>
</thead>
<tbody>
<tr>
<td>RRAM_cell</td>
<td>45nm</td>
<td>1b</td>
<td>mac</td>
<td>1.46E-2*</td>
<td>1.21E-2*</td>
</tr>
<tr>
<td>ADC</td>
<td></td>
<td></td>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DAC</td>
<td></td>
<td></td>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* scaled from ISAAC [Shafiee, MICRO 2016]
Estimation for PIM Accelerators

Architecture Description

Example Estimation Plug-in

Action Counts

<table>
<thead>
<tr>
<th>name</th>
<th>action</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>PE0.MAC</td>
<td>compute</td>
<td>1000</td>
</tr>
<tr>
<td>PE1.MAC</td>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>

Energy/Area Estimation

* scaled from ISAAC [Shafiee, MICRO 2016]

<table>
<thead>
<tr>
<th>name</th>
<th>energy (pJ)</th>
<th>area ((\mu m^2))</th>
</tr>
</thead>
<tbody>
<tr>
<td>PE0.MAC</td>
<td>14.6</td>
<td>1.21E-2</td>
</tr>
<tr>
<td>PE1.MAC</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Compound Component Description

<table>
<thead>
<tr>
<th>class</th>
<th>tech.</th>
<th>width</th>
<th>action</th>
<th>energy (pJ)</th>
<th>area ((\mu m^2))</th>
</tr>
</thead>
<tbody>
<tr>
<td>RRAM_cell</td>
<td>45nm</td>
<td>1b</td>
<td>mac</td>
<td>1.46E-2*</td>
<td>1.21E-2*</td>
</tr>
<tr>
<td>ADC</td>
<td></td>
<td></td>
<td></td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>DAC</td>
<td></td>
<td></td>
<td></td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>
Accelergy Modeling of PIM Architectures

• Parameterizable templates
  – Architecture Template allows architecture parameter sweeping, e.g.,
    • number of PE rows
    • number of PE columns
    • size of global buffer, etc.
  – Component design template allows implementation optimization, e.g.,
    • optimize DAC-based D2A conversion system
    • optimize the design of the flash ADC in the A2D conversion system, etc.
Energy Modeling Validation on PIM Design

• Validation on the ADC-based design proposed in CASCADE [Chou, MICRO2019]

• Design Specs
  – 80 64x64 1-bit Memristor Arrays
  – 1-bit DACs
  – 6-bit ADCs
  – 16-bit data representations

• Workload: VGG Net convolutional layers

• Energy estimation tables: extracted numbers from the paper/cited sources
Total Energy Estimation and Breakdown Validation

The architecture is correctly modeled:
- 95% accurate total energy estimation
- tracks the breakdown across different components

Total energy might not add up to exact 100.0% due to rounding
Energy Modeling Validation on PIM Design

Energy Breakdown Across VGG Convolutional Layers

Captures the energy breakdown of each convolutional layer

Published at [Wu, ISPASS 2020]
Summary

• Accelergy is an architecture-level energy estimator that
  – Accelerates accelerator design space exploration
  – Provides flexibility to
    • Describe and evaluate a wide range of accelerator designs
    • Support different technologies with user defined plug-ins, e.g., CMOS, RRAM, etc.
  – Achieves high accuracy energy estimations
    • 95% accurate for the Eyeriss accelerator and Cascade PIM accelerator

• The Timeloop-Accelergy system allows fast explorations on
  – High-level architecture properties, e.g., PE array size
  – Lower-level implementation optimizations on the components in the design, e.g., storage designs with local address generation

Acknowledgement: DARPA, Facebook, MIT Presidential Fellowship
Resources

• Tutorial Related
  – Tutorial Website: http://accelergy.mit.edu/isca20_tutorial.html
  – Tutorial Docker: https://github.com/Accelergy-Project/timeloop-accelergy-tutorial
    • Various exercises and example designs and environment setup for the tools

• Other
  – Infrastructure Docker: https://github.com/Accelergy-Project/accelergy-timeloop-infrastructure
    • Pure environment setup for the tools without exercises and example designs
  – Open Source Tools
    • Accelergy: http://accelergy.mit.edu/
    • Timeloop: https://github.com/NVlabs/timeloop
  – Papers: