Improving CGRA Energy Efficiency with Low-Overhead Fine-Grained Power Domains

Ankita Nayak

AHA Affiliates Meeting
July 13, 2022

Collaborators: Keyi Zhang, Raj Setaluri, Alex Carsello, Makai Mann, Christopher Torng, Stephen Richardson, Rick Bahr, Pat Hanrahan, Mark Horowitz, Priyanka Raina
Need for Energy Efficient Programmable Architectures

Many modern-day applications demand energy efficiency as well as flexibility
Space of Programmable Machines

Island-Style CGRA

- **Island-style CGRA**: 2D array of processing element (PE) tiles, memory (MEM) tiles, interconnect with horizontal and vertical routing tracks

- **Switch boxes (SBs)**: Implement connections between any two tiles

- **Connection boxes (CBs)**: Select PE/MEM inputs from routing tracks
Agenda

**Fine-Grain Power Domains**

- Improves energy-efficiency of CGRA with low-overhead fine-grained power domains

**Apply Fine-Grain Power Domains to Amber SoC**

- SoC with CGRA with Power domains
Agenda

Fine-Grain Power Domains

• Improves energy-efficiency of CGRA with low-overhead fine-grained power domains

Apply Fine-Grain Power Domains to Amber SoC

• SoC with CGRA with Power domains
Power Domains

- Collection of gates driven by same power and ground
  - Can be individually turned *on* or *off*

- Turn off power to part of chip not actively used
  - Saves leakage power
Why Power Domains?

- Improve energy efficiency by lowering transistor $V_{\text{th}}$
  - Same performance at lower $V_{\text{dd}}$
    - Lowers dynamic power

- If unit not active all the time
  - Dynamic power goes to zero
    - Want (need) leakage to go to zero too
  - Requires power gating to effectively manage the leakage power
Power Domains in CGRAs

Ideally add power switches to each tile and make it its own power domain!
Challenges in Introducing Fine-Grained Power Domains

Driving logic is powered down - off domain outputs may float between 0 and 1
Challenges in Introducing Fine-Grained Power Domains

Power domain boundaries need special isolation logic
Introducing Boundary Protection in an Island-Style CGRA

- Power domain boundaries need special isolation logic
  - Makes fine-grained power domains area- and timing-inefficient
Prior Work

- Turn off unused coarse functional units in FPGA/CGRA
  - Miniskar et al., Intra mode power saving methodology for CGRA-based reconfigurable processor architectures
  - Lopes et al., Evaluation of CGRA architecture for real-time processing of biological signals on wearable devices
  - Korol et al., MCEA: A Resource-Aware Multicore CGRA Architecture for the Edge

- Fine-grained power gating in FPGA
  - Ishihara et al., A low-power FPGA based on autonomous fine-grain power gating
  - Lin et al., Routing track duplication with fine-grained power-gating for FPGA interconnect power reduction
Introducing Boundary Protection in an Island-Style CGRA
Low-Overhead Boundary Protection Mechanism

All data inputs coming into a tile go only into SBs and CBs
Low-Overhead Boundary Protection Mechanism

Normal Operation Mode: Retain default mux functionality

Isolation Mode: If any SB/CB input is X:
1. SB/CB outputs always generate 0/1
2. All gate outputs in the mux must also generate 0/1
Redesign the multiplexers in the design

**Normal Operation Mode:** Regular Functionality

**Isolation Mode:** Clamping Functionality

---

Modified SB/CB Functionality in the Tile

Re-architected SB/CBs block X-propagation from off tiles
SMT-based Formal Verification

- Need technique to prove the mux transformations are correct
  - Use SMT-based model checker

- Challenge: Formal tools do not have a notion of X values
  - Need to formalize concept of X-propagation

- Encode SB/CB output conditions in isolation mode as properties in model checker
Compiler-like Framework for Power Domain Insertion

- **Motivation**: Create framework that incrementally makes design transformations to introduce power domains using compiler-like “passes”

- Leverage Domain Specific Language (DSL) (magma) to write passes for design transformations
Boundary Protection Circuit Transformation Pass
Power Switch Configuration Insertion Pass

- CGRA Tile
- ps_en_reg
- PS
- PS
- PS
- PS
- PS
- PS
Configurable Fine-Grained Power Domains

P&R tool has freedom to choose any ON-OFF topology
Handling Global Signals

Global signals that flow from top to bottom are impacted when top tiles are off.
Debug Signal Isolation Pass

All tiles are on

Stanford University
End-to-End Flow
Results – Area Savings & Performance Improvement

- Boundary Protection Mechanism
  - Baseline: No boundary protection
  - Conventional Method: Isolation cell-based
  - Proposed Method: Circuit transformation-based

- Area Savings

<table>
<thead>
<tr>
<th>Boundary Protection Techniques</th>
<th>PE</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Power Domains (PD)</td>
<td>1x</td>
<td>1x</td>
</tr>
<tr>
<td>PDs with Conventional Isolation Method</td>
<td>+9.2%</td>
<td>+5.7%</td>
</tr>
<tr>
<td>PDs with Our Method</td>
<td>+0.73%</td>
<td>+0.64%</td>
</tr>
</tbody>
</table>

Additional 4-6% area overhead from power switches for designs with power domains

- Performance Improvement
  - Conventional Method: 475 MHz
  - Proposed Method: 500+ MHz (No additional isolation logic on critical paths)
## Results – Power Savings

<table>
<thead>
<tr>
<th>Applications</th>
<th>% Tiles Utilized</th>
<th>Total Power Reduction</th>
<th>Leakage Power Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PE</td>
<td>Memory</td>
<td></td>
</tr>
<tr>
<td>CONV 1x2</td>
<td>17.7%</td>
<td>9.3%</td>
<td>26.2%</td>
</tr>
<tr>
<td>CONV 3x3</td>
<td>31.3%</td>
<td>29.7%</td>
<td>12.2%</td>
</tr>
<tr>
<td>Gaussian</td>
<td>35.1%</td>
<td>31.3%</td>
<td>10.8%</td>
</tr>
<tr>
<td>Demosiac</td>
<td>35.9%</td>
<td>33.6%</td>
<td>10.2%</td>
</tr>
<tr>
<td>Cascade</td>
<td>39.3%</td>
<td>37.5%</td>
<td>8.9%</td>
</tr>
<tr>
<td>DNN Conv</td>
<td>55.2%</td>
<td>53.0%</td>
<td>4.9%</td>
</tr>
<tr>
<td>Corner Detection</td>
<td>58.8%</td>
<td>54.7%</td>
<td>4.4%</td>
</tr>
<tr>
<td>Camera Pipe</td>
<td>90.0%</td>
<td>85.0%</td>
<td>0.8%</td>
</tr>
</tbody>
</table>

- **Leakage savings:**
  - **Off** PE tile: 22x
  - **Off** Mem tile: 46x

- **Leakage reduction:**
  - Up to 83%

- **Total power reduction:**
  - Up to 26%
Agenda

Fine-Grain Power Domains

- Improves energy-efficiency of CGRA with low-overhead fine-grained power domains

Apply Fine-Grain Power Domains to Amber SoC

- SoC with CGRA with Power domains
Amber SoC

- SoC: Processor, global buffer, SRAMs, 32x16 CGRA with PE and Mem Tiles
- Technology node: TSMC 16nm
CGRA in Amber SoC

- Fine-grained power gating for PE and MEM tiles
- Each tile is partitioned into power domains
  - ON: Logic for power switch configuration, debug circuits, tile ID cells
  - OFF: Rest of the logic

Design Consideration #1: Power Grid Implementation

- Homogeneous power grid common to both domains
  - Common IR analysis over two domains

- Horizontal VDD_SW stripes extended only till core edge of tile
  - No abutment to neighboring tile: Each tile can be individually turned on/off
Design Consideration #2: Power Switch (PS) Insertion

- Choose simple daisy chaining
  - Small power domains not sensitive to wake-up time

- Choose PS count that meets IR budget

- Choose PS insertion pitch and offset such that they overlap with vertical VDD stripes
Design Consideration #3: Well Substrate Connection

Connect to VDD through tap cells

Insert tap cells in boundary rows without power switches

Connect substrate to VDD with power switches instead of tap cells

AON Region

Switching Region

Stanford University
Power Domain Silicon Results

- Power domain-related features works on silicon as expected
  - Turning off unused tiles
  - Turning on off tiles through config registers
  - Global signal integrity when tiles are off

- Low leakage contribution in this chip
  - Next chip – use more low Vt cells to improve performance
  - Recover power through power gating
Conclusion

- CGRAs are still not as efficient as ASICs

- In addition to application-specific optimization explore generic optimization methods as well

- End-to-end system makes finding and testing these optimizations simpler
Thank You!