Snatch: Opportunistically Reassigning Power Allocation between Processor and Memory in 3D Stacks

Dimitrios Skarlatos, Renji Thomas, Aditya Agrawal, Shibin Qin, Robert Pilawa, Ulya Karpuzcu, Radu Teodorescu, Nam Sung Kim, and Josep Torrellas

UIUC, OSU, UMN, NVIDIA
Motivation: Cost of Power/Ground Pins in 3D stacks
Motivation: Cost of Power/Ground Pins in 3D stacks

- Size & cost of packages is proportional to # of pins
Motivation: Cost of Power/Ground Pins in 3D stacks

- Size & cost of packages is proportional to # of pins
- 3D Stacks: Disjoint Power/Ground pins for Processor and Memory
Motivation: Cost of Power/Ground Pins in 3D stacks

- Size & cost of packages is proportional to # of pins
- 3D Stacks: Disjoint Power/Ground pins for Processor and Memory
- Each dimensioned for the worst case
Motivation: Underutilization of Power Budget

- High Processor *or* Memory Power phases
Contribution: *Snatch*

- Dynamically and opportunistically divert power between processor and memory
Contribution: Snatch

- Dynamically and opportunistically divert power between processor and memory
- On-chip voltage regulator connects the two Power Delivery Networks
- Processor or Memory can consume more power for the same # of pins
Impact Compared to Conventional 3D Stacks

- For same # of power/ground pins:
  - Application can consume more power
  - Up to 23% application speedup
- For the same maximum power in Processor and Memory
  - Fewer pins, about 30% package cost reduction
Snatch Outline

- Implementation
- Operation
- Case 1:
  - Same Max Power in Processor and Memory, reduced # of pins
- Case 2:
  - Same # of pins, improved performance
- Evaluation
Snatch Outline

• Implementation

• Operation

• Case 1:
  • Same Max Power in Processor and Memory, reduced # of pins

• Case 2:
  • Same # of pins, improved performance

• Evaluation
Conventional Implementation

- **0.8-0.95V**: Processor die
- **1.1V**: Memory VR
- **5.5W**: DRAM0 die
- **4.5W**: DRAM1 die
- **12V**: BGA pins
- **12V**: C4 bumps
- Cross-section

Cross-section
Snatch implementation

Cross-section
Snatch implementation

• Small 2W on-chip bidirectional VR on Proc die

• Bulk of work from off-chip VRs
**Snatch**: Dynamic power reassignment

- Up/Down convert power *Snatched*
**Snatch**: Dynamic power reassignment

- Up/Down convert power *Snatched*

![Diagram showing power reassignment](image)

- Single On-Chip VR

- TSVs

- C4 bumps

- BGA pins

- Memory VR

- Processor VR

- DRAM0 die

- DRAM1 die

- Processor die

- PCB substrate
**Snatch: Cross-Section**

- Small 2W on-chip *bidirectional* VR on Proc die

Cross-section
**Snatch: Top Down**

- Small 2W on-chip *bidirectional* VR on Proc die

Cross-section: Single On-Chip VR

Top Down: 0.8-0.95V

- Processor VR: 5.5W
- Memory VR: 4.5W
- Processor VR: 1.1V
Snatch Outline

- Implementation
- Operation

Case 1:
- Same Max Power in Processor and Memory, reduced # of pins

Case 2:
- Same # of pins, improved performance

- Evaluation
Snatching Memory Power

- On processor intensive phase
Snatching Memory Power

- On **processor intensive** phase

- Snatch **Memory** Power → TurboBoost **Processor**
Snatching Processor Power

- On memory intensive phase

- Snatch Processor Power → TurboBoost Memory
**Snatching Decisions**

- Processor or Memory Intensive Phase?
Snatching Decisions

- Processor or Memory Intensive Phase?
- How much Power is available?
Snatching Decisions

- Processor or Memory Intensive Phase?
- How much Power is available?
- How much Power can we Snatch?
Conservative *Snatching* Algorithm

- Keep track of past power values of 10µs epochs
Conservative **Snatching** Algorithm

- Keep track of past power values of 10µs epochs
- Average for activity detection
Conservative *Snatching* Algorithm

- Keep track of past power values of 10µs epochs
- Average for activity detection
- MAX for power availability
Conservative *Snatching* Algorithm

- Keep track of past power values of 10µs epochs
- Average for activity detection
- MAX for power availability
- Avoid hysteresis
Snatch Outline

• Implementation

• Operation

• Case 1:
  • Same Max Power in Processor and Memory, reduced # of pins

• Case 2:
  • Same # of pins, improved performance

• Evaluation
Conventional Power Provisioning

- Processor provisioned for 7.5W
Conventional Power Provisioning

- Processor provisioned for **7.5W**
**Conventional Power Provisioning**

- Processor provisioned for **7.5W**
- Memory provisioned for **6.5W**
Conventional Power Provisioning

- Processor provisioned for 7.5W
- Memory provisioned for 6.5W
- Total = Processor + Memory = 14W
Snatch: Provisioning 3D Stacks Just Right

- Processor provisioned for 7.5W
- Memory provisioned for 6.5W
- Total = Processor + Memory = 14W
Snatch: Provisioning 3D Stacks Just Right

- Processor provisioned for 7.5W
- Memory provisioned for 6.5W
- Total = Processor + Memory = 14W
Snatch: Provisioning 3D Stacks Just Right

- Processor provisioned for 7.5W - 2W = 5.5W
- Memory provisioned for 6.5W
- Total = Processor + Memory = 14W
Snatch: Provisioning 3D Stacks Just Right

- Processor provisioned for $7.5W - 2W = 5.5W$
- Memory provisioned for $6.5W - 2W = 4.5W$
- Total = Processor + Memory = $14W$
Snatch: Provisioning 3D Stacks Just Right

- Processor provisioned for 7.5W - 2W = 5.5W
- Memory provisioned for 6.5W - 2W = 4.5W
- Total = Processor + Memory = 14W - 4W = 10W

Reduce Total Provisioning from 14W to 10W, approx same performance.
Snatch: Provisioning 3D Stacks Just Right

- Processor provisioned for 7.5W - 2W = 5.5W
- Memory provisioned for 6.5W - 2W = 4.5W
- Total = Processor + Memory = 14W - 4W = 10W

Reduce Total Provisioning from 14W to 10W, approx same performance

30% Reduction in Package Power/Ground Pins
Snatch Outline

- Implementation
- Operation

- Case 1:
  - Same Max Power in Processor and Memory, reduced # of pins

- Case 2:
  - Same # of pins, improved performance

- Evaluation
Conventional Power Provisioning

- Processor & Memory provisioned for **5.5W** & **4.5W**
Conventional Power Provisioning

• Processor & Memory provisioned for 5.5W & 4.5W
Snatch: Provide Additional Power

- Processor & Memory provisioned for **5.5W & 4.5W**
- *Snatch* power
Snatch: Opportunistically boost performance

- Processor & Memory provisioned for 5.5W & 4.5W
- *Snatch* power and boost performance
**Snatch: Boost Performance with Same # of Pins**

- Processor & Memory provisioned for **5.5W & 4.5W**
- *Snatch* power and boost performance
- Same # of pins as conventional
Snatch: Boost Performance with Same # of Pins

- Processor & Memory provisioned for **5.5W** & **4.5W**
- *Snatch* power and boost performance
- Same # of pins as conventional

**Higher performance for the same package cost**
**Snatch: Boost Performance with Same # of Pins**

- Processor & Memory provisioned for **5.5W** & **4.5W**
- *Snatch* power and boost performance
- Same # of pins as conventional

Higher performance for the same package cost
IR-drop and EM characteristics remain the same
Snatch Outline

- Implementation
- Operation
- Case 1:
  - Same Max Power in Processor and Memory, reduced # of pins
- Case 2:
  - Same # of pins, improved performance
- Evaluation
Evaluation Methodology

• Case 2: **Same # of pins**, improved performance
  
  • Processor: 22nm LP 8-core w/ SESC + McPAT
  
  • Memory: 4GB 2-layer WideIO2 w/ DRAMSim2
  
  • Benchmarks: SPLASH-2, NAS, and SPEC
Performance

<table>
<thead>
<tr>
<th>Speedup</th>
<th>Average(Splash + NAS)</th>
<th>Average(SPEC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Performance

Baseline

P = {5.5W, 1.2GHz}
M = {4.5W, 400MHz}

Average(Splash + NAS)
Average(SPEC)
Performance

Baseline
- $P = \{5.5W, 1.2GHz\}$
- $M = \{4.5W, 400MHz\}$

Turbo Boost
- $P = \{5.5W, 1.2-1.5GHz\}$
- $M = \{4.5W, 400-900MHz\}$
- DVFS Within Power Budget
Performance

**Baseline**
- $P = \{5.5W, 1.2GHz\}$
- $M = \{4.5W, 400MHz\}$

**Turbo Boost**
- $P = \{5.5W, 1.2-1.5GHz\}$
- $M = \{4.5W, 400-900MHz\}$
-DVFS Within Power Budget

**Snatch**
- $P = \{5.5W, 1.2-1.5GHz\}$
- $M = \{4.5W, 400-900MHz\}$
- DVFS and Snatch up to 2W
Snatch boosts performance on average, by 25% against Baseline and 8% against Turbo Boost for Splash and NAS benchmarks.
Snatch boosts performance on average, by 10% against Baseline for SPEC benchmarks. Negligible gains against Turbo Boost.

**Baseline**
- $P = \{5.5W, 1.2GHz\}$
- $M = \{4.5W, 400MHz\}$

**Turbo Boost**
- $P = \{5.5W, 1.2-1.5GHz\}$
- $M = \{4.5W, 400-900MHz\}$
- DVFS and Snatch up to 2W

**DVFS and Snatch**

**Average (Splash + NAS)**
- Baseline
- Turbo Boost
- Snatch

**Average (SPEC)**
- Baseline
- Turbo Boost
- Snatch

**Speedup**

- 0.4
- 0.6
- 0.7
- 0.9
- 1.0
- 1.2
- 1.3

**Turbo Boost**

- 25% increase
- 10% increase

**Snatch**

- 8% increase
Snatching Activity Overview

% of Total Time Snatching

0 22.5 45 67.5 90

Barnes BT CG Cholesky FFT FMM FT IS LU LU(NAS) MG Radiosity Radix Raytrace SP W-Nsquared W-Spatial AvgM->P AvgP->M mcf mile IBM bzip2

57
Snatching Activity Overview

% of Total Time Snatching

Barnes  BT  CG  Cholesky  FFT  FMM  FT  IS  LU  LU(NAS)  MG  Radiosity  Radix  Raytrace  SP  W-Nsquared  W-Spatial  AvgM->P  mcf  mile  bzip2  AvgP->M
Snatching Activity Overview

% of Total Time Snatching

- Barnes
- BT
- CG
- Cholesky
- FFT
- FMM
- FT
- IS
- LU
- LU(NAS)
- MG
- Radiosity
- Radix
- Raytrace
- SP
- W-Nsquared
- W-Spatial
- AvgM→P
- mcf
- mile
- ibm
- bzip2
- AvgP→M
Snatching Activity Overview

Application Snatch on average, 30% for Splash+NAS
9.4% for SPEC
More On the Paper

- Design and Implementation:
  - On-chip Voltage Regulator
  - *Snatch* Algorithm
- Additional Evaluation:
  - *Snatch* Algorithm
  - Power Delivery Network
  - Pin Reliability
  - 3D Stack Thermals
Summary

- **Snatch**: An opportunistic power reassignment design for 3D Stacked architectures
  - Small on-chip bidirectional VR
  - Processor - Memory phase detection and power availability estimation
  - Up to 23% application speedup
  - Alternatively, about 30% package cost reduction