Comparative Analysis of Contemporary Cache Power Reduction Techniques

Ph.D. Dissertation Proposal
Samuel V. Rodriguez
Motivation

Power dissipation is important across the board, not just portable devices!!

Portable Devices

Mid-end (e.g. Desktops)

High-end (e.g. servers)
Motivation

- Thermal Design Power (TDP) is now a priority specification
- AMD currently can’t compete in “Thin and light” notebooks because of their higher TDP’s
- AMD’s power advantage in initial dual-core offerings
- An entire Intel Pentium 4 design recently cancelled because of higher than expected TDP’s
• Breakdown of power consumption for a 4-wide 200MHz 3.3V 0.35um processor with 32kB/32kB/1MB caches
Motivation

• Fraction of die area and transistor count dedicated to caches is increasing

Photograph taken from Weiss2002
Presentation Outline

• Motivation (finished)
• Background
  – Power Dissipation
  – Cache/SRAM Implementation
• Contemporary Cache Power Reduction Schemes
• Proposed Work
• Q&A
Background (Power Dissipation)

- Need to account for both dynamic and static power dissipation!

Graph from Kim2004
Background (Power Dissipation)

- Causes of dynamic power
Background (Power Dissipation)

- Causes of dynamic power
Background (Power Dissipation)

- Causes of dynamic power

Diagram:
- Isc
- Idischarge
- Cloud
Background (Power Dissipation)

- \( \text{Power}_{\text{dyn}} \propto N \times C \times V_{DD}^2 \times f \)
  - \( \uparrow \uparrow \uparrow \) \( N \): Number of transistors
  - \( \downarrow \downarrow \) \( C \): Device capacitance
  - \( \downarrow \) \( V_{DD} \): Supply voltage
  - \( \uparrow \uparrow \) \( f \): Frequency

- Dynamic power trend: slow increase
Background (Power Dissipation)

- Causes of static power: leakage currents
Background (Power Dissipation)

- Subthreshold: 5x per generation
- Gate leakage: 500x per generation!!!
Background (Power Dissipation)

- Subthreshold leakage is increasing:

\[ \text{Id,sat} \propto (V_{gs} - V_{th}) = (V_{DD} - V_{th}) \]

- Increase: 5x per generation
Background (Power Dissipation)

- Gate leakage

Tox scaling resulting in increased gate leakage caused by oxide tunneling

- Gate leakage: \( 500x \) per generation!!!
Presentation Outline

- Motivation (finished)
- **Background**
  - Power Dissipation
  - Cache/SRAM Implementation
- Contemporary Cache Power Reduction Schemes
- Proposed Work
- Q&A
Background (Cache Implementation)

2-way set-associative Cache Read

Note: (external signal)
Background (Cache Implementation)

- Full CMOS 6T Memory Cell
Background (Cache Implementation)

Simplified 8 x 8b SRAM array
Background (Cache Implementation)

Simplified 8 x 8b SRAM array
SRAM partitioning: array is often divided into smaller "subarrays"
Presentation Outline

• Motivation (finished)
• Background (finished)
  – Power Dissipation
  – Cache/SRAM Implementation
• Contemporary Cache Power Reduction Schemes
• Proposed Work
• Q&A
## Cache Power Reduction Techniques

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Dynamic / Static?</th>
<th>Est. Power Savings</th>
<th>Exec-time increase?</th>
<th>State-Rotentive?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gated-Vdd</td>
<td>Static</td>
<td>N/A *</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>Cache decay</td>
<td>Static</td>
<td>80%</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>DRG-cache</td>
<td>Static</td>
<td>39%-59%</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>Drowsy cache</td>
<td>Static</td>
<td>60-75%</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Near-OPT precharge</td>
<td>Static</td>
<td>N/A **</td>
<td>YES</td>
<td>N/A</td>
</tr>
<tr>
<td>Way-halting</td>
<td>Dynamic</td>
<td>55%</td>
<td>NO</td>
<td>N/A</td>
</tr>
<tr>
<td>Data size detection</td>
<td>Dynamic</td>
<td>N/A</td>
<td>NO</td>
<td>N/A</td>
</tr>
</tbody>
</table>

* - paper only cites 62% energy-delay savings
** - paper only cites 92% reduction of bitline discharge
# Cache Power Reduction Techniques (cont...)

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Miss Ratio increase?</th>
<th>Access time increase?</th>
<th>Variable load-hit latency?</th>
<th>μARCH transparent?</th>
<th>Additional noise problems?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gated-Vdd</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>Cache decay</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>DRG-cache</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Drowsy cache</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>Near-OPT precharge</td>
<td>NO</td>
<td>NO*</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>Way-halting</td>
<td>NO</td>
<td>NO*</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>Data size detection</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
</tbody>
</table>

* - With proper design
Cache Power Reduction Techniques

First four techniques: Supply Gating

Stacking Effect:
Cache Power Reduction Techniques

1. Gated-Vdd (circuit)

-6TMC can be disabled by the gating transistor, resulting in less leakage
Cache Power Reduction Techniques


- Mask out part of the index to dynamically resize the cache
- Make this decision based on the cache Hit ratio
- Energy-delay reduced by 62%
Cache Power Reduction Techniques


Example:
If MASK removes the upper 2 bits of the index, only the lower _ sets of the cache can be accessed (all other sets are gated off)
Cache Power Reduction Techniques

2. Cache decay (concept)

- If we turn a cache block’s power off right after it is last accessed, we save leakage power *without* any performance penalty
Cache Power Reduction Techniques

2. Cache decay (circuit borrows Gated-Vdd techniques)

![Cache Decay Circuit Diagram]

- SLEEP
- WL
- BL
- BLB
- High-Vt PMOS
2. Cache decay (microarchitecture)

- Static power reduced by 80%!!
3. Data Retention Ground (DRG) (circuit)

- DRG gates the ground of the MC’s
- With careful sizing, state can be preserved!
- Technique is transparent!!
- Power is reduced by 39% to 59%
Cache Power Reduction Techniques

4. Drowsy caches (circuit)

- Drowsy caches (microarchitecture): Simple algorithm – periodically put *every* cache line into drowsy mode
- Static power reduced by 60% to 75%
5. Near-optimal Precharging

- Bitline leakage burns power even in unused cache subarray (additional power is needed during the precharge phase)
- For a given time interval, only a small fraction of subarrays are actually used
- Bitline discharge reduced by 92%
Cache Power Reduction Techniques

5. Near-optimal Precharging

-Near-optimal precharging: stop precharging infrequently-used subarrays
-Microarchitecture: counters to track subarray use, and a system to handle variable load-hit latency
Cache Power Reduction Techniques

6. Way-halting cache

- Perform early miss detection to stop access to cache ways that are certain to miss.
- Early miss detection performed by offloading a few tag bits into a faster array that performs tag comparison early in the access.
- Power reduced by 55%
Cache Power Reduction Techniques

7. Data Size Detection

- Not every operand uses up the maximum space provided by the wordlength (e.g. ~94% of the operands in 64-bit Alpha SpecInt95 benchmarks use 32-bit or less)

- Keep track of this information to turn off the upper bits of the datapath (saving on wordline, bitline and sense-amp power)

Plot from Brooks and Martonosi
# Cache Power Reduction Techniques

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Dynamic / Static?</th>
<th>Est. Power Savings</th>
<th>Exec-time increase?</th>
<th>State-Retentive?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gated-Vdd</td>
<td>Static</td>
<td>N/A *</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>Cache decay</td>
<td>Static</td>
<td>80%</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>DRG-cache</td>
<td>Static</td>
<td>39%-59%</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>Drowsy cache</td>
<td>Static</td>
<td>60-75%</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Near-OPT precharge</td>
<td>Static</td>
<td>N/A **</td>
<td>YES</td>
<td>N/A</td>
</tr>
<tr>
<td>Way-halting</td>
<td>Dynamic</td>
<td>55%</td>
<td>NO</td>
<td>N/A</td>
</tr>
<tr>
<td>Data size detection</td>
<td>Dynamic</td>
<td>N/A</td>
<td>NO</td>
<td>N/A</td>
</tr>
</tbody>
</table>

* - paper only cites 62% energy-delay savings

** - paper only cites 92% reduction of bitline discharge
# Cache Power Reduction Techniques (cont...)

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Miss Ratio increase?</th>
<th>Access time increase?</th>
<th>Variable load-hit latency?</th>
<th>µARCH transparent?</th>
<th>Additional noise problems?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gated-Vdd</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>Cache decay</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>DRG-cache</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Drowsy cache</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>Near-OPT precharge</td>
<td>NO</td>
<td>NO*</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>Way-halting</td>
<td>NO</td>
<td>NO*</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>Data size detection</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
</tbody>
</table>

* - *With proper design*
Presentation Outline

• Motivation (finished)
• Background (finished)
  – Power Dissipation
  – Cache/SRAM Implementation
• Contemporary Cache Power Reduction Schemes
• Proposed Work
• Q&A
Proposed Work

• Detailed comparative study of discussed low-power cache techniques (and various combinations)

• Metrics of comparison:
  – Power dissipation (including overheads)
  – Performance penalty (IPC and access time)
  – Die area overhead
  – Complexity
Proposed Work

• Contributions
  – Every scheme is put on the same playing field
  – Schemes are made up to date with the use of predictive 65nm/45nm technology
  – Improved evaluation accuracy
    • Gate leakage is now accounted for
    • Careful accounting for overheads
    • Use of a state-of-the-art memory system model
  – Data Size Detection is proposed
Q & A
Thank You