Reducing Memory Energy

• Improve the locality of memory accesses
  Minimize the number of memory accesses
• Architectural and circuit techniques
• Optimizing interactions of compiler and cache architecture
• Technology changes
Cache: Bit Line Segmentation

• RAM cells in each column are organized into blocks selected by word lines
• Only the memory cells in the activated block present a load on the bit line
  – lowers power dissipation (by decreasing bit line capacitance)
  – can use smaller sense amps
Bit Line Segmentation

- Address decoder identifies the segment targeted by the row address and isolates all but the targeted segment from the common bit line
- Reduces bit line capacitance
- Has minimal effect on performance
Other Circuit Optimizations

- Pulsed Word Line (PWL) and Isolated Bit Line (IBL)
  - Limit bit line swings
- HSPICE simulation of these designs for different organizations of 0.5Kbit SRAM
- 52% reduction when using DBL, PWL and IBL techniques
- Can capture influence in analytical model
Cache Block Buffering

- Check to see if data desired is in the data output latch from the last cache access (i.e., in the same cache block)
- Saves energy since not accessing tag and data arrays
  - minimal overhead hardware
- Can maintain performance of normal set associative cache
- Analytical model can vary block buffer parameters
Block Buffer Cache Structure

disable row decoders and BL precharge

Address issued by CPU

last_set_#

Hit

Desired word
Dcache Energy

8K 4way Dcache

Avg. 58.7% reduction
Influence of Hardware Optimizations:
Entire Memory View

![Graph showing memory system energy reduction with 8K 4way Dcache.]

- Avg. 11% reduction
Reducing Memory Energy

- Improve the locality of memory accesses
  - Minimize the number of memory accesses
- Architectural and circuit techniques
- Optimizing interactions of compiler and cache architecture
- Technology changes
Relative Energy Savings

- $(\text{Energy\_Reduction}\text{[w/Optimized Code]} - \text{Energy\_Reduction}\text{[w/Original Code]}) / \text{Energy\_Reduction}\text{[w/Original Code]})$

- Energy\_Reduction due to hardware schemes
Compiler Optimizations and Block Buffering

![Bar Chart]

- Relative Energy Savings
- Categories: tomcatv, brix, mxm, vpenta, adi
- Energy Savings: 0.05, 0.3, 0.15, 0.1, 0.25

Legend: 4K
Optimizing for Block Buffers

for(I=0; I<n; I++)
  for(J=0; J<n; J++)
    .. V[J][I];

for(J=0; J<n; J++)
  for(I=0; I<n; I++)
    .. V[J][I];

Improves locality within block buffer
MRU Cache

- Access only the Most Recently Used way in a multi-way cache
- If it misses, access all the other ways
- If prediction is successful, can decrease the energy cost per access for other ways
- But has performance penalty and could increase system energy due to this penalty
Most Recently Used Cache
Compiler Optimizations and MRU Cache

Relative Energy Savings

-0.05  0   0.05  0.1  0.15  0.2  0.25  0.3  0.35  0.4

Tomcatv  btrix  mxm  vpenta  adi

-0.05  0   0.05  0.1  0.15  0.2  0.25  0.3  0.35  0.4

2K  4K

2K  4K
Interaction of Compiler and Hardware Optimizations

• Block buffering saved 19% more energy when using compiler optimizations
  – Block buffer hit rates improve
  – Loop permutations had maximum impact

• MRU caches saved 21% more energy when using compiler optimizations
  – Way prediction was more successful
Reducing Memory Energy

• Improve the locality of memory accesses
  Minimize the number of memory accesses
• Architectural and circuit techniques
• Optimizing interactions of compiler and cache architecture
• Technology changes and low-power operating modes
Sensitivity to Technology Changes

![Graph showing energy consumption for different technology changes. The x-axis represents energy consumption values ranging from 2.475e-11 to 4.95e-9, and the y-axis represents different technology change configurations. The graph includes bars indicating data path, I-cache, D-cache, and main memory components.]
Technology Factor (Em=4.95e-9)

Energy Breakdown (%)
Technology Factor (Em=2.475e-11)

![Energy Breakdown Diagram](image-url)
Partitioned Memory Architecture
Alternatives

• All memory modules are ON all the time
• Mode Control Only
  – If not used, reduce power
  – No data/access pattern modifications
• Mode Control + Optimizations
  – Data Transformations (e.g., Clustering)
  – Loop Optimizations (e.g., Loop Splitting)
Power Modes

- **active**: 3.570 nJ
- **standby**: 0.830 nJ
- **napping**: 0.320 nJ
- **powerdown**: 0.005 nJ
- **disabled**: 0.000 nJ
Experimental Setup

Compiler-Directed Approach:

Input Code

- SUIF Pass (Pre-processing)
- SUIF Pass (Clustering)
- SUIF Pass (Mode Detection)

Output Code

- Energy Simulator (Coarse Grain)

Energy Consumption Statistics

Technology Parameters, Memory Configuration, & Prediction Mechanism
Experimental Setup

Self-Monitored Approach:

Input Code → SUIF Pass (Pre-processing) → SUIF Pass (Clustering) → Output Code

Energy Simulator (Detailed Mode) → Energy Consumption Statistics

Technology Parameters, Memory Configuration, & Prediction Mechanism
Array Clustering Heuristics

- Profile-Based
- Static Analysis-Based
  - Constructive Algorithm (Graph-Based)
  - Iterative Algorithm

Module/Bank Configuration is Important!
Array Access Profile (*vpenta*)

<table>
<thead>
<tr>
<th>Phase Number</th>
<th>Array Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>U1</td>
</tr>
<tr>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>2</td>
<td>X</td>
</tr>
<tr>
<td>3</td>
<td>X</td>
</tr>
<tr>
<td>4</td>
<td>X</td>
</tr>
<tr>
<td>5</td>
<td>X</td>
</tr>
<tr>
<td>6</td>
<td>X</td>
</tr>
<tr>
<td>7</td>
<td>X</td>
</tr>
<tr>
<td>8</td>
<td>X</td>
</tr>
</tbody>
</table>
Iterative Algorithm (*vpenta*)

U1  U2  U3  U4  U5  U6  U7  U8

- U4  U5
- U1  U6  U7
- U2  U8
- U3

U4  U5

U1  U6  U7

U2  U8

U3

U4  U5  U1  U6  U7  U2  U8  U3
# Bank Access Profile (vpenta) (unoptimized)

<table>
<thead>
<tr>
<th>Phase Number</th>
<th>Memory Banks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>B0</td>
</tr>
<tr>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>2</td>
<td>X</td>
</tr>
<tr>
<td>3</td>
<td>X</td>
</tr>
<tr>
<td>4</td>
<td>X</td>
</tr>
<tr>
<td>5</td>
<td>X</td>
</tr>
<tr>
<td>6</td>
<td>X</td>
</tr>
<tr>
<td>7</td>
<td>X</td>
</tr>
<tr>
<td>8</td>
<td>X</td>
</tr>
</tbody>
</table>
Bank Access Profile (*vpenta*)
(optimized)

<table>
<thead>
<tr>
<th>Phase Number</th>
<th>Memory Banks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>B0</td>
</tr>
<tr>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>2</td>
<td>X</td>
</tr>
<tr>
<td>3</td>
<td>X</td>
</tr>
<tr>
<td>4</td>
<td>X</td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>X</td>
</tr>
<tr>
<td>7</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>
# Benchmark Codes

<table>
<thead>
<tr>
<th>Benchmark Number</th>
<th>Benchmark Name</th>
<th>Data Size (MB)</th>
<th>Base Energy (mJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>adi</td>
<td>48.0</td>
<td>3.38</td>
</tr>
<tr>
<td>2</td>
<td>dtdtz</td>
<td>61.8</td>
<td>2.55</td>
</tr>
<tr>
<td>3</td>
<td>bmcm</td>
<td>39.9</td>
<td>3.93</td>
</tr>
<tr>
<td>4</td>
<td>btrix</td>
<td>47.7</td>
<td>2.49</td>
</tr>
<tr>
<td>5</td>
<td>eflux</td>
<td>33.6</td>
<td>413.23</td>
</tr>
<tr>
<td>6</td>
<td>full_search</td>
<td>33.0</td>
<td>337.75</td>
</tr>
<tr>
<td>7</td>
<td>matvec</td>
<td>16.0</td>
<td>675.75</td>
</tr>
<tr>
<td>8</td>
<td>mxm</td>
<td>48.0</td>
<td>10.70</td>
</tr>
<tr>
<td>9</td>
<td>phods</td>
<td>33.0</td>
<td>1586.25</td>
</tr>
<tr>
<td>10</td>
<td>tomcatv</td>
<td>56.0</td>
<td>119.80</td>
</tr>
<tr>
<td>11</td>
<td>vpenta</td>
<td>44.0</td>
<td>506.68</td>
</tr>
<tr>
<td>12</td>
<td>amhmtm</td>
<td>48.1</td>
<td>7.40</td>
</tr>
</tbody>
</table>
Energy Savings
(Mode Control only)

Savings between 12% and 75%
Energy Savings
(Mode Control + Clustering)

As much as 50% saving over pure mode control