Way-Predicting Set-Associative Cache for High Performance and Low Energy Consumption

Koji Inoue, Tohru Ishihara, and Kazuaki Murakami

Department of Computer Science and Communication Engineering Kyushu University

ppram@c.csce.kyushu-u.ac.jp

Conventional 4-Way Set-Associative Cache

Total energy for an access for decode for I/O pin drive

\[ E_{\text{cache}} = E_{\text{decode}} + E_{\text{memory}} + E_{\text{i/o}} \]

Step 1. Decode circuit

Step 2. Activate word line Activate senseamps pre(dis)charge bit lines

Step 3. Hit Miss

Step 4. Provide the required data Cache replacement

Activate of I/O pins
**Phased 4-Way Set-Associative Cache for Low Energy Consumption**

Energy consumption improvement by sacrificing the performance

**Cycle 1**

- Step 1: Address Decode
- Step 2: Read out of only tags
- Step 3: Tag comparison
- Step 4: Cache replacement
- Step 5: Read out of only the desired line

**Cycle 2**

- Step 4: Provide the required data

**Way-Predicting Set-Associative Cache - Concept -**

*How can we achieve high-performance and low energy consumption at the same time?*

Fast access by reading out both of tag and line simultaneously

- Conventional: Good!  Phased: Bad!

Low energy by avoiding unnecessary line read access

- Conventional: Bad!  Phased: Good!

Predict which way has the data desired by the processor before the cache access is started
4Way-Predicting Set-Associative Cache
- Operation -

Way Prediction
(Cache-line Base MRU Algorithm)

Cycle 1

Step 0
Step 1
Step 2
Step 3
Step 4
Step 5
Step 6

Prediction Hit
Prediction Miss
Cache Miss

Cycle 2

4Way-Predicting Set-Associative Cache
- Organization -

Way Prediction

MRU Algorithm
Evaluation Environment

**Cache Models**
- Conventional 4-way Set-Associative Cache (4SACache)
- Phased 4-way Set-Associative Cache (P4SACache)
- Way-Predicting 4-way Set-Associative Cache (WP4SACache)

*Cache Size: 16 K Byte, Cache-line Size: 32 Byte, Replacement Algorithm: LRU*

**Evaluation Items**
- **Performance** ($T_{cache}$): average number of clock cycles for an access
- **Energy** ($E_{cache}$): average energy consumption for an access

$$E_{cache} \sim E_{memory} = N_{tag} \times E_{tag} + N_{data} \times E_{data}$$

- Ave. number of tag-subarray accessed for an access
- Ave. number of line-subarray accessed for an access

---

**Static Analysis**

- **Energy and Performance Expression**

<table>
<thead>
<tr>
<th></th>
<th>$4SACache$</th>
<th>$P4SACache$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_{4SACache}$</td>
<td>$4 \times E_{tag} + 4 \times E_{data}$</td>
<td>$4 \times E_{tag} + E_{data} \times CHR$</td>
</tr>
<tr>
<td>$T_{4SACache}$</td>
<td>$1$</td>
<td>$1 + 1 \times CHR$</td>
</tr>
<tr>
<td>$E_{WP4SACache}$</td>
<td>$(E_{tag} + E_{data}) + (3 \times E_{tag} + 3 \times E_{data}) \times (1 - PHR)$</td>
<td>$1 + 1 \times (1 - PHR)$</td>
</tr>
</tbody>
</table>

**CHR:** Cache Hit Rate
**PHR:** Prediction Hit Rate
Static Analysis
- Best and Worst Case -

- Best Case (PHR = 100%)
  - 75% energy improvement without any performance degradation

- Worst Case (PHR = 0%)
  - 100% performance overhead without any energy improvement

Energy Consumption
(Eneg = 0.07Edata)

Performance

Compare with Conventional (4SACache)

Experimental Analysis
- Prediction Hit Rate -

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>PHR (%)</th>
<th>PMR (%)</th>
<th>CMR (%)</th>
<th>PHR (%)</th>
<th>PMR (%)</th>
<th>CMR (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P4SACache</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WP4SACache</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>95.87</td>
<td>3.49</td>
<td>0.62</td>
<td>86.34</td>
<td>9.41</td>
<td>4.25</td>
</tr>
</tbody>
</table>

Benchmarks
I-Cache D-Cache
Experimental Analysis
- Result of Instruction Cache -

Experimental Analysis
- Result of Data Cache -
Experimental Analysis
- Energy and Performance -

Average of all benchmarks

<table>
<thead>
<tr>
<th></th>
<th>Ecache</th>
<th>Tcache</th>
</tr>
</thead>
<tbody>
<tr>
<td>I-Cache</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conventional</td>
<td>30.3%</td>
<td>28.1%</td>
</tr>
<tr>
<td>Phased</td>
<td>104.1%</td>
<td></td>
</tr>
<tr>
<td>Way-Predicting</td>
<td>199.4%</td>
<td></td>
</tr>
<tr>
<td>Phased</td>
<td></td>
<td>195.8%</td>
</tr>
<tr>
<td>Average</td>
<td>143.1%</td>
<td>125.9%</td>
</tr>
</tbody>
</table>

Cache Power Consumption

Effect of on-chip caches to total chip power consumption

- DEC 21164 CPU* 25%
- StrongARM SA-110 CPU* 43%
- Bipolar ECL CPU** 50%

Energy Consumption Model

Components of the power dissipation

Ememory = 95.6%
Ememory = 97.7%


Average Energy Consumption for an access
Energy consumed for accessing a tag-subarray
Energy consumed for accessing a line-subarray

Ecache ~ Ememory = Ntag \times Etag + Ndata \times Edata

Ave. number of tag-subarray accessed for an access
Ave. number of line-subarray accessed for an access

Experimental Analysis
- Environment -

Benchmarks

SPECint95
099.go, 124.m88ksim, 126.gcc, 129.compress, 130.li, 132.jpeg, 134.perl, 147.vortex

SPECfp95
101.tomcatv, 102.swim, 103.su2cor, 104.hydro2d