Example

- Fully associative, write-back cache (empty at start)

<table>
<thead>
<tr>
<th>Write allocate</th>
<th>No Write allocate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memwrite[100]</td>
<td>miss (load 100)</td>
</tr>
<tr>
<td>Memwrite[100]</td>
<td>miss</td>
</tr>
<tr>
<td>Memread[200]</td>
<td>miss (load 200)</td>
</tr>
<tr>
<td>Memwrite[200]</td>
<td>hit</td>
</tr>
<tr>
<td>Memwrite[100]</td>
<td>hit</td>
</tr>
<tr>
<td>Memwrite[100]</td>
<td>hit</td>
</tr>
</tbody>
</table>

Alpha 21264 Data Cache

- On a miss...
  - Cache signal processor
  - 64 bytes from next lower level
  - 640 SW ~140MB/s (single transfer 16 bytes in 2.25ns)
  - 9ns for 64 bytes to arrive (6667MHz)
  - FIFO selection for block replacement (1-bit)
  - Write back: use write buffer (8 entries)
  - Write allocation
- Other details
  - 64K instruction cache

CPU Time

CPU execution time = (CPU clock cycles + memory stall cycles) x clock cycle

Simple in-order processors:

Modern superscalar processors:

Cache Measures

- Hit rate: fraction found in that level
- So high that usually talk about Miss rate
- Miss rate fallacy: as MIPS to CPU performance, miss rate to average memory access time in memory
- Average memory access time = Hit time + Miss rate x Miss penalty
- Miss penalty: time to replace a block from lower level, including time to replace in CPU
- Miss penalty = full miss latency
- Access time = time between read request and when desired word arrives
- Transfer time = time to transfer block
- Cycle time = min time between requests to memory

Cache Performance

cpu execution time = (CPU clock cycles + memory stall cycles) x clock cycle

mem stall cycles = (#misses x miss penalty)

mem stall cycles = (memory accesses x miss rate x miss penalty)
Example 1

<table>
<thead>
<tr>
<th>Size</th>
<th>16KB</th>
<th>32KB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Icache</td>
<td>0.00382</td>
<td>0.00136</td>
</tr>
<tr>
<td>Dcache</td>
<td>0.0384</td>
<td>0.0433</td>
</tr>
</tbody>
</table>

Miss rate = \( \frac{\text{misses}}{\text{instructions}} \)

\[ \text{Hit time} = 2 \] \[ \text{Miss time} = 100 \]

Unified cache

- Single ported

Miss rate = \( \frac{\text{misses}}{\text{instructions}} \)

\[ \text{Hit rate} = 0.9 \]

\[ \text{Miss rate} = 0.1 \]

Example 2

- Calculate CPI with perfect and real cache
  - With cache
    - miss rate = 5%
    - dmiss rate = 10%
    - Miss penalty = 40 cycles
  - Performance with/without cache?

\[ \text{Hit time} = 2 \] \[ \text{Miss time} = 100 \]

AMAT = hit time \times \text{miss rate} \times \text{miss penalty}

\[ \text{AMAT} = \% \times \% \times \% \times \text{AMAT} \]

\[ \text{AMAT}_{\text{perfect}} = 0.1 \times 0.9 \times 0.9 \times 0.9 = 0.06561 \]

\[ \text{AMAT}_{\text{real}} = 0.1 \times 0.9 \times 0.9 \times 0.9 = 0.06561 \]

Example 2

<table>
<thead>
<tr>
<th>Type</th>
<th>43%</th>
<th>21%</th>
<th>24%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accesses</td>
<td>1.57</td>
<td>1.57</td>
<td>1.57</td>
</tr>
<tr>
<td>Perfect cache</td>
<td>4.9</td>
<td>8</td>
<td>4</td>
</tr>
</tbody>
</table>

Summary

- CPU-Memory gap is major obstacle for performance, HW and SW
- Take advantage of program behavior: locality
- Time of program still only reliable performance measure
- 4Qs of memory hierarchy
- Lots of formulas (summary pg 412 H&P)