Cache Behavior

Constructing an effective cache requires the balancing of many properties. Bigger is always better, but how it is arranged is also important.

Recall Direct Mapped Cache

Direct Mapped Cache (4 word blks)

Benefits of Multiword Blocks

Increasing block size improves performance, to a point.
Larger blocks increase benefits of spatial locality.
Larger blocks = fewer blocks for a given cache size = greater likelihood a useful block is flushed when another block is brought in (conflict misses).

Memory request techniques --
Early restart
Requested word first

Miss rate vs block size

VAX traces (Agarwal, 1987)
## Memory Organizations

### CPU Cache Mux Cache Memory

<table>
<thead>
<tr>
<th>Clk</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Send Address</td>
</tr>
<tr>
<td>1</td>
<td>Access DRAM</td>
</tr>
<tr>
<td>2</td>
<td>Send Data Wd</td>
</tr>
</tbody>
</table>

**Miss Penalty:**
- 1+4x10+4x45 = 45
- 1+1x10+1 = 12
- 1+1x10+4x1 = 15

**Bytes per Cycle:**
- 4x4/45 = 0.35
- 4x4/12 = 1.33
- 4x4/15 = 1.0

## Alternative Designs

### Possible arrangements for cache elements

- **Direct Mapped**
- **2 Way Set-Associative**
- **4 Way Set-Associative**
- **Fully Associative; 8 Way Set-Associative**

## Set Associativity

- **Tag**
- **Index**
- **Address**
- **Data**

### Writing vs Reading Cache

- **Writing has two basic forms**
  - Write through
  - Write back
- **Since writing not on critical path, write buffers**
- **When an element is written, need it be kept in cache?**
  - Load on write … especially of block > word

## Associativity

- 8-way is pretty much the upper limit except for TLB and memory
- **Replacement policy**
  - Optimal
  - Least recently used
  - Random
  - Pgm control