Set-Associativity

- **Set-associativity**
  - Block can reside in one of few frames
  - Frame groups called **sets**
  - Each frame in set called a **way**
  - This is 2-way set-associative (SA)
  - 1-way → direct-mapped (DM)
  - 1-set → fully-associative (FA)

+ Reduces conflicts
  - Increases \( t_{aa} \), additional mux

---

Cache Performance Simulation

- Parameters: 32B cache, 4B blocks, **2-way set-associative**
  - Initial contents: 0000, 0010, 0020, 0030, 0100, 0110, 0120, 0130

<table>
<thead>
<tr>
<th>Cache contents</th>
<th>Address</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>[0000,0100], [0010,0110], [0020,0120], [0030,0130]</td>
<td>3020</td>
<td>Miss</td>
</tr>
<tr>
<td>[0000,0100], [0010,0110], [0120,3020], [0130,3030]</td>
<td>3100</td>
<td>Miss</td>
</tr>
<tr>
<td>[0100,2100], [0010,0110], [0120,3020], [0130,3030]</td>
<td>0012</td>
<td>Hit</td>
</tr>
<tr>
<td>[0100,2100], [0010,0110], [0120,3020], [0130,3030]</td>
<td>0020</td>
<td>Miss</td>
</tr>
<tr>
<td>[0100,2100], [0010,0110], [3020,0020], [0120,3030]</td>
<td>0030</td>
<td>Miss</td>
</tr>
<tr>
<td>[0100,2100], [0010,0110], [3020,0020], [0030,0030]</td>
<td>0110</td>
<td>Hit</td>
</tr>
<tr>
<td>[0100,2100], [0010,0110], [3020,0020], [0030,0030]</td>
<td>0100</td>
<td>Hit (avoid conflict)</td>
</tr>
<tr>
<td>[0100,2100], [0010,0110], [3020,0020], [0030,0030]</td>
<td>3000</td>
<td>Miss</td>
</tr>
<tr>
<td>[0100,2100], [0010,0110], [3020,0020], [0030,0030]</td>
<td>3100</td>
<td>Miss</td>
</tr>
<tr>
<td>[0100,2100], [0010,0110], [3020,0020], [0030,0030]</td>
<td>3020</td>
<td>Hit (avoid conflict)</td>
</tr>
<tr>
<td>[0100,2100], [0010,0110], [3020,0020], [0030,0030]</td>
<td>3020</td>
<td>Hit (avoid conflict)</td>
</tr>
</tbody>
</table>
**NMRU and Miss Handling**

- Add MRU field to each set
  - MRU data is encoded "way"
  - Hit? update MRU
  - Fill? write enable ~MRU

**Physical Cache Layout**

- Logical layout
  - Data and tags mixed together
- Physical layout
  - Data and tags in separate RAMs
  - Often multiple sets per line
  - As square as possible
  - Not shown here

**Full-Associativity**

- How to implement full (or at least high) associativity?
  - Doing it this way is terribly inefficient
  - 1K matches are unavoidable, but 1K data reads + 1K-to-1 mux?