Direct-Mapped Cache Lookup

Ex: Cache Size = 16 KB
Block Size = 32 bytes
# blocks = $\frac{16K}{32} = 512$

32-bit address

18

tag

9

block no.

5

block offset

1

tag

18

32 bytes

512 blocks

hit? data

-
(2-way) Set Associative Cache Lookup

Ex: Cache Size = 16 KB
    Block Size = 32 bytes
    # Sets = \frac{16 K}{32 \cdot 2} = 256
Fully Associative Cache Lookup

Ex: Cache Size = 16 KB
Block size = 32 bytes
# blocks = \( \frac{16 \text{K}}{32} = 512 \)
**Victim Cache**

**Motivation:**
Conflict misses usually isolated to a few sets

<table>
<thead>
<tr>
<th>Set</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- On access, check both caches (in parallel)
- Eviction from primary cache → victim cache
- Victim cache hit → primary cache
- If both caches miss, fetch from main memory → primary cache.
Multi-Level Caches

- Goal: Fast + large cache
- As you go down hierarchy:
  - Larger cache
  - Larger block size
  - Higher associativity
- Inclusion

![Diagram of multi-level cache hierarchy]

4 KB
1-cycle hit
write through
32-byte blocks

32 KB
4-cycle hit
write back
64-byte blocks

2 MB
10-cycle hit
write back
64-byte blocks

Avg Mem. Acc. Time = \( \text{hit}_{L1} + \text{miss rate}_{L1} (\text{hit}_{L2} + \text{miss rate}_{L2} (\text{hit}_{L3} + \text{miss rate}_{L3} \cdot \text{miss penalty})) \)