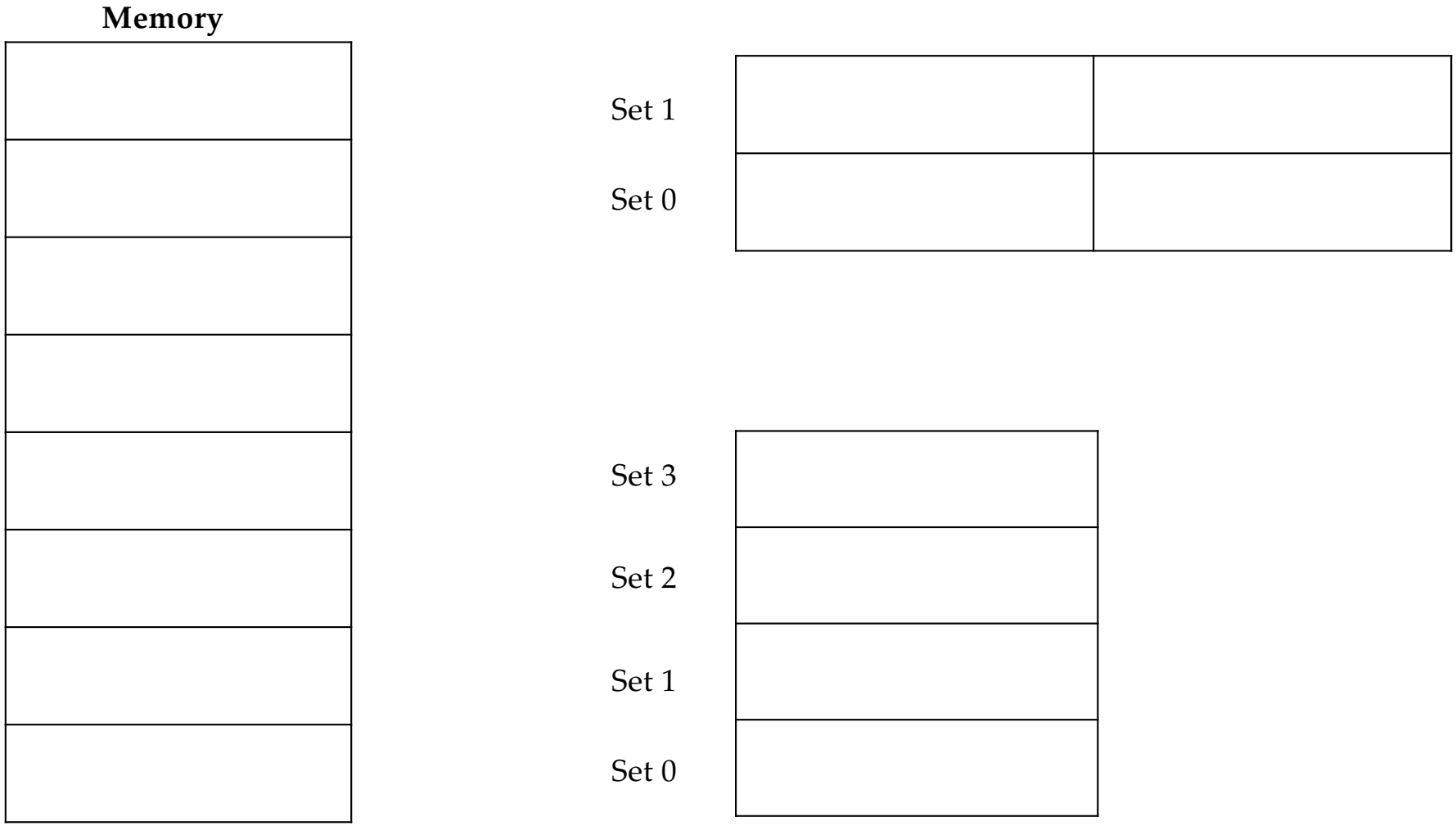
Set-Associative Caches

Cache Specifications

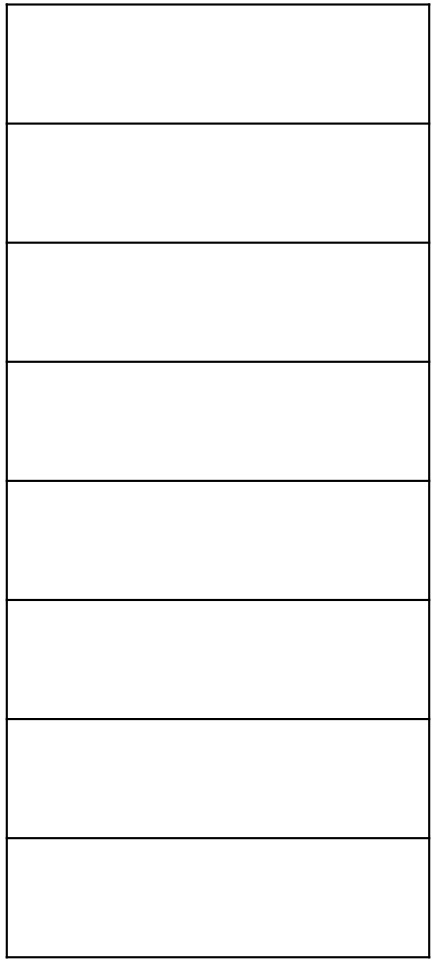
1. Capacity
   How much data can we store?
2. Number of Sets
   How many mappings between cache and main memory addresses?
3. Block Size
   Grab related words around a piece of data added to the cache.
4. Number of Blocks
   How many blocks can we store?
5. Degree of Associativity
   Different types of cache handle organization differently.

Cache Categorization

- Direct Mapped: each set contains only one block.
- *N*-way Set Associative Cache: each set contains *N* blocks
- Fully Associative Cache: only one set (*B*-way associative cache)


Memory
Set 1
Set 0
Set 3
Set 2
Set 1
Set 0

Memory



Differences from Direct-Mapped Cache

- Compared to a direct-mapped cache *of the same capacity,* a 2-way associative cache has:
  - Half as many sets
  - 1 bit less for the set
  - 1 bit more for the tag

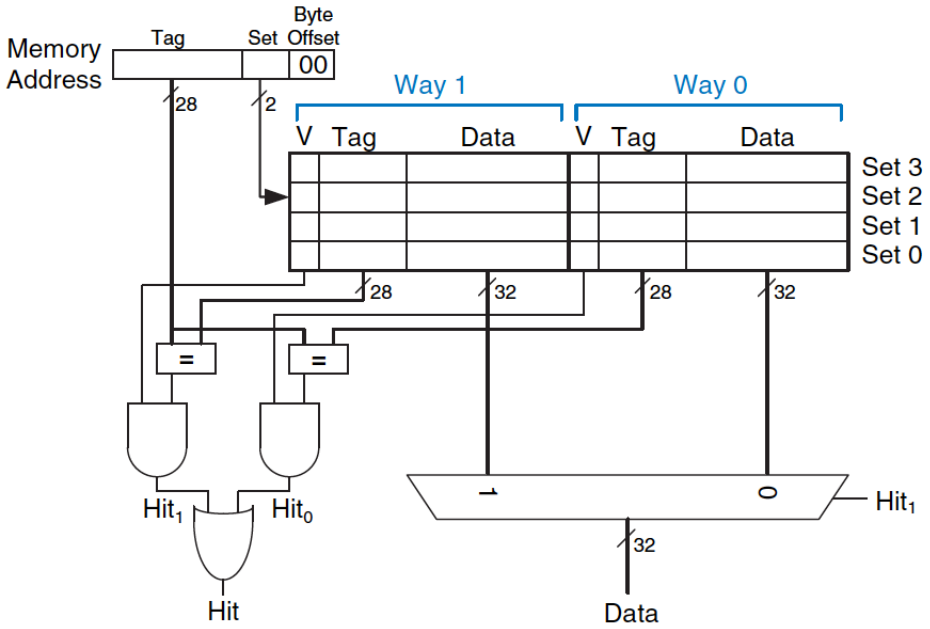
- How would a 4-way associative cache compare?

Multi-Way Set Associative Cache

N-way set associative cache reduces conflicts by providing *N* blocks in each set where data mapping to that set might be found.

Each memory address still maps to a specific set, but it can map to any one of the *N* blocks in the set.

N is the degree of associativity of the cache.

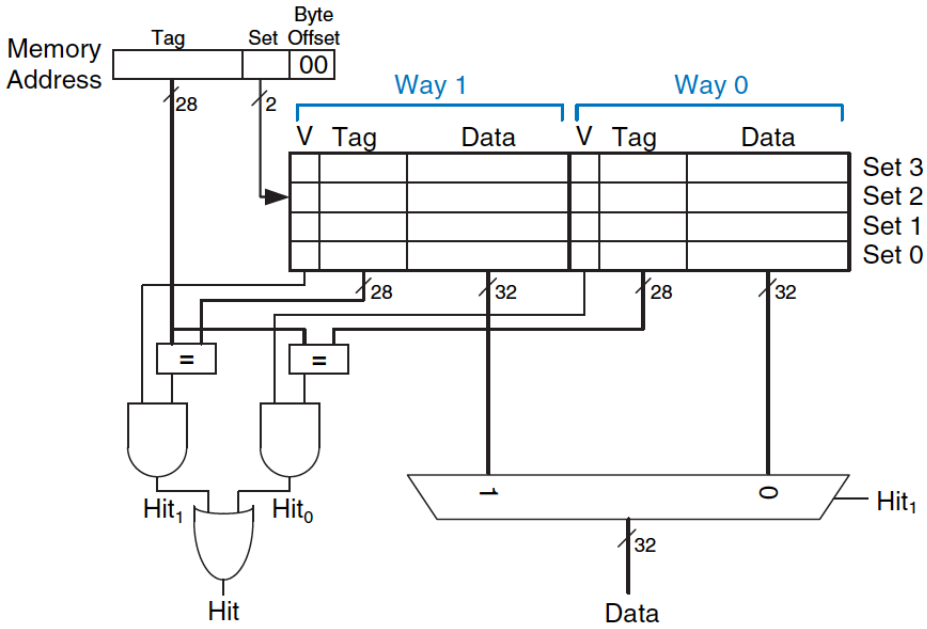


2-Way Set Associative Cache

Each set has two ways of associativity.

Cache reads blocks from both ways and checks tags and valid bits for a hit.

- If hit, multiplexer selects data from that way.



More tradeoffs

Set associative caches generally have lower miss rates than direct mapped of the same capacity

- We'll see an example of why this is the case next

But, usually slower and somewhat more expensive to build due to the MUX and comparators.