A Common Framework for Memory Hierarchies

Caching, paged virtual memory and TLBs all use the same underlying concepts.

Feature	Cache	Paged Mem	TLB
Size,Blocks	1K-100K	2K-250K	32-4000
Size, Bytes	8KB-8MB	8MB-8GB	128B-8000B
Blk Size, B	4-256	4KB-64KB	4-32
Miss Penalty	10-100clk	1M-10Mclk	10-100clk
Miss Rate	0.1%-10%	10^-4-10^-5%	0.01%-2%

(c) Copyright Larry Snyder 1998

Four Questions for Classification

- Where can a block be placed? *Block placement*
  - direct mapped, set associative, fully associative
- How is a block found? *Block identifcation*
  - indexing, set search, separate lookup table
- What block is replaced on a miss? *Block replacement*
  - LRU, Random, FIFO, MRU
- How are writes handled? *Write strategy*
  - write through or write back

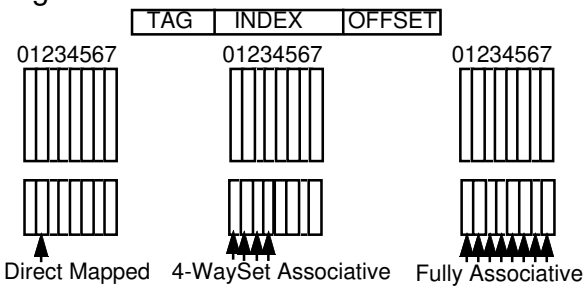
(c) Copyright Larry Snyder 1998

Block Placement

The extremes of cache mapping -- direct mapped and fully associative are end points on a spectrum.

Blocks are assigned to a cache by directly indexing any of its *n* sets and matching any of the *m* entries of the set by the tag.

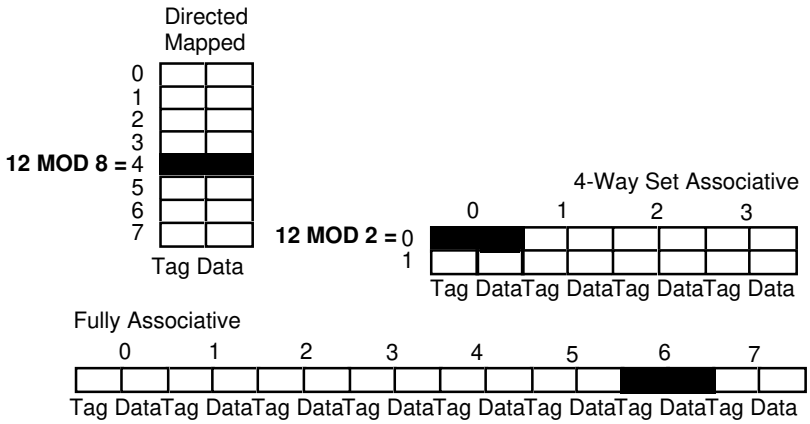
Indexing is "block number modulo number of sets"



(c) Copyright Larry Snyder 1998

Block Identification

Placement of a block whose address is 12 varies for direct, set associative, and fully associative



(c) Copyright Larry Snyder 1998

Block Replacement

- Replacement candidates are --
  - Any block in a fully associative cache
  - Any block of a set in set associative caches
  - The indexed block for direct mapped
- Replacement strategies --
  - Opt is best, but impossible
  - Least Recently Used (LRU) approximates Opt. Expensive
  - Random is easy, but impossible for software management
- For 2-way s.a., random has 1.1 times higher miss rate than LRU
- "Use" bit can approximate LRU

(c) Copyright Larry Snyder 1998

Write Strategy

- Write through simultaneously updates the cache and the lower level in the memory hierarchy on each write.
- Write back only updates the cache copy until the block is replaced, at which point the next lower level of the hierarchy is updated.
- Write through advantages --
  - Read misses are cheaper due to not waiting for write. Easier to implement, though it needs a write buffer.
- Write back advantages --
  - Multiple writes to a block require only one memory write.
  - Can utilize wider channel to lower level memory.
- Write back is always needed between memory & disk.
  - Dirty bit in page table determines if write back needed.

(c) Copyright Larry Snyder 1998

Mapping Choices in Hierarchy

- Tradeoff cost of miss *vs* cost of associativity.
- VM uses fully associative mapping
  - Reduces miss rate, because miss penalty is high
  - Mapping done in software
  - Large page size means page table size overhead is small
  - Note that page table is indexed, but full map provides fully associative placement
- Small caches (TLB) often use set associative placement
- Large caches never use fully associative placement
  - High cost and hit time penalties
  - Small performance advantage to set associative

(c) Copyright Larry Snyder 1998

The Three Cs

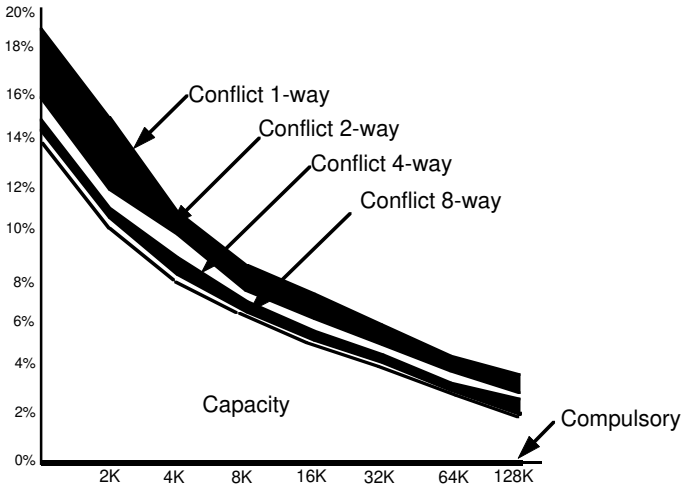
Missing in the cache can be caused by three different circumstances:

- Compulsory misses -- miss on first access
- Capacity misses -- miss due to cache not having enough blocks
- Conflict misses -- miss due to cache organization

In cache design, larger is always better ... but there are always trade-offs

(c) Copyright Larry Snyder 1998

Miss Rates


20%
18%
16%
14%
12%
10%
8%
6%
4%
2%
0%
Conflict 1-way
Conflict 2-way
Conflict 4-way
Conflict 8-way
Capacity
Compulsory
2K
4K
8K
16K
32K
64K
128K