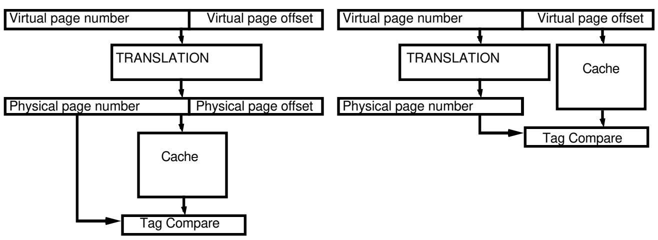
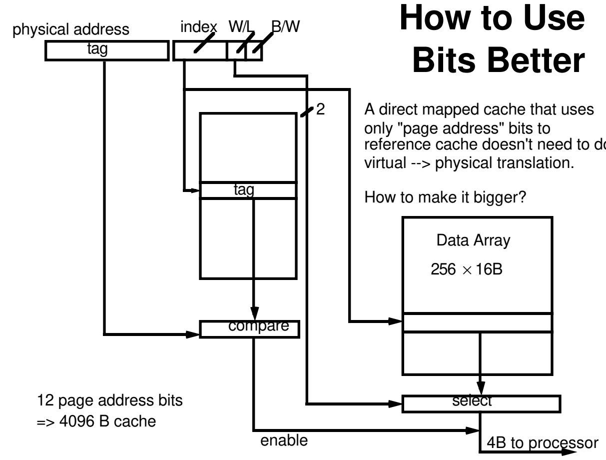
Engineering Considerations of Caching

Solving problems in generality is best, but to pick up a bit of extra performance, it is often advisable to take advantage of special circumstances.



How to Use Bits Better

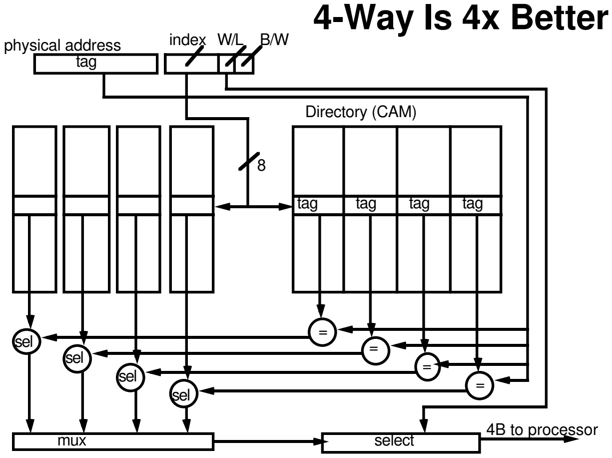


A direct mapped cache that uses only "page address" bits to reference cache doesn't need to do virtual --> physical translation.

How to make it bigger?

12 page address bits
=> 4096 B cache

4-Way Is 4x Better



Virtually Addressed Caches

Addressing the cache with the virtual address has the advantage that the address does not have to be translated except on a miss.

Problems -- aliases

- Multiple V addresses refer to same physical address.
- Same V address used by different processes to refer to different physical address.

Fixes --

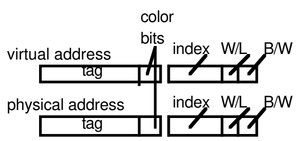
- Prohibit two processes in cache once.
- Keep process ID with each cache line.

Colored Pages

Suppose the OS chose pages so that the LSB of the tag portion of the virtual address matched the physical address . . . then more "untranslated" bits could be used to address the cache.

Coloring pages complicates the operating systems page replacement computation.
Colored pages restrict the usage of memory pages, and so forces suboptimal utilization.

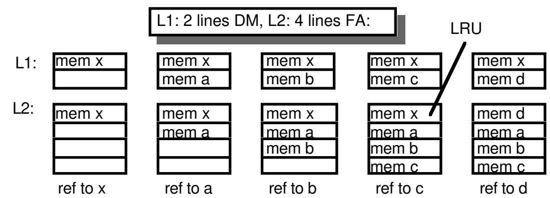
This is equivalent to about 1% reduction in memory size.



Two Level Caches

Two level caching is simply an application of caching to caching.
L2 cache should be "compatible" with L1 cache.

L2 should be substantially larger than L1.



Inclusion

Properties of L1 and L2 inclusion:

L2 line ≥ L1 line, else 1 L1 line depends on 2+ L2 lines.

L2 sets ≥ L1 sets, else multiple L1 sets depend on 1 L2 set.

*L2 size/(L2 assoc * L2 line size) ≥ L1 size/(L1 assoc * L1 line size)*

*L2 size/L2 assoc ≥ L1 size/L1 assoc * (L2 line size/L1 line size)*

L2 associativity ≥ L1 associativity.