CS152
Computer Architecture and Engineering
Lecture 22

Advanced Caching

April 23, 2003

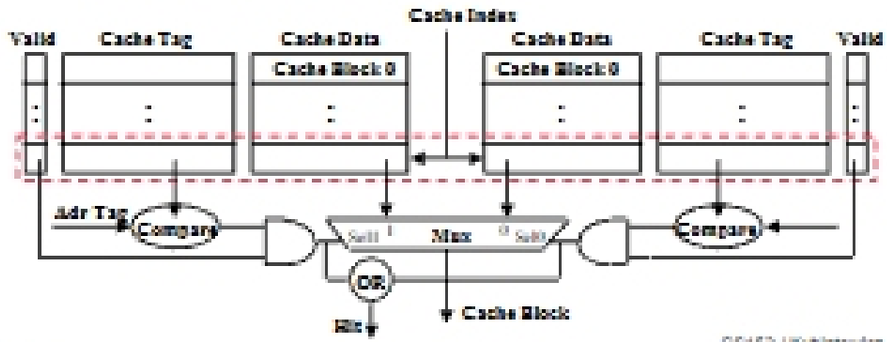
John Kubiatowicz (www.cs.berkeley.edu/~kubitron)

lecture slides: http://inst.eecs.berkeley.edu/~cs152/

Recap: Set Associative Cache

° N-way set associative: N entries for each Cache Index
- N direct mapped caches operates in parallel

° Example: Two-way set associative cache
- Cache Index selects a "set" from the cache
- The two tags in the set are compared to the input in parallel
- Data is selected based on the tag result



Recap: Cache Performance

Execution_Time =
Instruction_Count x Cycle_Time x
(Ideal CPI + Memory_Stalls/Inst + Other_Stalls/Inst)

Memory_Stalls/Inst =
Instruction Miss Rate x Instruction Miss Penalty +
Loads/Inst x Load Miss Rate x Load Miss Penalty +
Stores/Inst x Store Miss Rate x Store Miss Penalty

Average Memory Access time (AMAT) =
$Hit\ Time_{L1}$ + ($Miss\ Rate_{L1}$ x $Miss\ Penalty_{L1}$) =
($Hit\ Rate_{L1}$ x $Hit\ Time_{L1}$) + ($Miss\ Rate_{L1}$ x $Miss\ Time_{L1}$)

Recap: A Summary on Sources of Cache Misses

° Compulsory (cold start or process migration, first reference): first access to a block
- "Cold" fact of life: not a whole lot you can do about it
- Note: If you are going to run "billions" of instruction, Compulsory Misses are insignificant

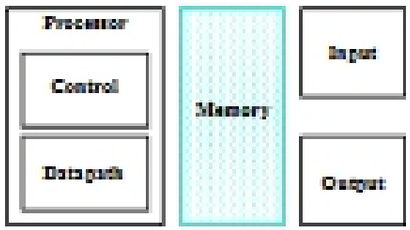
° Conflict (collision):
- Multiple memory locations mapped to the same cache location
- Solution 1: increase cache size
- Solution 2: increase associativity

° Capacity:
- Cache cannot contain all blocks access by the program
- Solution: increase cache size

° Coherence (Invalidation): other process (e.g., I/O) updates memory

The Big Picture: Where are We Now?

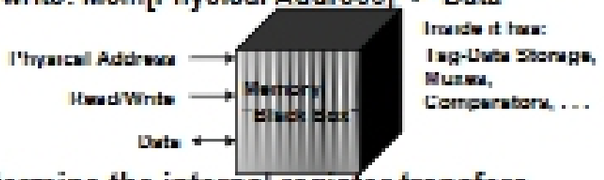
° The Five Classic Components of a Computer



° Today's Topics:
- Recap last lecture
- Virtual Memory
- Protection
- TLB
- Buses

How Do you Design a Memory System?

° Set of Operations that must be supported
- read: data <= Mem[Physical Address]
- write: Mem[Physical Address] <= Data



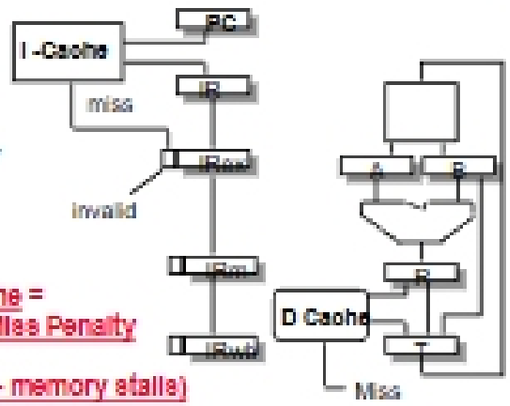
° Determine the internal register transfers

° Design the Datapath

° Design the Cache Controller

Impact on Cycle Time

Cache Hit Time:
directly tied to clock rate
Increases with cache size
Increases with associativity



Average Memory Access time =
Hit Time + Miss Rate x Miss Penalty

Time = IC x CT x (Ideal CPI + memory stalls)

Improving Cache Performance: 3 general options

Average Memory Access time =
Hit Time + (Miss Rate x Miss Penalty) =

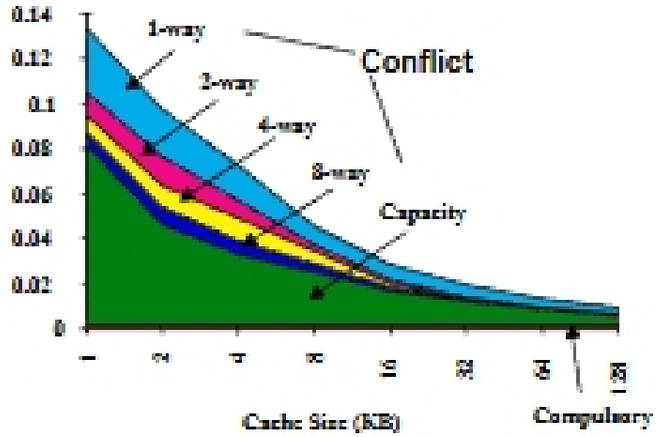
(Hit Rate x Hit Time) + (Miss Rate x Miss Time)

Options to reduce AMAT:
1. Reduce the miss rate,
2. Reduce the miss penalty, or
3. Reduce the time to hit in the cache.

Improving Cache Performance

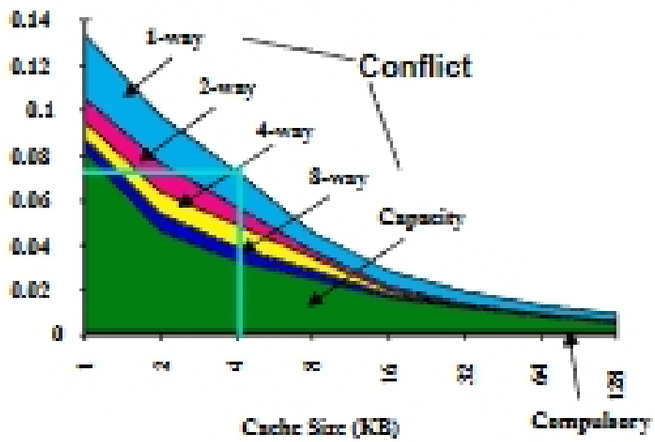
1. Reduce the miss rate,
2. Reduce the miss penalty, or
3. Reduce the time to hit in the cache.

3Cs Absolute Miss Rate (SPEC92)



2:1 Cache Rule

miss rate 1-way associative cache size X
= miss rate 2-way associative cache size X/2



3Cs Relative Miss Rate