Advanced Computer Architecture
ELEC3219 (2017/18)

Two-way (Fully) Associative Cache

Dr. Terrence Mak
tmak@ecs.soton.ac.uk
The Memory Hierarchy: Terminology

- **Block** (or line): the minimum unit of information that is present (or not) in a cache
- **Hit Rate**: the fraction of memory accesses found in a level of the memory hierarchy
  - **Hit Time**: Time to access that level which consists of Time to access the block + Time to determine hit/miss
- **Miss Rate**: the fraction of memory accesses *not* found in a level of the memory hierarchy  \( \Rightarrow 1 - \text{(Hit Rate)} \)
  - **Miss Penalty**: Time to replace a block in that level with the corresponding block from a lower level which consists of
    Time to access the block in the lower level + Time to transmit that block to the level that experienced the miss + Time to insert the block in that level + Time to pass the block to the requestor

Hit Time \( << \) Miss Penalty
Reducing Cache Miss Rates #1

1. Allow more flexible block placement

- In a **direct mapped cache** a memory block maps to exactly one cache block
- At the other extreme, could allow a memory block to be mapped to *any* cache block – fully associative cache

- A compromise is to divide the cache into *sets* each of which consists of n “ways” (*n-way set associative*). A memory block maps to a unique set (specified by the index field) and can be placed in any way of that set (so there are n choices)

  \[(\text{block address}) \mod (\# \text{ sets in the cache})\]
Set Associative Cache Example

Q1: Is it there?

Compare all the cache tags in the set to the high order 3 memory address bits to tell if the memory block is in the cache.

Q2: How do we find it?

Use next 1 low order memory address bit to determine which cache set (i.e., modulo the number of sets in the cache)

Main Memory

One word blocks
Two low order bits define the byte in the word (32b words)
Question (iii)
Four-Way Set Associative Cache

- $2^8 = 256$ sets each with four ways (each with one block)
Question (iv-v)
Range of Set Associative Caches

- For a fixed size cache, each increase by a factor of two in associativity doubles the number of blocks per set (i.e., the number or ways) and halves the number of sets – decreases the size of the index by 1 bit and increases the size of the tag by 1 bit.
Benefits of Set Associative Caches

- The choice of direct mapped or set associative depends on the cost of a miss versus the cost of implementation.

- Largest gains are in going from direct mapped to 2-way (20%+ reduction in miss rate).

Data from Hennessy & Patterson, Computer Architecture, 2003.
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. If you are going to run “millions” of instruction, compulsory misses are insignificant
  - Solution: increase block size (increases miss penalty; very large blocks could increase miss rate)

- **Capacity**:
  - Cache cannot contain all blocks accessed by the program
  - Solution: increase cache size (may increase access time)

- **Conflict** (collision):
  - Multiple memory locations mapped to the same cache location
  - Solution 1: increase cache size
  - Solution 2: increase associativity (stay tuned) (may increase access time)
FIGURE 5.31 The miss rate can be broken into three sources of misses. This graph shows the total miss rate and its components for a range of cache sizes. This data is for the SPEC2000 integer and floating-point benchmarks and is from the same source as the data in Figure 5.30. The compulsory miss component is 0.006% and cannot be seen in this graph. The next component is the capacity miss rate, which depends on cache size. The conflict portion, which depends both on associativity and on cache size, is shown for a range of associativities from one-way to eight-way. In each case, the labeled section corresponds to the increase in the miss rate that occurs when the associativity is changed from the next higher degree to the labeled degree of associativity. For example, the section labeled two-way indicates the additional misses arising when the cache has associativity of two rather than four. Thus, the difference in the miss rate incurred by a direct-mapped cache versus a fully associative cache of the same size is given by the sum of the sections marked eight-way, four-way, two-way, and one-way. The difference between eight-way and four-way is so small that it is difficult to see on this graph. Copyright © 2009 Elsevier, Inc. All rights reserved.
Average Memory Access Time (AMAT)

- A larger cache will have a longer access time. An increase in hit time will likely add another stage to the pipeline. At some point the increase in hit time for a larger cache will overcome the improvement in hit rate leading to a decrease in performance.
- Average Memory Access Time (AMAT) is the average to access memory considering both hits and misses

\[ \text{AMAT} = \text{Time for a hit} + \text{Miss rate} \times \text{Miss penalty} \]

- What is the AMAT for a processor with a 20 psec clock, a miss penalty of 50 clock cycles, a miss rate of 0.02 misses per instruction and a cache access time of 1 clock cycle?