APPENDIX D
VICTIM CACHE STRATEGIES

William Stallings

Supplement to
Computer Organization and Architecture, Ninth Edition
Prentice Hall 2012
ISBN: 013293633X
http://williamstallings.com/ComputerOrganization

This appendix looks at two cache strategies mentioned in Chapter 4: victim cache and selective victim cache.

D.1 VICTIM CACHE

Recall from Chapter 4 that an advantage of the direct mapping technique is simple and inexpensive to implement. Its main disadvantage is that there is a fixed cache location for any given block. Thus, if a program happens to reference words repeatedly from two different blocks that map into the same line, then the blocks will be continually swapped in the cache, and the hit ratio will be low (a phenomenon known as *thrashing*). On the other hand, with fully associative mapping, there is flexibility as to which block to replace when a new block is read into the cache. Replacement algorithms are designed to maximize the hit ratio. The principal disadvantage of associative mapping is the complex circuitry required to examine the tags of all cache lines in parallel.

The victim cache approach, proposed by Jouppi [JOUP90], is a strategy designed to combine the fast hit time of direct mapping yet still avoid thrashing. To achieve this objective, the direct-mapped cache is supplemented with a small associative cache known as the victim cache. A line removed from the direct-mapped is temporarily stored in the victim cache, which maintains a small number of lines using a FIFO (first-in-first-out) replacement strategy. Jouppi found that a 4-line victim cache removed 20% to 95% of misses in the direct-mapped cache.

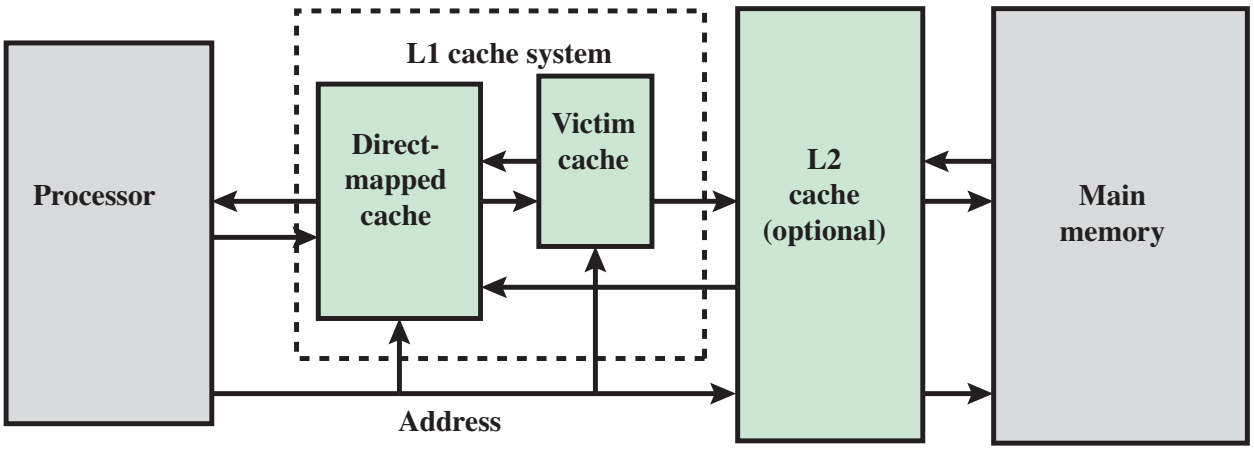

Figure D.1 Position of Victim Cache

Figure D.1 is a simple block diagram illustrating the location of the victim cache in the overall memory hierarchy. The victim cache can be considered to be part of the L1 cache system. The next lower level of the memory hierarchy can be an L2 cache or the main memory.

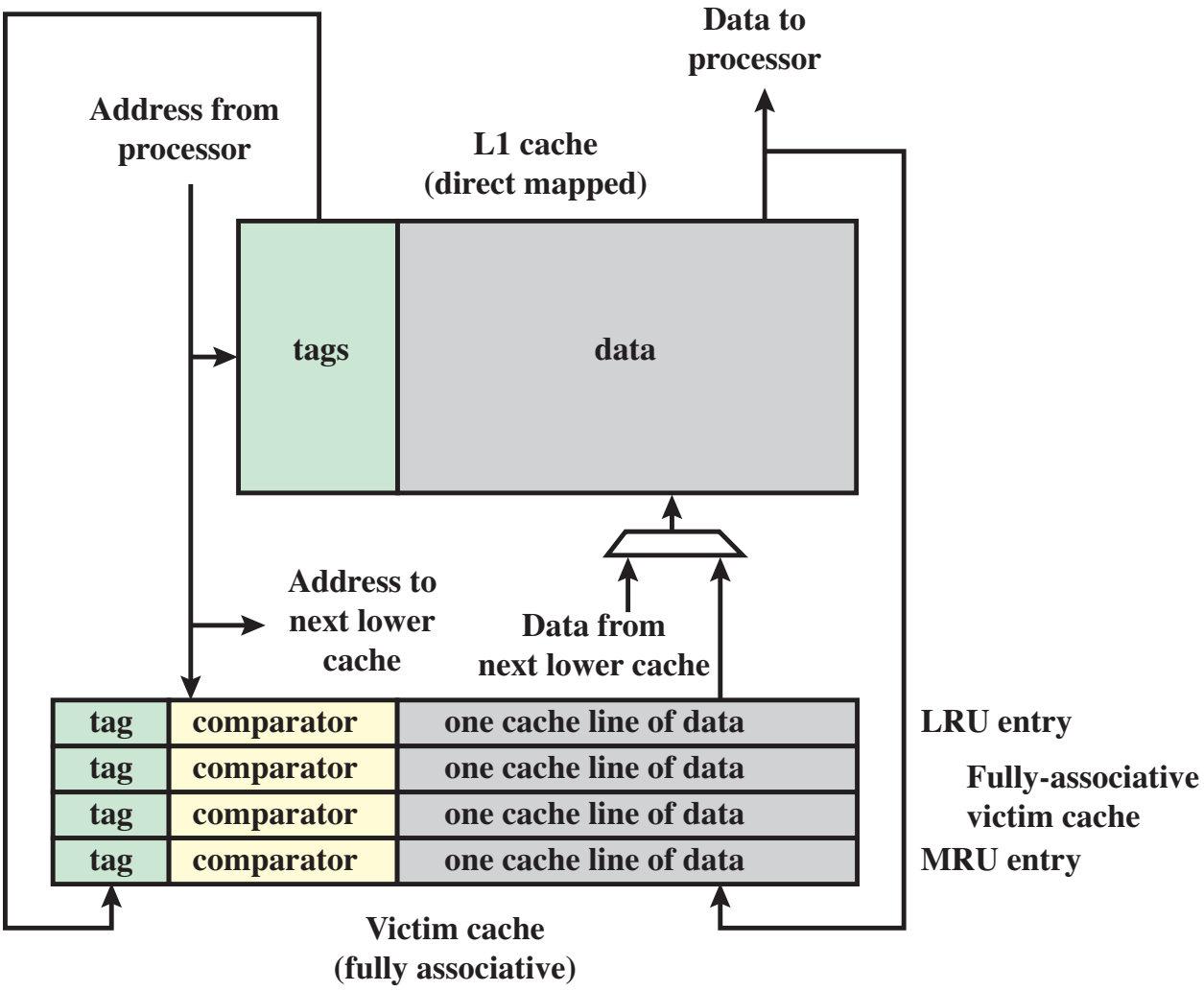

Figure D.2 Victim Cache Organization

Figure D.2 provides a more detailed look at the victim cache organization. In Jouppi's proposal, the victim cache contains four lines of data. The L1 cache is direct-mapped, so that each cache line consists of a block of data from memory plus a small tag (see Figures 4.9 and 4.10). The victim cache is associative, so that each line contains one block of data from memory plus a large tag (see Figures 4.11 and 4.12). For clarity, the tag in the victim cache is depicted as consisting of a tag equal in length to the direct-mapped cache and a comparator. In fact, looking back at Figures 4.10

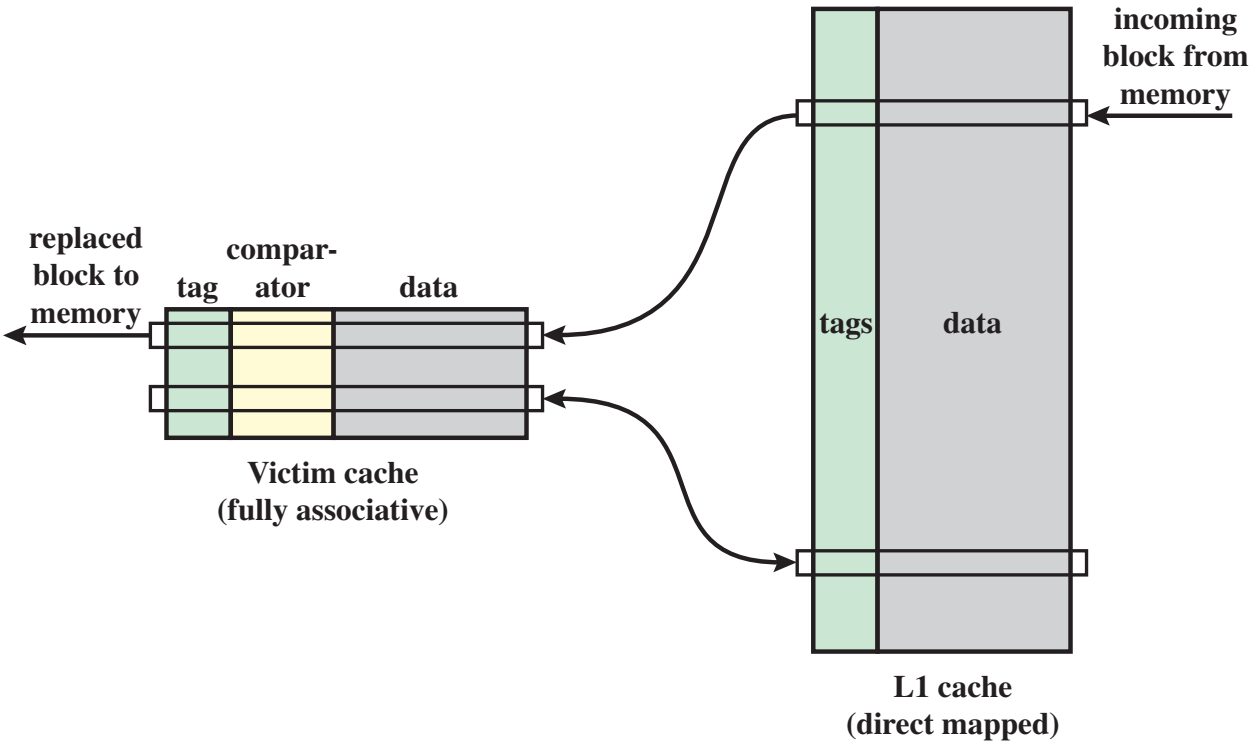

Figure D.3 Victim Cache Operation

and 4.12, the comparator is equivalent to the line field in the direct-mapped scheme. The tag plus comparator in the victim cache uniquely identify a block of memory, so that a memory reference from the processor can do a parallel search of all entries in the associative cache to determine if the desired line is present.

Figure D.3 suggests the operation of the victim cache. The data is arranged in such a way that the same line is never present in both the L1 cache and the victim cache at the same time. There are two cases to consider for managing the movement of data between the two caches:

Case 1: Processor reference to memory misses in both the L1 cache and the victim cache.

a. The required block is fetched from main memory (or the L2 cache if present) and placed into the L1 cache.

b. The replaced block in the main cache is moved to the victim cache. There is no replacement algorithm. With a direct-mapped cache, the line to be replaced is uniquely determined.

c. The victim cache can be viewed as a FIFO queue or, equivalently, a circular buffer. The item that has been in the victim cache the longest is removed to make room for the incoming line. The replaced line is written back the main memory if it is dirty (has been updated).

Case 2: Processor reference to memory misses the direct-mapped cache but hits the victim cache.

a. The block in the victim cache is promoted to the direct-mapped cache.

b. The replaced block in the main cache is swapped to the victim cache.

Note that with the FIFO discipline, the victim cache achieves true LRU (least recently used) behavior. Any reference to the victim cache pulls the referenced block out of the victim cache; thus the LRU block in the victim cache will, by definition, be the oldest one there.

The term victim is used for the following reason. When a new block is brought into the L1 cache, the replacement algorithm chooses a line to be replaced. That line is the "victim" of the replacement algorithm.

D.2 SELECTIVE VICTIM CACHE

[STIL94] proposes an improvement to the victim cache scheme known as selective victim cache. In this scheme, incoming blocks into the first-level cache are placed selectively in the main cache or the victim cache by the use of a prediction scheme based on their past history of use. In addition, interchanges of blocks between the main cache and the victim cache are also performed selectively.

Incoming Blocks from Memory

In the victim cache scheme, incoming blocks from memory (or L2 cache if present) are always loaded into the direct-mapped cache, with one of the direct cache blocks being replaced and moved to the victim cache, which in turn discards one of its blocks (writing it back to memory if necessary). The net effect is that when a new block is brought into the L1 cache, it is a victim cache block that is replaced in the total L1 cache system.

With the selective victim cache, a decision is made whether to replace the corresponding line in the direct-mapped cache (which is then moved to the victim cache) or to replace a line in the victim cache, choosing the LRU line for replacement. A prediction algorithm is used to determine which of the two conflicting lines is more likely to be referenced in the future. If the incoming line is found to have a higher probability than the conflicting line in the main cache, the latter is moved to the victim cache and the new line takes its place in the main cache; otherwise, the incoming line is directed to the victim cache.

Swap Between Direct-Mapped Cache and Victim Cache

In the victim cache scheme, if there is a miss in the direct-mapped cache and a hit in the victim cache, then there is a swap of lines between the direct-mapped cache and the victim cache. With the selective victim cache the prediction algorithm is invoked to determine if the accessed block in victim cache is more likely to be accessed in the future than the block in main cache it is conflicting with. If the prediction algorithm decides that the block in the victim cache is more likely to be referenced again than the conflicting block in the main cache, an interchange is performed between the two blocks; no such interchange is performed otherwise. In both cases the block in the victim cache is marked as the most recently used.

The prediction algorithm used in the selective victim cache scheme is referred to as the dynamic exclusion algorithm, proposed in [MAFA92]. Both [MAFA92] and [STIL94] have good descriptions of the algorithm.

References

JOUP90 Jouppi, N. "Improving Direct-Mapped Cache Performance by the Addition of a Small Fully-Associative Cache and Prefetch Buffers." *Proceedings, 17th Annual International Symposium on Computer Architecture*, May 1990.

MCFA92 McFarling, S. "Cache Replacement with Dynamic Exclusion." *Proceedings, 19th Annual International Symposium on Computer Architecture*, May 1992.

STIL94 Stiliadis, D., and Varma, A. "Selective Victim Caching: A Method to Improve the Performance of Direct-Mapped Caches." *Communications of the ACM*, January 1987.