6.823 Computer System Architecture
Victim Cache

http://csg.csail.mit.edu/6.823/

Although direct-mapped caches have an advantage of smaller access time than set-associative caches, they have more conflict misses due to their lack of associativity. In order to reduce these conflict misses, N. Jouppi proposed the *victim caching* where a small fully-associative back up cache, called victim cache, is added to a direct-mapped L1 cache to hold recently evicted cache lines.

The following diagram shows how a victim cache can be added to a direct-mapped L1 data cache. Upon a data access, the following chain of events takes place:

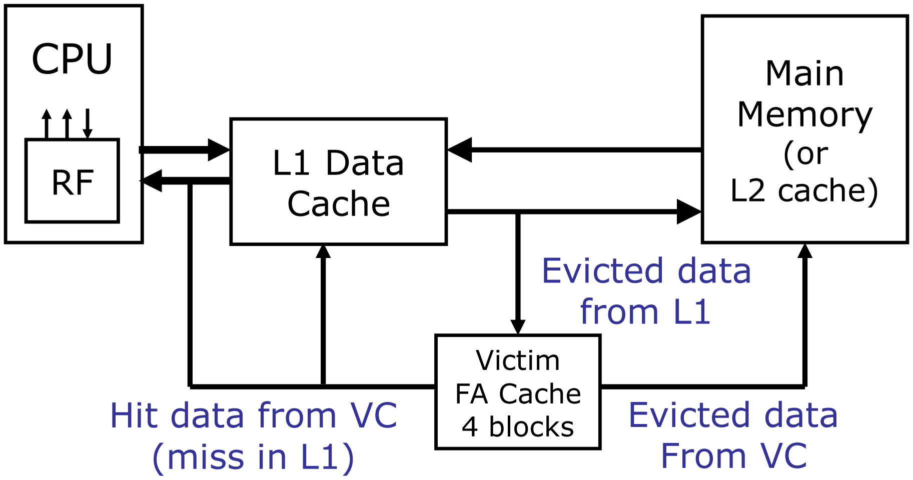

Figure H5-A: A Victim Cache Organization

1. The L1 data cache is checked. If it holds the data requested, the data is returned.
2. If the data is not in the L1 cache, the victim cache is checked. If it holds the data requested, the data is moved into the L1 cache and sent back to the processor. The data evicted from the L1 cache is put in the victim cache, and put at the end of the FIFO replacement queue.
3. If neither of the caches holds the data, it is retrieved from memory, and put in the L1 cache. If the L1 cache needs to evict old data to make space for the new data, the old data is put in the victim cache and placed at the end of the FIFO replacement queue. Any data that needs to be evicted from the victim cache to make space is written back to memory or discarded, if unmodified.

Note that the two caches are *exclusive*. That means that the same data cannot be stored in both L1 and victim caches at the same time.

Reference

1. Norm Jouppi, Improving direct-mapped cache performance by the addition of a small fully-associative cache and prefetch buffers, in the Proceedings of the 17th *International Symposium on Computer Architecture* (ISCA), pages 364--373, Seattle, Washington, May 1990.