

CLAIMS

1. A microprocessor, comprising:
 - a plurality of processing units;
 - a plurality of caches arranged into a plurality of cache layers, the plurality of caches including a plurality of first caches within a first cache layer and one or more second caches within a second cache layer, a particular second cache in the second cache layer serving as a backing store for at least a particular first cache in the first cache layer; and
 - control logic that configures at least the particular second cache to perform at least the following:
 - receive, from the particular first cache, a logging request referencing a particular memory address; and
 - based on the request, determine whether a cache line corresponding to the memory address is present in the particular second cache, and
 - when the cache line is not present in the particular second cache, perform one of:
 - cause the cache line to be logged when there does not exist a third cache that participates in logging and that serves as a backing store for at least the particular second cache, or
 - forward the request to the third cache when the third cache does exist; or
 - when the cache line is present in the particular second cache, perform at least one of:
 - cause the cache line to be logged when the cache line is (i) not determined by the particular second cache to be logged, or (ii) determined by the particular second cache to be logged but the particular second cache has not determined that the first cache is aware of a current value stored in the cache line of the particular second cache; or
 - determine that cache line need not be logged when (i) the cache line is determined by the particular second cache to be logged, and (ii) it is determined that the first cache is aware of the current value stored in the cache line of the particular second cache.
2. The microprocessor as recited in claim 1, wherein causing the cache line to be logged comprises:

logging the cache line within a trace buffer; and

marking the cache line as being logged within the particular second cache.

3. The microprocessor as recited in claim 1, wherein causing the cache line to be logged comprises one of:

instructing the first cache to log a value of the particular memory address directly;

or

instructing the first cache to log a reference to a prior log entry for the particular memory address.

4. The microprocessor as recited in claim 1, wherein causing the cache line to be logged when the cache line is not determined by the particular second cache to be logged comprises:

determining that the third cache layer exists, and

notifying the third cache that the cache line has been logged by value by the particular second cache.

5. The microprocessor as recited in claim 1, wherein the first cache layer comprises an L1 cache layer, and wherein the second cache layer comprises an L2 cache layer or an L3 cache layer.

6. The microprocessor as recited in claim 1, the control logic also configuring at least the particular second cache to:

receive a message from the first cache indicating that another cache line in the first cache that also corresponds to the memory address is marked as not logged within the first cache; and

based on the message, mark the cache line as not logged within the particular second cache.

7. The microprocessor as recited in claim 6, wherein the other cache line is marked as not logged within the first cache based on the other cache line being written to by a processing unit corresponding to the first cache while logging is disabled for the processing unit.

8. The microprocessor as recited in claim 1, the control logic also configuring at least the particular second cache to:

receive a message from the first cache requesting the cache line for writing; and

send the cache line to the first cache, the cache line being marked as not logged based at least on logging being disabled for the first cache.

9. A method of an upper cache layer determining how to log an influx by a

lower cache layer, based on a logging request by the lower cache layer, the method being implemented at a computing device that includes (i) a plurality of processing units, (ii) a plurality of caches arranged into a plurality of cache layers, the plurality of caches including a plurality of first caches within a first cache layer and one or more second caches within a second cache layer, a particular second cache in the second cache layer serving as a backing store for at least a particular first cache in the first cache layer, the method comprising:

receiving, from the particular first cache, a logging request referencing a particular memory address; and

based on the request, determining whether a cache line corresponding to the memory address is present in the particular second cache, and

when the cache line is not present in the particular second cache, performing one of:

causing the cache line to be logged when there does not exist a third cache that participates in logging and that serves as a backing store for at least the particular second cache, or

forwarding the request to the third cache when the third cache does exist; or

when the cache line is present in the particular second cache, performing at least one of:

causing the cache line to be logged when the cache line is (i) not determined by the particular second cache to be logged, or (ii) determined by the particular second cache to be logged but the particular second cache has not determined that the first cache is aware of a current value stored in the cache line of the particular second cache; or

determining that cache line need not be logged when (i) the cache line is determined by the particular second cache to be logged, and (ii) it is determined that the first cache is aware of the current value stored in the cache line of the particular second cache.

10. The method of claim 9, wherein causing the cache line to be logged comprises:

logging the cache line within a trace buffer; and

marking the cache line as being logged within the particular second cache.

11. The method of claim 9, wherein causing the cache line to be logged comprises one of:

instructing the first cache to log a value of the particular memory address directly;
or

instructing the first cache to log a reference to a prior log entry for the particular memory address.

12. The method of claim 9, wherein causing the cache line to be logged when the cache line is not determined by the particular second cache to be logged comprises:

determining that the third cache exists, and

causing the cache line to be logged by reference based on knowledge of the third cache.

13. The method of claim 9, wherein causing the cache line to be logged when the cache line is not determined by the particular second cache to be logged comprises:

determining that the third cache layer exists, and

notifying the third cache that the cache line has been logged by value by the particular second cache.

14. The method of claim 9, wherein the first cache layer comprises an L1 cache layer, and wherein the second cache layer comprises an L2 cache layer or an L3 cache layer.

15. The method of claim 9, the control logic also configuring at least the particular second cache to:

receive a message from the first cache indicating that another cache line in the first cache that also corresponds to the memory address is marked as not logged within the first cache; and

based on the message, mark the cache line as not logged within the particular second cache.