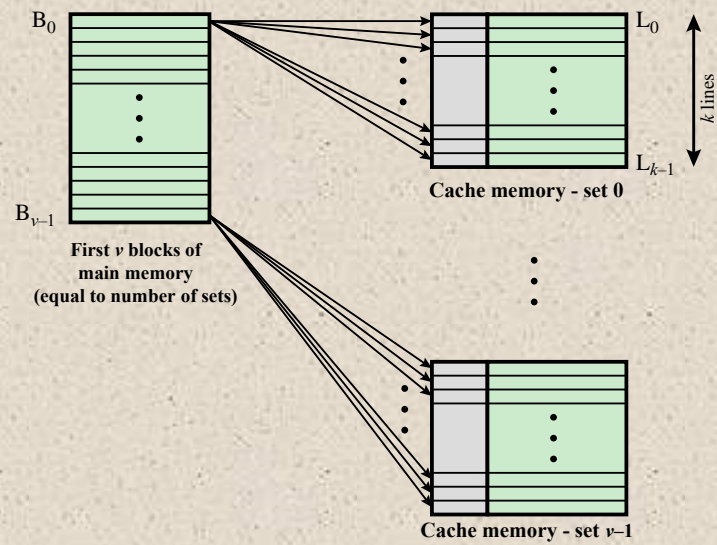


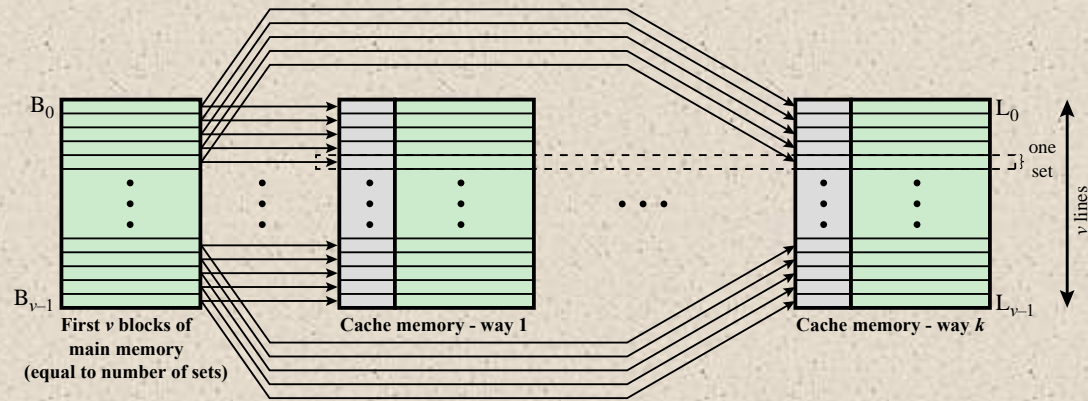


Set Associative Mapping

- Compromise that exhibits the strengths of both the direct and associative approaches while reducing their disadvantages
- Cache consists of a number of sets
- Each set contains a number of lines
- A given block maps to any line in a given set
- e.g. 2 lines per set
 - 2 way associative mapping
 - A given block can be in one of 2 lines in only one set



(a) v associative-mapped caches



(b) k direct-mapped caches

**Figure 4.13 Mapping From Main Memory to Cache:
 k -way Set Associative**

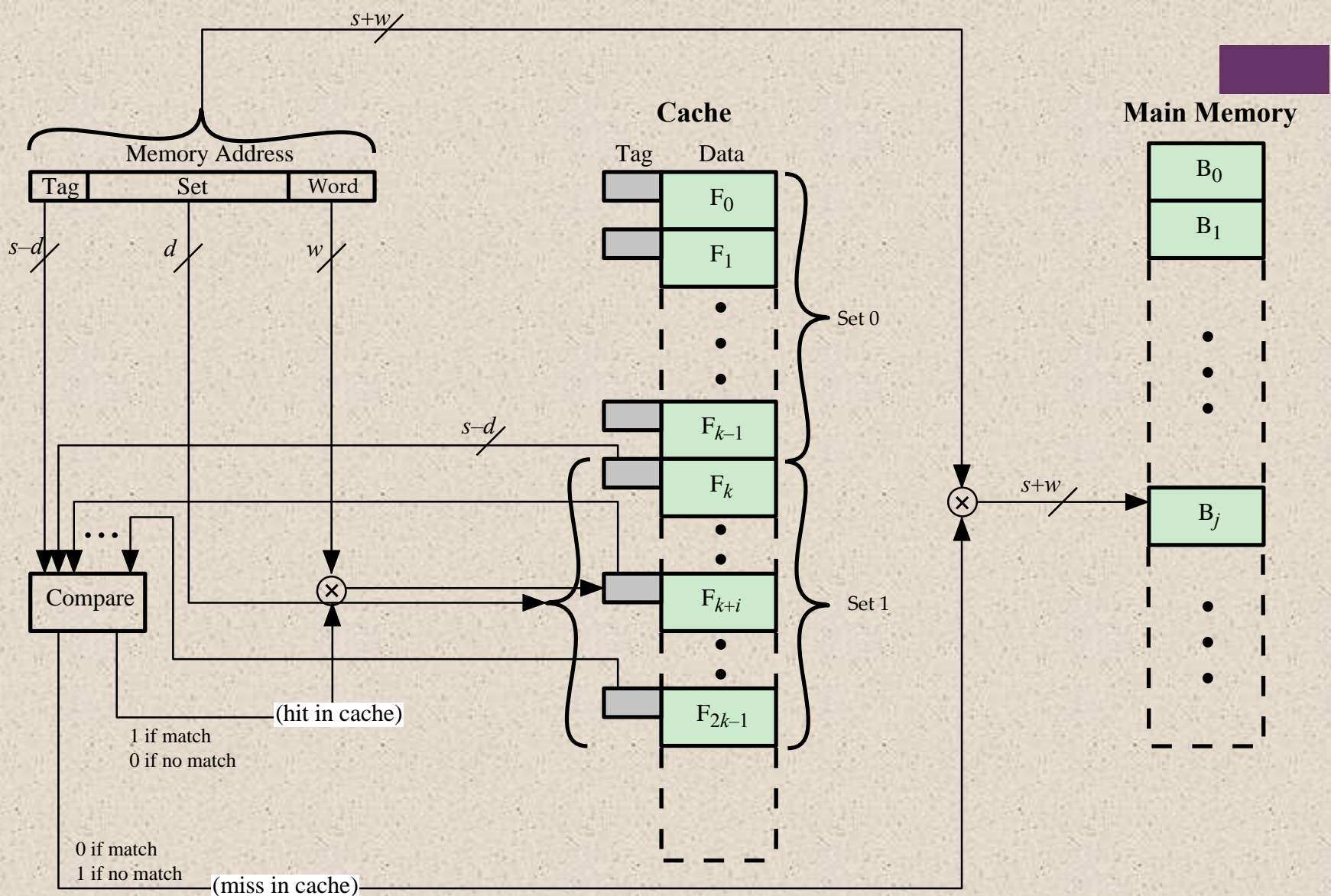


Figure 4.14 k -Way Set Associative Cache Organization



Set Associative Mapping Summary



- Address length = $(s + w)$ bits
- Number of addressable units = 2^{s+w} words or bytes
- Block size = line size = 2^w words or bytes
- Number of blocks in main memory = $2^{s+w}/2^w = 2^s$
- Number of lines in set = k
- Number of sets = $v = 2^d$
- Number of lines in cache = $m = kv = k * 2^d$
- Size of cache = $k * 2^{d+w}$ words or bytes
- Size of tag = $(s - d)$ bits



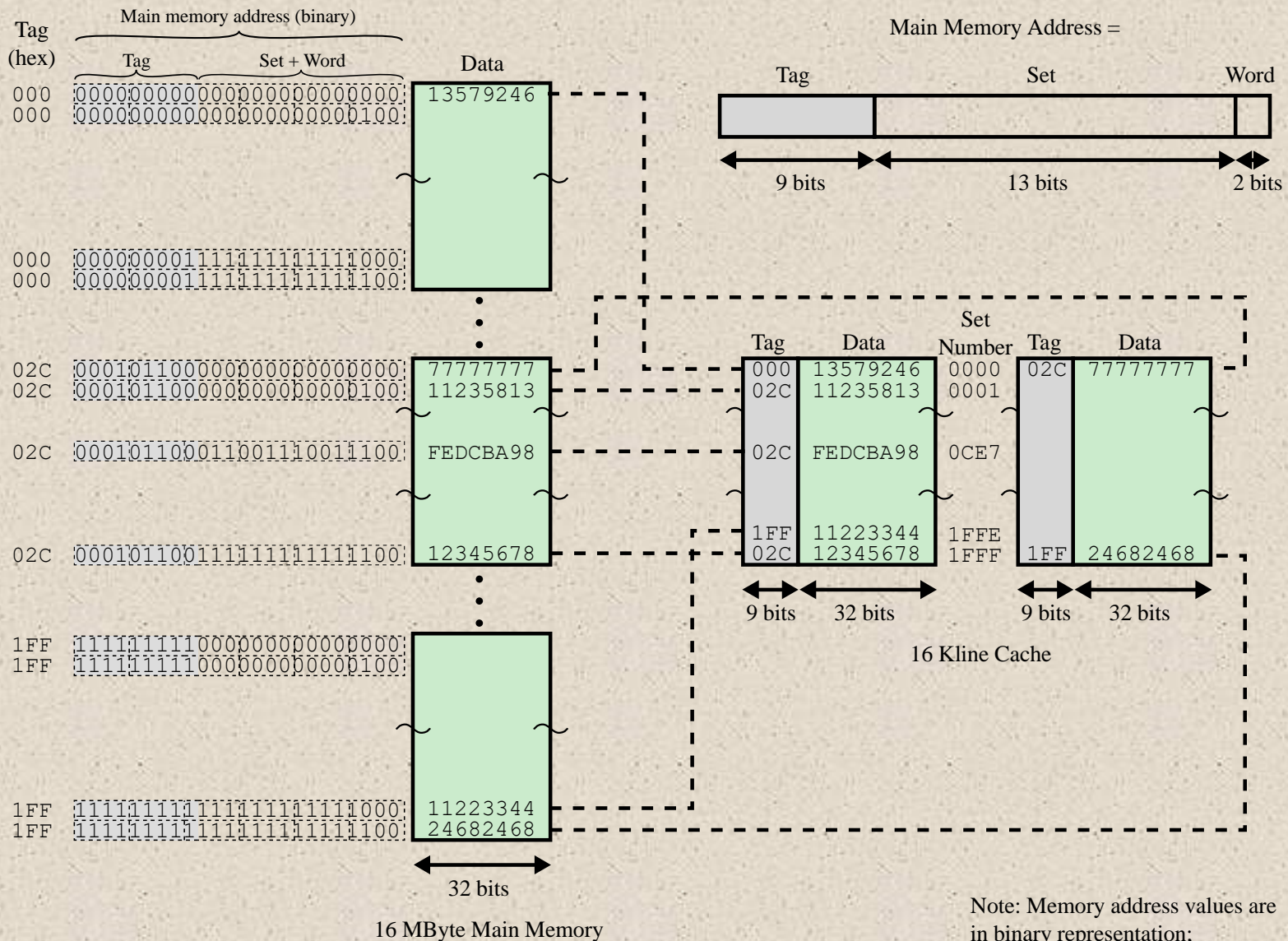


Figure 4.15 Two-Way Set Associative Mapping Example

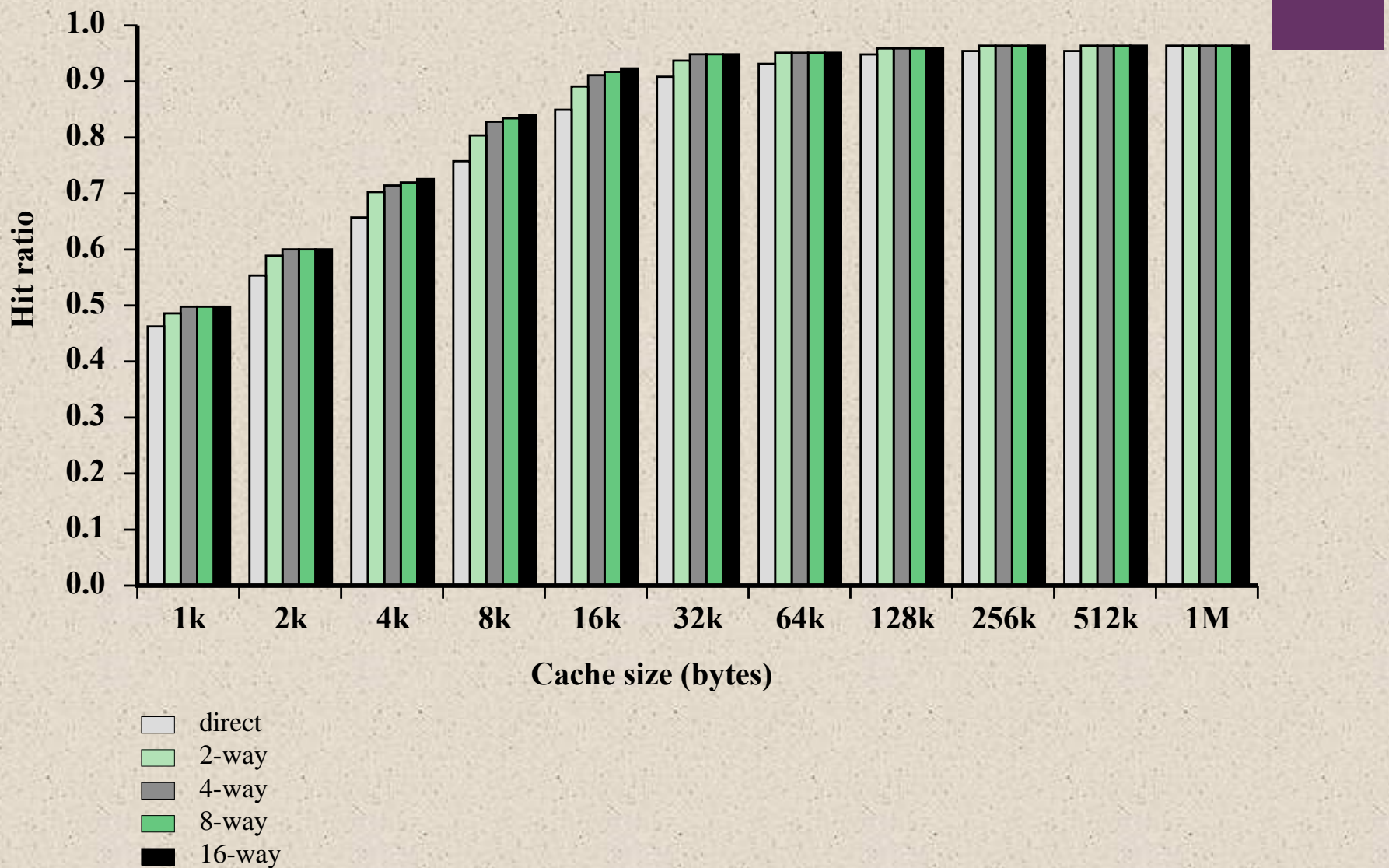
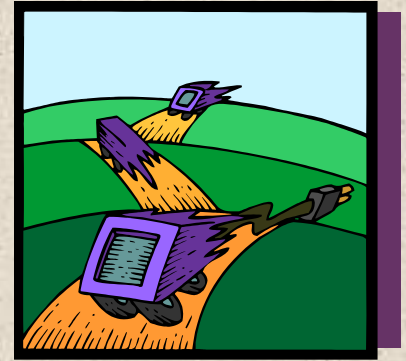


Figure 4.16 Varying Associativity over Cache Size



Replacement Algorithms



- Once the cache has been filled, when a new block is brought into the cache, one of the existing blocks must be replaced
- For direct mapping there is only one possible line for any particular block and no choice is possible
- For the associative and set-associative techniques a replacement algorithm is needed
- To achieve high speed, an algorithm must be implemented in hardware

+ The most common replacement algorithms are:

- Least recently used (LRU)
 - Most effective
 - Replace that block in the set that has been in the cache longest with no reference to it
 - Because of its simplicity of implementation, LRU is the most popular replacement algorithm
- First-in-first-out (FIFO)
 - Replace that block in the set that has been in the cache longest
 - Easily implemented as a round-robin or circular buffer technique
- Least frequently used (LFU)
 - Replace that block in the set that has experienced the fewest references
 - Could be implemented by associating a counter with each line

Write Policy

When a block that is resident in the cache is to be replaced there are two cases to consider:



If the old block in the cache has not been altered then it may be overwritten with a new block without first writing out the old block



If at least one write operation has been performed on a word in that line of the cache then main memory must be updated by writing the line of cache out to the block of memory before bringing in the new block

There are two problems to contend with:



More than one device may have access to main memory



A more complex problem occurs when multiple processors are attached to the same bus and each processor has its own local cache - if a word is altered in one cache it could conceivably invalidate a word in other caches

+ Write Through and Write Back



- Write through
 - Simplest technique
 - All write operations are made to main memory as well as to the cache
 - The main disadvantage of this technique is that it generates substantial memory traffic and may create a bottleneck
- Write back
 - Minimizes memory writes
 - Updates are made only in the cache
 - Portions of main memory are invalid and hence accesses by I/O modules can be allowed only through the cache
 - This makes for complex circuitry and a potential bottleneck

Line Size



When a block of data is retrieved and placed in the cache not only the desired word but also some number of adjacent words are retrieved

As the block size increases more useful data are brought into the cache

Two specific effects come into play:

- Larger blocks reduce the number of blocks that fit into a cache
- As a block becomes larger each additional word is farther from the requested word

As the block size increases the hit ratio will at first increase because of the principle of locality

The hit ratio will begin to decrease as the block becomes bigger and the probability of using the newly fetched information becomes less than the probability of reusing the information that has to be replaced

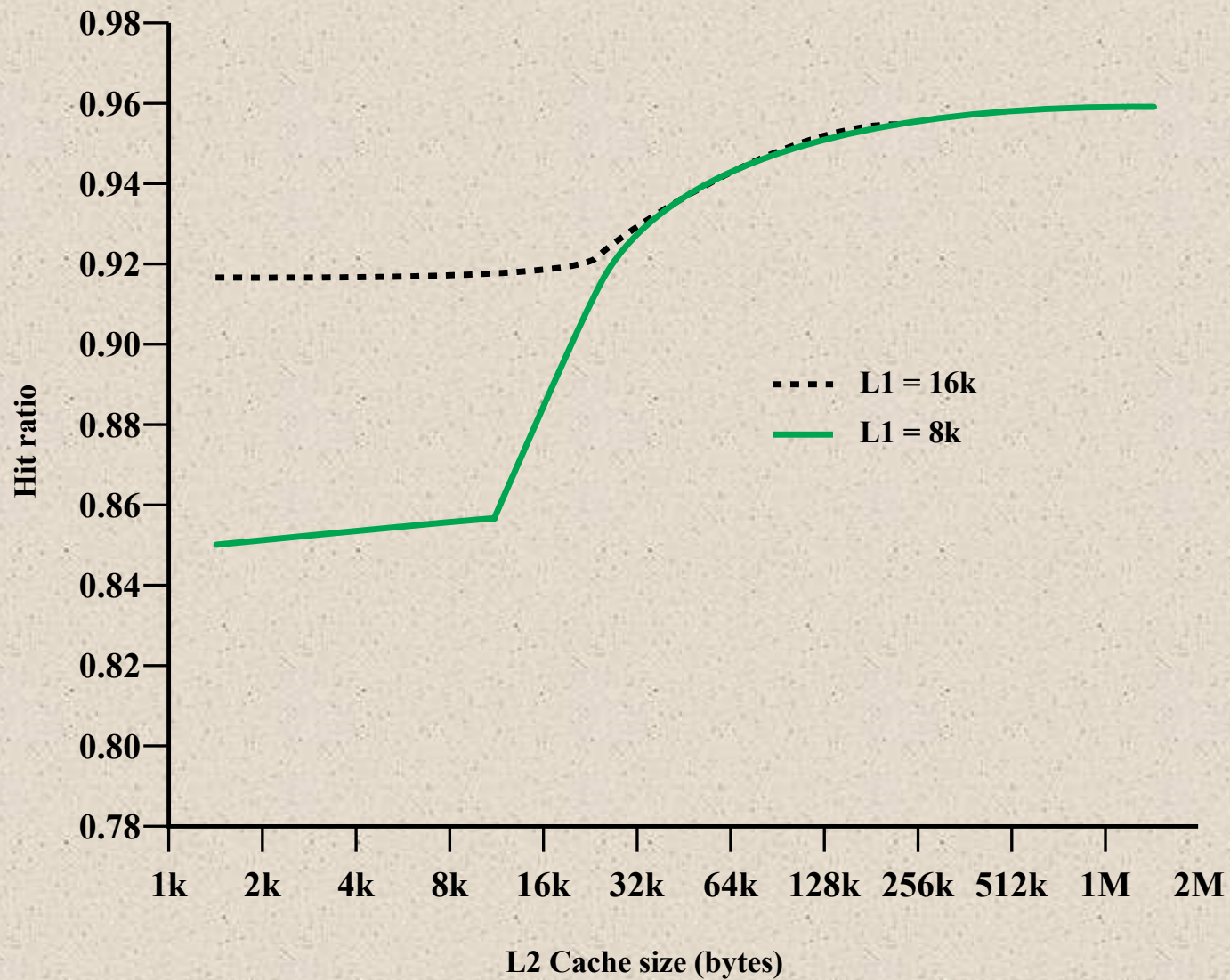


Figure 4.17 Total Hit Ratio (L1 and L2) for 8 Kbyte and 16 Kbyte L1

+ Summary

Chapter 4

Cache Memory

- Computer memory system overview

- Characteristics of Memory Systems
- Memory Hierarchy

- Cache memory principles

- Elements of cache design

- Cache addresses
- Cache size
- Mapping function
- Replacement algorithms
- Write policy
- Line size
- Number of caches