Computer Organization #1 *

FIT - HUI

May, 2013

Cache Memory

1 Average Memory Access Time

1. For a system with two levels of cache, define T_{c1} = first-level cache access time; T_{c2} = second-level cache access time; T_m = memory access time; H_1 = first-level cache hit ratio; H_2 = combined first/second level cache hit ratio. Provide an equation for T_a for a read operation.

2. Consider an L1 cache with an access time of 1 ns and a hit ratio of H = 0.95. Suppose that we can change the cache design (size of cache, cache organization) such that we increase H to 0.97, but increase access time to 1.5 ns. What conditions must be met for this change to result in improved performance?

   Explain why this result makes intuitive sense.

3. Consider a single-level cache with an access time of 2.5 ns, a line size of 64 bytes, and a hit ratio of H = 0.95. Main memory uses a block transfer capability that has a first word (4 bytes) access time of 50 ns and an access time of 5 ns for each word thereafter.

   - What is the access time when there is a cache miss? Assume that the cache waits until the line has been fetched from main memory and then re-executes for a hit.
   - Suppose that increasing the line size to 128 bytes increases the H to 0.97. Does this reduce the average memory access time?

4. On the Motorola 68020 microprocessor, a cache access takes two clock cycles. Data access from main memory over the bus to the processor takes three clock cycles in the case of no wait state insertion; the data are delivered to the processor in parallel with delivery to the cache.

   - Calculate the effective length of a memory cycle given a hit ratio of 0.9 and a clocking rate of 16.67 MHz.
   - Repeat the calculations assuming insertion of two wait states of one cycle each per memory cycle. What conclusion can you draw from the results?

*Following problems are taken from Chapter 4, William Stallings, eighth edition

2 Cache Mapping: DM and AM

1. For the hexadecimal main memory addresses 111111, 666666, BBBBBB, show the following information, in hexadecimal format:

   - Tag, Line, and Word values for a direct-mapped cache, using the format of Fig. 1.
   - Tag and Word values for an associative cache, using the format of Fig. 2.

2. Consider a machine with a byte addressable main memory of 2^{16} bytes and block size of 8 bytes. Assume that a direct mapped cache consisting of 32 lines is used with this machine.

   - How is a 16-bit memory address divided into tag, line number, and byte number?
   - Into what line would bytes with each of the following addresses be stored?
     0001 0001 0001 1011
     1100 0011 0011 0100
     1101 0000 0001 1101
     1010 1010 1010 1010
   - Suppose the byte with address 0001 1010 0001 1010 is stored in the cache. What are the addresses of the other bytes stored along with it?
   - How many total bytes of memory can be stored in the cache?
   - Why is the tag also stored in the cache?

3. Consider a memory system that uses a 32-bit address to address at the byte level, plus a cache that uses a 64-byte line size.

   - Assume a direct mapped cache with a tag field in the address of 20 bits. Show the address format and determine the following parameters: number of addressable units, number of blocks in main memory, number of lines in cache, size of tag.
   - Assume an associative cache. Show the address format and determine the following parameters: number of addressable units, number of blocks in main memory, number of lines in cache, size of tag.

4. Consider a computer with the following characteristics: total of 1Mbyte of main memory; word size of 1 byte; block size of 16 bytes; and cache size of 64 Kbytes.

   - For the main memory addresses of F0010, 01234, and CABBE, give the corresponding tag, cache line address, and word offsets for a direct-mapped cache.
   - Give any two main memory addresses with different tags that map to the same cache slot for a direct-mapped cache.

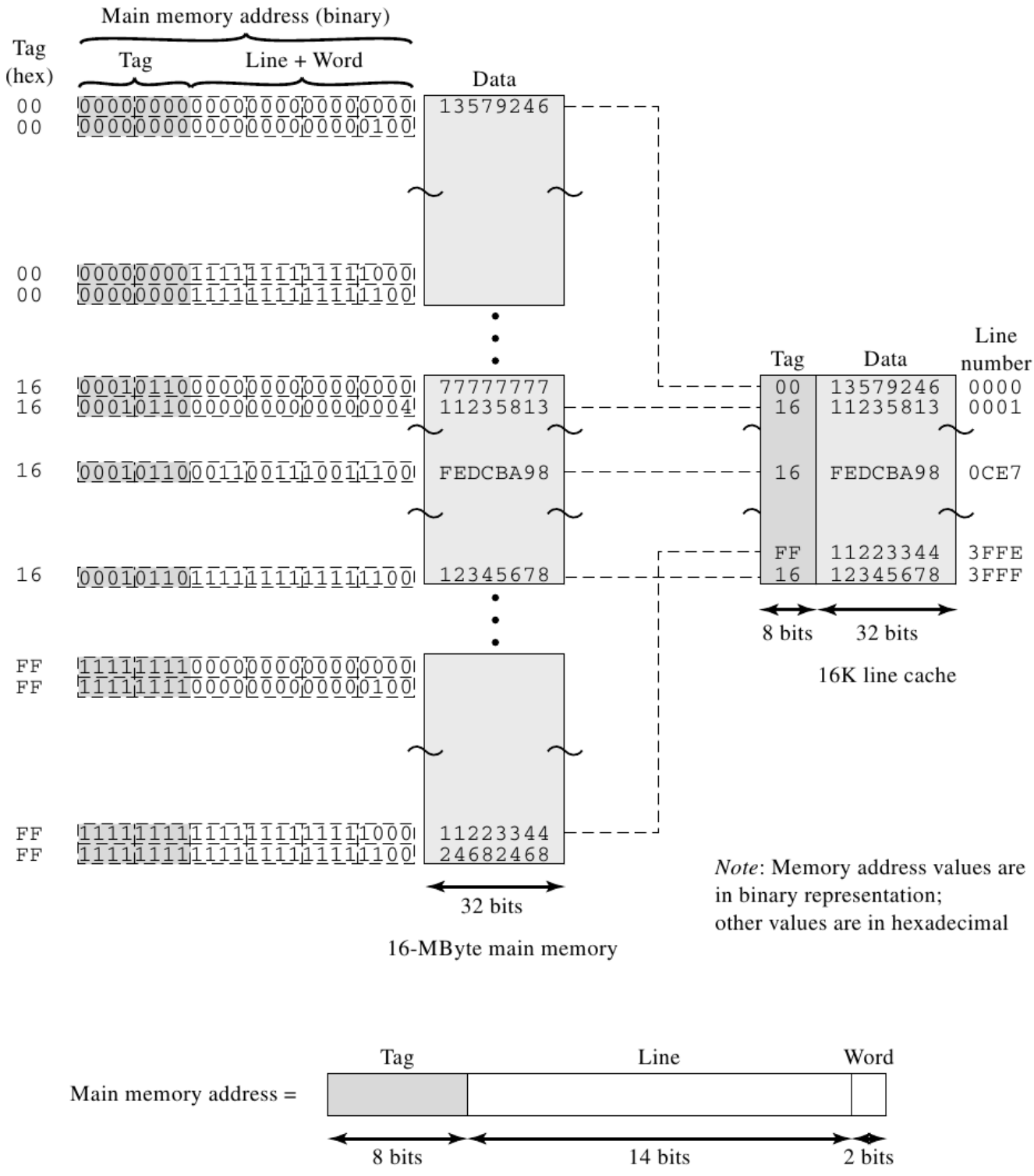

Figure 1: Direct Mapping Example

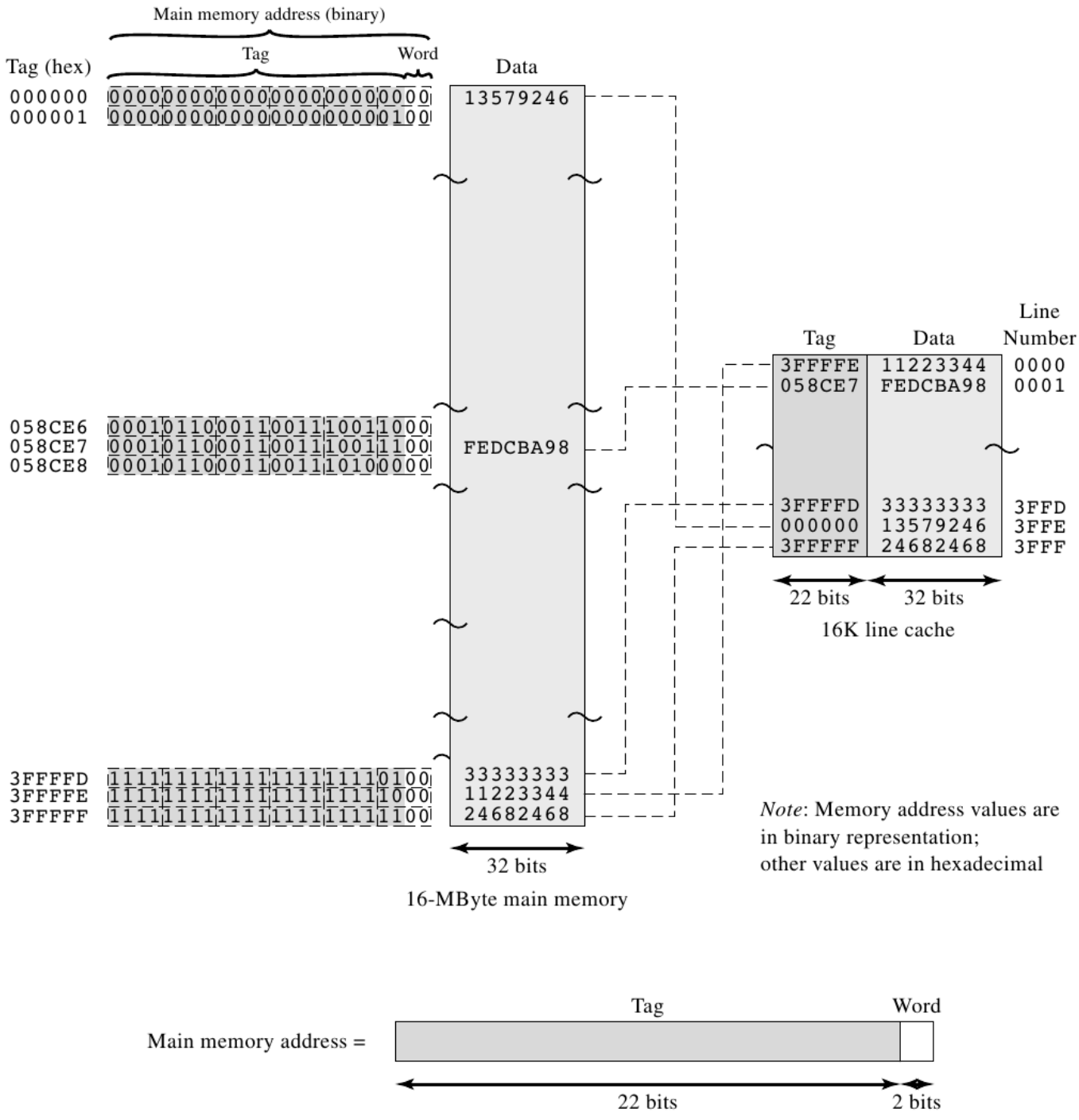

Figure 2: Associative Mapping Example