More Cache Schemes

Advantages of direct mapping:
1. The technique is simple
2. The mapping scheme is easy to implement

Disadvantage of direct mapping:
1. Each block of main memory maps to a fixed location in the cache; therefore, if two different blocks map to the same location in cache and they are continually referenced, the two blocks will be continually swapped in and out (known as thrashing).

(2) Associative Mapping

With associative mapping, any block of memory can be loaded into any line of the cache. In this case, a memory address is simply a tag and a word (note: there is no field for line #). To determine if a memory block is in the cache, each of the tags are simultaneously checked for a match.

To summarize:

Address Length is (s + w) bits

Number of addressable units is 2^{s+w} bytes

Block size = line size = 2^w bytes

Number of blocks in main memory is $\frac{2^{s+w}}{2^w} = 2^s$

Number of cache lines is undetermined

Tag size is (s) bits

Let's go through the following example:

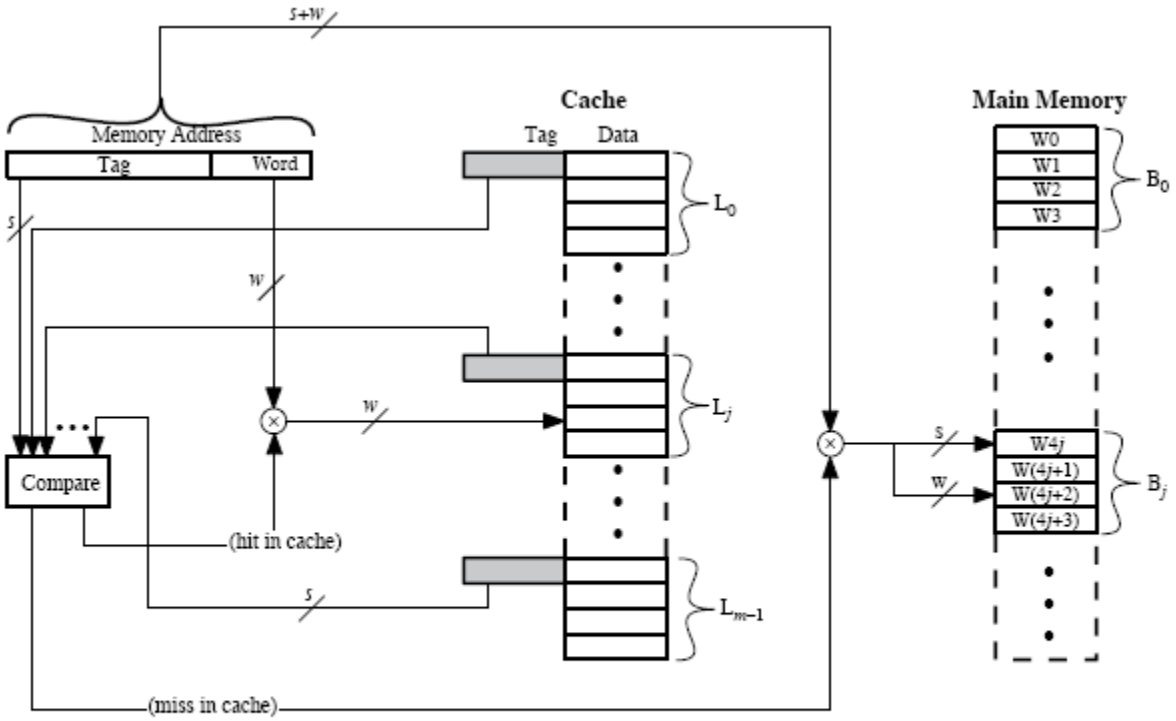


Figure 4.9 Fully Associative Cache Organization [HWAN93]

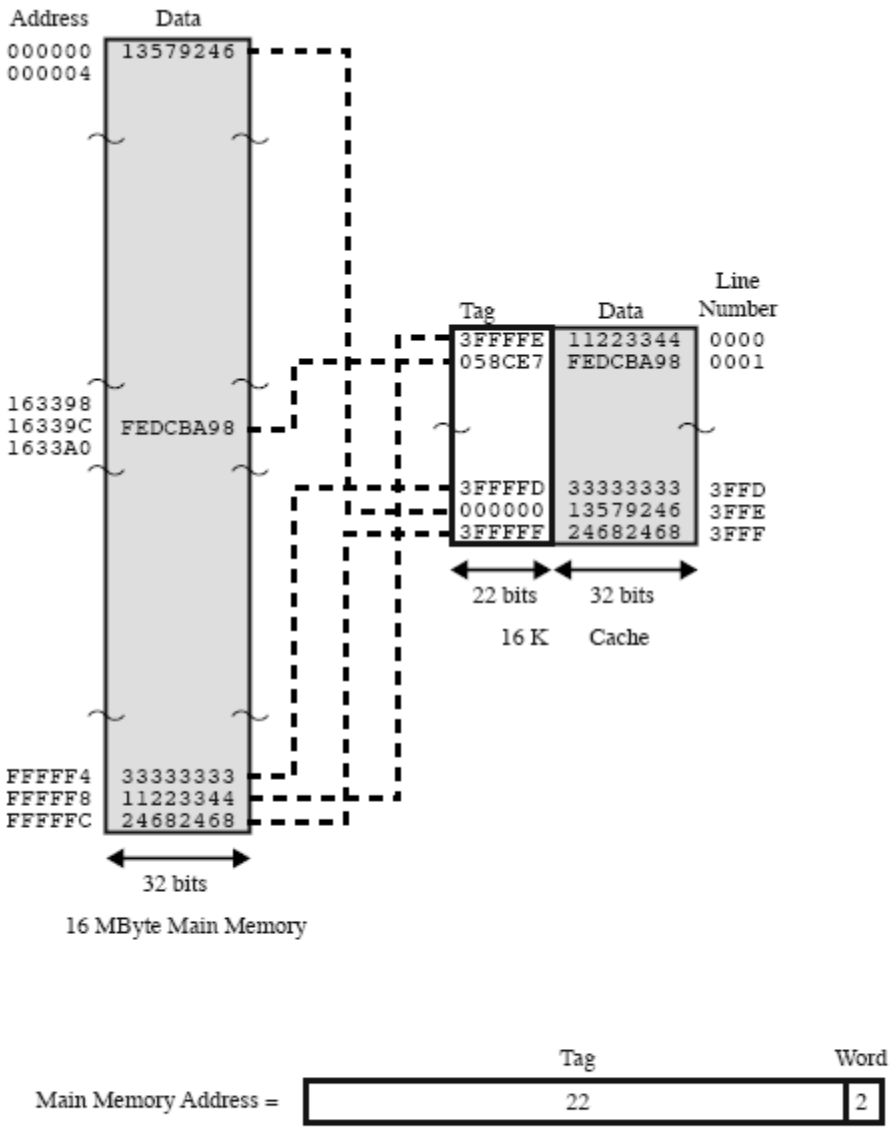


Figure 4.10 Associative Mapping Example

Advantage of associative mapping:

1. There is flexibility when mapping a block to any line of the cache

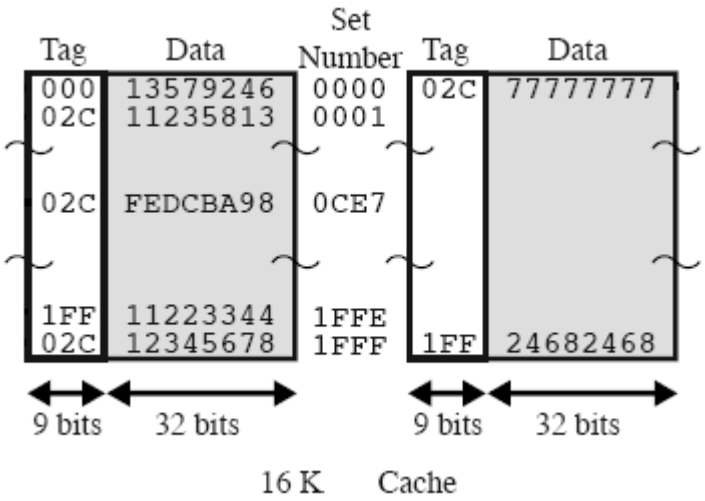
Disadvantages of associative mapping:

1. A replacement algorithm must be used to determine which line of cache to swap out
2. More space is needed for the tag field
3. The most important disadvantage is the complex circuitry needed to examine all of the tags in parallel in the cache

3) Set Associative Mapping

Set associative mapping utilizes the strengths of both direct and associative mapping while trying to reduce their disadvantages. With set associative, the cache is divided into v sets where each set consists of k lines.

Here is a two-way set associative cache that we will go into detail a little later.



The relationships are as follows:

m = v x k
i = j module v

where

i = cache set number
j = main memory block number
m = number of lines in the cache
v = set number
k = number of lines in a set (two-way set associative means two lines per set)

To summarize:

Address Length is (s + w) bits

Number of addressable units is 2^{s+w} bytes

Block size = line size = 2^w bytes

Number of blocks in main memory is $\frac{2^{s+w}}{2^w} = 2^s$

Number of lines in a set = k

Number of sets is $v = 2^d$

Number of lines in the cache is kv = k x 2^d
Size of tag is (s – d) bits

Let's go through the following example:

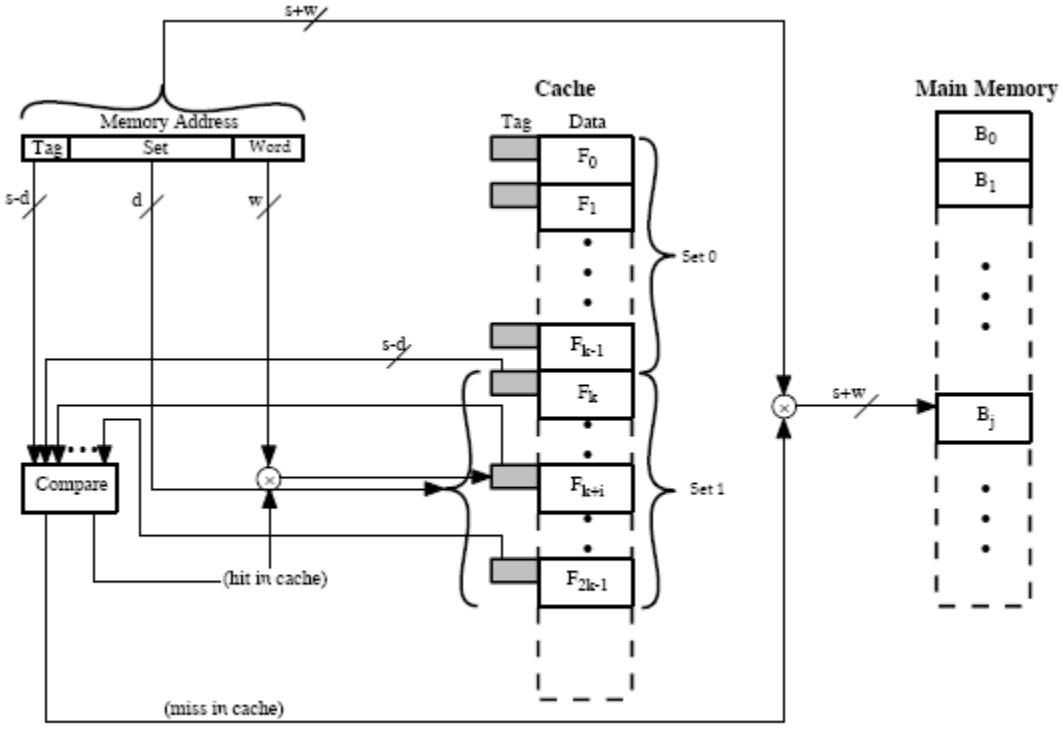
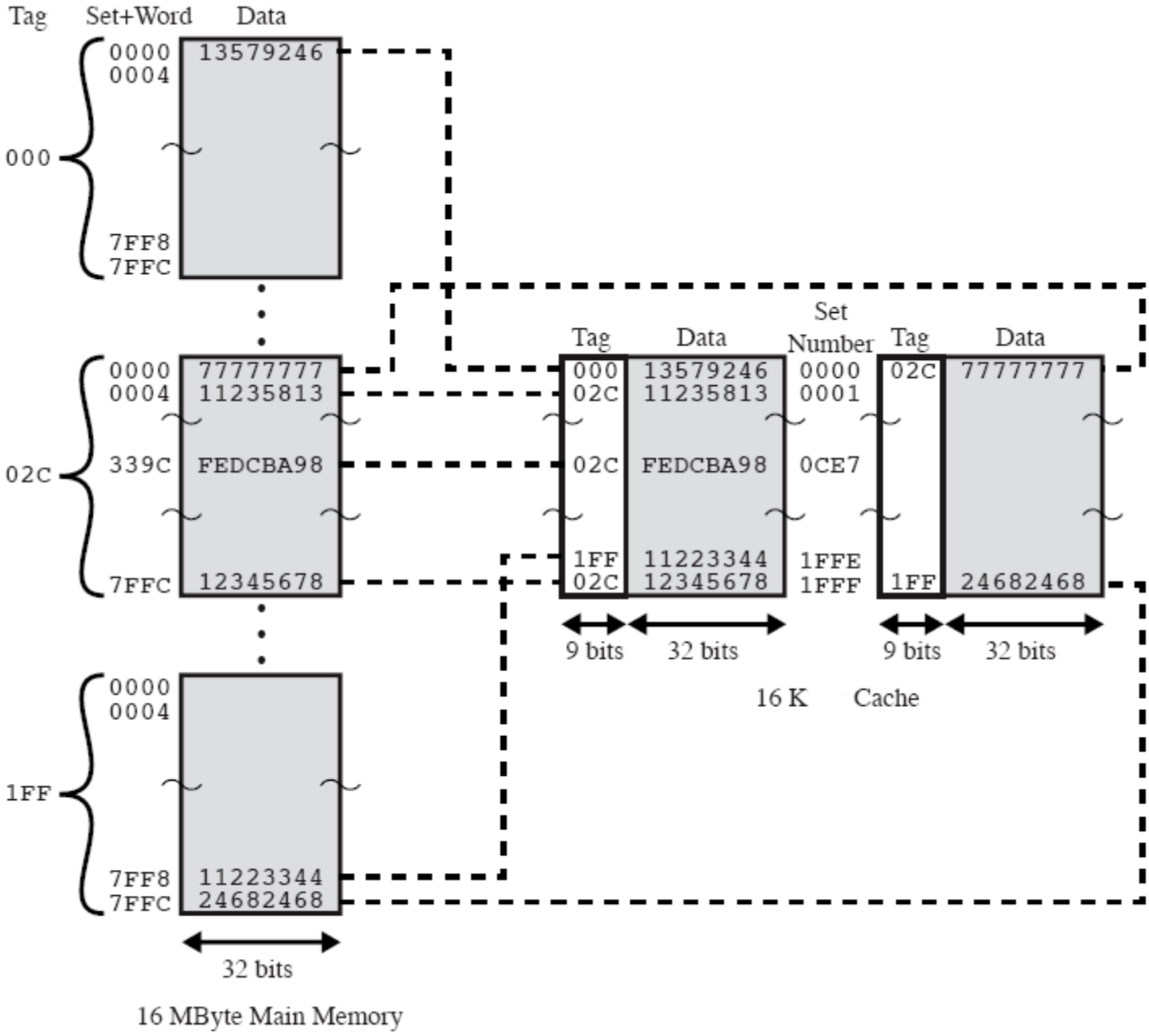


Figure 4.11 *k*-Way Set Associative Cache Organization



Reading for next time is pp. 123-131

Replacement Algorithms & Write Policies
Multilevel caches
Pentium 4 and PowerPC Cache Organizations

1. The following cache represents a 2-way set associative cache, i.e., there are two lines per set. Notice that the set ID values start at 01101101_2 and increment every other row. This is meant to imply that you are looking at a group of lines/sets toward the middle of the cache and not the entire cache. There are 14 bits for the tag, 8 bits for the set id, and 2 bits for the word id. Answer the following 3 questions based on this cache.

Tag (binary values)	**Set ID (binary values)**	**Word within block**			
		00	**01**	**10**	**11**
10100001001001	01101101	00_{16}	61_{16}	$C2_{16}$	23_{16}
11100001100100	01101101	10_{16}	71_{16}	$D2_{16}$	33_{16}
11001011010110	01101110	20_{16}	81_{16}	$E2_{16}$	43_{16}
11100101101011	01101110	30_{16}	91_{16}	$F2_{16}$	53_{16}
11110110110100	01101111	40_{16}	$A1_{16}$	02_{16}	63_{16}
10100111010101	01101111	50_{16}	$B1_{16}$	12_{16}	73_{16}
10101010111110	01110000	84_{16}	$E5_{16}$	46_{16}	$A7_{16}$
10101010010011	01110000	94_{16}	$F5_{16}$	56_{16}	$B7_{16}$
01110001001000	01110001	$A4_{16}$	$A5_{16}$	66_{16}	$C7_{16}$
00001101101101	01110001	$B4_{16}$	15_{16}	76_{16}	$D7_{16}$
01011010010010	01110010	$C4_{16}$	25_{16}	86_{16}	$E7_{16}$
10101111001011	01110010	$D4_{16}$	35_{16}	96_{16}	$F7_{16}$

1. A copy of the data from memory address 7121C5 (hex) is contained in the portion of the cache shown above. Enter the value that was retrieved from that address in the space below as a two-digit hexadecimal number. _____

2. How many lines are contained in this cache?

    a. 8
    b. 256
    c. 512
    d. 1024
    e. 16K
    f. Cannot be determined

3. How many blocks are contained in the memory space?
    a. 2^24
    b. 2^22
    c. 2^14
    d. 2^8
    e. 2^2
    f. Cannot be determined

2. Suppose physical addresses are 32 bits wide. Suppose there is a cache containing 256K words of data (not including tag bits), and each cache block contains 4 words. For each of the following cache configurations,

a. direct mapped
b. 2-way set associative
c. 4-way set associative
d. fully associative

specify how the 32-bit address would be partitioned. For example, for a direct mapped cache, you would need to specify which bits are used to select the cache entry and which bits are used to compare against the tag stored in the cache entry.

3. The Magiccomputer Corporation manufacturers a machine with a cache that contains four lines. Main memory consists of 16 blocks.

Suppose a program starts out by referencing blocks 0, 3, 4, 9, and 5, in that order. After this series of references is made, which blocks are in the cache, and what are the tags (in binary) if the cache organization is Direct Mapping.

Line	**Tag**	**Block Number**
0		
1		
2		
3		

After this series of references is made, which blocks are in the cache, and what are the tags (in binary) if the cache organization is 2-way set associative mapping.

Line	**Tag**	**Block Number**
0		
1		
2		
3		