Computer Organization and Architecture (EET 2211)

Chapter 4
CACHE MEMORY

Mapping Function

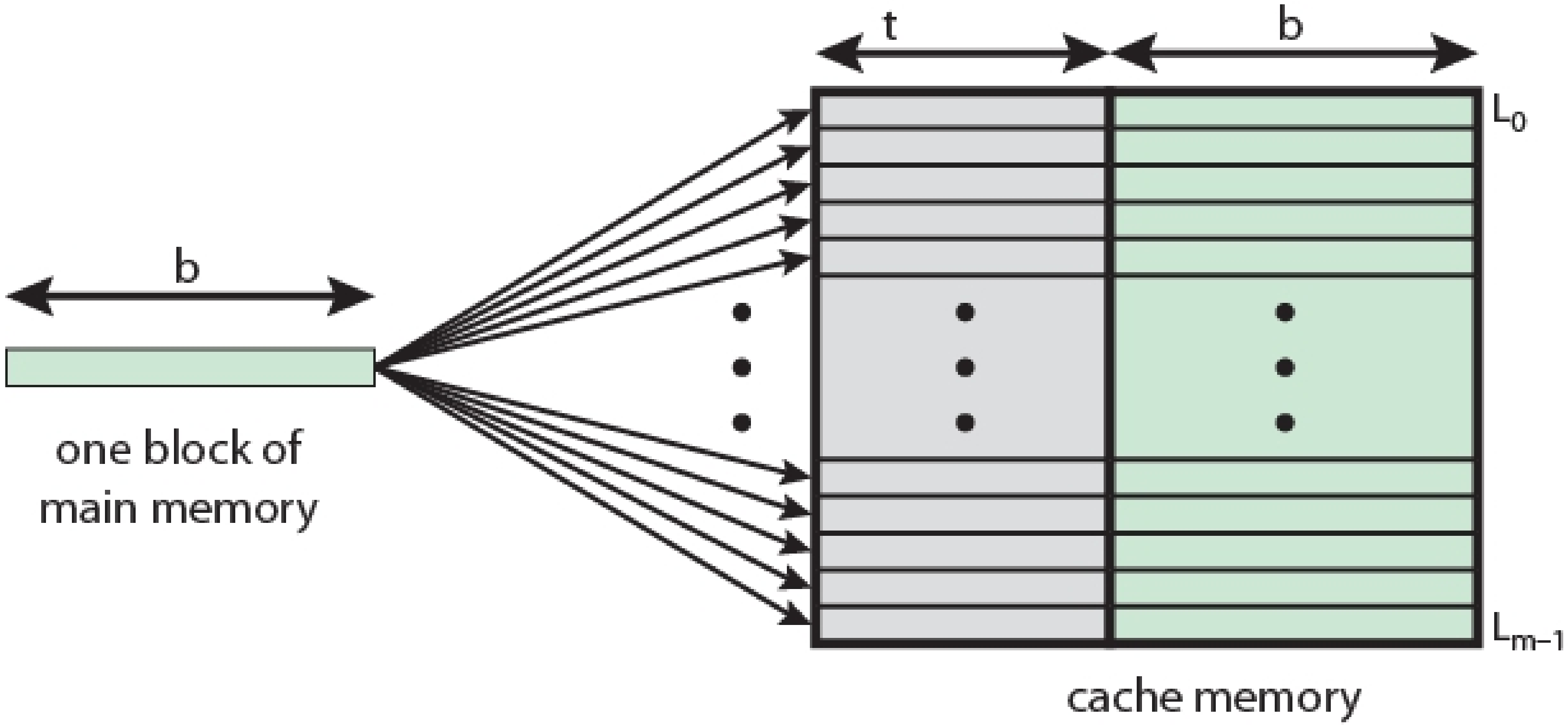
- Algorithm needed for mapping main memory blocks to cache lines
- A means is needed to determining which main memory block currently occupies a cache line
- Three Techniques:

  1. Direct
  2. Associative
  3. Set Associative

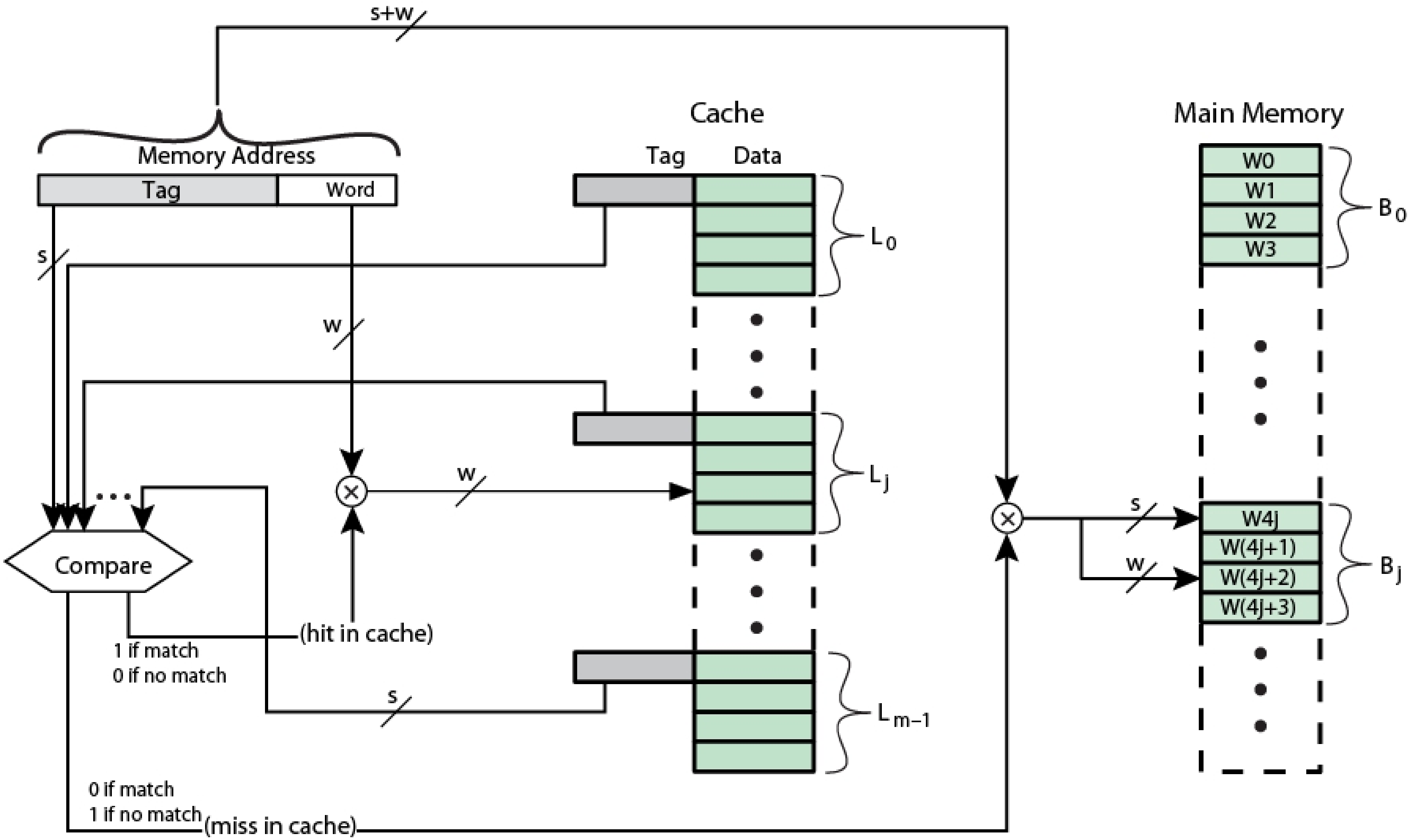
Associative Mapping

- A main memory block can load into any line of cache
- Memory address is interpreted as tag and word
- Tag uniquely identifies block of memory
- Every line’s tag is examined for a match
- Cache searching gets expensive

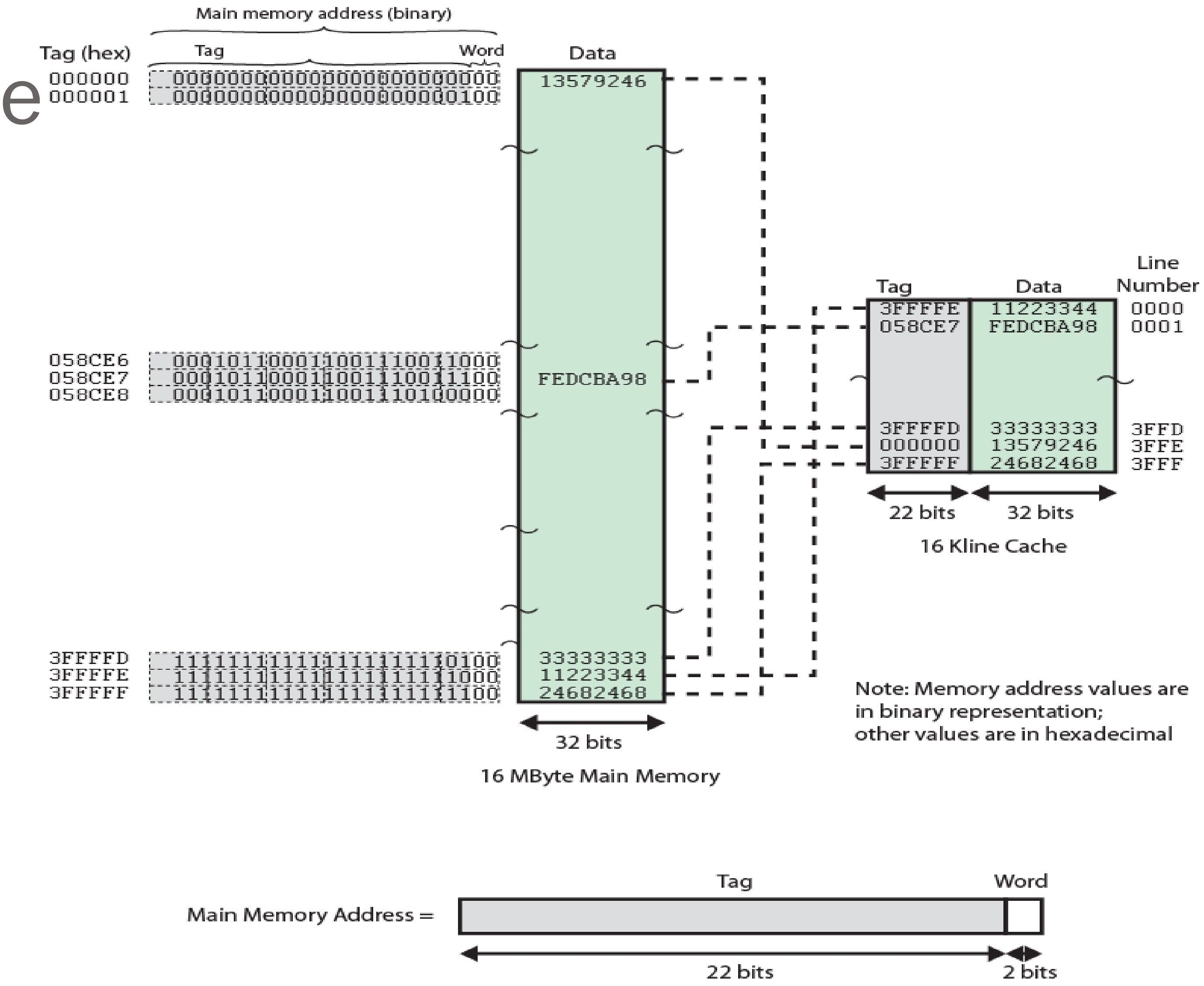
Associative Mapping from Cache to Main Memory

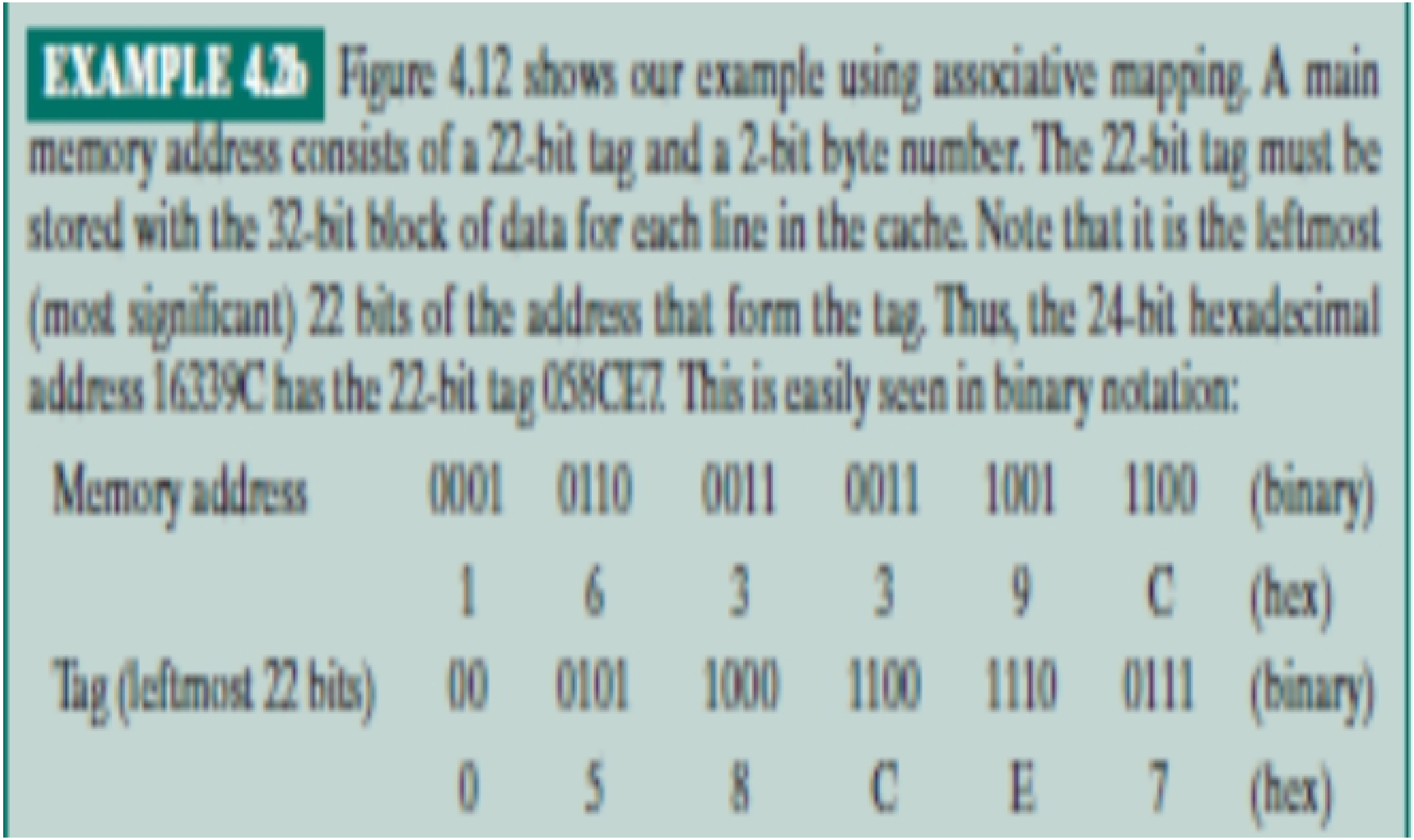


Fully Associative Cache Organization



Associative Mapping Example





EXAMPLE 4.2b Figure 4.12 shows our example using associative mapping. A main memory address consists of a 22-bit tag and a 2-bit byte number. The 22-bit tag must be stored with the 32-bit block of data for each line in the cache. Note that it is the leftmost (most significant) 22 bits of the address that form the tag. Thus, the 24-bit hexadecimal address 16339C has the 22-bit tag 058CE7. This is easily seen in binary notation:

Memory address	0001	0110	0011	0011	1001	1100	(binary)
	1	6	3	3	9	C	(hex)
Tag (leftmost 22 bits)	00	0101	1000	1100	1110	0111	(binary)
	0	5	8	C	E	7	(hex)

Addressing Structure

Tag 22 bit	Word 2 bit

- 22 bit tag stored with each 32 bit block of data
- Compare tag field with tag entry in cache to check for hit
- Least significant 2 bits of address identify which 16 bit word is required from 32 bit data block
- e.g.

Address	Tag	Data	Cache line
FFFFFC	FFFFFC	24682468	3FFF

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 cache = undetermined
- Size of tag = s bits

There is flexibility as which block to replace when a new block is read into the cache.

Disadvantage- complex circuitry required to examine the tags of all cache lines in parallel.

Set Associative Mapping

- Cache is divided into a number of sets
- Each set contains a number of lines
- A given block maps to any line in a given set
  - e.g. Block B can be in any line of set i
- e.g. 2 lines per set
  - 2 way associative mapping
  - A given block can be in one of 2 lines in only one set

Set Associative Mapping Example

- In this case the cache consists of number of sets, each of which consists of a number of lines. The relationships are

  $m = v * k$

  $i = j \text{ modulo } v$

  where

  i = cache set number

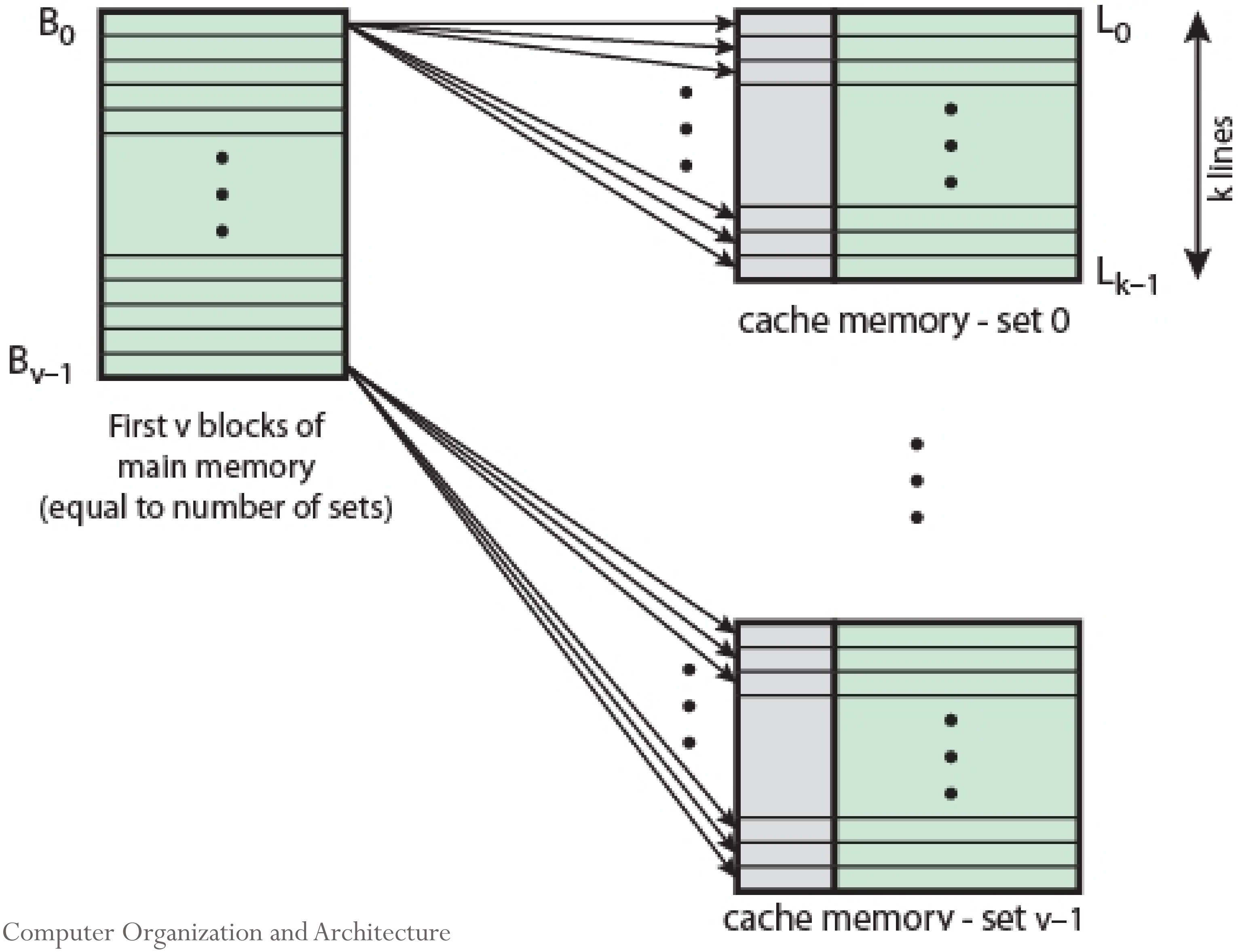
  j = main memory block number

  m = number of lines in the cache

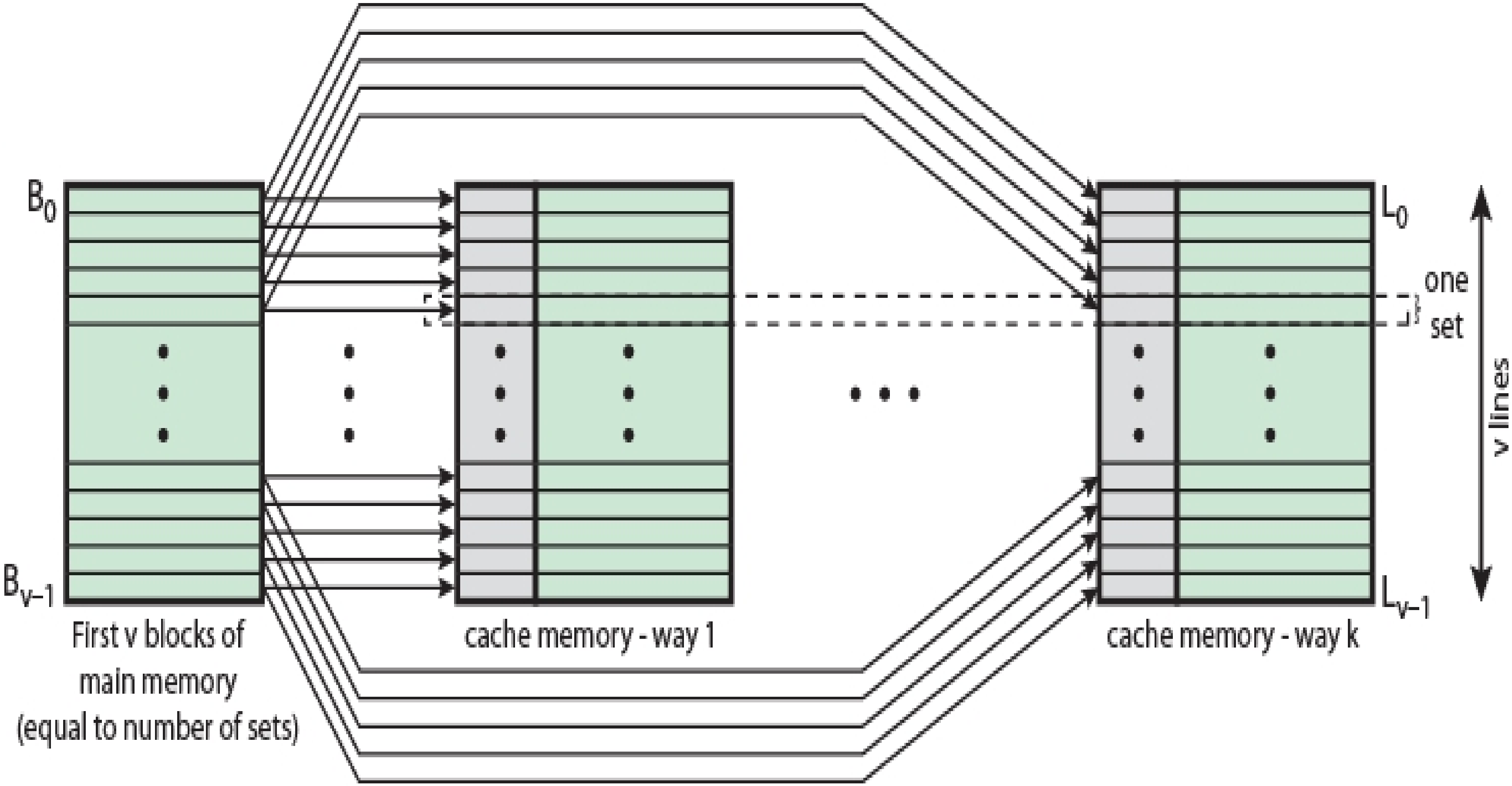
  v = number of sets

  k = number of lines in each set

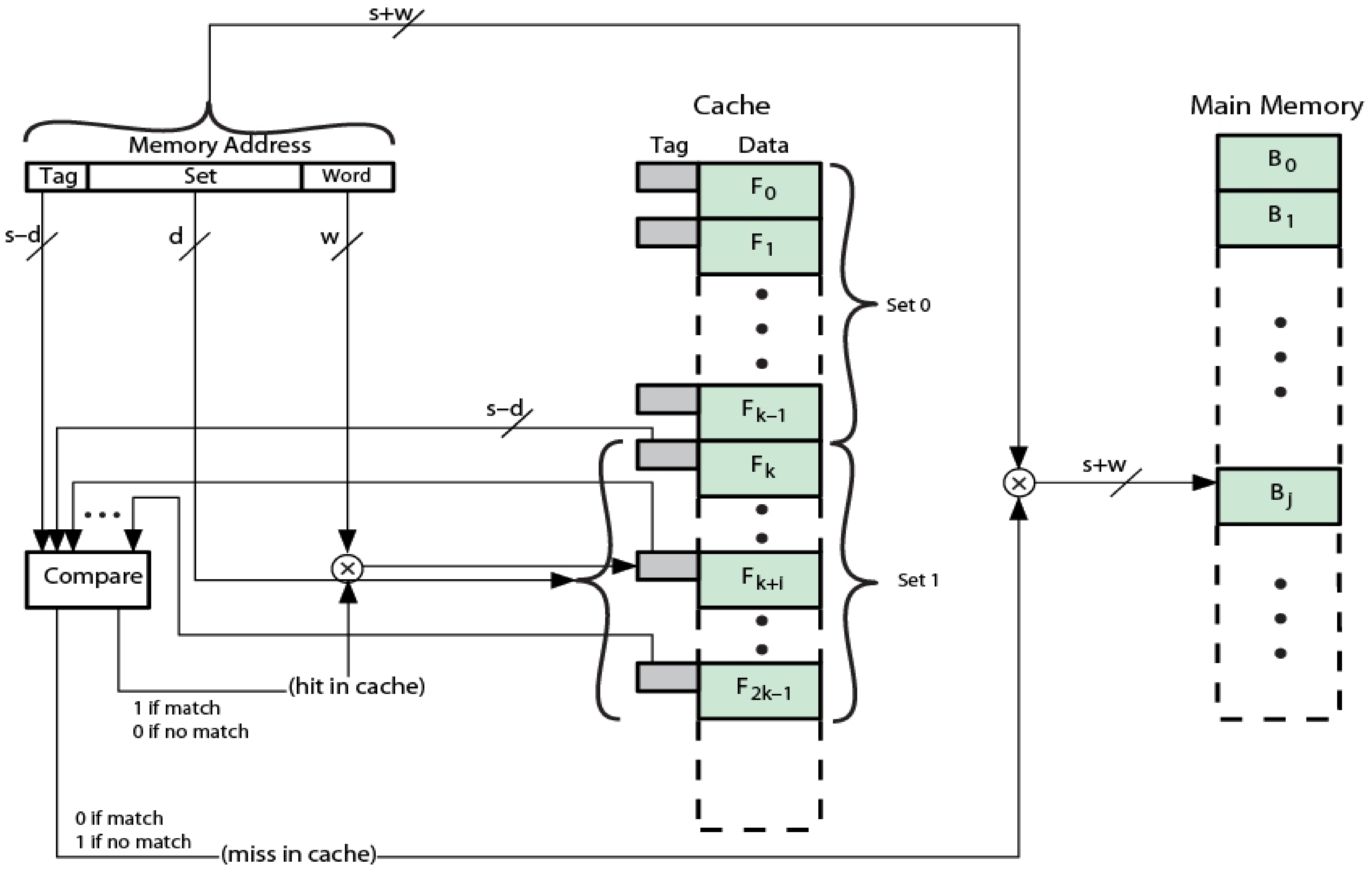
Mapping From Main Memory to Cache : v Associative



Mapping From Main Memory to Cache : k-way Associative



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
- 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)
- Size of tag = (s – d) bits

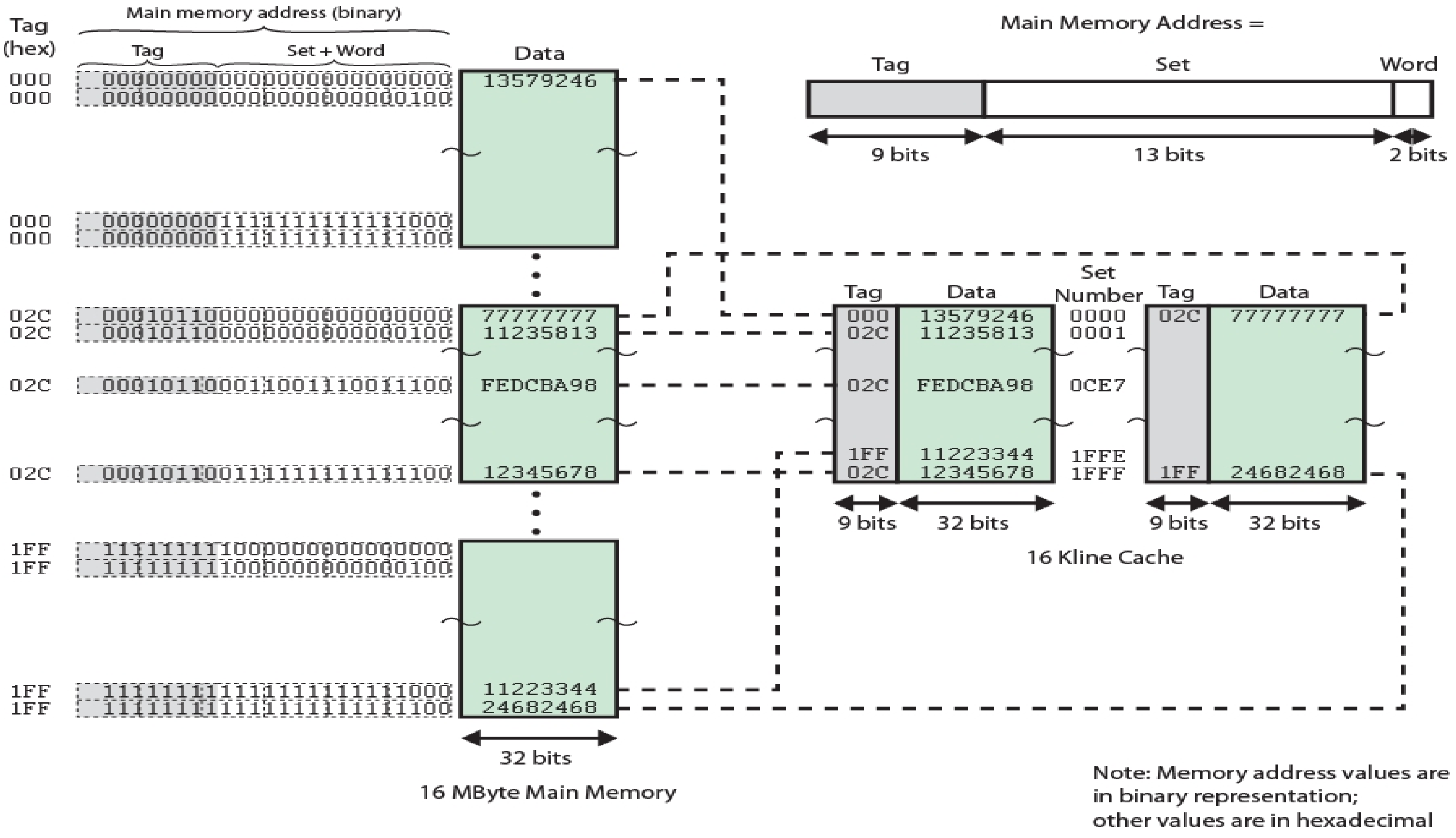
Set Associative Mapping Address Structure

Tag 9 bit	Set 13 bit	Word 2 bit

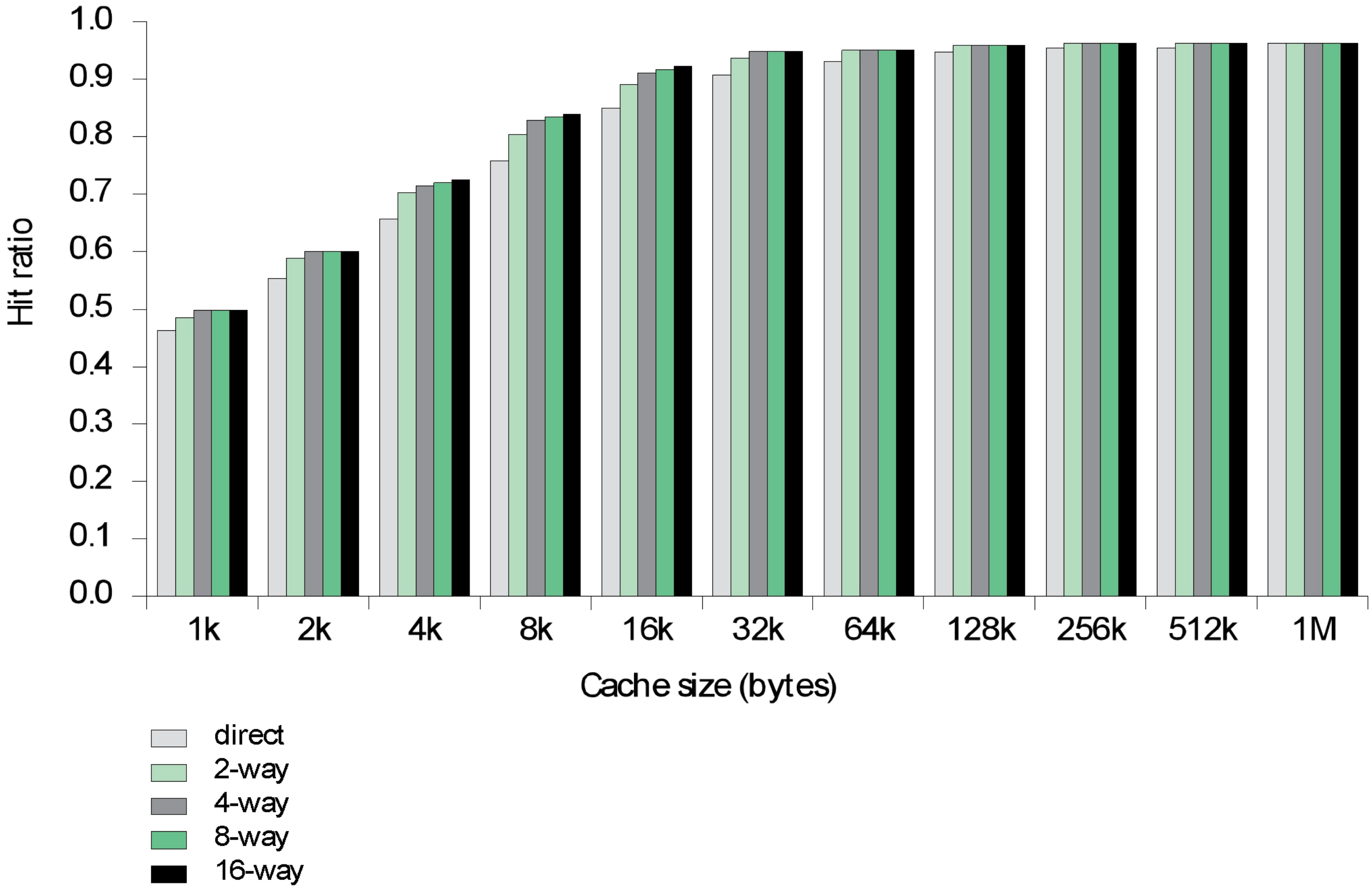
- Use set field to determine cache set to look in
- Compare tag field to see if we have a hit
- e.g

Address	Tag	Data	Set number
1FF 7FFC	1FF	12345678	1FFF
001 7FFC	001	11223344	1FFF

Two Way Set Associative Mapping Example


Tag (hex)
Main memory address (binary)
Tag
Set + Word
Data
000 000000000000000000000000 13579246
000 000000000000000000000100
000 000000001111111111111000
000 000000001111111111111100
02C 000101100000000000000000 77777777
02C 000101100000000000000100 11235813
02C 000101100011001110011100 FEDCBA98
02C 000101100111111111111100 12345678
1FF 111111111000000000000000
1FF 111111111000000000000100
1FF 111111111111111111111000 11223344
1FF 111111111111111111111100 24682468
32 bits
16 MByte Main Memory
Main Memory Address =
Tag
Set
Word
9 bits
13 bits
2 bits
Tag
Data
Set Number
Tag
Data
000 13579246 0000 02C 77777777
02C 11235813 0001
02C FEDCBA98 0CE7
1FF 11223344 1FFE
02C 12345678 1FFF 1FF 24682468
9 bits
32 bits
9 bits
32 bits
16 Kline Cache
Note: Memory address values are in binary representation; other values are in hexadecimal

Varying Associativity over Cache Size



Direct and Set Associative Cache Performance Differences

- Significant up to at least 64kB for 2-way
- Difference between 2-way and 4-way at 4kB much less than 4kB to 8kB in cache size
- Cache complexity increases with associativity
- Not justified against increasing cache to 8kB or 16kB
- Beyond about 32kB gives no improvement in performance

Thank You !