COMP212 Computer Organization and Systems

Tutorial/Lab #4

Problems about Cache Memory

4.2 A two-way set associative cache has lines of 16 bytes and a total size of 8 bytes. The 64-Mbyte main memory is byte-addressable. Show the format of main memory address.

Ans.

A two-way set => 2 lines / set in the cache;

The cache has lines of 16 bytes and a total size of 8K bytes
=> There are a total of 8K bytes/16 bytes = 512 lines ($2^3*2^{10}/2^4$) in the cache;

'2 lines / set in the cache' AND 'There are a total of 512 lines in the cache'
=> The cache consists of 512 lines/2 lines/set = 256 sets
=> 8 bits ($256=2^8$) are needed to identify the set number.

64-Mbyte main memory is byte-addressable
=> There are 64M units to be addressed
=> A 26-bit (2^6*2^{20}) address is needed;

'The cache has lines of 16 bytes' AND '64-Mbyte main memory'
=> Main memory consists of 64-Mbyte/16 bytes = 2^{22} blocks
=> The length of Tag + Set = 22
=> The length of Tag = 22-8 = 14;

The word field length is 26 – 22 = 4 bits

(Another way to calculate word field length:
The cache has lines of 16 bytes => The main memory block size is 16 bytes;
The main memory is byte-addressable => The word length (size) is 1 byte;
=> There are 16/1=16 words in one block;
=> 4 bits ($16=2^4$) are needed to identify the word number.)

Therefore, the format of main memory is

Tag	Set	Word
14	8	4

4.3 For the hexadecimal main memory address 111111, 666666, BBBBBB, show the following information, in hexadecimal format:

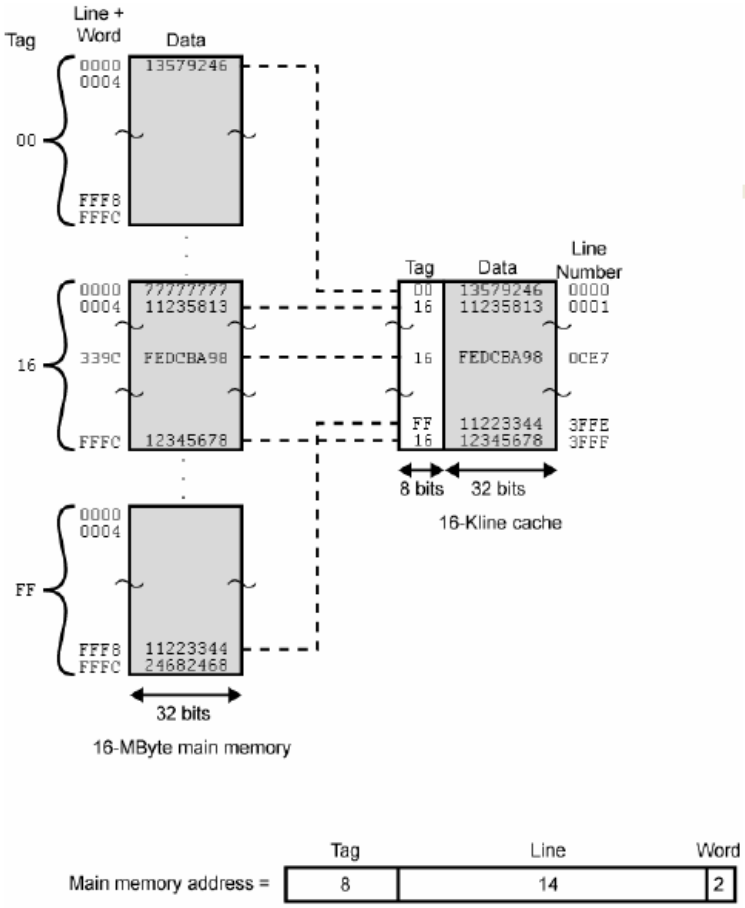
a) Tag, Line, and Word values for a direct-mapped cache, using the format of Figure 4.8 (slide 27).

Ans.

Address (in hexadecimal format)	111111	666666	BBBBBB
Address (in binary format)	0001 0001 0001 0001 0001 0001	0110 0110 0110 0110 0110 0110	1011 1011 1011 1011 1011 1011
Tag	0001 0001 (11_h)	0110 0110 (66_h)	1011 1011 (BB_h)
Line	0001 0001 0001 00 (444_h)	0110 0110 0110 01 (1999_h)	1011 1011 1011 10 ($2EEE_h$)
Word	01 (1_h)	10 (2_h)	11 (3_h)

4.4 List the following values:

a) For the direct cache example of Fig.4.8: address length, number of addressable units, block size, number of blocks in main memory, number of lines in cache, size of tag.



Direct Mapping Example

Ans.

Address length: 24 bits

Number of addressable unit: 2^{24}, i.e. 16M unit

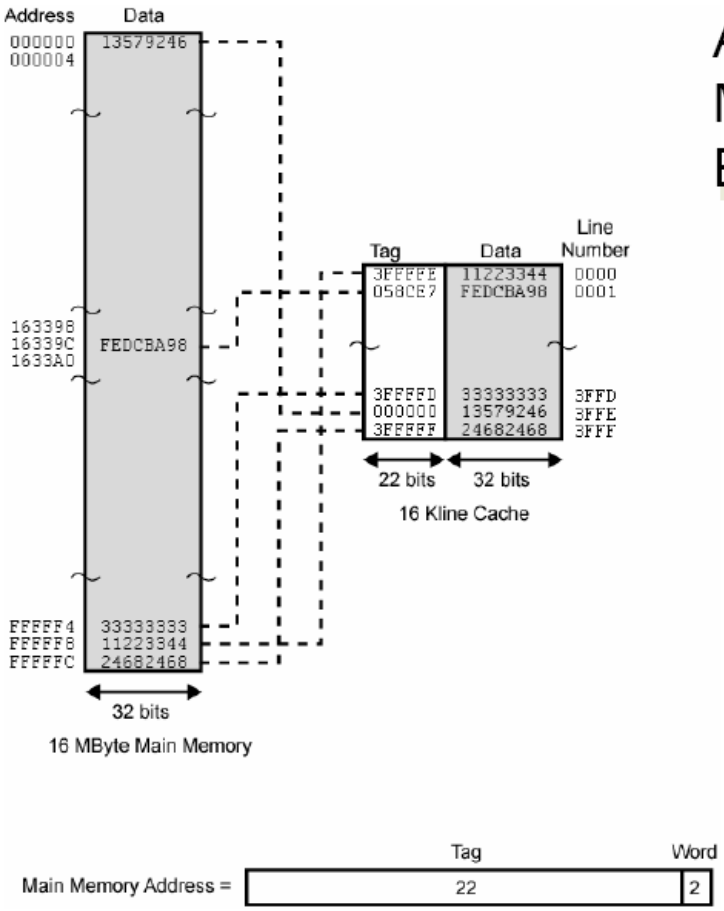
Block size: 2^2, i.e. 4 words

Number of blocks in main memory: $2^{24}/2^2$, i.e. 2^{22}

Number of lines in cache: 2^{14}, i.e. 16K lines

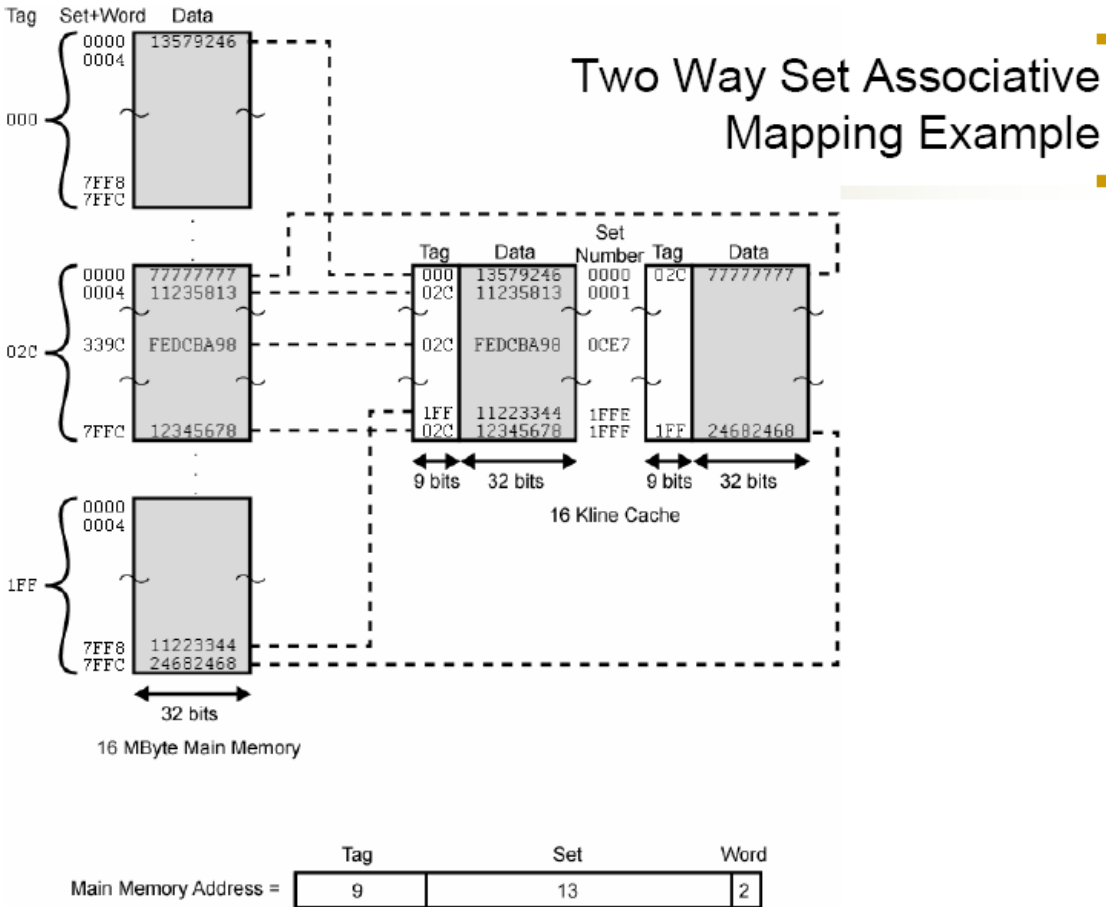
Size of tag: 8 bits

b) For the associative cache example of Fig.4.10: address length, number of addressable units, block size, number of blocks in main memory, number of lines in cache, size of tag.



Ans.
Address length: 24 bits
Number of addressable unit: 2^{24}, i.e. 16M unit
Block size: 2^2, i.e. 4 words
Number of blocks in main memory: $2^{24}/2^2$, i.e. 2^{22}
Number of lines in cache: 2^{14}, i.e. 16K lines
Size of tag: 22 bits

c) For the associative cache example of Fig.4.12: address length, number of addressable units, block size, number of blocks in main memory, number of lines in set, number of set, number of lines in cache, size of tag.



Ans.
Address length: 24 bits
Number of addressable unit: 2^{24}, i.e. 16M unit
Block size: 2^2, i.e. 4 words
Number of blocks in main memory: $2^{24}/2^2$, i.e. 2^{22}
Number of lines in set: 2
Number of set: 2^{13}
Number of lines in cache: 2^{14}, i.e. 16K lines
Size of tag: 9 bits

4.5 Consider a 32-bit microprocessor that has an on-chip 16-KByte four-way set-associative cache. Assume that the cache has a line size of four 32-bit words. Draw a block diagram of this cache showing its organization and how the different address fields are used to determine a cache hit/miss. Where in the cache is the word from memory location ABCDE8F8 mapped?

Ans.

A four-way set => 4 lines/set in the cache;

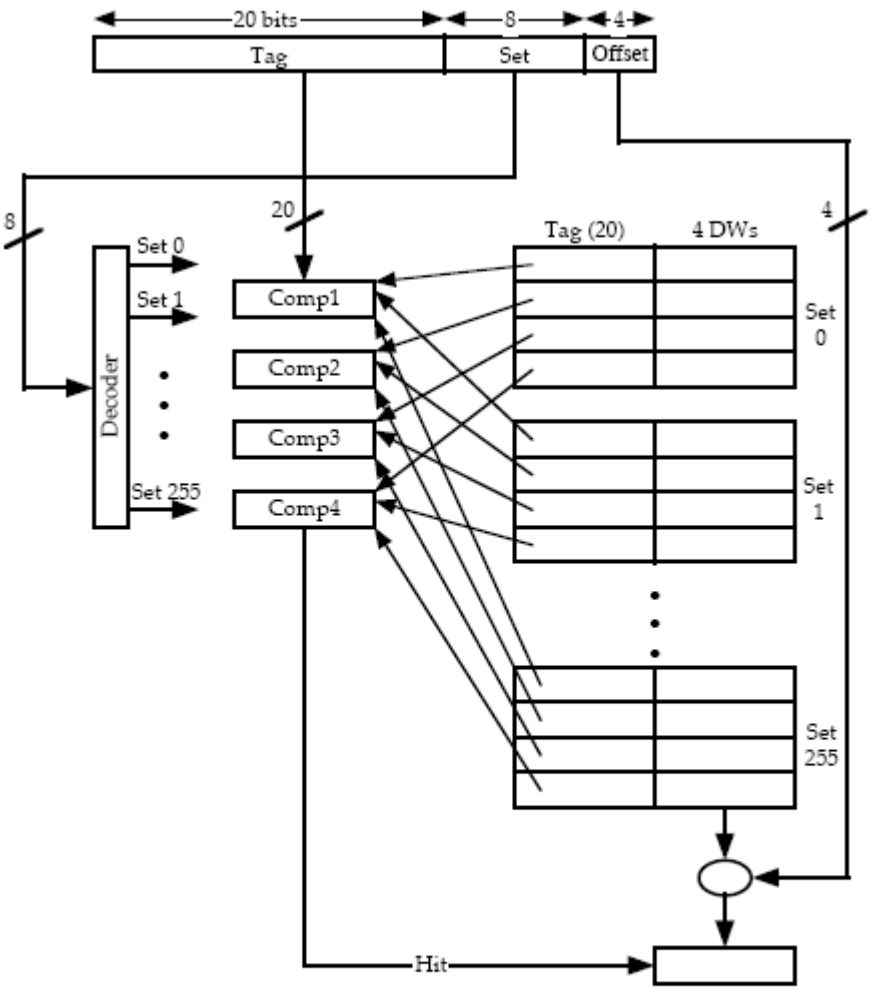
'16-KByte cache' AND 'the cache has a line size of four 32-bit words'
=> There are 16K*8/4*32 = $2^{14}*2^3/2^2*2^5 = 2^{10}$ = 1024 lines in the cache

‘4 lines/set’ AND ‘1024 lines in the cache’
=> There are 1024/4 = 256 = 2^8 sets in the cache
=> Set length is 8

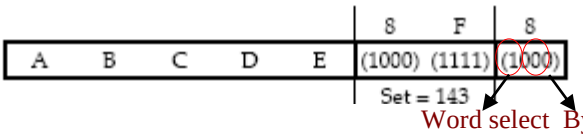
Line size is 4*32 bits = 16 bytes
=> 4 bits are needed to identify a byte;

ABCDE8F8 => memory address is 32 bits

So the tag length is 32-8-4=20



Example: doubleword from location ABCDE8F8 is mapped onto: set 143, any line, doubleword 2:



4.6 Given the following specifications for an external cache memory: four-way set-associative, line size of two 16-bit words; able to accommodate a total of 4K 32-bit words; used with 24-bit addresses. Design the cache structure.

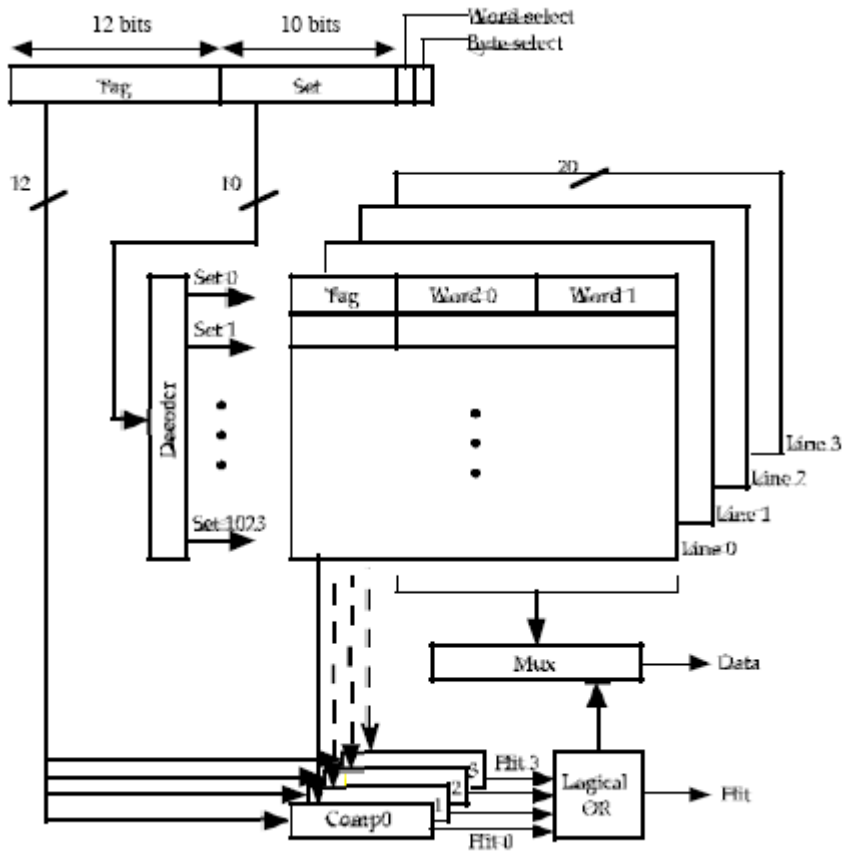
Ans.
A four-way set => 4 lines/set in the cache;

Line size is 2*16bits = 4 byte
=> 2 bits are needed to identify a byte

A total of 4K 32-bit words => cache size is $2^2*2^{10}*2^2 = 2^{14}$ bytes

So there are $2^{14}/4 = 2^{12}$ lines totally, which is equal to $2^{12}/4 = 2^{10}$ sets
=> 10 bits are needed to identify a set

So the Tag length is 24-2-10=12



4.8 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.

a) How is a 16-bit memory address divided into tag, line number, and byte number?
b) 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

c) 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?
d) How many total bytes of memory can be stored in the cache?
e) Why is the tag also stored in the cache?

Ans.

a) Address length: 16 bits
Word: 3 rightmost bits (as the block size if 8 bytes)
Line: 5 middle bits (as there are 32 lines)
Tag: 8 leftmost bits (as 16 – 3 – 5=8)

b)

0001 0001 **0001 1**011 → line 3
1100 0011 **0011 0**100 → line 6
1101 0000 **0001 1**101 → line 3
1010 1010 **1010 1**010 → line 21

c)

bytes with addresses 0001 1010 0001 1**000**
0001 1010 0001 1**001**
0001 1010 0001 1**010**
…
0001 1010 0001 1**111**
are stored in the cache.

d)

32 lines × 8 bytes = 256 bytes

e) It is because 2 items with two different memory addresses can be stored in the same place in the cache. The tag is used to distinguish between them.

4.11 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.
a) 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 block in main memory, number of lines in cache, size of tag.

Ans.

a) Address format: Tag=20 bits, Word=6 bits (64-byte line size), Line=32-20-6=6 bits
Number of addressable units: 2^{32}, i.e. 4Giga unit
Number of block in main memory: $2^{32}/2^{6}$, i.e. 2^{26}
Number of lines in cache: 2^{6}=64 lines
Size of tag: 20 bits

b) Assume an associative cache. Show the address format and determine the following parameters: number of addressable units, number of block in main memory, number of lines in cache, size of tag.

Ans.

b) Address format: Word=6 bits (64-byte line size), Tag=32-6=26 bits
Number of addressable units: 2^{32}, i.e. 4Giga unit
Number of block in main memory: $2^{32}/2^{6}$, i.e. 2^{26}
Number of lines in cache: undetermined
Size of tag: 26 bits

c) Assume a 4-way set associative cache with a tag field in the address of 9 bits. Show the address format and determine the following parameters: number of addressable units, number of block in main memory, number of lines in cache, size of tag.

Ans.

c) Address format: Tag= 9 bits, Word=6 bits (64-byte line size), Set=32-9-6=17 bits
Number of addressable units: 2^{32}, i.e. 4Giga unit
Number of block in main memory: $2^{32}/2^{6}$, i.e. 2^{26}
Number of lines in set: 4
Number of sets in cache: 2^{17}
Number of lines in cache: $4*2^{17}=2^{19}$
Size of tag: 9 bits

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

a) 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.
b) Give any two main memory addresses with different tags that map to the same cache slot for a direct-mapped cache.

Ans.

a) Word=4 bits,
Number of cache lines=64Kbytes/16bytes=4K lines=4096 lines. To address them, we need 12 bits. Thus, the tag will take up 20-12-4=4 bits
For MM address
F0010=1111 0000 0000 0001 0000 → Tag=1111 (F), Line=001, Word Offset=0
01234=0000 0001 0010 0011 0100 → Tag=0000 (0), Line=123, Word Offset=4
CABBE=1100 1010 1011 1011 1110 → Tag=1100 (C), Line=ABB, Word Offset=E

b) We need to pick any address where the slot is the same, but the tag (and optionally, the word offset) is different. For the line/slot=1111 1111 1111 (altogether 12 bits), we have
Address 1: Word offset=1111; Slot=1111 1111 1111; Tag=0000
Address=0FFFF
Address 2: Word offset=0001; Slot=1111 1111 1111; Tag=0011
Address=3FFF1

4.19 A computer has a cache, main memory, and a disk. If a referenced word is in the cache, 20 ns are required to access it. If it is in the main memory but not in the cache, 60 ns are required to load it into the cache, and then the reference is started again. If the word is not in the main memory, 12 ms are needed to load it from the disk to main memory, followed by 60 ns to copy it to the cache, and then the reference is started again. The cache hit ratio is 0.9 and the main memory hit ratio is 0.6. What is the average time needed to access a referenced word?

Ans.

There are three cases to consider:

Location of referenced word	Probability	Total time for access in ns
In cache	0.9	20
Not in cache, but in main memory	(0.1)(0.6) = 0.06	60 + 20 = 80
Not in cache or main memory	(0.1)(0.4) = 0.04	12ms + 60 + 20 = 12,000,080

So the average access time would be:

Avg = (0.9)(20) + (0.06)(80) + (0.04)(12000080) = 480026 ns