

Computer Science & Information Technology

Computer Organization & Architecture

Cache Organization

DPP 02

- Q1** Consider a 512KB direct mapped cache with block size of 32 bytes. The main memory address is of 34-bits. The size of index and tag in bits are?
 (A) 15, 14 (B) 14, 15
 (C) 19, 14 (D) 14, 19
- Q2** Consider a direct mapped cache of size 256MB. Cache controller maintains 8-bits tag for each block in cache. The maximum size of main memory (byte addressable) supported in the system is _____GB?
- Q3** The size of memory required at cache controller to store metadata is 2KBytes. The metadata includes tag bits, 1 modified bit and 1 valid bit. The cache contains 1K blocks of 16bytes each and organized as direct mapped. The size of main memory is _____ Mbytes?
- Q4** Consider a direct mapped cache of size 256KB. The CPU generates x- bit addresses. The number of tag bits in main memory address are 14 bits then value of x is _____?
- Q5** Assume a computer has 32-bit addresses. Each block stores 64 bytes. A direct-mapped cache has 512 blocks. Match the block (line) of the cache (in decimal) we look for each of the given hexadecimal addresses in the table?
 (A) 1A2BC012: 256, FFFF00FF: 3, 12345678: 345, C109D532: 340
 (B) 1A2BC012: 512, FFFF00FF: 7, 12345678: 243, C109D532: 320
 (C) 1A2BC012: 128, FFFF00FF: 5, 12345678: 345, C109D532: 420
 (D) 1A2BC012: 255, FFFF00FF: 1, 12345678: 247, C109D532: 240
- Q6** Consider a cache with 2^{13} blocks of size 32Bytes each. The CPU generates addresses of 32-bits. The cache controller stores 1 valid bit, 1 modified bit and tag-bits for each metadata entry. The cache controller has a maximum memory of 18Kbytes to store the metadata. The cache is organized as k-way set associative. Maximum value of k to utilize the cache controller memory in optimized manner is _____?
- Q7** Consider a direct mapped write back data cache of size 2KB with the block size of 128 bytes. The cache is considered to be empty initially. The byte addressable main memory has size 1Mbytes. Further consider that there is an array A[35][20] with each element occupies 4 bytes. The base address of array is $(1A300)_{16}$. The array is accessed 3 times. And between the accesses, there is no any data cache changes happen. Hit ratio (correct upto 1 decimal place) of cache for this array access is _____%?
- Q8** Consider a direct mapped cache of size 32 Kbytes. The cache uses 64 bytes block. Consider main memory address is of 20 bits. For the main memory address $(94347)_{10}$, the tag is $(\quad)_{10}$?
- Q9** Consider a 4-way set associative cache of size 256 Kbytes. The cache uses 32 bytes block.


[Android App](#)
[iOS App](#)
[PW Website](#)

Consider main memory address is of 20 bits. For the main memory address $(86147)_{10}$, the tag is $(\quad)_{10}$?

- Q10** Consider a fully associative cache of size 16Kbytes. The cache uses 128 bytes block. Consider main memory address is of 20 bits. For the main memory address $(5182)_{10}$, the tag is $(\quad)_{10}$?
- Q11** A 2-way set associative cache with LRU cache replacement contains 8 blocks. The CPU requests for main memory blocks in following sequence: 8, 12, 0, 1, 5, 8, 1, 12, 8, 0, 1, 3, 6, 7, 3, 11, 2, 3, 6, 2, 8, 7, 11
Number of cold, capacity and conflict misses respectively are?
(A) 10, 3, 2 (B) 10, 2, 3

(C) 9, 3, 2

(D) 9, 2, 3

- Q12** Consider a direct mapped cache of size 2MBytes. The CPU generates x-bit addresses. The number of tag bits in main memory address are 14 bits then value of x is ____?
- Q13** Consider a set-associative cache of size 16KB ($1\text{KB}=2^{10}$ bytes) with cache block size of 32 bytes. Assume that the cache is byte-addressable and a 32-bit address is used for accessing the cache. If the width of the tag field is 20 bits, the associativity of the cache is _____?
- Q14** Assume a computer has 32 bits addresses. Each block stores 128 bytes. A 4-way set associative cache has size 512Kbytes. The set of the cache would we look for the address $(AC1016AF)_{16}$ is $(\quad)_{10}$?



Answer Key

Q1 (B)
Q2 64~64
Q3 256~256
Q4 32~32
Q5 (A)
Q6 4~4
Q7 97.8~97.8

Q8 2~2
Q9 1~1
Q10 40~40
Q11 (B)
Q12 35~35
Q13 4~4
Q14 45~45



[Android App](#) | [iOS App](#) | [PW Website](#)

Hints & Solutions

Q1 Text Solution:

The 34-bits main memory address is divided into 3 parts like this:

Tag	Block	Byte
15	14	5

Block size = 32bytes = 2^5 bytes, hence byte number is of 5 bits

Number of blocks in cache = $512\text{KB} / 32\text{B} = 16\text{K} = 2^{14}$, hence number of bits for block number = 14 bits. This is only index for direct mapped cache.

Tag bits = $34 - (14+5) = 15$ bits

Q2 Text Solution:

The main memory address is divided into 3 parts as below:

Tag	Block	Byte
8		

We know that bits for block and byte combined = $\log(\text{cache size}) = \log 256\text{M} = 28$ bits

Hence entire main memory address = $8 + 28 = 36$ bits

Main memory size = $2^{36} = 64\text{GB}$

Q3 Text Solution:

Number of blocks in cache = $1\text{K} = 2^{10}$, hence number of bits for block = 10 bits

Block size = 16 bytes = 2^4 , hence number of bits for byte = 4 bits

The main memory address is divided into 3 parts as below:

Tag	Block	Byte
	10	4

Tag bits we will calculate from tag directory size.

Tag directory size = Number of blocks in cache * (Tag bits + extra bits)

$2\text{Kbytes} = 1\text{k} * (\text{Tag} + 1 + 1)$ bits

$2\text{K} * 8 \text{ bits} = 1\text{K} * (\text{Tag} + 1 + 1)$ bits

$16 = \text{Tag} + 2$

Tag = 14 bits

Now let's calculate the main memory address = $14 + 10 + 4 = 28$ bits

Main memory size = $2^{28} = 256\text{MB}$

Q4 Text Solution:

The main memory address is divided into 3 parts as below:

Tag	Block	Byte
14		

We know that bits for block and byte combined = $\log(\text{cache size}) = \log 256\text{K} = 18$ bits

Hence entire main memory address = $14 + 18 = 32$ bits

Q5 Text Solution:

Number of blocks in cache = $512 = 2^9$, hence number of bits for block = 9 bits

Block size = 64 bytes = 2^6 , hence number of bits for byte = 6 bits

The 32-bits main memory address is divided into 3 parts as below:

Tag	Block	Byte
17	9	6

For address $(1A2BC012)_{16} = 0001\ 1010\ 0010\ 1011\ 1100\ 0000\ 0001\ 0010$

0001 1010 0010 1011	100 0000	01 0010
1	00	
17	9	6

Block number = $(100000000)_2 = (256)_{10}$



Android App

iOS App

PW Website

For address $(FFFF00FF)_{16} = 1111\ 1111\ 1111\ 1111$
 $0000\ 0000\ 1111\ 1111$

1111 1111 1111 1111 0	000 0000 11	11 1111
17	9	6

Block number = $(000000011)_2 = (3)_{10}$

For address $(12345678)_{16} = 0001\ 0010\ 0011\ 0100$
 $0101\ 0110\ 0111\ 1000$

0001 0010 0011 0100 0	101 0110 01	11 1000
17	9	6

Block number = $(101011001)_2 = (345)_{10}$

For address $(C109D532)_{16} = 1100\ 0001\ 0000$
 $1001\ 1101\ 0101\ 0011\ 0010$

1100 0001 0000 1001 1	101 0101 00	11 0010
17	9	6

Block number = $(101010100)_2 = (340)_{10}$

Q6 Text Solution:

Block size = 32 bytes = 2^5 , hence number of bits
 for byte = 5 bits

Tag bits we will calculate from tag directory size.

Tag directory size = Number of blocks in cache *
 (Tag bits + extra bits)

18Kbytes = $2^{13} * (\text{Tag} + 1 + 1)$ bits

$18K * 8 \text{ bits} = 2^{13} * (\text{Tag} + 1 + 1)$ bits

$18 = \text{tag} + 2$

Tag = 16 bits

The 32-bits main memory address is divided into
 3 parts as below:

Tag	set	Byte
16		5

For set offset number of bits needed = 32 –
 (16+5) = 11 bits

Hence number of sets = $2^{11} = 2^{13}/k$, hence $k = 4$

Q7 Text Solution:

Number of blocks in cache = 2KB/128B = 16

Array size = $35 * 20 = 700$ element = $700 * 4$ bytes
 = 2800 bytes

To store entire array in main memory, the number
 of blocks needed = $\text{ceil}(2800 \text{ bytes} / 128 \text{ bytes}) = 22$

For 1st access of array, for each block there will be
 1 miss hence for accessing 22 blocks of array 22
 miss will be experienced, but there are total 700
 elements in array hence for remaining $700 - 22 = 678$
 elements there will be hit in cache.

Now for second reference only $22 - 16 = 6$ blocks
 will experience miss 2 times, hence total miss for
 2nd reference = 12 and hits = $700 - 12 = 688$

For 3rd reference also number of hits and miss
 will be same as 2nd reference, which are 12 and
 688 respectively.

Total number of hits = $678 + 688 + 688 = 2054$

Hit ratio = $2054 / 2100 = 0.978 = 97.8\%$

Q8 Text Solution:

Number of blocks in cache = 32KB / 64B = 512

Main memory block number = $\text{floor}(94347/64) = 1474$

Tag in main memory address = $1474 / 512 = 2$

Q9 Text Solution:

Number of blocks in cache = 256KB / 32 B = 8k =
 8192

Number of sets in cache = $8192 / 4 = 2048$

Main memory block number = $86147/32 = 2692$

Tag = Main memory block number / number of
 sets in cache

= $2692 / 2048$

= 1

Q10 Text Solution:



Android App

iOS App

PW Website

Main memory block number = $5182 / 128 = 40$
 In fully associative mapping, the main memory block number is tag only. Hence tag is 40.

Q11 Text Solution:

Number of sets in cache = $8/2 = 4$

In set associative mapping, cache memory set number = main memory block number % 4

For the references of the given blocks, the miss and replacements are as follows:

Main memory block number	Cache set number	Type of miss or hit	Replaced block
8	$8\%4 = 0$	Cold Miss	None
12	$12\%4 = 0$	Cold Miss	None
0	$0\%4 = 0$	Cold Miss	Replace 8
1	$1\%4 = 1$	Cold Miss	None
5	$5\%4 = 1$	Cold Miss	None
8	$8\%4 = 0$	Conflict Miss	Replace 12
1	$1\%4 = 1$	Hit	None
12	$12\%4 = 0$	Conflict Miss	Replace 0
8	$8\%4 = 0$	Hit	None
0	$0\%4 = 0$	Conflict Miss	Replace 12
1	$1\%4 = 1$	Hit	None
3	$3\%4 = 3$	Cold Miss	None
6	$6\%4 = 2$	Cold Miss	None
7	$7\%4 = 3$	Cold Miss	None
3	$3\%4 = 3$	Hit	None
11	$11\%4 = 3$	Cold Miss	Replace 7
2	$2\%4 = 2$	Cold Miss	None
3	$3\%4 = 3$	Hit	None
6	$6\%4 = 2$	Hit	None
2	$2\%4 = 2$	Hit	None

8	$8\%4 = 0$	Hit	None
7	$7\%4 = 3$	Capacity Miss	Replace 11
11	$11\%4 = 3$	Capacity Miss	Replace 3

Number of cold miss = 10

Number of capacity miss = 2

Number of conflict miss = 3

Q12 Text Solution:

The main memory address is divided into 3 parts as below:

Tag	Block	Byte
14		

We know that bits for block and byte combined = $\log(\text{cache size}) = \log 2M = 21$ bits

Hence entire main memory address = $14 + 21 = 35$ bits, hence $x = 35$

Q13 Text Solution:

The main memory address is divided into 3 parts as below:

Tag	Set	Byte
20		

We know that bits for set and byte combined = $\log(\text{cache size}) - \log(\text{associativity})$

Hence the entire main memory address = 32 bits
 $32 \text{ bits} = 20 + \log(16K \text{ size}) - \log(\text{associativity})$

$$32 = 20 + 14 - \log(\text{associativity})$$

$$\log(\text{associativity}) = 2$$

$$\text{associativity} = 4$$

Q14 Text Solution:

Number of blocks in cache = $512KB/128B = 2^{12}$,

Number of sets in cache = $2^{12} / 4 = 2^{10}$, hence number of bits for set offset = 10 bits

Block size = 128 bytes = 2^7 , hence number of bits for byte = 7 bits



The 32-bits main memory address is divided into 3 parts as below:

Tag	Set	Byte
15	10	7

For address $(AC1016AF)_{16} = 1010\ 1100\ 0001\ 0000\ 0001\ 0110\ 1010\ 1111$

1010 1100 0001 000	0 0001 0110 1	01 0010
15	10	7

Set number = $(0000101101)_2 = (45)_{10}$

[Android App](#)| [iOS App](#)| [PW Website](#)