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 32 Bytes 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 ____%?

Answer Key

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

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

Number of blocks in cache = 512KB / 32 B = 16K = 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 (cache size) = log 256M = 28 bits

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

Main memory size = 2^{36} = 64GB

Q3 Text Solution:

Number of blocks in cache = 1K = 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)

2Kbytes = 1k × (Tag + 1 + 1) bits

2K × 8 bits = 1K × (Tag + 1 + 1) bits

16 = Tag + 2

Tag = 14 bits

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

Main memory size = 2^{28} = 256MB

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 (cache size) = log 256K = 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 1	100 0000 00	01 0010
17	9	6

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

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} × (Tag + 1 + 1) bits

18K × 8 bits = 2^{13} × (Tag + 1 + 1) bits

18 = 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 = ceil (2800 bytes /128 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%