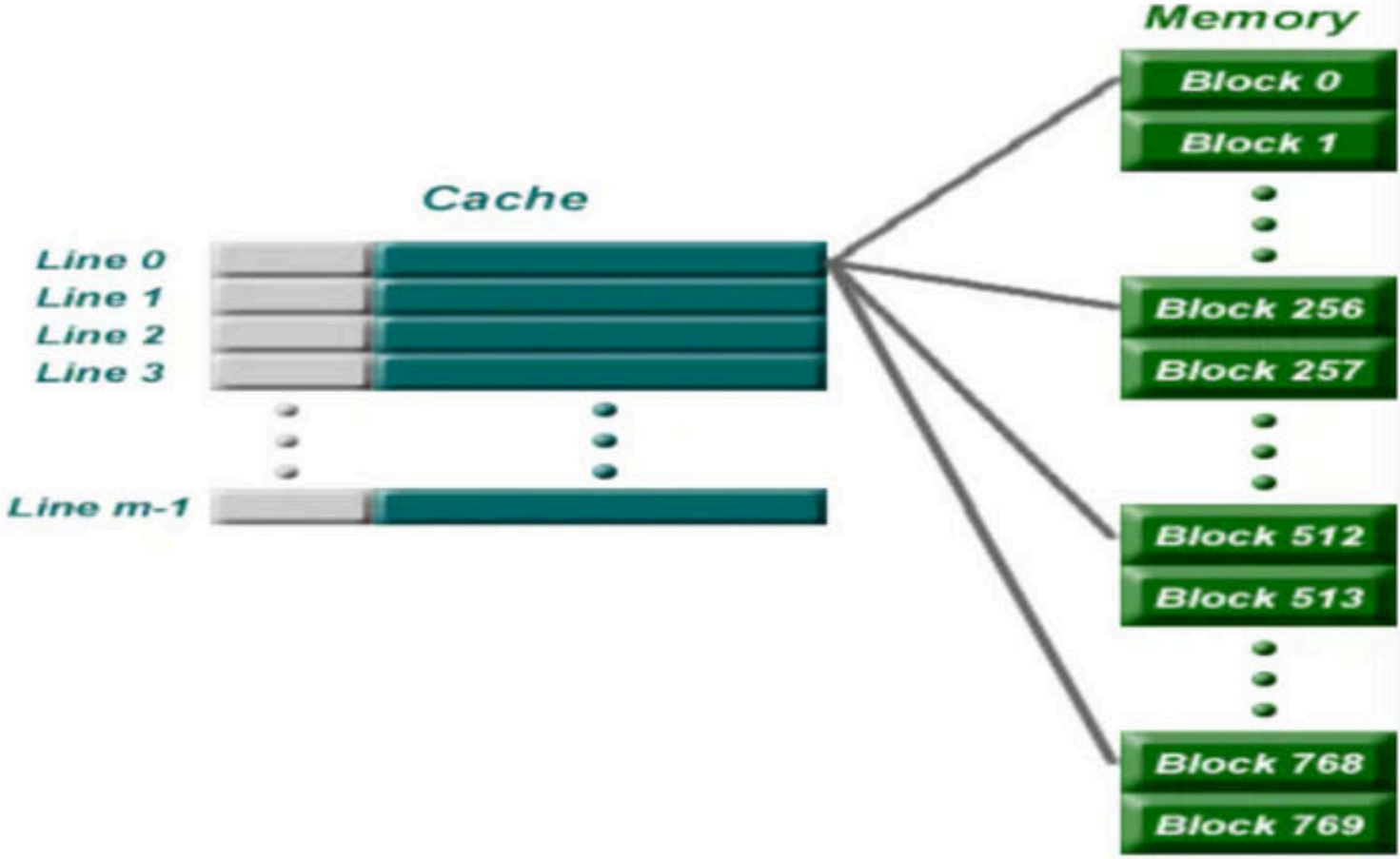
Cache Management - Part I

Complete Course on Computer Architecture for GATE

Sanchit Jain • Lesson 5 • Nov 28, 2021

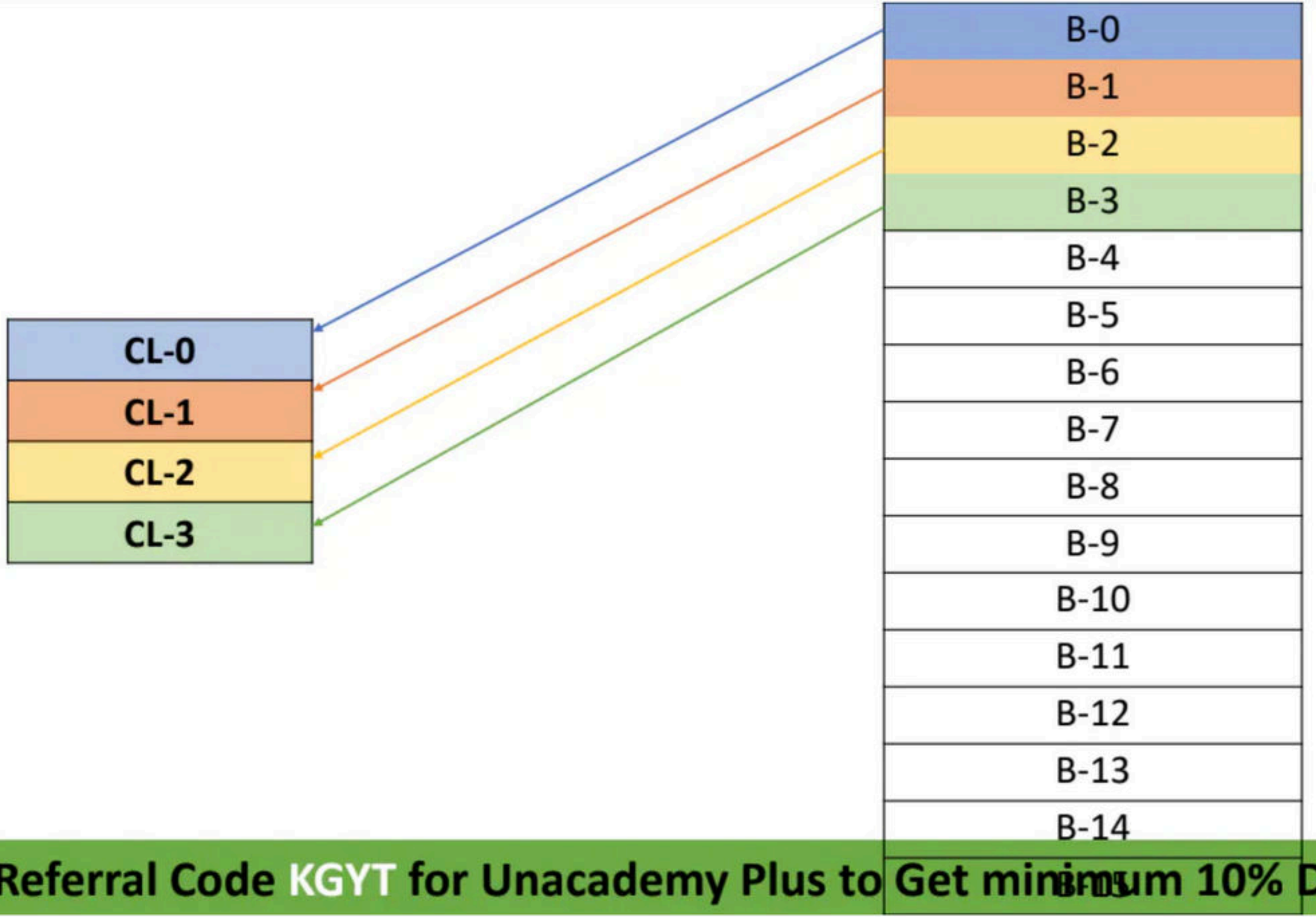
Direct Mapping

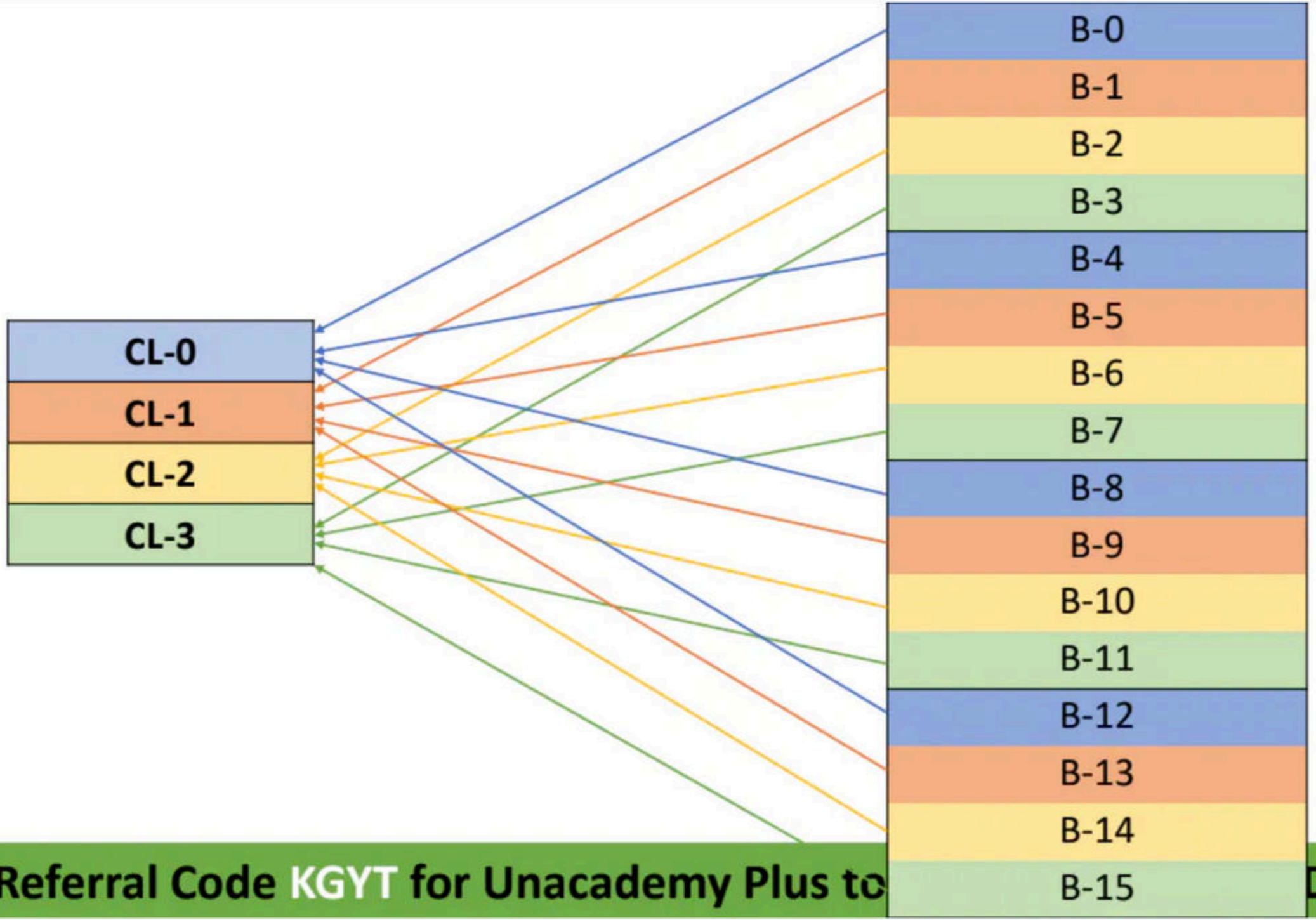
- In direct mapping scheme the main memory blocks are directly mapped onto a particular cache memory line.
- It is also known as many to one mapping.



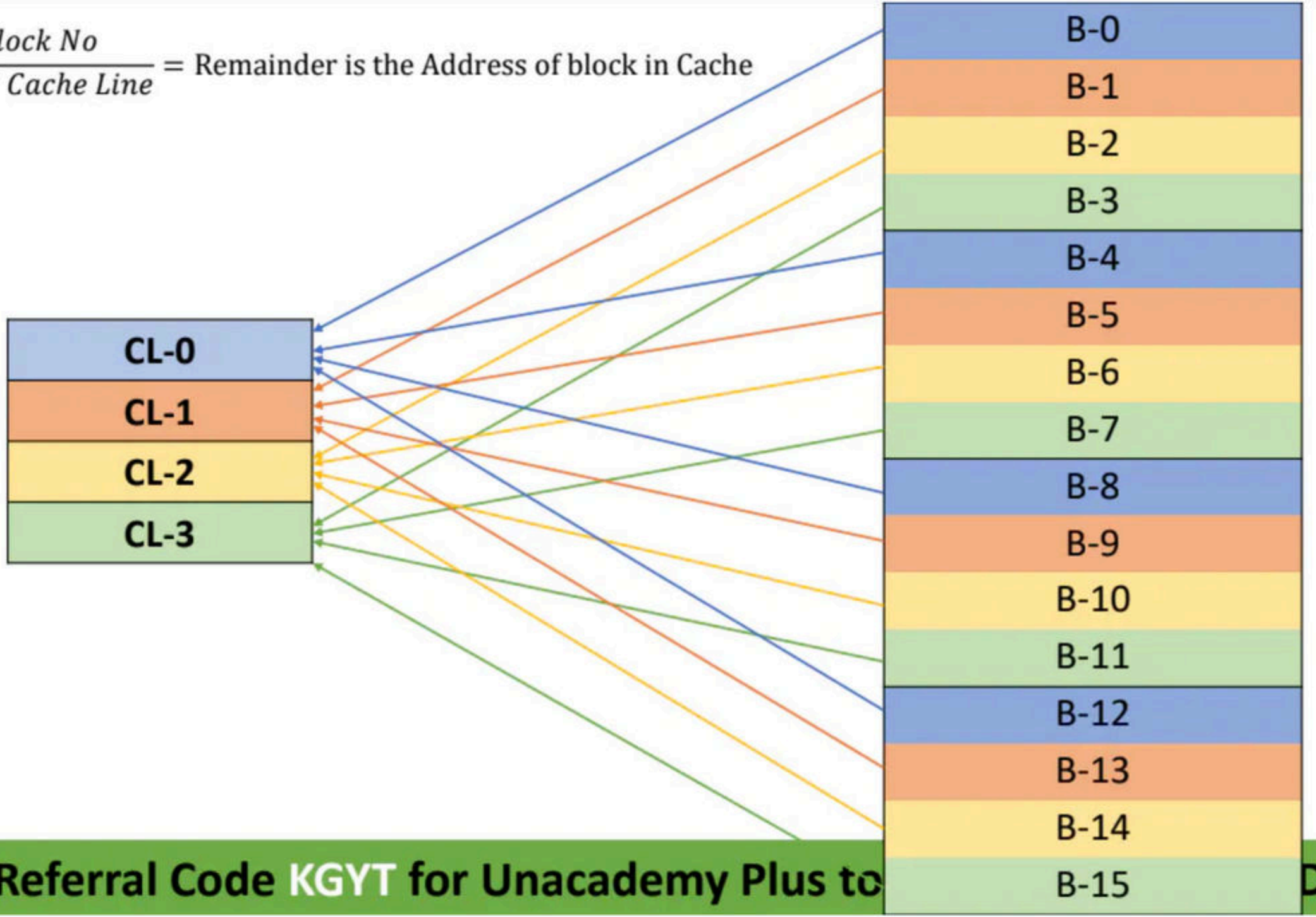
CL-0
CL-1
CL-2
CL-3

B-0
B-1
B-2
B-3
B-4
B-5
B-6
B-7
B-8
B-9
B-10
B-11
B-12
B-13
B-14
B-15





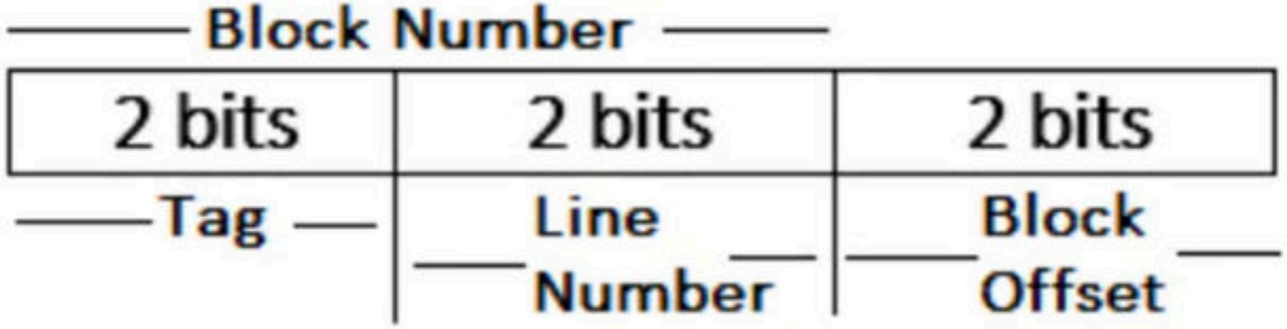
$$\frac{Block\ No}{No\ of\ Cache\ Line} = \text{Remainder is the Address of block in Cache}$$

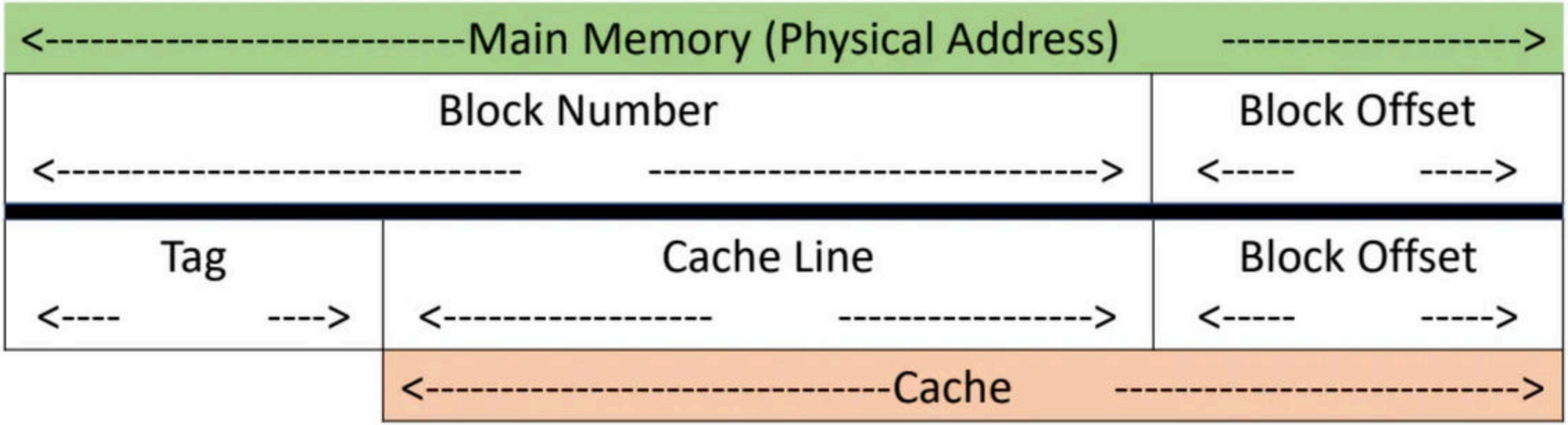


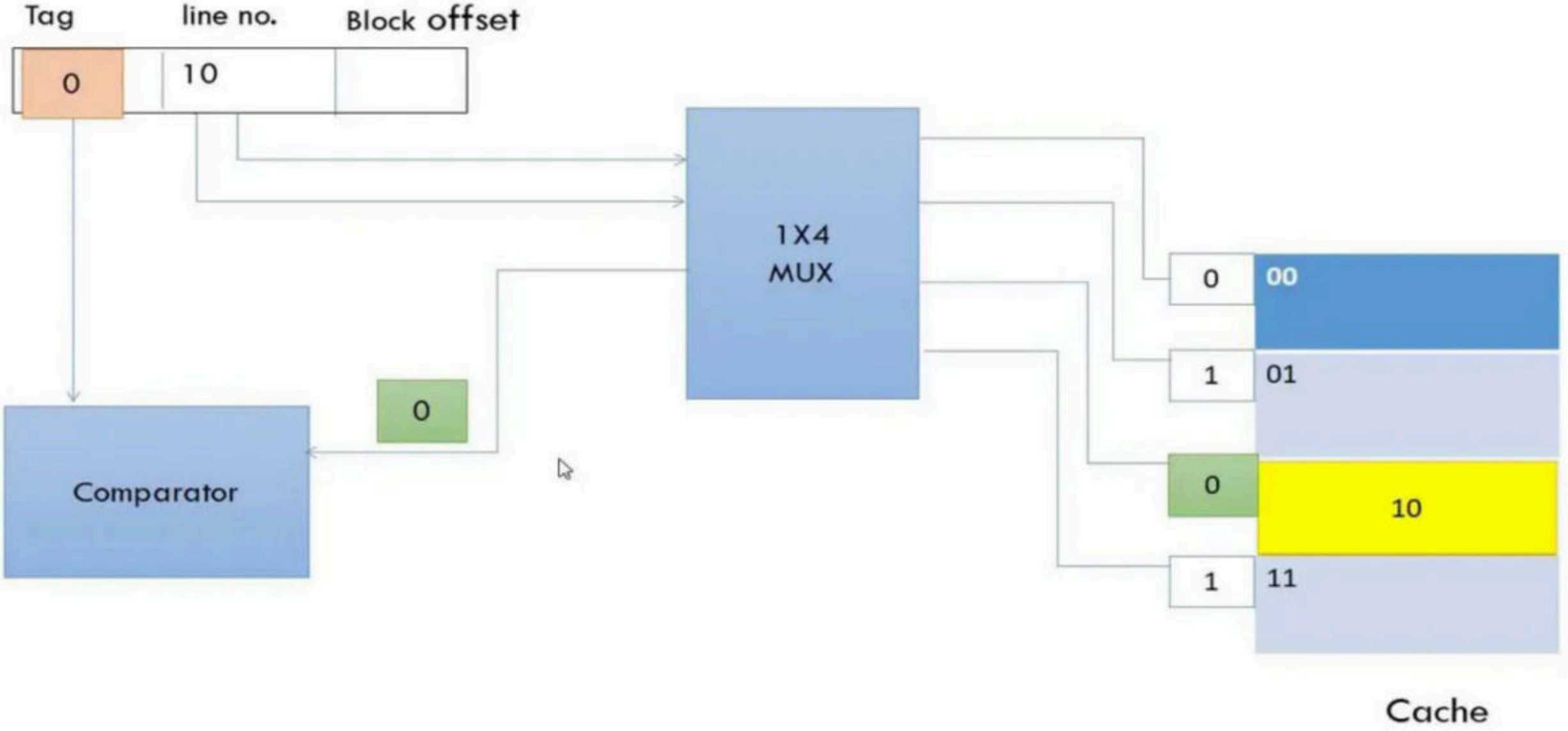
Cache								
CL-0	B-0	W-0	B-4	W-16	B-8	W-32	B-12	W-48
		W-1		W-17		W-33		W-49
		W-2		W-18		W-34		W-50
		W-3		W-19		W-35		W-51
CL-1	B-1	W-4	B-5	W-20	B-9	W-36	B-13	W-52
		W-5		W-21		W-37		W-53
		W-6		W-22		W-38		W-54
		W-7		W-23		W-39		W-55
CL-2	B-2	W-8	B-6	W-24	B-10	W-40	B-14	W-56
		W-9		W-25		W-41		W-57
		W-10		W-26		W-42		W-58
		W-11		W-27		W-43		W-59
CL-3	B-3	W-12	B-7	W-28	B-11	W-44	B-15	W-60
		W-13		W-29		W-45		W-61
		W-14		W-30		W-46		W-62
		W-15		W-31		W-47		W-63

Cache Memory	
CL-0	B-0 / B-4 / B-8 / B-12
CL-1	B-1 / B-5 / B-9 / B-13
CL-2	B-2 / B-6 / B-10 / B-14
CL-3	B-3 / B-7 / B-11 / B-15

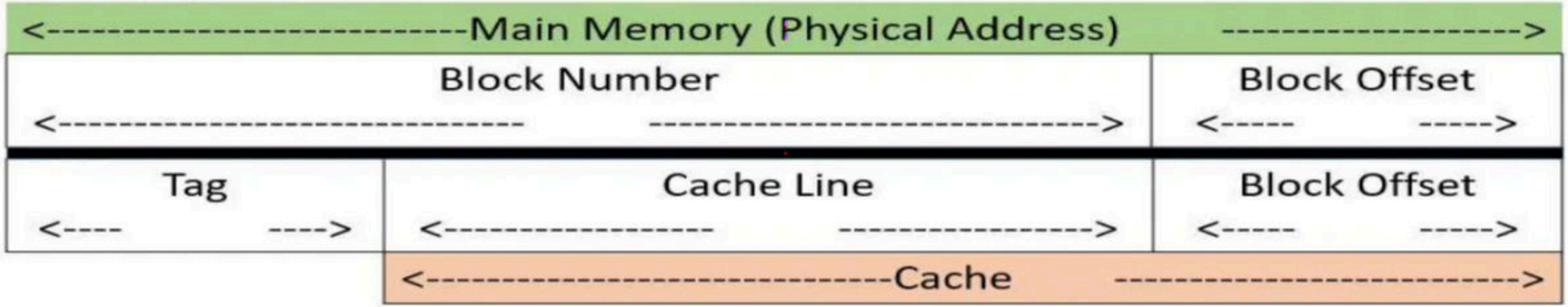
Cache Memory		
CL-0	TAG	W-
		W-
		W-
		W-
CL-1	TAG	W-
		W
		W-
		W-
CL-2	TAG	W-
		W-
		W-
		W-
CL-3	TAG	W-
		W-
		W-
		W-







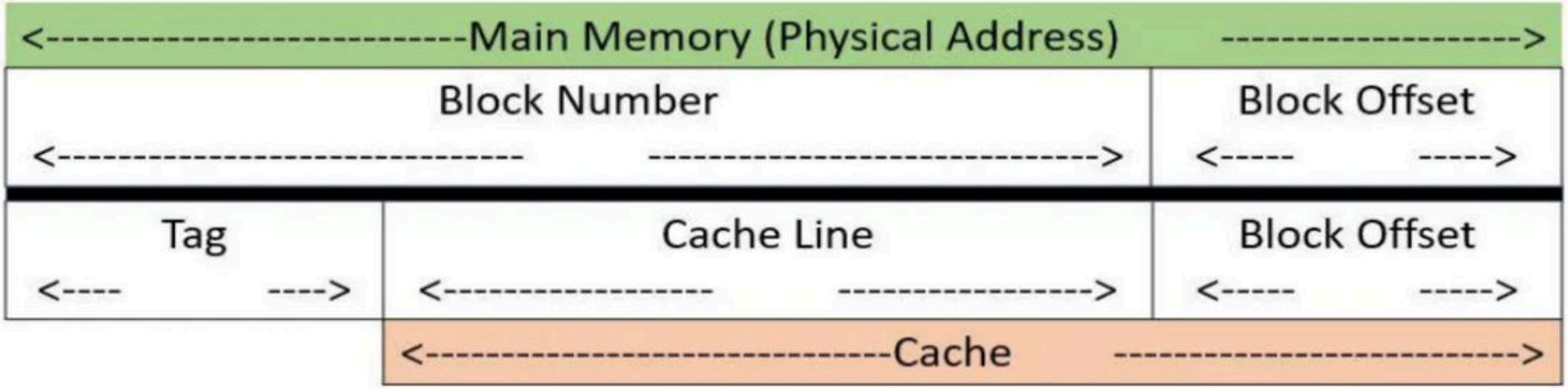
MM Size	Cache Size	Block Size	No of bits in Tag	Tag Directory Size
16 GB	32 MB	4 KB		
128 MB	256 KB	512 B		
32 GB	128 MB	1 KB		
256 MB	16 KB	1 KB		
4 GB	8 MB	2 KB		
512 KB	2 KB	128 B		



MM Size	Cache Size	Block Size	No of bits in Tag	Tag Directory Size
128 KB	16 KB	256 B	3	$3 * 2^6$
32 GB	32 KB	1 KB	20	$20 * 2^5$
2^{26}B	512 KB	1 KB	7	$7*2^9$
16 GB	2^{24} B	4 KB	10	$10*2^{12}$
64 MB	2^{16} B	?	10	?
2^{26} B	512 KB	?	7	?

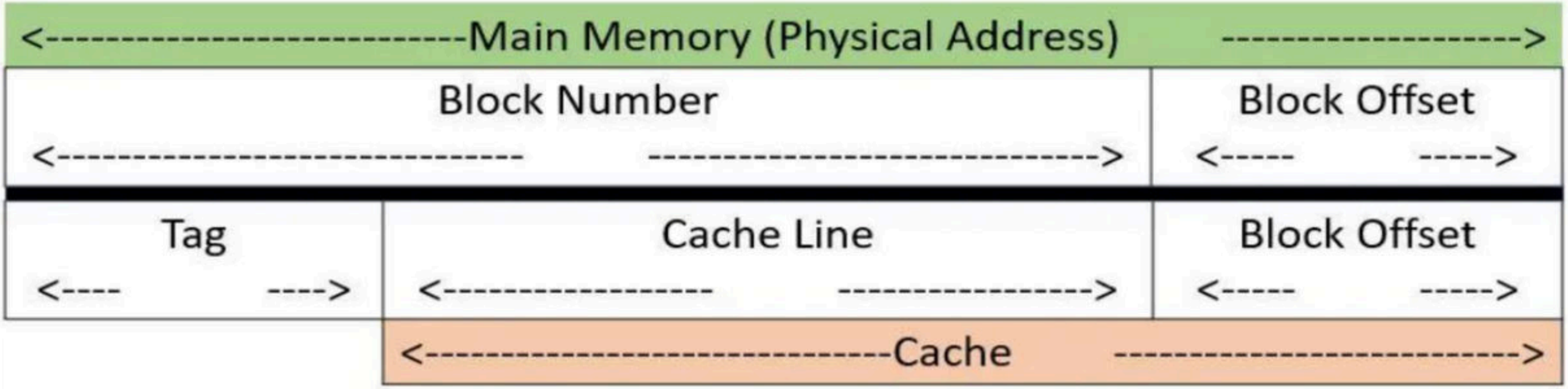
Q.41 Consider a computer system with a byte-addressable primary memory of size 2^{32} bytes. Assume the computer system has a direct-mapped cache of size 32 KB (1 KB = 2^{10} bytes), and each cache block is of size 64 bytes.
The size of the tag field is _____ bits.

(GATE-2021)



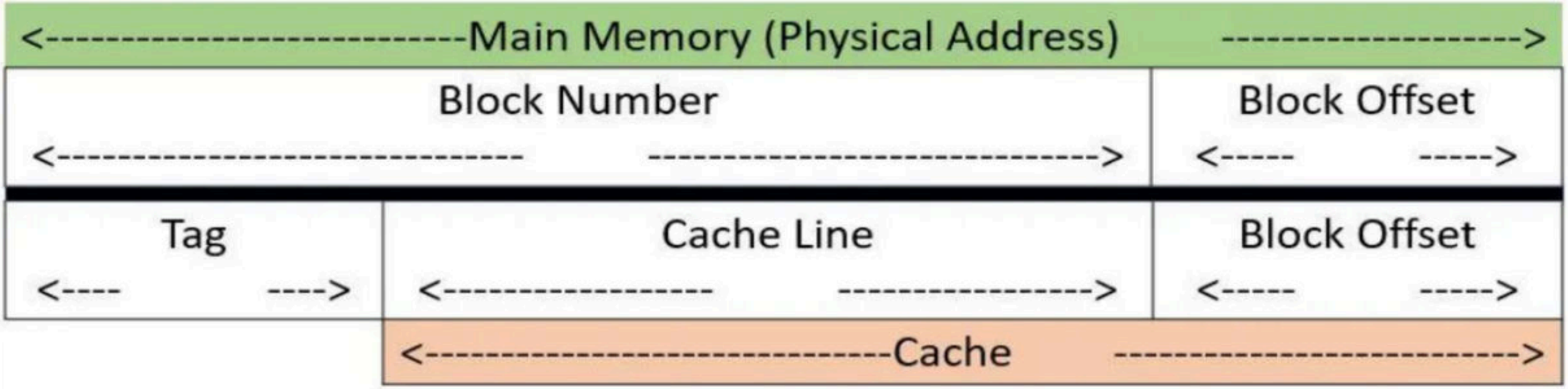
Q Consider a machine with byte addressable memory of 2^{32} bytes divided into blocks of size 32 bytes. Assume a direct mapped cache having 512 cache lines is used with this machine. The size of tag field in bits is _____ **(Gate-2017) (2 Marks)**

(A) 12 **(B)** 16 **(C)** 18 **(D)** 24



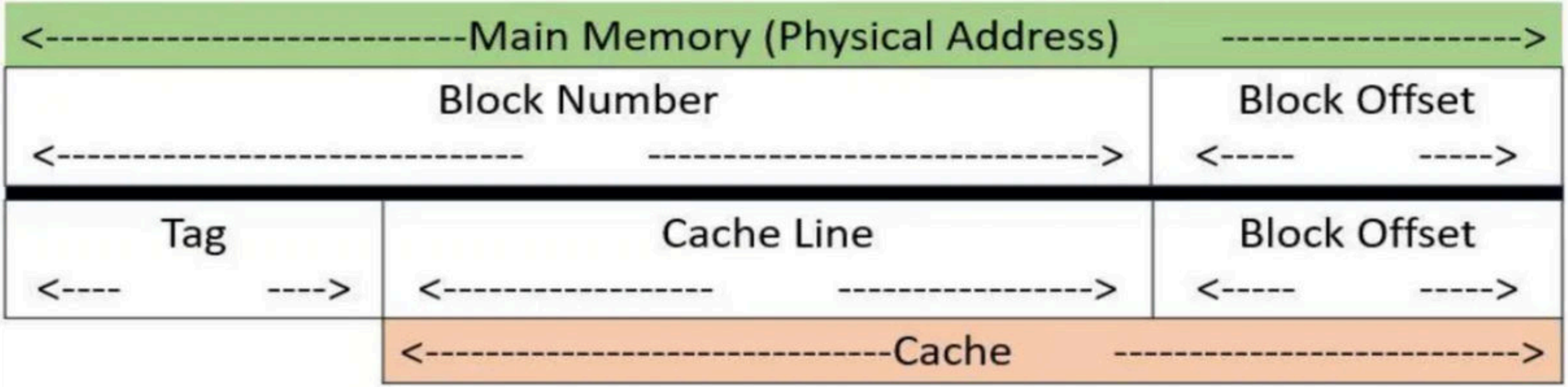
Q Consider a direct mapped cache of size 32 KB with block size 32 bytes. The CPU generates 32 bit addresses. The number of bits needed for cache indexing and the number of tag bits are respectively **(Gate-2005) (2 Marks)**

(A) 10, 17 **(B)** 10, 22 **(C)** 15, 17 **(D)** 5, 17



Q Consider a machine with a byte addressable main memory of 2^{20} bytes, block size of 16 bytes and a direct mapped cache having 2^{12} cache lines. Let the addresses of two consecutive bytes in main memory be $(E201F)_{16}$ and $(E2020)_{16}$. What are the tag and cache line address (in hex) for main memory address $(E201F)_{16}$? **(Gate-2015) (1 Marks)**

(A) E, 201 **(B)** F, 201 **(C)** E, E20 **(D)** 2, 01F



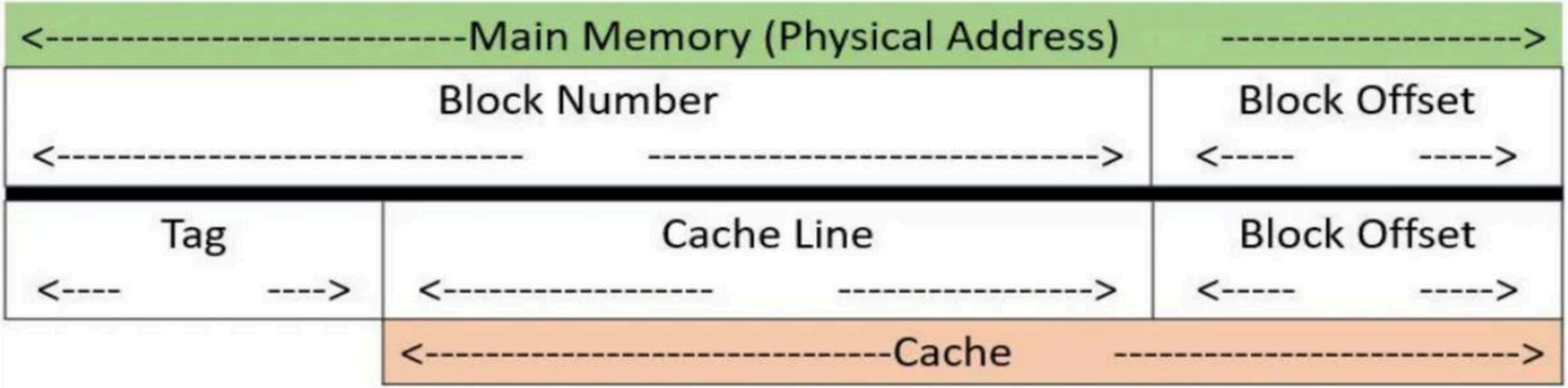
Q An 8KB direct-mapped write-back cache is organized as multiple blocks, each of size 32-bytes. The processor generates 32-bit addresses. The cache controller maintains the tag information for each cache block comprising of the following.

1 Valid bit

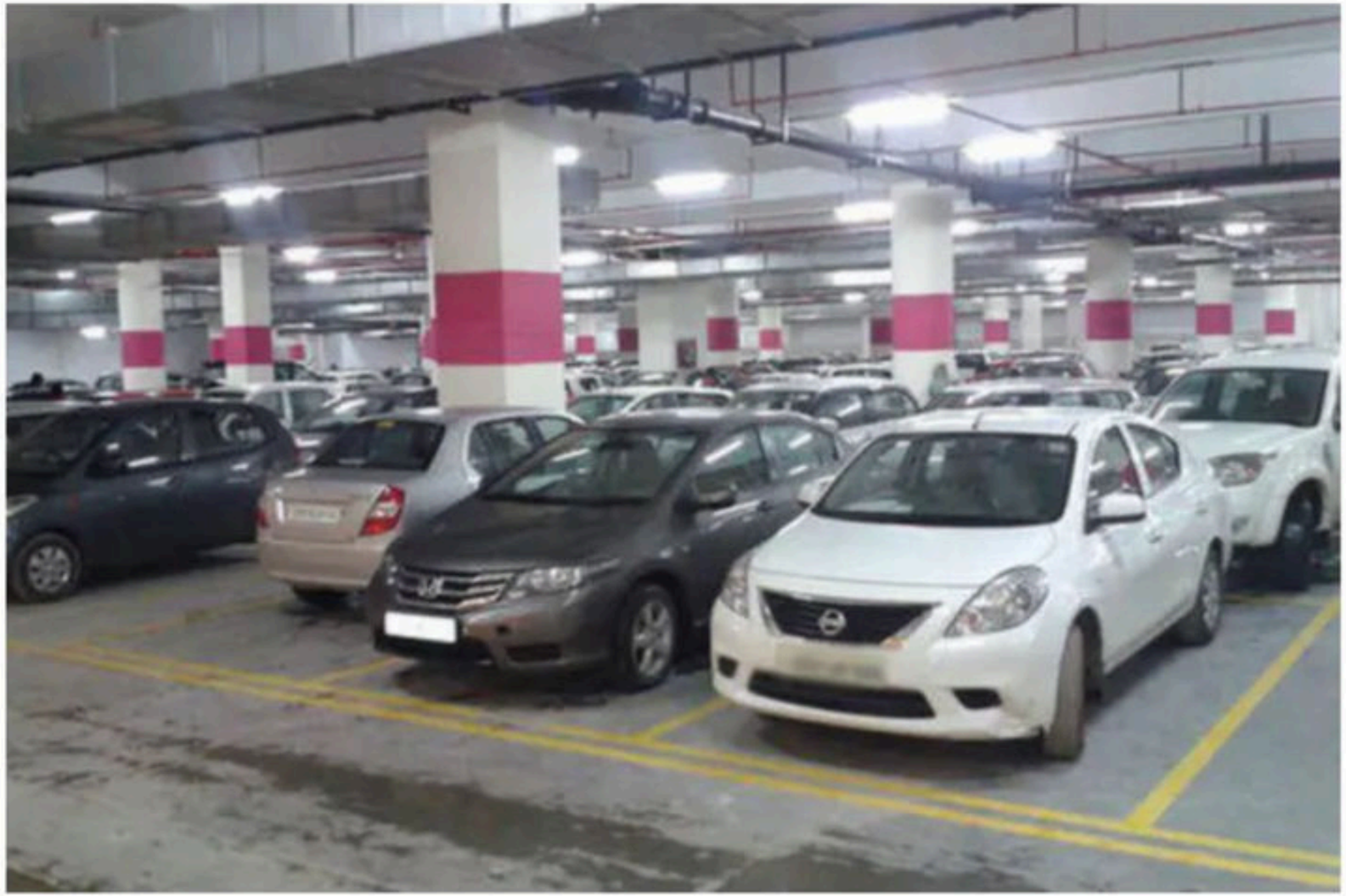
1 Modified bit

As many bits as the minimum needed to identify the memory block mapped in the cache. What is the total size of memory needed at the cache controller to store meta-data (tags) for the cache? **(Gate-2011) (2 Marks)**

(A) 4864 bits **(B)** 6144 bits **(C)** 6656 bits **(D)** 5376 bits



Break

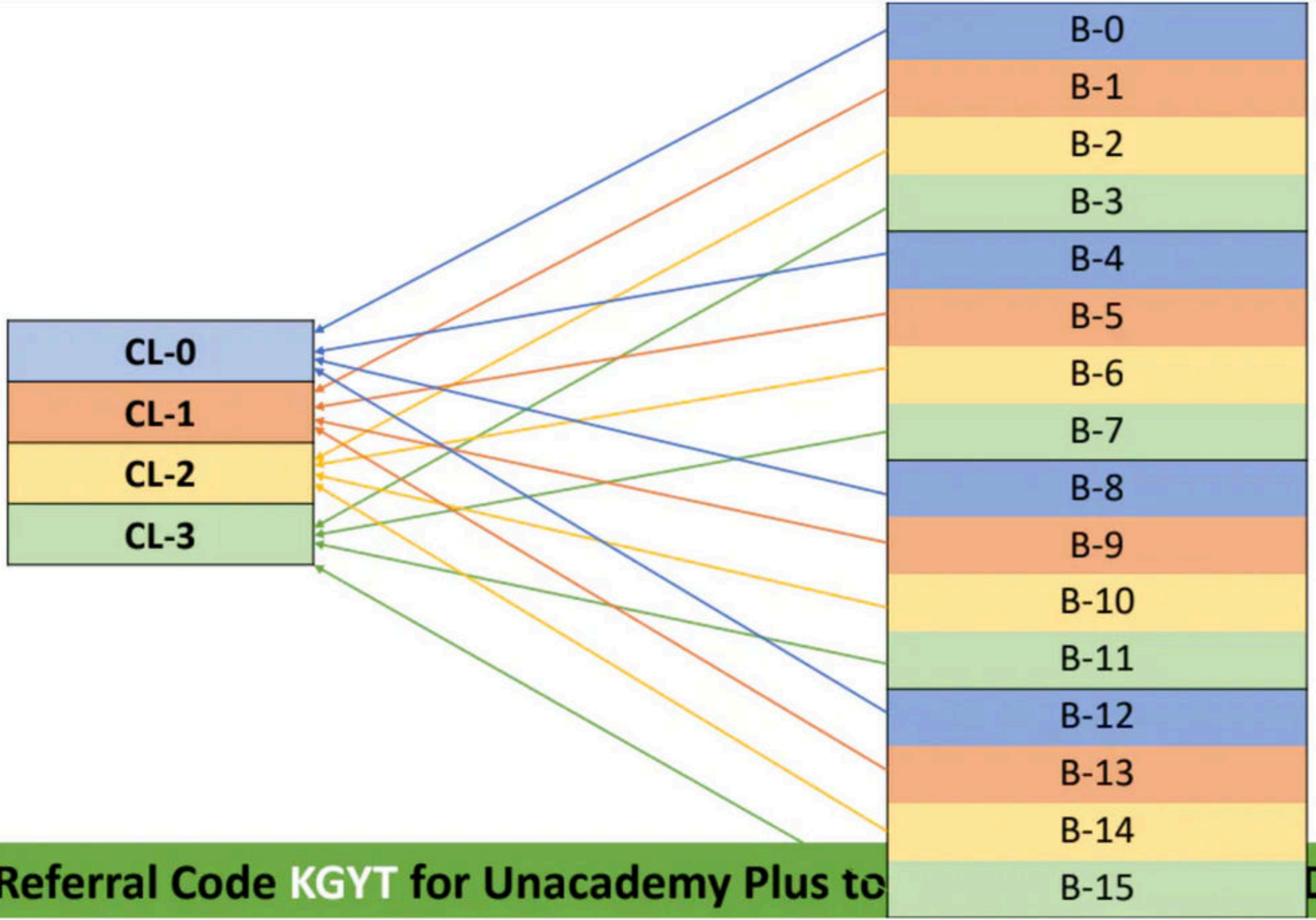


Associative Mapping

- To overcome the problem of conflict-miss in direct mapping we have Associative Mapping.
- A block of main memory can be mapped to any freely available cache line. This makes fully associative mapping more flexible than direct mapping.
- It is also known as many to many mappings.

CL-0
CL-1
CL-2
CL-3

B-0
B-1
B-2
B-3
B-4
B-5
B-6
B-7
B-8
B-9
B-10
B-11
B-12
B-13
B-14
B-15



Cache Memory		
CL-0	TAG = Block no	W-
		W-
		W-
		W-
CL-1	TAG = Block no	W-
		W
		W-
		W-
CL-2	TAG = Block no	W-
		W-
		W-
		W-
CL-3	TAG = Block no	W-
		W-
		W-
		W-

Main Memory				
B-0	W-0	W-1	W-2	W-3
B-1	W-4	W-5	W-6	W-7
B-2	W-8	W-9	W-10	W-11
B-3	W-12	W-13	W-14	W-15
B-4	W-16	W-17	W-18	W-19
B-5	W-20	W-21	W-22	W-23
B-6	W-24	W-25	W-26	W-27
B-7	W-28	W-29	W-30	W-31
B-8	W-32	W-33	W-34	W-35
B-9	W-36	W-37	W-38	W-39
B-10	W-40	W-41	W-42	W-43
B-11	W-44	W-45	W-46	W-47
B-12	W-48	W-49	W-50	W-51
B-13	W-52	W-53	W-54	W-55
B-14	W-56	W-57	W-58	W-59
B-15	W-60	W-61	W-62	W-63

Cache								
CL-0	B-0	W-0	B-4	W-16	B-8	W-32	B-12	W-48
		W-1		W-17		W-33		W-49
		W-2		W-18		W-34		W-50
		W-3		W-19		W-35		W-51
CL-1	B-1	W-4	B-5	W-20	B-9	W-36	B-13	W-52
		W-5		W-21		W-37		W-53
		W-6		W-22		W-38		W-54
		W-7		W-23		W-39		W-55
CL-2	B-2	W-8	B-6	W-24	B-10	W-40	B-14	W-56
		W-9		W-25		W-41		W-57
		W-10		W-26		W-42		W-58
		W-11		W-27		W-43		W-59
CL-3	B-3	W-12	B-7	W-28	B-11	W-44	B-15	W-60
		W-13		W-29		W-45		W-61
		W-14		W-30		W-46		W-62
		W-15		W-31		W-47		W-63

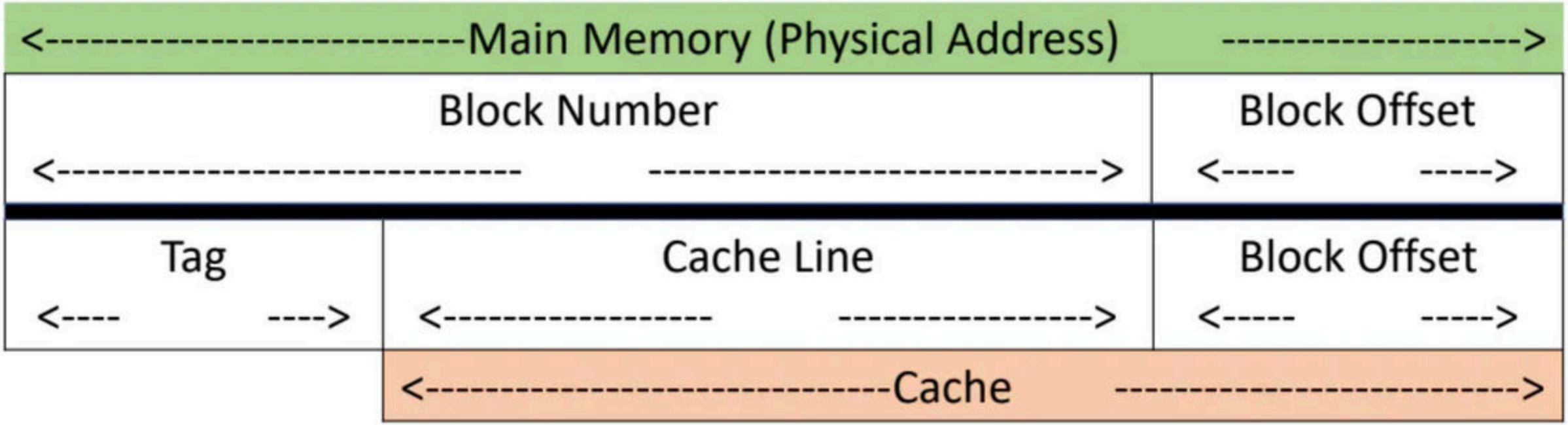
Cache Memory	
CL-0	B-0 / B-4 / B-8 / B-12
CL-1	B-1 / B-5 / B-9 / B-13
CL-2	B-2 / B-6 / B-10 / B-14
CL-3	B-3 / B-7 / B-11 / B-15

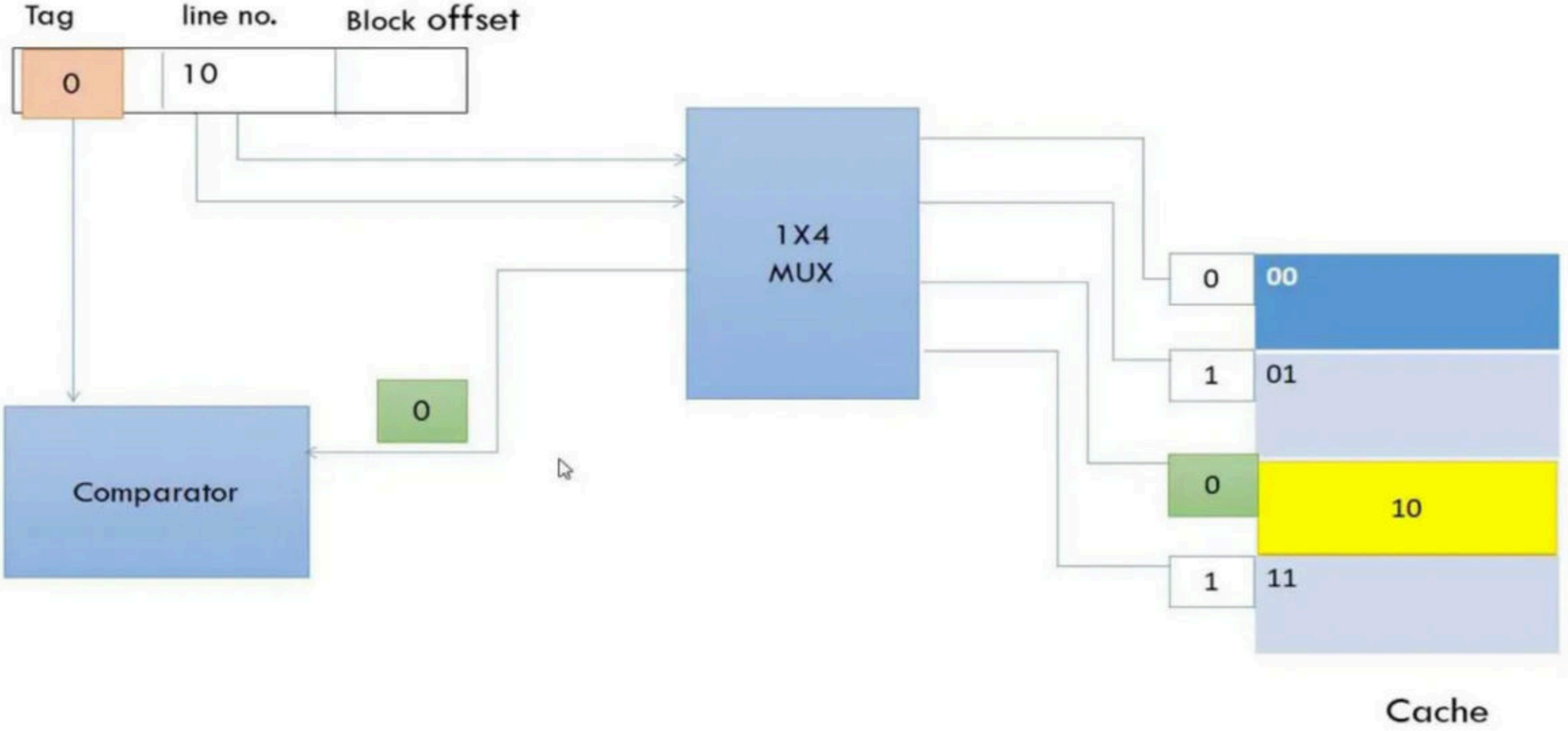
Cache Memory		
CL-0	TAG	W-
		W-
		W-
		W-
CL-1	TAG	W-
		W
		W-
		W-
CL-2	TAG	W-
		W-
		W-
		W-
CL-3	TAG	W-
		W-
		W-
		W-

- A replacement algorithm is needed to replace a block if the cache is full.

- In fully associative mapping we only have two fields: Tag/Block Number field and a Block offset field.

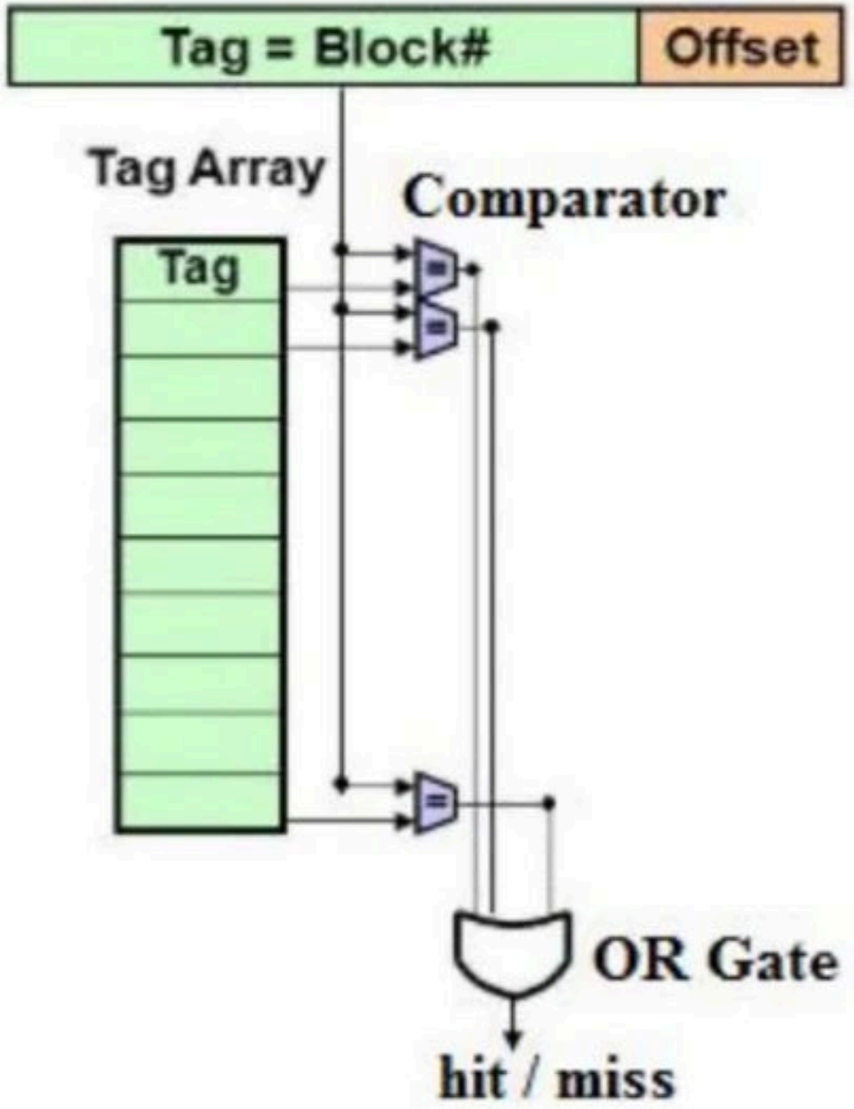
- Here the number of bits in tag = number of bits to require to represent block number

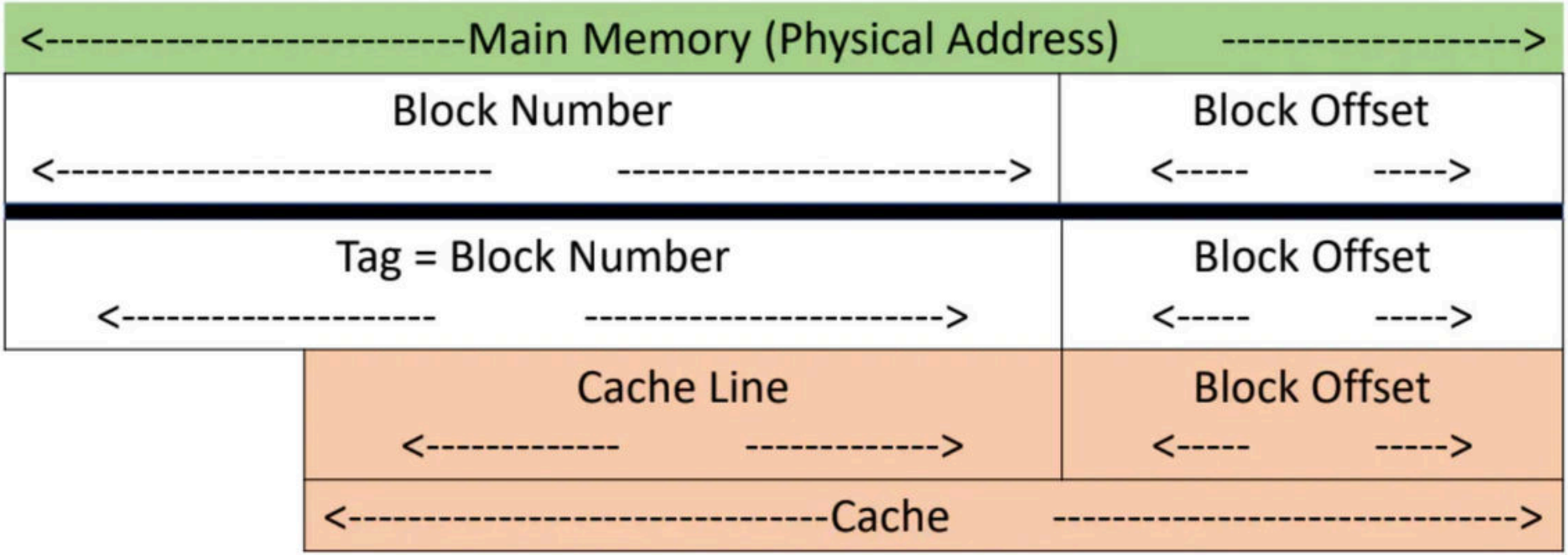




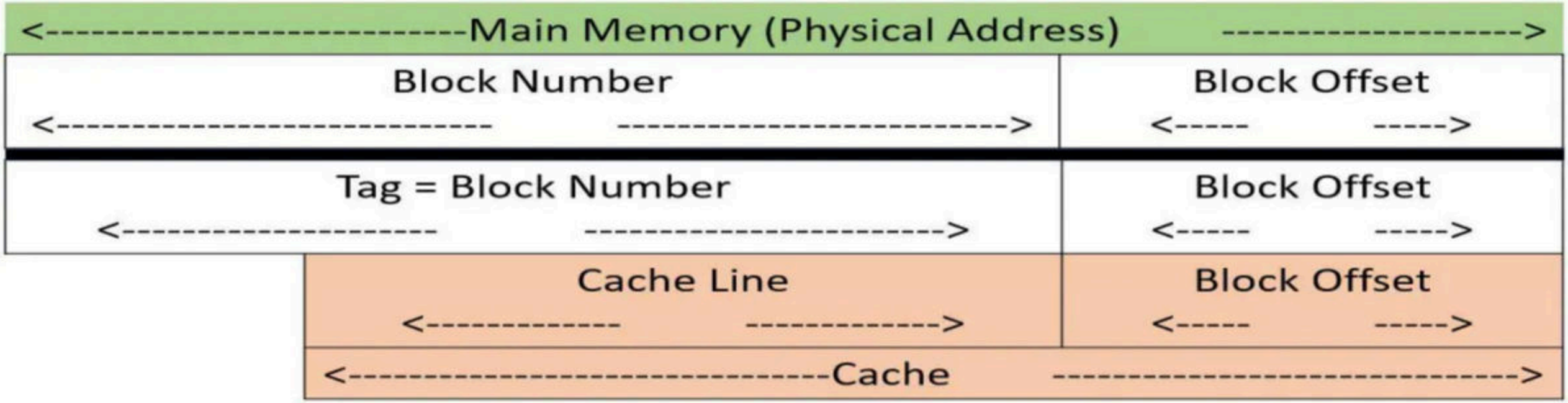
Hardware Architecture

- If we have ‘n’ lines in cache then ‘n’ number of comparators are required.
- Size of comparator = Size of Tag
- If we have ‘n’ bit tag then we require ‘n’ bit comparator.

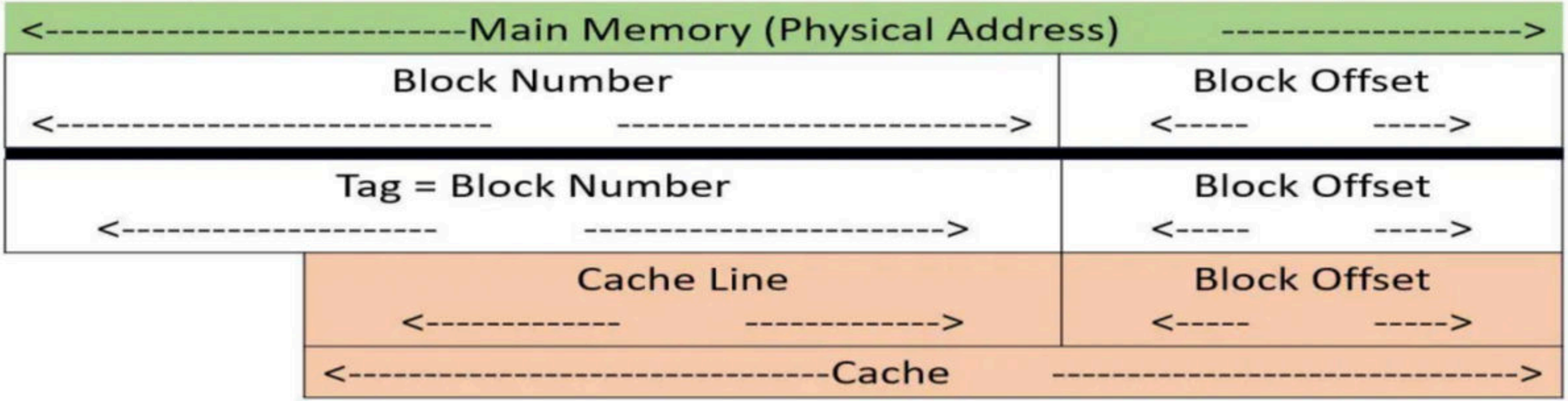




Q Consider a fully associative mapped cache of size 16 KB with block size 256 bytes. The size of main memory is 128 KB. Find out the: Number of bits in tag and Tag directory size?



MM Size	Cache Size	Block Size	No of bits in Tag	Tag Directory Size	Comp
128 KB	16 KB	256 B			
32 GB	32 KB	1 KB			
	512 KB	1 KB	17		
16 GB		4 KB	22		
64MB			10		
	512 KB		7		



MM Size	Cache Size	Block Size	No of bits in Tag	Tag Directory Size	Comp
128 KB	16 KB	256 B	9	9 * 64	64
32 GB	32 KB	1 KB	25	25 * 32	32
128 MB	512 KB	1 KB	17	512 *17	512
16 GB	?	4 KB	22	?	?
64MB	?	64 KB	10	?	?
?	512 KB		7	?	?

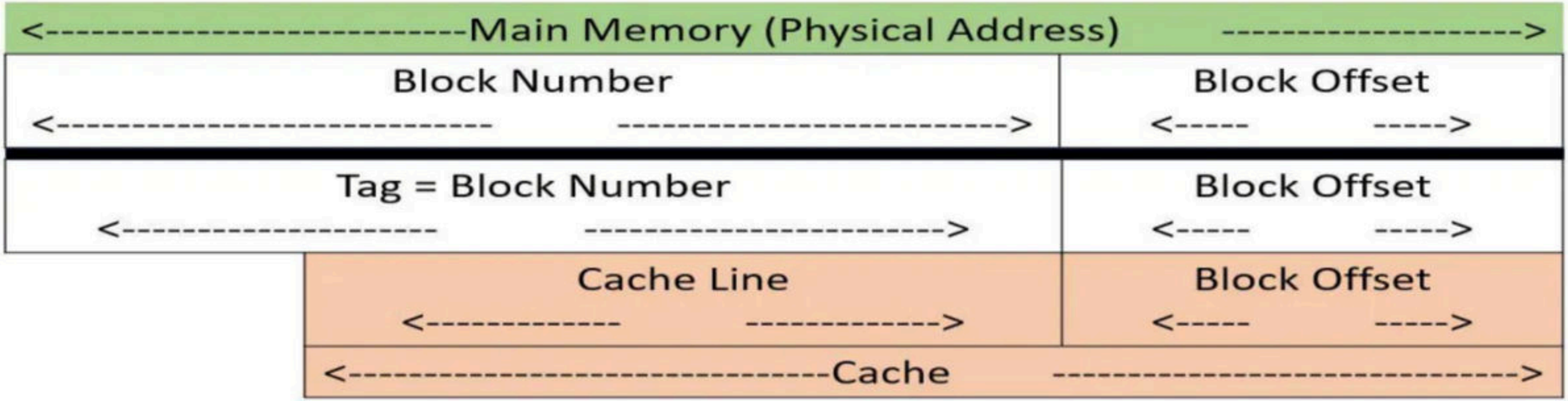
Hit latency = Time taken by one comparator (Comparators are in Parallel) + time taken by OR Gate

Example: Let us assume that main memory size is 32GB, block size is 32KB and the propagation delay for the comparators and OR gate is 10 K ns and 10 ns respectively. Calculate the tag size and hit latency?

Main Memory = 32 GB = $2^{30} * 2^5 = 2^{35}$ = 35 bits Physical Address.

Block Size = 32 KB = 2^{15} = 15 bits block size.

So, the block number = 35 – 15 = 20 bits



Propagation Delay = 10 * 20 = 200 ns.

Hit latency: 200 ns + 10 ns (Propagation Delay of OR gate) = 210 ns.

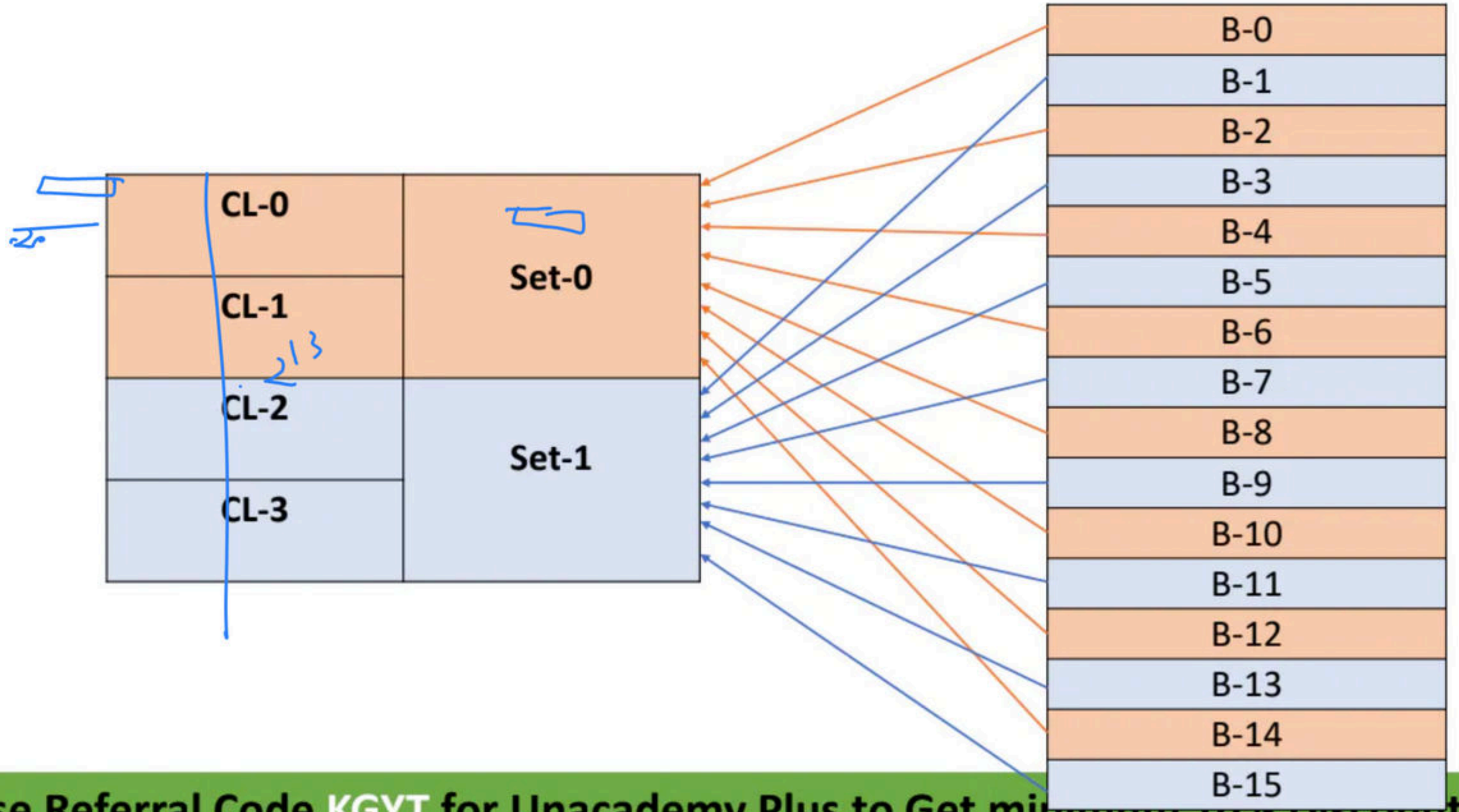
Disadvantage

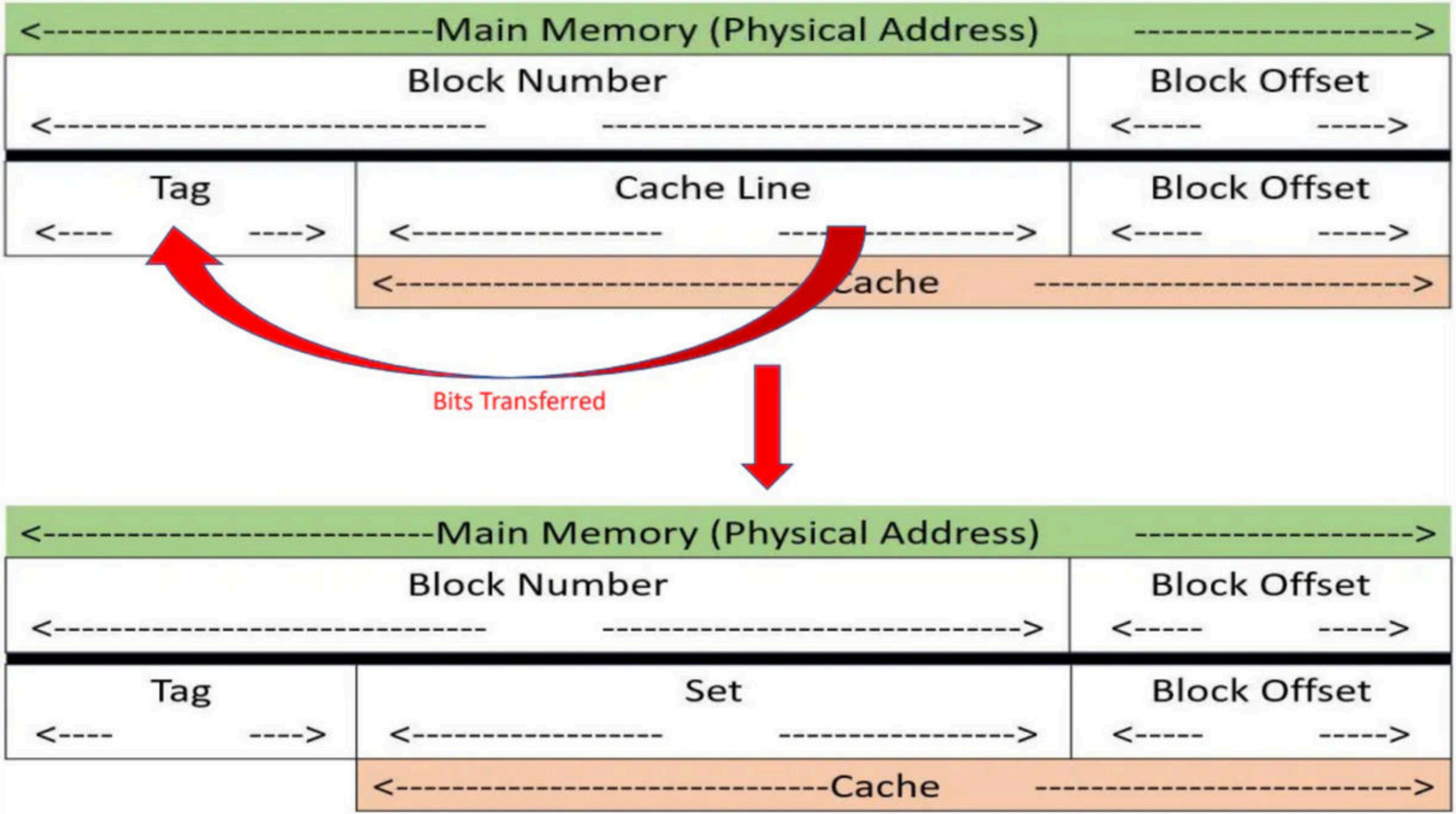
- Hardware cost is high as compared to direct mapping.
- Tag directory size is more as compared to direct mapping.

Break

Set Associative Mapping

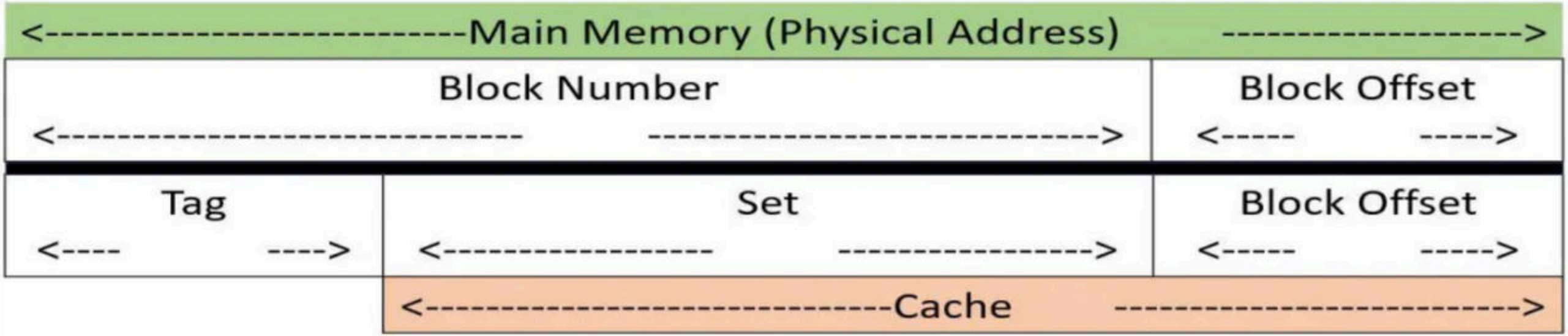
- In k-way set associative mapping, cache lines are grouped into sets where each set contains "k" number of lines.
- A particular block of main memory can map to only one particular set of the cache.
- However, within that set, the memory block can map to any freely available cache line.




<----------Main Memory (Physical Address)----------
Block Number
Block Offset
Tag
Cache Line
Block Offset
Cache
Bits Transferred
<----------Main Memory (Physical Address)----------
Block Number
Block Offset
Tag
Set
Block Offset
<----------Cache----------

Formulas

- Number of Sets = No of Lines in Cache / No of Cache line in a set(k)(k-way set associative)

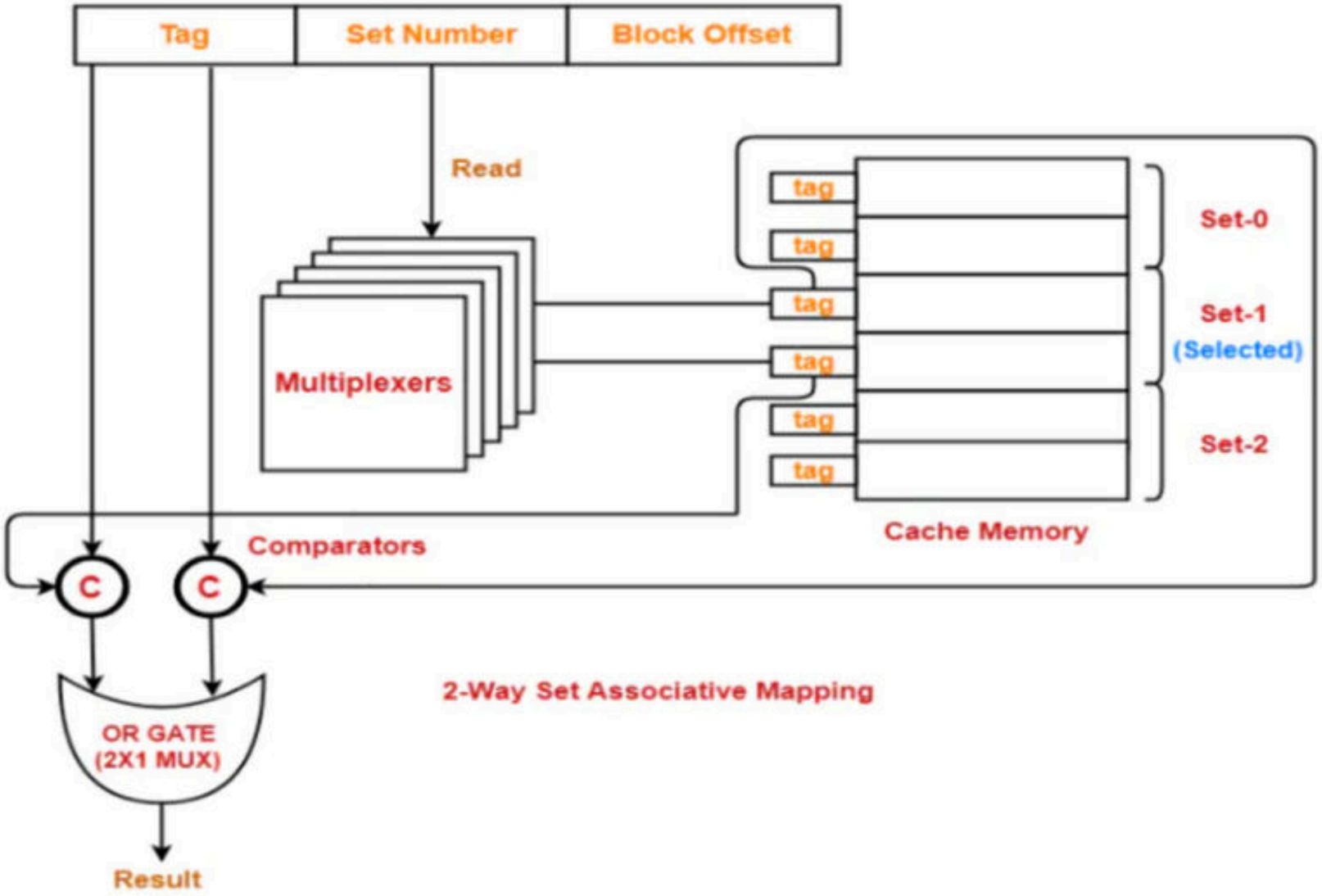


Main Memory				
B-0	W-0	W-1	W-2	W-3
B-1	W-4	W-5	W-6	W-7
B-2	W-8	W-9	W-10	W-11
B-3	W-12	W-13	W-14	W-15
B-4	W-16	W-17	W-18	W-19
B-5	W-20	W-21	W-22	W-23
B-6	W-24	W-25	W-26	W-27
B-7	W-28	W-29	W-30	W-31
B-8	W-32	W-33	W-34	W-35
B-9	W-36	W-37	W-38	W-39
B-10	W-40	W-41	W-42	W-43
B-11	W-44	W-45	W-46	W-47
B-12	W-48	W-49	W-50	W-51
B-13	W-52	W-53	W-54	W-55
B-14	W-56	W-57	W-58	W-59
B-15	W-60	W-61	W-62	W-63

Cache			
Set Number-0	CL-0	TAG = Block no – CL	W-
			W-
			W-
			W-
	CL-1	TAG = Block no – CL	W-
			W
			W-
			W-
Set Number-1	CL-2	TAG = Block no – CL	W-
			W-
			W-
			W-
	CL-3	TAG = Block no - CL	W-
			W-
			W-
			W-

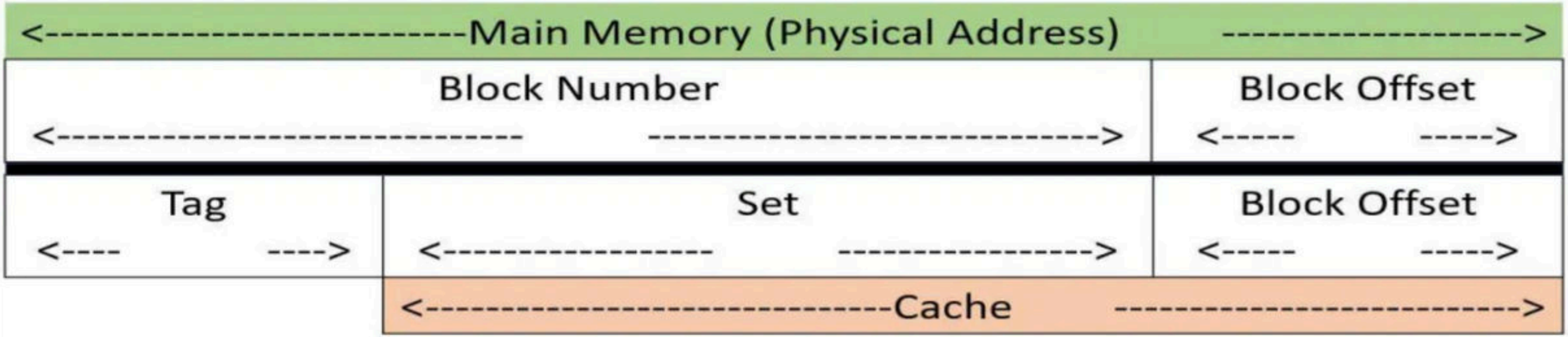
	Cache									
Set Number-0	CL-0	Tag	B-0	W-0	B-2	W-8	B-4	W-16	B-6	W-24
				W-1		W-9		W-17		W-25
				W-2		W-10		W-18		W-26
				W-3		W-11		W-19		W-27
	CL-1	Tag	B-8	W-32	B-10	W-40	B-12	W-48	B-14	W-56
				W-33		W-41		W-49		W-57
				W-34		W-42		W-50		W-58
				W-35		W-43		W-51		W-59
Set Number-1	CL-2	Tag	B-1	W-4	B-3	W-12	B-5	W-20	B-7	W-28
				W-5		W-13		W-21		W-29
				W-6		W-14		W-22		W-30
				W-7		W-15		W-23		W-31
	CL-3	Tag	B-9	W-36	B-11	W-44	B-13	W-52	B-15	W-60
				W-37		W-45		W-53		W-61
				W-38		W-46		W-54		W-62
				W-39		W-47		W-55		W-63

Hardware Architecture

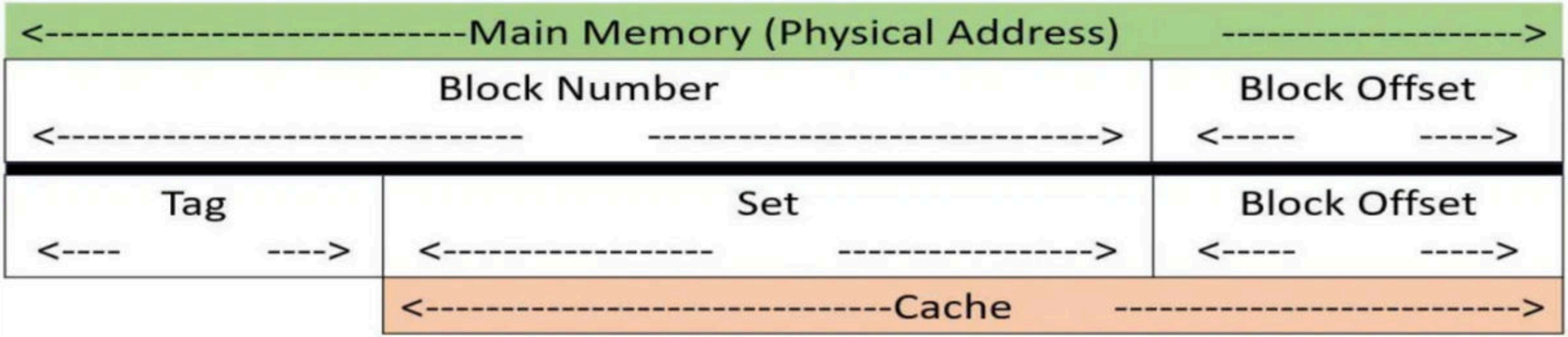


2-Way Set Associative Mapping

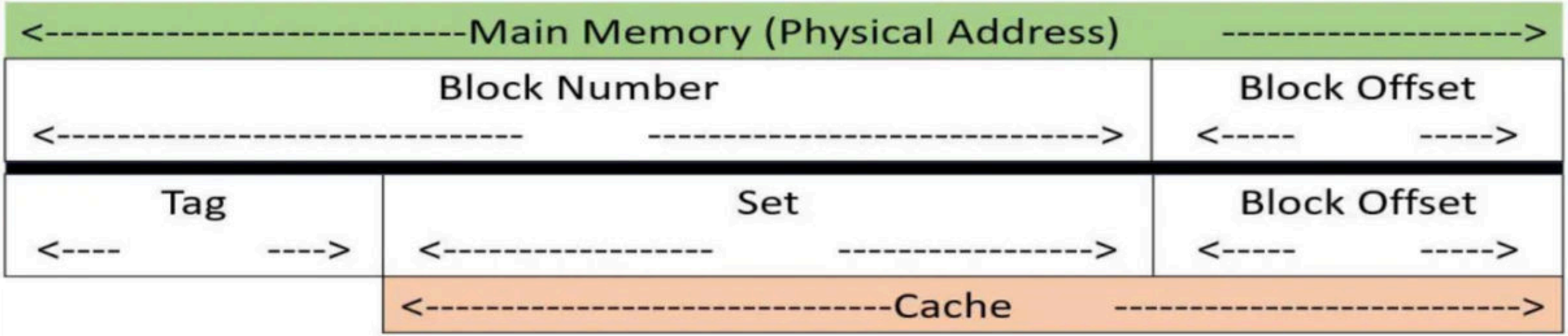
Q Consider the main memory size is of 128 KB, the cache size is of 16 KB, the block size is of 256 B, the set size is 2. Find Tag.



Q Main Memory = 32 GB, Cache Size = 32 KB, Block Size = 1 KB and it is a 4-way set associative cache?



MM Size	Cache Size	Block Size	No of bits in Tag	Tag Directory Size	Set Associative
128 KB	16 KB	256 B			2-way
32 GB	32 KB	1 KB			4-way
	512 KB	1 KB	7		8-way
16 GB		4 KB	10		4-way
64MB			10		4-way
	512 KB		7		8-way

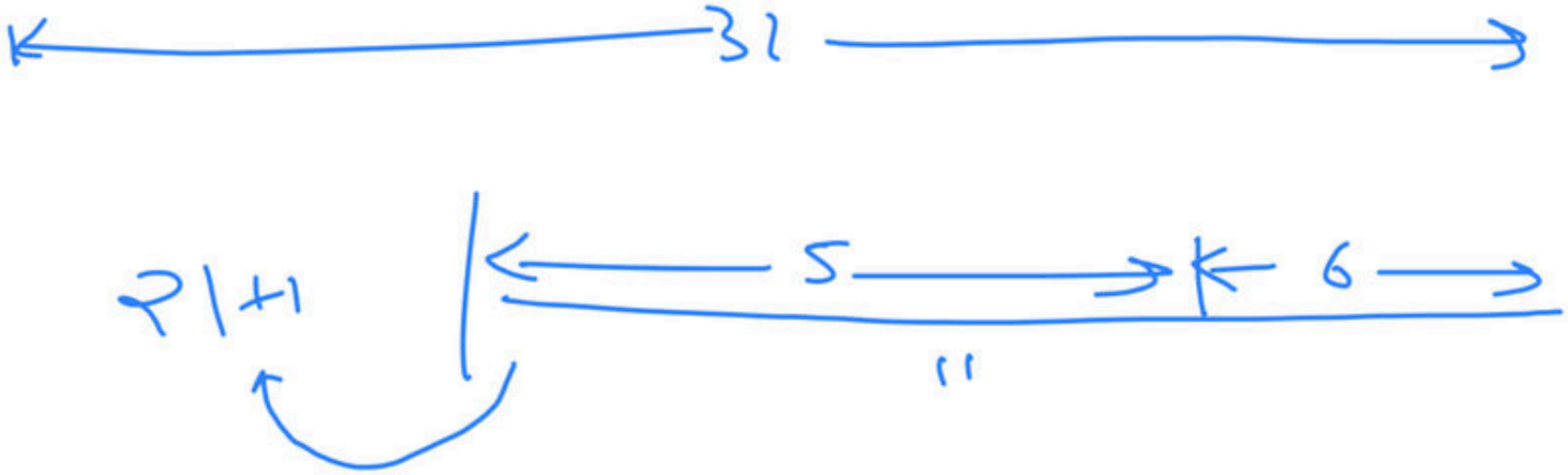


MM Size	Cache Size	Block Size	No of bits in Tag	Tag Directory Size	Set Associative
128 KB	16 KB	256 B	4	$4 * 2^6$	2-way
32 GB	32 KB	1 KB	22	22 * 32	4-way
2^{23} B	512 KB	1 KB	7	$7 * 2^9$	8-way
16 GB	2^{26} B	4 KB	10	$10 * 2^{14}$	4-way
64MB	?	?	10	?	4-way
?	512 KB	?	7	?	8-way

Q.55 Consider a set-associative cache of size 2 KB (1 KB = 2^{10} bytes) with cache block size of 64 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 22 bits, the associativity of the ache is 2.

(GATE- 2021)

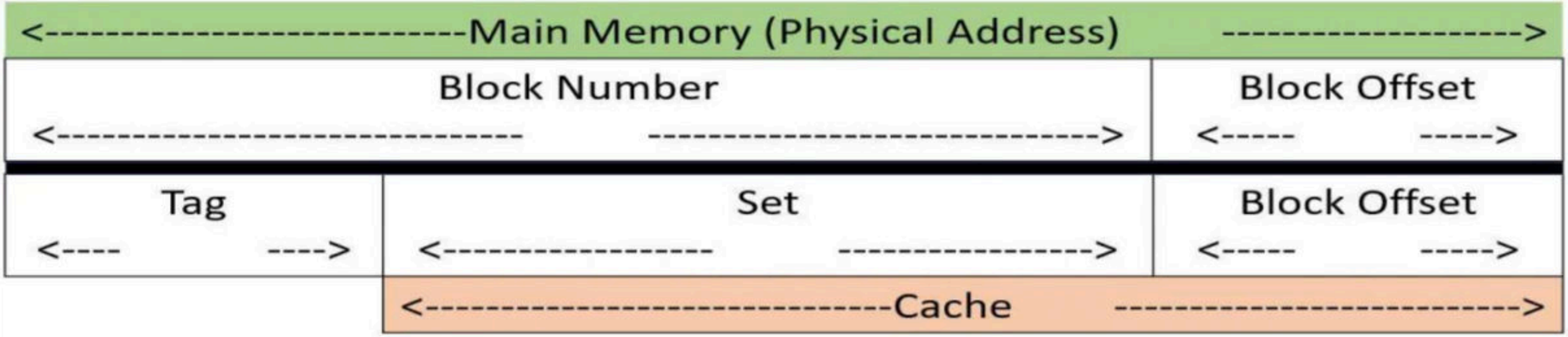
Ans. (2)



Q A computer system with a word length of 32 bits has a 16 MB byte- addressable main memory and a 64 KB, 4-way set associative cache memory with a block size of 256 bytes. Consider the following four physical addresses represented in hexadecimal notation. **(Gate-2020) (2 Marks)**

A_1 = 0x42C8A4, A_2 = 0x546888, A_3 = 0x6A289C, A_4 = 0x5E4880

(A) A_1 and A_4 are mapped to different cache sets.
(B) A_2 and A_3 are mapped to the same cache set.
(C) A_3 and A_4 are mapped to the same cache set.
(D) A_1 and A_3 are mapped to the same cache set.



Q The size of the physical address space of a processor is 2^P bytes. The word length is 2^W bytes. The capacity of cache memory is 2^N bytes. The size of each cache block is 2^M words. For a K-way set-associative cache memory, the length (in number of bits) of the tag field is **(Gate-2018) (2 Marks)**

(A) $P - N - \log_2 K$

(B) $P - N + \log_2 K$

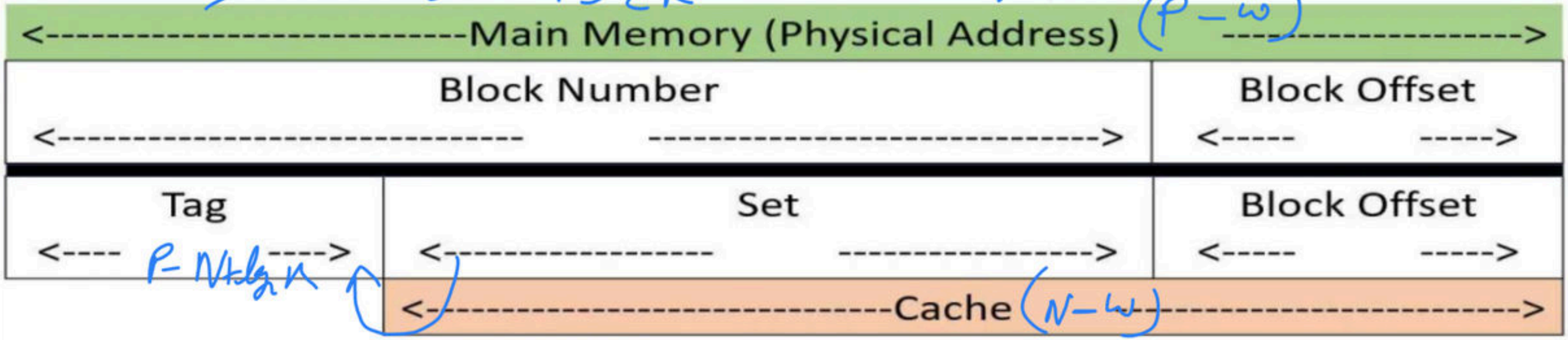
(C) $P - N - M - W - \log_2 K$

(D) $P - N - M - W + \log_2 K$

$$\frac{2^P}{2^W} = 2^{P-W}$$

$$= (P-W) - (N-W) + \log_2 K$$

$$P - W - N + W + \log_2 K = P - N + \log_2 K$$



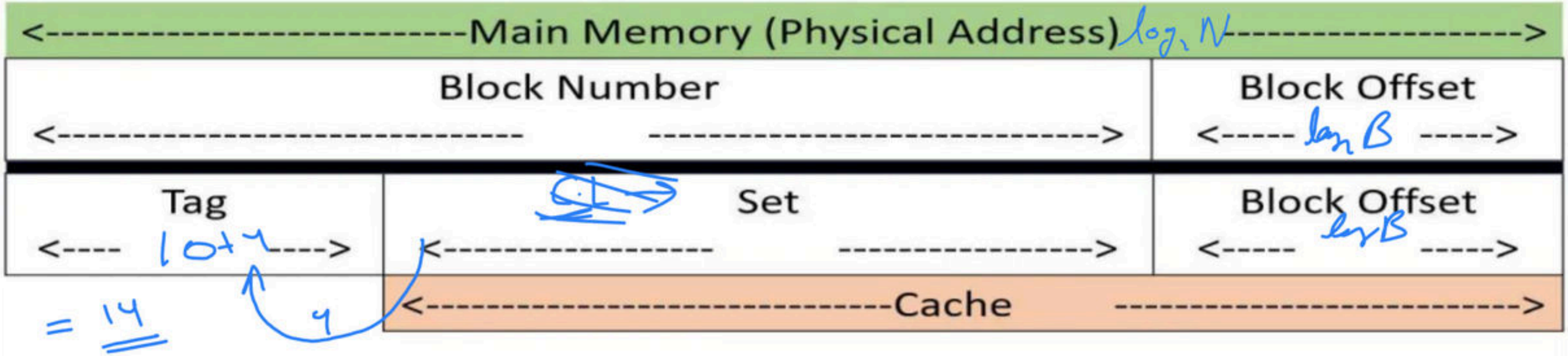
Q A cache memory unit with capacity of N words and block size of B words is to be designed. If it is designed as direct mapped cache, the length of the TAG field is 10 bits. If the cache unit is now designed as a 16-way set-associative cache, the length of the TAG field is 14 bits. **(Gate-2017) (1 Marks)**

9 → a) 16

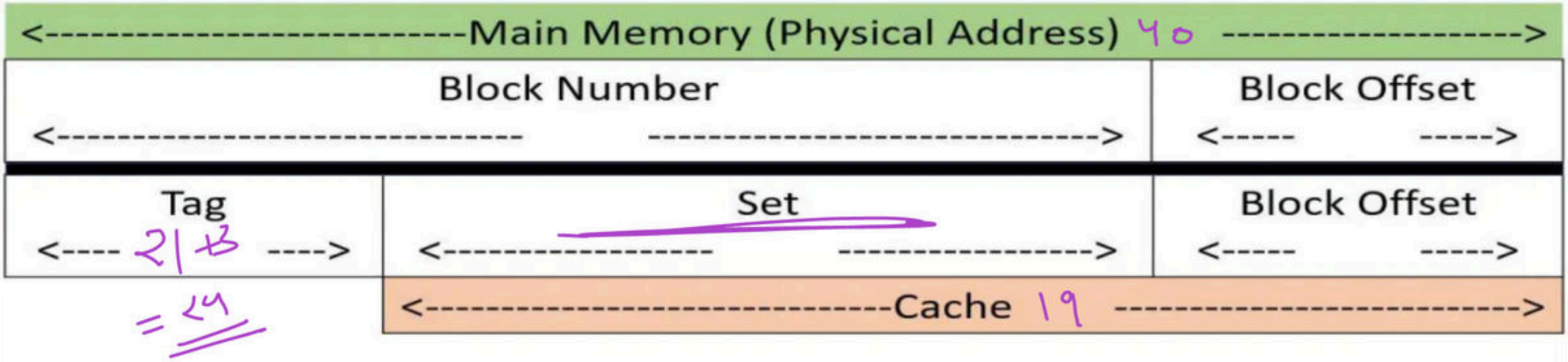
75 → b) 14

7 → c) 4

9 → d) ...



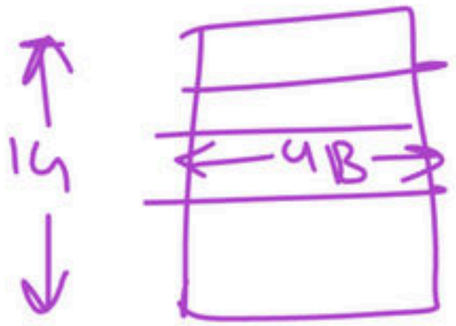
Q The width of the physical address on a machine is 40 bits. The width of the tag field in a 512 KB 8-way set associative cache is 24 bits **(Gate-2016) (2 Marks)**



Q A 4-way set associative cache memory unit with a capacity of 16 KB is built using a block size of 8 words. The word length is 32 bits. The size of the physical address space is 4 GB. The number of bits for the TAG field is _____ **(Gate-2014) (1 Marks)**

$\frac{16KB}{4B} = 4K$

$\frac{4GB}{4B}$



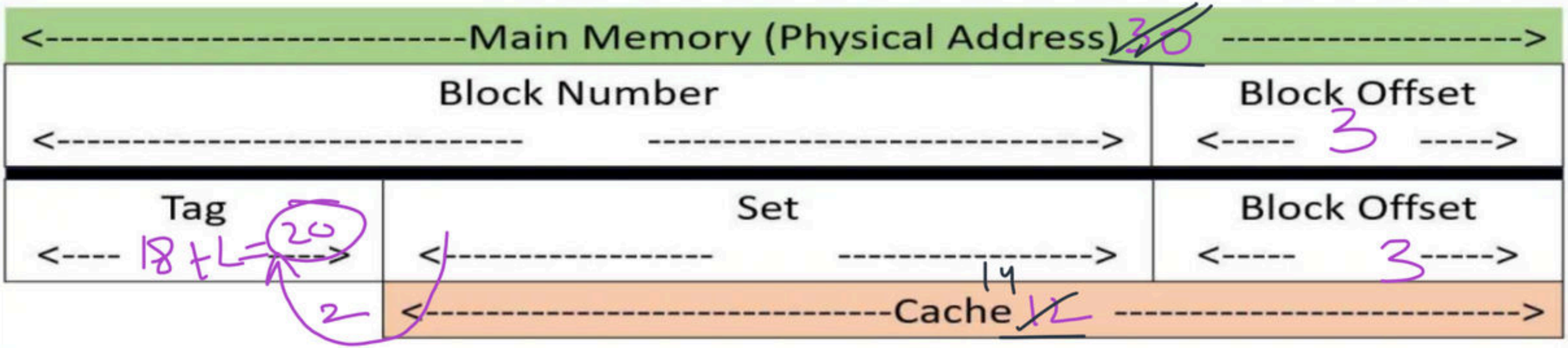
1 → a) 16

19 → b) 18

2 → c) 20

1 → a) None of the above

32



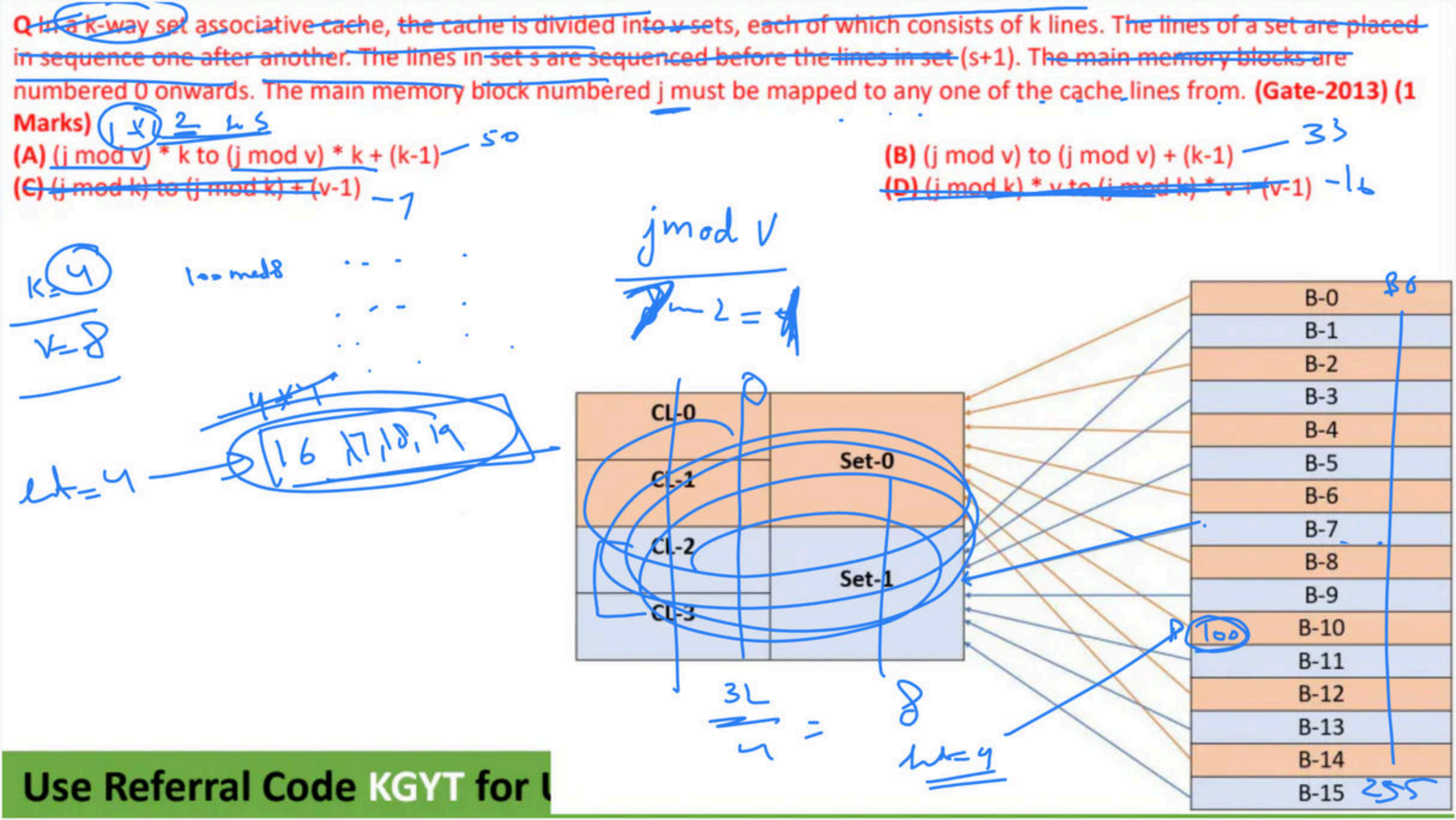
Q In a k-way set associative cache, the cache is divided into v sets, each of which consists of k lines. The lines of a set are placed in sequence one after another. The lines in set s are sequenced before the lines in set (s+1). The main memory blocks are numbered 0 onwards. The main memory block numbered j must be mapped to any one of the cache lines from. **(Gate-2013) (1 Marks)**

(A) (j mod v) * k to (j mod v) * k + (k-1)

(B) (j mod v) to (j mod v) + (k-1)

(C) (j mod k) to (j mod k) + (v-1)

(D) (j mod k) * v to (j mod k) * v + (v-1)

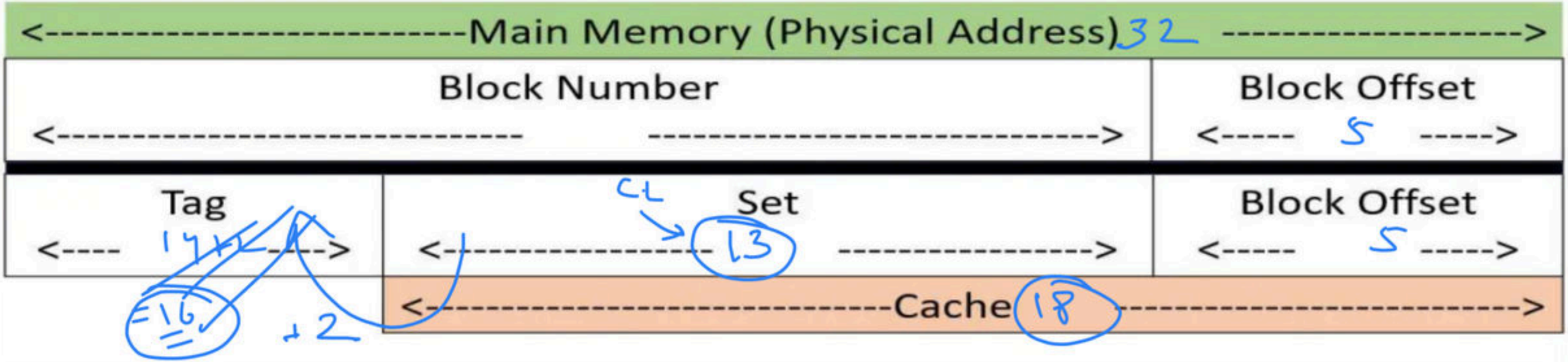


Q A computer has a 256 KByte, 4-way set associative, write back data cache with block size of 32 Bytes. The processor sends 32-bit addresses to the cache controller. Each cache tag directory entry contains, in addition to address tag, 2 valid bits, 1 modified bit and 1 replacement bit. The number of bits in the tag field of an address is **(Gate-2012) (2 Marks)**

(A) 11 **(B)** 14 **(C)** 16 **(D)** 27

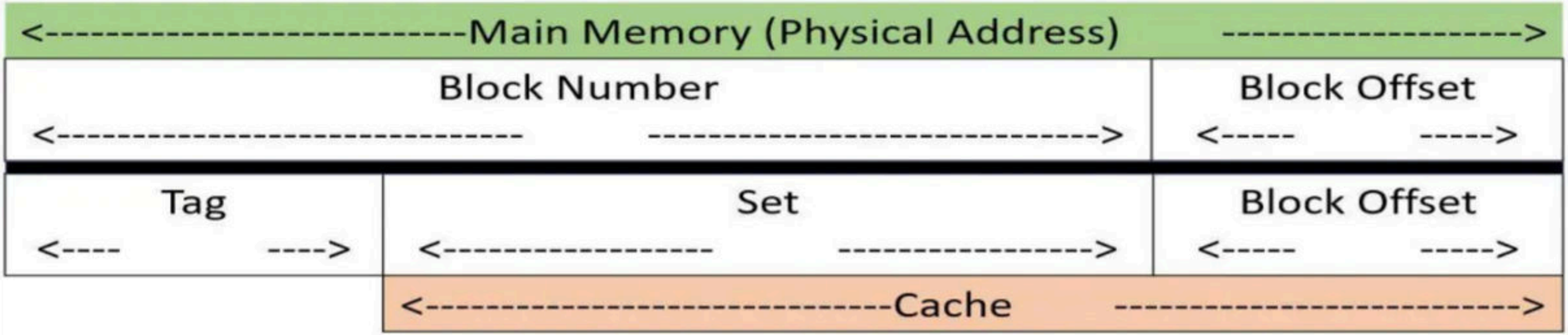
20×2^{13}

20 8k = 16kbit



Q The size of the cache tag directory is **(Gate-2012) (2 Marks)**

a) 160 Kbits **b)** 136 Kbits **c)** 40 Kbits **d)** 32 Kbits



Q Consider a computer with a 4-ways set-associative mapped cache of the following characteristics: a total of 1 MB of main memory, a word size of 1 byte, a block size of 128 words and a cache size of 8 KB. The number of bits in the TAG, SET and WORD fields, respectively are: **(Gate-2008) (2 Marks)**

(A) 7, 6, 7

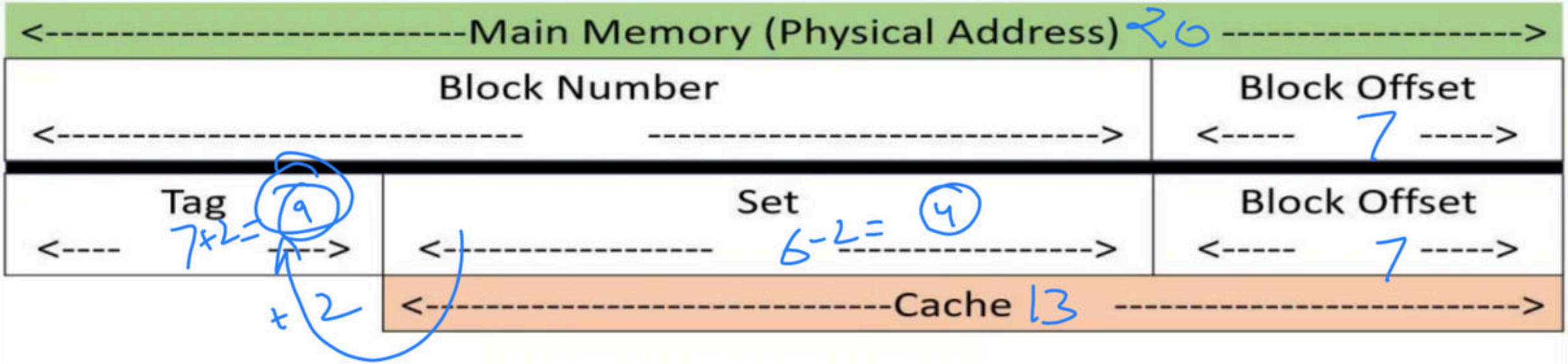
(B) 8, 5, 7

(C) 8, 6, 6

(D) 9, 4, 7

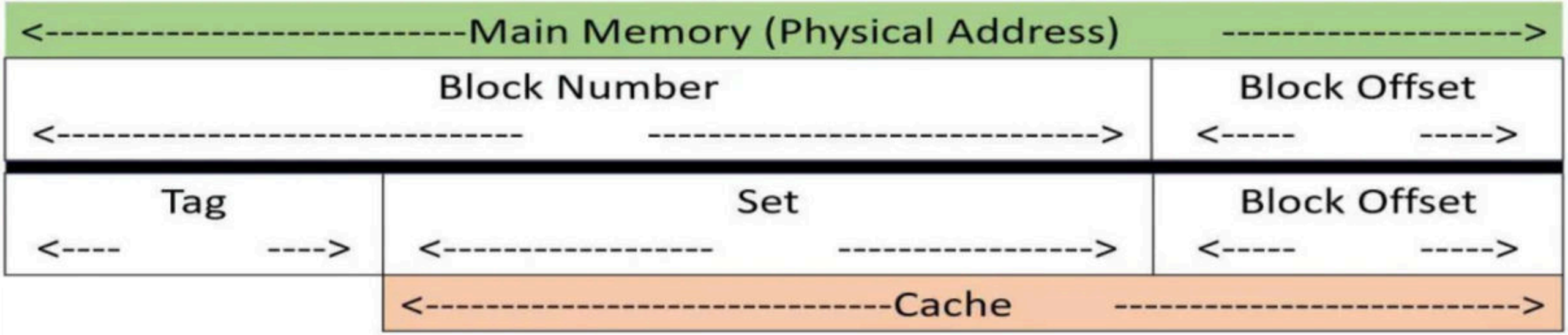
9, 6, 7

9, 4, 7



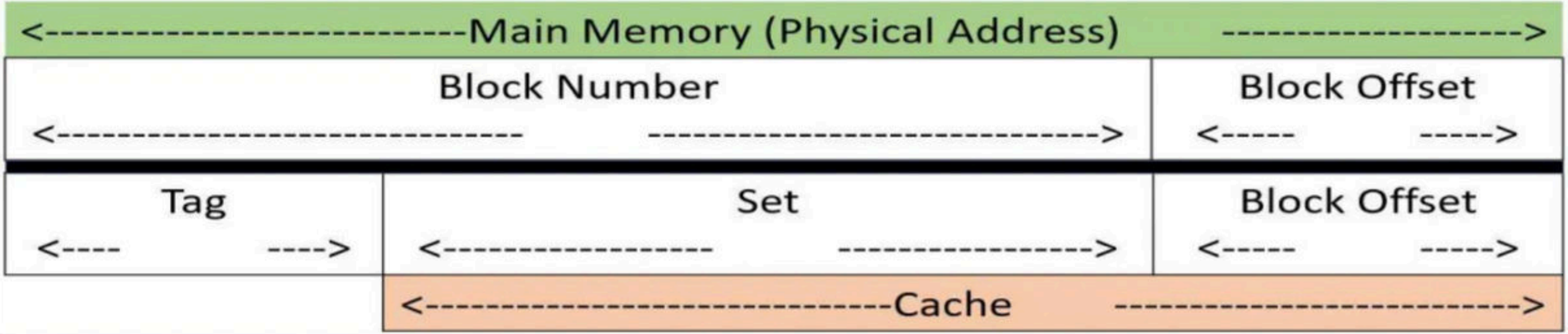
Q Consider a computer with a 4-ways set-associative mapped cache of the following characteristics: a total of 1 MB of main memory, a word size of 1 byte, a block size of 128 words and a cache size of 8 KB. While accessing the memory location 0C795H by the CPU, the contents of the TAG field of the corresponding cache line is: **(Gate-2008) (2 Marks)**

a) 000011000 **b)** 110001111 **c)** 00011000 **d)** 110010101



Q Consider a 4-way set associative cache consisting of 128 lines with a line size of 64 words. The CPU generates a 20-bit address of a word in main memory. The number of bits in the TAG, LINE and WORD fields are respectively: **(Gate-2007) (1 Marks)**

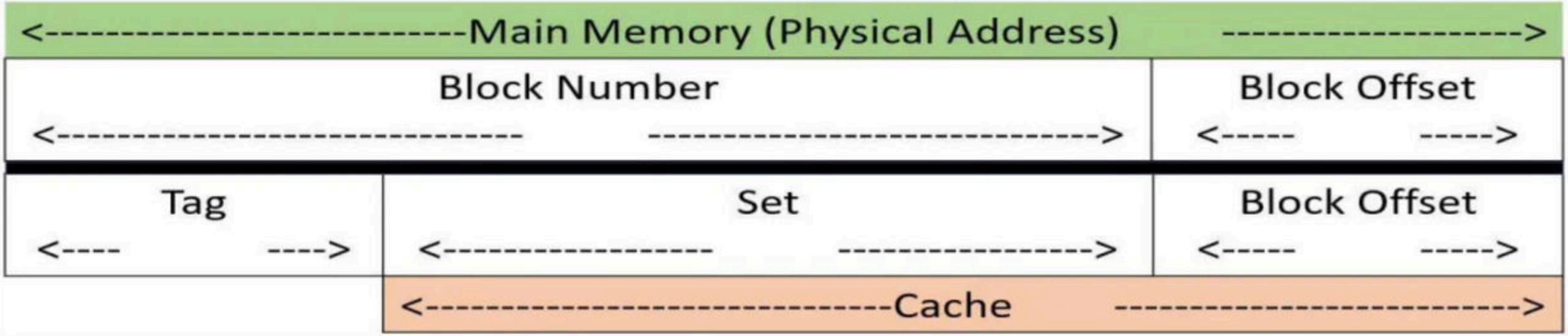
(A) 9,6,5 **(B)** 7, 7, 6 **(C)** 7, 5, 8 **(D)** 9, 5, 6



Q A computer system has a level-1 instruction cache (1-cache), a level-1 data cache (D-cache) and a level-2 cache (L2-cache) with the following specifications:
The length of the physical address of a word in the main memory is 30 bits. The capacity of the tag memory in the I-cache, D-cache and L2-cache is, respectively,
(Gate-2006) (2 Marks)

(A) 1 K x 18-bit, 1 K x 19-bit, 4 K x 16-bit
(B) 1 K x 16-bit, 1 K x 19-bit, 4 K x 18-bit
(C) 1 K x 16-bit, 512 x 18-bit, 1 K x 16-bit
(D) 1 K x 18-bit, 512 x 18-bit, 1 K x 18-bit

	Capacity	Mapping Method	Block size
I-cache	4K words	Direct mapping	4 Words
D-cache	4K words	2-way set associative mapping	4 Words
L2-cache	64K words	4-way set associative mapping	16 Words



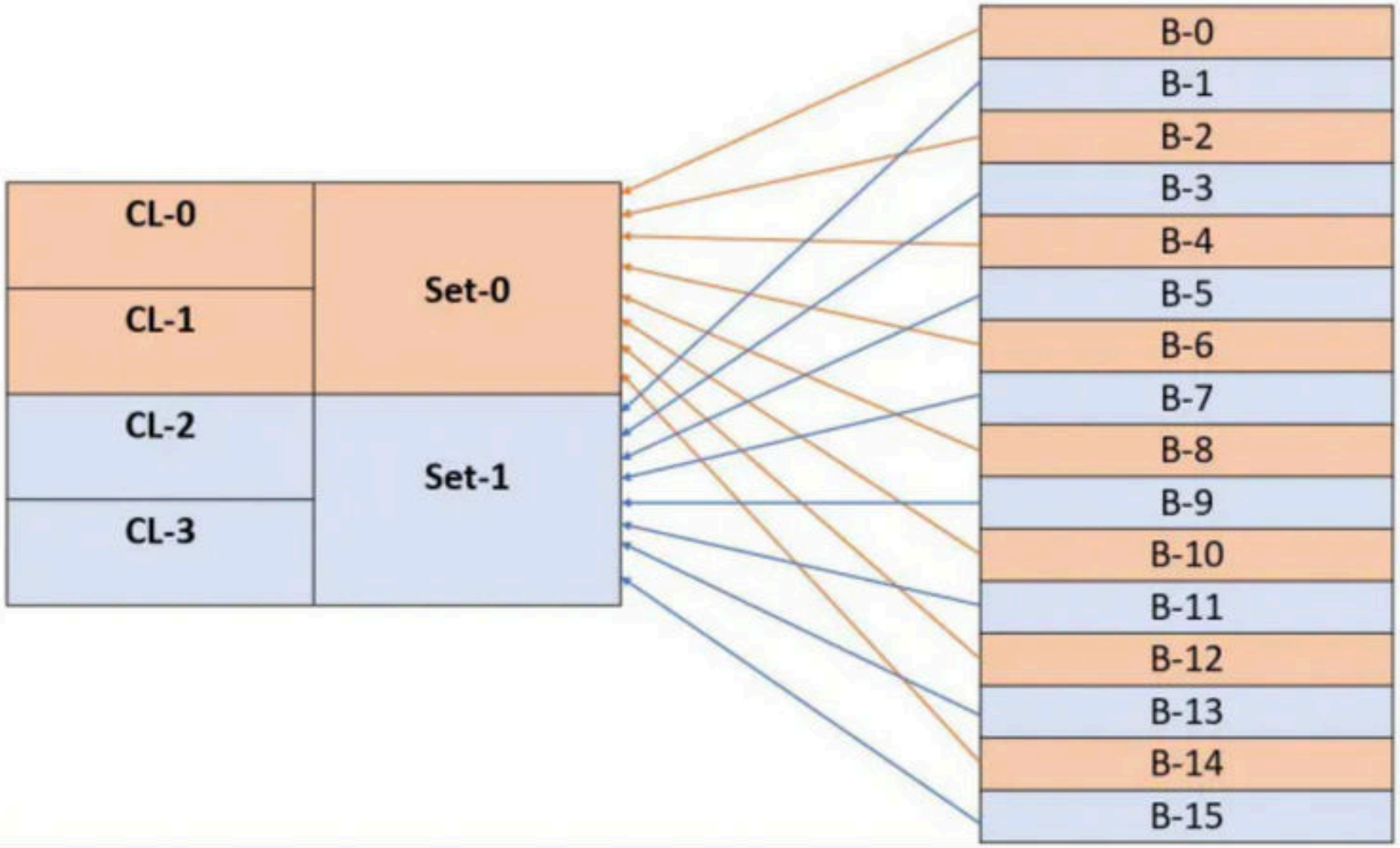
Q The main memory of a computer has 2 cm blocks while the cache has 2c blocks. If the cache uses the set associative mapping scheme with 2 blocks per set, then the block k of main memory maps to the set: **(Gate-1999) (1 Marks)**

(A) (k mod m)of the cache

(B) (k mod c)of the cache

(C) (k mod 2c)of the cache

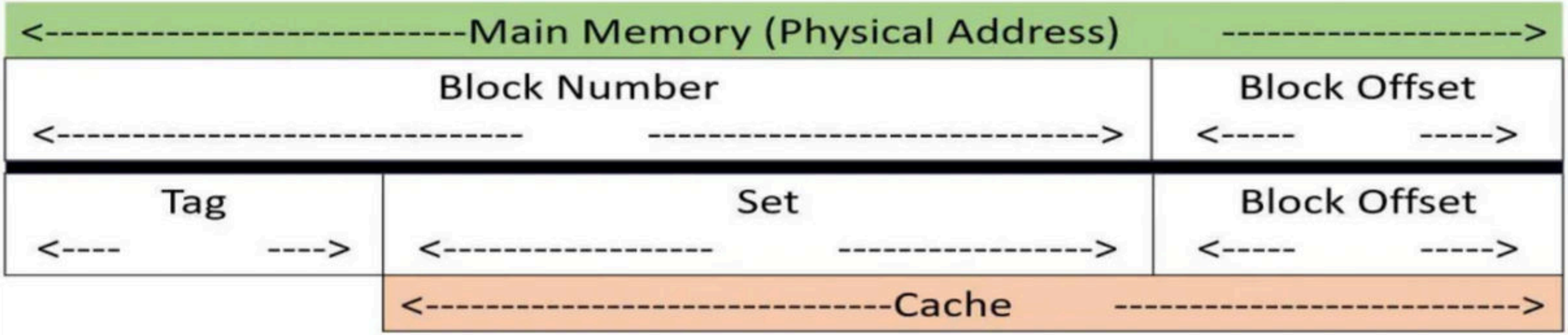
(D) (k mod 2cm)of the cache



Q A block-set associative cache memory consists of 128 blocks divided into four block sets. The main memory consists of 16,384 blocks and each block contains 256 eight-bit words. **(Gate-1990) (2 Marks)**

i) How many bits are required for addressing the main memory?

ii) How many bits are needed to represent the TAG, SET and WORD fields?



Q In designing a computer's cache system, the cache block (or cache line) size is an important parameter. Which one of the following statements is correct in this context? **(Gate-2014) (1 Marks)**

(A) A smaller block size implies better spatial locality

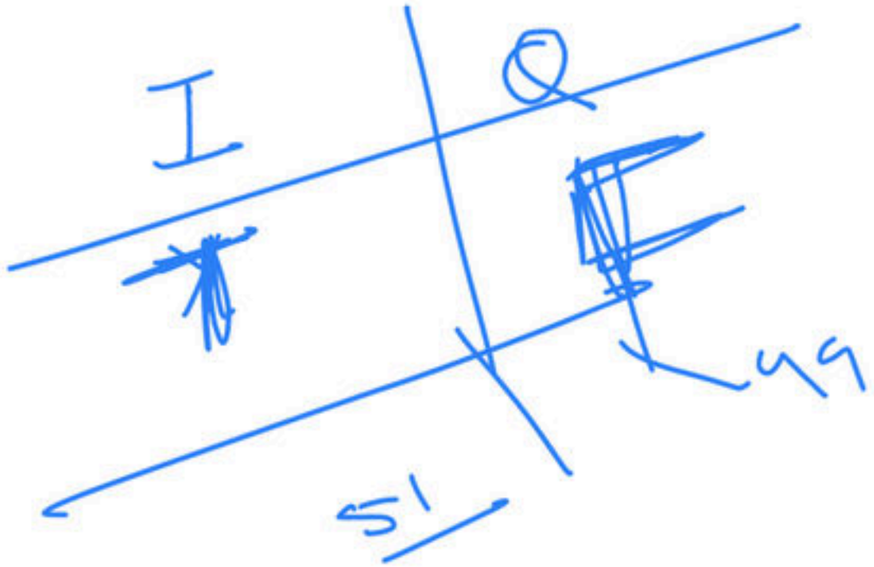
(B) A smaller block size implies a smaller cache tag and hence lower cache tag overhead

(C) A smaller block size implies a larger cache tag and hence lower cache hit time

(D) A smaller block size incurs a lower cache miss penalty

Cache Replacement Policies

- In direct mapped cache, the position of each block is predetermined hence no replacement policy exists.
- In fully associative and set associative caches there exists policies.
- When a new block is brought into the cache and all the positions that it may occupy are full, then the controller needs to decide which of the old blocks it can overwrite.



A.M

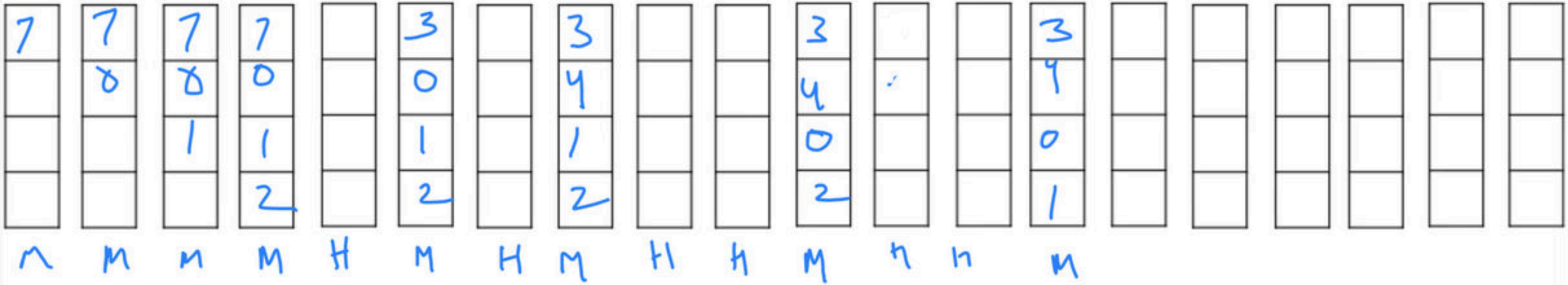
SAM

FIFO Policy — A.M

- The block which have entered first in the memory will be replaced first.
- This can lead to a problem known as "**Belady's Anomaly**", it states that if we increase the number of lines in cache memory the cache miss will increase.

Example: Let the blocks be in the sequence: 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1 and the cache memory has 4 lines.

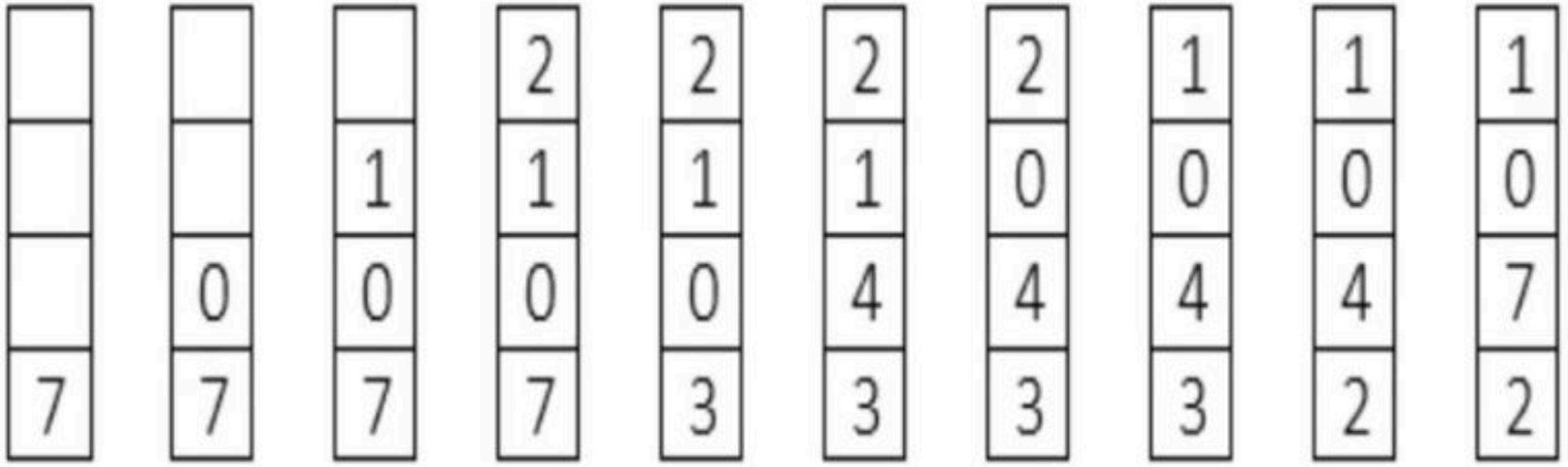
7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1



FIFO Policy

- The block which have entered first in the memory will be replaced first.
- This can lead to a problem known as "**Belady's Anomaly**", it states that if we increase the number of lines in cache memory the cache miss will increase.

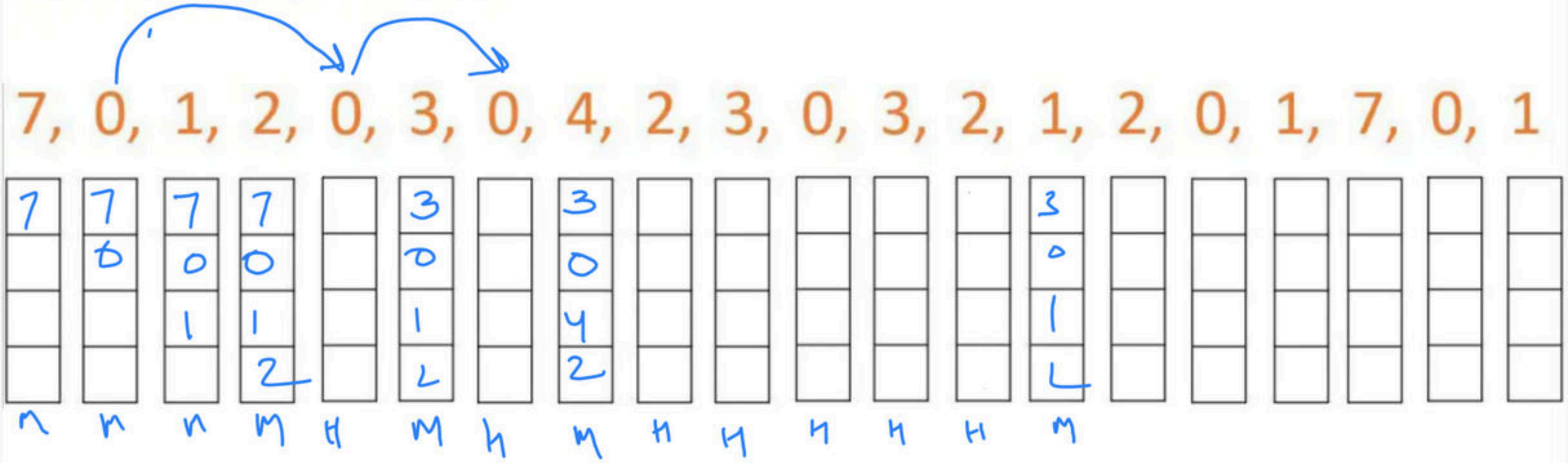
Example: Let the blocks be in the sequence: 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1 and the cache memory has 4 lines.



LRU (Least Recently Used)

- The page which was not used for the longest period of time in the past will get replaced first.

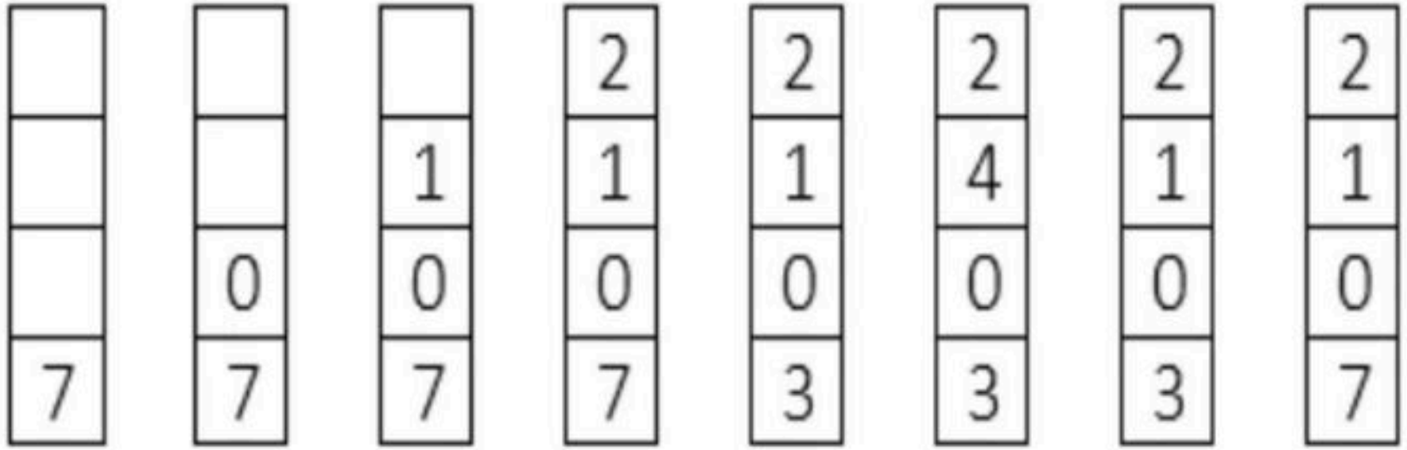
Example: Let the blocks be in the sequence: 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1 and the cache memory has 4 lines.



LRU (Least Recently Used)

- The page which was not used for the longest period of time in the past will get replaced first.

Example: Let the blocks be in the sequence: 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1 and the cache memory has 4 lines.

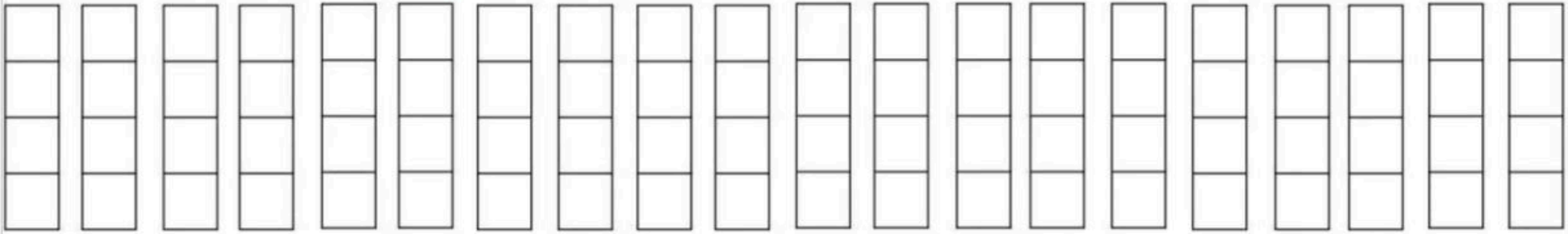


Most Recently Used (MRU)

- The page which was used recently will be replaced first.

Example: Let the blocks be in the sequence: 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1 and the cache memory has 4 lines.

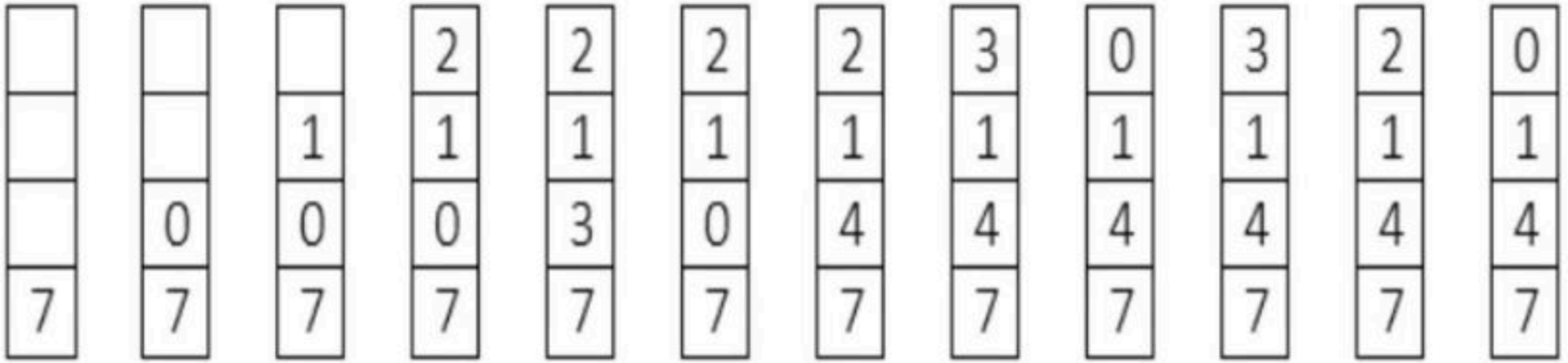
7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1



Most Recently Used (MRU)

- The page which was used recently will be replaced first.

Example: Let the blocks be in the sequence: 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1 and the cache memory has 4 lines.

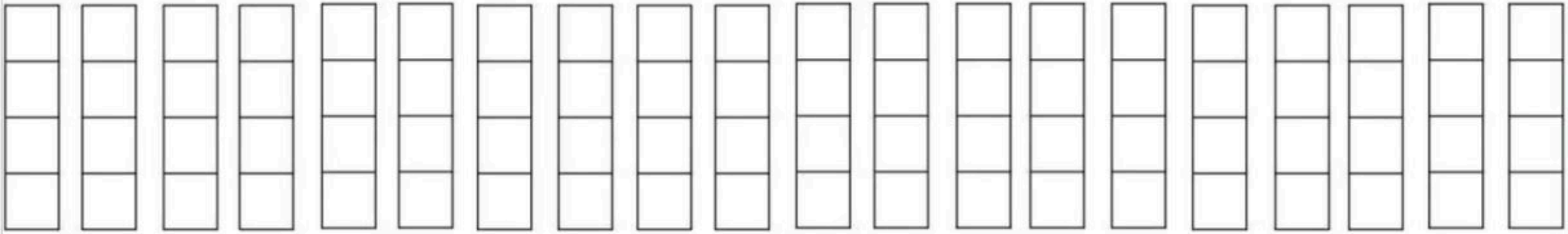


Optimal Algorithm

- The page which will not be used for the longest period of time in future references will be replaced first.
- The optimal algorithm will provide the best performance but it is difficult to implement as it requires the future knowledge of pages which is not possible.
- It is used as a benchmark for cache replacement algorithms.

Example: Let the blocks be in the sequence: 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1 and the cache memory has 4 lines.

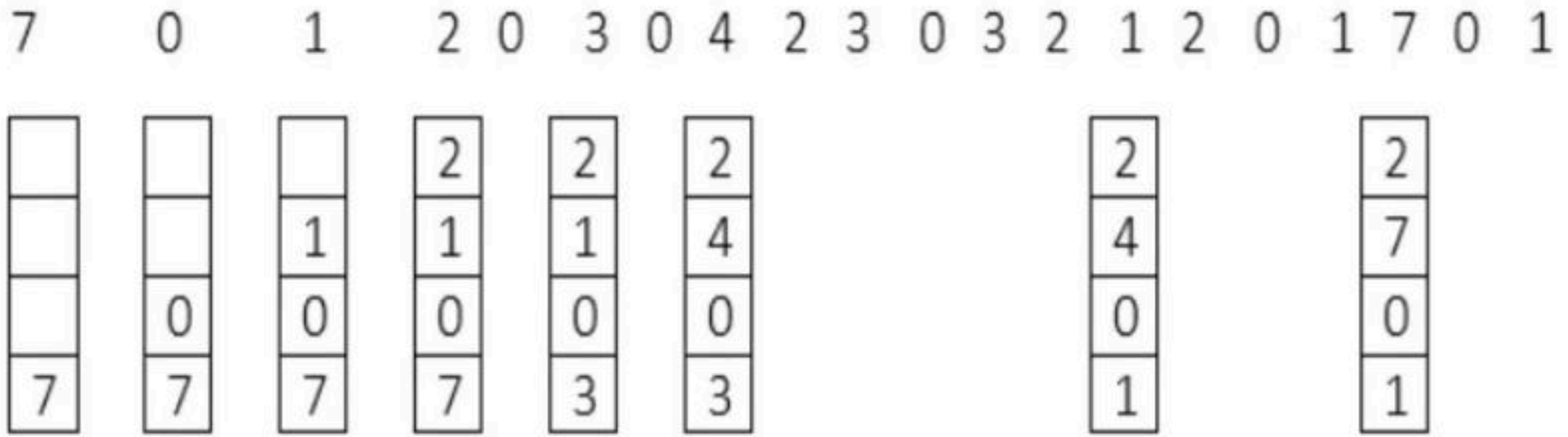
7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1



Optimal Algorithm

- The page which will not be used for the longest period of time in future references will be replaced first.
- The optimal algorithm will provide the best performance but it is difficult to implement as it requires the future knowledge of pages which is not possible.
- It is used as a benchmark for cache replacement algorithms.

Example: Let the blocks be in the sequence: 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1 and the cache memory has 4 lines.



Break

Types of Miss

- **Compulsory Miss**
  - When CPU demands for any block for the first time then definitely a miss is going to occur as the block needs to be brought into the cache, it is known as Compulsory miss.
- **Capacity Miss**
  - Occur because blocks are being discarded from cache because cache cannot contain all blocks needed for program execution.
- **Conflict Miss**
  - In the case of set associative or direct mapped block placement strategies, conflict misses occur when several blocks are mapped to the same set or block frame; also called collision misses or interference misses.

Q Consider a 2-way set associative cache with 256 blocks and uses LRU replacement, Initially the cache is empty. Conflict misses are those misses which occur due the contention of multiple blocks for the same cache set. Compulsory misses occur due to first time access to the block. The following sequence of accesses to memory blocks
(0, 128, 256, 128, 0, 128, 256, 128, 1, 129, 257, 129, 1, 129, 257, 129) is repeated 10 times. The number of conflict misses experienced by the cache is _____. **(Gate-2017) (2 Marks)**

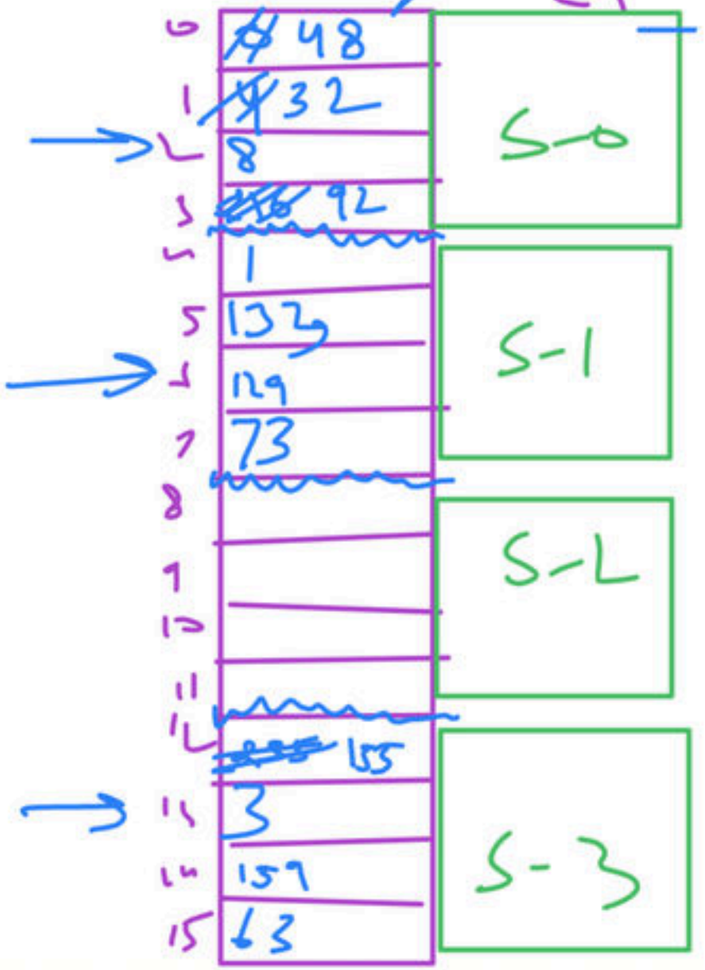
Q Consider a 4-way set associative cache (initially empty) with total 16 cache blocks. The main memory consists of 256 blocks and the request for memory blocks is in the following order: 0, 255, 1, 4, 3, 8, 133, 159, 216, 129, 63, 8, 48, 32, 73, 92, 155. Which one of the following memory blocks will NOT be in cache if LRU replacement policy is used? **(Gate-2009) (2 Marks)**

(A) 3

(B) 8

(C) 129

(D) 216



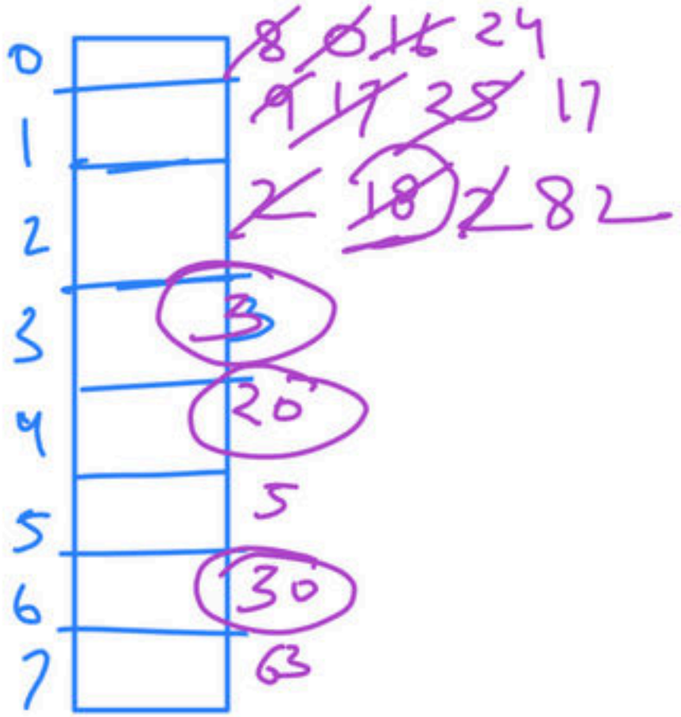
Q Consider a Direct Mapped Cache with 8 cache blocks (numbered 0-7). If the memory block requests are in the following order 3, 5, 2, 8, 0, 63, 9,16, 20, 17, 25, 18, 30, 24, 2, 63, 5, 82,17, 24. Which of the following memory blocks will not be in the cache at the end of the sequence? **(GATE-2007) (2 Marks)**

(A) 3 **(B)** 18 **(C)** 20 **(D)** 30

Q Consider a Direct Mapped Cache with 8 cache blocks (numbered 0-7). If the memory block requests are in the following order 3, 5, 2, 8, 0, 63, 9,16, 20, 17, 25, 18, 30, 24, 2, 63, 5, 82,17, 24. Which of the following memory blocks will not be in the cache at the end of the sequence? **(GATE-2007) (2 Marks)**

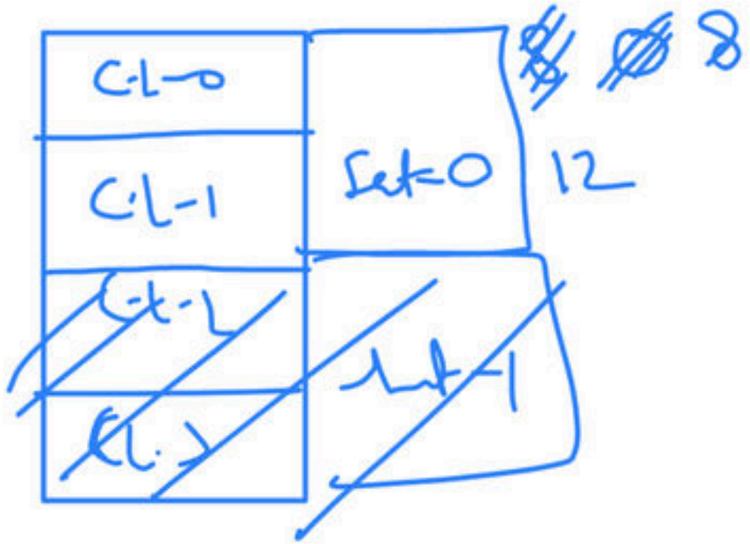
(A) 3 **(B)** 18 **(C)** 20 **(D)** 30

3, 5, 2, 8, 0, 63, 9, 16, 20, 17, 25, 18, 30, 24, 2, 63, 5, 82, 17, 24



Q Consider a small two-way set-associative cache memory, consisting of four blocks. For choosing the block to be replaced, use the least recently used (LRU) scheme. The number of cache misses for the following sequence of block addresses is 8, 12, 0, 12, 8 **(Gate-2004) (2 Marks)**

(A) 2 **(B)** 3 **(C)** 4 **(D)** 5



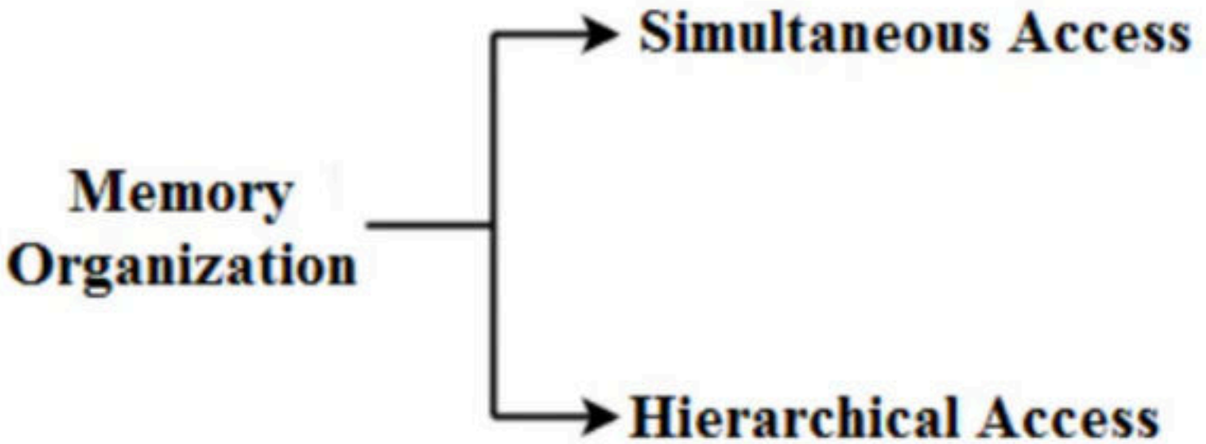
8, 12, 0, 12, 8

M, M, H, H, M

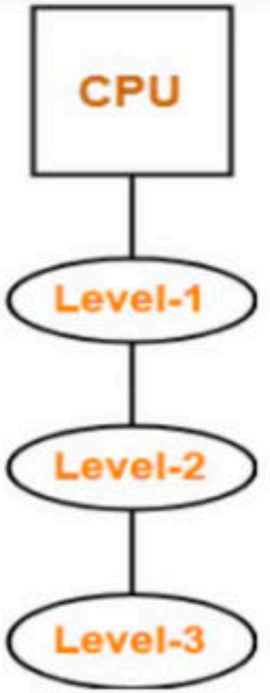
Break

Memory Organization

- Memory is organized at different levels.
- CPU may try to access different levels of memory in different ways.
- On this basis, the memory organization is broadly divided into two types



Hierarchical Access Memory Organization



- In this memory organization, memory levels are organized as
  - Level-1 is directly connected to the CPU.
  - Level-2 is directly connected to level-1.
  - Level-3 is directly connected to level-2 and so on

- Whenever CPU requires any word,
  - It first searches for the word in level-1.
  - If the required word is not found in level-1, it searches for the word in level-2.
  - If the required word is not found in level-2, it searches for the word in level-3 and so on.

• T_1 = Access time of level L_1 • S_1 = Size of level L_1 • C_1 = Cost per byte of level L_1 • H_1 = Hit rate of level L_1	• T_2 = Access time of level L_2 • S_2 = Size of level L_2 • C_2 = Cost per byte of level L_2 • H_2 = Hit rate of level L_2	• T_3 = Access time of level L_3 • S_3 = Size of level L_3 • C_3 = Cost per byte of level L_3 • H_3 = Hit rate of level L_3

Effective Memory Access Time

Average time required to access memory per operation =

$$H_1 * T_1 + (1 - H_1) * H_2 * (T_1 + T_2) + (1 - H_1)\,(1 - H_2) * H_3 * (T_1 + T_2 + T_3)$$

$$H_1 * T_1 + (1 - H_1) * H_2 * (T_1 + T_2) + (1 - H_1)\,(1 - H_2) * (T_1 + T_2 + T_3)$$

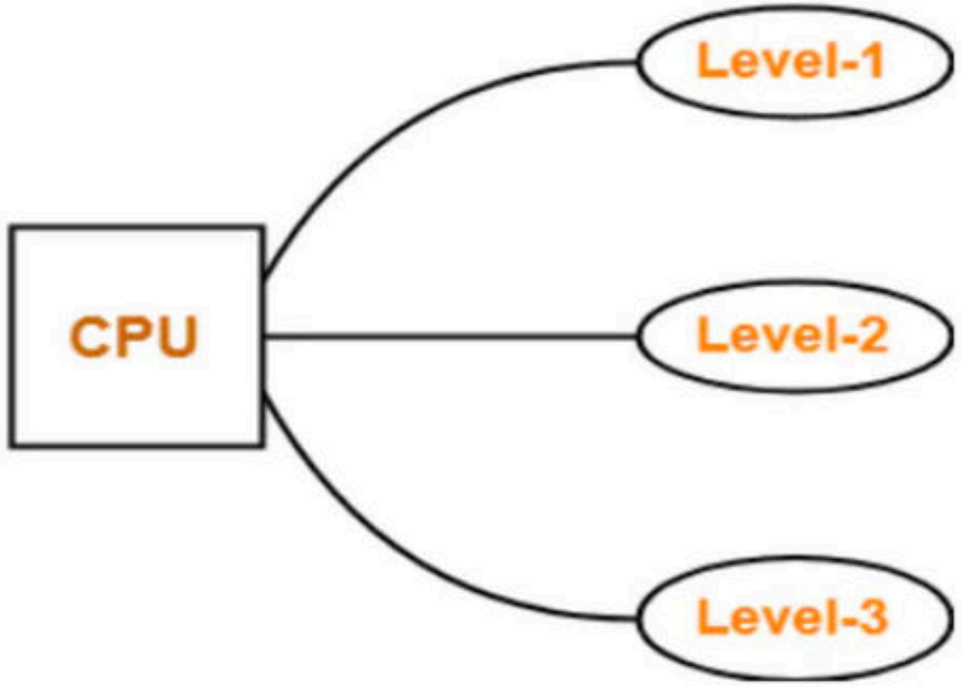
Average Cost per Byte

- Average cost per byte of the memory =

$$\frac{C_1S_1 + C_2S_2 + C_3S_3}{S_1 + S_2 + S_3}$$

Simultaneous Access Memory Organization

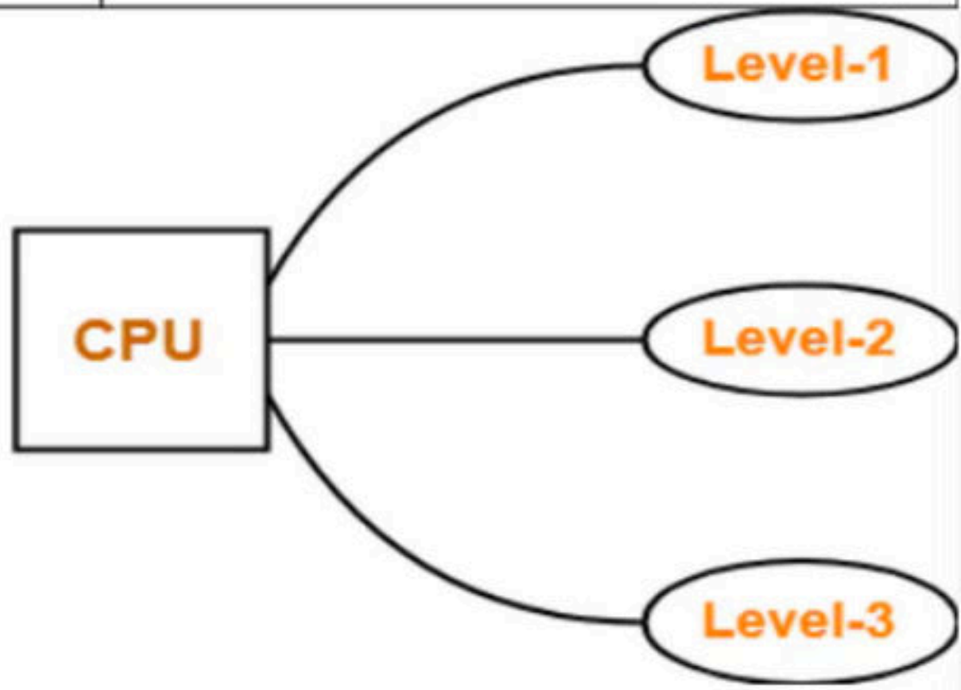
- In simultaneous access all the levels of memory are directly connected to the CPU, whenever CPU requires any word, it starts searching for it in all the levels simultaneously.



• T_1 = Access time of level L_1 • S_1 = Size of level L_1 • C_1 = Cost per byte of level L_1 • H_1 = Hit rate of level L_1	• T_2 = Access time of level L_2 • S_2 = Size of level L_2 • C_2 = Cost per byte of level L_2 • H_2 = Hit rate of level L_2	• T_3 = Access time of level L_3 • S_3 = Size of level L_3 • C_3 = Cost per byte of level L_3 • H_3 = Hit rate of level L_3

Effective Memory Access Time (EMAT) =

$$H_1 * T_1 + (1 - H_1) * H_2 * T_2 + (1 - H_1)(1 - H_2) * H_3 * T_3$$



In any memory organization, the data item being searched will definitely be present in the last level (or secondary memory).

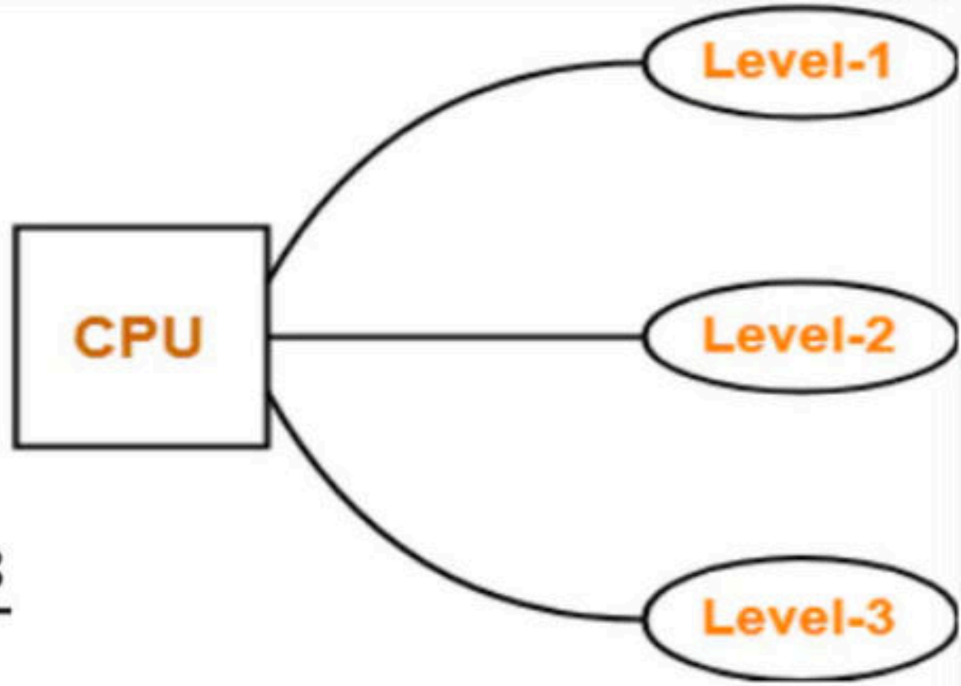
Thus, hit rate for the last level is always 1. So,

$$H_1 * T_1 + (1 - H_1) * H_2 * T_2 + (1 - H_1)(1 - H_2) * T_3$$

Average Cost per Byte

- Average cost per byte of the memory =

$$\frac{C_1S_1 + C_2S_2 + C_3S_3}{S_1 + S_2 + S_3}$$



Example: Calculate the EMAT for a machine with a cache hit rate of 80% where cache access time is 5ns and main memory access time is 100ns, both for simultaneous and hierarchical access.

Q Assume that for a certain processor, a read request takes 50 nanoseconds on a cache miss and 5 nanoseconds on a cache hit. Suppose while running a program, it was observed that 80% of the processor's read requests result in a cache hit. The average read access time in nanoseconds is_____. **(GATE-2015) (2 Marks)**

Q Consider a system with 2 level cache. Access times of Level 1 cache, Level 2 cache and main memory are 0.5 ns, 5 ns and 100 ns respectively. The hit rates of Level 1 and Level 2 caches are 0.7 and 0.8 respectively. What is the average access time of the system ignoring the search time within the cache? **(NET-DEC-2018)**

a) 35.20 ns **b)** 7.55 ns **c)** 20.75 ns **d)** 24.35 ns

Cache Coherence Problem

- If multiple copy of same data is maintained at different level of memories then inconsistency may occur, this problem is known as cache coherence problem.

- Cache coherence problem can be resolved using the following techniques:
  - Write Through
  - Write Back

Write Through

- Write through is used to maintain the consistency between the cache and main memory.
- According to it if the cache copy is updated, at the same time main memory is also updated.

- **Advantages**
  - It provides the highest level of consistency.

- **Disadvantages**
  - It requires more number of memory access.

Write Back

- Write back is also used to maintain the consistency between the cache and main memory.
- According to it all the changes performed on cache are reflected back to the main memory in the end.

- **Advantage**
  - Less number of memory accesses and less write operations.

- **Disadvantage**
  - Inconsistency may occur.

Q In _____ method, the word is written to the block in both the cache and main memory, in parallel. **(NET-JULY-2016)**

(a) Write through

(b) Write back

(c) Write protected

(d) Direct mapping

Q A hierarchical memory system that uses cache memory has cache access time of 50 nano seconds, main memory access time of 300 nano seconds, 75% of memory requests are for read, hit ratio of 0.8 for read access and the write-through scheme is used. What will be the average access time of the system both for read and write requests? **(NET-DEC-2014)**

(A) 157.5 n.sec **(B)** 110 n.sec **C)** 75 n.sec **(D)** 82.5 n.sec

Q For inclusion to hold between two cache levels L_1 and L_2 in a multi-level cache hierarchy, which of the following are necessary?
(Gate-2008) (2 Marks)

I. L_1 must be a write-through cache

II. L_2 must be a write-through cache

III. The associativity of L_2 must be greater than that of L1

IV. The L_2 cache must be at least as large as the L1 cache

(A) IV only
(B) I and IV only
(C) I, III and IV only
(D) I, II, III and IV

Q In a two-level cache system, the access times of L_1 and L_2 is 1 and 8 clock cycles, respectively. The miss penalty from the L_2 cache to main memory is 18 clock cycles. The miss rate of L_1 cache is twice that of L_2. The average memory access time (AMAT) of this cache system is 2 cycles. The miss rates of L_1 and L_2 respectively are: **(Gate-2017) (2 Marks)**

(A) 0.111 and 0.056

(B) 0.056 and 0.111

(C) 0.0892 and 0.1784

(D) 0.1784 and 0.0892

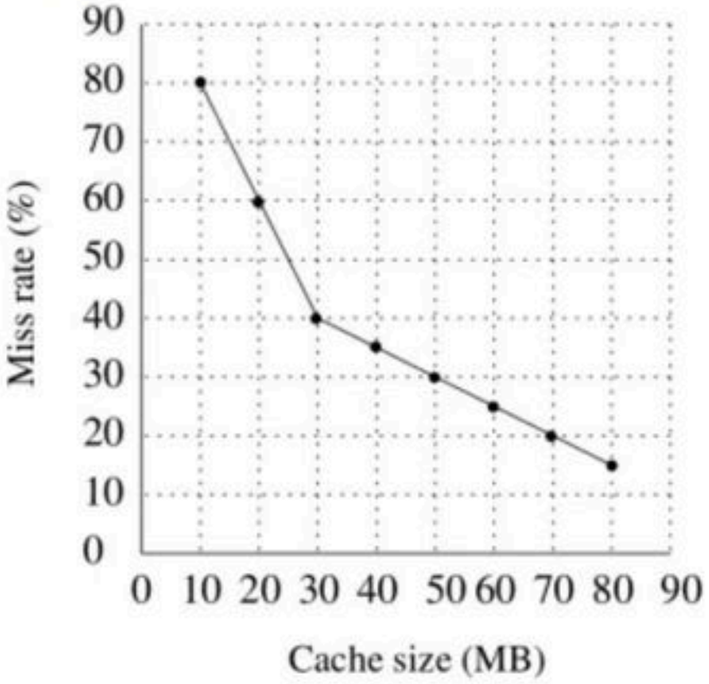
Q The read access times and the hit ratios for different caches in a memory hierarchy are as given below:

Cache	Read access time (in nanoseconds)	Hit ratio
I-cache	2	0.8
D-cache	2	0.9
L2-cache	8	0.9

The read access time of main memory in 90 nanoseconds. Assume that the caches use the referred-word-first read policy and the writeback policy. Assume that all the caches are direct mapped caches. Assume that the dirty bit is always 0 for all the blocks in the caches. In execution of a program, 60% of memory reads are for instruction fetch and 40% are for memory operand fetch. The average read access time in nanoseconds (up to 2 decimal places) is _____ **(Gate-2017) (2 Marks)**

Q Consider a two-level cache hierarchy L_1 and L_2 caches. An application incurs 1.4 memory accesses per instruction on average. For this application, the miss rate of L_1 cache 0.1, the L_2 cache experience on average. 7 misses per 1000 instructions. The miss rate of L_2 expressed correct to two decimal places is _____. **(Gate-2017) (1 Marks)**

Q A file system uses an in-memory cache to cache disk blocks. The miss rate of the cache is shown in the figure. The latency to read a block from the cache is 1 ms and to read a block from the disk is 10 ms. Assume that the cost of checking whether a block exists in the cache is negligible. Available cache sizes are in multiples of 10 MB.



The smallest cache size required to ensure an average read latency of less than 6 ms is _____ MB. **(Gate-2016) (2 Marks)**

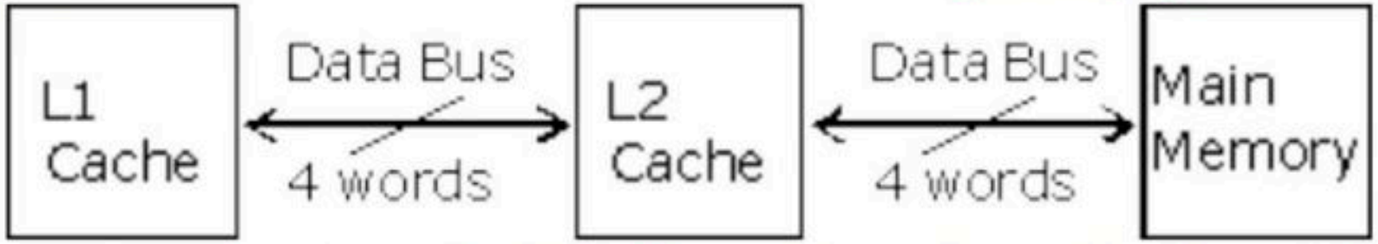
Q The memory access time is 1 nanosecond for a read operation with a hit in cache, 5 nanoseconds for a read operation with a miss in cache, 2 nanoseconds for a write operation with a hit in cache and 10 nanoseconds for a write operation with a miss in cache. Execution of a sequence of instructions involves 100 instruction fetch operations, 60 memory operand read operations and 40 memory operands write operations. The cache hit-ratio is 0.9. The average memory access time (in nanoseconds) in executing the sequence of instructions is _____. **(Gate-2014) (2 Marks)**

Q A computer system has an L_1 cache, an L_2 cache, and a main memory unit connected as shown below. The block size in L_1 cache is 4 words. The block size in L_2 cache is 16 words. The memory access times are 2 nanoseconds. 20 nanoseconds and 200 nanoseconds for L_1 cache, L_2 cache and main memory unit respectively.

When there is a miss in L_1 cache and a hit in L_2 cache, a block is transferred from L_2 cache to L_1 cache. What is the time taken for this transfer? **(Gate-2010) (2 Marks)**

(A) 2 nanoseconds
(B) 20 nanoseconds
(C) 22 nanoseconds
(D) 88 nanoseconds

Q A computer system has an L_1 cache, an L_2 cache, and a main memory unit connected as shown below. The block size in L_1 cache is 4 words. The block size in L_2 cache is 16 words. The memory access times are 2 nanoseconds. 20 nanoseconds and 200 nanoseconds for L_1 cache, L_2 cache and main memory unit respectively.



Q When there is a miss in both L_1 cache and L_2 cache, first a block is transferred from main memory to L_2 cache, and then a block is transferred from L_2 cache to L_1 cache. What is the total time taken for these transfers? **(Gate-2010) (2 Marks)**

(A) 222 nanoseconds
(B) 888 nanoseconds
(C) 902 nanoseconds
(D) 968 nanoseconds

Q Consider a system with 2 level caches. Access times of $Level_1$ cache, $Level_2$ cache and main memory are 1 ns, 10ns, and 500 ns, respectively. The hit rates of $Level_1$ and $Level_2$ caches are 0.8 and 0.9, respectively. What is the average access time of the system ignoring the search time within the cache? **(Gate-2004) (1 Marks)**

(A) 13.0 ns **(B)** 12.8 ns **(C)** 12.6 ns **(D)** 12.4 ns

Q A dynamic RAM has a memory cycle time of 64 nsec. It has to be refreshed 100 times per msec and each refresh takes 100 nsec. What percentage of the memory cycle time is used for refreshing? **(Gate-2005) (1 Marks)**

(A) 10 **(B)** 6.4 **(C)** 1 **(D)** .64

Q A cache line is 64 bytes. The main memory has latency 32ns and bandwidth 1G.Bytes/s. The time required to fetch the entire cache line from the main memory is **(Gate-2006) (2 Marks)**

(A) 32 ns **(B)** 64 ns **(C)** 96 ns **(D)** 128 ns

Q.29 Assume a two-level inclusive cache hierarchy. L1 and L2, where L2 is the larger of the two. Consider the following statements.

S_1 : Read misses in a write through L1 cache do not result in writebacks of dirty lines to the L2.

S_2 : Write allocate policy must be used in conjunction with write through caches and no-write allocate policy is "used with writeback caches.

(GATE- 2021)

Which of the following statements is correct?

(a) S_1 is true and S_2 is true.

(b) S_1 is true and S_2 is false.

(c) S_1 is false and S_2 is true.

(d) S_1 is false and S_2 is false.