Cache Implementation
LAB EXERCISE - 7

—

CED17I017

Specifications

Direct mapping L1 cache (split cache)

- Main memory
  - Size: 128 KB
  - 8 words/block
- Data cache
  - Size: 32 KB
  - 8 words/line
  - 1 read port and 1 write port
- Instruction cache
  - Size: 16 KB
  - 8 words/line
  - 1 read port

- 1 word: 2 Bytes
- Use Valid and Dirty bits in the cache
- I have used Write Back with write allocate policy

Calculations

- **Main memory**

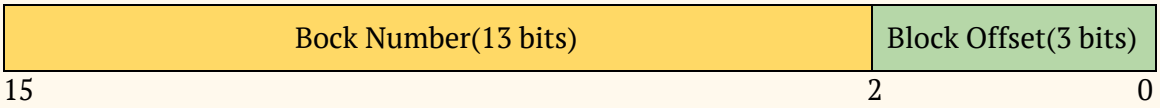
$$Main\ Memory\ Size = 128\ KB = 2^{16}\ words$$

$$Block\ Size = 8\ words$$

$$No.\ of\ Blocks = \frac{Main\ Memory\ Size}{Block\ Size} = \frac{2^{16}}{8} = 2^{13}$$

$$No.\ of\ bits\ required\ for\ Block\ no. = \log_2(No.\ of\ Blocks) = \log_2(2^{13}) = 13\ bits$$

$$Block\ offset = \log_2(Block\ Size) = \log_2(8) = 3\ bits$$



- **Data Cache**

$$Data\ Cache\ Size = 32\ KB = 2^{14}\ words$$

$$Block\ Size = 8\ words$$

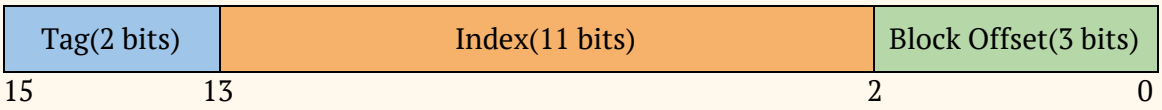
$$No.\ of\ Lines = \frac{Data\ Cache\ Size}{Block\ Size} = \frac{2^{14}}{8} = 2^{11}$$

$$No.\ of\ bits\ required\ for\ Line\ no.(Index) = \log_2(No.\ of\ Lines) = \log_2(2^{11}) = 11\ bits$$

$$Block\ offset = 3\ bits$$

$$No\ of\ bits\ in\ Block\ no. = Tag + Index$$

$$Tag = No\ of\ bits\ in\ Block\ no. - Index = 13 - 11 = 2\ bits$$



- **Instruction Cache**

$Instruction\ Cache\ Size = 16\ KB = 2^{13}\ words$

$Block\ Size = 8\ words$

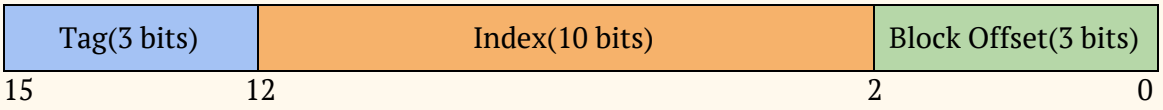
$No.\ of\ Lines = \frac{Instruction\ Cache\ Size}{Block\ Size} = \frac{2^{13}}{8} = 2^{10}$

$No.\ of\ bits\ required\ for\ Line\ no.(Index) = \log_2(No.\ of\ Lines) = \log_2(2^{10}) = 10\ bits$

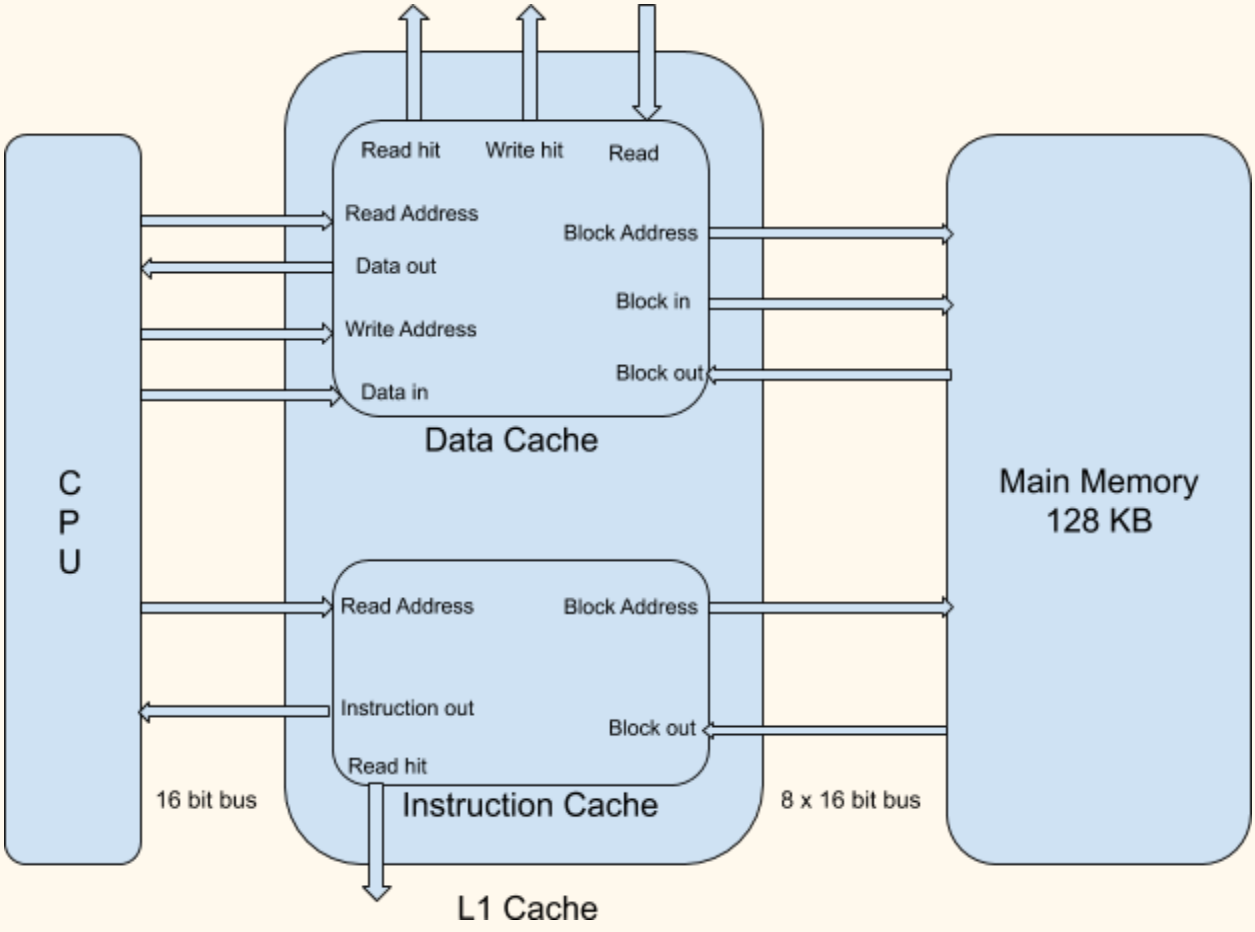
$Block\ offset = 3\ bits$

$No\ of\ bits\ in\ Block\ no. = Tag + Index$

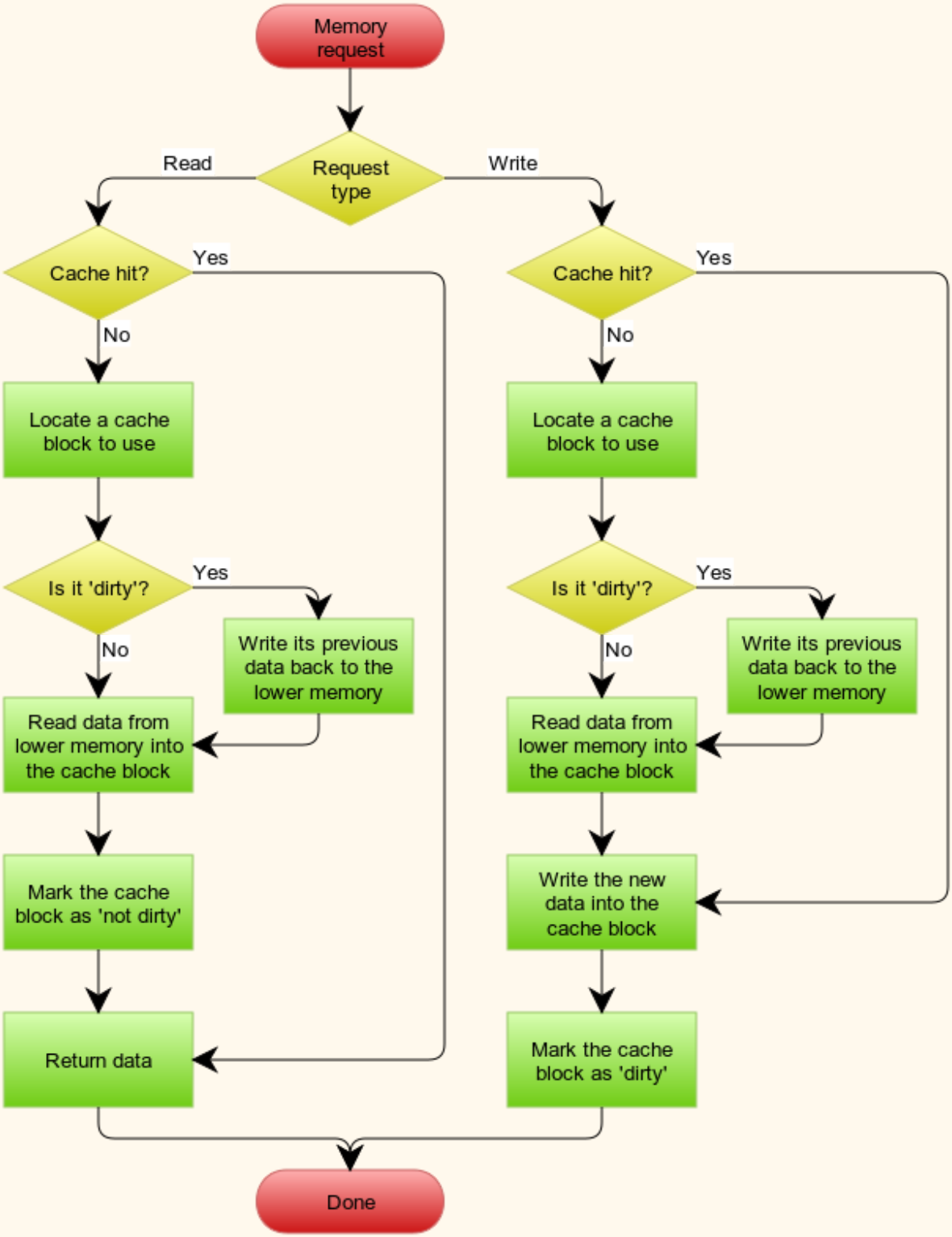
$Tag = No\ of\ bits\ in\ Block\ no. - Index = 13 - 10 = 3\ bits$



Memory Diagram



Flow chart of write-back cache with write allocation(used for data cache)



Code Explanation

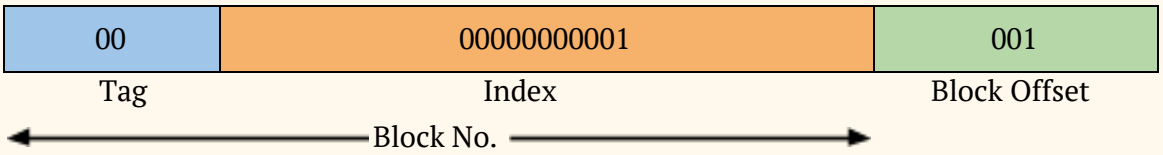
The code has thoroughly been explained by comments in the file cache.v

Output Explanation

```
(base) firoz@320-15ikb:~/Desktop/faltu/firoz/sem_6/Computer Arch../lab_7$ iverilog tb.v
(base) firoz@320-15ikb:~/Desktop/faltu/firoz/sem_6/Computer Arch../lab_7$ ./a.out
10 dc_is_read = 1  dc_read_address = 0000000000001001  dc_is_read_hit = 0  dc_data_out = 789a
30 dc_is_read = 0  dc_write_address = 0000000000001001  dc_is_write_hit = 1  dc_data_in = 0000
50 dc_is_read = 1  dc_read_address = 0100000000001001  dc_is_read_hit = 0  dc_data_out = xxxx
70 dc_is_read = 1  dc_read_address = 0000000000001001  dc_is_read_hit = 0  dc_data_out = 0000
--------------------
90 ic_read_address =0000000000010100  ic_is_read_hit = 0  ic_inst_out = abcd
110 ic_read_address =0000000000010100  ic_is_read_hit = 1  ic_inst_out = abcd
--------------------
130 dc_is_read = 1  dc_read_address = 0000000000000101  dc_is_read_hit = 0  dc_data_out = 89ab
  ic_read_address =0000000000011100  ic_is_read_hit = 0  ic_inst_out = 9abc
(base) firoz@320-15ikb:~/Desktop/faltu/firoz/sem_6/Computer Arch../lab_7$ ^C
```

1st input (data cache):

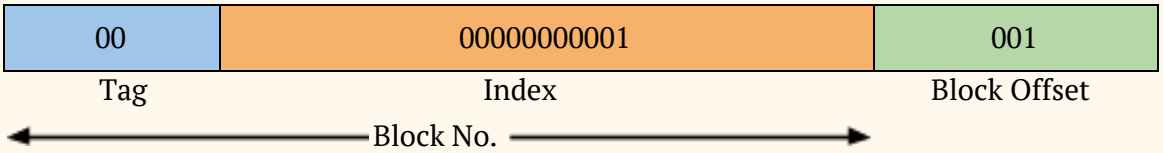
Read mode indicated by dc_is_read=1, Read Address = 0x9 = 00_00000000001_001



Initially, the cache was empty, hence this is a compulsory miss. Therefore a read miss indicated by dc_is_read_hit=0. The data present in the requested location is 789a.

2nd input (data cache):

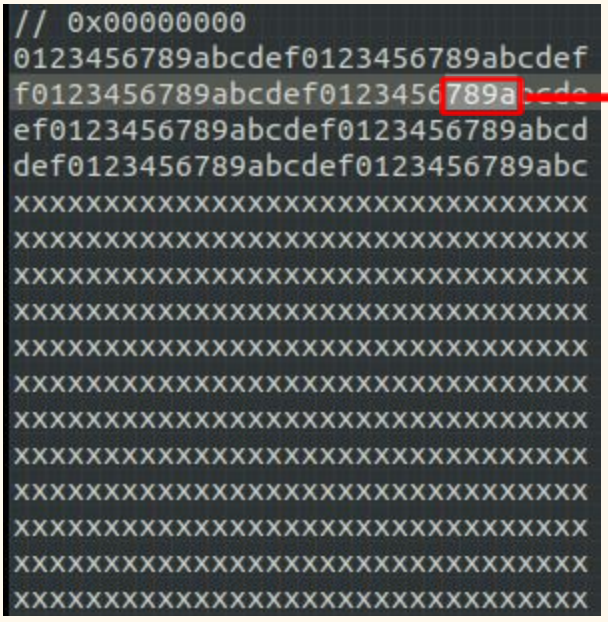
Write mode indicated by dc_is_read=0, Write Address = 0x9 = 00_00000000001_001



Data to be written = 0x0000

This will be a write hit as the block is present in the cache indicated by dc_is_write_hit = 1.

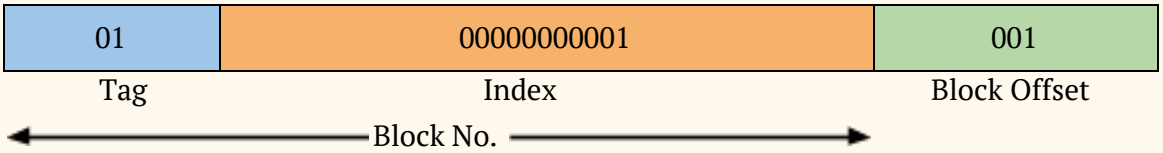
This update will only be there in the cache and the dirty bit would be set, the update wouldn't be reflected in the main memory till the block is replaced in the cache.



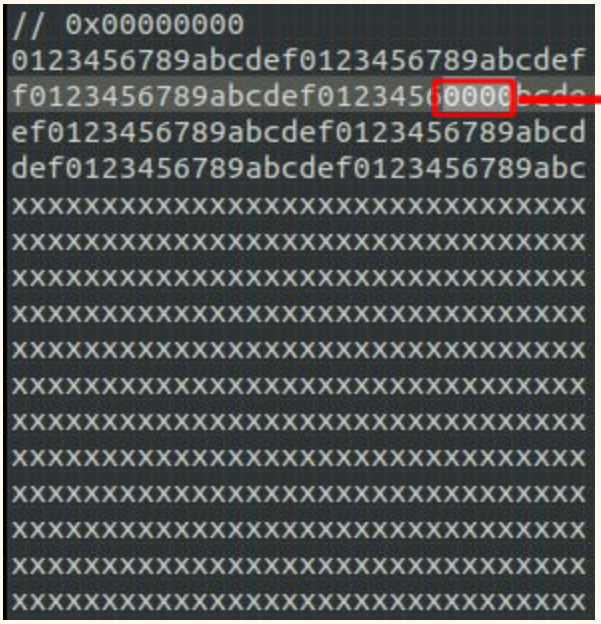
No change as the block is not replaced yet

3rd input (data cache):

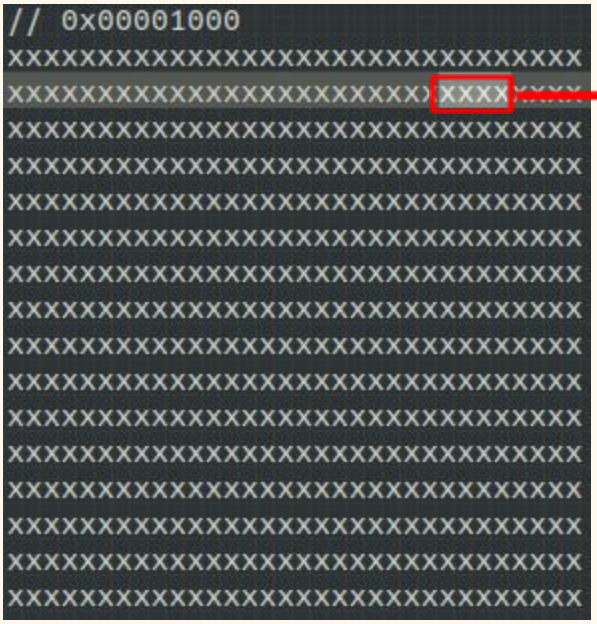
Read mode indicated by dc_is_read=1, Read Address = 0x4009 = 01_00000000001_001



As the block containing 0x4009 address is directly mapped to the same line as that of the block containing the address 0x9, but the tags are different hence it will be a read miss indicated by is_read_hit = 0. As the block in cache is dirty, it will be written into the main memory and then the new block will be loaded into the cache. The data present in the requested location is xxxx.



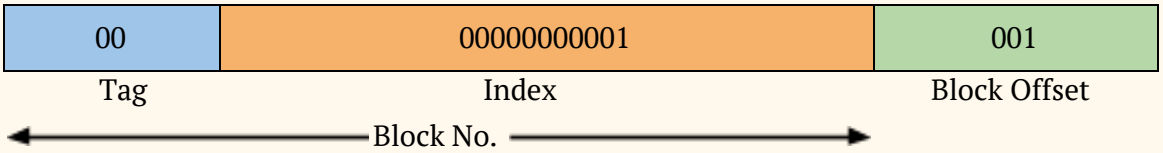
Data updated in main memory



Block No. 4097 Word No. 1

4th input (data cache):

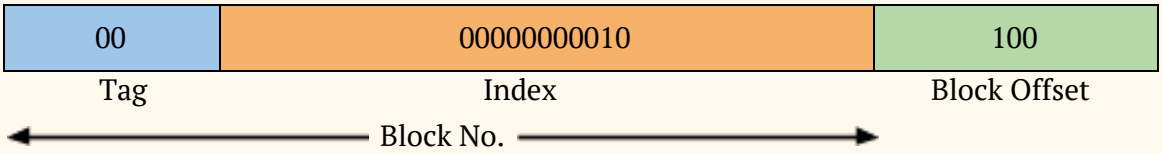
Read mode indicated by dc_is_read=1, Read Address = 0x0009 = 00_00000000001_001



As the block containing 0x4009 address is directly mapped to the same line as that of the block containing the address 0x9, but the tags are different hence it will be a read miss indicated by is_read_hit = 0. The data present in the requested location is 0000.

5th input (instruction cache):

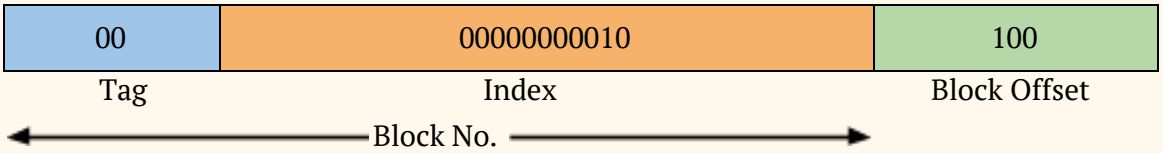
Read mode indicated by ic_is_read=1, Read Address = 0x0014 = 00_00000000010_100



Initially, the cache was empty, hence this is a compulsory miss. Therefore a read miss indicated by ic_is_read_hit=0. The data present in the requested location is abcd.

6th input (instruction cache):

Read mode indicated by ic_is_read=1, Read Address = 0x0014 = 00_00000000010_100



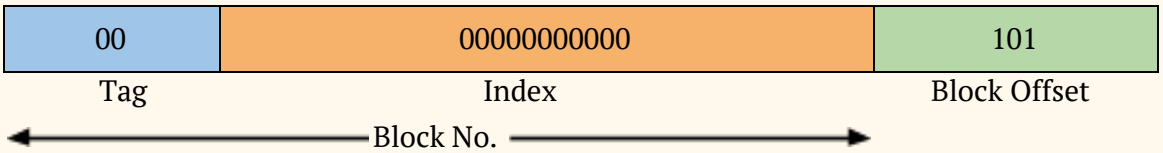
It will be a read hit as the block is already present in the cache indicated by ic_read_hit = 1. The data present in the requested location is abcd.

7th input (both data and instruction cache):

Here instruction and data access can occur in parallel without contention. This improves performance in pipelined processors because instruction and data accesses can occur in the same cycle in different stages of the pipeline.

For data cache:

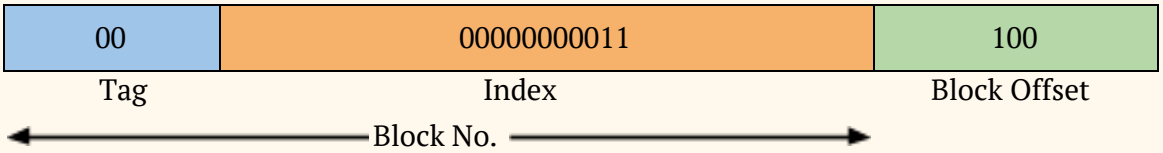
Read mode indicated by dc_is_read=1, Read Address = 0x0005 = 00_00000000000_101



This will be a read miss indicated by ic_read_hit=0 as the tag is not matched because this cache line is being accessed for the first time. The requested data will be fetched from main memory to cache. The requested data is 89ab.

For instruction cache:

Read mode indicated by ic_is_read=1, Read Address = 0x00c = 00_00000000011_100



This will be also a read miss indicated by ic_read_hit=0 as the tag is not matched because this cache line is being accessed for the first time. The requested data will be fetched from main memory to cache. The requested data is 9abc.