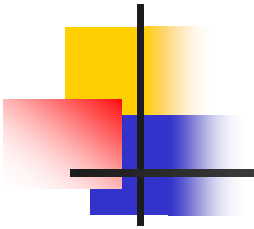


Computer Architecture
Lecture 15

8 bytes cache

64 bytes main mem.

6 bits main mem address

Tag	Index	Byte Offset

Block size 2 bytes

Cache rows=

CacheSize/(Associativity*BlockSize)

IndexBits=log2(CacheRows)

4 rows

2 bits index

CPU calls for addresses:

0, 2,15,9, 12, 25, 31, 8,

22, 26, 63, so on..

To find out which mem. block gets mapped to which cache row/set...

Expand the CPU address in binary...

0--> 000000

Tag	Index	Byte Offset
3 bits	2 bits	1 bit

For next CPU address (2)... expanding it in 6 bits (6 bcz, CPU address is of 6-bits)...

000 01 0

Tag, Index , Byte offset

so, the index bits (01) says, the block containing main mem. address (2) goes to Index (01) in the cache with Tag 000

For next CPU address (15)... expanding it again... 001111 gives...

001	11	1
Tag,	Index,	ByteOffset

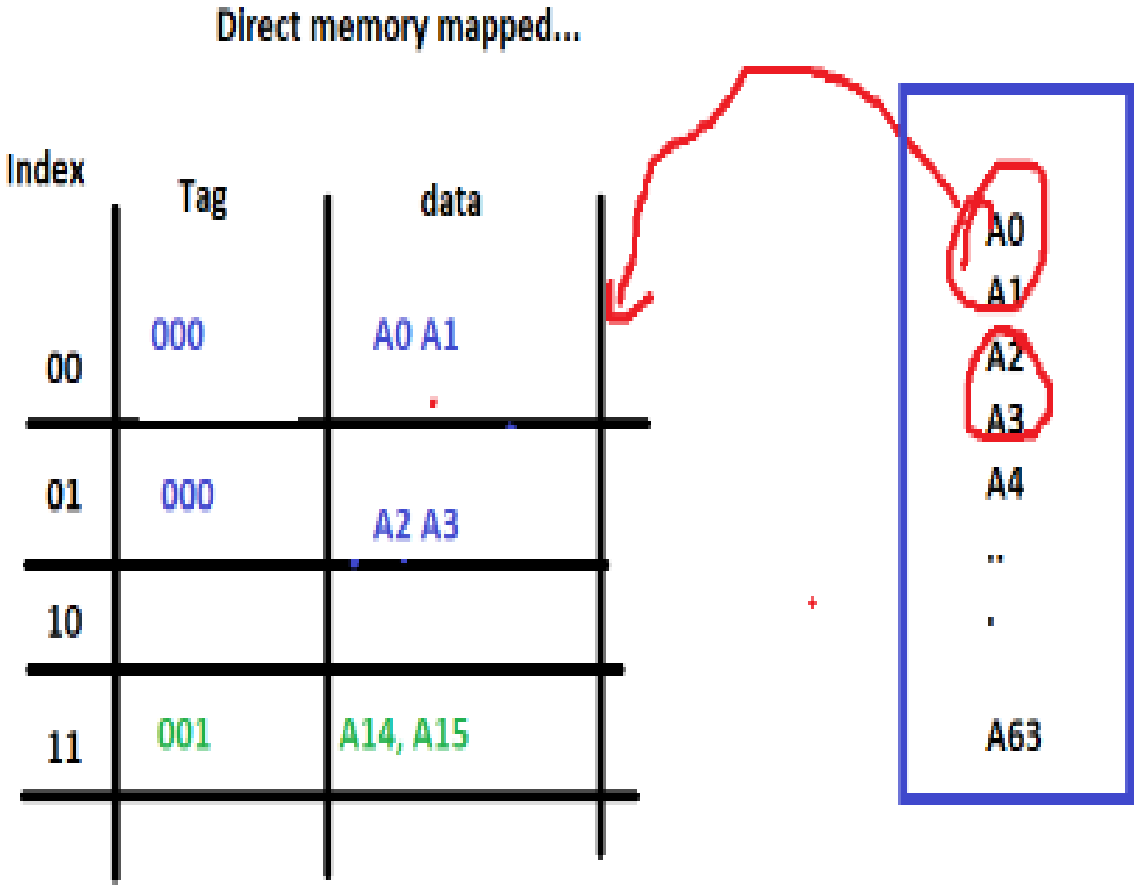
So, the block containg mem. address (15) goes to Index (11) with Tag 001...

Now the question is which mem. location will be fetched with (15)?? either it will be (14, 15) or (15, 16)??

method1: Calculate it from start with 2 block size of two bytes... first block (0,1), (2,3), (4,5),... (14,15),... and (62, 63) the last block..

method2: Write the number in binary 001111... ignore byte offset bits.., remaining number is 00111 which is decimal 7??

multiplying this decimal 7 with block size will give you starting address of block... 7* 2= 14.. the block containg (15) starts from address (14)..


Direct memory mapped...
Index
Tag
data
00
000
A0 A1
01
000
A2 A3
10
11
001
A14, A15
A0
A1
A2
A3
A4
..
.
A63

Consider the main memory of 32 byte size. Here is the series of address references generated sequentially by CPU are: 0,1,2,3,4,5,6,7,0,1,2,8,10,15,12. Assuming cache size is 8 bytes, initially empty, and Block size is 1-byte, Fill the data in Cache lines and calculate the Hit Rate, for the fully associative cache using Least Recently Used (LRU) replacement policy.

Clearly label the data that is replaced in Cache lines.

Table 1: ***main memory 32 bytes size***

Address	Data
00000	A0
00001	A1
00010	A2
00011	A3
.	
.	
.	
.	
.	.
.	.
.	.
11111	A31

Following is the Table of **Fully Associative Cache.**

Index	Tag	Data
000		
001		
010		
011		
.		
.		
111		