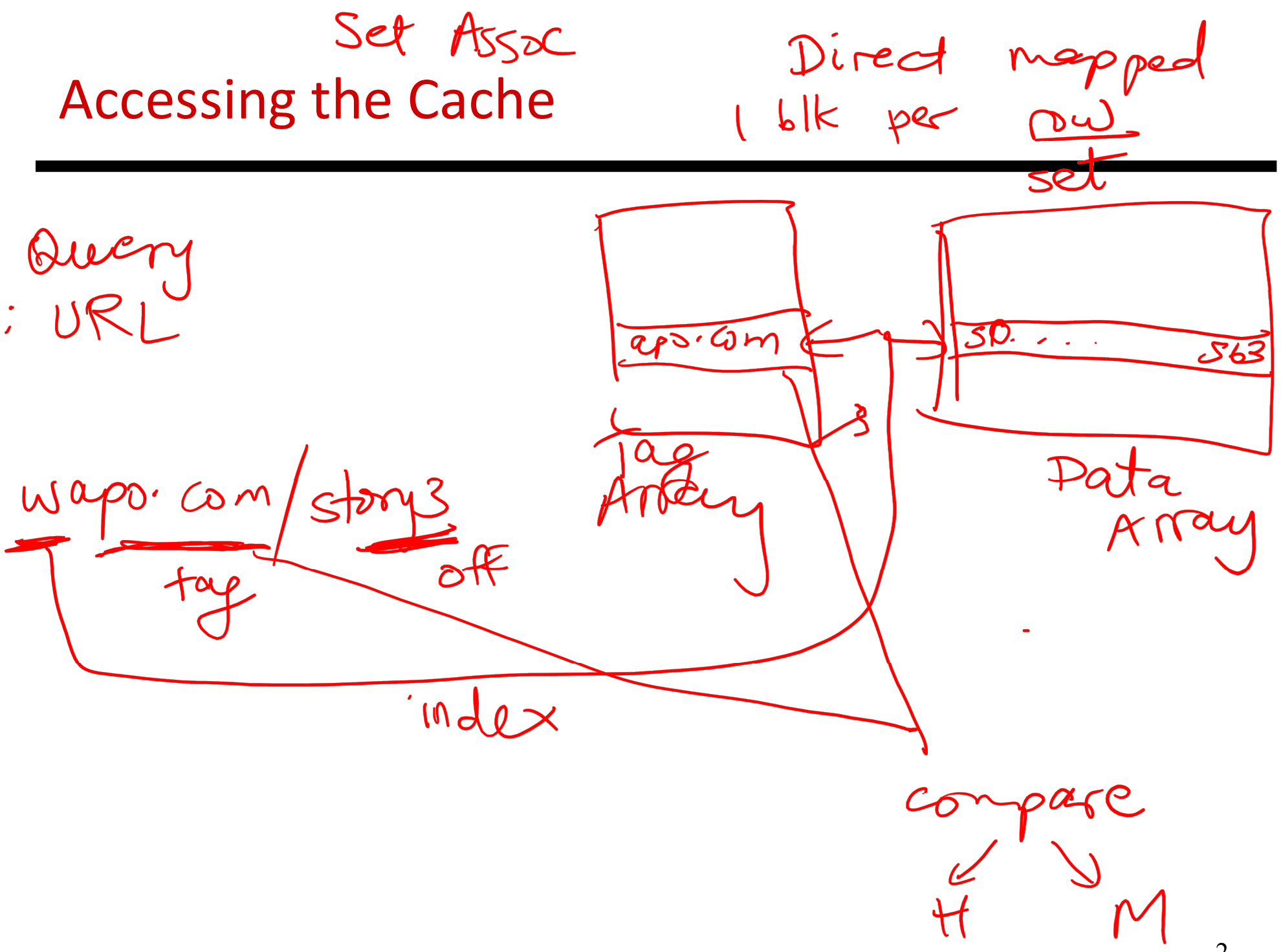
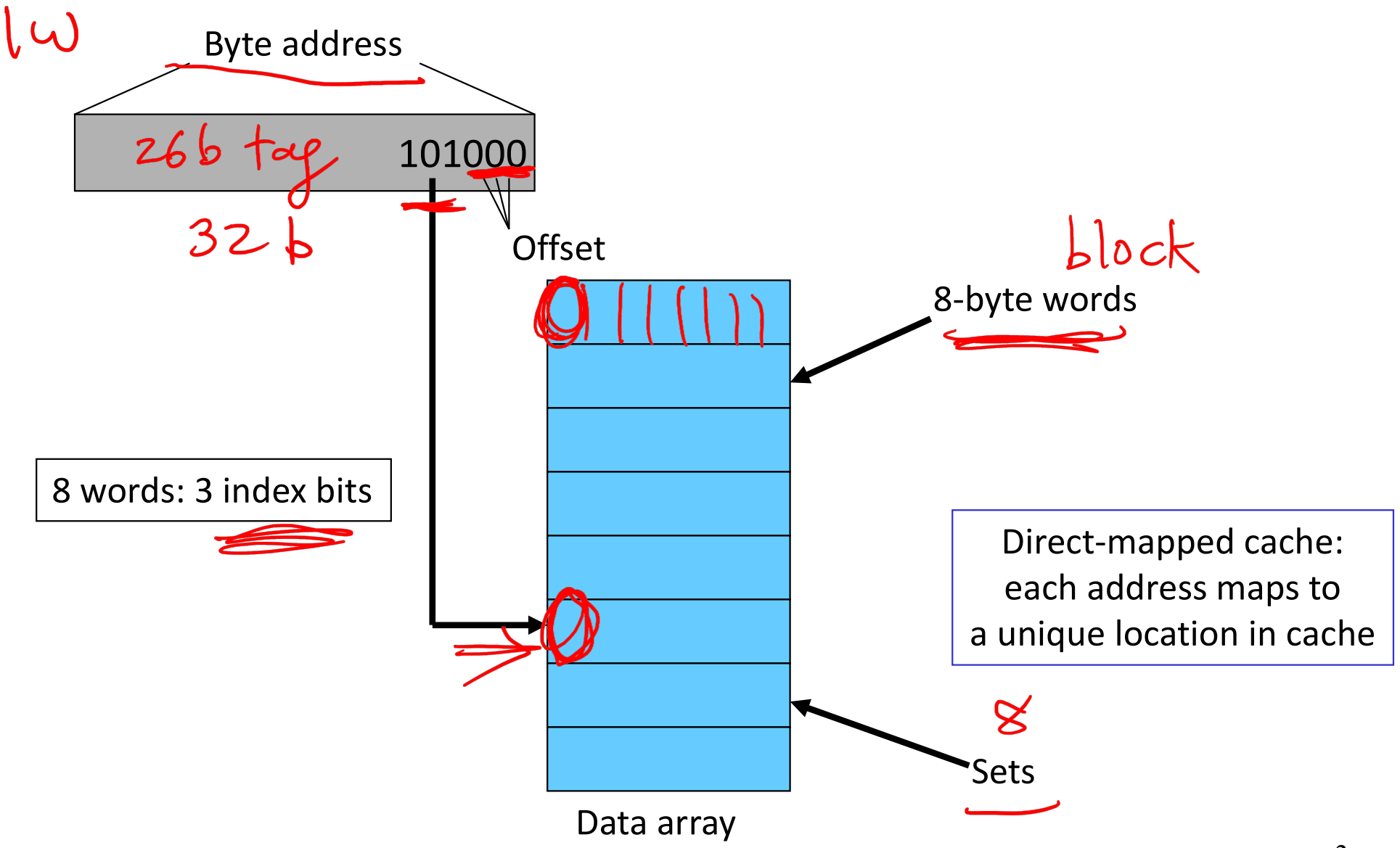
Lecture 22: Cache Examples

- Today's topics:
  - Cache access
  - Example problems in cache design

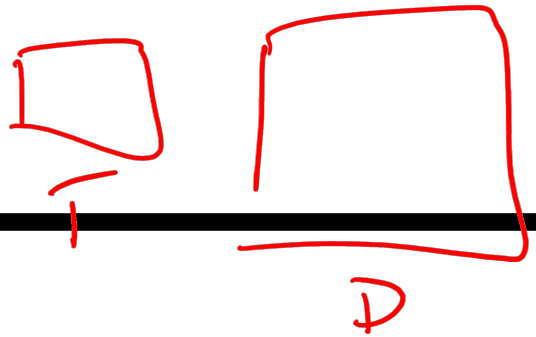
Accessing the Cache

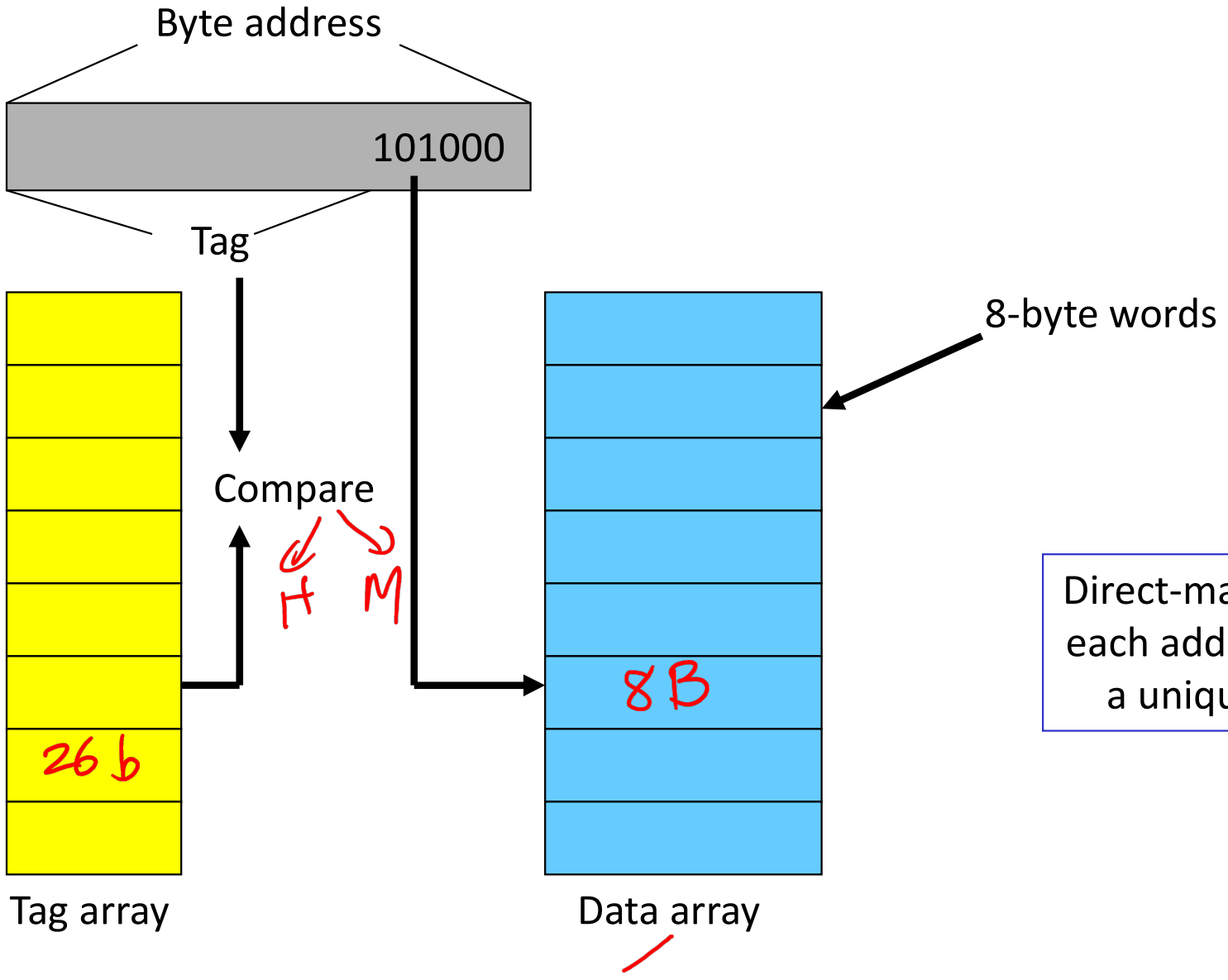


Accessing the Cache



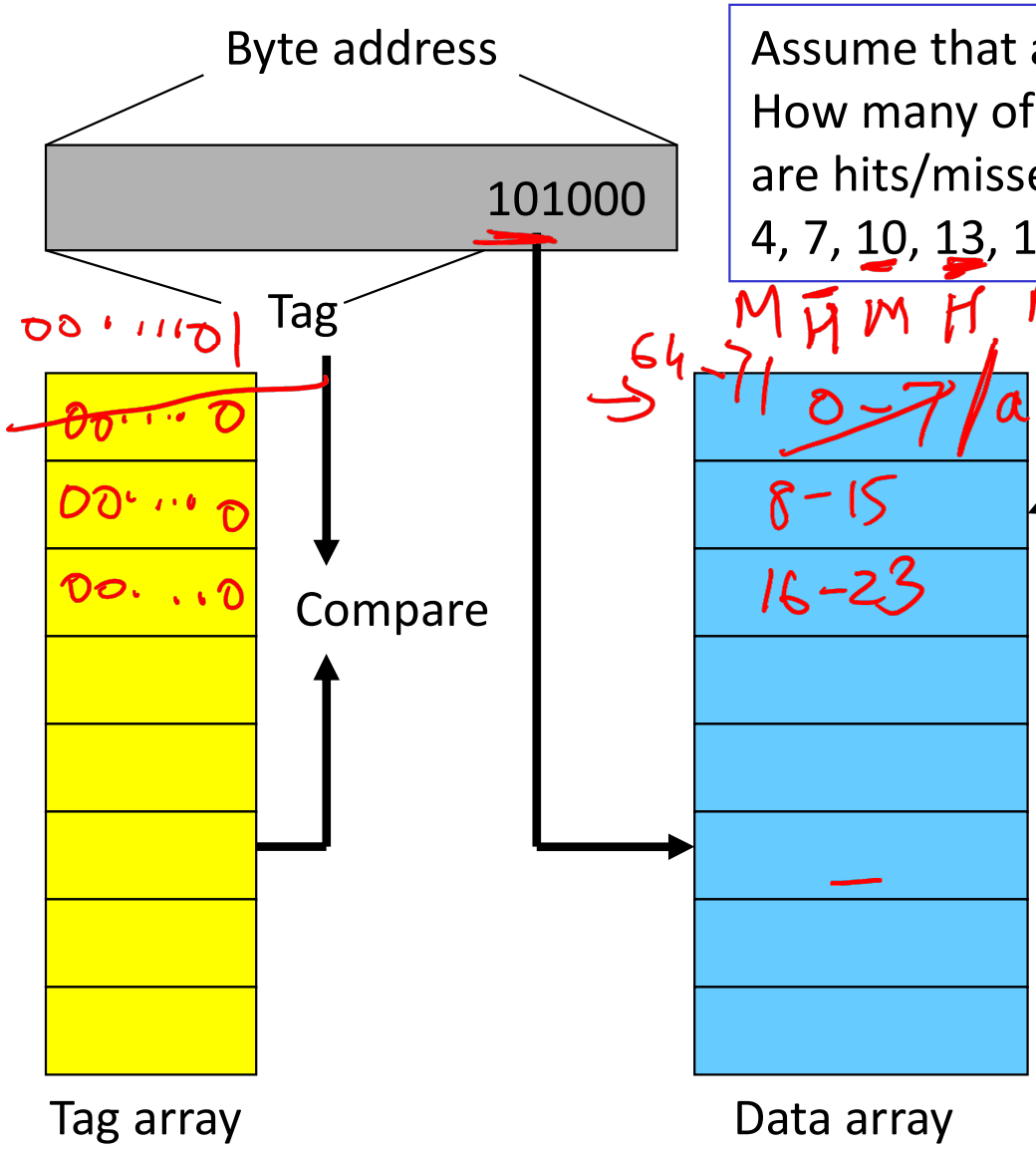
The Tag Array





Direct-mapped cache: each address maps to a unique address

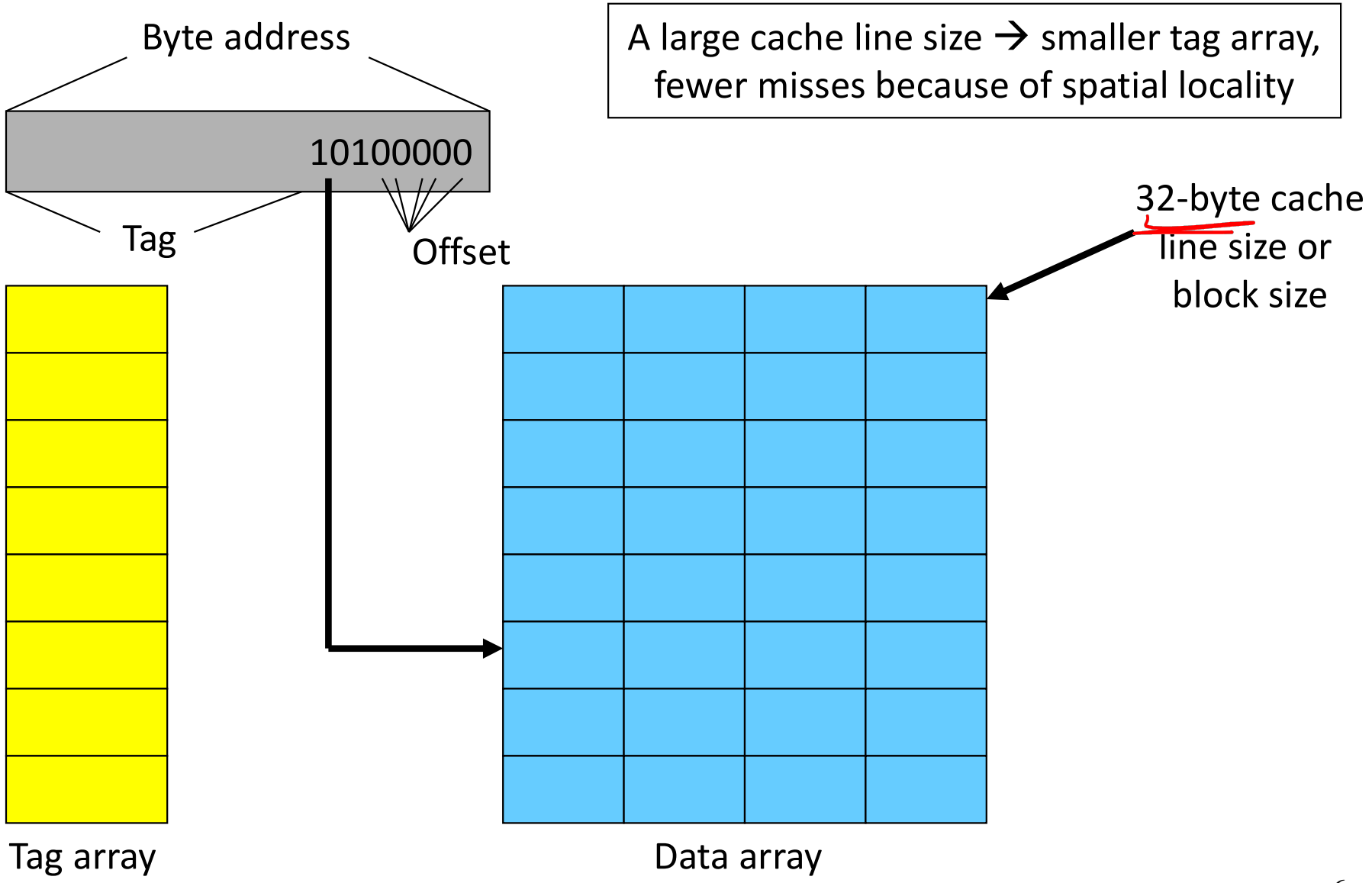
Example Access Pattern



Assume that addresses are 8 bits long
How many of the following address requests are hits/misses?
4, 7, 10, 13, 16, 68, 73, 78, 83, 88, 4, 7, 10...

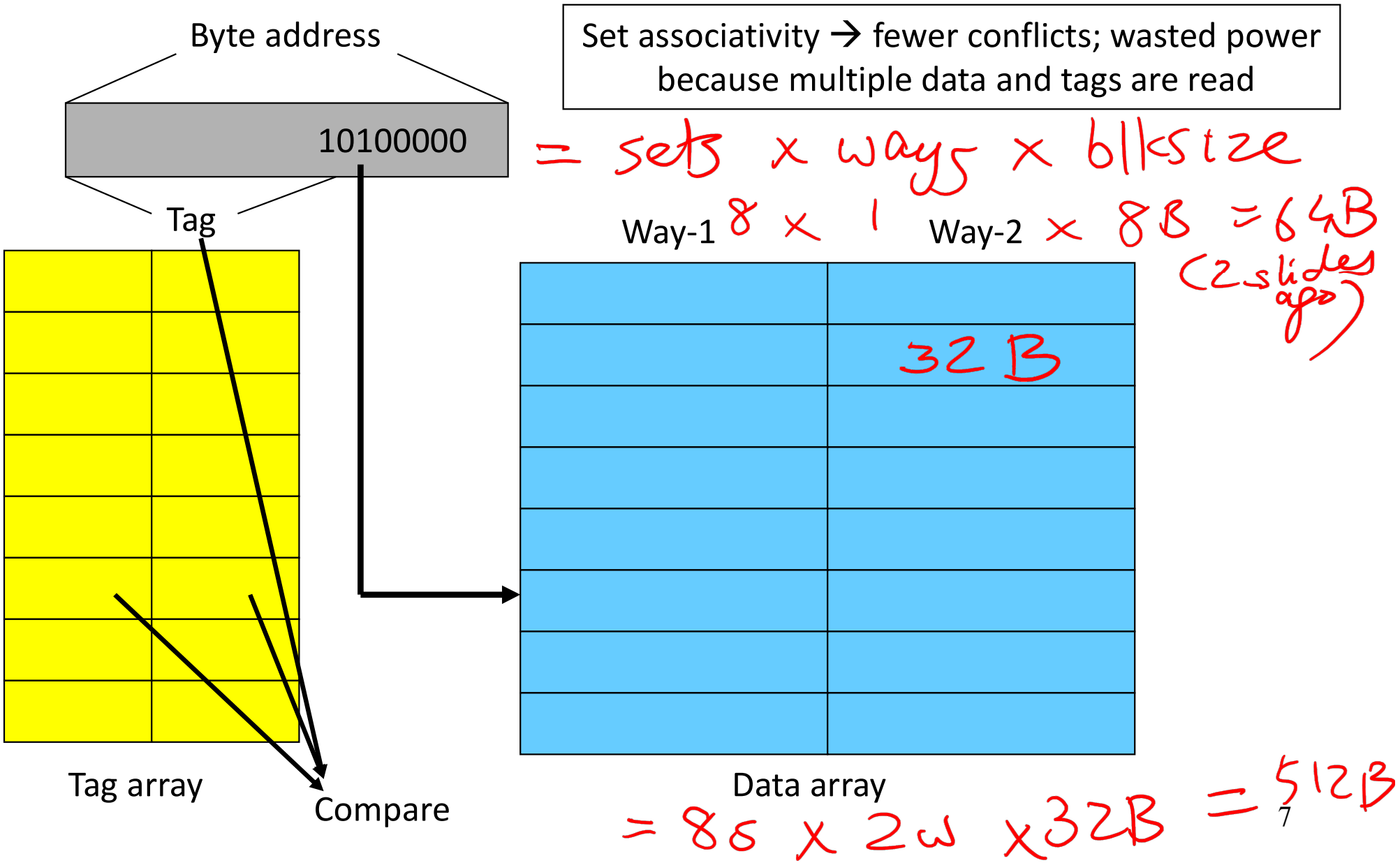
Direct-mapped cache: each address maps to a unique address

Increasing Line Size



Associativity

Cache capacity (size) = rows x cols x blksize



Set associativity → fewer conflicts; wasted power because multiple data and tags are read

= sets x ways x blksize

8 x 1 x 8B = 64B (2 slides ago)

= 8s x 2w x 32B = 512B

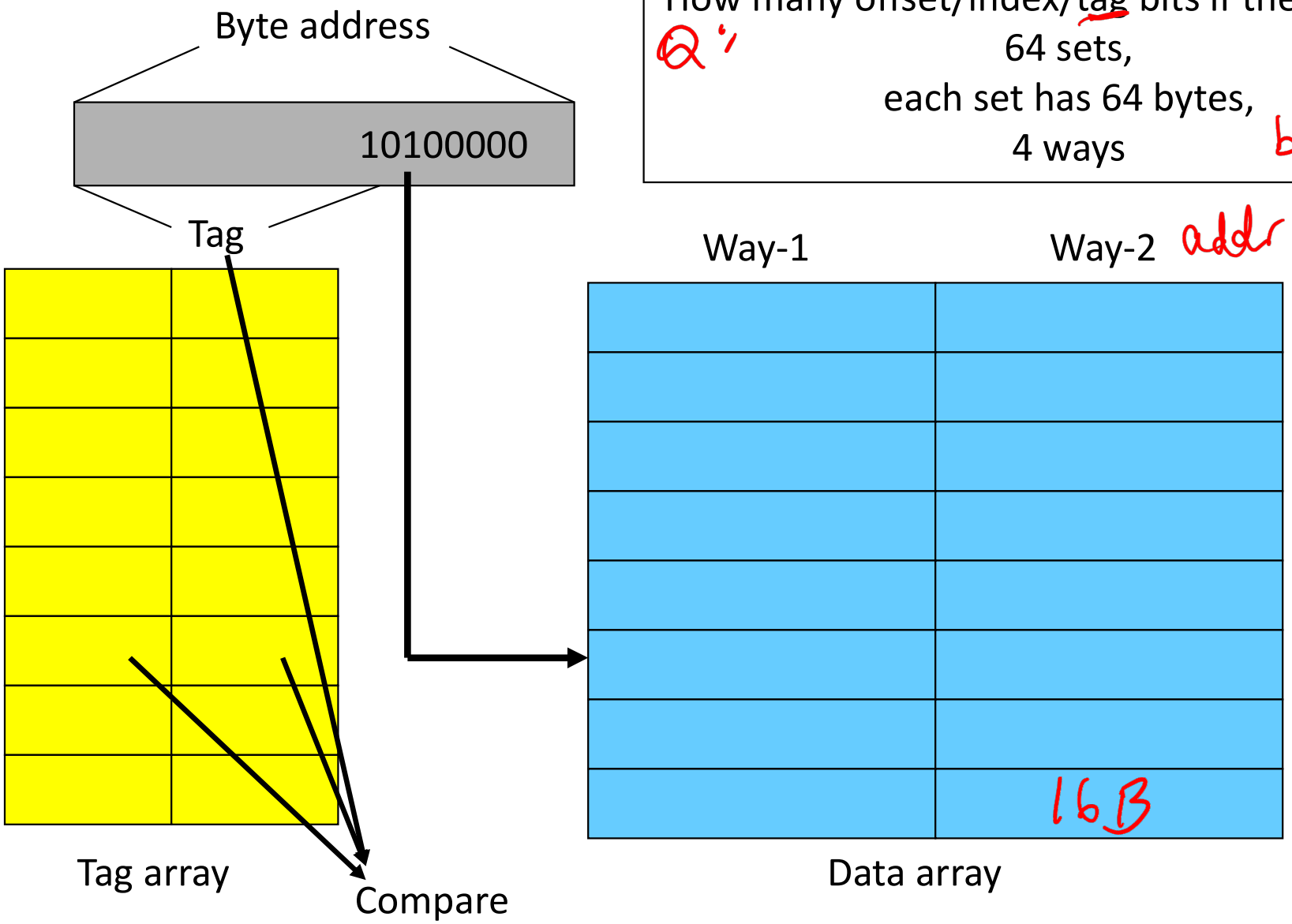
Associativity

Cache size = 64 s x 4 w x 16 B
= 4096 B = 4 KB

How many offset/index/tag bits if the cache has
64 sets,
each set has 64 bytes,
4 ways

Q:

blk size = 16B



addr:
22b 6b 4b
tag index off

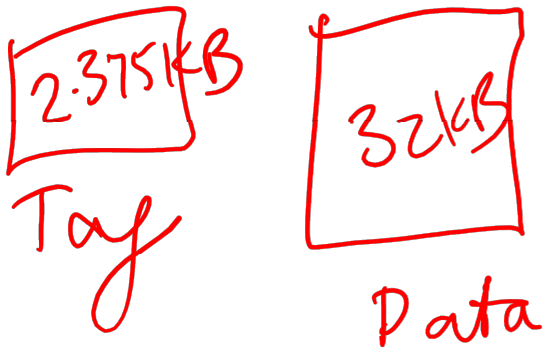
index = log2(sets)

Example 1

32KB = sets x 4 x 32B

sets = 32KB / (4 x 32B) = $2^5 \times 2^{10} / 2^2 \times 2^5 = 2^8 = 256$

- 32 KB 4-way set-associative data cache array with 32 byte line sizes (blksize)
- How many sets? 256
- How many index bits, offset bits, tag bits? 8 5 19
- How large is the tag array? = 256 x 4 x 19b = 19 Kb = 2.375 KB



Data array

Cache size = #sets x #ways x blocksize

Index bits = $\log_2$(sets)

Offset bits = $\log_2$(blocksize)

Addr width = tag + index + offset

$2^{indexbits}$ = sets

$2^{offsetbits}$ = blksize

32b

Tag array size = #sets x #ways x tagsize

Example 1

- 32 KB 4-way set-associative data cache array with 32 byte line sizes

  cache size = #sets x #ways x block size

- How many sets? 256

- How many index bits, offset bits, tag bits?

8	5	19
$\log_2$(sets)	$\log_2$(blksize)	addrsize-index-offset

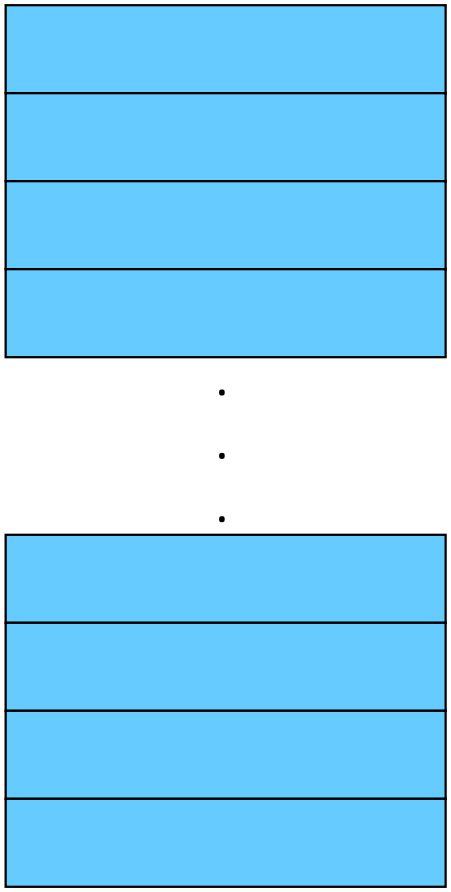
- How large is the tag array?

  tag array size = #sets x #ways x tag size

  = 19 Kb = 2.375 KB

Example 2

Show how the following addresses map to the cache and yield hits or misses.
The cache is direct-mapped, has 16 sets, and a 64-byte block size.
Addresses: 8, 96, 32, 480, 976, 1040, 1096

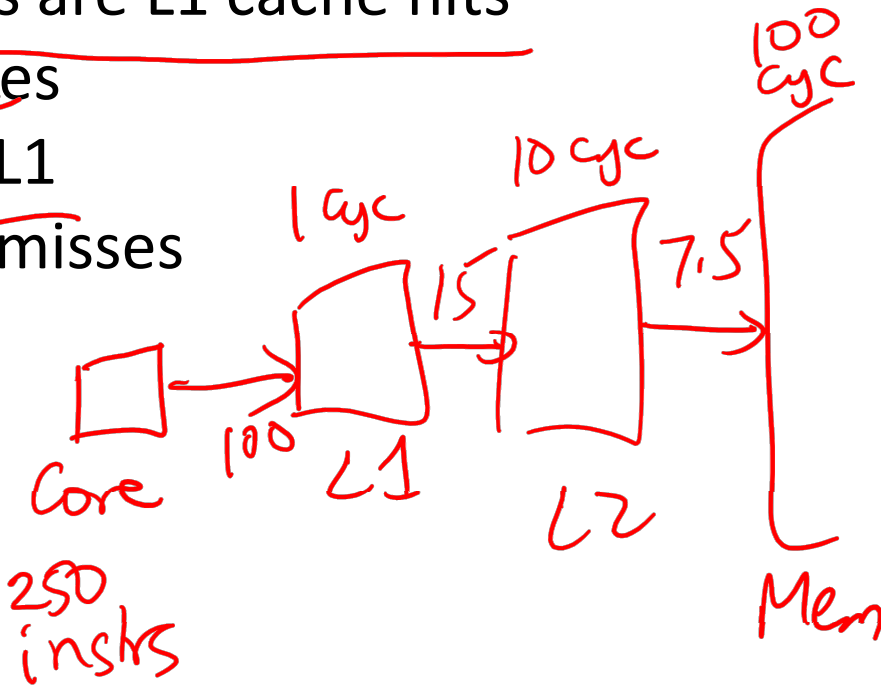


Offset = address % 64 (address modulo 64, extract last 6)
Index = address/64 % 16 (shift right by 6, extract last 4)
Tag = address/1024 (shift address right by 10)

	32-bit address			
	22 bits tag	4 bits index	6 bits offset	
8:	0	0	8	M
96:	0	1	32	M
32:	0	0	32	H
480:	0	7	32	M
976:	0	15	16	M
1040:	1	0	16	M
1096:	1	1	8	M

Example 3

- A pipeline has CPI 1 if all loads/stores are L1 cache hits
  40% of all instructions are loads/stores
  85% of all loads/stores hit in 1-cycle L1
  50% of all (10-cycle) L2 accesses are misses
  Memory access takes 100 cycles
  What is the CPI?



stalls from

misses = 15 x 10 cyc

+ 7.5 x 100 cyc

= 900 cyc

Exec time = 250 + 900

CPI = 1150 / 250 = 4.6

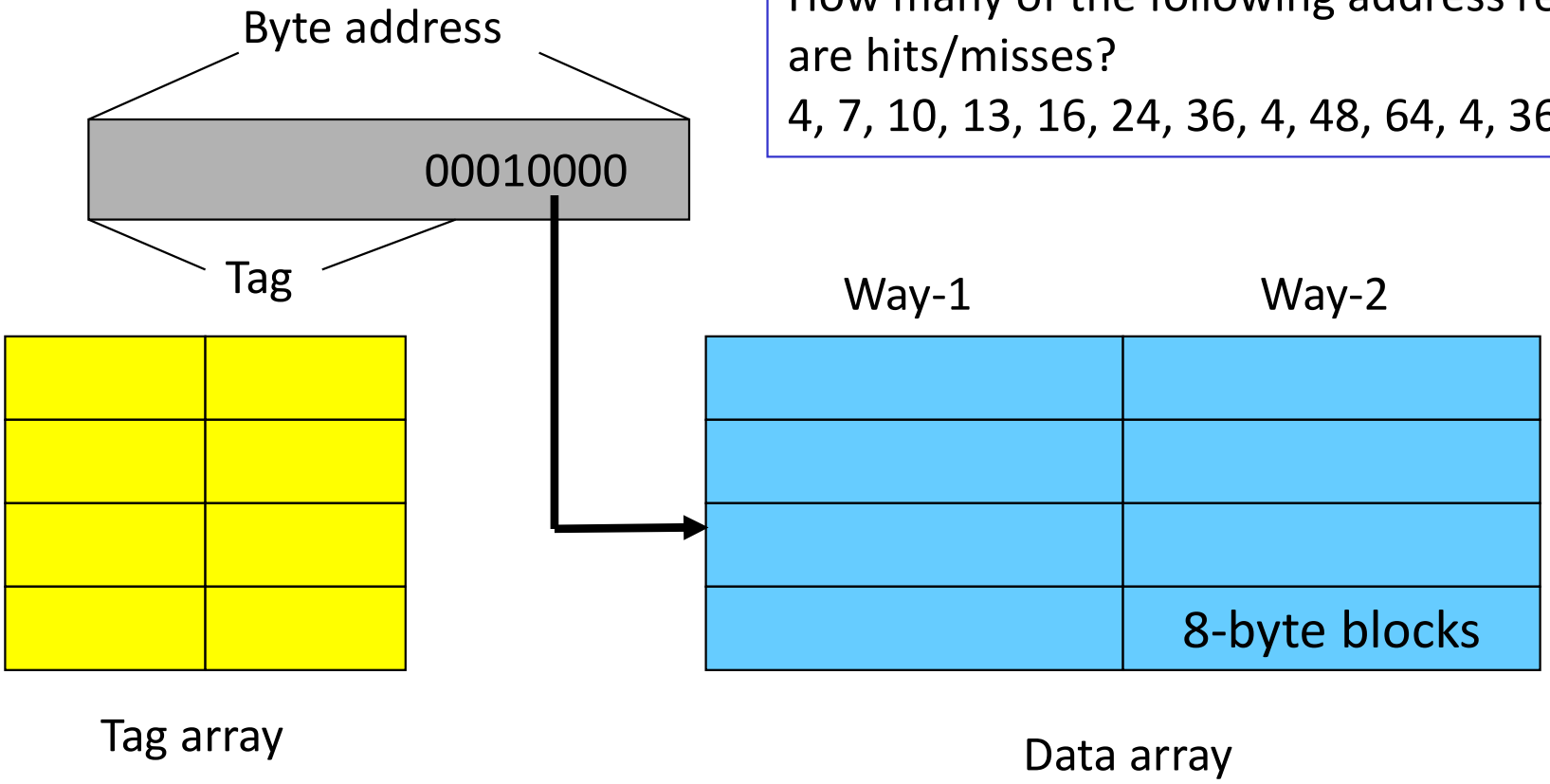
Example 3

- A pipeline has CPI 1 if all loads/stores are L1 cache hits
  40% of all instructions are loads/stores
  85% of all loads/stores hit in 1-cycle L1
  50% of all (10-cycle) L2 accesses are misses
  Memory access takes 100 cycles
  What is the CPI?

Start with 1000 instructions
1000 cycles (includes all 400 L1 accesses)
+ 400 (ld/st) x 15% x 10 cycles (the L2 accesses)
+ 400 x 15% x 50% x 100 cycles (the mem accesses)
= 4,600 cycles
CPI = 4.6

Example 4

Assume that addresses are 8 bits long
How many of the following address requests are hits/misses?
4, 7, 10, 13, 16, 24, 36, 4, 48, 64, 4, 36, 64, 4



Example 4

Assume that addresses are 8 bits long
How many of the following address requests are hits/misses?
4, 7, 10, 13, 16, 24, 36, 4, 48, 64, 4, 36, 64, 4
M H M H M M M H M M H M M M