High Performance Computing
Lecture 27

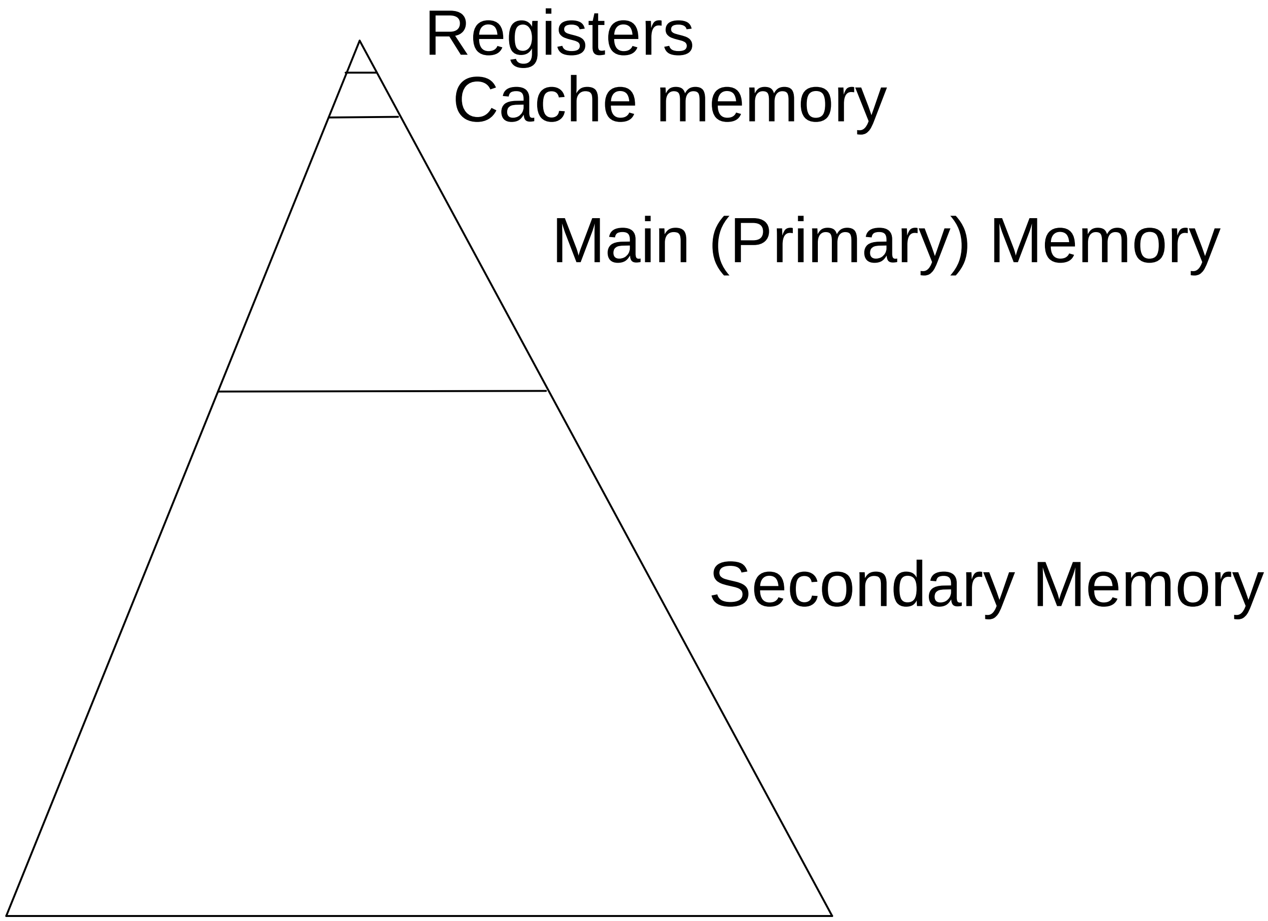
Matthew Jacob

Indian Institute of Science

Memory Hierarchy

- CPU registers
  - few in number (typically 16/32/128)
  - subcycle access time (nsec)
- Cache memory
  - on-chip memory
  - 10’s of KBytes (to a few MBytes)
  - access time of a few cycles
- Main memory
  - 100’s of MBytes storage (to a few GBytes)
  - access time several 10’s of cycles
- Secondary storage (like disk)
  - 100’s of GBytes storage (to a few TBytes)
  - access time of msecs

Memory Hierarchy



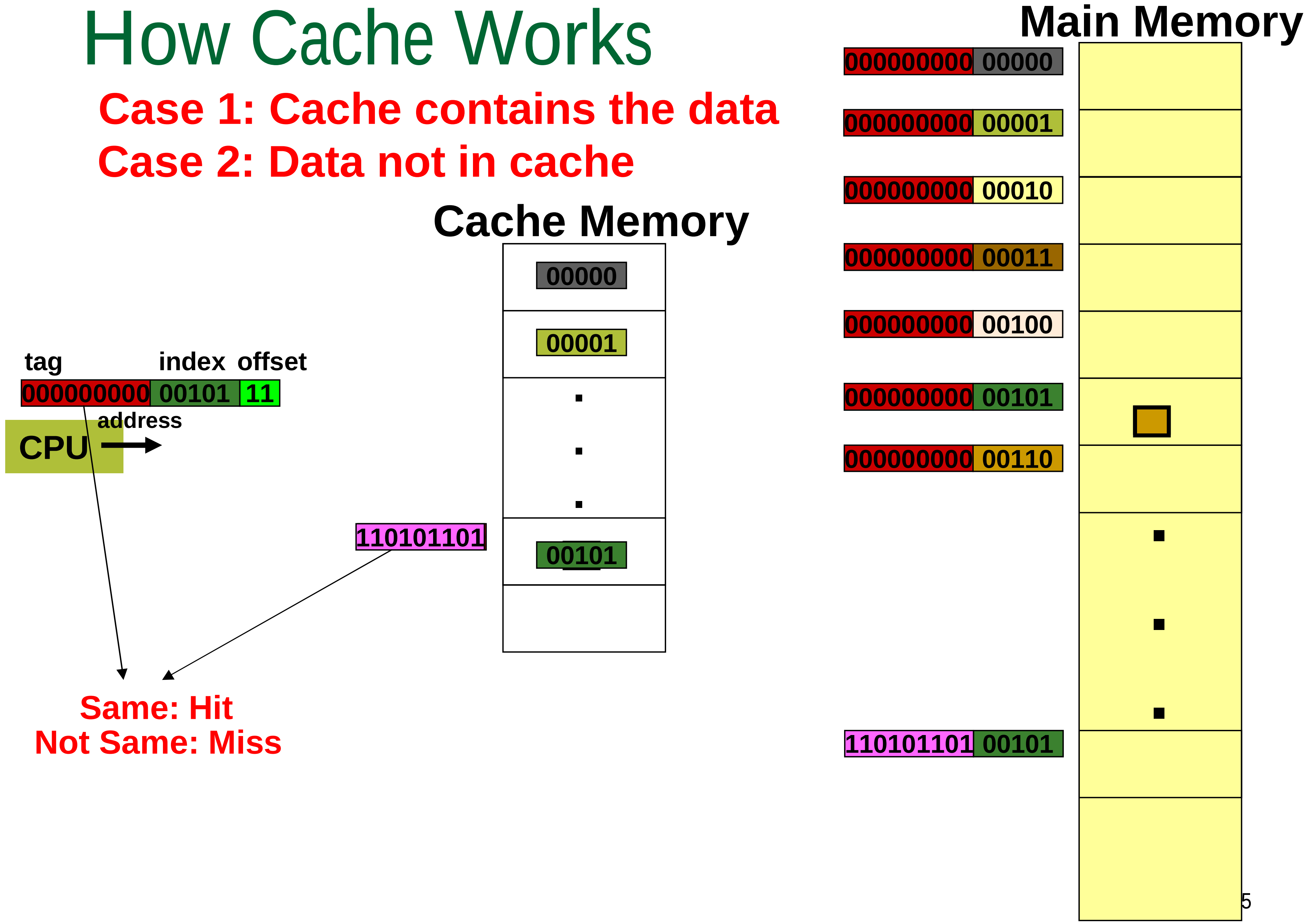
Why Hierarchy?

- Consider hierarchy in a business organization
  - Purpose: Right quantity of right quality human resource to achieve the required performance
- Realities of storage: Size-speed tradeoff
  - Disks: large storage, slow speed, low cost
  - Silicon memory: high cost for large, fast memory
  - So, cost effective memory hierarchy with
    - Small amount of very fast memory
    - Affordable amount of medium speed memory
    - Huge amounts of very slow memory

How Cache Works

Case 1: Cache contains the data

Case 2: Data not in cache



Cache Terminology

- Cache hit: A memory reference where the required data is found in the cache
- Cache Miss: A memory reference where the required data is not found in the cache
- Hit Ratio: # of hits / # of memory references
- Miss Ratio = (1 - Hit Ratio)
- Hit Time: Time to access data in cache
- Miss Penalty: Time to bring a block to cache

The 4 Qs of Cache Organization

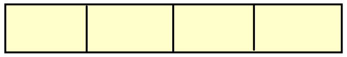
1. Where can a memory block be placed in the cache? (Block Placement)
   - Direct mapped, Set Associative

2. How is a block identified in the cache?
   - Tag, valid bit, tag checking hardware

3. What is the replacement policy used?
   - LRU, FIFO, Random …

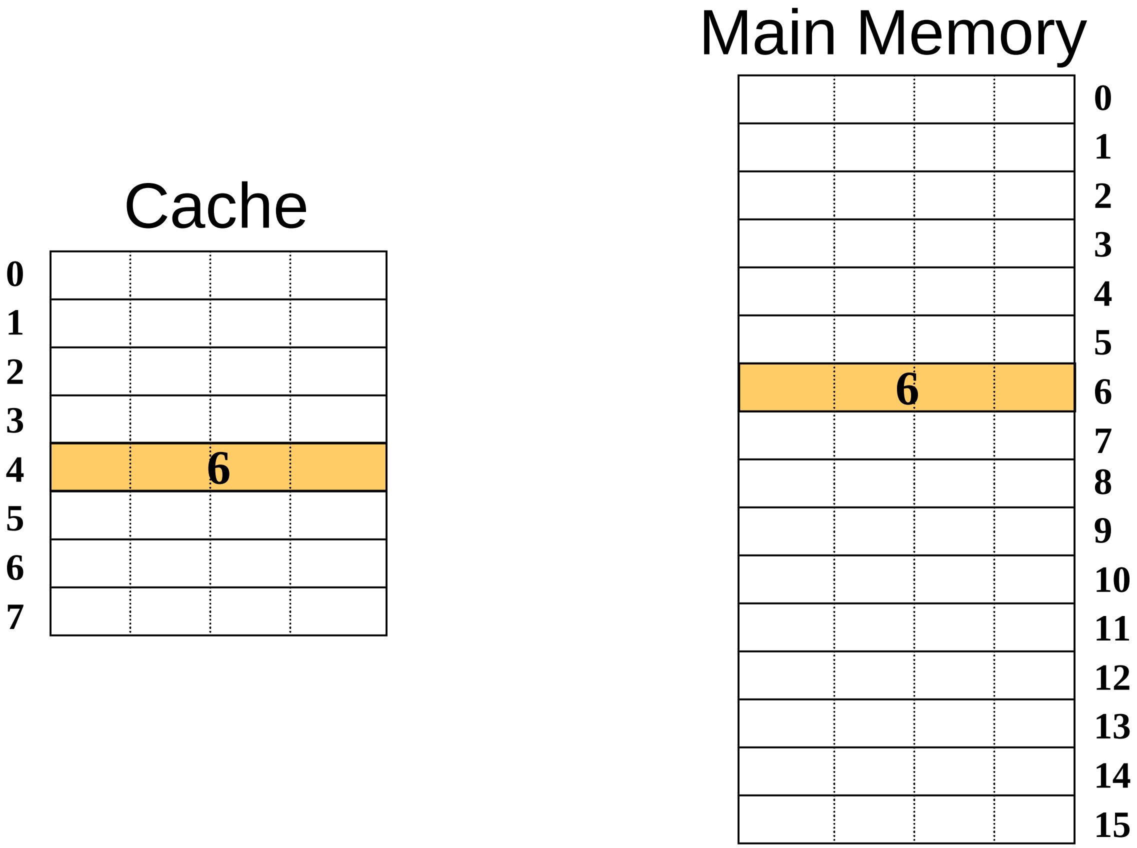
4. What happens on writes to the cache?
   - Cache Hit: When is main memory updated?
   - Cache Miss: What happens on a write miss?

Notation

Block 

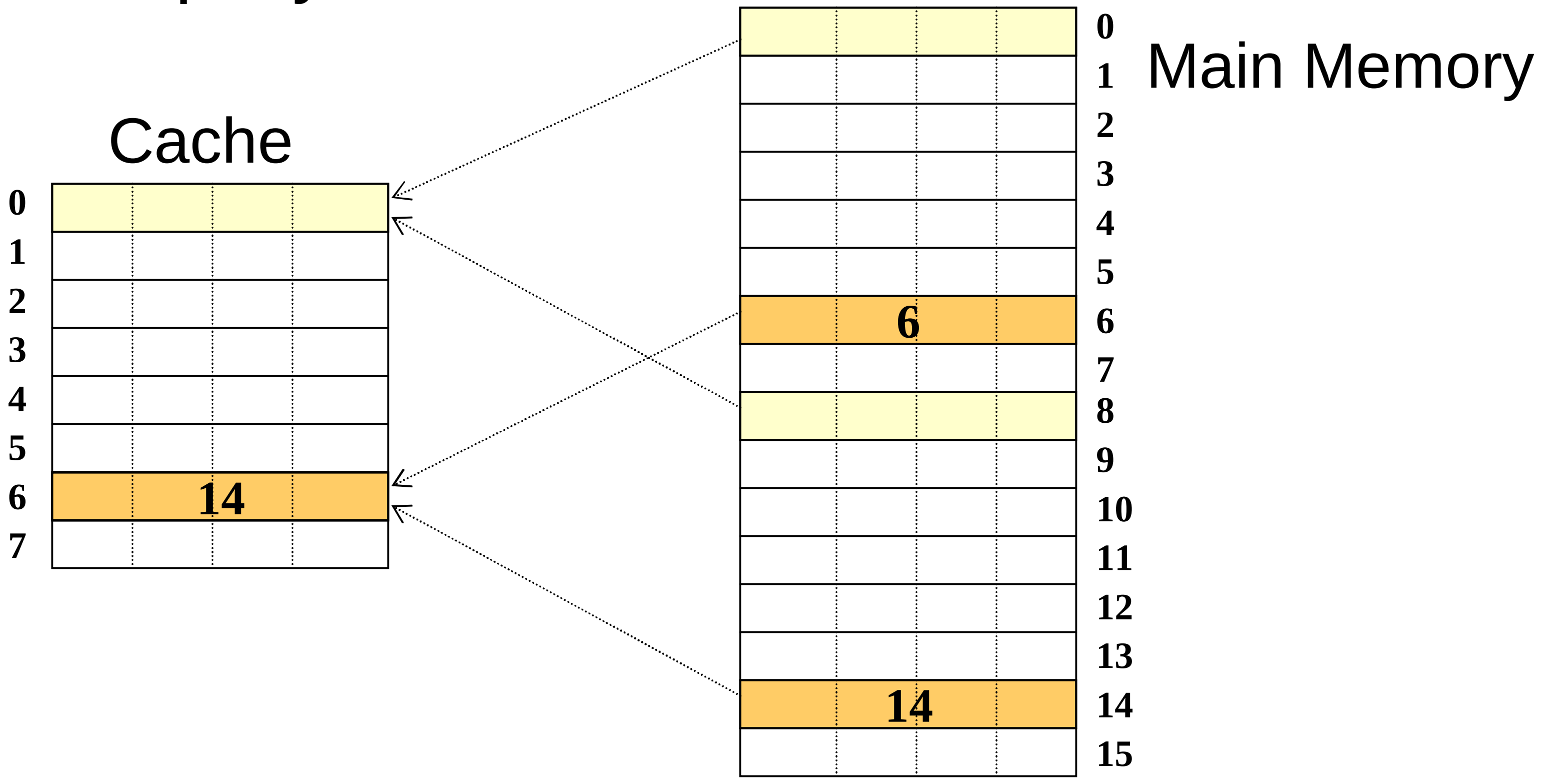
(Depicted as a block of size 4 bytes for convenience in drawing)

8 block cache memory and 16 block main memory



Block Placement: Direct Mapping (DM)

- Suppose that the cache is large enough to hold N blocks from main memory
  - i.e., Cache size = N blocks
- Direct Mapping: Memory block M is placed uniquely in cache block M mod N



Identifying Block in DM Cache

Assume 32 bit address space, 16 KB cache, 32byte cache block size.

Number of cache blocks = 16KB / 32B = 512

Index field: to identify the unique cache block

$\log_2 512 = 9$ bits

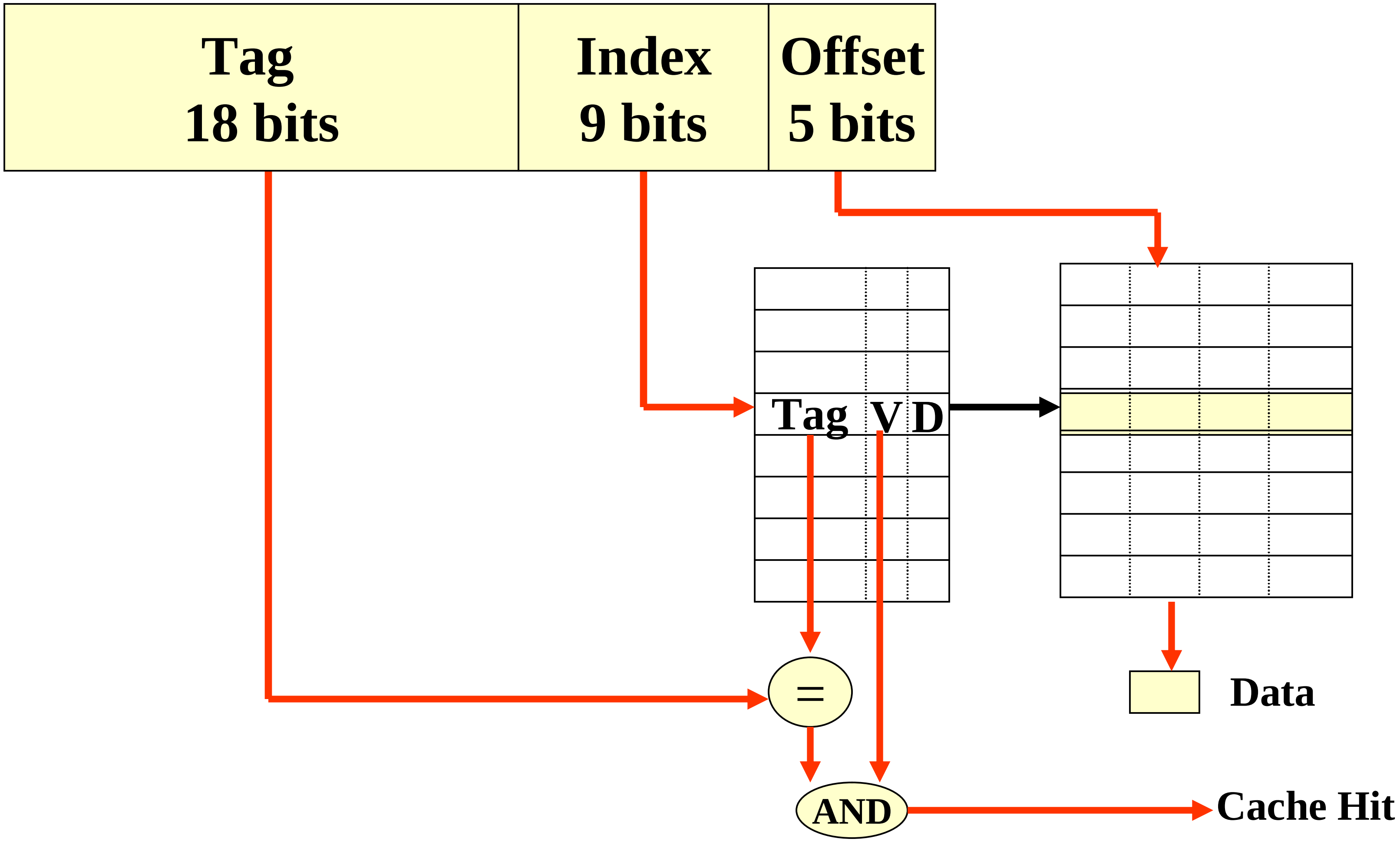
Offset field: to identify the desired byte in cache block

$\log_2 32 = 5$ bits

Tag field: to identify which memory block is currently in this cache block (remaining 18 bits)

Tag	Index	Offset

Accessing Block in DM Cache


Tag
18 bits
Index
9 bits
Offset
5 bits
Tag
V
D
=
AND
Data
Cache Hit