Chapter 5 - MEMORY

Memory Hierarchy = A structure that uses multiple levels of memory.

(Distance from processor ↑ → Size of memories & access time ↑, Cost per bit ↓)

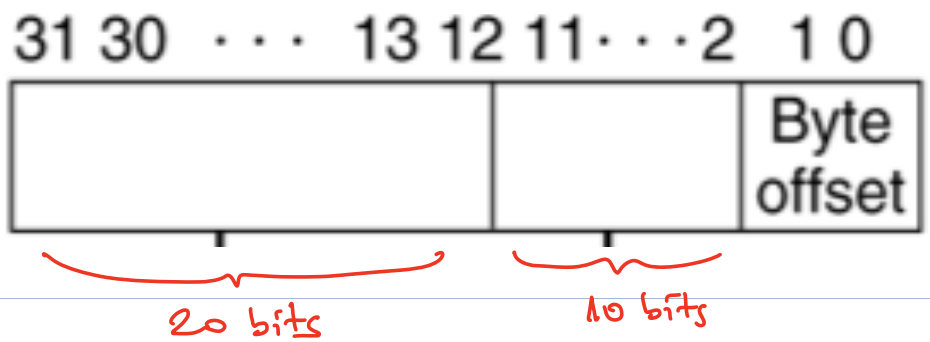
Block (or line) = minimum unit of information that can be present or not present in cache.

Hit Rate = Fraction of memory accesses found in a level of the memory hierarchy.

Direct-Mapped Cache = A cache structure in which each memory location is mapped to exactly one location in the cache.

Tag = A field in a table used for a memory hierarchy that contains the address info. required to identify whether the associated block in the hierarchy corresponds to a requested word.

Decimal address of reference	Binary address of reference	Hit or miss in cache	Assigned cache block (where found or placed)
22	10110_{two}	miss (5.9b)	$(10110_{two} \bmod 8) = 110_{two}$
26	11010_{two}	miss (5.9c)	$(11010_{two} \bmod 8) = 010_{two}$
22	10110_{two}	hit	$(10110_{two} \bmod 8) = 110_{two}$
26	11010_{two}	hit	$(11010_{two} \bmod 8) = 010_{two}$
16	10000_{two}	miss (5.9d)	$(10000_{two} \bmod 8) = 000_{two}$
3	00011_{two}	miss (5.9e)	$(00011_{two} \bmod 8) = 011_{two}$
16	10000_{two}	hit	$(10000_{two} \bmod 8) = 000_{two}$
18	10010_{two}	miss (5.9f)	$(10010_{two} \bmod 8) = 010_{two}$
16	10000_{two}	hit	$(10000_{two} \bmod 8) = 000_{two}$



The cache holds 1024 words = 4 KiB

Block size = 1 word & cache has $1024 = 2^{10}$ words

(10-Bit for index 32 − 10 − 2 = 20 bit for TAG)

Block = (Block Addr) mod (# of blocks in cache)

Block Addr = Byte Addr / Bytes per block

(Block Addr) is the block containing all addresses between

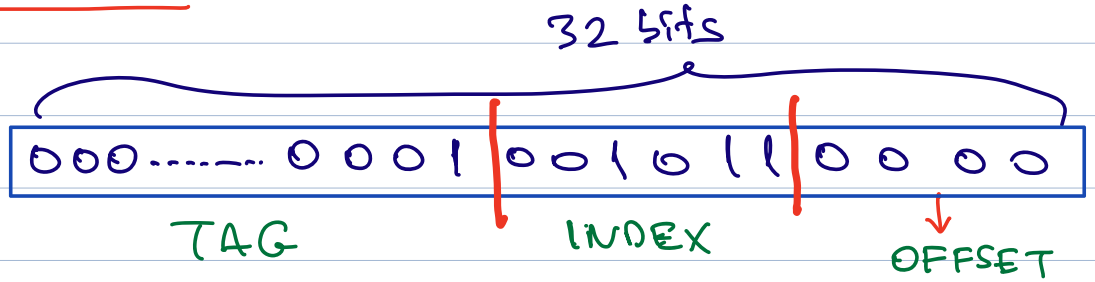
$$\left[\frac{\text{byte addr}}{\text{bytes per block}}\right] \times \text{Bytes per block}$$

$$\left[\frac{\text{byte addr}}{\text{bytes per block}}\right] \times \text{Bytes per block} + (\text{Bytes per block} - 1)$$

Question: Address → 1200 mapped to which block?

Assume 64 blocks & 1 block = 16 bytes given

Answer



Index = 11
(Block number)

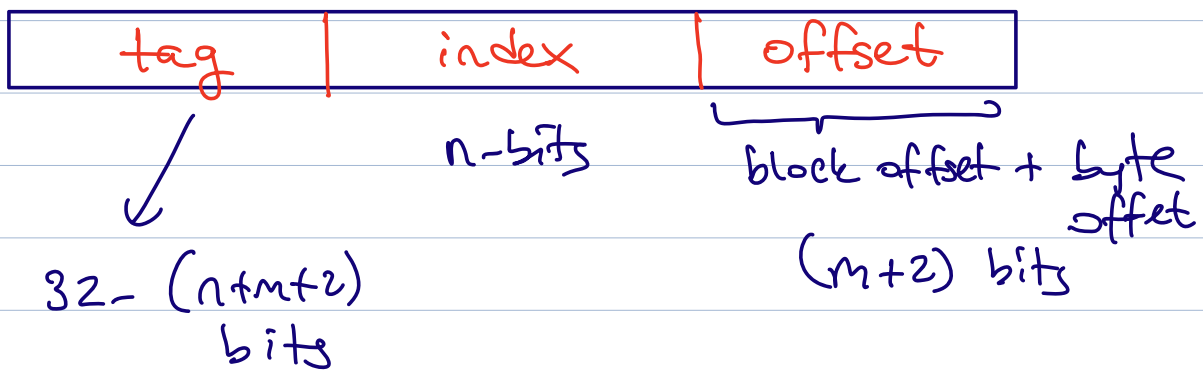
That block maps all addess between
1200 and 1215

Total #of Bits for a Cache

Total # of bits needed for a cache is a function of Cache size & address size

* 32-bit addess
* Direct-mapped Cache
* Cache size is 2^n blocks (n-bit for index)
* Block size is 2^m words (2^{m+2} bytes)
  (m-bits word within block,
  2 bits byte port of the addr)

Size of tag field: $32-(n+m+2)$



Total # of bits in a direct-mapped cache

$= 2^n \times$ (block size + tag size + valid field size)

$= 2^n \times (2^m \times 32 + (32 - n - m - 2) + 1)$

$= 2^n \times (2^m \times 32 + 31 - n - m)$