Lecture 21: March 30, 2021
Computer Architecture and Organization-I
Biplab K Sikdar

0.3.4 Content Addressable Memory (CAM)

Content addressable memory (CAM), also known as associative memory.

A CAM cell has storage capability as well as circuitry to realize matching its content with an argument. A CAM block is shown in Figure 23.

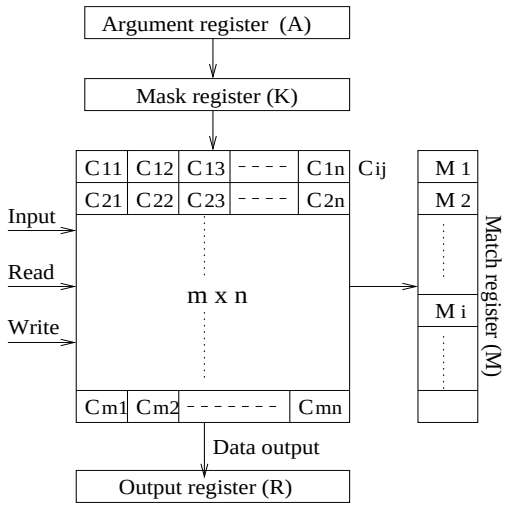


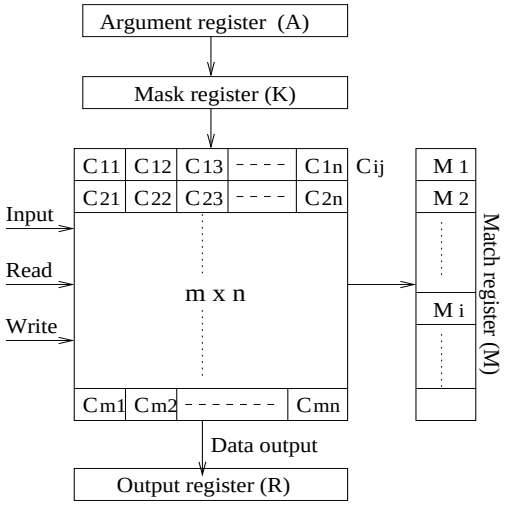
Figure 23: Content addressable memory (CAM)

Each word ($C_{i1}C_{i2}C_{i3} \cdots C_{in}$) is compared in parallel with argument register A ($A_1A_2A_3 \cdots A_n$) content.

Also gives due consideration of mask register K.

$K_j = 0 \Rightarrow j^{th}$ bit (C_{ij}) of word (i) is not to be compared.

If there is a match between A and word i, respective bit M_i of M is set to 1.



Let consider,

$$\begin{aligned} A &= 1010\ 1101\ 0011\ 1110 \\ K &= 1111\ 0000\ 0000\ 0000 \\ word1 &= 1100\ 1101\ 0011\ 1110 \\ word2 &= 1010\ 0000\ 0000\ 1100 \end{aligned}$$

Here, all K_1, K_2, K_3 and K_4 are 1.

That is, only four MSBs of A are to be compared with corresponding bits of $word_1$/$word_2$.

For $word_1$, there is no match - and M_1 of M is set to 0.

For $word_2$, a match is found and M_2 is set to 1.

Match logic: Match logic is shown in Figure 24(b).

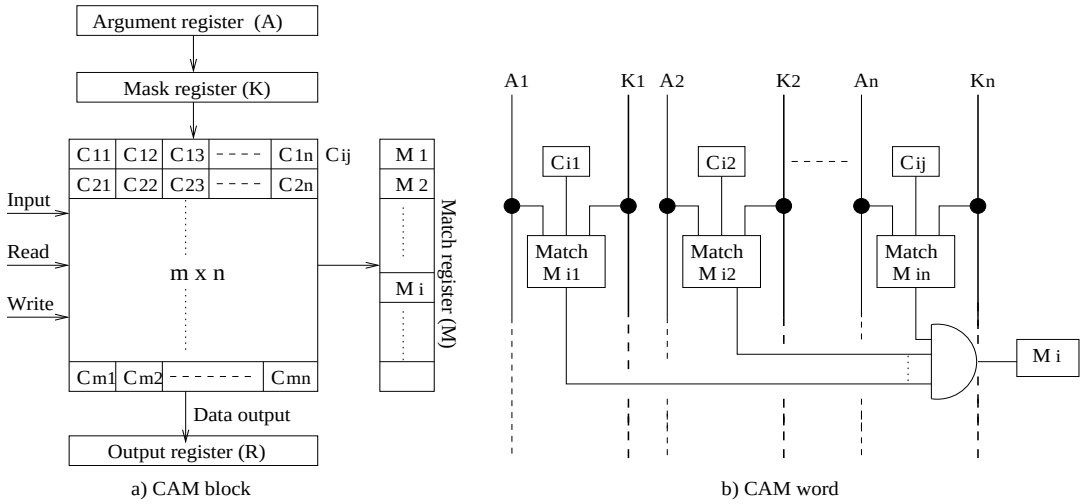


Figure 24: Content addressable memory (CAM)

Match logic M_{ij} compares j^{th} cell of i^{th} memory word with j^{th} bit (A_j) of A.

If all M_{ij}s, for an i are 1 -that is $\Pi_j M_{ij}$ = 1, then M_i = 1.

Word i of a CAM is matches with argument A -that is, M_i is set if in Figure 24(b)

$M_{ij} = A_j C_{ij} + A'_j C'_{ij} + K'_j$ is true for all j = 1 to n.

0.4 Cache Memory

Follow Figure 25.

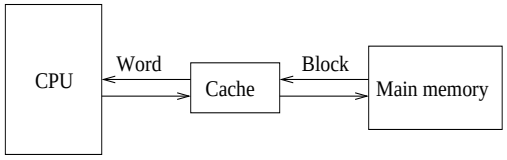


Figure 25: Cache main memory data transfer

Objective of the memory hierarchy (cache-MM) is to increase cache hit ratio (h).

$$h = \frac{N_1}{N_1+N_2}$$

N_1 = number of times the word soughted by the CPU is available in cache.

N_2 = number of times the word soughted by CPU is not available (miss) in cache.

miss ratio = 1-h.

miss penalty = time to transfer a block from MM to cache + time to deliver the item to CPU from cache.

Miss penalty is to be minimized.

Average access time is

AMAT = $h \times t_{cache}$ + (1-h)$\times t_2$.

t_2 = time taken to fetch from main memory to CPU bypassing cache or through cache as per the system design and architecture.

Several functions guide the performance of a cache based system -

cache size, block size, associativity: mapping function, replacement policy.

Cache to MM write policy (write-back or write-through).

0.4.1 Mapping function

1. Direct mapping
2. Associative mapping
3. Set associative mapping (m-way associative mapping).

0.4.2 Direct mapping

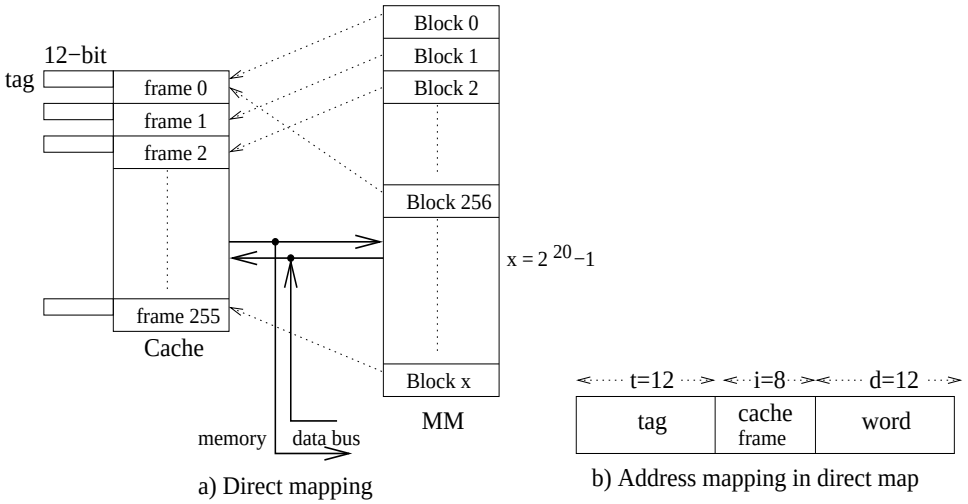


Figure 26: Direct mapping technique

Example: MM is of 2^{32} byte, page size = 4KB and cache is of 1MB (2^{20}).

Number of cache frames is

$$\frac{2^{20}}{2^{12}} = 2^8 = 256$$

Block i is mapped onto the frame (i modulo 256).

Number of blocks in MM is

$$\frac{2^{32}}{2^{12}} = 2^{20} = 1\text{M},$$

Therefore,

$$\frac{2^{20}}{2^{8}} = 2^{12} = 4\text{K}$$

MM blocks compete for a single cache frame.

That is, cache frame 0 can be occupied by MM blocks 0, 256, 512, $\cdots$.

12-bit tag is associated with each frame to signify currently residing block number.

Direct mapping is a many-to-one mapping.

0.4.3 Associative mapping

A main memory (MM) block can potentially reside in any cache frame.

So, if 1M main memory block is competing for 256 cache frames, 20 tag bits are required to identify an MM block in cache.

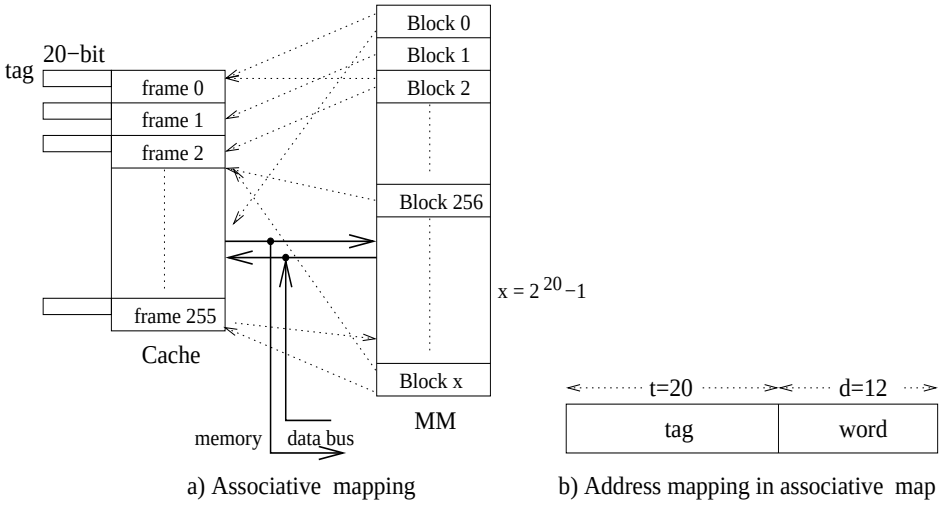


Figure 27: Associative mapping technique

Here, tags are stored in CAM so that all the tags can be compared simultaneously.