Lecture 11-12: September 1, 2021

Computer Architecture and Organization-II

Biplab K Sikdar

Cache Design

The memory subsystem of a m/c is shown in (Figure 1).

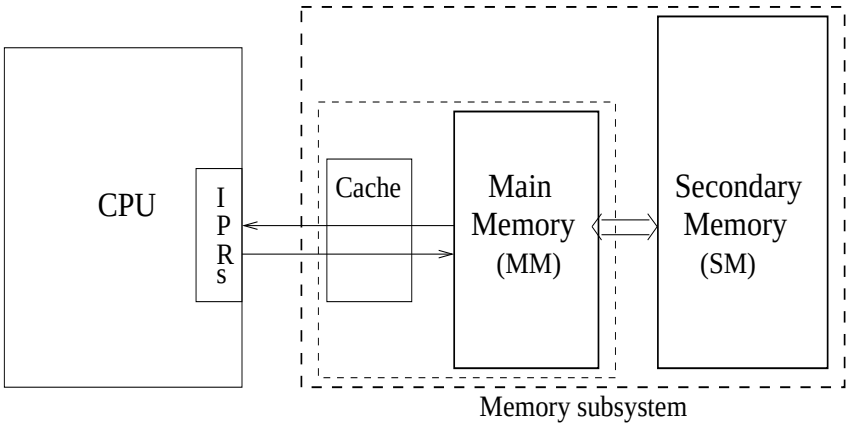


Figure 1: Computer memory subsystem

Access time:

t_A = t_2 - t_1 (Figure 2)

t_1: Memory read/write request generated by CPU,

t_2: Completion of read/write operation.

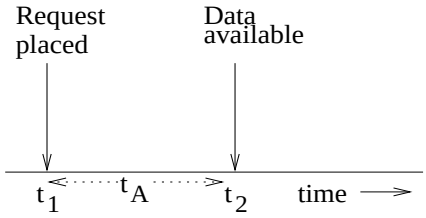


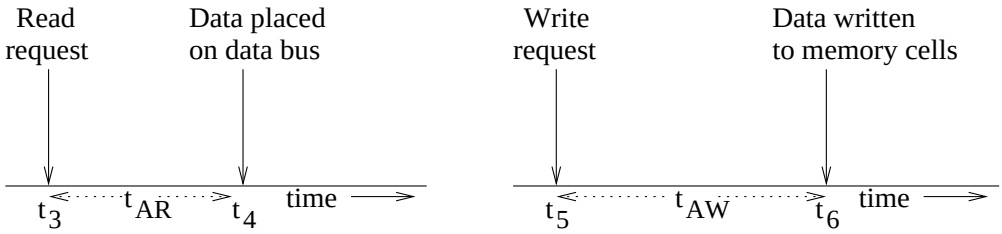
Figure 2: Memory access time

For memory, we define

read access time (t_{AR}, Figure 3(a)) and *write access time* (t_{AW}, Figure 3(b)).

$$t_A = \frac{t_{AR} + t_{AW}}{2}.$$

$$t_{AR} \simeq t_{AW} = t_A.$$



a) Memory read access time b) Memory write access time

Figure 3: Memory read/write access time

Cycle time

Minimum time delay between initiations of two successive operations.

Access time plus additional time required before a second access can commence.

e.g. non-destructive write in a DRO memory, refresh time in DRAM, etc.

Cycle time t_c is, therefore,

$$t_c = t'_2 - t_1 \text{ (Figure 4).}$$

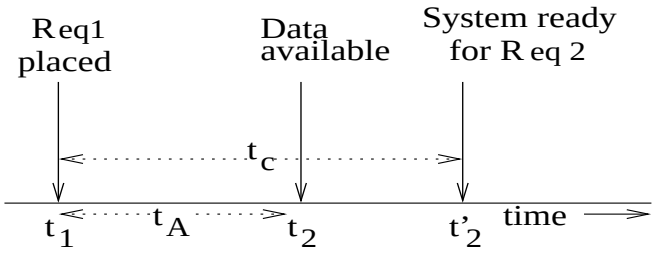


Figure 4: Cycle time

Memory cost vs access time

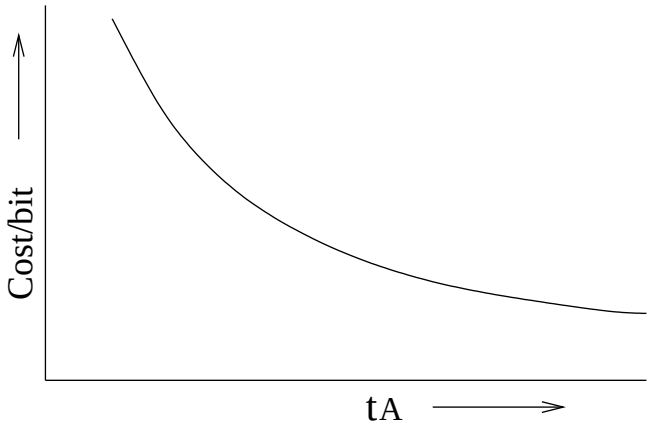


Figure 5: Memory cost and access time

0.1 Cache

Figure 6: CPU, cache and main memory interfacing.

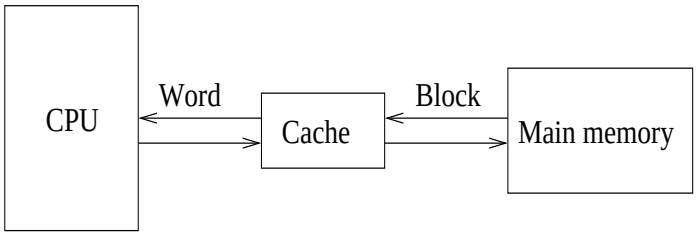


Figure 6: Cache main memory data transfer

During program execution when CPU refers item x, the item is searched in cache.

If x is found in cache, it is sent to CPU.

On the other hand, if x is not in cache, then CPU has to wait for x.

Search for x then goes to main memory.

If x is in main memory (MM), it is transferred to cache and then to CPU.

Real implementation: from MM a block containing x is transferred to cache.

But from cache to CPU only item x, referred by CPU, is transferred.

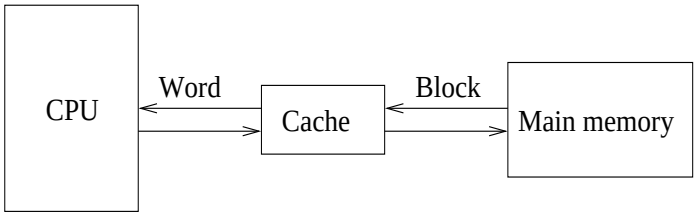
0.1.1 Cache performance

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

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

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

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



Miss rate/ratio

miss ratio = 1-h.

Miss penalty

For miss, the block containing requested item is to be transfered from MM to cache.

It adds to miss penalty

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

Average memory access time (AMAT)

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

t_2 = time taken to fetch from MM to CPU

(bypassing cache or through cache as per the system design).

Hit ratio is to be $\simeq 1$ so that average memory access time (AMAT) is minimized.

0.1.2 Mapping function

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

Direct mapping

A block i of MM is mapped onto frame $i(modulo\ X)$ of cache memory (Figure 7).

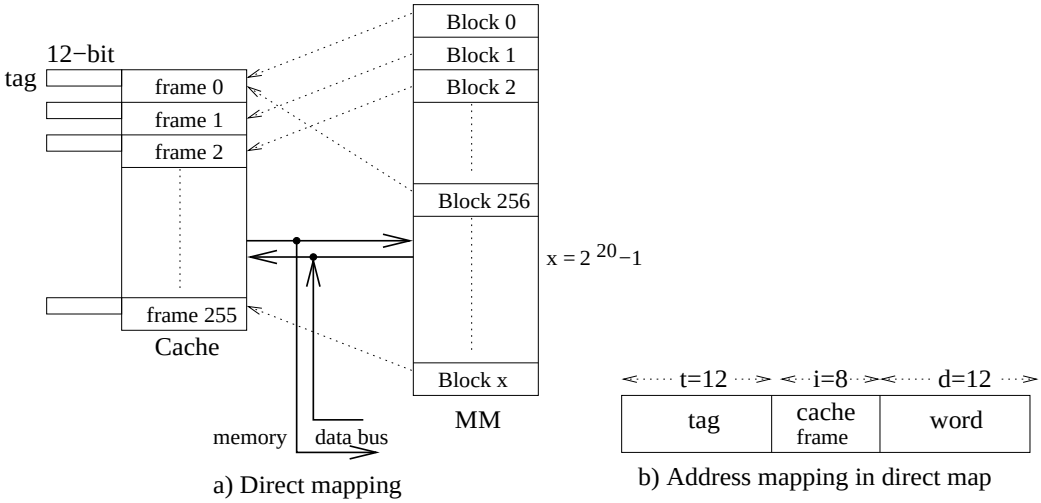


Figure 7: Direct mapping

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

Therefore, number of cache frames is

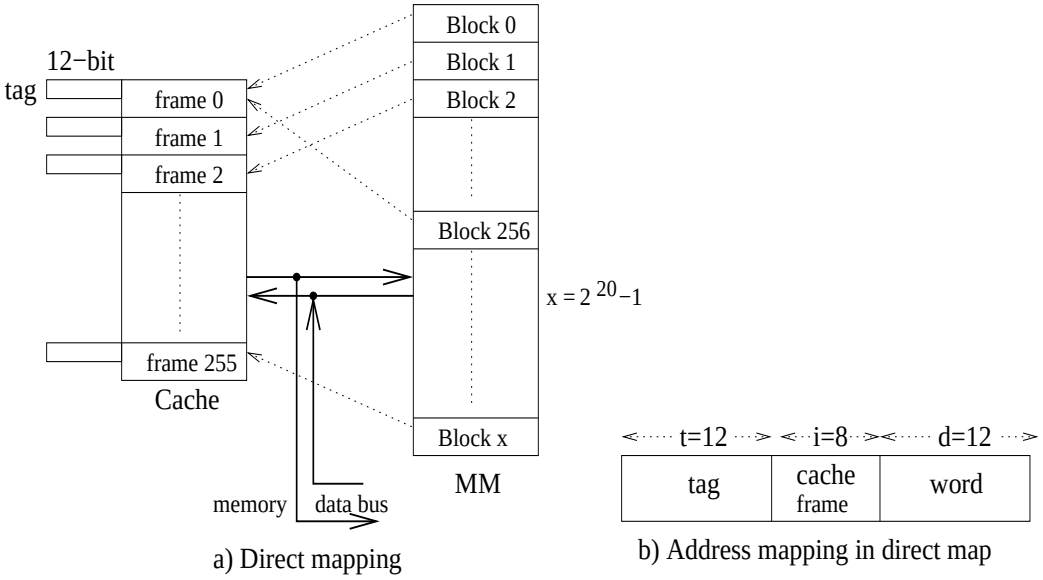
$$\frac{2^{20}}{2^{12}} = 2^8 = 256$$

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

Cache frame 0 can be occupied by MM blocks 0, 256, 512, $\cdots$.

It defines a 12-bit tag is to be associated with each cache frame.

It signifies currently residing block out of 4K MM blocks competing for the frame.



Memory address generated by CPU is of 32 bits.

Its most significant 12-bit is tag (Figure 7(b)).

Next 8-bit determines designated frame of the cache where the item can be found.

Least significant 12-bit determines actual position of the item within the block.

Associative mapping

Here, a MM block can potentially reside in any cache frame (Figure 8).

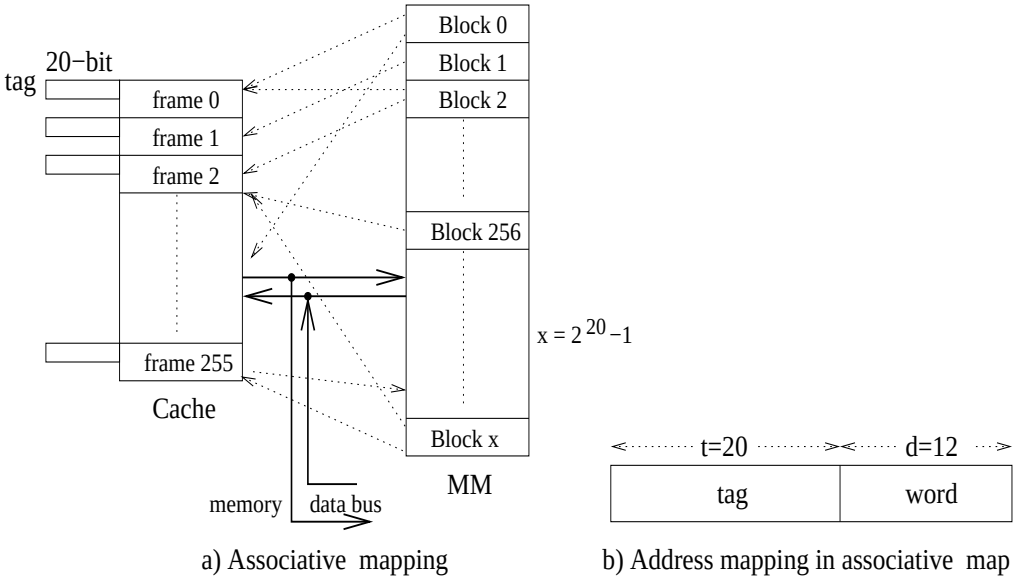


Figure 8: Associative mapping technique

So, if 1M main memory block is competing for 256 cache frames,

20 tag bits are required to identify an MM block in cache.

While CPU generates 32-bit address to get item x,

most significant 20 bits of address is compared with tag of each cache.

A match (F) signifies that desired memory block is present in cache frame F.

Finally, least significant 12-bit of address determines actual position of item x in F.

If there is no match, respective block is transferred to a free frame.

m-way set-associative mapping

If we realize m-way associative mapping for same example,

then it is $m\times$4K to m. A set of m frames of cache forms a group.

Figure 9 shows 2-way set-associative mapping. It is an 8K to 2 mapping.

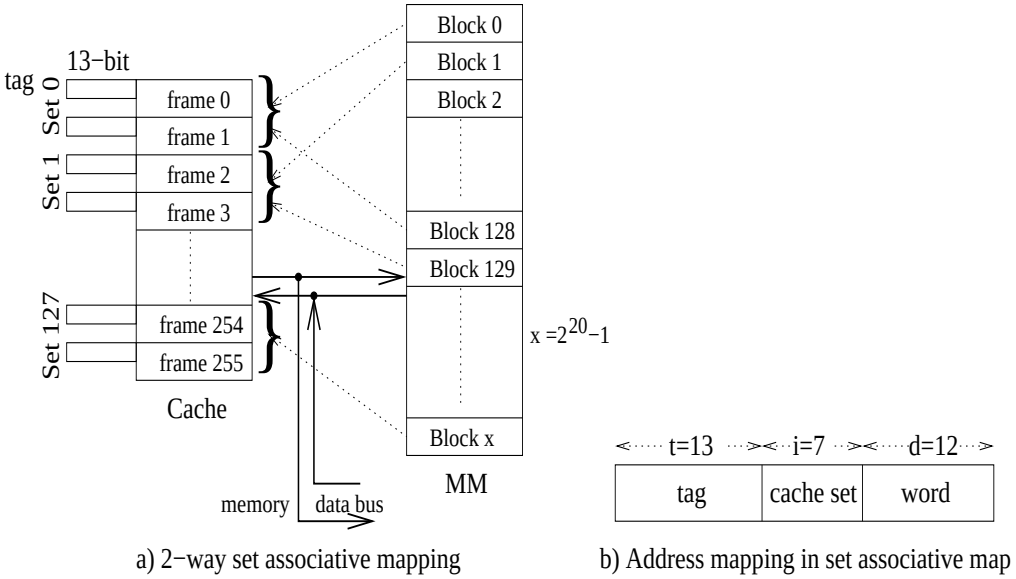


Figure 9: m-way set associative mapping technique

It allows a block of MM to reside in any frame of a set of 2 specified for the block.

In Figure 9(a), MM is of 2^{32} bytes, block size is 4KB and cache is of 1MB = 2^{20}byte.
Number of sets is = 128 = 2^7.

That is,

$\frac{2^{20}}{2^7} = 2^{13}$ MM blocks are competing for a set.

This implies tag t = 13 (Figure 9).
The 7-bit set - called *index* field.

Index determines which set of cache might contain MM block of the item referred.

Comparison of mapping techniques is in Figure 10.

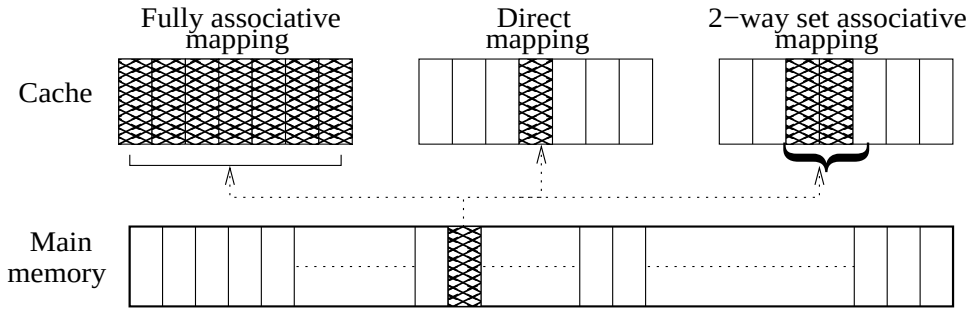


Figure 10: Cmparison of mapping techniques

0.1.3 Cache write policies

a) *write-back*: If CPU modifies data in cache, there is no change in MM.

Data in cache can be written in MM only when block is considered for replacement.

b) *write-through*: While writing at cache, similar write operation is done in MM.

It signifies - each write operation in cache is a write miss.

The Requirement

Programmers demand - access to unlimited fast memory.

Virtual memory system creates illusion that a m/c is of unlimited very fast memory.

This is realized with introduction of memory hierarchy.

0.2 Improving Cache Performance

Average memory access time

AMAT/$t_{effective}$ = hit-ratio $\times$ hit-time + miss-rate $\times$ miss-penalty.

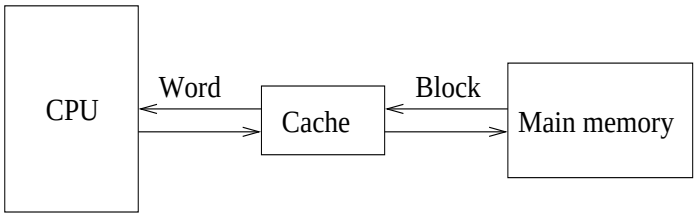
That is,

$$t_{effective} = \text{h} \times t_{cache} + (1\text{-h}) \times t_2$$

t_{cache}: cache hit time,

t_2: miss penalty.

Miss penalty: time to transfer block from MM to cache + item from cache to CPU.



To reduce $t_{effective}$, we can have following options:

1. Reducing miss rate (1-h),
2. Reducing miss penalty (t_2),
3. Reducing hit time (t_{cache}).

0.3 Reducing miss rate

Three categories of misses (3Cs) - compulsory, conflict and capacity miss.

a) **Compulsory miss**

Very first access to a block B is compulsory miss.

Also known as cold start miss.

It cannot be avoided (Figure 11).

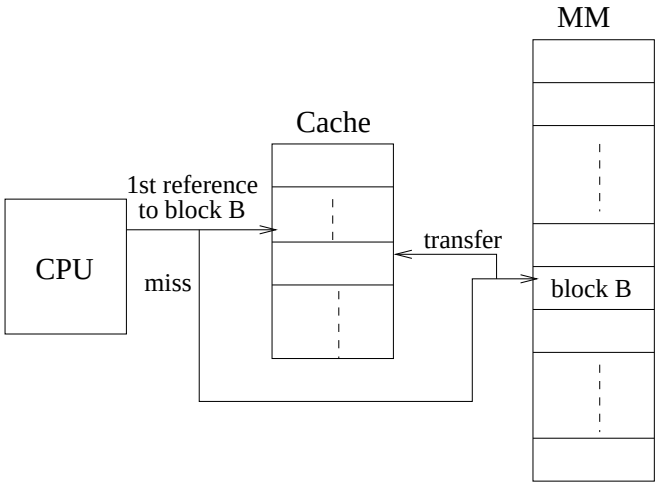


Figure 11: Compulsory miss

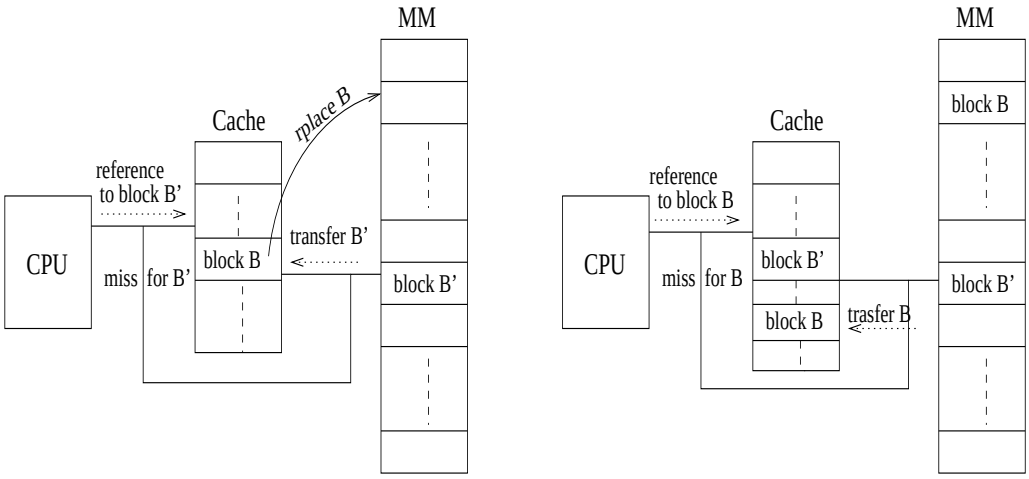
b) **Capacity miss**

Cache cannot contain all blocks needed during execution of a program.

Cache is of limited capacity.

Block B is replaced to find space for incoming block B' (Figure 12(a)).

Next reference to B results in a miss - it is a case of capacity miss (Fig 12(b)).



a) Block B' is referred: B' is not in cache and no free space in cache

b) Block B is referred: B is not in cache (miss)

Figure 12: Capacity miss

c) **Conflict miss**

Multiple memory blocks may map to same cache frame.

In direct mapping, multiple blocks mapped to single cache frame (Figure 13).

This causes a conflict miss.

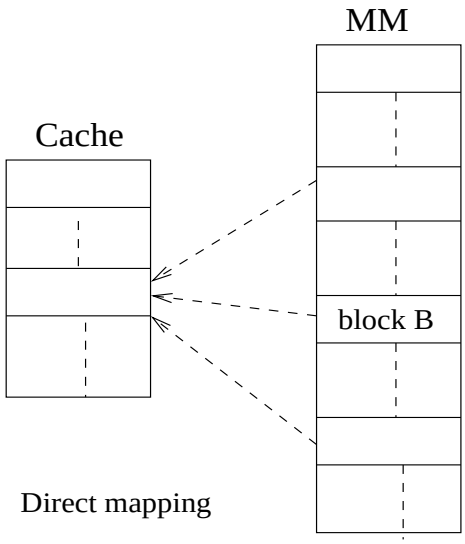


Figure 13: Conflict miss

There is a 4^{th} C (cache coherence). It also affects miss rate.

Block size, higher associativity etc. affect the 3Cs.

QUIZ