ELEC 263 COMPUTER ARCHITECTURE AND ORGANIZATION

Dr. Mohamed Al-Meer
403-4275
almeer@qu.edu.qa, almeerqatar@gmail.com

CHAPTER 12:
MEMORY ORGANIZATION

12-5 Cache Memory

- Analysis has shown that references to memory at given interval of time is confined within few localized areas in memory. (Locality of Reference)
- Programs has loops and repeated subroutine calls encountered frequently. So computer repeatedly refers to set of instructions in memory of the loop
- Same applies for subroutine; every time a subroutine is called the instructions of teh subroutine will be executed.
- Memory reference of data also tend to be localized . lookup tables data repeatedly refer to same area in memory.

"Temporal Locality"

- The information which will be used in near future is likely to be in use already(e.g. Reuse of information in loops)

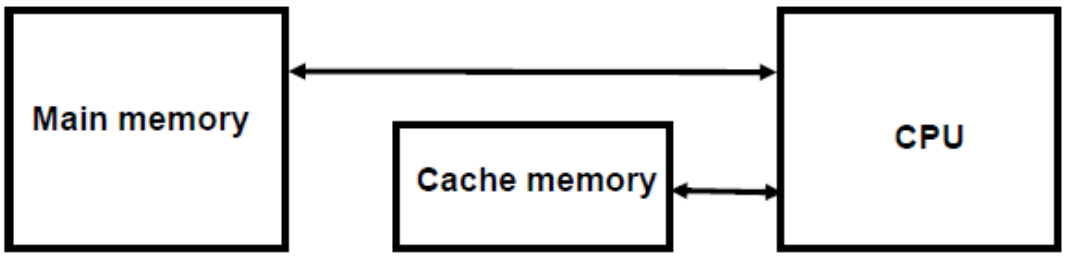
"Spatial Locality"

- If a word is accessed, adjacent(near) words are likely accessed soon
    - (e.g. Related data items (arrays) are usually stored together;

- instructions are executed sequentially

"Cache Memory"

- The property of Locality of Reference makes the Cache memory systems work
- Cache is a fast small capacity memory that should hold those information which are most likely to be accessed.
- Then the average memory access time can be reduced dramatically
- Placed between processor and main memory
- Have access time less than main memory access time by a factor of (5 – 10)



"Cach and memory Access"

- All the memory accesses are directed first to Cache. If the word is in Cache; Access cache to provide it to CPU.
- If the word is not in Cache; bring a block (or a line) including that word to replace a block now in Cache. Block varies between (1 – 16 words)
- Two important questions must be taken care of:
    1. How can we know if the word that is required is there?
    2. If a new block is to replace one of the old blocks, which one should we choose?

"Cache Memory system Performance"

Hit Ratio :

"% of memory accesses satisfied by Cache memory system"

- If processor searches for a word in cache and finds it then we call this state as "hit"; in the same way, if the word is not found in cache then must it be pulled from main memory and placed in cache, then we call this state as "miss"

Te = Tc + (1 -h) Tm

Where:

Te: Effective memory access time in Cache memory system
Tc: Cache access time
Tm: Main memory access time

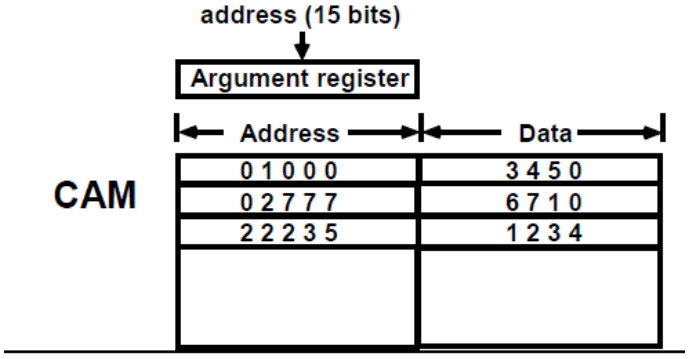
Example: Assume a cached processor system with cache access time of Tc = 0.4 us, and a memory access time of Tm = 1.2us, and cache hit ratio of h = 0.85%. Then get the effective access time?

Answer:

Te = 0.4 + (1 -0.85) * 1.2 = 0.58 us

"Mapping Function: Associative Mapping"

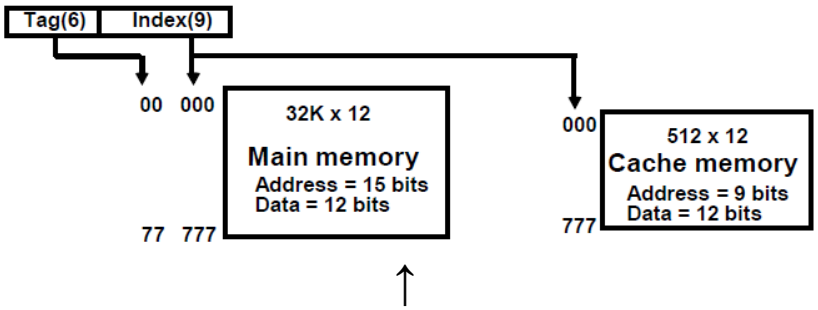
- Any block location in Cache can store any block in memory
  - Most flexible
- Mapping Table is implemented in an associative memory
  - Fast, very Expensive
- Mapping Table
  - Stores both address and the content of the memory word.
- Cache controller search for the existence of the address part. If found then content is accessed; otherwise the main memory is accessed and the address-data pair is placed in next available place in cache.



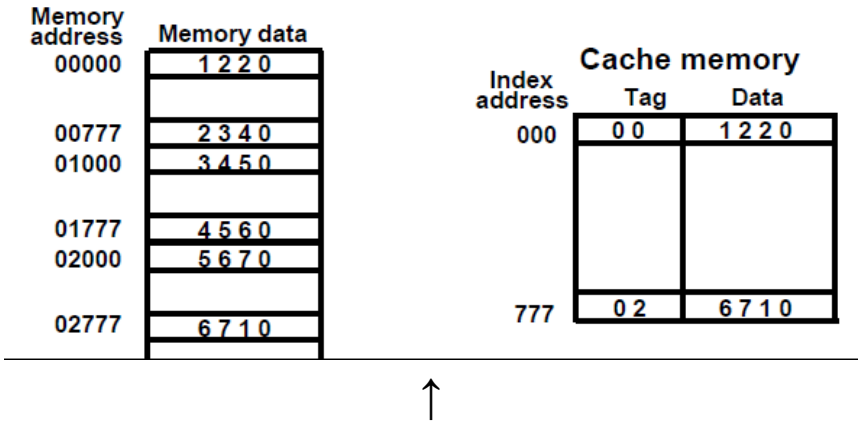
- If cache is full then a decision must be taken to which line in cache to be replaced by the new content (need a replacement algorithm).
    - Like Round-Robin order?

"Mapping Function: Direct Mapping"

- Associative memories are expensive compared to RAMs
- Each memory block has only one place to load in Cache
- Mapping Table is made of RAM instead of CAM
- n-bit memory address consists of 2 parts; (k) bits of Index field and (n-k) bits of Tag field
- n-bit addresses are used to access main memory and k-bit Index is used to access the Cache. Tag part of the address is stored with the data in cache line that it represent.



Addressing Relationship



Direct Map Cache Organization

- Note that location 00000 will be placed in cache in location 000. And its content 1220 will be placed in cache at that address
- Note that content of 02777 which is 6710 will be placed in address 777 in cache.

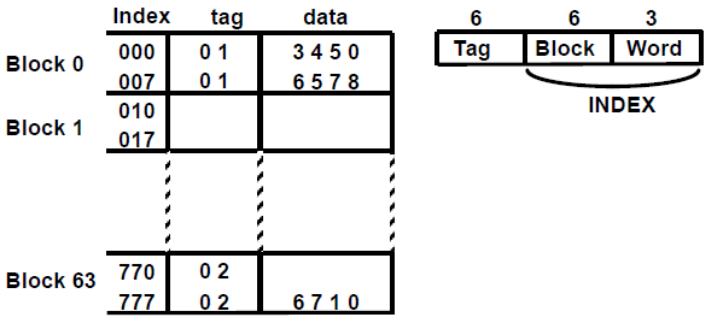
- Note that the next main memory addresses 00000, 01000, and 02000 when referenced will occupy the same cache line address of 000. But they have different TAG parts.

Operation of Direct mapped caches

- address is generated from CPU with 2 parts INDEX and TAG
- cache is accessed using INDEX; the required word is accessed and TAG if compared with cache TAG
- IF matches then it is a "Hit"
    - Data is pulled from this line to processor
- If not a match then it is a "Miss"
    - The required address content is read into cache. ?It may replace an old content with the same INDEX but not the TAG
    - A copy is presented to the processor

Example:

- Next figure shows a direct mapping with block size = 8 bytes
- Each line in cache stores TAG and DATA
- Data will be content of 8 bytes addressed sequentially in main memory
- We have 64 cache lines
- TAG stored with data ion cache line is the remaining address of memory location after the 3 bits word and 6 bits block



Disadvantage

- Hit ratio may drop if 2 or more address words whose address has the same INDEX but different TAG are accessed repeatedly.

- 2 main memory address can have the same index part but different tag part will occupy the same cache line

Cache Write policies

Write Through

- Memory is always updated
- When writing into memory
    - If Hit, then both cache and memory is written in parallel
    - If Miss, then memory is written
    - For a read miss, missing block may be overloaded onto a cache block
- Slow as memory is always accessed

Write-Back (Copy-Back)

- When writing into memory
    - If Hit, only Cache is written
    - If Miss, missing block is brought to cache and write into Cache

- For a read miss, replaced block must be written back to the memory. And this is the only time the block is updated.
- Memory is not up-to-date, i.e., the same item in cache and memory may have different value