CS147 - Lecture 19

Kaushik Patra
(kaushik.patra@sjsu.edu)

1

- Memory Hierarchy & Characteristics
- Introduction to Cache Memory

Reference Books / Source:

1) *Chapter 4 of 'Computer Organization & Architecture' by Stallings*
2) *Chapter 7 of 'Computer Organization & Design' by Patterson and Hennessy/*

Memory Hierarchy & Characteristics ...

2

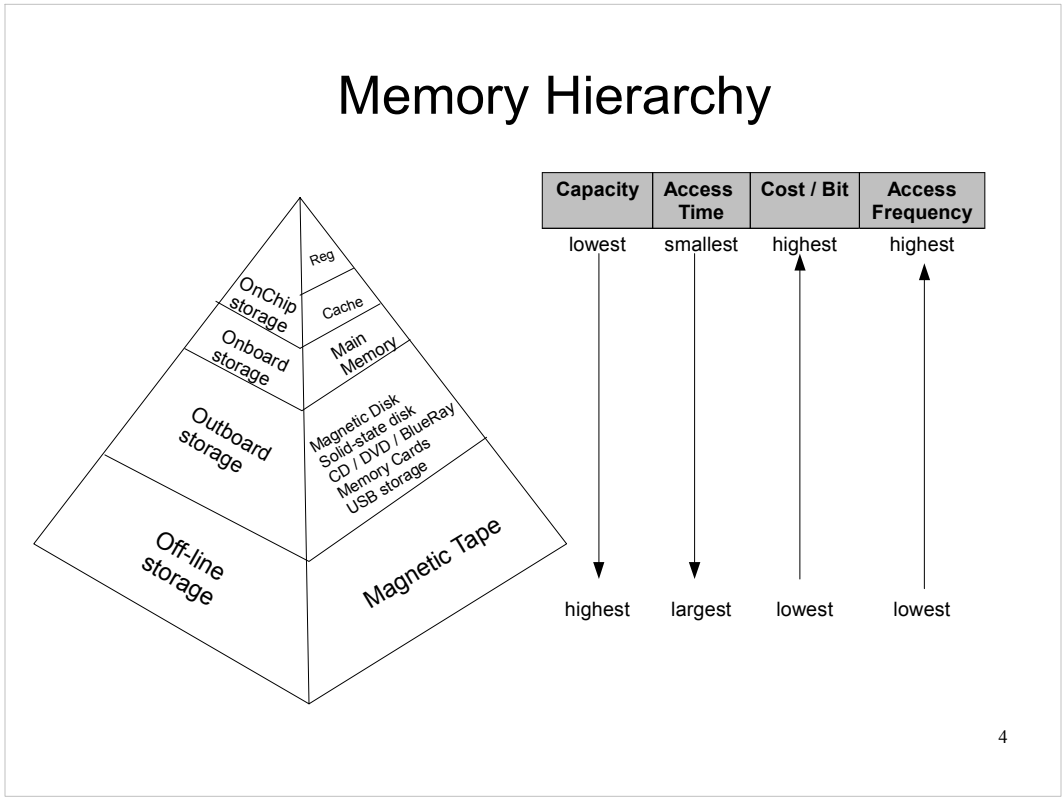
Storage / Memory Avatars

3

- Storage technology has been evolved in parallel with processor evolution. We can see multiple varieties of storage system around us – some of them are very distinguishable externally (e.g. hard disk, CD/DVD, memory cards, USB memory stick, etc.) and others are not so distinguishable (e.g. RAM, Cache-memory, EPROM, etc.)

- In early days semiconductor memories were considered to be internal to a computer. Now a days semiconductor memory can be found outside of computer system as add-on (e.g. memory card, solid-state disk drive).

- In early days, storage technology revolved around magnetic media (tape, floppy disk, hard-disk, etc.). However, in this era, all these magnetic medias are gradually replaced by semiconductor storage media as online storage (or live storage). Magnetic storage mediums are used more as the off-line storage (or backups).



• Storages are classified into three groups depending on their location on whole system.

- In-board storages are the storage available on board and on chip. Register file, cache and main memory are included in this category. They are usually volatile in nature – means that the memory content is gone once power is off (except a little part of the internal memory ROM which is needed for the system boot up process).
- Out-board storages are the storages that are attached to the system externally (and sometimes temporarily too). Hard disk (magnetic or solid state), CD, DVD, BlueRay, memory cards, USB storages are included in this category. They are permanent in nature – means that the memory (alternately they are called storage) content is not gone with power down of the whole system. It retains all it contents to be used one the system is powered up. It is also online storage in the sense that it takes active part during program execution and data saving / retrieval.
- Off-line storages are the storages that are stored away from system as permanent record. Magnetic tapes are usually used for this purpose. In industry the whole computing system is backed up daily / weekly on tape device. The tapes are marked with date and stored away in secured warehouse for record keeping. Also, system can be restored with this known good revision of backup (e.g. system is down with virus / malware attack).

• With respect to the location relative to the processor, register file is the closest and magnetic tape archiving system is the farthest. We say that the register file is the upper most level and the magnetic tape is at the lowest level in the memory hierarchy. In terms of capacity, lowest level in the hierarchy can store highest amount of information. As we go upward in the hierarchy, capacity diminishes. The lowest capacity is of the register file. However, memory access time, is faster at the upper level and slower at the lower level. In addition, upper level memory is costlier (in terms of dollars / bit) and lower level memory is cheaper. In the system, upper level memories are accessed much more frequently that lower level memory.

Memory Characteristic – Unit Transfer

- **Word**
  - In general the size of integer supported by the machine
- **Addressable units**
  - Can vary – byte, word, block
  - $N = 2^A$; where A is number of bits in address value and N is the number of addressable unit.
- **Unit of transfer**
  - Number of units can be transferred with single operation.

5

• A unit of information related to memory is usually the integer storage size in terms of bits. For example, a 32-bit machine has 32-bit unit information or in other terms 32-bit machine has word size of 32-bit. Similarly for a 64-bit machine, word size is 64-bit.

• It is not always true that the memory system provides access at word granularity level. Often the addressable units are wither lower (byte addressable) or higher (like block / page addressable). Usually number of addressable units (N) can be at most 2^A , where A is the number of bits in the address.

• Very often, memory system transfers multiple units at a time (on a single operation). Number of units transferred in one operation is another important characteristic of memory system.

Memory Characteristic – Access Type

- **Sequential**
  - Need to go through every data unit sequentially to reach desired location – e.g. storage tape access.
- **Direct**
  - Can go directly to general vicinity of the location of the data and then data is retrieved by sequential searching – e.g. hard disk / CD-ROM
- **Random**
  - Each addressable location has unique address – fixed access time, e.g. USB storage, flash cards, main memory, etc.
- **Associative**
  - Data is retrieved based on a part of its content – e.g. cache memory

6

• For a sequential access type, storage system needs to go through every data unit till it reaches to the desired data unit. For example, for a tape storage, the reading head must move from the current position through every data unit till the desired data unit is reached (it may need to do forward rewind or reverse rewind). This means the data access time varies from address to address.

• A direct access system is an improvement on top of the sequential storage system. In this system, read mechanism / hardware can go directly into the vicinity of the data requested and then locate requested data sequentially within a smaller boundary. Magnetic (hard disk, floppy disk, etc.) or optical (CD, DVD, BlueRay, etc.) disk system are the examples of direct access system. Direct access system provides faster operation compare to sequential access storage. However the access time still varies from address to address.

• A random access system assigns unique address / location to each addressable data unit. It has fixed access time for every address. All the semiconductor memory / storage (RAM, ROM, Flash memory, etc.) are random access type. This type of memories provides much faster access time compare to direct and sequential access memory.

  • This is a general collection of all types of semiconductor memory. This random access type can be classified further (w.r.t. Specific memory technology) as flash memory, D-RAM (Dynamic RAM), and S-RAM(Static RAM). Each of this memory technology gives different combination of speed, capacity, and volatility. Flash memory has higher capacity, lower speed (compare to RAM), but it is permanent (non-volatile). On the other hand, any type of RAM is volatile. DRAM has higher capacity but lower speed compare to SRAM.

•Associative memory is a very special type of storage technique where data is retrieved based on the requested data content. It is analogous to has table where a data is retrieved based on a key pattern matching. Cache memory is implemented using this type of storage technique.

Memory Characteristic – Performance

- **Access Time (Latency) [T_A]**
  - Time to complete read or write operation from the time point when the address is present to the memory (for Random or associative access memory).
  - Time to place read/write mechanism at the desired place (for non random access memory)
- **Memory Cycle Time [T_C]**
  - Only related to random access memory of certain type (e.g. D-RAM).
  - This is the additional time before next access can be done (refresh time or T_R) plus the access time.
  - $T_C = T_A + T_R$
- **Transfer Rate [T_N]**
  - For random access it is $1/T_C$
  - For non-random access average transfer time for N bits is (T_A + N/R); where T_A is the average access time, N is the number of bits to be transferred, R is the transfer rate in bits per second (***bps***) for the given device. Reciprocal of it is the average transfer rate for non-random access memory.

7

• Performance metrics have different meanings for random access and non-random access memory / storage.

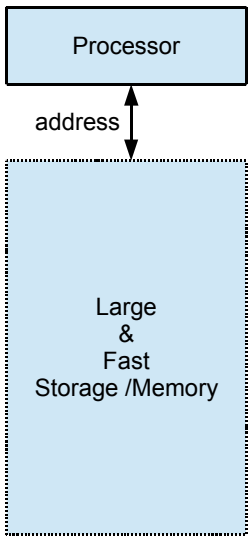
• For a random access type storage the following are the meaning of performance metrics.
  - Access time is the time delay between address of the operation (read or write) is given to memory and the completion of the operation.
  - Some type of memory (especially D-RAM) needs to have refresh cycle to renew its storage information. Over time, D-RAM tends to loose information. Therefore it needs to refresh / renew its storage information. During this refresh time, memory does not allow any other operation and usually it is done after every operation performed. Hence, RAM is not only related to access time (or latency) but also refresh time. The sum of these two metric is called a cycle time for memory – it is the time delay between two successive memory operation.
  - Transfer rate is the number of memory unit transferred to / from the memory per unit time. For a RAM, unit is usually word. Thus it is the reciprocal of cycle time for the memory.

• For a non random access memory following are the meaning of performance metrics.
  - Access time is the time of place read/write mechanism into desired place to perform the operation. Since it is electromechanical in nature, this operation is usually slower (order of milliseconds) compare to random-access memory.
  - To compute total time to transfer Transfer rate is expressed in terms of bits and second. If R is the transfer rate (bits per second) of the device then addition of average access time and (N/R) is the average time to transfer N bits. Reciprocal of this is the average transfer time of N bits is average transfer rate of non-random access memory.

Cache Memory ...

8

Cache – Motivation

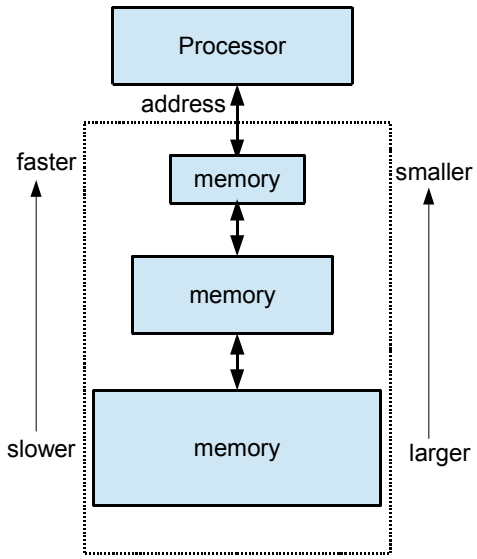


- Programmer always demands two features from memory / storage
  - Faster access
  - Unlimited capacity
- These two features are in contradiction.

9

- To develop complex softwares and algorithm, programmers are always in constant demand for faster unlimited memory – since major part of the program involves in read memory and storing data. However, these two demands are in contradiction. The memory technology which can provide unlimited capacity can not be faster. On the other hand, the memory technology which can give faster access can not provide unlimited (not even large) capacity.

Cache – Solution



- Solution is to create an illusion of having faster speed and larger capacity.
- Two observations from memory hierarchy:
  - Up in the hierarchy (nearer to processor) memory is faster but small.
  - Down in the hierarchy memory is slower but large.

10

• The solution is to create an illusion to the processor of having faster yet large storage capacity. Processor issues address for memory operation and utilize the result out of the operation on the provided address. It is not particularly aware of where from the data is coming (for read operation) or data is going (for write operation). Hence, if it is possible to combine different technology to provide faster unlimited access, there will not be any problem from processor's perspective.

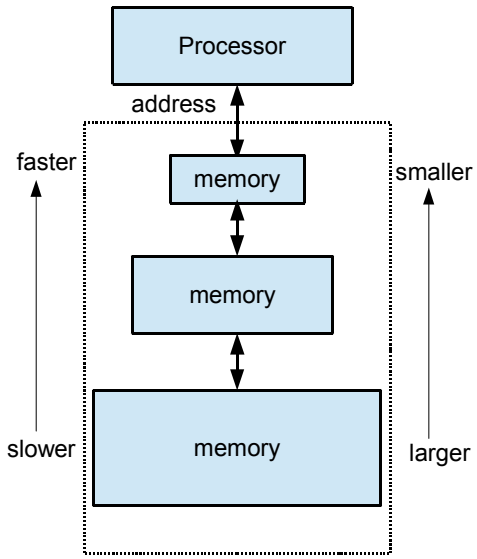
• We have two key observations from the memory hierarchy.

- Technology which can provide faster access are smaller - we can not make them bigger from cost perspective.
- Technology which can provide big storage capacity are slower.

• If we make the processor's primary interaction with the faster memory, we reduce the access time / cycle time for memory operation. It is usually the case that to put faster memory nearer to the processor. We can connect different flavor of memory technology in hierarchical fashion with faster & smaller memory at the top (nearer to processor) and slower & larger memory at the bottom (far from processor) and putting every other type of memory in-between in order of performance and capacity. In this way we present a large storage pool to the processor.

- Now the key question is - can we manage with the smaller memory size of the faster memory?

Cache – Solution



- Information (program and data) could be copied from lower memory to higher memory.
  - Two adjacent levels are involved in information duplication process.
- Issue
  - Since higher memory is smaller, all the information may not fit from lower memory related to a program (code + data).

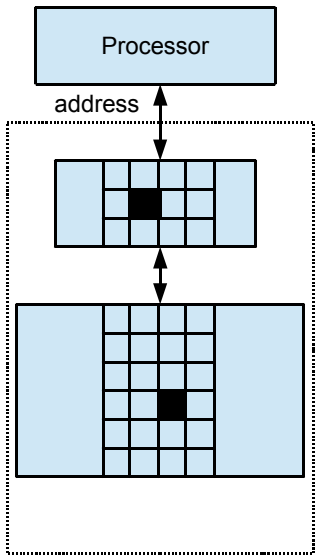
11

- Same information (instruction or data) can be copied between two adjacent level of memories in the hierarchy. In the memory hierarchy data can be copied between two adjacent levels. That says, in this example, data can not be copied between the top most and bottom most memory.

- One observation is that information is duplicated at different level of hierarchies. The bottom most layer is the master copy of the information. All the upper level memory contains copy from its lower level.

- Other observation is that the whole software program and data associated with its execution, may not fit from lower memory to higher memory. Since we are providing illusion of having large capacity, by adding appropriate memory technology at the bottom of the hierarchy, programs should not be restricted to be fit into smaller size of top most memory in the hierarchy.

Cache – Solution



- The unit information copied from lower memory to higher memory is called ***block***.
  - The size of the block is determined by cache architect.
- Upon processor request of information from an address, there are two outcomes.
  - It is present in the higher memory (***cache hit***)
  - It is not present in the higher memory (***cache miss***) and needed to be copied from lower level memory

12

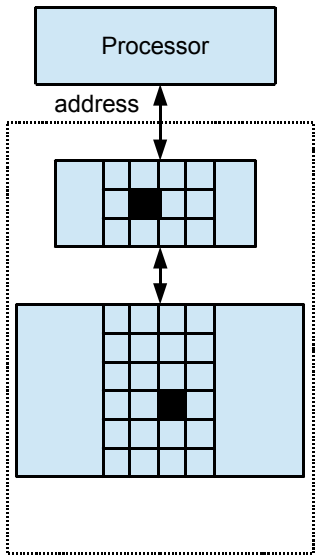
• Since data copy process is restricted between two adjacent memory level, we can restrict our discussion with two levels in a memory hierarchy.

• The unit information copied from lower memory and back may not be just one word. It may be multiple word and the unit is called a block. The size of block is determined by cache architect.

• When a processor request for a data from / to an address, it may not present in the upper most memory in the hierarchy. If it is not present, it is called a cache miss. If such miss occurs, corresponding block must be copied from lower memory to higher memory. During this copy process, the processor usually goes into wait (the total time to complete the block operation is called the *miss penalty*). However, there are many other techniques (like out of order execution or hardware threading) to keep the processor busy and comes back to current memory operation once the copy between memories are done.

• When a processor request for a data from / to an address, if the corresponding block is present in the upper most memory, it is called a cache hit. There is no associated penalty for a cache hit.

Cache – Solution

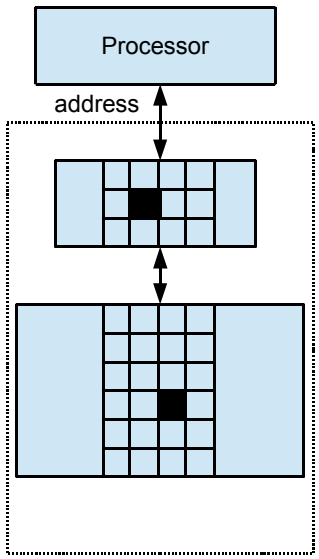


- The ***hit rate*** or ***hit ratio*** is the fraction of memory access found in higher level memory.
- The ***miss rate*** or ***miss ratio*** is the fraction of memory access not found in the upper level.
  - (1- hit rate)
  - Associated with a miss penalty, a time during which required memory block is transferred from lower memory to higher memory.

13

• Hit rate or hit ratio is very important metric to express cache effectiveness. It is defined as the fraction of time memory access is found in the top most memory. The higher this hit rate, more effective is the cache. This is because, a miss causes expensive copy operation between memory.

Cache – Solution



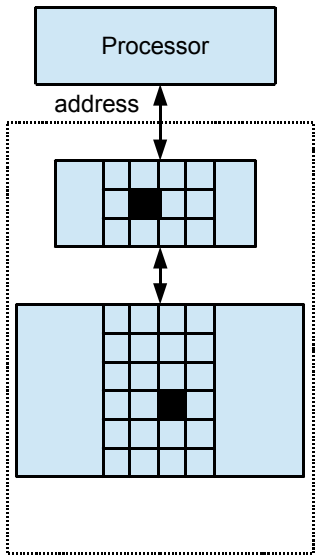
- For a system, upper level memory cycle time is 1ns and miss penalty is 510ns. A processor in this system is executing 1K R-type instruction (which is accessing memory for fetching just the instructions). Compute total time of instruction fetch operation with hit rate of 0.8 and 0.4.
  - 103.0us for 0.8 hit rate
  - 307.0 us for 0.4 hit rate
- Degradation in hit rate can severely degrade the over all system performance.
  - In this example, 2x degradation in hit rate degrading instruction fetch by almost 3x.

14

• For hit rate 0.8
  - 0.8*1K instructions were found in the cache. Hence total time to fetch this fraction of instructions is (800*1ns) = 800ns.
  - 0.2*1K instructions were not found in the cache. Therefor 200 times the operation incurred miss penalty of 510ns. Hence for these 200 instructions, fetch time for each instruction is (510 + 1)ns = 511ns. So total time to fetch these 200 instruction is (511*200)ns = 102200ns.
  - Hence total time to fetch 1K instructions is (800 + 102200)ns = 103000ns = 103.0us

• For hit rate 0.4
  - 0.4*1K instructions were found in the cache. Hence total time to fetch this fraction of instructions is (400*1ns) = 400ns.
  - 0.6*1K instructions were not found in the cache. Therefor 600 times the operation incurred miss penalty of 510ns. Hence for these 600 instructions, fetch time for each instruction is (510 + 1)ns = 511ns. So total time to fetch these 600 instruction is (511*600)ns =306600ns
  - Hence total time to fetch 1K instructions is (400 + 306600)ns = 307000ns = 307us.

Cache – Solution



- What keeps the higher hit rate in the system?
  - Temporal locality
  - Spatial locality

```
1            I=0;
2            N = 1000;
3     Loop : C[I] = A[I] + B[I];
4            I = I + 1;
5            if (I<N) goto Loop;
```

- Line 3,4 was executed 1000 times.
- Line 5 was executed 1001 times.
- Array C, A and B are accessed 1000 times.
- I is accessed 5001 times
- N is accessed accessed 1002 times.

15

• The observation from previous example is that the degradation hit rate can severely affect the overall performance. Therefore, hit rate should be kept as high as possible during execution of a software program.

• There are two observations in software execution that makes the cache technique useful in creating illusion of high performance high capacity storage system by keeping the hit ratio higher.

- Temporal locality – it says that if an item is referenced, it will tend to be referenced again soon. e.g. with in a loop, same instructions are executed multiple times with close temporal proximity. In the above example, variable I and N are referenced multiple times within close temporal proximity, thus exhibits a temporal locality.

- Spatial locality – it says that if an item is referenced, item whose addresses are close by will tend to be referenced soon. e.g. adjacent instructions in a program are executed with close temporal proximity. In the above example, array elements of A, B and C are referenced in close temporal proximity, thus exhibits a spatial locality.

CS147 - Lecture 19

Kaushik Patra
(kaushik.patra@sjsu.edu)

16

- Memory Hierarchy & Characteristics
- Introduction to Cache Memory

Reference Books / Source:

1) *Chapter 4 of 'Computer Organization & Architecture' by Stallings*
2) *Chapter 7 of 'Computer Organization & Design' by Patterson and Hennessy/*