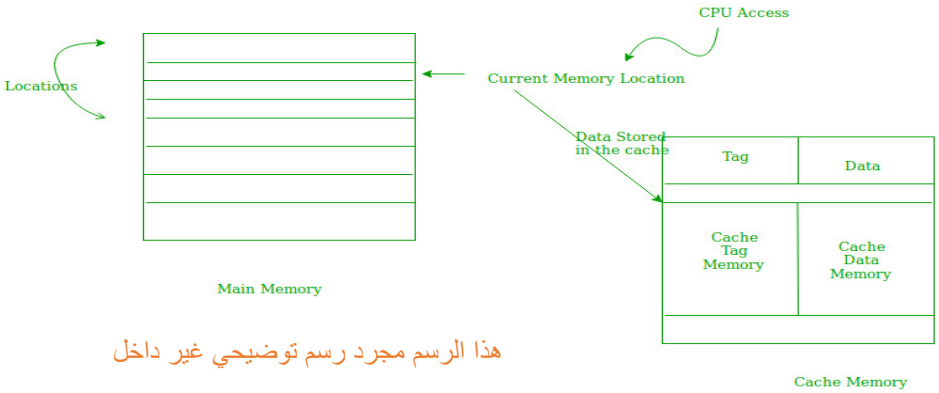
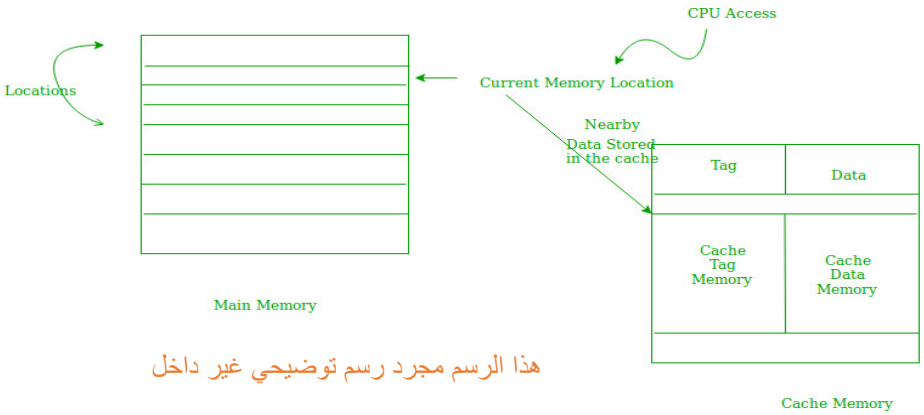
PART 3

Temporal Locality – Temporal locality means current data or instruction that is being fetched may be needed soon. So, we should store that data or instruction in the cache memory so that we can avoid again searching in main memory for the same data.



هذا الرسم مجرد رسم توضيحي غير داخل

Spatial Locality – Spatial locality means instruction or data near to the current memory location that is being fetched, may be needed soon in the near future. This is slightly different from the temporal locality. Here we are talking about nearly located memory locations while in temporal locality we were talking about the actual memory location that was being fetched.



هذا الرسم مجرد رسم توضيحي غير داخل

1) **Location**

1. CPU registers
2. Cache
3. Internal Memory(main)
4. External (secondary)

2) **Capacity**

- **Word size**: Typically, equal to the number of bits used to represent a number and to the instruction length.
- **Bytes:** For addresses of length A (in bits), the number of addressable units is **2^A.**

3) **Unit of Transfer**

1. **Word**
2. **Block**

4) **Access Method**

*** Sequential Access**

- Access must be made in a specific linear sequence.
- Time to access an arbitrary record is highly variable.

*** Direct access**

- Individual blocks or record have an address based on physical location.
- Access is by direct access to general vicinity of desired information, then some search.
- Access time is still variable, but not as much as sequential access.

*** Random access**

- Each addressable location has a unique, physical location.
- Access is by direct access to desired location,
- Access time is constant and independent of prior accesses.

*** Associative**

- Desired units of information are retrieved by comparing a sub-part of unit.
- Location is needed.
- Most useful for searching.

5) **Performance**

*** Access Time (Latency)**

- For random access memory, latency is the time it takes to perform. A read or write operation that is the time from the instant that the address is presented to the memory to the instant that the data have been stored available for use.

*** Memory Cycle Time**

- Access time + additional time required before a second access can begin(refresh time, for example).

*** Transfer Rate**

- Generally measured in bit/second.

Typical Members of the Memory Hierarchy

Some characteristics of key elements of the memory hierarchy:

- The fastest, smallest, and most expensive type of memory consists of the registers internal to the processor.
- Next will be typically multiple layers of cache. Level 1 cache (L1 cache), closest to the processor registers, is almost always divided into an instruction cache and a data cache. This split is also common for L2 caches. L3 cache and some have an L4 cache.
- Main memory is the principal internal memory system of the computer. Each location in main memory has a unique address. Main memory is visible to the programmer, whereas cache memory is not.
- External, nonvolatile memory is also referred to as secondary memory or auxiliary memory. These are used to store program and data files and are usually visible to the programmer only in terms of files and records, as opposed to

Memory level	Typical technology	Unit of transfer with next larger level (typical size)	Managed by
Registers	CMOS	Word (32 bits)	Compiler
Cache	Static RAM (SRAM); Embedded dynamic RAM (eDRAM)	Cache block (32 bytes)	Processor hardware
Main memory	DRAM	Virtual memory page (1 kB)	Operating system (OS)
Secondary memory	Magnetic disk	Disk sector (512 bytes)	OS/user
Offline bulk memory	Magnetic tape		OS/User

As one goes down the hierarchy, the following occur:

a. Decreasing cost per bit

b. Increasing capacity

c. Increasing access time

d. Decreasing frequency of access of the memory by the processor

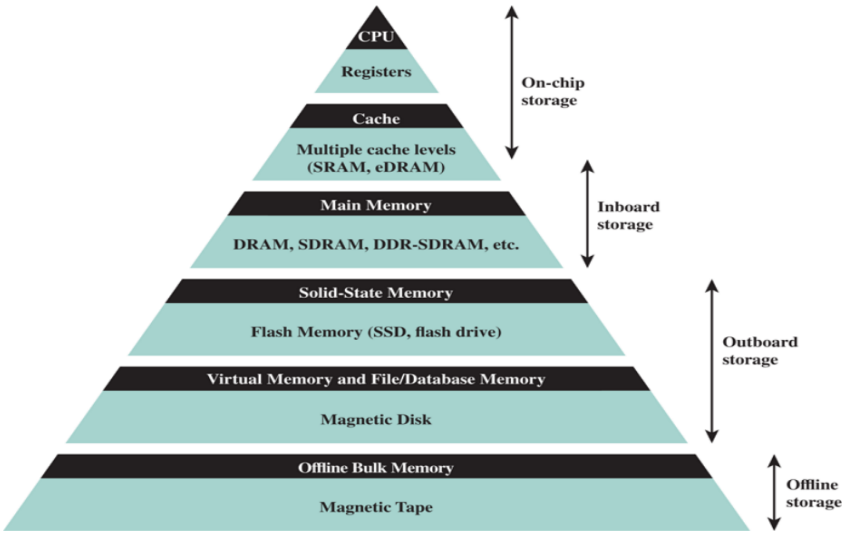


Figure 3.2 The Memory Hierarchy

Design Principles for a Memory Hierarchy

- Three principles guide the design of a memory hierarchy :

1. **Locality:** Locality is the principle that makes effective use of a memory hierarchy possible.
2. **Inclusion:** This principle dictates that all information items are originally stored in level M_n, where n is the level most remote from the processor. During the processing, subsets of M_n are copied into M_{n-1}. similarity, subsets of M_{n-1} are copied into M_{n-2}, and so on. This is expressed concisely as $M_i \subseteq M_{i+1}$. Thus, this is in contrast to our simple example of **Figure 4.1**, where Bob moved a folder from the file cabinet to his desk. With the memory hierarchy, units of data are copied rather than moved, so that the data unit that is moved to M_i remains in M_{i+1}. Thus, if a word is found in M_i, then copies of the same word also exist in all subsequent layers $M_{i+1}, M_{i+2}, \ldots, M_n$.
3. **Coherence:** Copies of the same data unit in adjacent memory levels must be consistent. If a word is modified in the cache, copies of that word must be updated immediately or eventually at all higher levels.

his leads to two requirements:

Vertical coherence: If one core makes a change to a cache block of data at L2, that update must be returned to L3 before another L2 retrieves that block.

Horizontal coherence: If two L2 caches that share the same L3 cache have copies of the same block of data, then if the block in one L2 cache is updated, the other L2 cache must be alerted that its copy is obsolete.