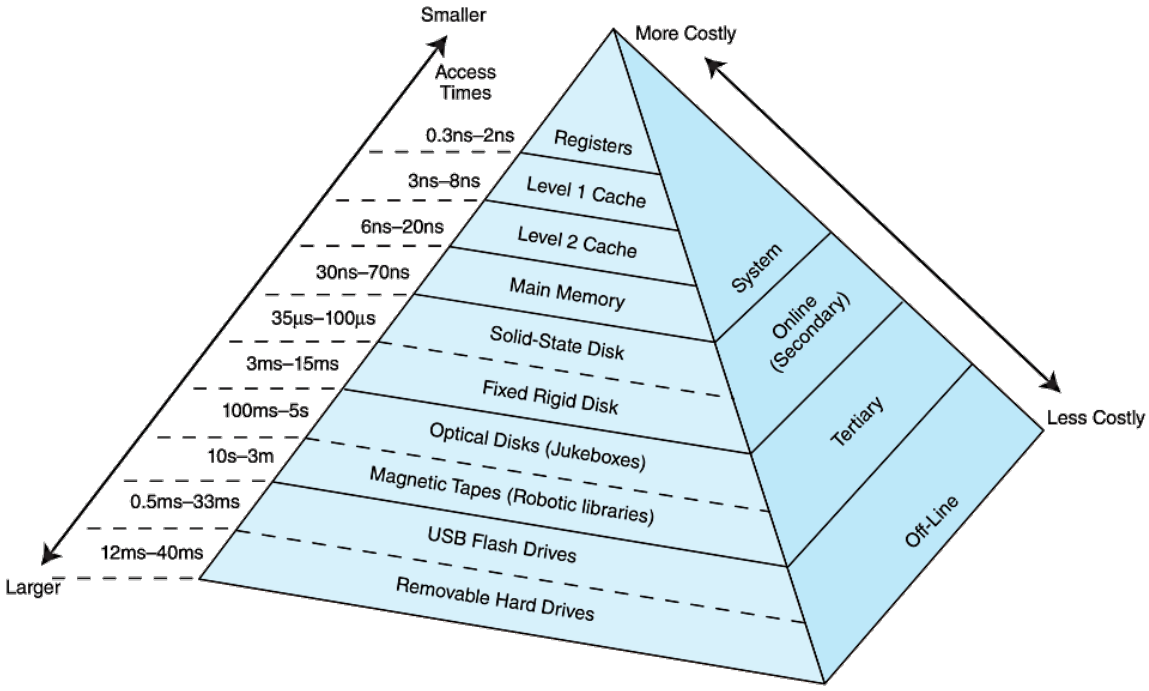
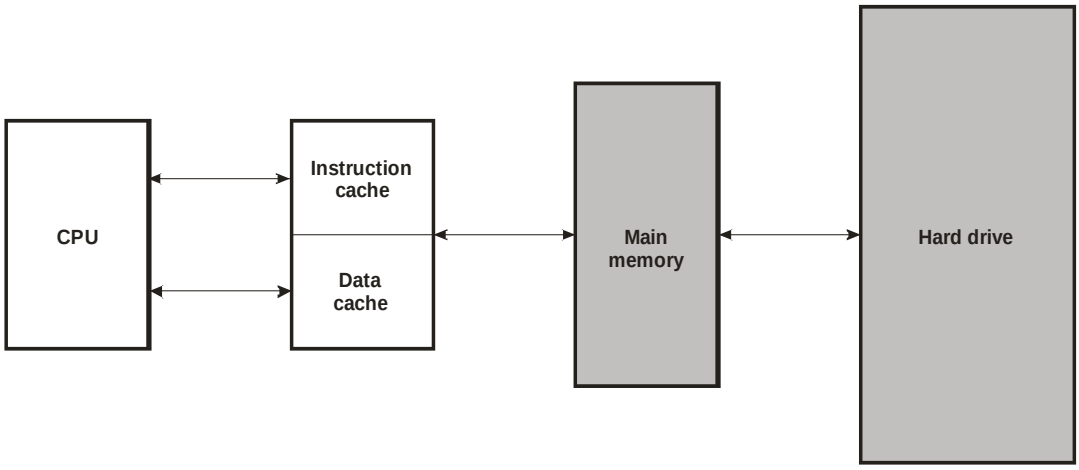


Figure 6.1 The Memory Hierarchy

Term	Definition
Hit	The requested data resides in a given level of memory. For example, a c*ache-hit* occurs when data are found at a particular address in *cache*.
Miss	The requested data are not found in the given level of memory. For example, a page fault refers to the event that when data in the main memory is not resident and must be copied from hard disk.
Hit rate	The probability that data at an address will be found in a given level of memory. For example, a cache hit rate of 0.995 means that 995 times out of 1000 data will be found in the cache.
Miss rate	The probability that data at an address will **NOT** be found in a given level of memory. The miss rate is equal to 1 – hit rate. For example, given a cache hit rate of 0.995, the cache-miss rate is 0.005 or five times out of 1000 memory accesses data are not stored in the cache.
Hit time	The time required to access the requested information in a given level of memory.
Miss penalty	The time required to process a miss, which includes replacing a block in an upper level of memory, plus the additional time to deliver the requested data to the processor. (The time to process a miss is typically significantly larger than the time to process a hit.)



Memory Hierarchy

- Two caches permits one instruction and one operand to be fetched, or one instruction and operand to be stored, in a single clock cycle if the caches are fast enough.
- To make caches faster, we design the cache to be internal to the CPU, so that no off-chip accesses are required for most memory references. Typically designers require more cache memory than technology will support on the CPU chip. Thus, caches, themselves are divided into on-chip and off-chip partitions.

Access Speed

Symbol	Typical Value	Definition
$\boldsymbol{P_C}$	0.95	Probability that the address referenced is in *cache*
$\boldsymbol{P_M}$	$1 - 5 \times 10^{-7}$	Probability that the address referenced is in *main* memory.
$\boldsymbol{T_C}$	2 ns	Cache access time
$\boldsymbol{T_M}$	10 ns	Main store access time
$\boldsymbol{T_D}$	13 ms	Hard disk access time
$\boldsymbol{T_{eff}}$		Memory system effective access time

1. Simple: assume that the memory system only includes cache and main store. Assume that $P_M = 0.05$

$$T_{eff} = P_C T_C + (1 - P_C) T_M$$
$$T_{eff} = 0.95 \times 2ns + (1 - 0.95) \times 10ns$$
$$T_{eff} = 1.9ns + 0.5ns = 2.4ns$$

2. Complex: assume that the memory system includes cache, main store and disk.

$$T_{eff} = P_C T_C + (1 - P_C) T_{Meff}$$
$$T_{Meff} = P_M T_M + (1 - P_M) T_D$$

$$T_{Meff} = (1 - 5 \times 10^{-7}) \times 10ns + 5 \times 10^{-7} \times 13ms$$
$$T_{Meff} = (1 - 5 \times 10^{-7}) \times 10ns + 5 \times 10^{-7} \times 13ms = 16.5ns$$

$$T_{eff} = P_C T_C + (1 - P_C) T_{Meff}$$

$$T_{eff} = 0.95 \times 2ns + 0.05 \times 16.5ns = 2.725ns$$

6.3.1 Locality of Reference

Term	Discussion
Locality of Reference	Locality of reference refers to the high probability that the next memory reference will be close to the current memory reference.
Temporal locality	Recently accessed items tend to be accessed again in the near future.
Spatial locality	Accesses tend to be clustered in the address space (for example, as in arrays or loops).
Sequential locality	Instructions tend to be accessed sequentially.

The locality principle enables the functionality of the memory hierarchy. It memory accesses were truly random, then the memory hierarchy would provide no advantage. Without locality of reference, all memory references would be equally likely making the assignment of a block of memory, near to the current reference, to very fast memory, without value.