Memory Hierarchy

- Design constraints on a computer’s memory
  - How much?
  - How fast?
  - How expensive?
- If the capacity is there, applications will likely be developed to use it
- Memory must be able to keep up with the processor
- Cost of memory must be reasonable in relationship to the other components

Memory Relationships

Faster access time = greater cost per bit

Greater capacity = smaller cost per bit

Greater capacity = slower access speed

The Memory Hierarchy

- What if use multiple memory components?
- A memory hierarchy
- We could use a reasonably fast memory system that is also cost effective.

The Memory Hierarchy

- Going down the hierarchy:

a. Decreasing cost per bit
b. Increasing capacity
c. Increasing access time
d. Decreasing frequency of access to the memory by the processor

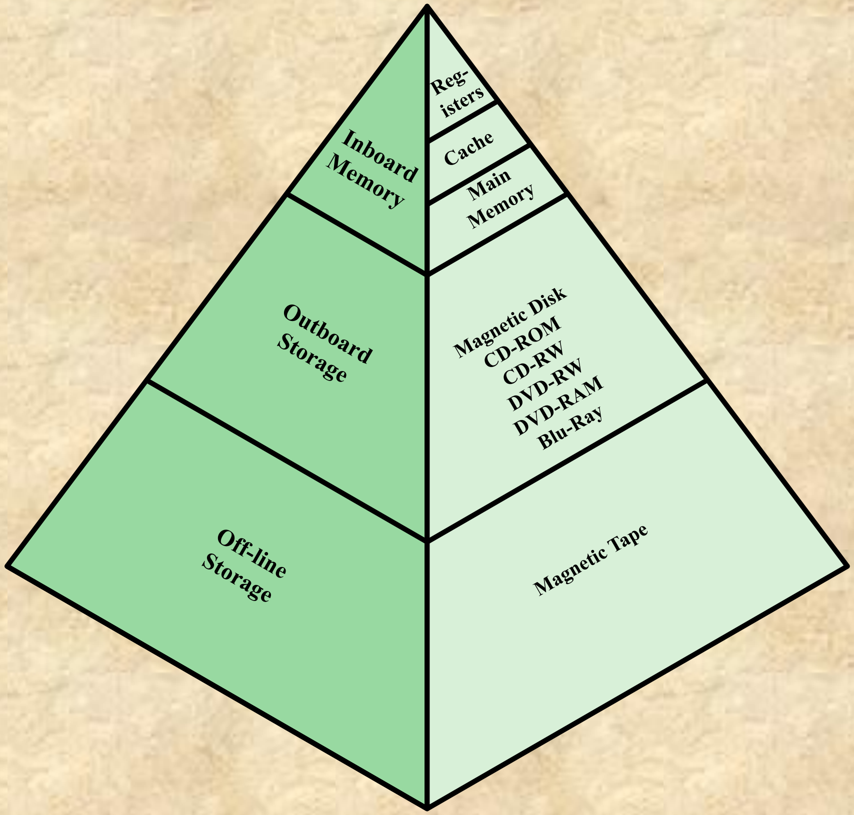


Figure 1.14 The Memory Hierarchy

Hit Ratio (H) and Average Access Time

- Let's do some calculations

- Hypothetical memory system
  - Two levels
  - Level 01 : 1000 bytes : 0.1 μs access time
  - Level 02: 100000 bytes : 1 μs access time
  - H = fraction of memory accesses that are found in fastest memory

- Average access time (A)
  - $A = H \cdot (0.1\ \mu s) + (1 - H) \cdot (0.1\ \mu s + 1\ \mu s)$

- If 95% of memory access is in the level 01, then
  - $0.15\ \mu s = 0.95 \cdot (0.1\ \mu s) + (1 - 0.95) \cdot (0.1\ \mu s + 1\ \mu s)$

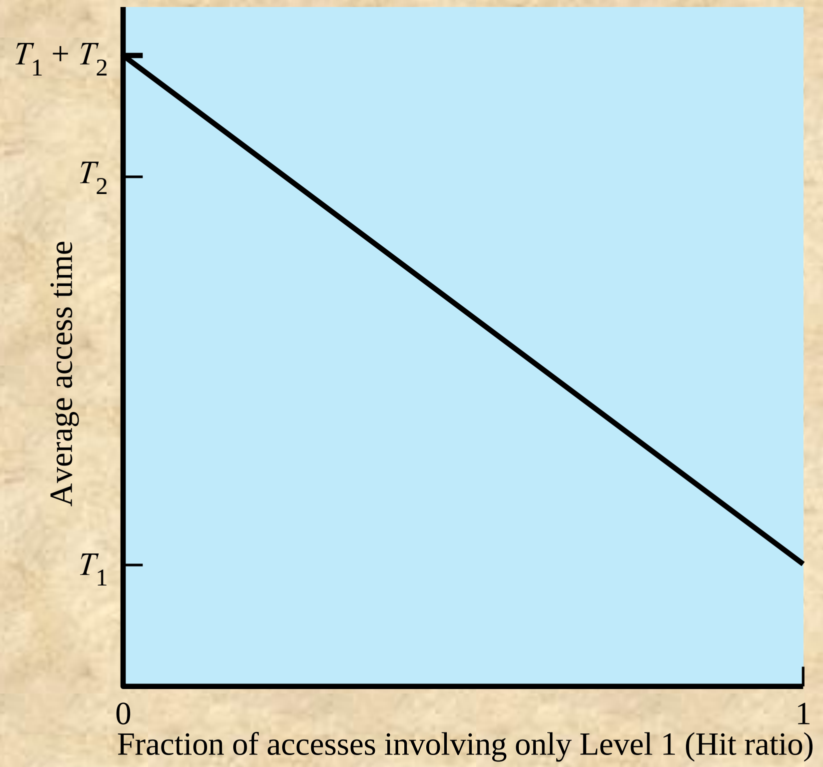


Figure 1.15 Performance of a Simple Two-Level Memory

Memory hierarchy

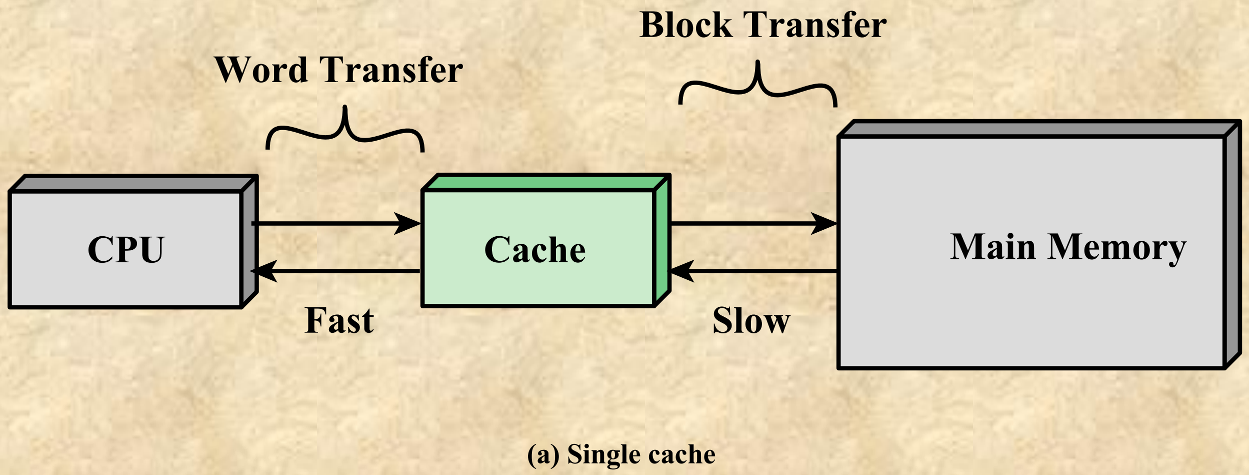
- The average access time is closer to faster memory
- This works in principle, If the conditions (a) to (d) applies
- Condition (d) is generally valid.

Principle of Locality

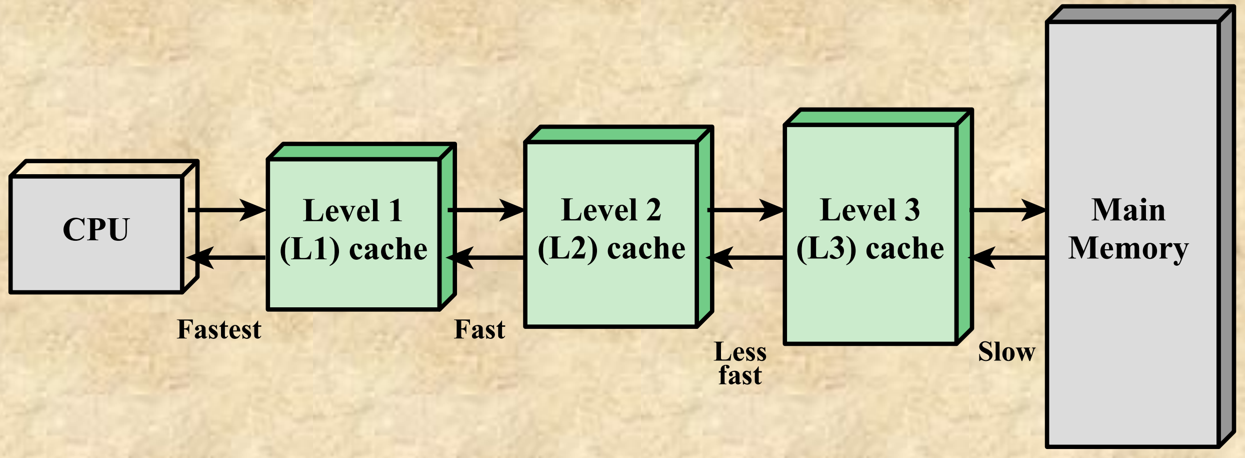
- Memory references by the processor tend to cluster
- Data is organized so that the percentage of accesses to each successively lower level is substantially less than that of the level above
- Can be applied across more than two levels of memory

Cache Memory

- Largely invisible to the OS
- Interacts with other memory management hardware
- Processor must access memory at least once per instruction cycle
- Processor execution is limited by memory cycle time
- Exploit the principle of locality with a small, fast memory

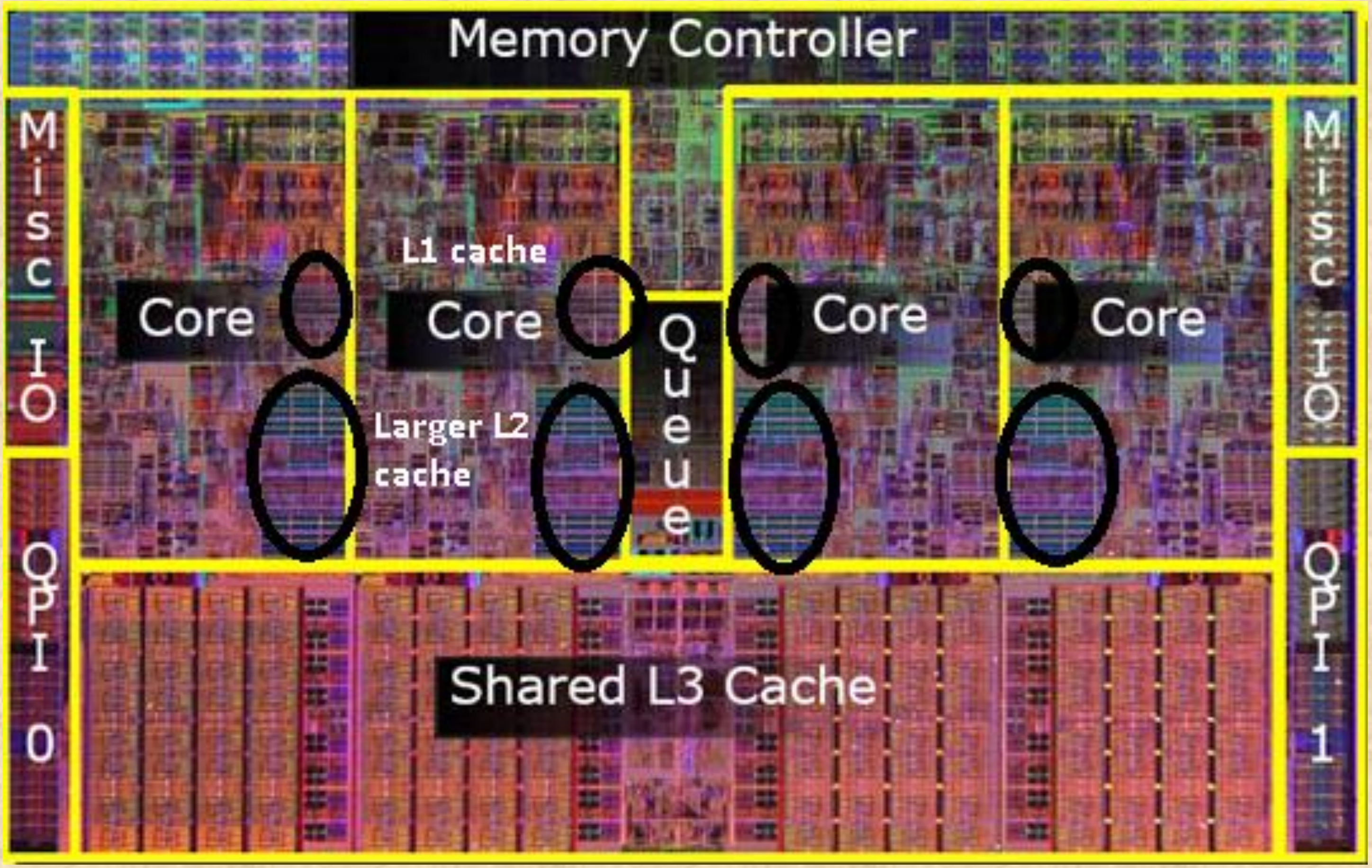


(a) Single cache



(b) Three-level cache organization

Figure 1.16 Cache and Main Memory



https://medium.com/software-design/why-software-developers-should-care-about-cpu-caches-8da04355bb8a

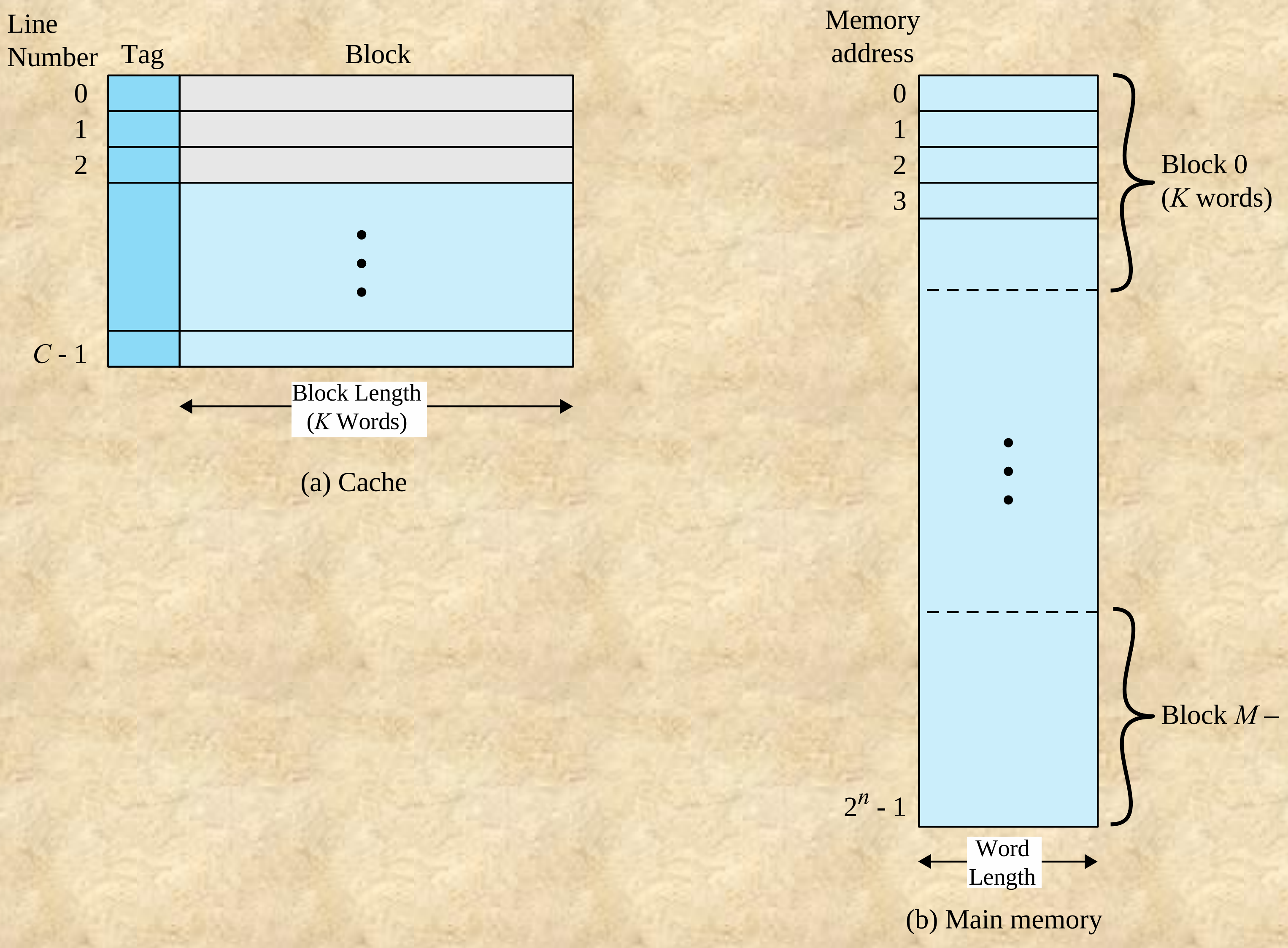


Figure 1.17 Cache/Main-Memory Structure

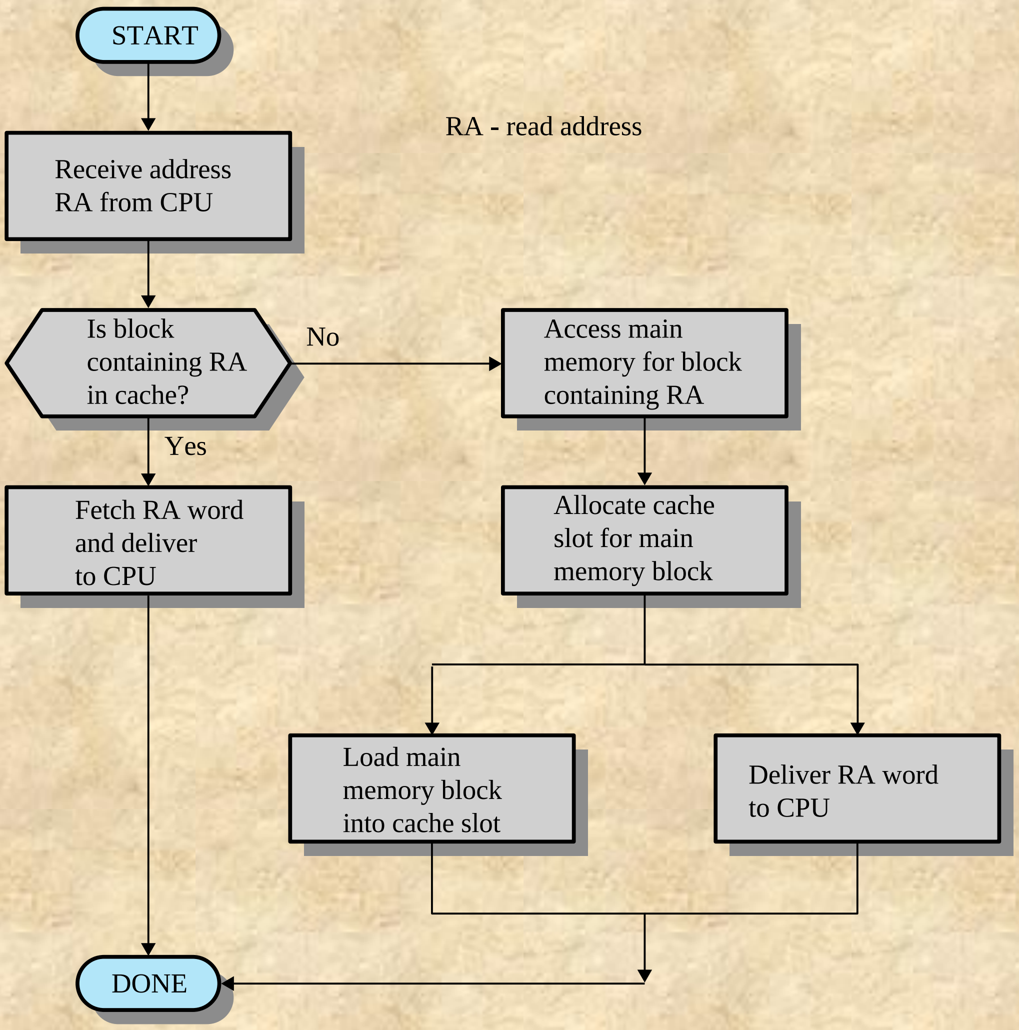
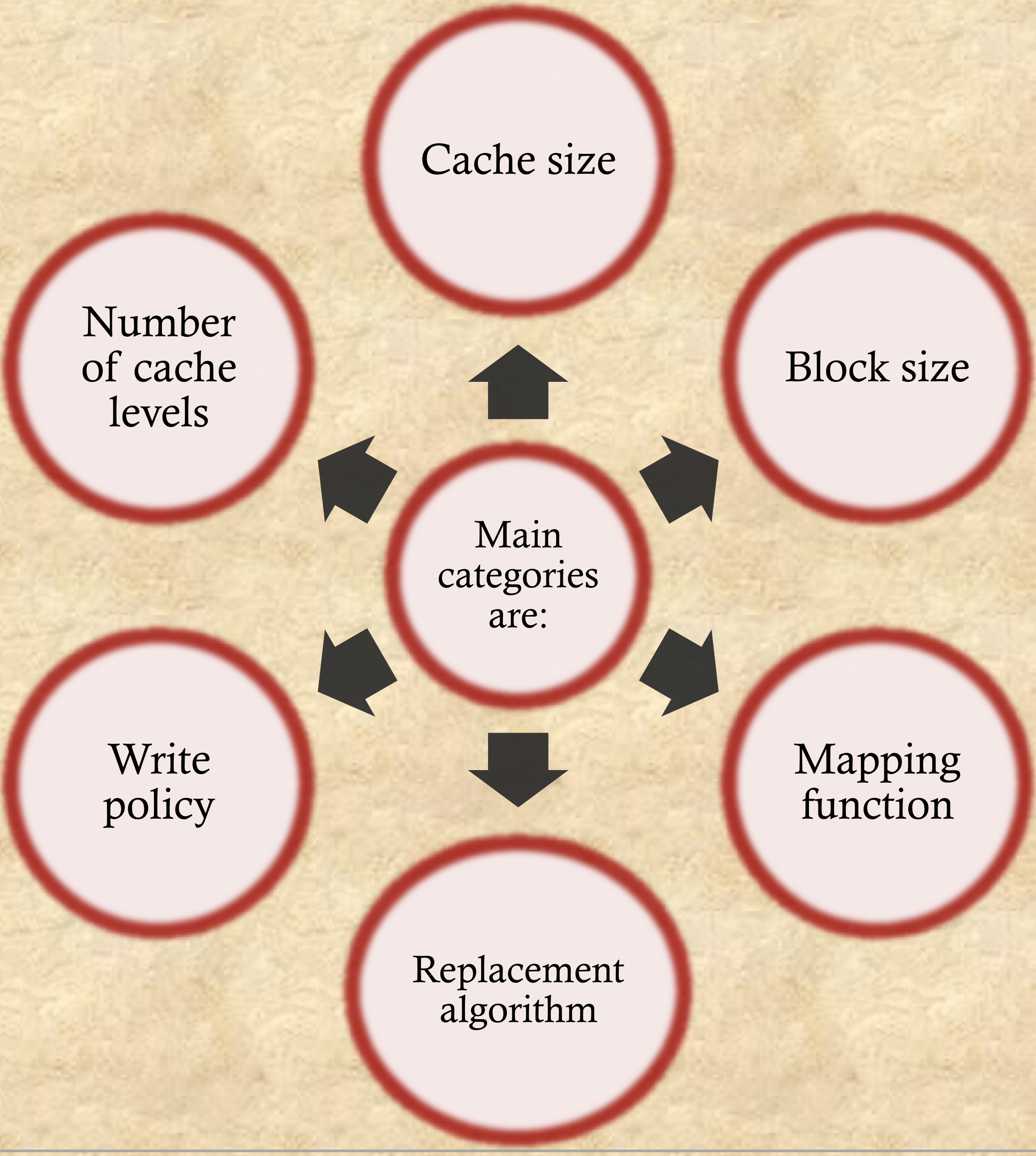

START
RA - read address
Receive address RA from CPU
Is block containing RA in cache?
No
Access main memory for block containing RA
Yes
Fetch RA word and deliver to CPU
Allocate cache slot for main memory block
Load main memory block into cache slot
Deliver RA word to CPU
DONE

Figure 1.18 Cache Read Operation

Cache Design



Cache and Block Size

Cache Size

- Small caches have significant impact on performance

Block Size

- The unit of data exchanged between cache and main memory

Mapping Function

- Determines which cache location the block will occupy

Two constraints affect design:

- When one block is read in, another may have to be replaced
- The more flexible the mapping function, the more complex is the circuitry required to search the cache

Replacement Algorithm

- Least Recently Used (LRU) Algorithm
  - Effective strategy is to replace a block that has been in the cache the longest with no references to it
  - Hardware mechanisms are needed to identify the least recently used block
    - Chooses which block to replace when a new block is to be loaded into the cache

Write Policy

Dictates when the memory write operation takes place

- Can occur every time the block is updated
- Can occur when the block is replaced
  - Minimizes write operations
  - Leaves main memory in an obsolete state

Analysis of Two-level memory

- Level 1 (M1): faster, expensive, smaller
  - T_1 : access time for M1
  - C_1 : average cost per bit for M1
  - S_1 : size of M1
- Level 2 (M2): slower, cheaper, larger
  - T_2 : access time for M1
  - C_2 : average cost per bit for M1
  - S_2 : size of M1

Analysis of Two-level memory

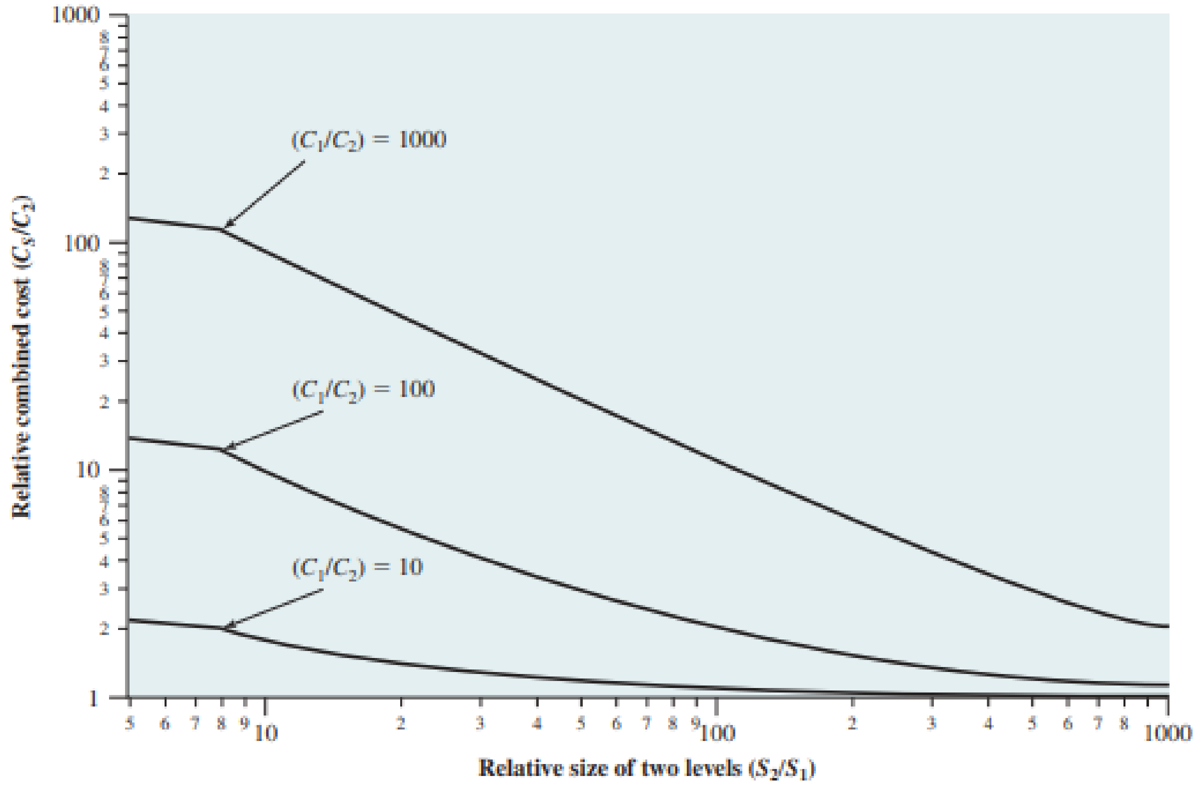


Figure 1.22 **Relationship of Average Memory Cost to Relative Memory Size for a Two-Level Memory**

Analysis of Two-level memory

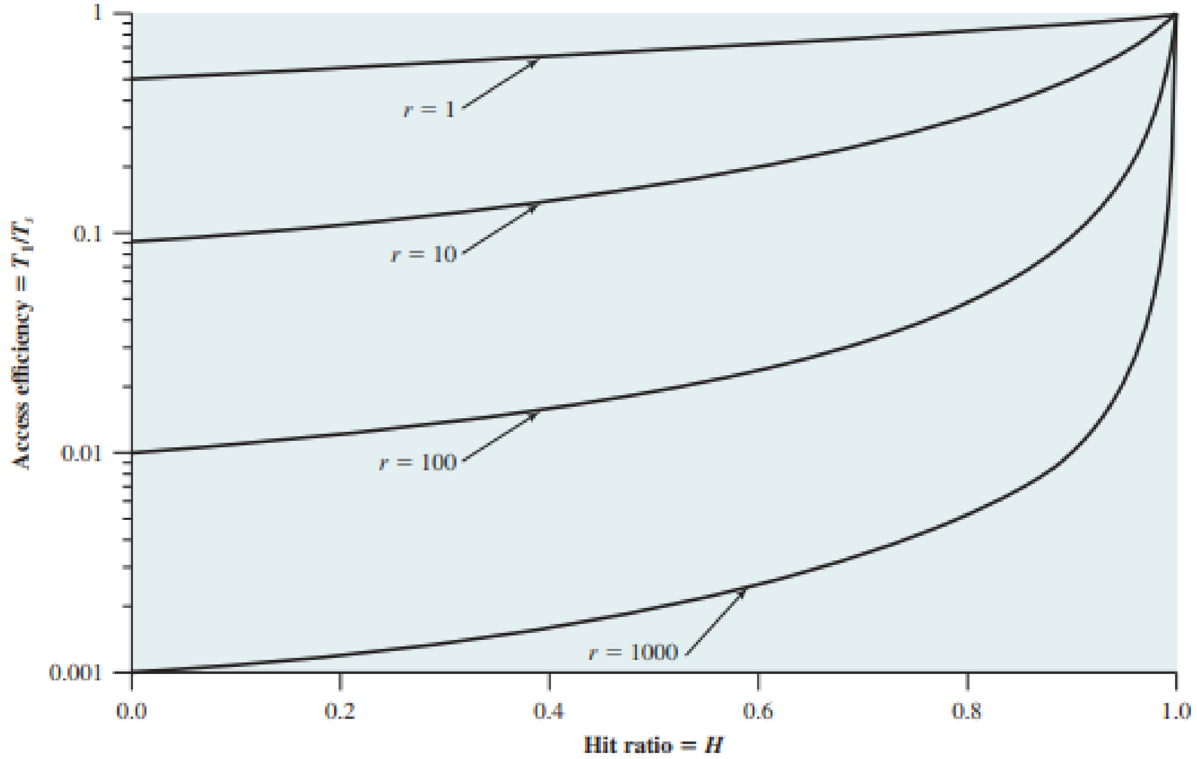


Figure 1.23 Access Efficiency as a Function of Hit Ratio (r = T2/T1)