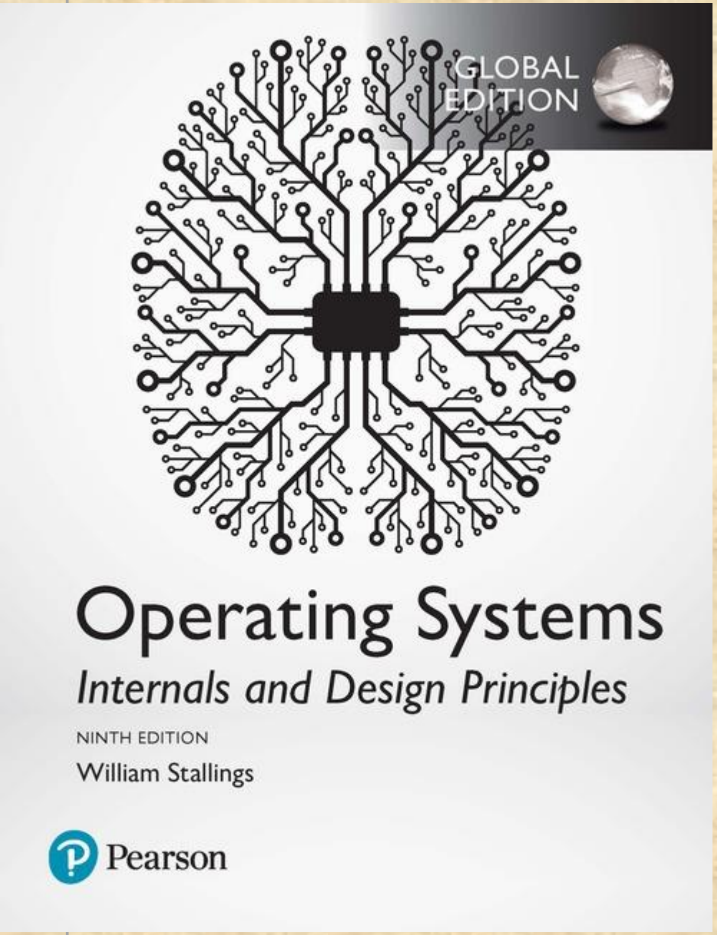


Chapter 1 - Part 2 Computer System Hardware

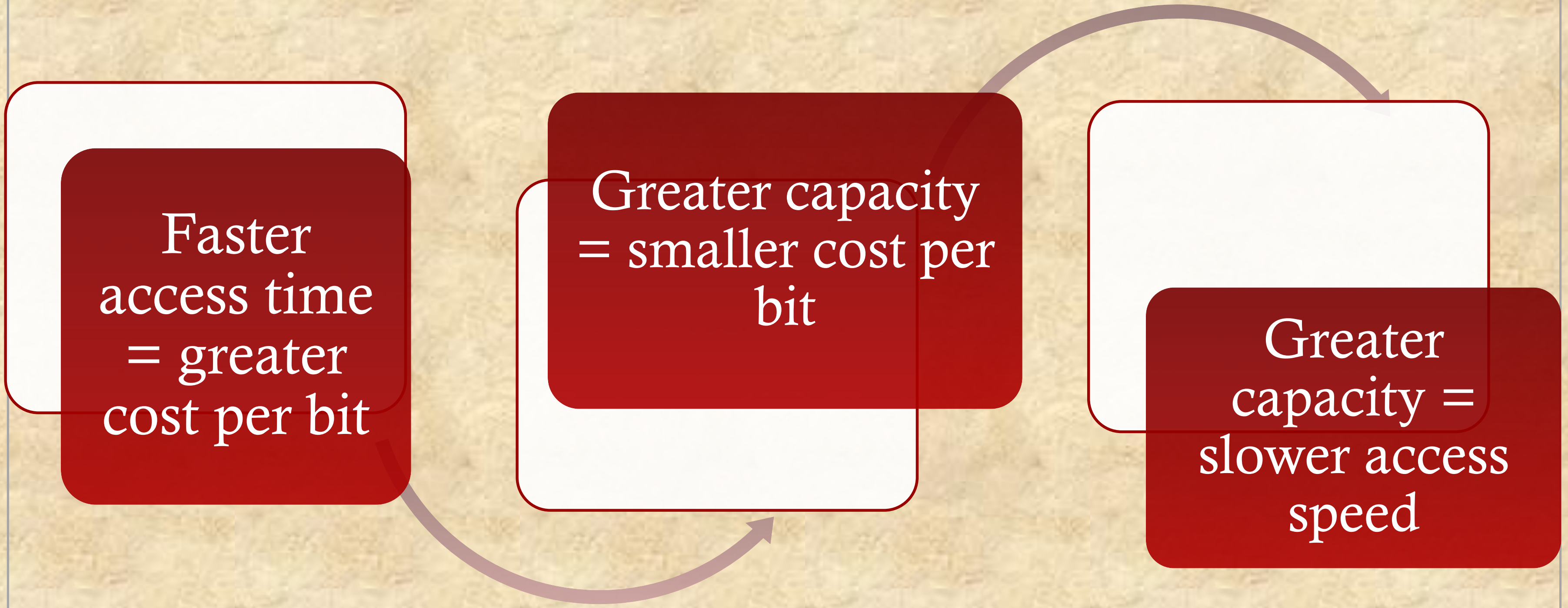
Ninth Edition, Global Edition

By William Stallings

Computer Memory

- Design constraints on memory are:
  - Capacity
  - Speed
  - Price
- Memory access time must keep up with the processor speed
- Cost of memory must be reasonable in relationship to other hardware components

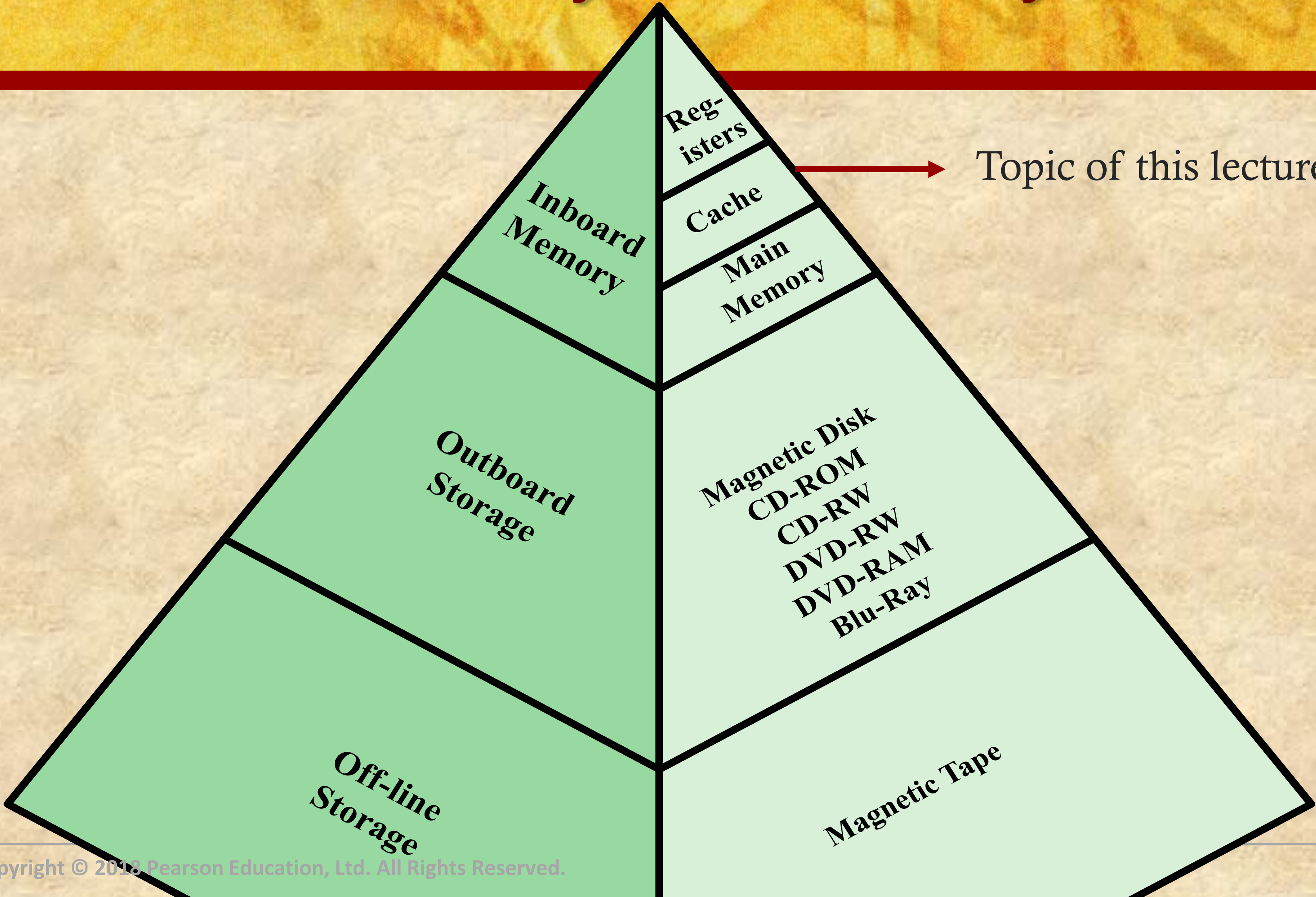
Memory Trade-Offs


Faster access time = greater cost per bit
Greater capacity = smaller cost per bit
Greater capacity = slower access speed

No perfect memory solution exists

Approach for best performance: multiple levels of different memory types

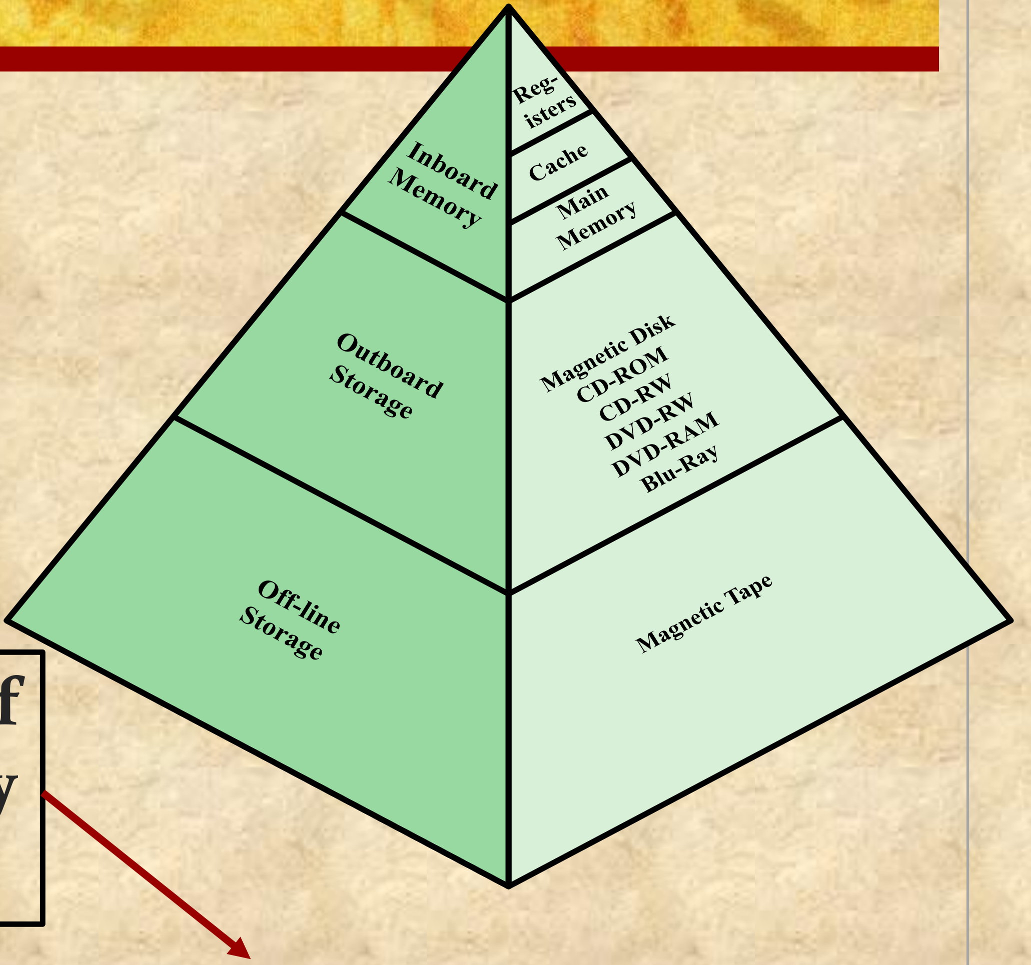
Memory Hierarchy



Memory Hierarchy

Going down the hierarchy:

- Decreasing cost per bit
- Increasing capacity
- Increasing access time
- **Decreasing frequency of access to the memory by the processor**

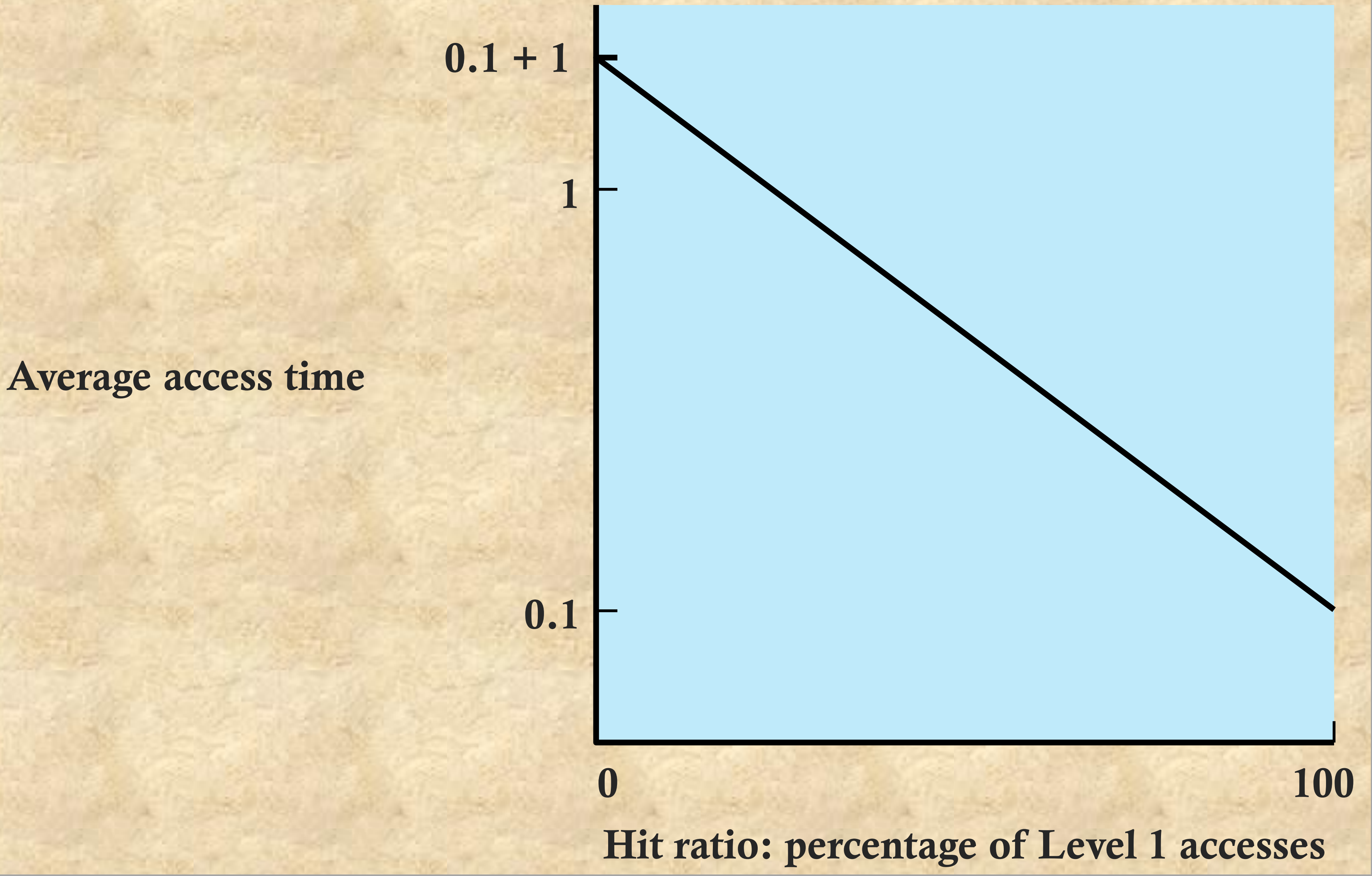


Design question that we address in this Chapter

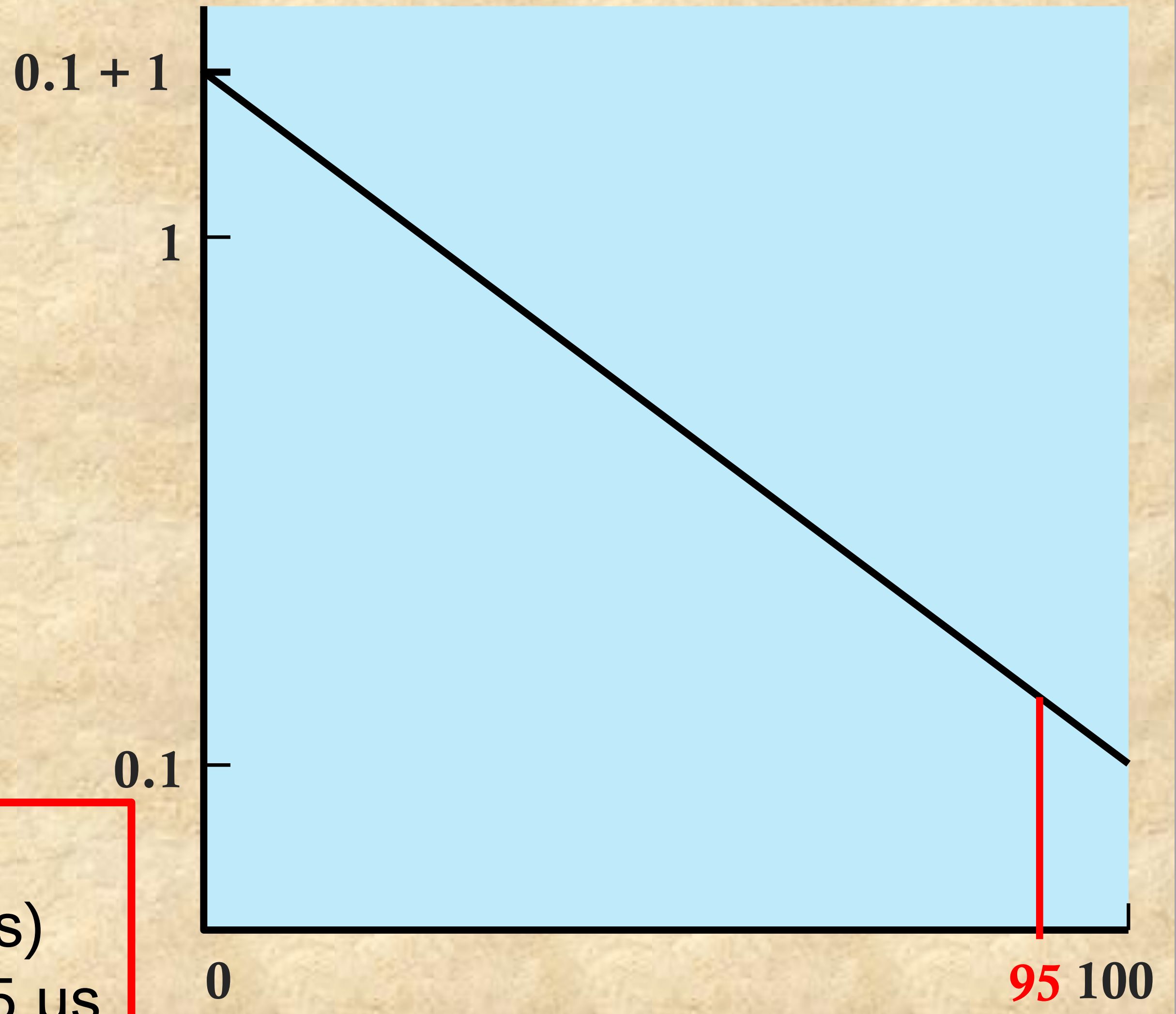
Two-Level Memory Example

- Processor has access to **two levels of memory**
- **Level 1** contains **1,000 bytes** and has an access time of **0.1 microseconds** **(small and fast memory: cache)**
- **Level 2** contains **100,000 bytes** and has an access time of **1 microsecond** **(large and slow memory: main memory)**
- If data is found in **Level 1** then processor accesses it **directly**
- If data is found in **Level 2** then it is first **transferred to Level 1** and then accessed by the processor

Performance of Two-Level Memory



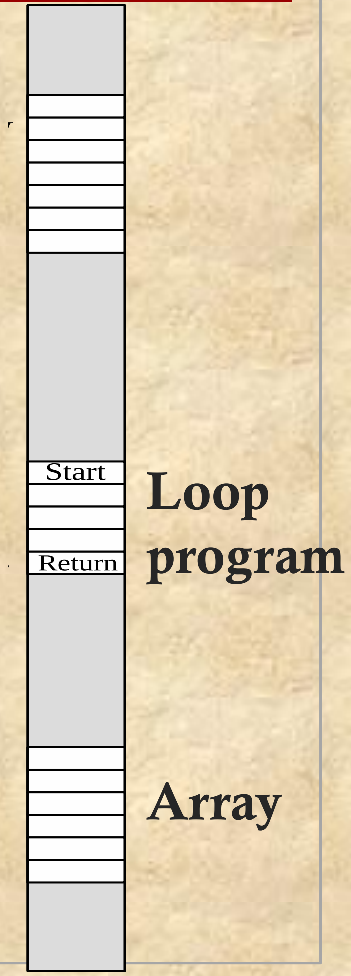
Performance of Two-Level Memory



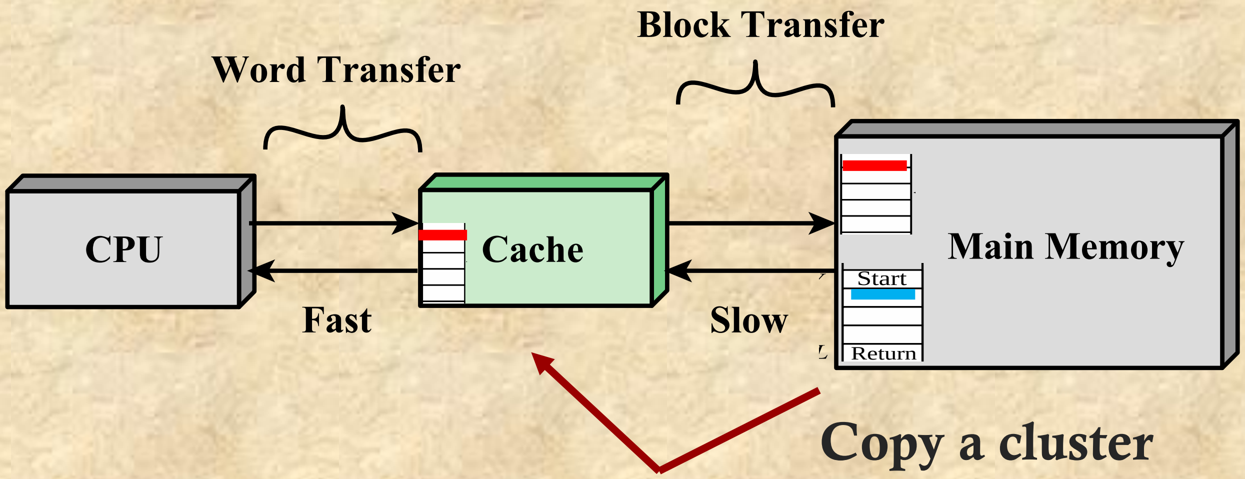
(0.95) (0.1 μs) +
(0.05) (0.1 μs + 1 μs)
= 0.095 μs + 0.055 μs
= 0.15 μs

Principle of Locality

- How to achieve a high hit ratio?
- Observation: memory accesses by the processor tend to **cluster** (loops, iterative array access, etc.)
- Organize data across memory levels such that current **cluster** in use is in the cache memory

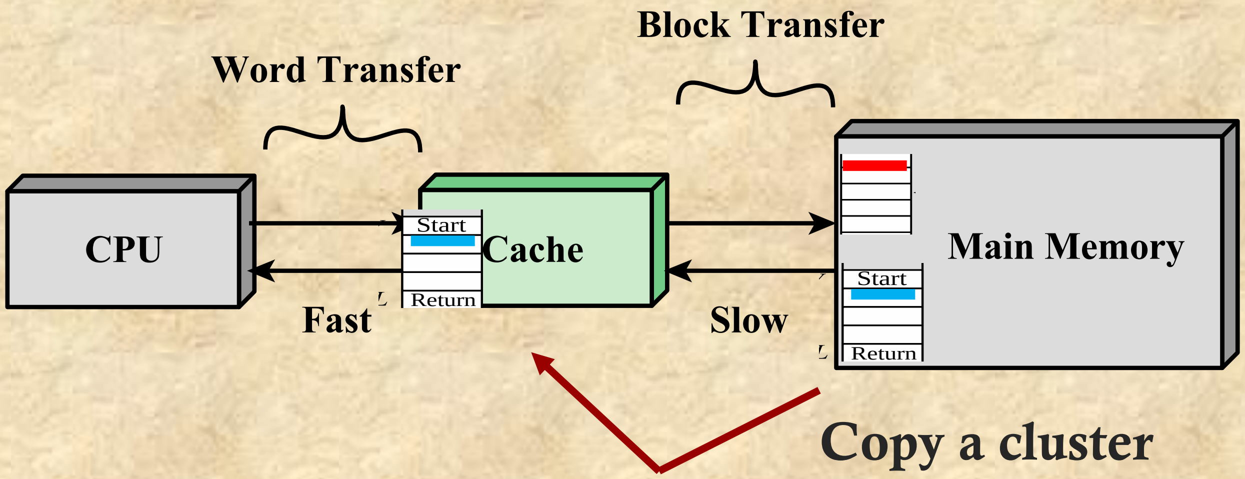


Cache-Main Memory

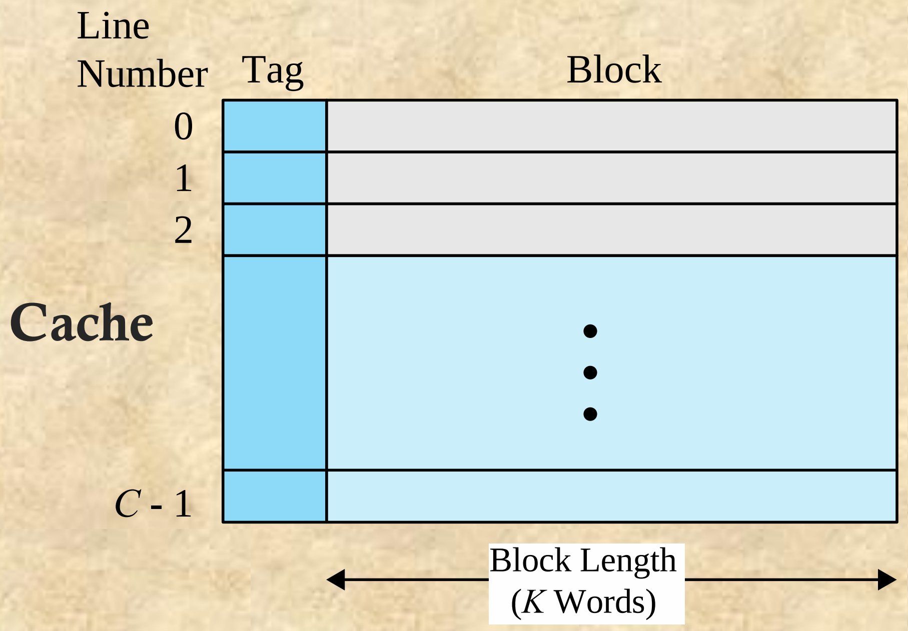


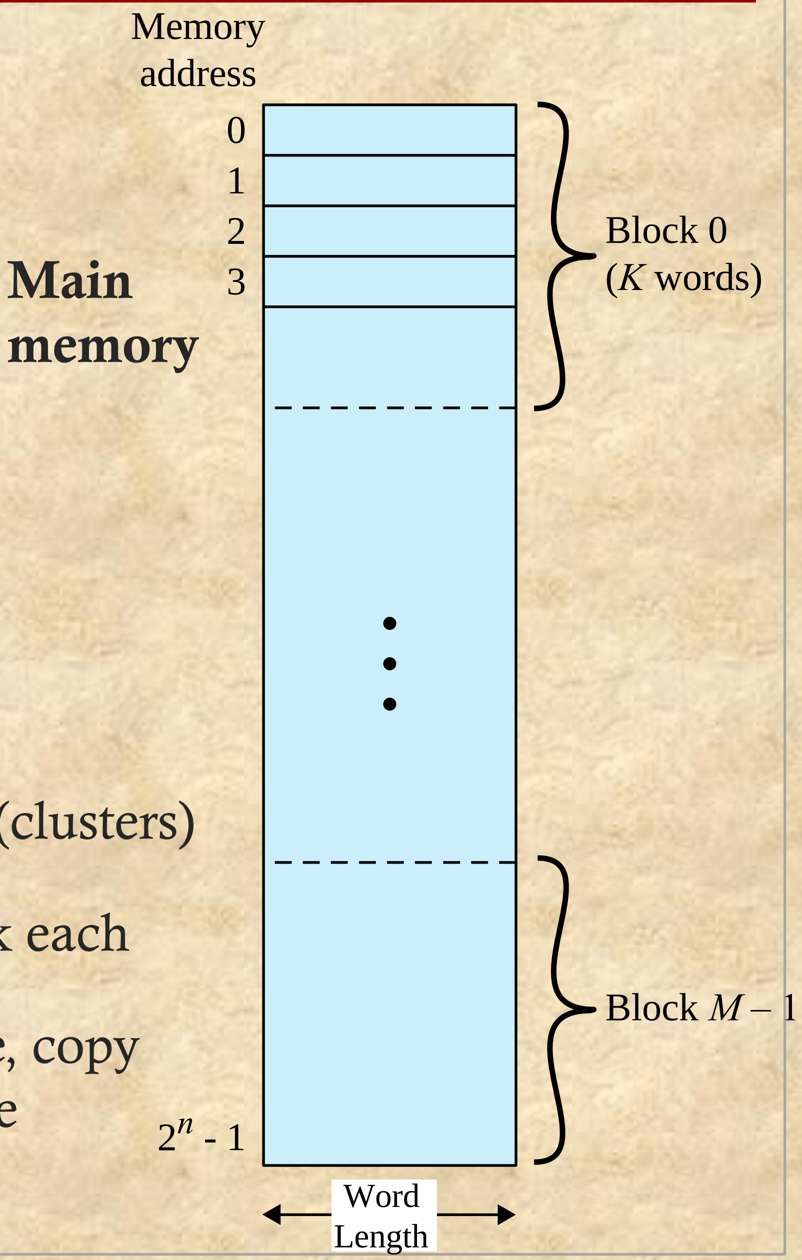
- Small cache memory contains copy of portion of larger main memory that is likely to be accessed by the processor in future
- Copy clusters into cache based on principle of locality

Cache-Main Memory



- Small cache memory contains copy of portion of larger main memory that is likely to be accessed by the processor in future
- Copy clusters into cache based on principle of locality





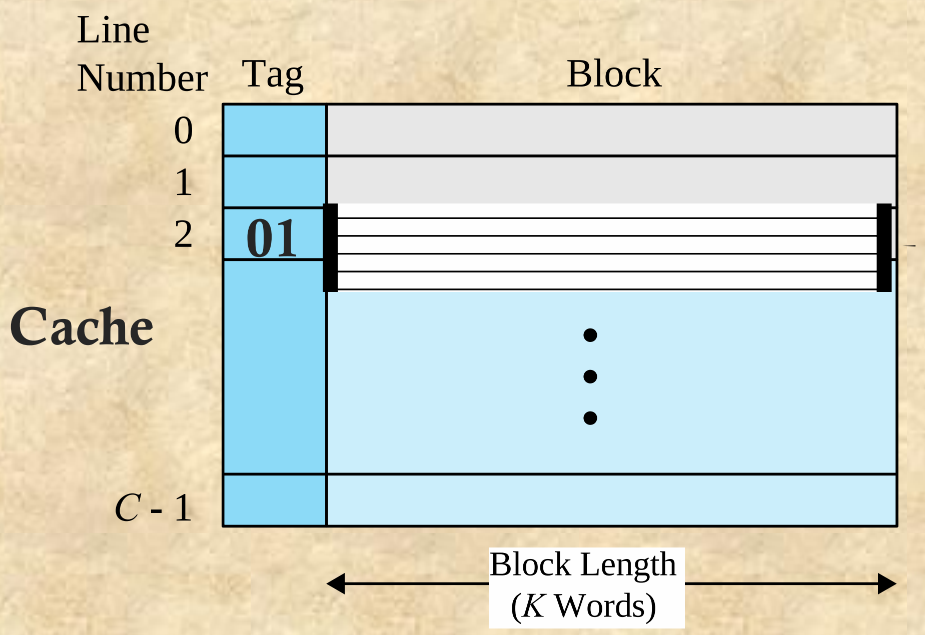
Cache-Main Memory Organization:

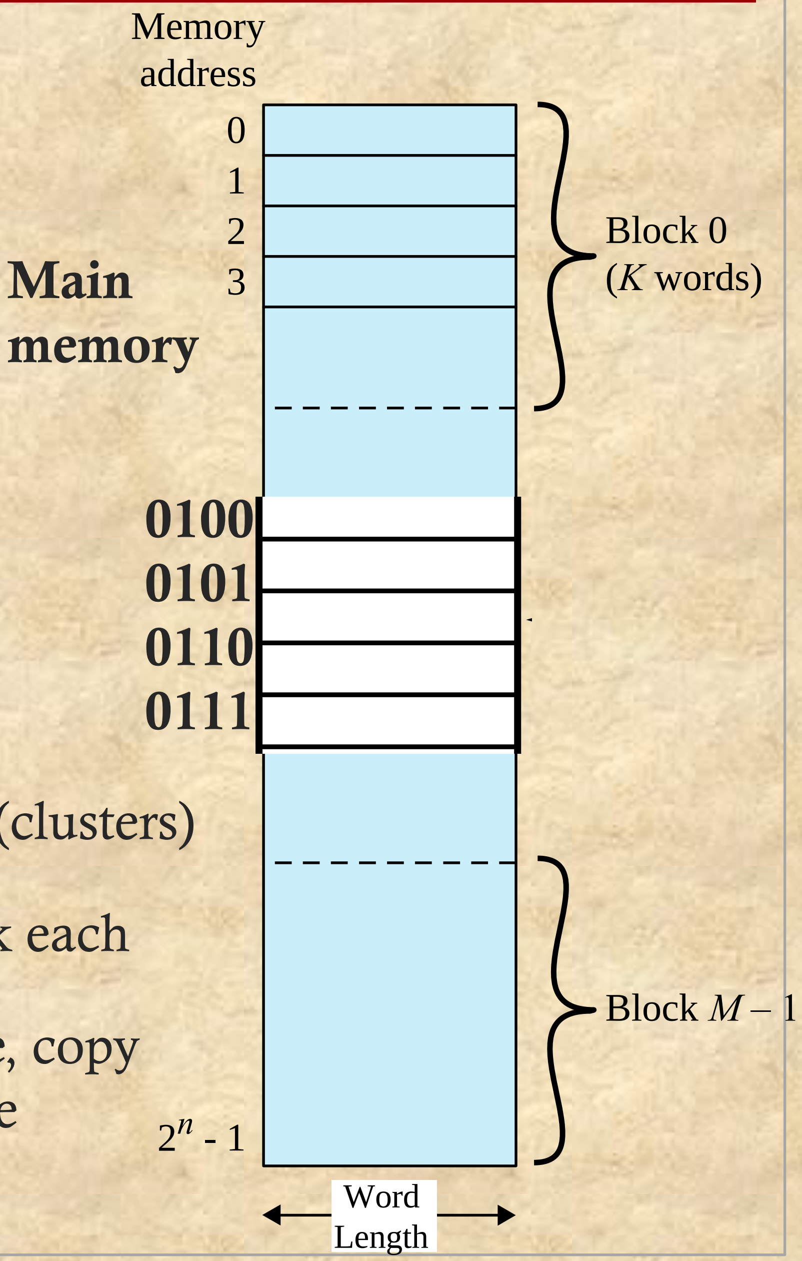
Divide main memory into blocks of words (clusters)

Cache consists of slots that can hold a block each

If a word is accessed that is not in the cache, copy the block containing the word into the cache

Add a binary tag that identifies the block





Cache-Main Memory Organization:

Divide main memory into blocks of words (clusters)

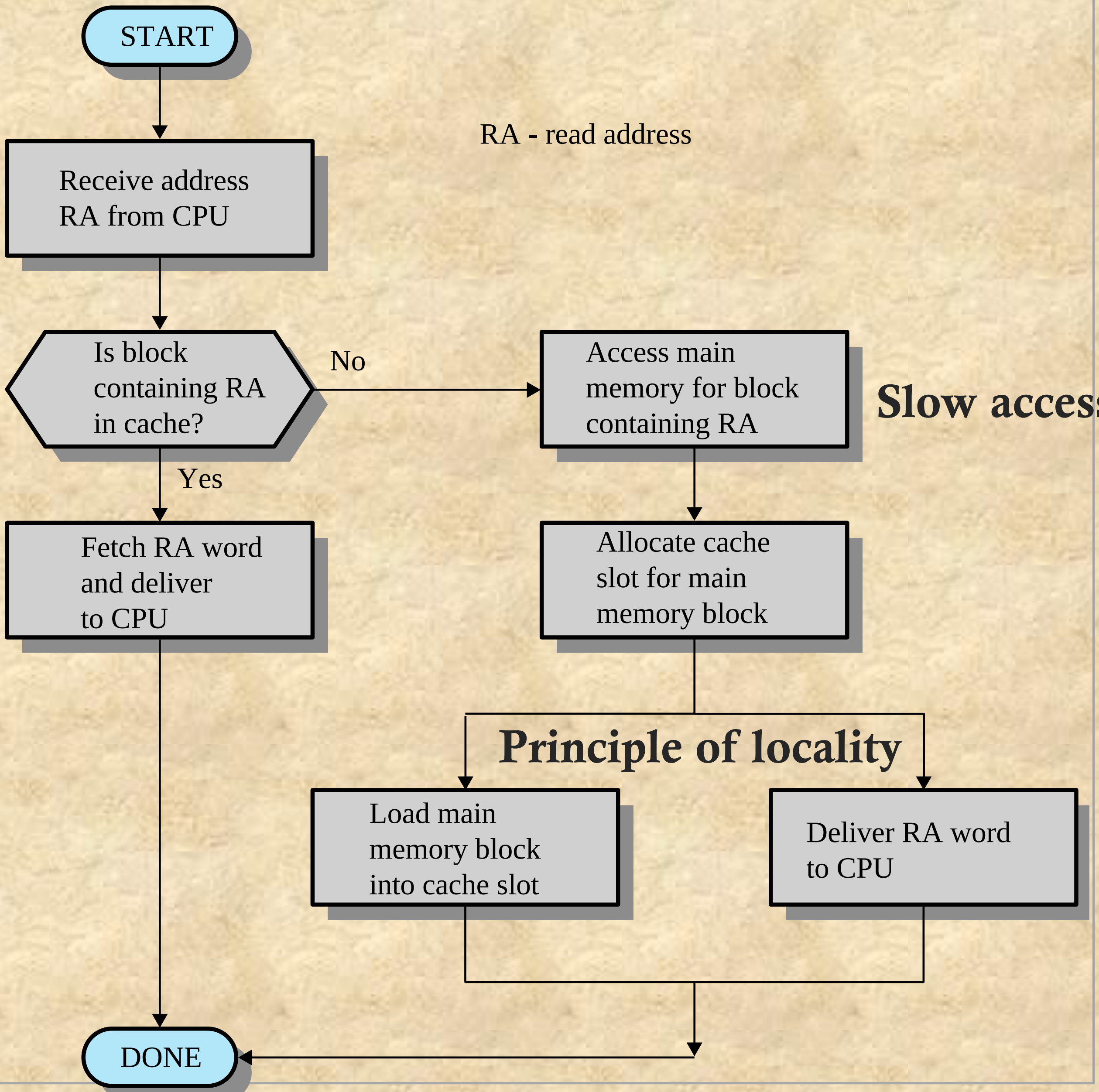
Cache consists of slots that can hold a block each

If a word is accessed that is not in the cache, copy the block containing the word into the cache

Add a binary tag that identifies the block

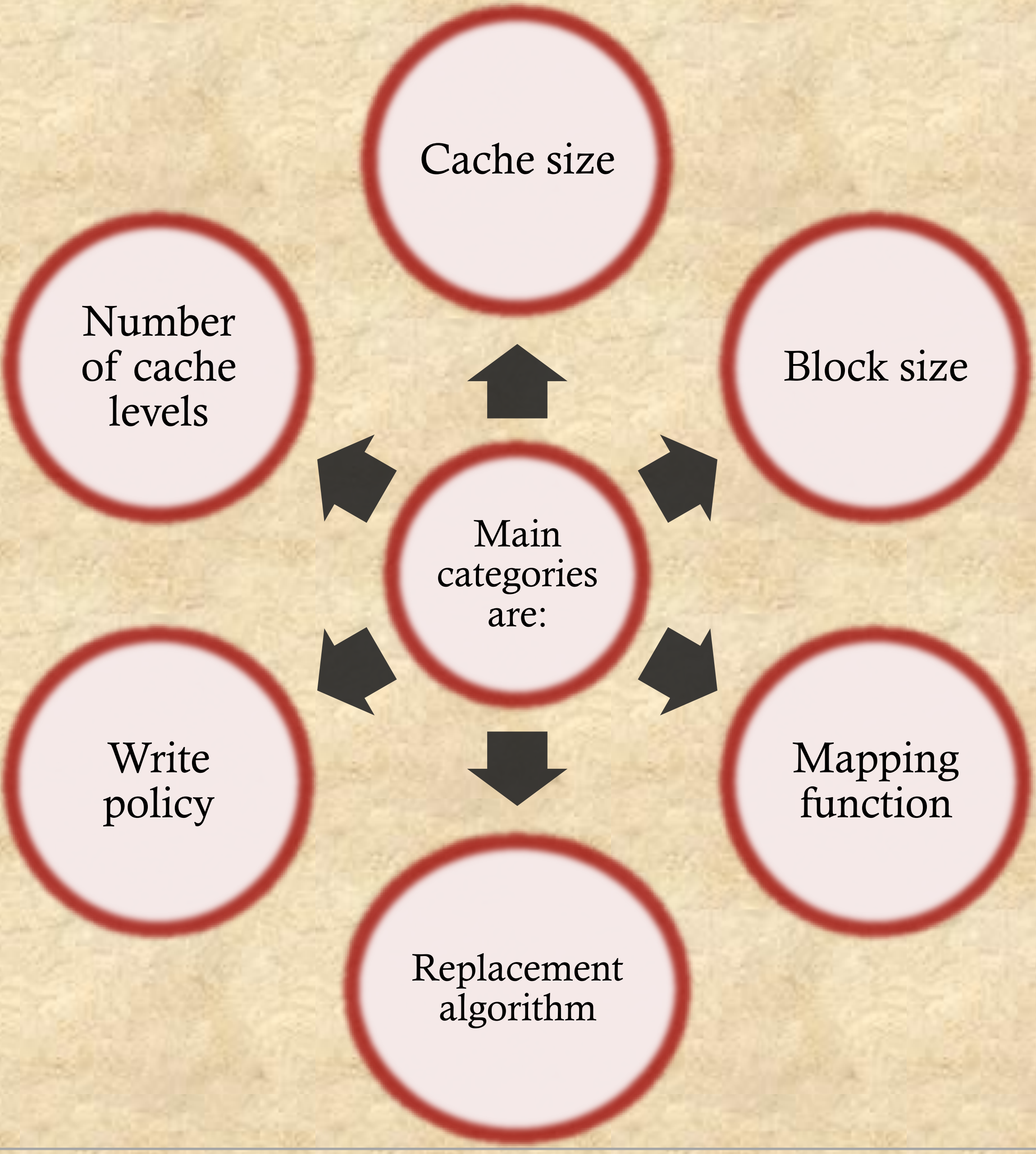
Example:

Read from memory address RA


START
Receive address RA from CPU
Is block containing RA in cache?
No
Yes
Access main memory for block containing RA
Slow access
Fast access
Fetch RA word and deliver to CPU
Allocate cache slot for main memory block
Principle of locality
Load main memory block into cache slot
Deliver RA word to CPU
DONE

RA - read address

Cache Design Questions



Cache and Block Size

Cache Size

- Small caches already have significant impact on performance

Block Size

- The size of data exchanged between cache and main memory

Mapping Function

- Determines which cache slot the block will occupy

Two constraints :

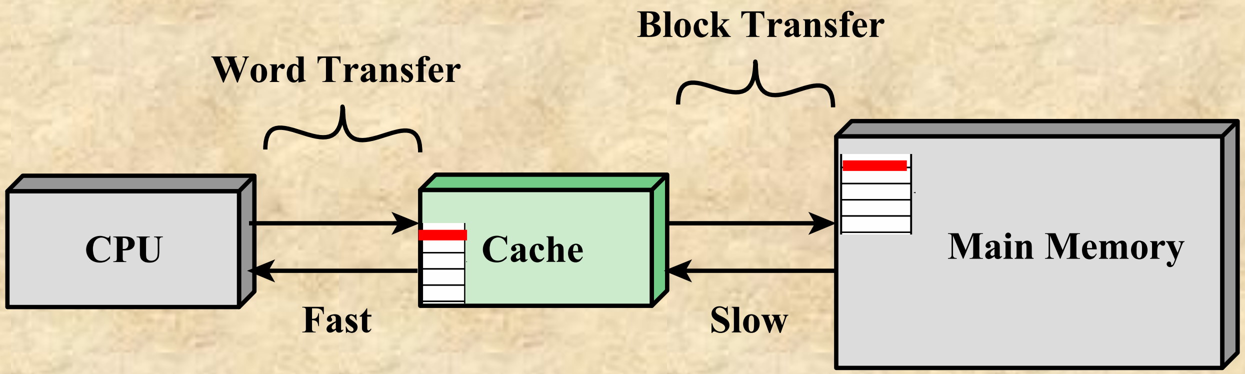
When one block is read in, another may have to be **replaced**

The more flexible the mapping function, the more complex is the search in the cache

Replacement Algorithm

- **Least Recently Used** (LRU) Algorithm
  - Strategy is to replace a block that has been in the cache the longest with no access to it

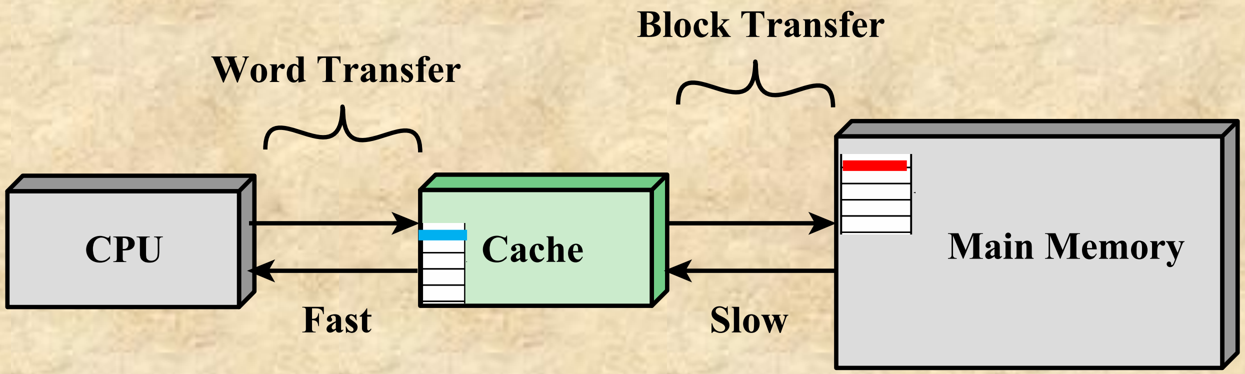
Write Policy



Dictates when an updated cache block is written back to main memory

- Every time the cache block is updated
- Only when the cache block is replaced: may cause problems in **multiprocessor systems**

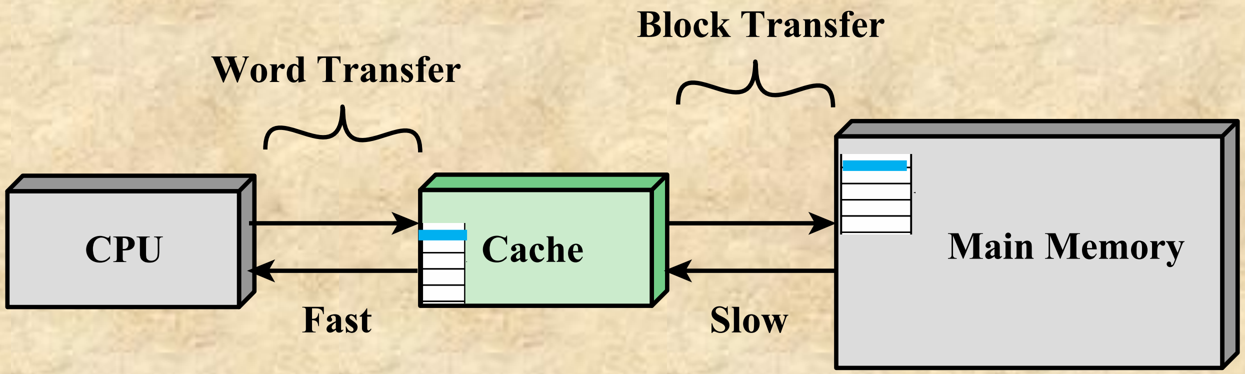
Write Policy



Dictates when an updated cache block is written back to main memory

- Every time the cache block is updated
- Only when the cache block is replaced: may cause problems in **multiprocessor systems**

Write Policy



Dictates when an updated cache block is written back to main memory

- Every time the cache block is updated
- Only when the cache block is replaced: may cause problems in **multiprocessor systems**

Multiprocessor Systems: Symmetric Multiprocessors (SMP)

- A stand-alone computer system with the following characteristics:
  - Two or more processors of similar capability
  - Processors share main memory and are connected by a system bus
  - Processors share access to I/O devices
  - All processors can perform the same functions
  - Communication between processors via shared memory

SMP Advantages

Performance

- Work can be done in parallel

Scaling

- Vendors can offer a range of products with different price and performance characteristics

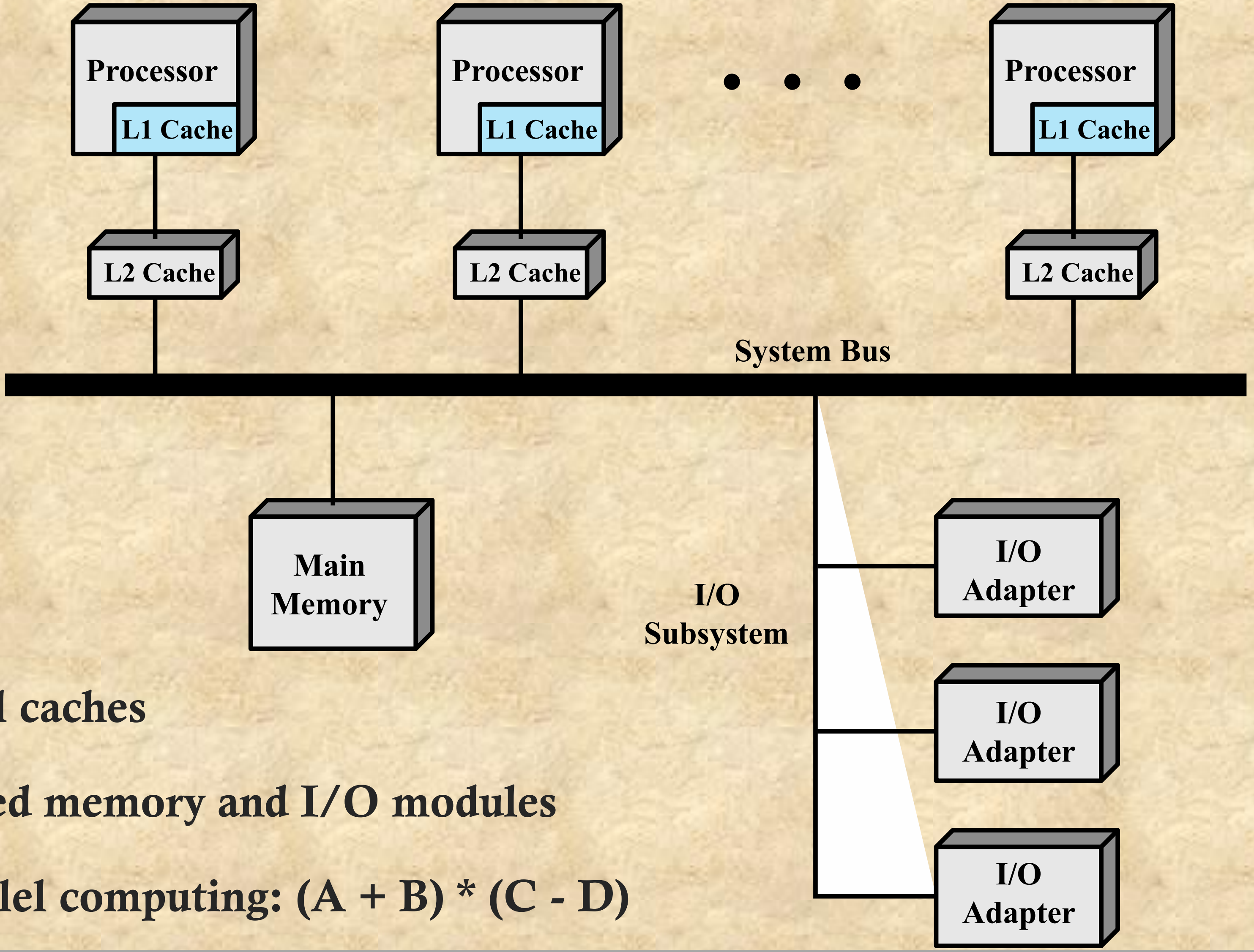
Availability

- Failure of a single processor does not halt the machine

Incremental Growth

- Each additional processor enhances the performance

SMP Organization



Local caches

Shared memory and I/O modules

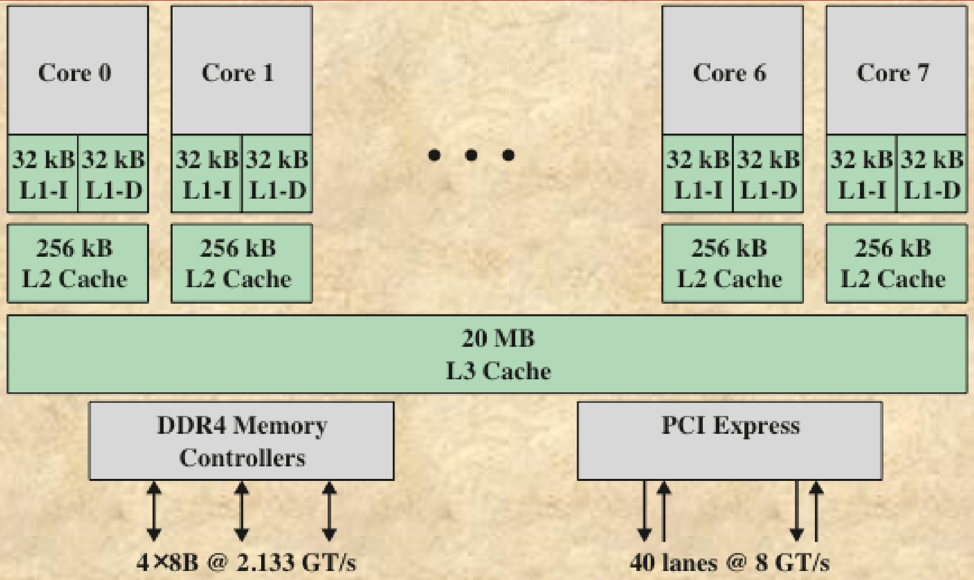
Parallel computing: (A + B) * (C - D)

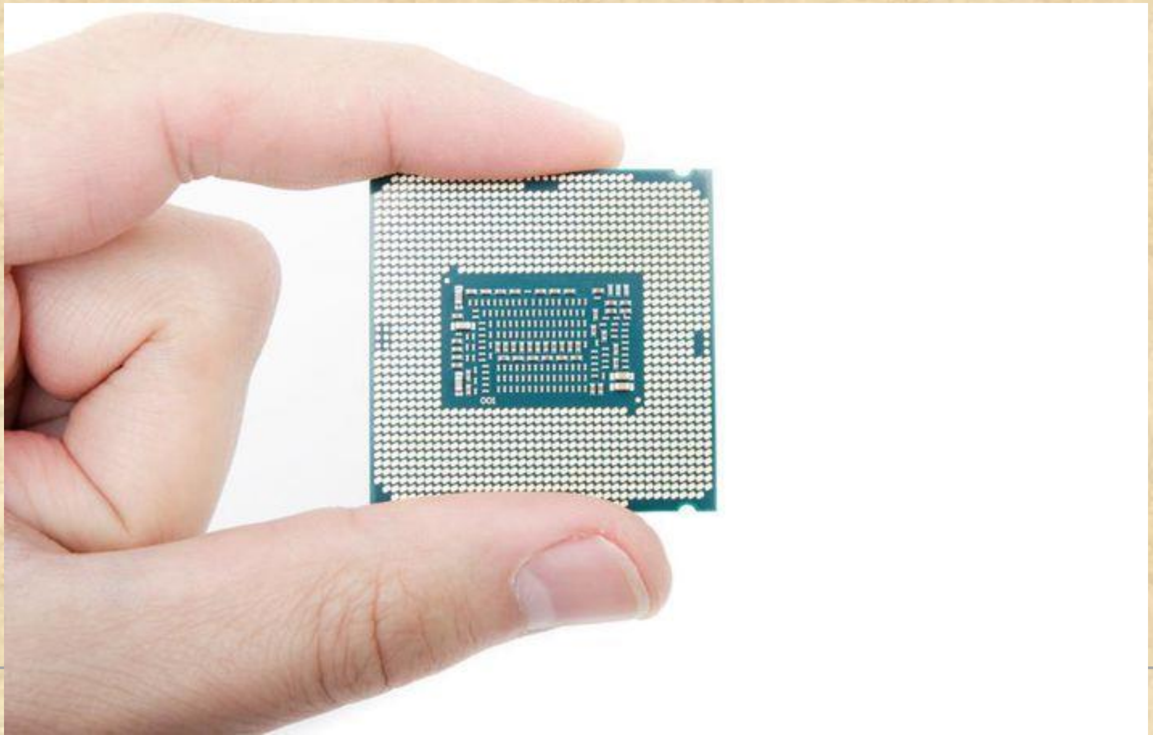
Multiprocessor Systems: Multicore Computer

- Combines two or more processors (cores) on a single chip
  - Each core consists of all of the components of an independent processor
- Exclusive and shared caches
- Fast communication between cores
- External communication to other chips via controllers

Example:

Intel Core i7





Chapter 1 Summary

- Basic hardware elements
- Evolution of the microprocessor
- Instruction execution
- Interrupts
  - Interrupts and the instruction cycle
  - Interrupt processing
  - Multiple interrupts
- Memory hierarchy
- Cache memory
  - Motivation
  - Cache principles
  - Cache design
- Multiprocessor and multicore organization
  - Symmetric multiprocessors
  - Multicore computers