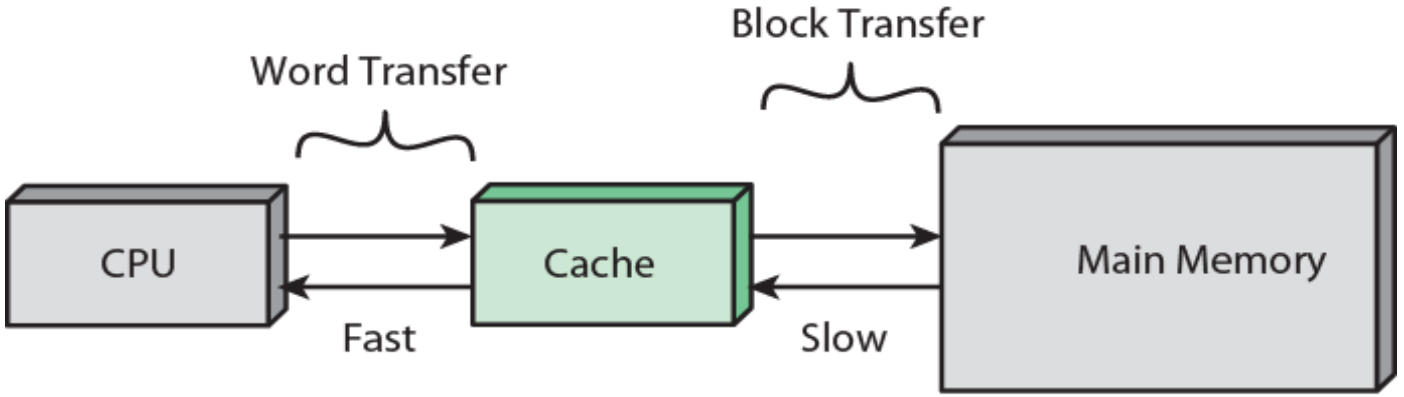
CPE 110408343
Fundamentals of Computer Architecture

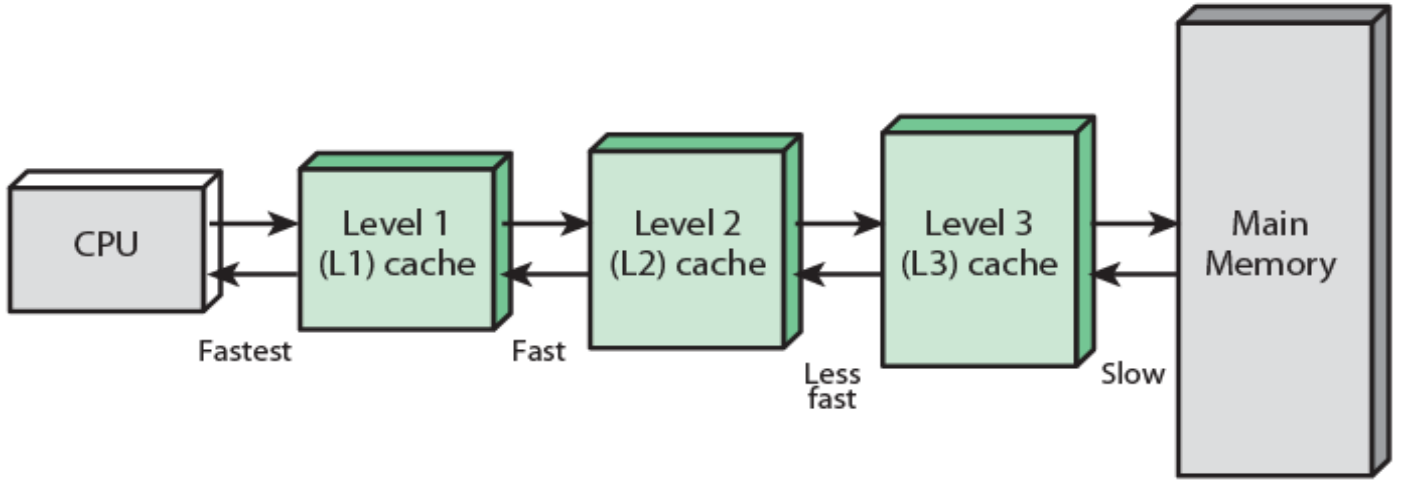
Dr. Khalil Ahmad Yousef

[Computer Engineering Department,
Hashemite University]

Big Picture: Cache and Main Memory



(a) Single cache

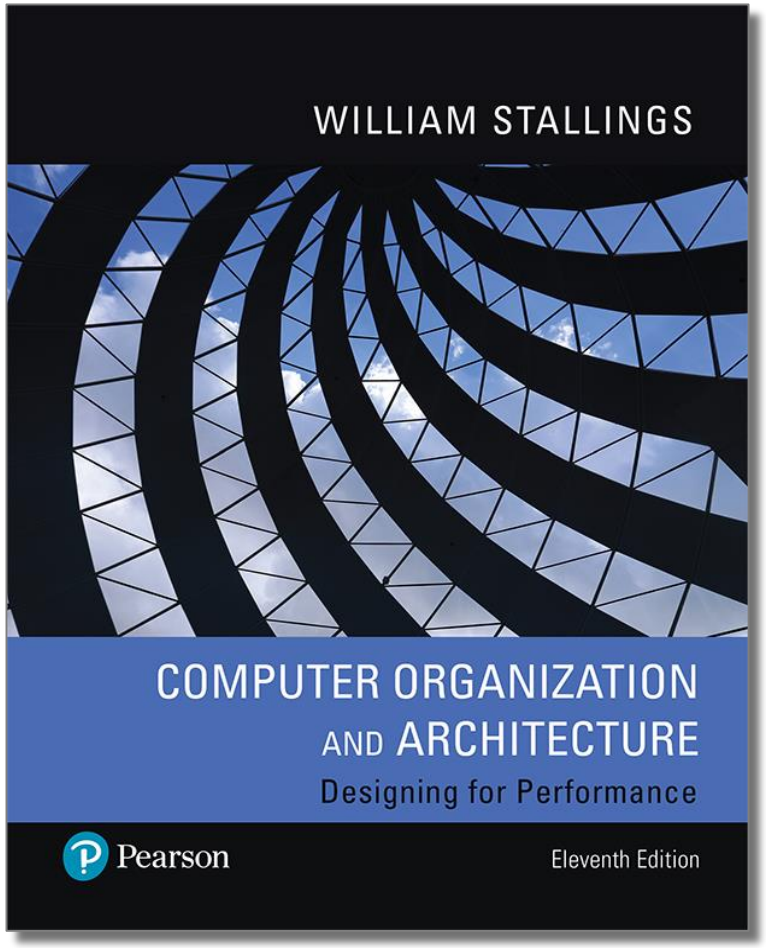


(b) Three-level cache organization

Computer Organization and Architecture

Designing for Performance

11th Edition



Chapter 4

The Memory Hierarchy: Locality and Performance

Principle of Locality (1 of 2)

- Also referred to as the *locality of reference*
- Reflects the observation that during the course of execution of a program, memory references by the processor tend to cluster (loops, subroutines, and data structures)
  - Programs typically contain a number of iterative loops and subroutines.
    - Once a loop or subroutine is entered, there are repeated references to a small set of instructions.
  - Similarly, operations on tables and arrays involve access to a clustered set of data word
- **Locality is based on three assertions:**
  - During any interval of time, a program references memory location non-uniformly
  - As a function of time, the probability that a given unit of memory is referenced tends to change slowly
  - The correlation between immediate past and immediate future memory reference patterns is high and tapers off as the time interval increases

Principle of Locality (2 of 2)

- Two forms of locality
  - Temporal locality
    - Refers to the tendency of a program to reference in the near future those units of memory referenced in the recent past
    - Constants, temporary variables, and working stacks are also constructs that lead to this principle
  - Spatial locality
    - Refers to the tendency of a program to reference units of memory whose addresses are near one another
    - Also reflects the tendency of a program to access data locations sequentially, such as when processing a table of data

Temporal and Spatial Localities

- Temporal Locality (Locality in Time):

  ⇒ Keep most recently accessed data items closer to the processor (if we use it now, we'll want to use it again soon)

- Spatial Locality (Locality in Space):

  ⇒ Move blocks consisting of contiguous words to the upper levels

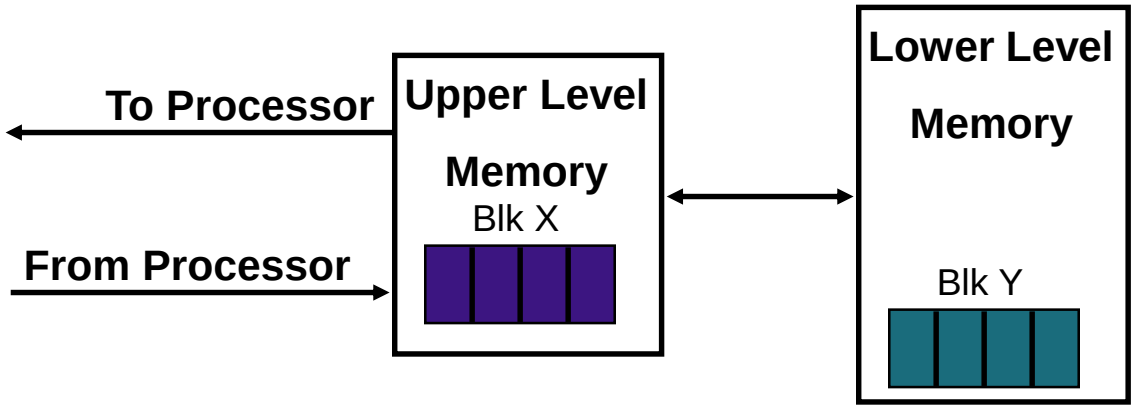


Figure 4.1

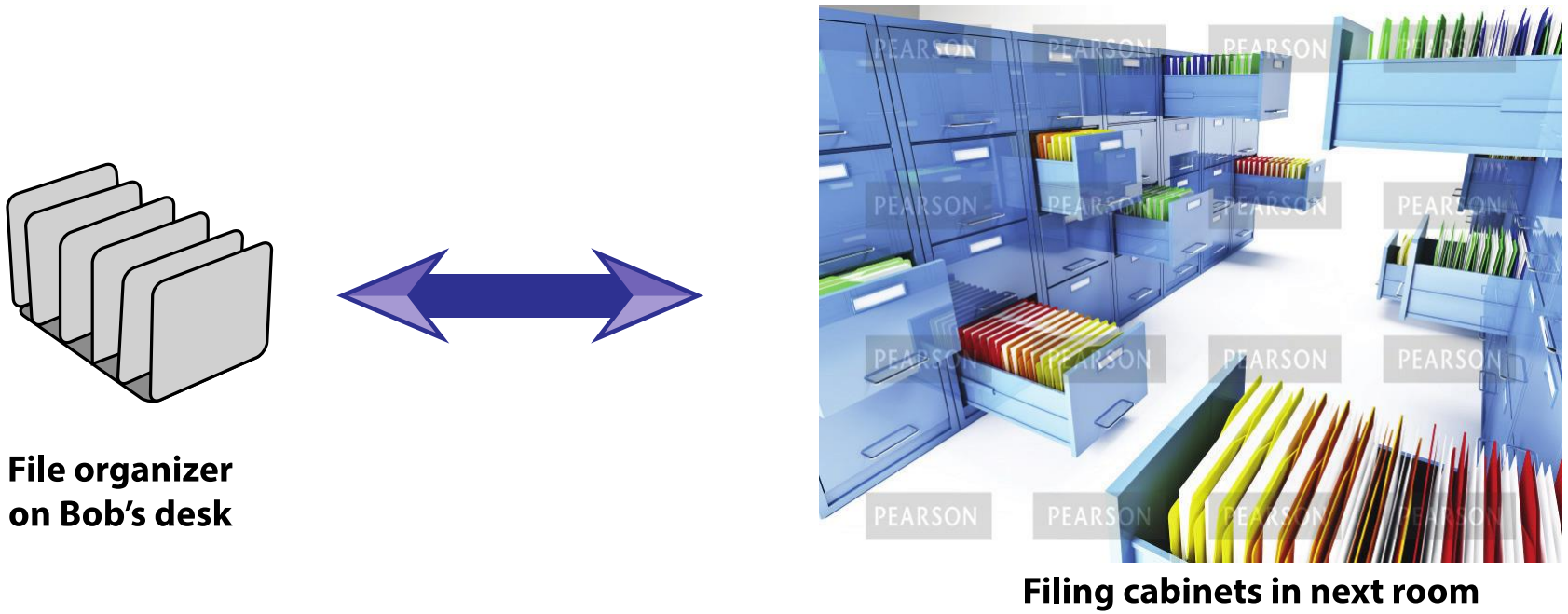


Figure 4.1 Moving File Folders Between Smaller, Faster-Access Storage and Larger, Slower-Access Storage

Table 4.1
Key Characteristics of Computer Memory Systems

Location Internal (e.g., processor registers, cache, main memory) External (e.g., optical disks, magnetic disks, tapes) **Capacity** Number of words Number of bytes **Unit of Transfer** Word Block **Access Method** Sequential Direct Random Associative	**Performance** Access time Cycle time Transfer rate **Physical Type** Semiconductor Magnetic Optical Magneto-optical **Physical Characteristics** Volatile/nonvolatile Erasable/nonerasable **Organization** Memory modules

Characteristics of Memory Systems

- Location
  - Refers to whether memory is internal and external to the computer
  - Internal memory is often equated with main memory
  - Processor requires its own local memory, in the form of registers
  - Cache is another form of internal memory
  - External memory consists of peripheral storage devices that are accessible to the processor via I/O controllers
- Capacity
  - Memory is typically expressed in terms of bytes
- Unit of transfer
  - For internal memory the unit of transfer is equal to the number of electrical lines into and out of the memory module

Method of Accessing Units of Data

Sequential access

- Memory is organized into units of data called records
- Access must be made in a specific linear sequence
- Access time is variable

Direct access

- Involves a shared read-write mechanism
- Individual blocks or records have a unique address based on physical location
- Access time is variable

Random access

- Each addressable location in memory has a unique, physically wired-in addressing mechanism
- The time to access a given location is independent of the sequence of prior accesses and is constant
- Any location can be selected at random and directly addressed and accessed
- Main memory and some cache systems are random access

Associative

- A word is retrieved based on a portion of its contents rather than its address
- Each location has its own addressing mechanism and retrieval time is constant independent of location or prior access patterns
- Cache memories may employ associative access

Capacity and Performance:

The two most important characteristics of memory

Three performance parameters are used:

Access time (latency)

- For random-access memory it is the time it takes to perform a read or write operation
- For non-random-access memory it is the time it takes to position the read-write mechanism at the desired location

Memory cycle time

- Access time plus any additional time required before second access can commence
- Additional time may be required for transients to die out on signal lines or to regenerate data if they are read destructively
- Concerned with the system bus, not the processor

Transfer rate

- The rate at which data can be transferred into or out of a memory unit
- For random-access memory it is equal to 1/(cycle time)

Memory

- The most common forms are:
  - Semiconductor memory
  - Magnetic surface memory
  - Optical
  - Magneto-optical
- Several physical characteristics of data storage are important:
  - Volatile memory
    - Information decays naturally or is lost when electrical power is switched off
  - Nonvolatile memory
    - Once recorded, information remains without deterioration until deliberately changed
    - No electrical power is needed to retain information
  - Magnetic-surface memories
    - Are nonvolatile
  - Semiconductor memory
    - May be either volatile or nonvolatile
  - Nonerasable memory
    - Cannot be altered, except by destroying the storage unit
    - Semiconductor memory of this type is known as read-only memory (ROM)
- For random-access memory the organization is a key design issue
  - Organization refers to the physical arrangement of bits to form words

Memory Hierarchy

- Design constraints on a computer's memory can be summed up by three questions:
  - How much, how fast, how expensive
- There is a trade-off among capacity, access time, and cost
  - Faster access time, greater cost per bit
  - Greater capacity, smaller cost per bit
  - Greater capacity, slower access time

Figure 4.6

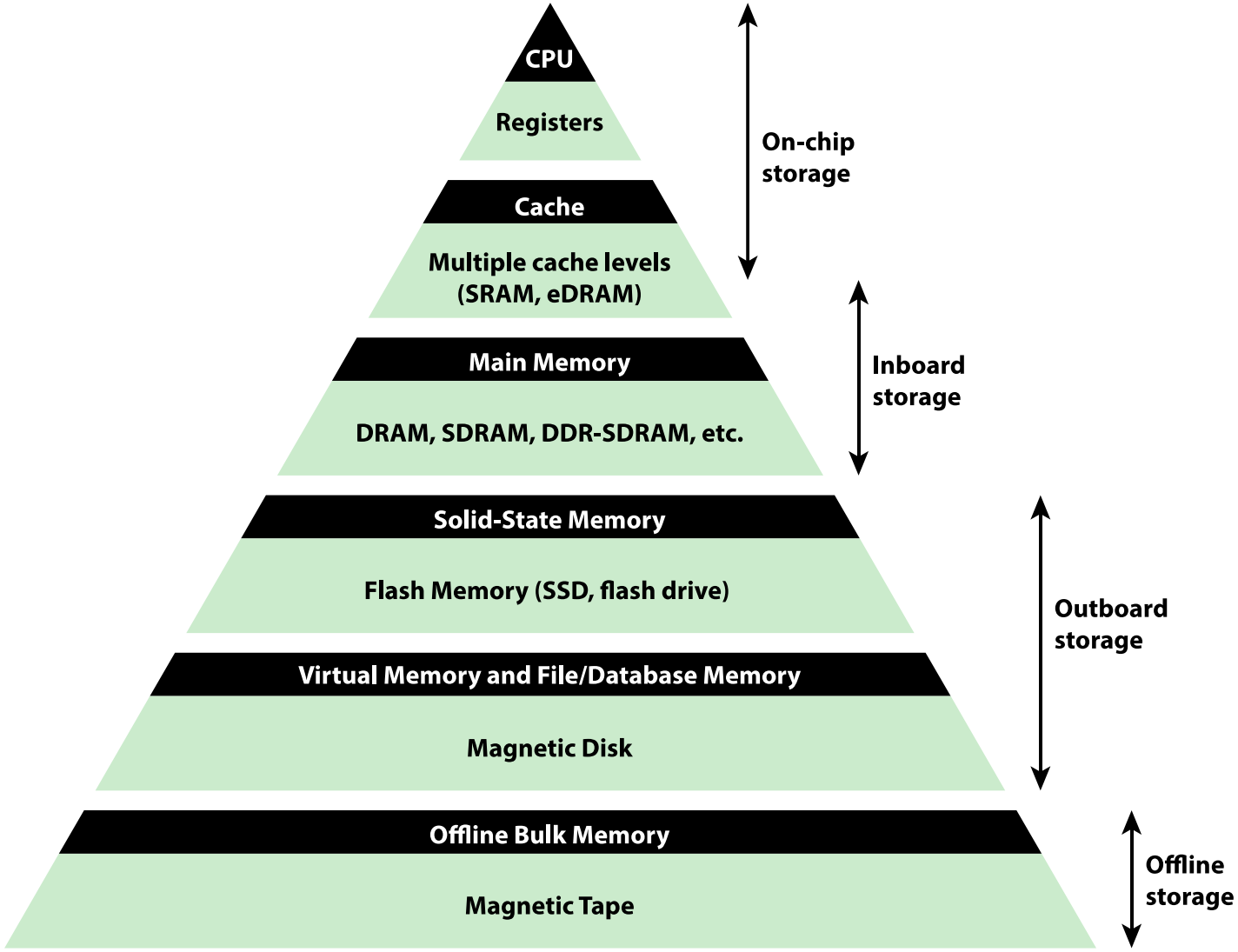


Figure 4.6 The Memory Hierarchy

Figure 4.7

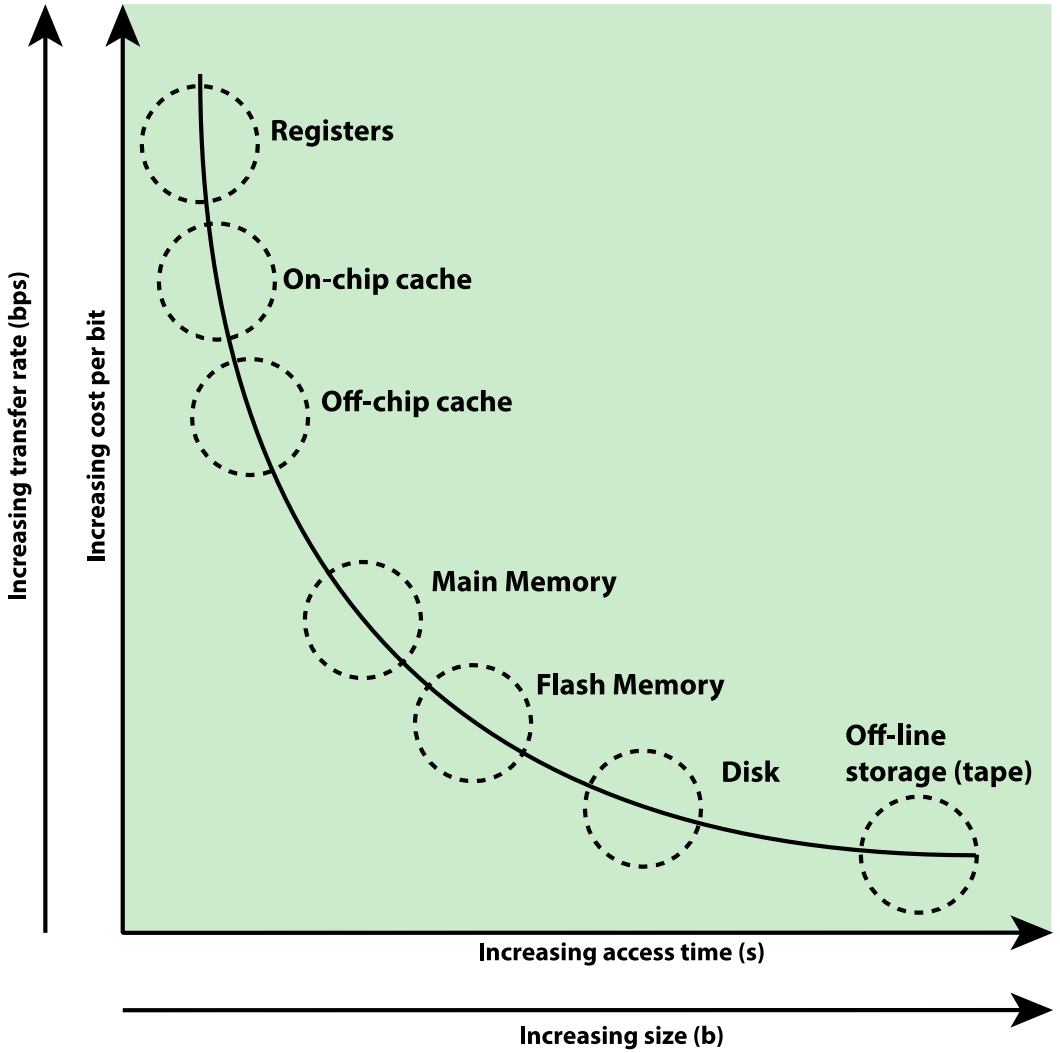


Figure 4.7 Relative Cost, Size, and Speed Characteristics Across the Memory Hierarchy

The Memory Hierarchy

- Assume a hypothetical system has two levels of memory
  - Level 2 should contain all instructions and data
  - Level 1 doesn't have room for everything, so when a new cluster (block of data) is required, the cluster it replaces must be sent back to the level 2
- These principles can be applied to much more than just two levels
- If performance is based on amount of memory rather than speed, lower levels can be used to simulate larger sizes for higher levels, e.g., virtual memory

Two-Level Memory Access

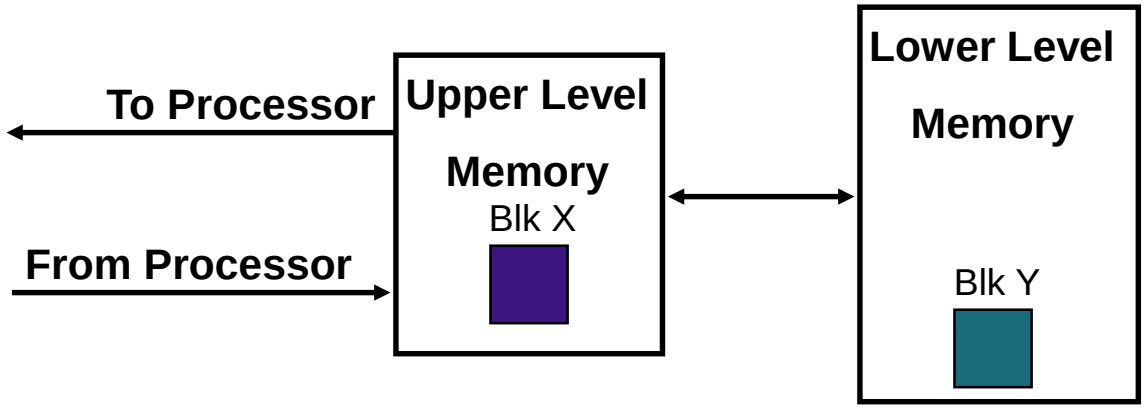
- A cache acts as a buffer between main memory and processor, creating a two-level internal memory
  - Exploits locality to provide improved performance over a comparable one-level memory
  - The main memory cache mechanism is part of the computer architecture, implemented in hardware and typically invisible to the operating system
- Two other instances of a two-level memory approach that also exploit locality and that are, at least partially, implemented in the operating system are virtual memory and the disk cache

Operation of Two-Level Memory

- The locality property can be exploited in the formation of a two-level memory
- The upper-level memory (M1) is smaller, faster, and more expensive (per bit) than the lower-level memory (M2)
- M1 is used as temporary store for part of the contents of the larger M2
- When a memory reference is made, an attempt is made to access the item in M1
  - If this succeeds, then a quick access is made
  - If not, then a block of memory locations is copied from M2 to M1 and the access then takes place via M1
- Because of locality, once a block is brought into M1, there should be a number of accesses to locations in that block, resulting in fast overall service

The Memory Hierarchy: Terminology

- Hit: data is in some block in the upper level (Blk X)
  - Hit Rate: fraction of memory accesses found in upper level
  - Hit Time: Time to access the upper level which consists of
    - RAM access time + Time to determine hit/miss



- Miss: data is not in the upper level so needs to be retrieve from a block in the lower level (Blk Y)
  - Miss Rate = 1 - (Hit Rate)
  - Miss Penalty: Time to bring in a block from the lower level and replace a block in the upper level with it + Time to deliver the block the processor
  - Hit Time << Miss Penalty

Memory: Important Performance Metric

- Average Access Time Across Multiple Levels

Figure 4.14

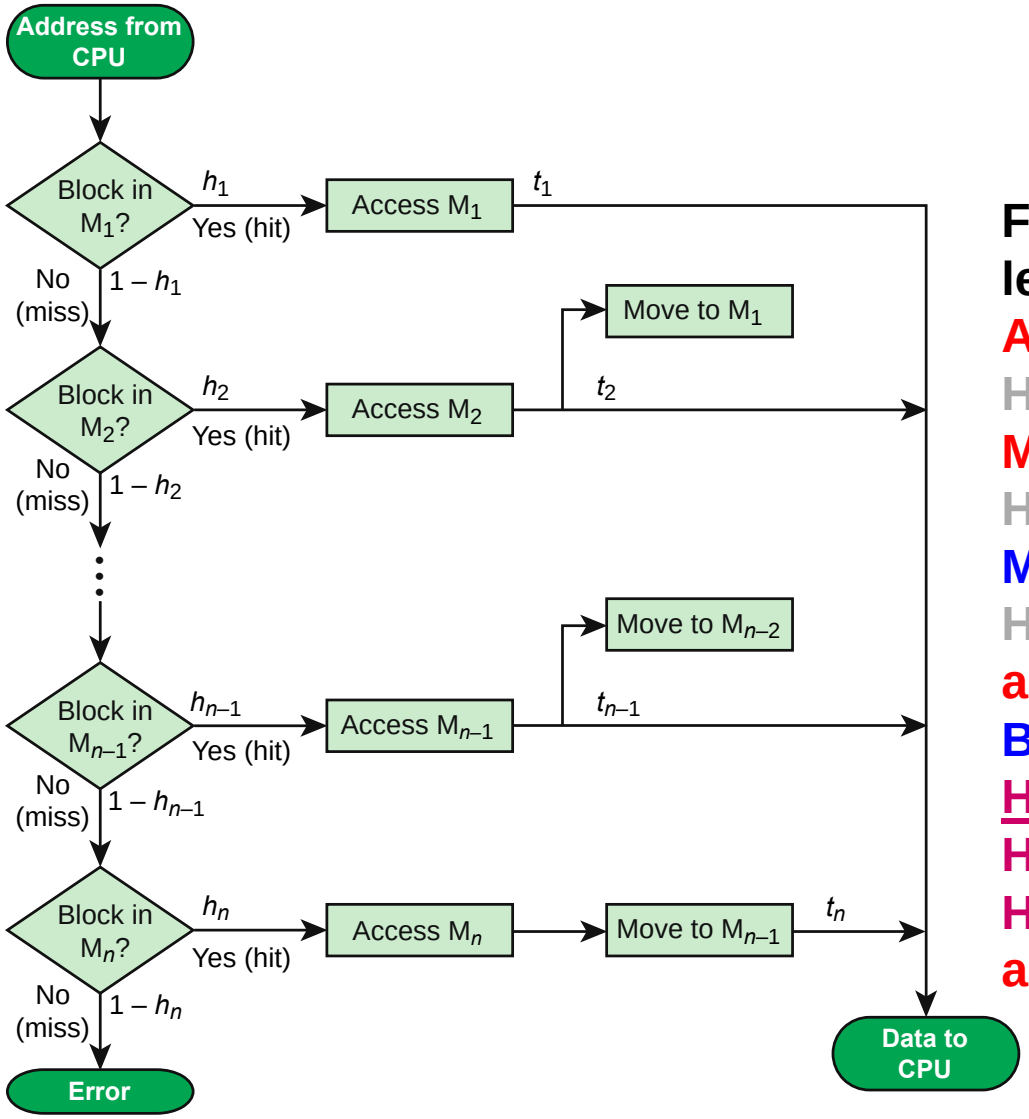


Figure 4.14 Multilevel Memory Access Performance Model

For the computation of the average access time across multiple levels of memories, please use the following set of equations:

Average Access Time across all levels =

Hit Rate L1* Hit Time L1 + Miss Rate L1 * Miss Penalty L1

Miss Penalty L1 =

Hit Rate L2* Hit Time L2 + Miss Rate L2 * Miss Penalty L2

Miss Penalty L2 =

Hit Rate L3* Hit Time L3 + Miss Rate L3 * Miss Penalty L3

and so on.

Be careful that

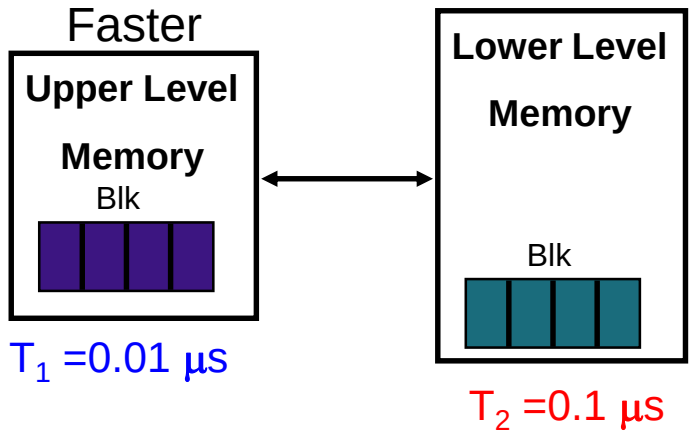
Hit Time L1 = t_1

Hit Time L2 = $t_1 + t_2$

Hit Time L3 = $t_1 + t_2 + t_3$

and so on

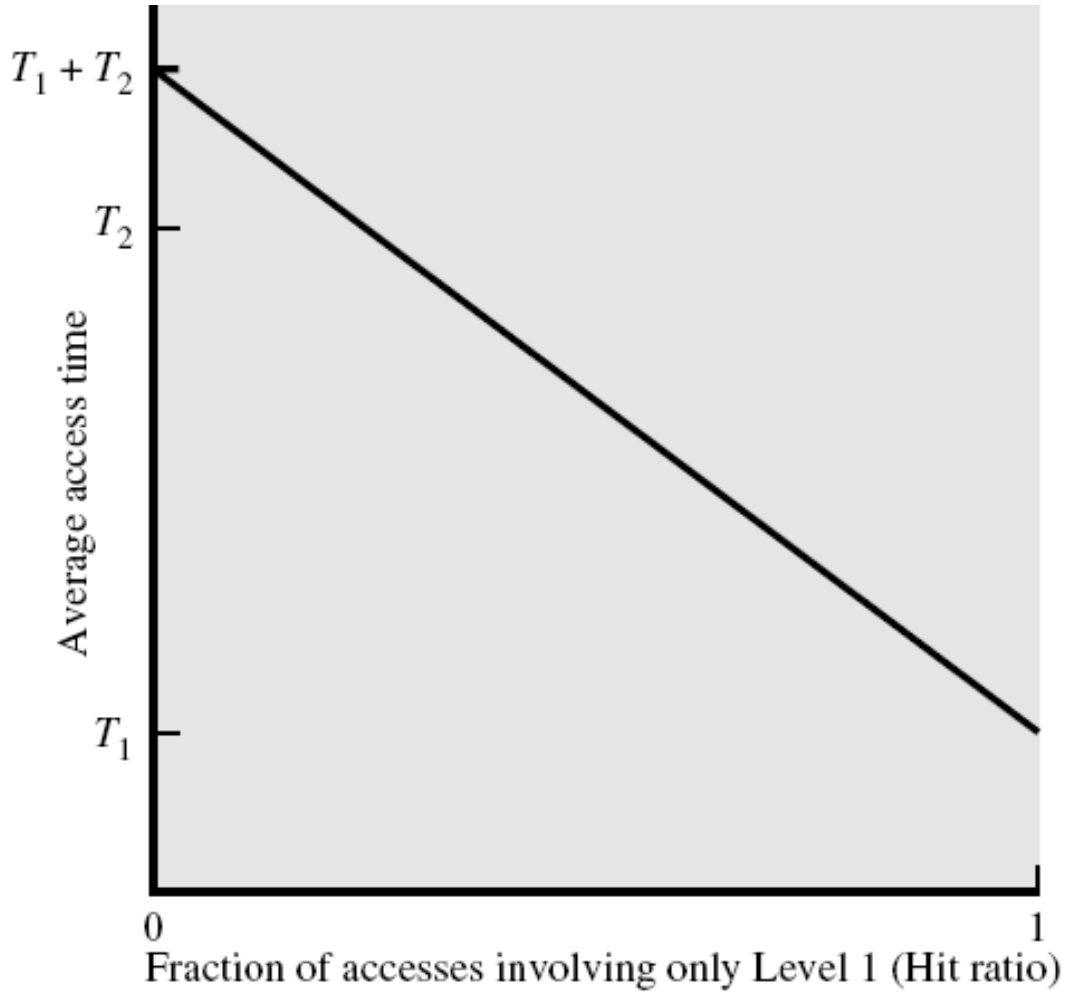
Example: Memory Hierarchy



- Suppose that the processor has access to two levels of memory.
  - Level 1 (M1) contains 1000 words and has an access time of 0.01 μs;
  - level 2 (M2) contains 100,000 words and has an access time of 0.1 μs.
  - Assume that if a word to be accessed is in level 1, then the processor accesses it directly. If it is in level 2, then the word is first transferred to level 1 and then accessed by the processor.
  - If 95% of the memory accesses are found in the faster level, then the average access time across the two levels might be:
- Solution:
  - (0.95)(0.01 μs) + (0.05)((0.01 + 0.1) μs) = 0.0095 + 0.0055 = 0.015 μs

Performance of a Simple Two-Level Memory

- This figure shows the general shape of the curve that covers the situation in the previous example.
- The figure shows the average access time to a two-level memory as a function of the hit ratio H, where *H is defined as the fraction of all memory accesses that are found in the faster memory (e.g., the cache),*
- T_1 is the access time to level 1, and T_2 is the access time to level 2.
- As can be seen, for high percentages of level 1 access, the average total access time is much closer to that of level 1 than that of level 2.



Summary and Case Studies

Figure 4.9

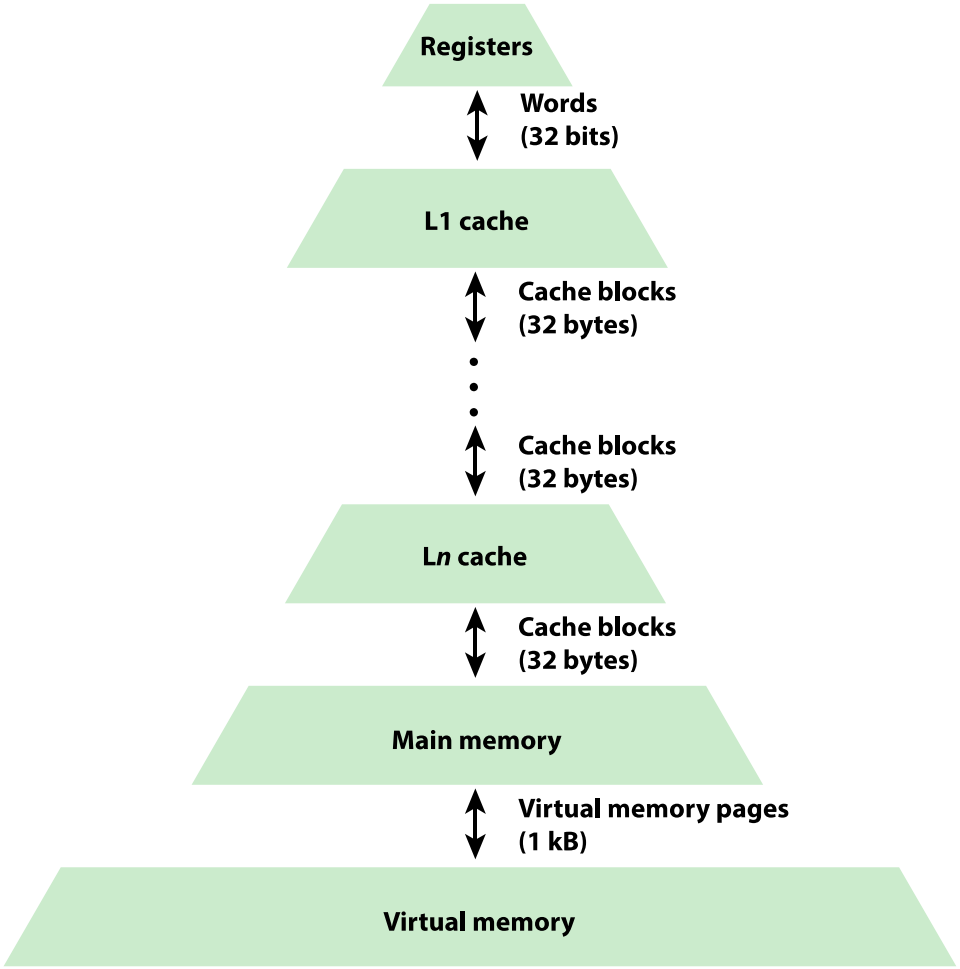


Figure 4.9 Exploiting Locality in the Memory Hierarchy (with typical transfer size)

Table 4.2
Characteristics of Memory Devices in a Memory Architecture

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

Table 4.2 Characteristics of Memory Devices in a Memory Architecture

Memory

- The use of three levels exploits the fact that semiconductor memory comes in a variety of types which differ in speed and cost
- Data are stored more permanently on external mass storage devices
- External, nonvolatile memory is also referred to as **secondary** memory or **auxiliary** memory

Figure 4.10

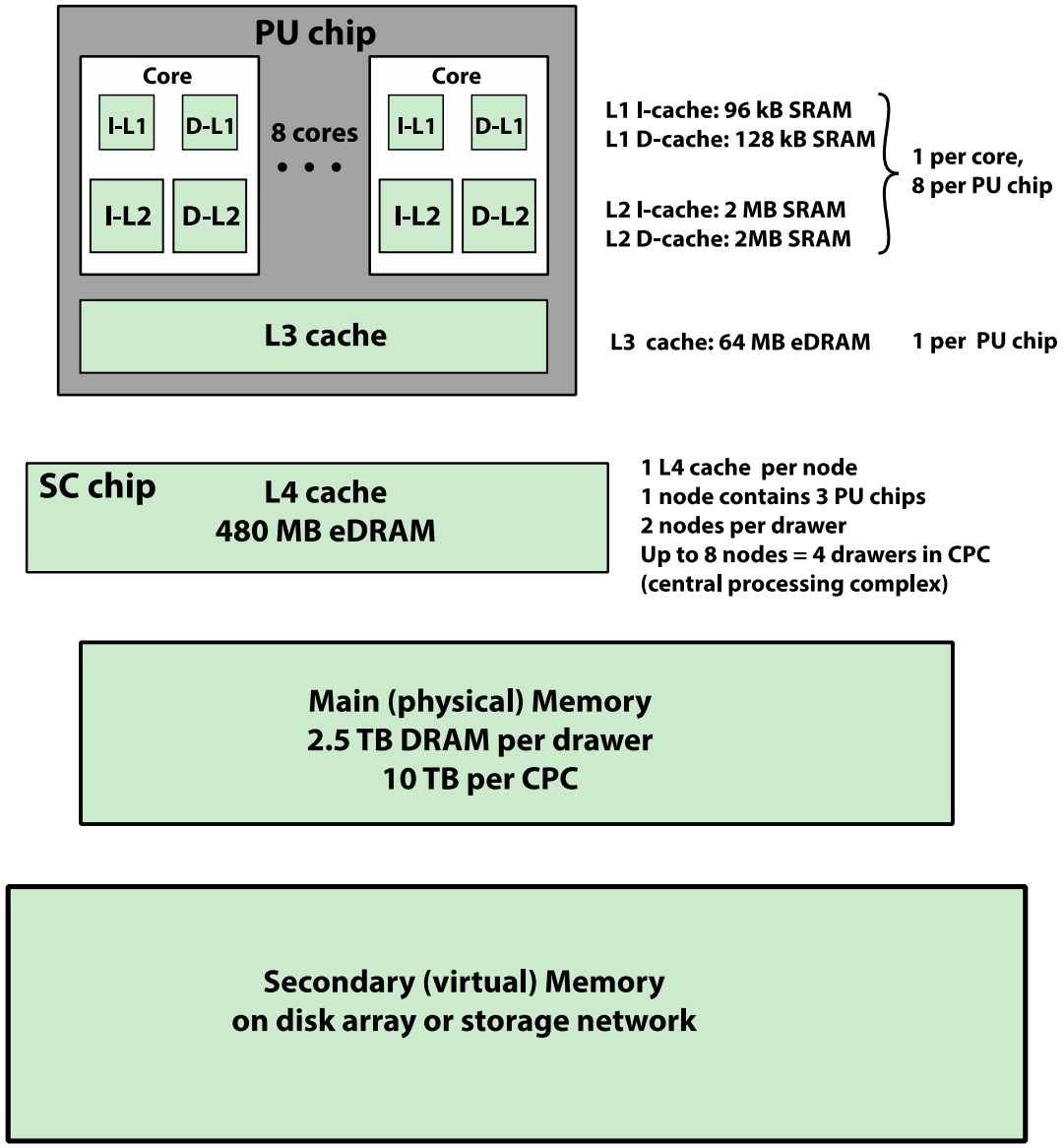
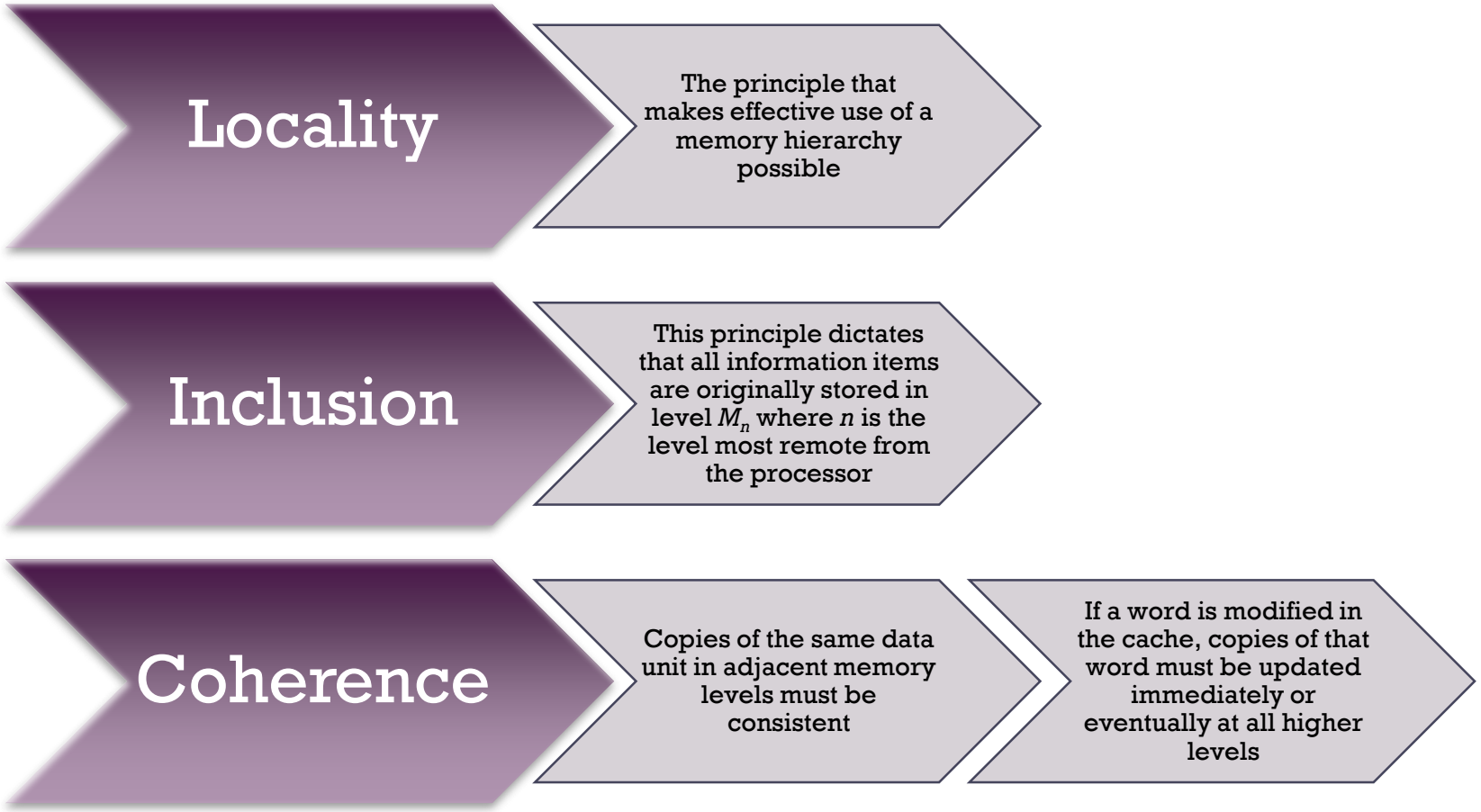


Figure 4.10 IBM z13 Memory Hierarchy

Design Principles for a Memory Hierarchy



The End