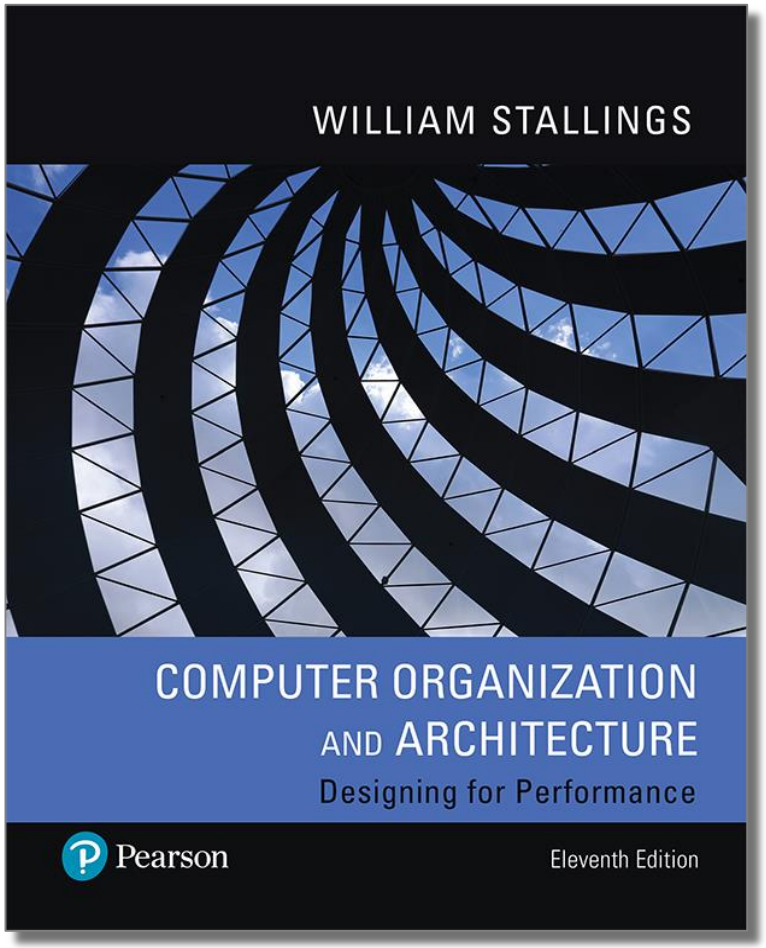
Computer Organization and Architecture

Designing for Performance

11th Edition



Chapter 4

The Memory Hierarchy: Locality and Performance

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

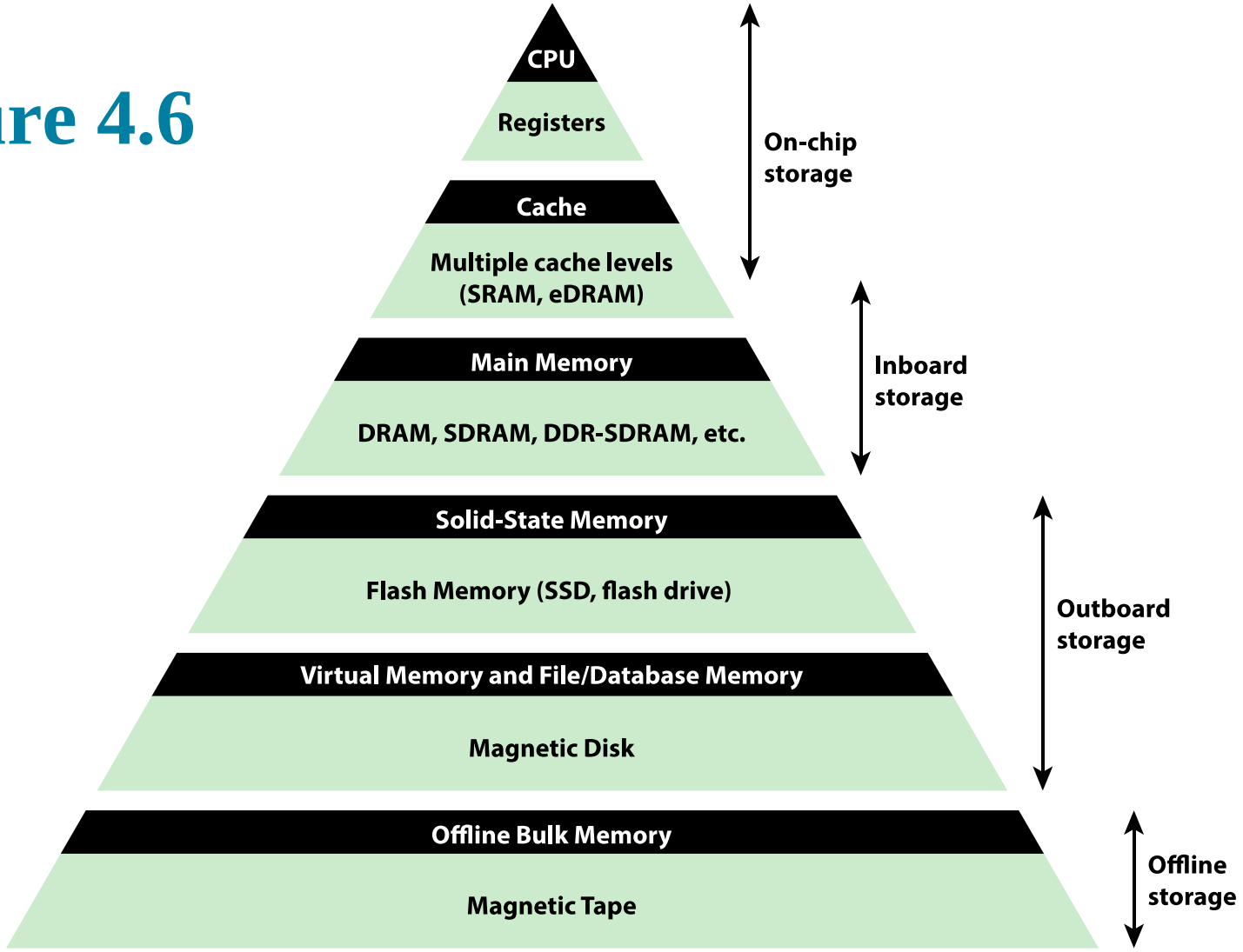


Figure 4.6 The Memory Hierarchy

Figure 4.7

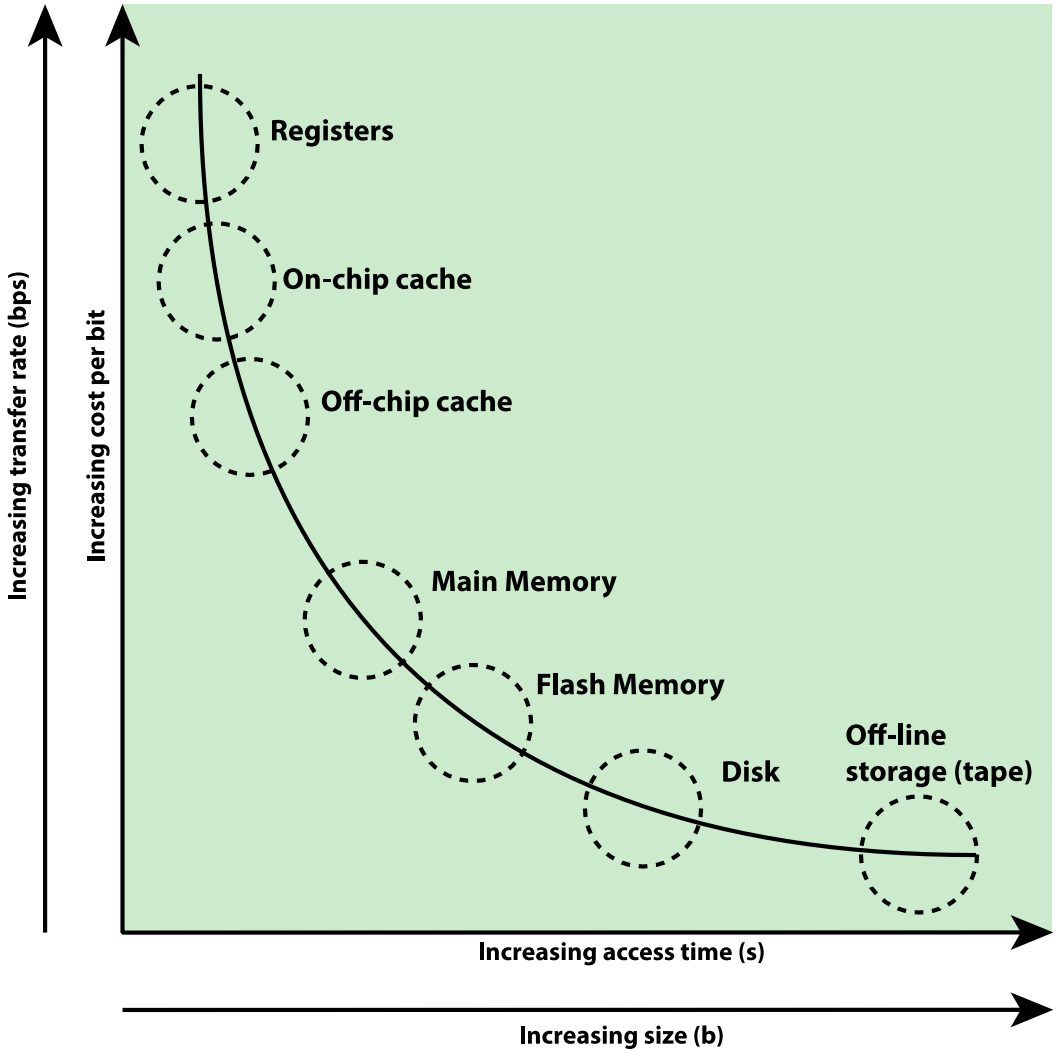


Figure 4.7 Relative Cost, Size, and Speed Characteristics Across the Memory Hierarchy

Memory Access Time

- Suppose that the processor has access to two levels of memory. Level 1 contains X words and has an access time of 0.01 μs; level 2 contains 1000 x X 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.
- Suppose 95% of the memory accesses are found in level 1. Then the average time to access a word can be expressed as:

(0.95)(0.01 μs) + (0.05)(0.01 μs + 0.1 μs) = 0.0095 + 0.0055 = 0.015 μs

Goal? Hit level 1 as much as possible. Hit level 2 as least as possible.

Figure 4.8

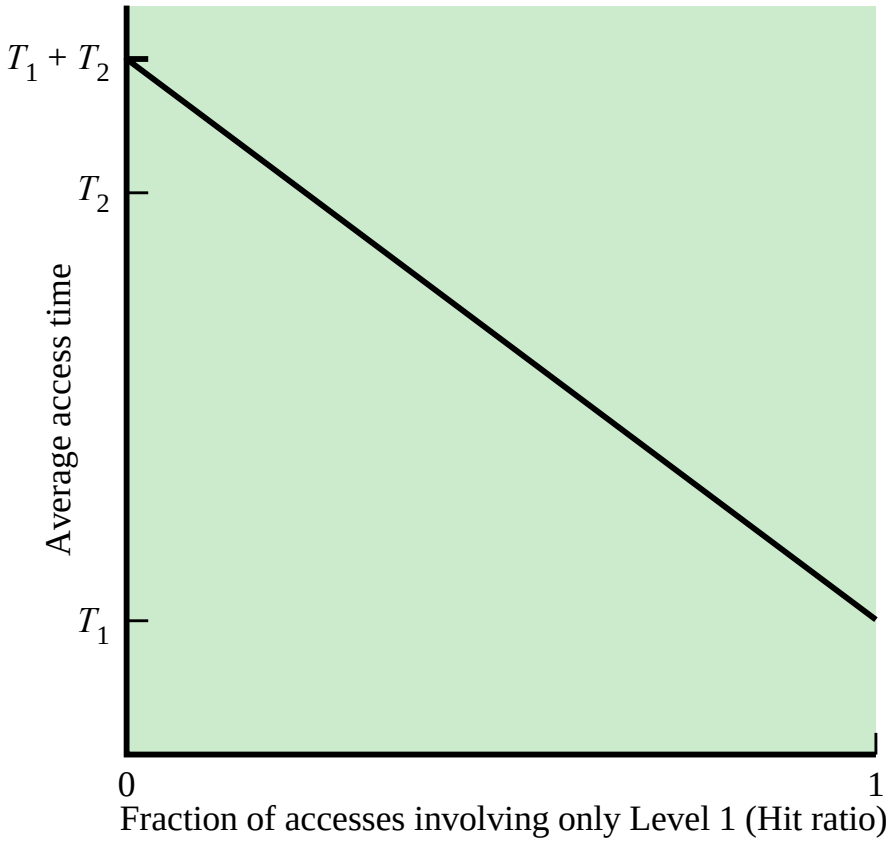


Figure 4.8 Performance of a Simple Two-Level Memory

Figure 4.9

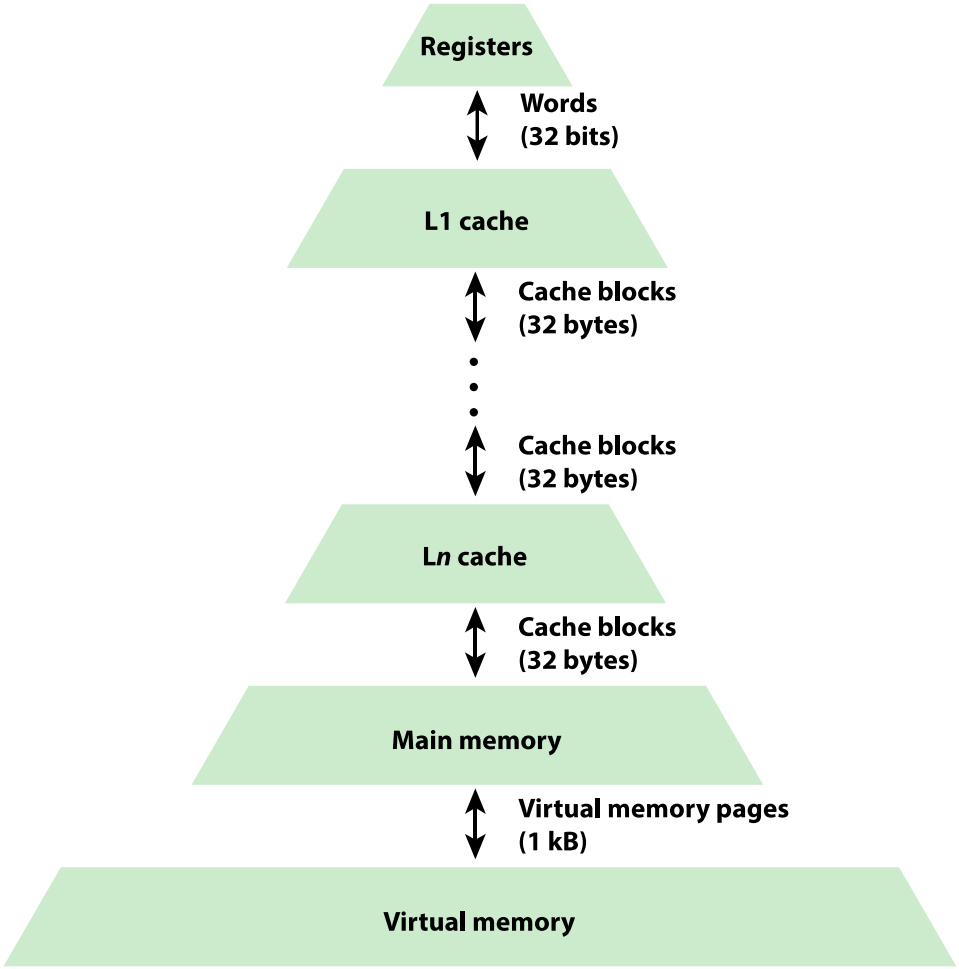


Figure 4.9 Exploiting Locality in the Memory Hierarchy (with typical transfer size)

Memory Hierarchy

- The fastest, smallest, and most expensive type of memory consists of the **registers** internal to the processor. Typically, a few dozen, although some machines contain hundreds of registers.
- 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.
- L2 cache is similar to L2 with respect to the split.

Memory Hierarchy

- L3 cache and L4 cache generally are not split between instruction and data and may be shared by multiple processors.
- Traditionally, cache memory has been constructed using a technology called static random access memory (SRAM).
- More recently, higher levels of cache on many systems have been implemented using embedded dynamic RAM (eDRAM), which is slower than SRAM but faster than the DRAM used to implement the main memory of the computer.

Memory Hierarchy

- 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. The various levels of cache are controlled by hardware and are used for staging the movement of data between main memory and processor registers to improve performance.

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

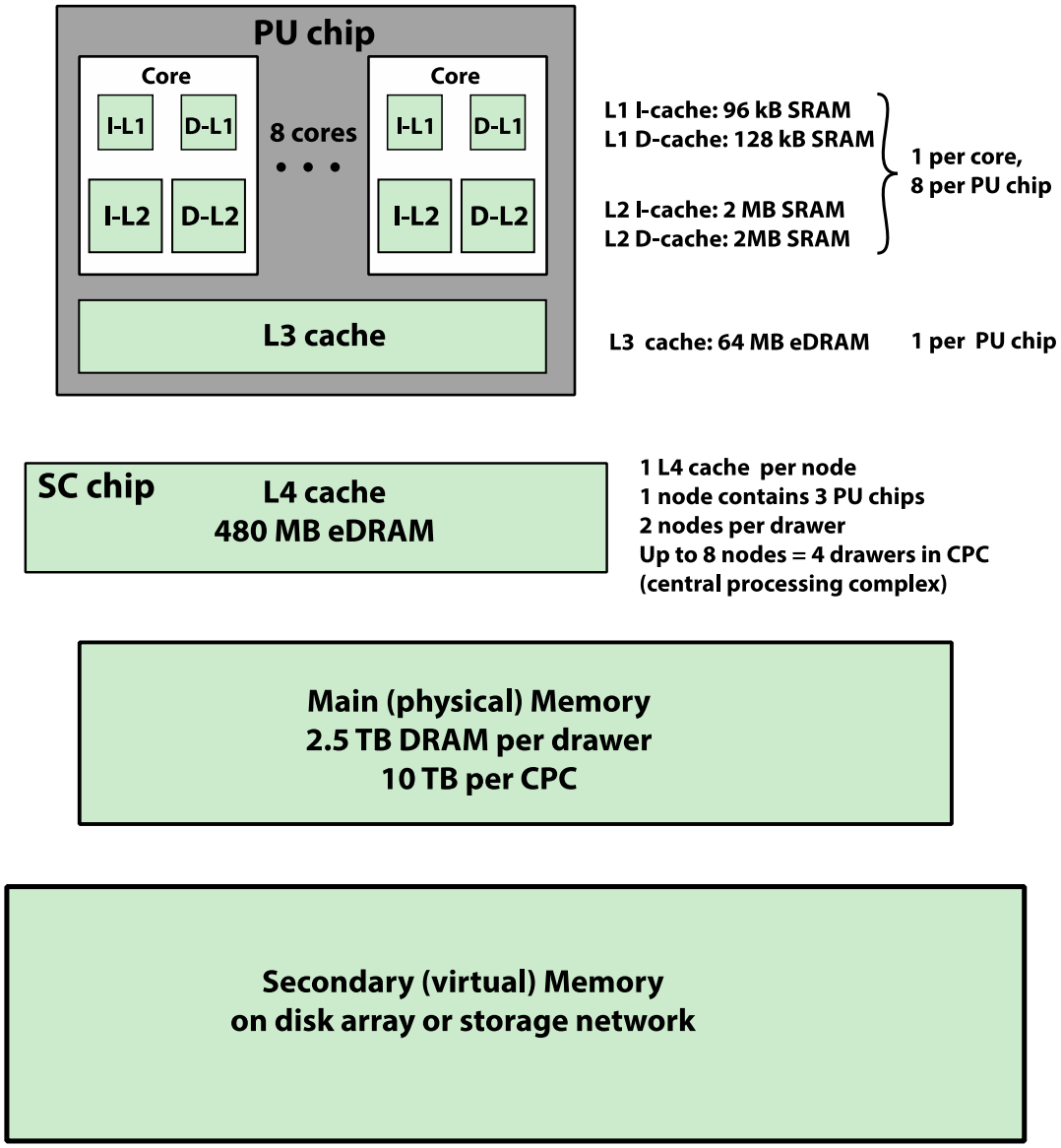
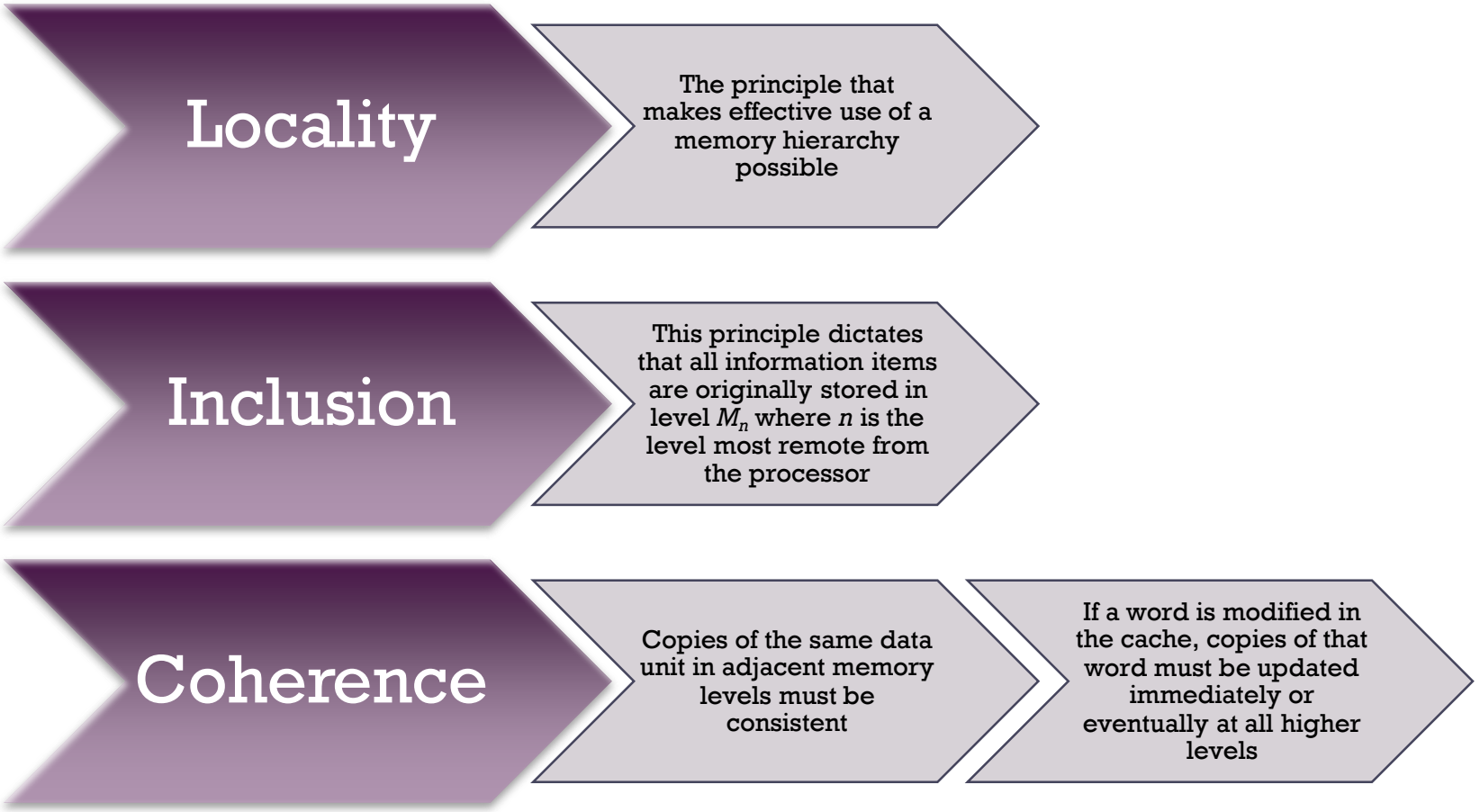


Figure 4.10 IBM z13 Memory Hierarchy

Design Principles for a Memory Hierarchy



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

Figure 4.11

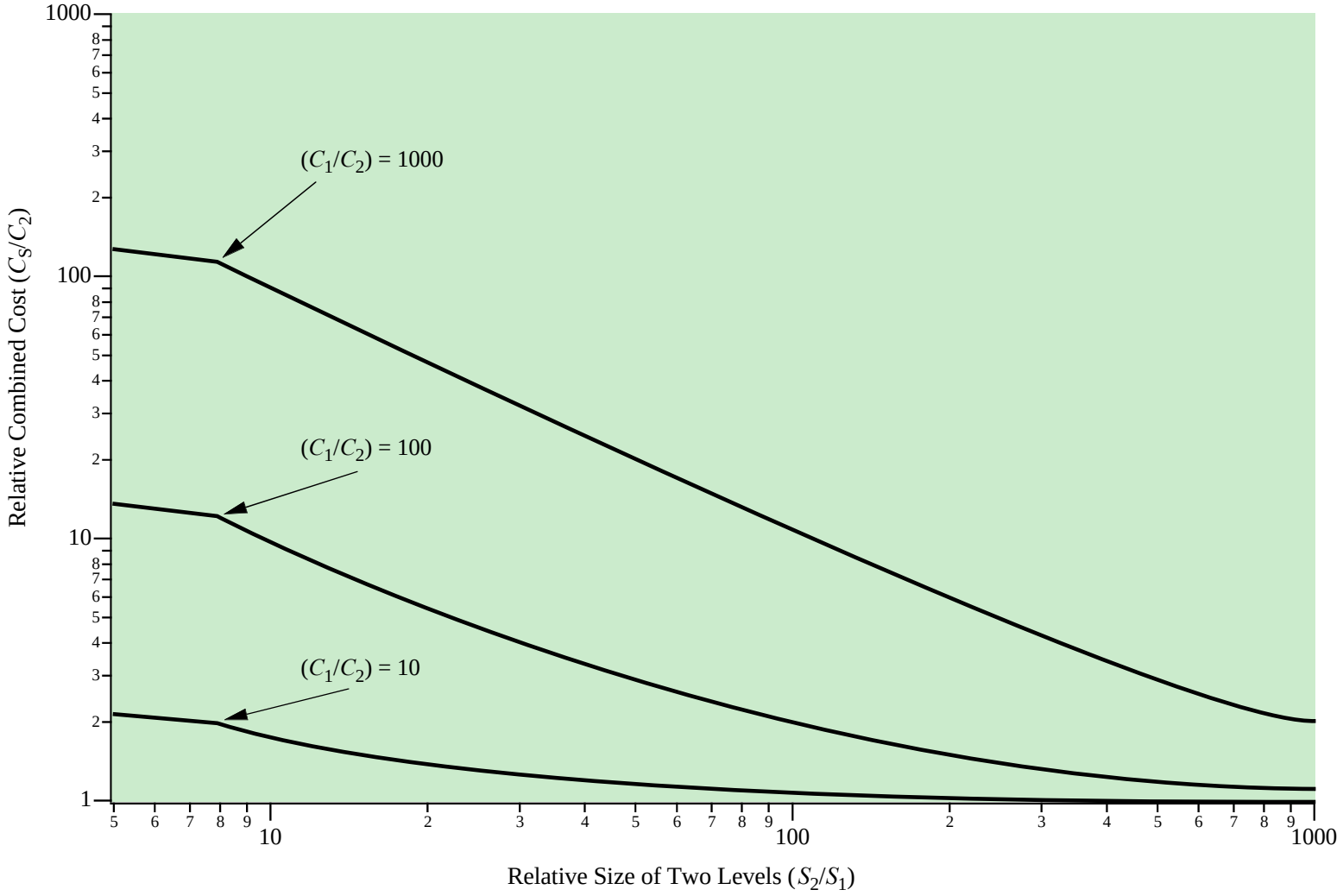


Figure 4.11 Relationship of Average Memory Cost to Relative Memory Size for a Two-Level Memory

Figure 4.12

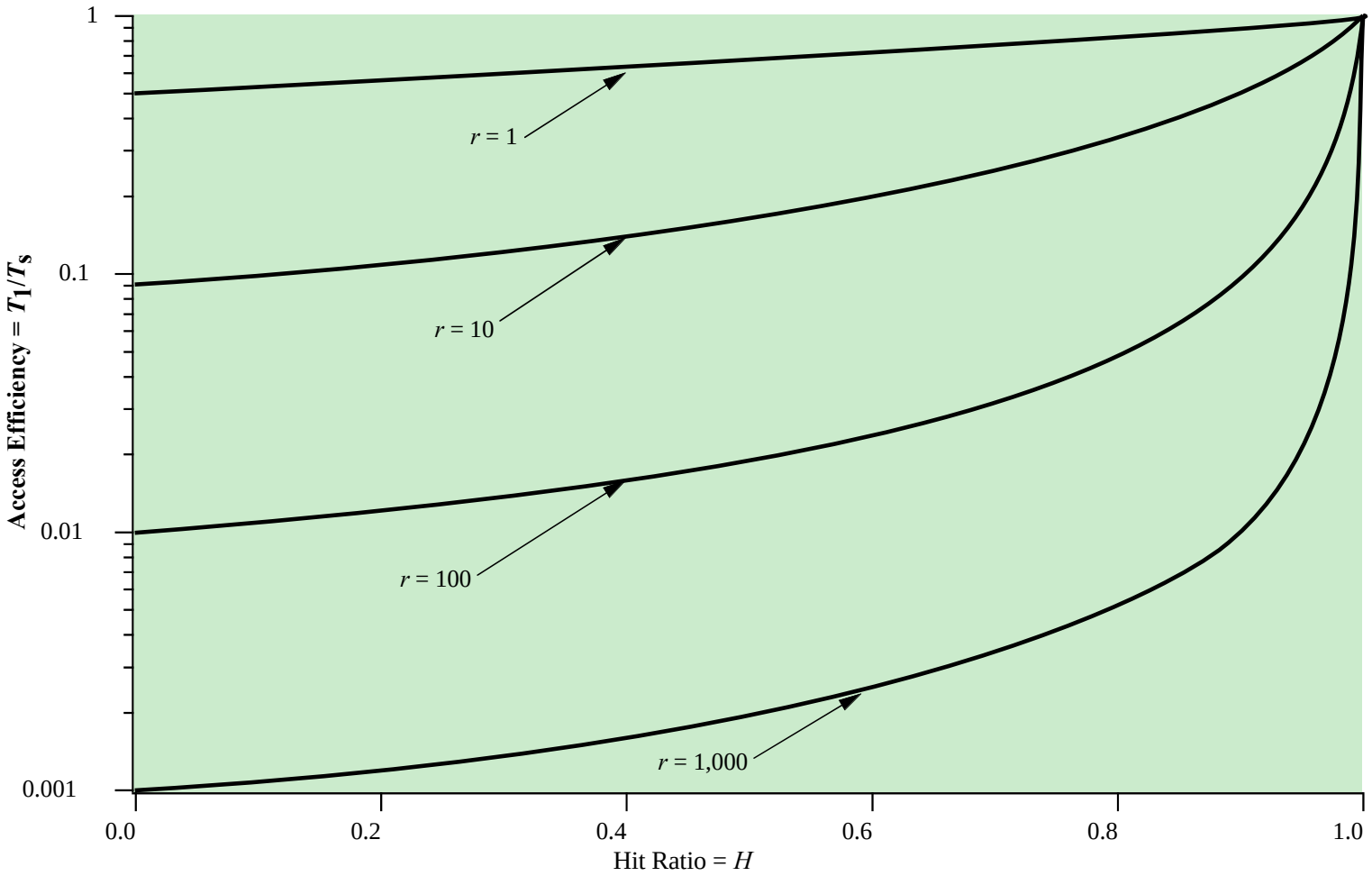


Figure 4.12 Access Efficiency as a Function of Hit Ratio ($r = T_2/T_1$)

Figure 4.13

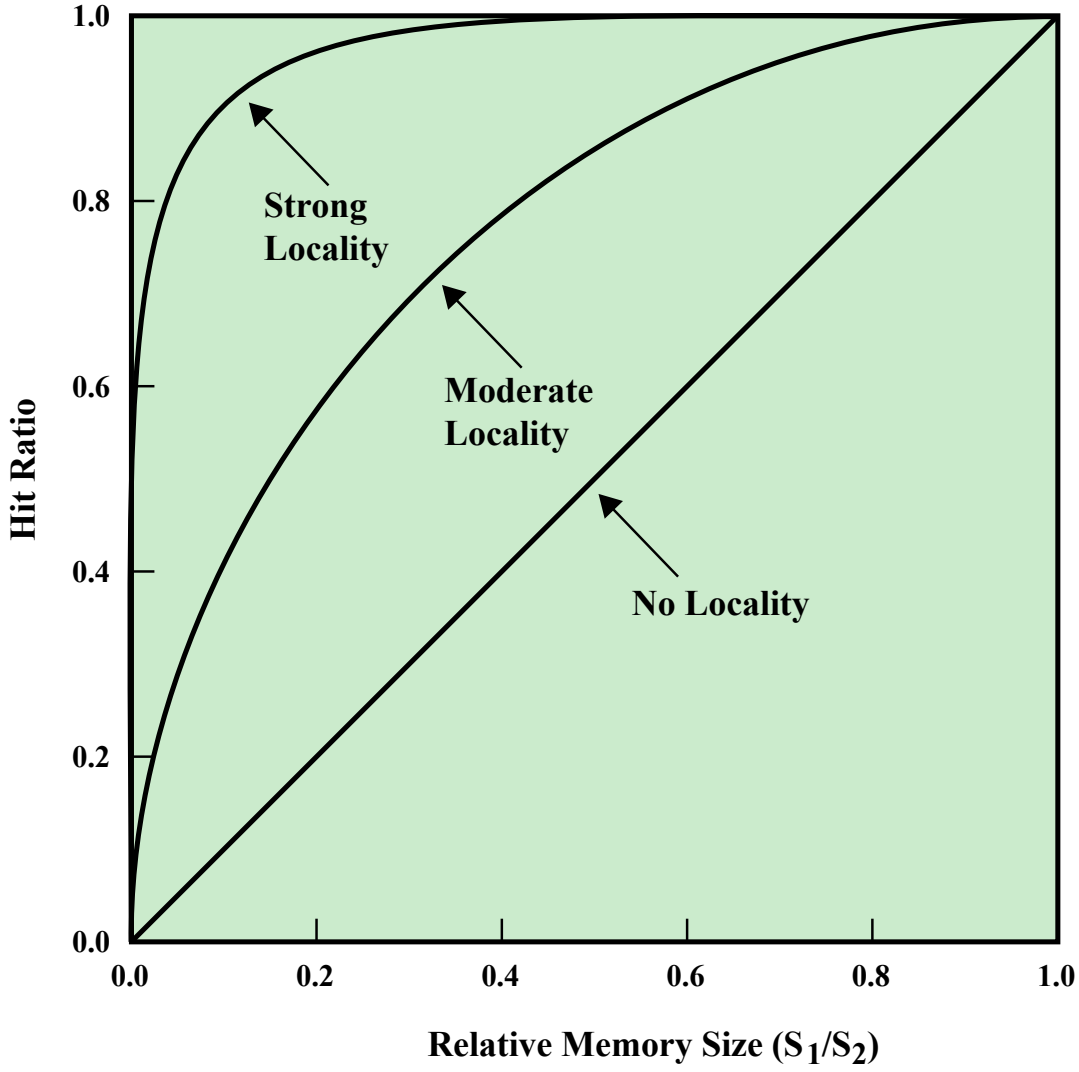


Figure 4.13 Hit Ratio as a Function of Relative Memory Size

Figure 4.14

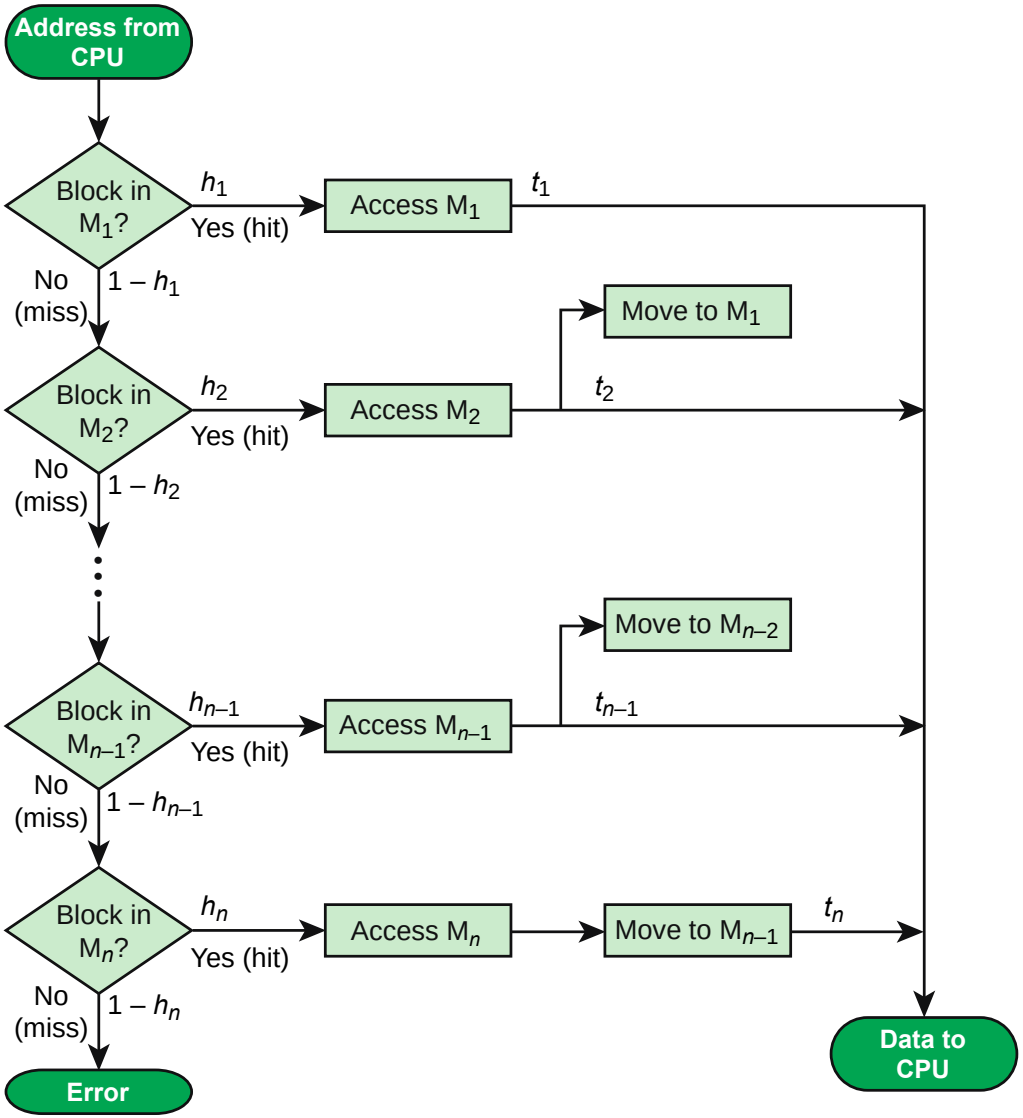


Figure 4.14 Multilevel Memory Access Performance Model

Summary

Chapter 4

- Principle of locality
- Characteristics of memory systems
- Performance modeling of a multilevel memory hierarchy
  - Two-level memory access
  - Multilevel memory access

The Memory Hierarchy: Locality and Performance

- The memory hierarchy
  - Cost and performance characteristics
  - Typical members of the memory hierarchy
  - The IBM z13 memory hierarchy
  - Design principles for a memory hierarchy

Copyright