

### LEARNING OBJECTIVES

After studying this chapter, you should be able to:

- ◆ Present an overview of the main characteristics of computer memory systems and the use of a memory hierarchy.
- ◆ Describe the basic concepts and intent of cache memory.
- ◆ Discuss the key elements of cache design.
- ◆ Distinguish among direct mapping, associative mapping, and set-associative mapping.
- ◆ Explain the reasons for using multiple levels of cache.
- ◆ Understand the performance implications of multiple levels of memory.

Although seemingly simple in concept, computer memory exhibits perhaps the widest range of type, technology, organization, performance, and cost of any feature of a computer system. No single technology is optimal in satisfying the memory requirements for a computer system. As a consequence, the typical computer system is equipped with a hierarchy of memory subsystems, some internal to the system (directly accessible by the processor) and some external (accessible by the processor via an I/O module).

This chapter and the next focus on internal memory elements, while Chapter 6 is devoted to external memory. To begin, the first section examines key characteristics of computer memories. The remainder of the chapter examines an essential element of all modern computer systems: cache memory.

## 4.1 COMPUTER MEMORY SYSTEM OVERVIEW

### Characteristics of Memory Systems

The complex subject of computer memory is made more manageable if we classify memory systems according to their key characteristics. The most important of these are listed in Table 4.1.

The term **location** in Table 4.1 refers to whether memory is **internal or external to the computer**. **Internal memory** is often equated with **main memory**, but there are other forms of internal memory. The **processor** requires its **own local memory**, in the form of **registers** (e.g., see Figure 2.3). Further, as we will see, the **control unit** portion of the **processor** may also **require** its **own internal memory**. We will defer discussion of these latter two types of internal memory to later chapters. **Cache** is another form of **internal memory**. **External memory** consists of **peripheral storage devices**, such as **disk** and **tape**, that are **accessible** to the processor via **I/O controllers**.

An obvious characteristic of memory is its **capacity**. For **internal memory**, this is typically expressed in terms of **bytes** (1 byte = 8 bits) or **words**. Common **word lengths** are **8, 16, and 32 bits**. **External memory capacity** is typically expressed in terms of **bytes**.

**Table 4.1** Key Characteristics of Computer Memory Systems

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

A related concept is the **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. This may be equal to the word length, but is often larger, such as 64, 128, or 256 bits. To clarify this point, consider three related concepts for internal memory:

- **Word:** The “natural” unit of organization of memory. The size of a word is typically equal to the number of bits used to represent an integer and to the instruction length. Unfortunately, there are many exceptions. For example, the CRAY C90 (an older model CRAY supercomputer) has a 64-bit word length but uses a 46-bit integer representation. The Intel x86 architecture has a wide variety of instruction lengths, expressed as multiples of bytes, and a word size of 32 bits.
- **Addressable units:** In some systems, the addressable unit is the word. However, many systems allow addressing at the byte level. In any case, the relationship between the length in bits  $A$  of an address and the number  $N$  of addressable units is  $2^A = N$ .
- **Unit of transfer:** For main memory, this is the number of bits read out of or written into memory at a time. The unit of transfer need not equal a word or an addressable unit. For external memory, data are often transferred in much larger units than a word, and these are referred to as blocks.

Another distinction among memory types is the **method of accessing** units of data. These include the following:

- **Sequential access:** Memory is organized into units of data, called records. Access must be made in a specific linear sequence. Stored addressing information is used to separate records and assist in the retrieval process. A shared read–write mechanism is used, and this must be moved from its current location to the desired location, passing and rejecting each intermediate record. Thus, the time to access an arbitrary record is highly variable. Tape units, discussed in Chapter 6, are sequential access.
- **Direct access:** As with sequential access, direct access involves a shared read–write mechanism. However, individual blocks or records have a unique

address based on physical location. Access is accomplished by direct access to reach a general vicinity plus sequential searching, counting, or waiting to reach the final location. Again, access time is variable. Disk units, discussed in Chapter 6, are direct access.

- **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. Thus, any location can be selected at random and directly addressed and accessed. Main memory and some cache systems are random access.
- **Associative:** This is a random access type of memory that enables one to make a comparison of desired bit locations within a word for a specified match, and to do this for all words simultaneously. Thus, a word is retrieved based on a portion of its contents rather than its address. As with ordinary random-access memory, 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.

From a user's point of view, the two most important characteristics of memory are capacity and performance. Three performance parameters are used:

- **Access time (latency):** For random-access memory, this is the time it takes to perform a read or write operation, that is, the time from the instant that an address is presented to the memory to the instant that data have been stored or made available for use. For non-random-access memory, access time is the time it takes to position the read-write mechanism at the desired location.
- **Memory cycle time:** This concept is primarily applied to random-access memory and consists of the access time plus any additional time required before a second access can commence. This additional time may be required for transients to die out on signal lines or to regenerate data if they are read destructively. Note that memory cycle time is concerned with the system bus, not the processor.
- **Transfer rate:** This is the rate at which data can be transferred into or out of a memory unit. For random-access memory, it is equal to  $1/(\text{cycle time})$ . For non-random-access memory, the following relationship holds:

$$T_n = T_A + \frac{n}{R} \quad (4.1)$$

where

$T_n$  = Average time to read or write  $n$  bits

$T_A$  = Average access time

$n$  = Number of bits

$R$  = Transfer rate, in bits per second (bps)

A variety of physical types of memory have been employed. The most common today are semiconductor memory, magnetic surface memory, used for disk and tape, and optical and magneto-optical.

Several **physical characteristics** of data storage are **important**. In a **volatile memory**, information **decays naturally** or is **lost** when **electrical power** is switched off. In a **nonvolatile memory**, information once **recorded** remains without deterioration until **deliberately changed**; **no electrical power** is **needed** to **retain information**. Magnetic-surface memories are **nonvolatile**. Semiconductor memory (memory on integrated circuits) 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)**. Of necessity, a **practical nonerasable memory** must also be **nonvolatile**.

For **random-access memory**, the **organization** is a key design issue. In this **context**, **organization** refers to the **physical arrangement** of **bits** to **form words**. The **obvious arrangement** is **not always used**, as is explained in Chapter 5.

### The Memory Hierarchy

The design constraints on a computer's memory can be summed up by three questions: How much? How fast? How expensive?

The question of how much is somewhat open ended. If the capacity is there, applications will likely be developed to use it. The question of how fast is, in a sense, easier to answer. To achieve greatest performance, the memory must be able to keep up with the processor. That is, as the processor is executing instructions, we would not want it to have to pause waiting for instructions or operands. The final question must also be considered. For a practical system, the cost of memory must be reasonable in relationship to other components.

As might be expected, there is a trade-off among the three key characteristics of memory: capacity, access time, and cost. A variety of technologies are used to implement memory systems, and across this spectrum of technologies, the following relationships hold:

- Faster access time, greater cost per bit;
- Greater capacity, smaller cost per bit;
- Greater capacity, slower access time.

The dilemma facing the designer is clear. The designer would like to use memory technologies that provide for large-capacity memory, both because the capacity is needed and because the cost per bit is low. However, to meet performance requirements, the designer needs to use expensive, relatively lower-capacity memories with short access times.

The way out of this dilemma is not to rely on a single memory component or technology, but to employ a **memory hierarchy**. A typical hierarchy is illustrated in Figure 4.1. As one goes down the hierarchy, the following occur:

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

Thus, smaller, more expensive, faster memories are supplemented by larger, cheaper, slower memories. The key to the success of this organization