ISBN/ISSN: 9780138146467

13–1 THE BASIC COMPUTER

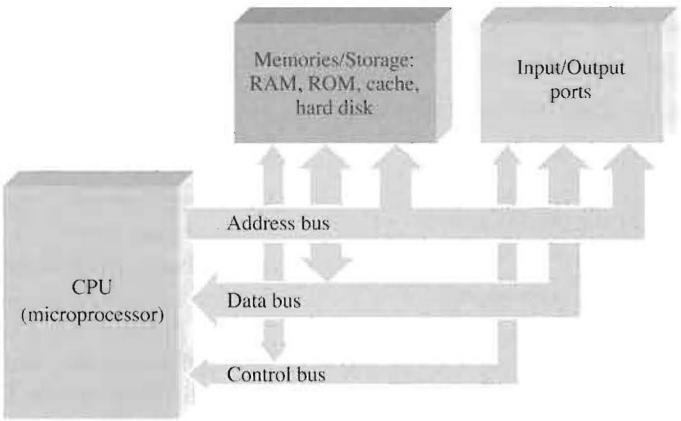
Special-purpose computers control various functions in automobiles or appliances, control manufacturing processes in industry, provide games for entertainment, and are used in navigation systems such as GPS (Global Positioning System), to name a few areas. However, the most familiar type of computer is the general-purpose computer that can be programmed to do many different types of things.

After completing this section, you should be able to

- Describe the basic elements in a computer
- Discuss what each part of a computer does
- Explain what a peripheral device is

All computers consist of basic functional blocks that include a central processing unit (CPU), memory, and input/output ports. These functional blocks are connected together with three internal buses, as shown in the block diagram of Figure 13–1. The three buses are the data bus, the address bus, and the control bus. Input and output devices are connected through the input/output ports. A **port** is a physical interface on a computer through which data are passed to and from peripherals.

▶ FIGURE 13–1
Basic computer block diagram.



COMPUTER NOTE

Grace Hopper, a mathematician and pioneer programmer, developed considerable troubleshooting skills as a naval officer working with the Harvard Mark I computer in the 1940s. She found and documented in the Mark I's log the first real computer bug. It was a moth that had been trapped in one of the electromechanical relays inside the machine, causing the computer to malfunction. From then on, when asked if anything was being accomplished, those working on the computer would reply that they were "debugging" the system. The term stuck, and finding problems in a computer (or other electronic device), particularly the software, would always be known as debugging.

Instructions and data are stored in memory in specific locations determined by the **program**, a list of instructions designed to solve a specific problem. Each location has a unique address associated with it. Instructions are obtained by the CPU by placing an address on the address bus. Instructions are transferred via the data bus as they are requested by the CPU. The CPU executes the instructions sequentially; frequently, the instructions modify data stored in memory or obtained from an input device. Processed data may be stored back in memory or sent to an output device via the data bus. Signals on the control bus are generated by the CPU to coordinate all of these operations.

Central Processing Unit (CPU)

The **CPU** is the "brain" of the computer; it oversees everything that the computer does. The CPU is a microprocessor with associated circuits that control the running of the computer software programs. Basically, the CPU obtains (fetches) each program instruction from memory and carries out (executes) the instruction.

After completing one instruction, the CPU moves on to the next one and in most cases can operate on more than one instruction at the same time. This "fetch and execute" process is repeated until all of the instructions in a specific program have been executed.

For example, an application program may require the sum of a series of numbers. The instructions to add the numbers are stored in the form of binary codes that direct the CPU to fetch a series of numbers from memory, add them, and store the sum back in memory.

Memories and Storage

Several types of memories are used in a typical computer. The *RAM* (random-access memory) stores binary data and programs temporarily during processing. Data are numbers and other information, and programs are lists of instructions. Data can be written into and read out of a RAM at any time. The RAM is volatile, meaning that the information is lost if power is turned off or fails. Therefore, any data or program that needs to be saved should be moved to nonvolatile memory (such as a CD or hard disk) before power is removed.

The *ROM* (read-only memory) stores a permanent system program called the **BIOS** (Basic Input/Output System) and certain locations of system programs in memory. The ROM is nonvolatile, which means it retains what is stored, even when the power is *off*. As the name implies, the programs and data in ROM cannot be altered. Sometimes it is referred to as "firmware" because it is permanent software for a given system.

The *BIOS* is the lowest level of the computer's operating system. It contains instructions that tell the CPU what to do when power is first applied; the first instruction executed is in the BIOS. It controls the computer's basic start-up functions that include a self-test and a disk self-loader to bring up the rest of the operating system. In addition, the BIOS stores locations of system programs that handle certain requests from peripherals called **interrupts,** which cause the current processing to be temporarily stopped.

The *cache* memory is a small RAM that is used to store a limited amount of frequently used data that can be accessed much faster than the main RAM. The cache stores "close at hand" information that will be used again instead of having to retrieve it from farther away in the main memory. Most microprocessors have internal cache memory called level-1, or simply L1. External cache memory is in a separate memory chip and is referred to as level-2, or L2.

The *hard disk* is the major storage medium in a computer because it can store large amounts of data and is nonvolatile. The high-level operating systems as well as applications software and data files are all stored on the hard disk.

Removable storage is part of most computer systems. The most common types of removable storage media are the CDs and Zip disks (magnetic storage media). CDs are available as CD-ROMs (Compact Disk–Read-Only Memory) and as CD-RWs (Rewritable) and can store huge amounts of data (typically 650 MB). Zip drives typically store 250 MB.

Input/Output Ports

Generally, the computer sends data to a peripheral device through an output port and receives information through an input port. Ports can be configured in software to be either an input or output port. The keyboard, mouse, video monitor, printer, and other peripherals communicate to the CPU through individual ports. Ports are generally classified as either serial ports, with a single data line, or parallel ports, with multiple data lines.

Buses

Peripherals are connected to the computer ports with standard interface buses. A bus can be thought of as a highway for digital signals that consists of a set of physical connections, as well as electrical specifications for the signals. Examples of serial buses are FireWire and USB (Universal Serial Bus), which has essentially become a de facto standard. The most common parallel bus is simply called the *parallel bus,* which connects to a port commonly referred to as the printer port (although this port can be used by other peripherals.) Another example of a parallel bus, for connecting lab instruments to a computer, is called the General Purpose Interface Bus (GPIB).

The three basic types of internal buses that interconnect the CPU with memory and storage and with input and output ports are the address bus, data bus, and control bus. These buses are usually lumped into what is called the *local bus*. The address bus is used by the CPU to specify memory locations or addresses and to select ports. The data bus is used to transfer program instructions and data between the CPU, memories, and ports. The control bus is used for transferring control signals to and from the CPU.

Computer Software

In addition to the hardware, another major aspect of a computer is the software. The software makes the hardware perform. The two major categories of software used in computers are system software and applications software.

System Software The system software is called the operating system of a computer and allows the user to interface with the computer. The most common operating systems used in desktop and laptop computers are Windows and MacOS. Many other operating systems are used in special-purpose computers and in mainframe computers.

System software performs two basic functions. It manages all the hardware and software in a computer. For example, the operating system manages and allots space on the hard disk. It also provides a consistent interface between applications software and hardware. This allows an applications program to work on various computers that may differ in hardware details.

The operating system on your computer allows you to have several programs running at the same time. This is called *multitasking*. For example, you can be using the word processor while downloading something from the Internet and printing an e-mail message.

Applications Software Applications software is used to accomplish a specific job or task. Table 13–1 lists several types of applications software.

TABLE 13–1
Applications software.

APPLICATION	FUNCTION	EXAMPLES
Accounting	Tax preparation, bookkeeping	Quickbooks, Turbotax, MYOB
Circuit simulation	Create and test electronic circuits	Multisim
Data management	Manipulate large databases	Filemaker, Access
Desktop publishing	Prepare newsletters, flyers, books, and other printed material	Quark XPress, Pagemaker
Drawing	Prepare technical drawings and pictures	CorelDraw, Freehand, Illustrator
Multimedia	Digital video editing, produce moving images in presentations	Premier, Dreamweaver, After Effects
Photography	Manipulate digital pictures, add special effects to pictures	Photoshop, Image Expert
Presentations	Prepare slide shows and technical presentations	PowerPoint, Harvard Graphics
Speech recognition	Converts speech to text	NaturallySpeaking
Spreadsheet	Manipulate numbers and words in an array	Excel, Lotus 123
Website preparation	Tools to create web pages and websites on the Internet	FrontPage, Acrobat
Word processing	Prepare text documents and letters	Microsoft Word

Sequence of Operation When you first turn on your computer, this is what happens:

1. BIOS from ROM is loaded into RAM and a self-test is performed to check all major components and memory. Also, the BIOS provides information about storage, boot sequence, and the like.
2. The operating system (such as Windows) on the hard disk is loaded into RAM.
3. Application programs (such as Microsoft Word) are stored on the hard disk. When you select one, it is loaded into RAM. Sometimes, only portions are loaded as needed.
4. Files required by the application are loaded from the hard disk into RAM.
5. When a file is saved and the application is closed, the file is written back to the hard disk and both the application and the file are removed from RAM.

The Computer System

The block diagram in Figure 13–2 shows the main elements in a typical computer system and how they are interconnected. For a computer to accomplish a given task, it must communicate with the "outside world" by interfacing with people, sensing devices, or devices to be controlled in some way. To do this, there is a keyboard, a mouse, a video monitor, a printer, a modem, and a CD drive in most basic systems. These are called **peripherals.**

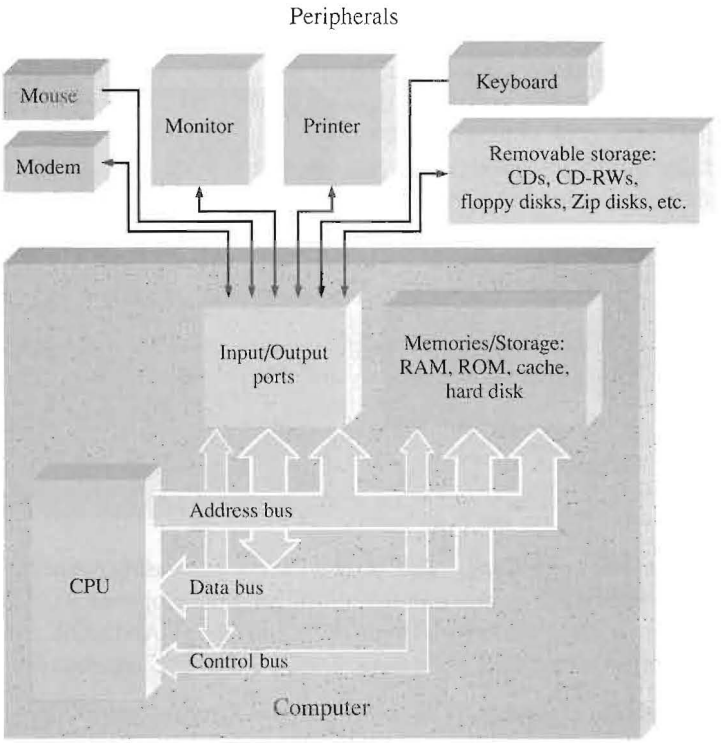


FIGURE 13–2
Basic block diagram of a typical computer system including common peripherals. The computer itself is shown within the gray block.

SECTION 13–1 CHECKUP
Answers are at the end of the chapter.

1. What are the major elements or blocks in a computer?
2. What is the difference between RAM and ROM?
3. What are peripherals?
4. What is the difference between computer hardware and computer software?

13–2 THE MICROPROCESSOR

The **microprocessor** is a digital integrated circuit that can be programmed with a series of instructions to perform various operations on data. A microprocessor is the CPU of a computer. It can do arithmetic and logic operations, move data from one place to another, and make decisions based on certain instructions.

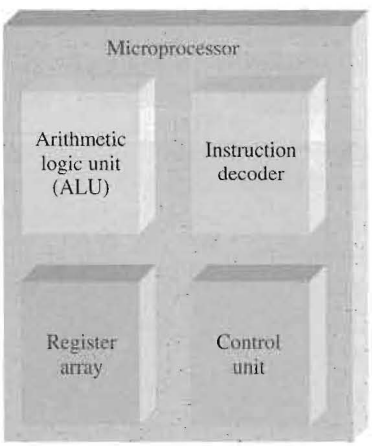
After completing this section, you should be able to

- Describe the basic elements of a microprocessor
- Discuss microprocessor buses
- Discuss a microprocessor instruction set
- Discuss multi-core processors
- Describe pipelining, multitasking, and multithreading

Basic Elements

A microprocessor consists of several units, each designed for a specific job. The specific units, their design and organization, are called the *architecture* (do not confuse the term with the VHDL element). The architecture determines the instruction set and the process for executing those instructions. Four basic units that are common to all microprocessors are the arithmetic logic unit (ALU), the instruction decoder, the register array, and the control unit, as shown in Figure 13–3.

FIGURE 13–3



Arithmetic Logic Unit The **ALU** is the key processing element of the microprocessor. It is directed by the control unit to perform arithmetic operations (addition, subtraction, multiplication, and division) and logic operations (NOT, AND, OR, and exclusive-OR), as well as many other types of operations. Data for the ALU are obtained from the register array.

Instruction Decoder The instruction decoder can be considered as part of the ALU, although we are treating it as a separate function in this discussion because the instructions and the decoding of them are key to a microprocessor's operation. The microprocessor accomplishes a given task as directed by programs that consist of lists of instructions stored in memory. The instruction decoder takes each binary instruction in the order in which it appears in memory and decodes it.

Register Array The **register array** is a collection of registers that are contained within the microprocessor. During the execution of a program, data and memory addresses are temporarily stored in registers that make up this array. The ALU can access

the registers very quickly, making the program run more efficiently. Some registers are classed as general-purpose, meaning they can be used for any purpose dictated by the program. Other registers have specific capabilities and functions and cannot be used as general-purpose registers. Still others are called program invisible registers, used only by the microprocessor and not available to the programmer.

Control Unit The **control unit** is "in charge" of the processing of instructions once they are decoded. It provides the timing and control signals for getting data into and out of the microprocessor and for synchronizing the execution of instructions.

Microprocessor Buses

The three buses mentioned earlier are the connections for microprocessors to allow data, addresses, and instructions to be moved.

The Address Bus The **address bus** is a "one-way street" over which the microprocessor sends an address code to a memory or other external device. The size or width of the address bus is specified by the number of conductive paths or bits. Early microprocessors had sixteen address lines that could select 65,536 (2^{16}) unique locations in memory. The more bits there are in the address, the higher the number of memory locations that can be accessed. The number of address bits has advanced to the point where processors have up to 64 address bits and can access over 1.8×10^{19} memory locations.

The Data Bus The **data bus** is a "two-way street" on which data or instruction codes are transferred into the microprocessor or the result of an operation or computation is sent out. The original microprocessors had 8-bit data buses. Today's microprocessors have up to 64-bit data buses.

The Control Bus The **control bus** is used by the microprocessor to coordinate its operations and to communicate with external devices. The control bus has signals that enable either a memory or an input/output operation at the proper time to read or write data. Control bus lines are also used to insert special wait states for slower devices and prevent bus contention, a condition that can occur if two or more devices try to communicate at the same time.

Microprocessor Programming

All microprocessors work with an instruction set that implements the basic operations. The Pentium, for example, has hundreds of variations of its instruction set divided into seven basic groups.

- Data transfer
- Arithmetic and logic
- Bit manipulation
- Loops and jumps
- Strings
- Subroutines and interrupts
- Control

Each instruction consists of a group of bits (1s and 0s) that is decoded by the microprocessor before being executed. These binary code instructions are called *machine language* and are all that the microprocessor recognizes. The first computers were programmed by actually writing instructions in binary code, which was a tedious job and prone to error. This primitive method of programming in binary code has evolved to a higher form where coded instructions are represented by English-like words to form what is known as *assembly language*. This will be discussed further in Section 13–4.