Chapter Outline

1.1. Personal Computer System 5
   1.1.1. PC Hardware 6
   1.1.2. PC Motherboard 7
   1.1.3. PC Memory 9

1.2. Word-Processor Operation 10
   1.2.1. Starting the Computer 10
   1.2.2. Starting the Application 10
   1.2.3. Data Input 11
   1.2.4. Data Storage 11
   1.2.5. Data Processing 11
   1.2.6. Data Output 12

1.3. Microprocessor Systems 13
   1.3.1. System Operation 14
   1.3.2. Program Execution 15
   1.3.3. Execution Cycle 16

1.4. Microcontroller Applications 17
   1.4.1. Microcontroller Application Design 17
   1.4.2. Programming a Microcontroller 23

Questions 1 25
Activities 1 26

Chapter Points

- A microprocessor system consists of data input, storage, processing and output devices, under the control of a CPU.
- The main unit of a desktop PC is a modular system, consisting of the motherboard, power supply and disk drives.
- The motherboard carries the microprocessor (CPU), RAM, BIOS ROM, bus controllers and I/O interfaces.
- The CPU communicates with the main system chips via a shared set of address and data bus lines.
- The microcontroller provides most of the features of a conventional microprocessor system on one chip.
In this chapter, we will start with something familiar, looking at how a personal computer (PC) works when running a word processor, to establish a few technical concepts that are used in microcontrollers (MCUs). Hopefully, most readers will be familiar with this, and will know how the application functions from the user’s point of view. Some basic microcontroller system ideas will be introduced by analyzing how software interacts with computer hardware, allowing the user to enter, store and process documents. For example, we will see why different kinds of memory are needed to support the system operation. If you are familiar with these concepts, you can skip this chapter.

The PC also provides the hardware platform for the PIC® program development system. The programs for the PIC are written using a text editor, and the machine code program is created and downloaded to the PIC chip using the PC. The PIC development system hardware can be seen connected in Figure 1.1. We will see how this works later.

We will also have a quick look at a basic microcontroller system, set up to operate as a simple equivalent of the microprocessor-based PC system, to see how it compares. Here, the microcontroller has a keypad with only 12 keys instead of a keyboard, and a seven-segment display instead of a screen. Its memory is much smaller than the PC, yet it can carry out the same basic tasks. In fact, it is far more versatile; the Intel™ processors used in the PC are designed specifically for that system. The microcontroller can be used in a great variety of circuits, and it is much cheaper.

Figure 1.1
Laptop with PIC demo system attached
1.1. Personal Computer System

The conventional desktop system comprises a main unit, separate keyboard and mouse, and monitor. The main unit has connectors for these (when wireless peripherals are not available) and universal serial bus (USB) ports for memory sticks, printers, scanners, etc., as well as hardwired (Ethernet), or wireless (Wi-Fi) network interfaces. The circuit board (motherboard) in the main unit carries a group of chips which work together to provide digital processing of information and control of input and output devices. A power supply for the motherboard and the peripheral devices is included in the main unit.

The laptop has the same components in a compact form, with integrated keyboard and screen, while tablet computers are even more compact with a touch-sensitive screen and no keyboard. The difference between a microprocessor and microcontroller system is illustrated quite well by comparing a desktop computer with a touch-screen game console or mobile phone. The facilities and applications are similar, they just differ in scale and complexity.

A block diagram (Figure 1.2a) is a good way to show such a system in simplified form, so we can identify the main components and how they connect. In the case of the disk drives and network, for example, the information flow is bidirectional, representing the process of saving data to, and retrieving data from, the hard disk or server. The internal architecture of a microcontroller is shown in its data sheet as a block diagram.

Any microprocessor or microcontroller system must have software to run on the hardware. In a desktop, this is stored on a hard disk inside the main unit; this can hold a large amount of data that is retained when the power is off. There are two main types of software required: the operating system (e.g. Microsoft® Windows) and the application (e.g. Microsoft® Word). As well as the operating system and application software, the hard disk stores the data created by the user, in this case, document files.

The keyboard is used for data input, and the screen displays the resulting document. The mouse provides an additional input device, allowing control operations to be selected from menus or by clicking on icons and buttons. This provides a much more user-friendly interface than earlier computers, which had a command-line interface. Then, actions were initiated by typing a text command such as ‘dir’ to show a directory (folder) of files. Network specialists still use this type of interface as it allows batch files (list of commands) to be created to control system operation. The network interface allows us to download data or applications from a local or remote server, or share resources such as printers over a local area network (LAN) and provide access to a wide area network (WAN), usually the Internet. In the domestic environment, a modem is currently needed to connect to the Internet via a telephone line or cable service. The network browser (e.g. Microsoft® Internet Explorer) is then another essential application.
1.1.1. PC Hardware

Inside the PC main unit (Figure 1.2b), the traditional motherboard has slots for expansion boards and memory modules to be added to the system. The power supply and disk drives are fitted separately into the main unit frame. The keyboard and mouse interface are integrated on the motherboard. In older designs, expansion boards carried interface circuits for the disk drives.
drives and external peripherals such as the display and printer, but these functions are now increasingly incorporated into the motherboard itself. Peripherals are now usually connected via USB or wirelessly.

The desktop PC is a modular system, which allows the hardware to be put together to meet the individual user's requirements, with components sourced from different specialist suppliers, and allows subsystems, such as disk drives and keyboard, to be easily replaced if faulty. This also allows easy upgrading (e.g. fitting extra memory chips) and also makes the PC architecture well suited to industrial applications. In this case, the PC can be ‘ruggedized’ (put into a more robust casing) for use on the factory floor. This modular architecture is one of the reasons for the success of the desktop PC hardware, which has continued in the same basic form for many years, as a universal processor platform. The laptop is the main alternative for the general user, but it is not so flexible, and tends to be replaced rather than upgraded. Another reason for its success is the dominance of Microsoft operating systems, which have developed in conjunction with the Intel-based hardware, providing a standard platform for domestic, commercial and industrial computers.

1.1.2. PC Motherboard

The main features of typical motherboards are shown in Figure 1.3. The heart of the system is the microprocessor, a single chip, or central processing unit (CPU). The CPU controls all the other system components, and must have access to a suitable program in memory before it can do anything useful. The blocks of program required at any one time are provided by both the operating system and the application software, which are downloaded to random access memory (RAM) from the hard disk as required. The programs consist of lists of machine code instructions (binary code) that are executed in sequence by the CPU.

The Intel CPU has undergone rapid and continuous development since the introduction of the PC in the early 1980s. Intel processors are classified as complex instruction set computer (CISC) chips, which means they have a relatively large number of instructions that can be used in a number of different ways. This makes them powerful, but relatively slow compared with processors that have fewer instructions; these are classified as reduced instruction set computer (RISC) chips, of which the PIC microcontroller is an example.

The CPU needs memory and input/output devices for getting data in, storing it and sending it out again. The main memory block is made up of RAM chips, which are generally mounted in Dual In-line Memory Modules (DIMMs). As far as possible, input/output (I/O) interfacing hardware is fitted on the motherboard (keyboard, mouse, USB, etc., preferably wireless), but additional peripheral interfacing boards may be fitted in the expansion card slots to connect the main board to extra disk drives and other specialist peripherals, traditionally using the PCI bus, a parallel data highway 32 bits wide.
All these parts are connected together via a pair of bus controller chips, which handle parallel data transfers between the CPU and the system. The ‘northbridge’ provides fast access to RAM and the graphics (screen) interface, while its partner, the ‘southbridge’, handles slower peripherals such as the disk drives, network and PCI bus. The motherboard itself can be represented as a block diagram (Figure 1.4) to show how the components are interconnected.

The block diagram shows that the CPU is connected to the peripheral interfaces by a set of bus lines. These are groups of connections on the motherboard, which work together to transfer the
data from the inputs, such as the keyboard, to the processor, and from the processor to memory. When the data has been processed and stored, it can be sent to an output peripheral, such as the screen.

Buses connect all the main chips in the system together, but, because they mainly operate as shared connections, can only pass data to or from one peripheral interface or memory location at a time. This arrangement is used because separate connections to all the main chips would need an impossible number of tracks on the motherboard. The disadvantage of bus connection is that it slows down the program execution speed, because all data transfers use the same set of lines, and only one data word can be present on the bus at any one time. To help compensate for this, the bus connections are as wide as possible. For example, a 64-bit bus, operating at 100 MHz ($10^8$ Hz), can transfer 6.4 gigabits ($6.4 \times 10^9$ bits) per second. The current generation of Intel® CPUs also use multiple (typically 4) 64-bit cores in one chip to improve performance.

1.1.3. PC Memory

There are two principal types of memory in the PC system. The main memory block is RAM, where input data is stored before and after processing in the CPU. The operating system and application program are also copied to RAM from disk for execution, because access to data in RAM is faster. Unfortunately, RAM storage is ‘volatile’, which means that the data and application software disappear when the PC is switched off, and these have to be reloaded each time the computer is switched back on.

This means that some read-only memory (ROM), which is non-volatile, is needed to get the system started at switch on. The basic input/output system (BIOS) ROM chip contains enough
code to check the system hardware and load the main operating system software from disk. It also contains some basic hardware control routines so that the keyboard and screen can be used before the main system has been loaded.

The hard disk is a non-volatile, read and write storage device, consisting of a set of metal disks with a magnetic recording surface, read/write heads, motors and control hardware. It provides a large volume of data storage for the operating system, application and user files. The applications are stored on disk and then selected as required for loading into memory; because the disk is a read and write device, user files can be stored, applications added and software updates easily installed. Standard hard disk drives can now hold over 1 TB (1 terabyte = $10^{12}$ bytes) of data.

The PC system quickly becomes ever more elaborate, and this description may well already be out of date in some respects. However, the basic principles of microprocessor system operation are the same as established in the earliest digital computers, and these also apply to microcontrollers, as we will see.

### 1.2. Word-Processor Operation

In order to understand the operation of the PC microprocessor system, we will look at how the word-processor application uses the hardware and software resources. This will help us to understand the same basic processes that occur in microcontrollers.

#### 1.2.1. Starting the Computer

When the PC is switched on, the RAM is empty. The operating system, application software and user files are all stored on the hard disk, so the elements needed to run the word processor must be transferred to RAM for quick access when using the application. The BIOS gets the system started. It checks that the hardware is working properly, loads (copies) the main operating system software (e.g. Windows) from hard disk into RAM, which then takes over. As you will probably have noticed, this all takes some time; this is because of the amount of data transfer required and the relatively slow access to the hard drive.

#### 1.2.2. Starting the Application

Windows displays an initial screen with icons and menus, which allow the application to be selected by clicking on a shortcut. Windows converts this action into an operating system command which runs the executable file (WINWORD.EXE, etc.) stored on disk. The application program is transferred from disk to RAM, or as much of it as will fit in the available memory. The word-processor screen is displayed and a new document file can be created or an existing one loaded by the user from disk.
1.2.3. Data Input

The primary data input is from the keyboard, which consists of a grid of switches that are scanned by a dedicated microcontroller within the keyboard unit. This chip detects when a key has been pressed, and sends a corresponding code to the CPU via a serial data line in the keyboard cable, or wirelessly. The serial data is a sequence of voltage pulses on a single wire, which represent a binary code, each key generating a different code. The keyboard interface converts this serial code to parallel form for transfer to the CPU via the system data bus. It also signals separately to the CPU that a keycode is ready to be read into the CPU, by generating an ‘interrupt’ signal. This serial-to-parallel (or parallel-to-serial) data conversion process is required in all the interfaces that use serial data transfer, such as the keyboard, screen and network (see the appendices for more information on binary coding, and serial and parallel data).

The mouse is a convenient pointer controller for selecting options on screen and drawing graphics. The original mouse used two rollers set at right angles, with perforated disks attached. The holes were detected using an opto-sensor, sending pulses representing movement in two directions to the CPU. This mechanism has been replaced with direct optical sensing of variations in the surface under the mouse, using complex software to extract the direction and speed information. This also eliminates unreliable mechanical components.

Data input from a network or USB source is also in serial form, while the internal disk interface is traditionally in parallel, direct onto the peripheral bus. The parallel connection is inherently faster, since data bits are transferred simultaneously on all bus lines.

1.2.4. Data Storage

The character data is received by the CPU from the keyboard, or other interface, in parallel form, via the internal data bus. It is stored in a CPU register and then copied back to RAM. RAM locations are numbered and accessed via the system address bus, a set of lines that select a location as a binary number. This is why the CPU has so many pins: for speed of transfer, all data and address pins, and control lines, are separately connected to the northbridge controller via the frontside bus, and hence to the RAM. The data is stored in RAM as charge on the gate of an electronic switch, a field effect transistor (FET; see Appendix B). When charged, the FET is switched on, and this state can be read back at a later time. The addressing system accesses an array of these switches in rows and columns to store and retrieve bits of data. Each byte has a unique address.

1.2.5. Data Processing

The data processing in the CPU required by a simple text editor is minimal; the input characters are simply stored as binary code and displayed, with a separate graphics processor
converting the character code to a corresponding symbol on the screen. Nevertheless, the word-processor program has to handle different fonts, word wrapping at the end of lines and so on. It also has to handle text, page and document formatting, menu systems and the user interface. Editing embedded graphics is a bit more complex, since each pixel needs handling separately. The most demanding applications are those where the real world is simulated in a computer model in order to make predictions about the behavior of complex systems. Weather forecasting is an extreme example; the fact that we can still only forecast accurately a few days ahead illustrates the limitations of such system modeling, even on the most powerful computers.

The circuit simulation software used in this book, Proteus VSM, combining traditional circuit analysis with an interactive interface, is a good example of system modeling in a PC. It takes a circuit created as a schematic and applies network analysis (lots of simultaneous equations) to predict its operation when constructed. For digital elements, logic modeling is needed, and then the analogue and digital domains are co-simulated. Component characteristics and input variables are typically represented by 32-bit binary numbers, which correspond to decimal numbers in exponential form (as on a scientific calculator). The processor needs to be able to manipulate these circuit variables simultaneously to represent the circuit conditions at a series of points in time. The output is calculated and displayed via animated circuit components or virtual instruments, or graphically. Numerous examples are to follow!

1.2.6. Data Output

Going back to the word processor, the characters must be displayed on the screen as they are typed in, so the character codes stored in memory are also sent to the screen via the graphics interface. The display is made up of single colored dots (pixels) organized in lines across the screen, which are accessed in sequence, forming a scanned display. The shape of the character on screen must be generated from its code in memory, and sent out on the correct set of lines at the right time. The display must therefore be created as a two-dimensional image made up from a serial data stream which sets the color of each pixel on the screen in turn, line by line, where each line of text occupies a set of adjacent lines. The exact arrangement depends on the font type and size.

If a file is transferred on a network, it must also be sent in serial form. The characters (letters) in a text file are normally sent as ASCII code, along with formatting information and network control codes. ASCII code represents one character as one byte (8 bits) of binary code, and is therefore a very compact form of the data. The code for the letter ‘A’ (upper case), for example, is 01000001.

The printer works in a similar way to the screen, except that the output is generated as lines of dots of ink on a page. In an inkjet printer, you can see the scanning operation