Computers and computer systems

About this free course

This free course provides a sample of level 2 study in Computing & IT
http://www.open.ac.uk/courses/find/computing-and-it

This version of the content may include video, images and interactive content that may not be optimised for your device.

You can experience this free course as it was originally designed on OpenLearn, the home of free learning from The Open University:
http://www.open.edu/openlearn/science-maths-technology/computing-and-ict/computing/computers-and-computer-systems/content-section-0.

There you'll also be able to track your progress via your activity record, which you can use to demonstrate your learning.

The Open University

Walton Hall, Milton Keynes

MK7 6AA

Head of Intellectual Property, The Open University

The Open University

United Kingdom by Hobbs the Printers Limited, Brunei Road, Totton, Hampshire SO40 3WX

Contents

Introduction

There is more to computers and processors than simply PCs. In fact computers are ubiquitous in everyday life. This unit challenges how we view computers through the examples of processors in kitchen scales and digital cameras, as well as a work of art that, at heart, is a computer.

This OpenLearn course provides a sample of level 2 study in Computing & IT.

Tell us what you think! We'd love to hear from you to help us improve our free learning offering through OpenLearn by filling out this short survey.

Learning Outcomes

After studying this course, you should be able to:

- understand what all the terms highlighted in bold in the text mean
- understand the fundamental hardware components that make up a computer’s hardware and the role of each of these components
- understand the difference between an operating system and an application program, and what each is used for in a computer
- describe some examples of computers and state the effect that the use of computer technology has had on some common products
- identify the principal components of a given computer system and draw a diagram after the style of Figures 6 and 12 to represent the data flows between them.

1 Computers and processors: introduction

Computers have become a vital part of everyday life. It is almost inconceivable that you could spend a day without at least one event being influenced by a computer. Perhaps the word 'computer' automatically conjures up the image of a personal computer sitting on a desk, but in fact it is the computers you cannot see that influence your life the most. Typical examples of common products that may use these 'invisible' computers are:

- cars
- washing machines
- bar-code reading systems
- DVD players/writers
- central-heating controllers
- microwave ovens
- games consoles.

This is a very short extract from a very long list, but even this limited set of examples shows how significant the use of computers has become. Without computers many everyday products such as mobile phones and personal digital assistants (PDAs) would not exist, dramatic progress in the development of products such as artificial limbs could not have happened, and you would not have the luxury of many conveniences now taken for granted, such as email.

The computers which form the basis of those used today were mainly developed in the 1940s. The following quote taken from that era shows how difficult it was to conceive of the way in which computers would develop in the following decades.

> I think there is a world market for maybe five computers.
>
> (Thomas Watson, Chairman of IBM, 1943)

Even later on, in the mid 1970s, some still failed to comprehend the size of the future computer market.

> There is no reason for any individual to have a computer in his home.
>
> (Ken Olsen, President of Digital Equipment Corporation, 1977)

And although a diminution in size was anticipated, it was considerably underestimated.

> Where a calculator on the ENIAC is equipped with 18 000 vacuum tubes and weighs 30 tons, computers in the future may have only 1000 vacuum tubes and perhaps weigh 1½ tons.
>
> (Popular Mechanics, March 1949)

Figure 1 shows a picture of the ENIAC computer mentioned above. You can see it is rather larger than the personal computer available today! Completed in the US in 1945, it was one of the earliest electronic computers. Its name stands for Electronic Numerical Integrator And Calculator, and it was designed to calculate ballistic firing tables in the Second World War. It could perform mathematical operations such as addition, subtraction, multiplication and division, and it could find the square root and compare two values for equality.

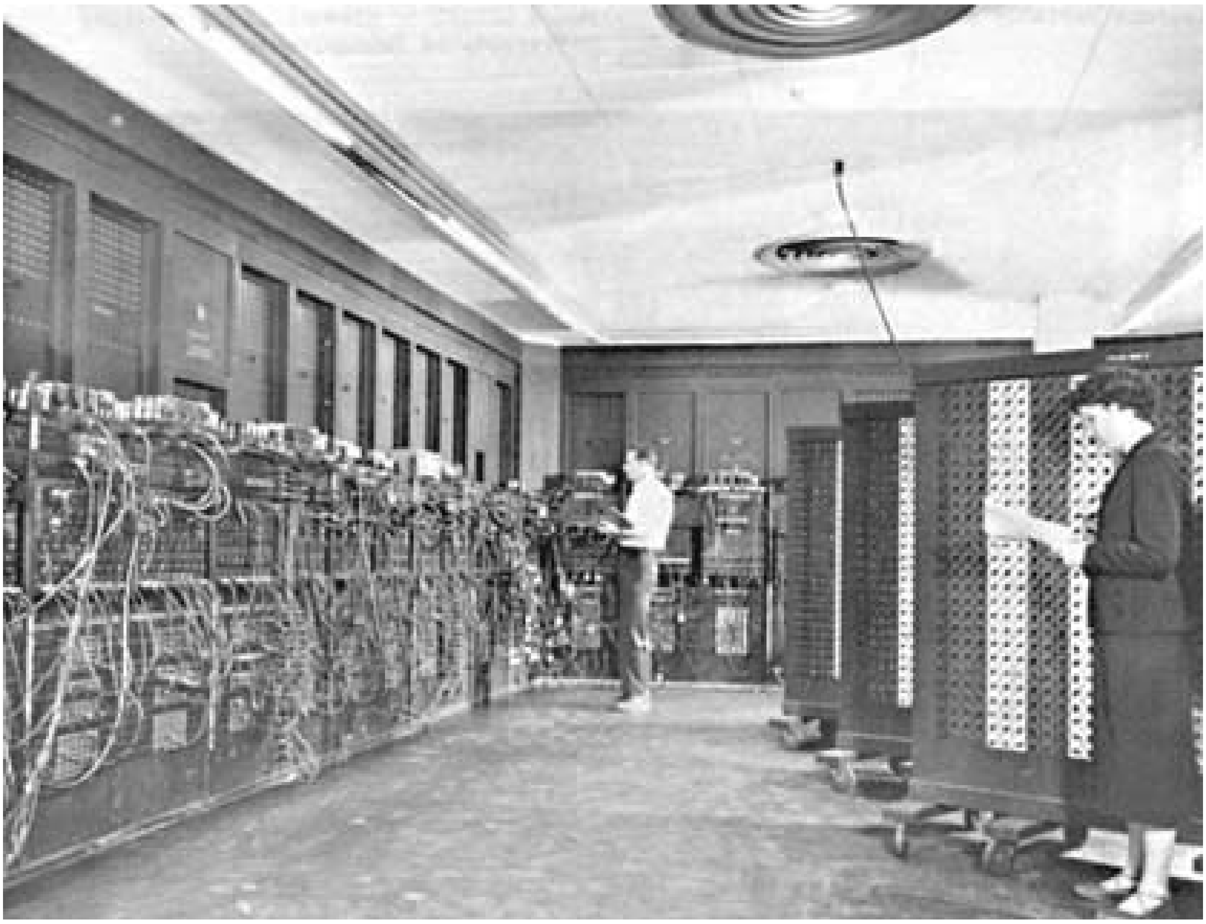

Figure 1 The ENIAC computer

As part of the ENIAC fiftieth anniversary celebrations, students and staff at the University of Pennsylvania fabricated a version of this computer using modern manufacturing processes. The component at the heart of this later version measures 7.44 millimetres by 5.29 millimetres! A personal computer was connected to this component to provide the modern equivalent of the cable connections shown on the left hand side of Figure 1 and display the ENIAC's outputs.

Computers like ENIAC were built because of the need for powerful automatic calculators. Another famous use for an early computer was the work at Bletchley Park in the UK to break the German diplomatic codes used in the Second World War.

A **computer** is a machine that manipulates data following a list of instructions that have been programmed into it. **Input devices** are used to input data into a computer; the keyboard of a personal computer, the scanner of a bar-code reading system and the switches or buttons of a microwave oven are some examples. The list of instructions the computer follows is called a **computer program**. So, for example, a bar-code reader that sends the name of a scanned bar-coded product to a display will have been programmed with a set of instructions that makes it:

- take in data via the bar-code scanner;
- use the data from the bar-code scanner to look up the name of the product in a list that cross references bar-code data to product name;
- generate a new form of data that is compatible with showing the product name on the display;

- send the newly generated data to the display to show the product name.

In this example the display is being used as an **output device**. There are many different types of output device. The actuator that switches on a pump of a computer-controlled central-heating system is one example; the sound system that generates the beep of an electronic heart monitor is another.

I have used the word ‘data’ several times now in the context of the computer receiving input data, generating data and outputting data. A computer can only work with information that is presented to it in a very strictly controlled format. When information is in this format it is called data. Quite simply, a computer cannot perform its task if the information it needs has not been transformed into the required data form. You will find out more about the format of data a computer needs later on in this block.

As computers have developed, a critical change in their role has been their use in communication: many of the applications that run on personal computers (PCs) help us communicate with each other, and also with other computers. The following quote from Danny Hillis, a pioneer in the development of some of the fastest and most powerful computers, gives some insight into how computers can turn everyday objects into part of a communication system.

> I went to my first computer conference at the New York Hilton about 20 years ago. When somebody there predicted the market for microprocessors [these are the major component of all computers] would eventually be in the millions, someone else said, ‘Where are they all going to go? It's not like you need a computer in every doorknob!’ Years later, I went back to the same hotel. I noticed the room keys had been replaced by electronic cards you slide into slots in the doors.
>
> There was a computer in every doorknob.
>
> (Danny Hillis, circa 1999)

Of course you do not know exactly the configuration of the computers in the doorknobs of the Hilton; it could be that they simply verified that the card should give the holder access to that particular room. Alternatively the doorknob computer could communicate with another computer, telling it that the occupant had just entered the room. This second computer could then ring the telephone to pass on any recorded messages, activate a display showing if the occupant had received any e-mail or perhaps run the bath!

As you study this course you will find out how computers and the components within them carry out their allotted tasks, and you will also develop an understanding of how improvements in computer technologies have allowed computers to become smaller, more powerful and cheaper.

2 Computers and computer systems

2.1 Processors

Figure 2 shows an advertisement for a personal computer (PC) published in February 2004. The main features of the computer are listed in this advert. The first item on the list is ‘Intel® Pentium® 4 Processor 2.80 GHz’. So this computer uses an Intel Pentium 4 Processor, running at a speed of 2.80 GHz. A **processor** is an essential component of a computer; it carries out or **executes** the instructions that make up the computer program. PCs use one main processor and several other ‘supporting’ processors, and adverts for PCs often specify what main processor they use. The speed of the processor (2.80 GHz in this instance) is a measure of how fast the processor can carry out each instruction. (Don't worry if you don't understand the term ‘GHz’ and other specialised terms used in the advert such as ‘512 MB DDR RAM’. These will be explained as the unit progresses.)

Figure 2 An advertisement for a personal computer

You may remember that the quote from Danny Hillis in Section 1 mentioned a microprocessor. The term **microprocessor** was introduced when processors were first made on a single silicon chip, with the prefix ‘micro’ emphasising their small size. Today, however, the fact that a processor can be made on a single silicon chip is taken for granted and the term ‘microprocessor’ is not so often used. This course will generally use the term ‘processor’.

All computers, not just PCs, contain processors, so all those ‘invisible’ computers I listed earlier will contain a processor. However, the processor they use will not necessarily be the same as that used in a PC. For example, the processor used within a central-heating controller would not be the same as the main processor used in the personal computer you are using to study this course. The processor in the personal computer has to carry out a much more complex set of tasks and execute its instructions much more quickly than the processor in the central heating controller. As a result the PC's processor is likely to be physically larger and more costly. However you will see later in the course that the complexity and speed of operation of processors has increased dramatically. As a result, the ‘simple’ processor in an electronic central heating controller may be very similar to a processor which was considered ‘state of the art’ a decade or two previously.

2.2 Memory

You should now be beginning to build up a picture of what a computer is: you know it needs input and output devices to communicate with the world outside and a processor to carry out the instructions that are programmed into it. But where are these instructions stored within the computer? The answer is that they are stored within what is called the computer's **main memory**, along with any data needed to carry them out.

However, the main memory in computers like PCs is much too small to hold all of the programs and associated data that their users need. In addition, main memory does not hold onto its contents when the computer is switched off. So users must be able to call up the programs they want, and also store and read back the files they have generated with these programs, from some form of capacious and retentive memory. This memory is called **secondary memory**, and there are two types, removable and permanent. With removable secondary memory the user can store files and then ‘remove’ them from the PC, either to ensure there are copies if the computer fails, or to transport the files to another PC. New software can also be installed from removable secondary memory. Removable secondary memory includes floppy disks, CD-ROMs, memory cards and DVDs. In contrast, permanent secondary memory is ‘permanently’ attached to the PC and is usually only removed if the PC is undergoing some maintenance or repair. A typical example of permanent secondary memory is a computer's **hard disk** – so called because it consists of one or more rigid magnetic disks rotating about a central axle. It is common practice to copy the files stored on permanent secondary memory onto some removable secondary memory as a backup in case of disk failure.

Although programs and associated data are stored on the hard disk when not actually in use, both the programs and the data must be copied into the computer's main memory before the processor can execute the instructions or use the data.

The third and fourth items in the list of features of the PC in Figure 2 show that this PC has 512 MB of main memory and a hard drive that provides 80 GB of secondary memory. Note that I used the term ‘hard disk’, but in the advert the term ‘hard drive’ is used to refer to the permanent secondary memory. These terms are often used synonymously, though in fact there is a subtle difference which I'll explain shortly.

Activity 1 (Exploratory)

What other secondary memory device or devices are used by the PC in the advert shown in Figure 2?

Item 7 on the list of features shows that the PC also uses DVDs and CDs as removable secondary memory.

All computers have main memory, but not all will have secondary memory. In an 'invisible' computer such as the central-heating controller, the software is already stored in the main memory when the computer is purchased. The software is said to be already installed. The PC you are using on this course will have come with some software already installed on it – the software the PC needs to start up when you switch it on. But a key difference between an 'invisible' computer like the one in the central-heating controller and the PC is that users cannot install any additional software on the 'invisible' computer, whereas they can and do install their own choice of software onto a PC. They do this by copying computer programs into the secondary memory. Such programs are then taken into the main memory when the program is run.

2.3 Computer systems

So far, I have introduced the major components of a computer, namely a processor along with input and output devices, plus main and secondary memory. I now want to explore three of these components a little further, starting with input devices.

Input devices have to collect some information from outside the computer and present it to the computer as data which is in a form the processor can work with. (Strictly speaking, 'data' is the plural of the Latin word 'datum'. But in the world of computers 'data' is very often used in the singular, and this unit follows that practice.) It is useful to think of these as two separate functions:

- 'capturing' the information;
- 'translating' the information into a data form the processor can use.

Sometimes these two functions are done by a single physical entity; sometimes by two separate entities. When I am talking about input functions I shall use the term you have already met, 'input device', for whatever *captures* the information and the term **input subsystem** for whatever does the *translation*.

Similarly, output devices have to collect data from the processor in the processor's format and translate it into something that is meaningful outside the computer, and again it's useful to think of these as two separate functions:

- 'translating' from the data form the processor uses;
- 'presenting' the information.

Again, when I am talking about output functions I shall use the term 'output device' for whatever *presents* the information and the term **output subsystem** for whatever does the *translation*.

In the case of secondary memory, there are also two functions, though they are rather different. The secondary memory's function is simply to. *hold* stored data, and a **secondary memory subsystem** is used to *prepare* the data for storage and get it stored (when data is being sent to the memory) or to collect stored data and prepare that data for use by the processor (when data is being sent from the memory). Here is the subtle difference between ‘hard disk’ and ‘hard drive’ that I mentioned earlier: the hard disk is the secondary memory and the hard drive is the secondary memory subsystem. But remember that many people use the two synonymously (and hence ambiguously). Often the subtle distinction doesn't matter in a particular context, but it's worth being alert to the fact that the two terms are not strictly synonymous.

The diagram in Figure 3 shows all the functional blocks of a computer. That is, it shows all the *functions* performed within a computer. Some of the components in any particular example of a computer may perform more than one of the functions shown in Figure 3, so there may not always be a separate physical entity associated with each function shown in the diagram. For example, sometimes an output device and its associated output subsystem are housed together; in some small computers the processor and the main memory are even housed together. But, with the possible exception of secondary memory, any computer will have the functionality shown in Figure 3.

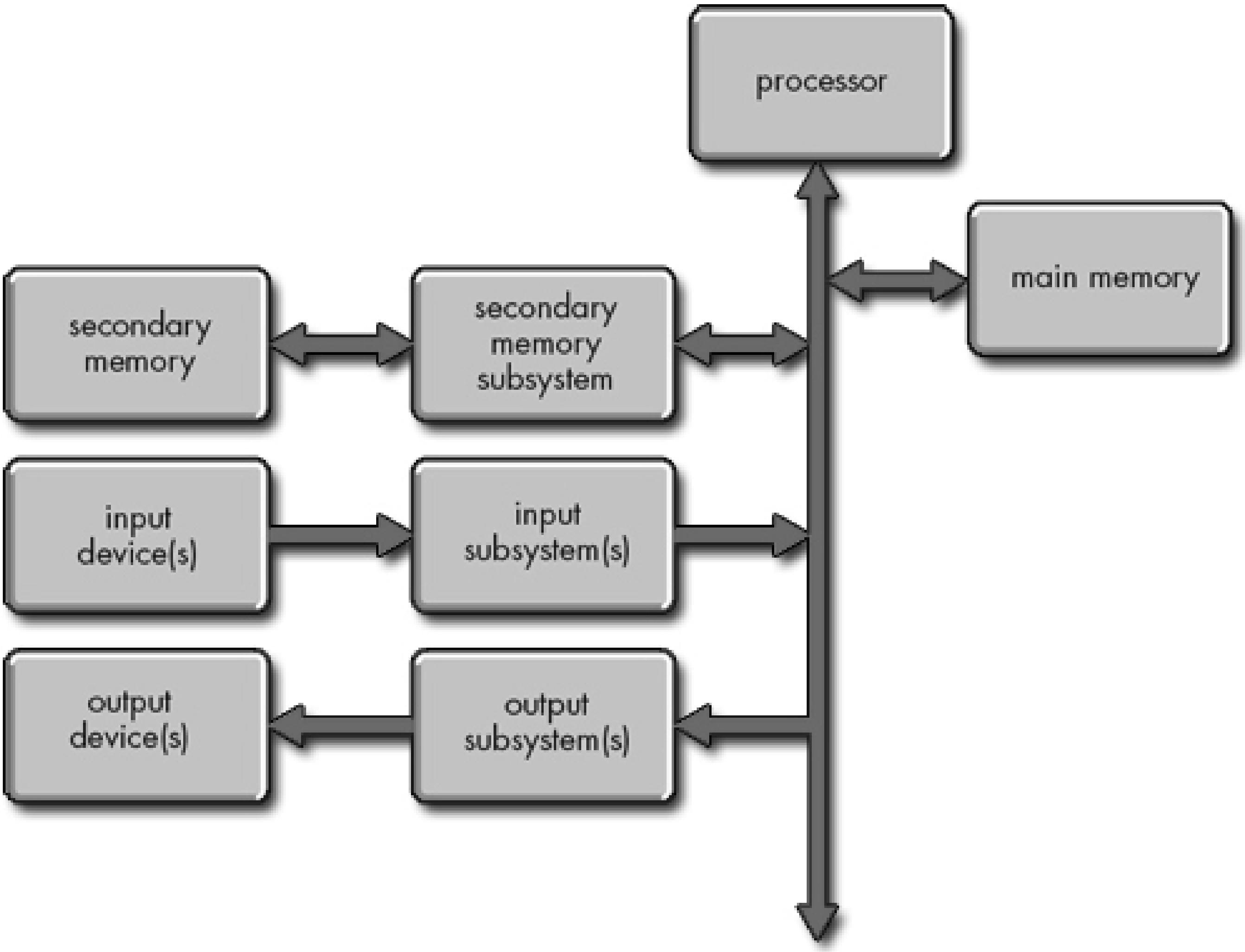

Figure 3 A functional block diagram of a computer which also shows the flow of data within the computer

In Figure 3 the interconnecting lines show the data flows. The thick line running vertically down the page from the processor represents the **computer bus**. This is a data path that connects the input and output subsystems and the secondary memory subsystem to the computer's processor and main memory. It allows data to be transmitted from one part of the computer to another. Notice that this path has an arrow at each end, indicating that data can travel in both directions along it. The arrows on the other paths indicate that data can also travel both to and from the secondary memory, but data only travels in from the input devices and out to the output devices.

Figure 3 is an important diagram, and you will see it, and variations of it, again later in the unit. Figure 3 illustrates the fact that computers, however complex and 'clever' they may seem, do only the following tasks:

- *receive* data from the outside world via their input devices;
- *store* that data in their memory;
- *manipulate* that data in their processor, probably creating and storing more data while doing so;
- *present* data back to the outside world via their output devices.

2.4 Computer systems (*contd*)

As I have already mentioned, the functional blocks shown in Figure 3 relate very closely to, even though they are not necessarily identical with, the computer's physical components. The computer's physical components are normally known collectively as the **hardware**. **Software** is a term often used to refer to a computer program or a collection of computer programs which enable a computer to carry out its tasks. As the course progresses you will find out a great deal about computer hardware and software, including the processor and the programs it runs. You will generate some programs yourself and look at the operation of the processor to see how it executes the instructions within them. Through this you will gain an understanding of what form the data used by the processor and memory must take, and hence understand the role of the input and output subsystems.

Box 1: Electrical signals and computers

The data that travels along the main computer bus does so in the form of electrical signals which can have one of two possible values: a near-zero voltage, known as 'voltage low', and a rather higher voltage, probably around 3 volts, known as 'voltage high'. (3 volts is used to represent a '1' that is most common at the time of writing [early 2004]. A couple of decades ago the value was usually nearer to 5 volts, and older systems still use this value. Some newer systems are already using voltages near 2 volts and this may well become the most common value in a few years.) Any electrical signal where the number of possible values that can be used is limited is known as a **digital** signal. If the number of possible values is limited to just two, as on the computer bus, then the signal is known as a binary digital signal, or simply a **binary** signal. (The 'bi' in 'binary' means 'two'.)

So the electrical signals that travel along the computer bus, and hence to the output subsystems and from the input subsystems, are all binary signals. Unfortunately, the electrical signals produced by input devices or needed by output devices are not necessarily binary, or even digital. Hence an important task of the input and output subsystems is to transform between the binary signals on the computer bus and whatever signals are used by their particular input or output device.

Activity 2 (Self assessment)

For the PC shown in Figure 2, write down which of the functional blocks in Figure 3 the following items of hardware relate to (for simplicity, items that provide input functionality can be assumed to relate to input devices, rather than input subsystems, and similarly for items that provide output functionality):

Keyboard, monitor, 80 GB hard drive, 512 MB DDR RAM, speakers, mouse, DVD and CD.

Answer

The keyboard and mouse relate to input devices. The monitor and speakers relate to output devices.

To decide whether the 80 GB hard drive relates to the secondary memory, to the secondary memory subsystem or to the combination of both you need to make an intelligent guess about what those who wrote the advert meant. In this case I suspect they probably meant the combination of the two.

The 512 MB DDR RAM relates to main memory.

The DVD and the CD both relate to secondary memory.

Activity 3 (Exploratory)

The following items are listed under the features of the PC shown in Figure 2: a sound card and a modem. What problem arises if you try to relate these two items to the functional blocks of Figure 3?

The problem is that the sound card is used for *both* input and output (it takes inputs from a microphone and delivers outputs to one or more speakers). The modem is also used for both input and output (it both sends signals to and receives signals from the telephone network). Therefore these items do not at first seem to fit neatly with Figure 3. But remember that the diagram is showing *functionality*. So you can relate the input functionality of the sound card to the ‘input subsystem’ item in Figure 3 (the microphone would be the input device), and the output functionality of the sound card to the ‘output subsystem’ (the speaker or speakers would be the output device). In the case of the modem its input functionality relates to the ‘input device’ in Figure 3, and its output functionality to the ‘output device’.

The terms **input-output device** and **input-output subsystem** are sometimes used where items have both input and output functionality. Hence a sound card is an input-output subsystem.

Finally, just as you are just becoming familiar with all of the terms I have been introducing, I need to add a word of caution. When you read books or other literature about computers you may find some of the terms I have defined used differently. This is not necessarily a problem, and is common when technical terms become part of everyday language. However, throughout your study of this unit you do need to make sure that you use the terms as defined here.

One term I have not used here that you might come across is **computer system**. Historically some people used the terms ‘computer’ and computer system ‘computer system’ rather differently. But that is no longer the case, and nowadays the word ‘system’ tends to be omitted. A good example is the use of the term ‘personal computer’, which would several years ago have often been described as a ‘personal computer system’.

3 Some facts about processors

3.1 Processor statistics

In Sections 3.1 and 3.2 you are going to find out a little more about one of the key components of a computer: the processor, which manipulates data according to a list of instructions called a program.

Here is a mini-quiz which explores some facts about processors.

Question 1

Which of the numbers given below is closest to the number of processors sold worldwide in 2000?

A	20 million	B	40 million
C	125 million	D	1 billion

The answer to Question 1 is D. The processor market is vast; it is estimated that around 1 billion processors were sold in 2000.

Question 2

Which of the numbers given below is closest to the number of processors installed in a BMW 7-series car in 1999?

A	1	B	5
C	100	D	5000

The correct answer is C, 100 processors. The BMW 7-series manufactured in 1999 uses 65 processors. Not all these are used in the engine management system; for example the processor in the automatic transmission communicates with the processors behind each side view mirror, so they tilt down and inward whenever the driver puts the car into reverse gear. Also the processor in the car radio communicates with the processors controlling the brakes so the audio volume can be adjusted to compensate for additional road noise resulting from the application of the brakes.

Question 3

Which of the numbers given below is closest to the number of processors you own?

A	1	B	4
C	50	D	1000

Perhaps your answer to this will vary from mine, which is C, 50 processors. If you answered A, just one processor, you may have been thinking about the PC you are using to study this course. However, in addition to their main processor PCs contain at least another seven processors. You also need to consider the domestic products you may own. There are processors in some toasters, washing machines, home entertainment systems, tumble dryers, central-heating controllers, video and DVD players, microwave ovens, electronic clocks, TVs, children's toys, computer games, phones, satellite systems and so on, and of course remember the ones you may have in your car. I quite quickly identified around 30 processors that I own, and if I thought about it for longer I would probably find a few more, but it was obvious that the total would be much less than answer D, 1000 processors.

Perhaps you got all the answers to the quiz correct; perhaps all your answers were wrong. It doesn't matter. What is important is that you now appreciate:

- the huge number of applications that can use processors and hence how vast the processor market is;
- that the market for processors is not limited to personal computers;
- and that the market for processors used in personal computers is very much smaller than that for processors used in other applications.

3.2 What does a processor look like?

So what do these devices that are manufactured in such vast quantities look like? Processors are manufactured as integrated circuits. Essentially they are circuits, around the size of a fingernail, which contain many millions of electronic components manufactured as one very complex circuit. Figure 4(a) shows how a processor manufactured as an integrated circuit is packaged so it can be used as a component in an electronic circuit. The pins of the package are connected to the integrated circuit using gold bonding wire. (Gold is most commonly used, but sometimes aluminium is used instead.) Some of the pins are used to supply the electrical power to the device, while signals are input to and output from the processor via other pins. Figure 4(c) shows the integrated circuit I mentioned in Section 1, the one developed as part of the ENIAC fiftieth anniversary project.

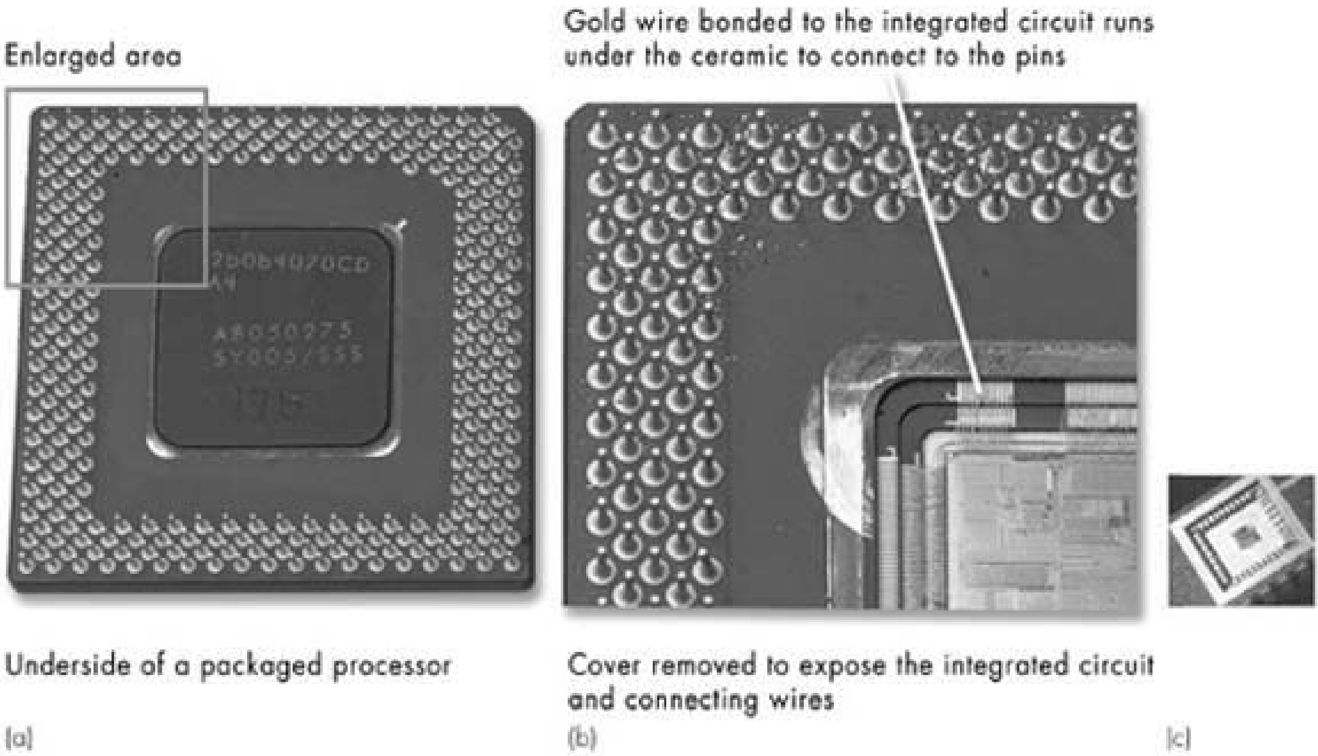

Figure 4 (a) The inside of a packaged processor; (b) detail of the integrated circuit and connecting wires; (c) the integrated circuit created as part of the ENIAC fiftieth anniversary project

Figure 5 shows the packaged processor assembled with other components on the motherboard of a computer. A motherboard is the major circuit board inside a computer and it holds the processor, the computer bus, the main memory and many other vital components.

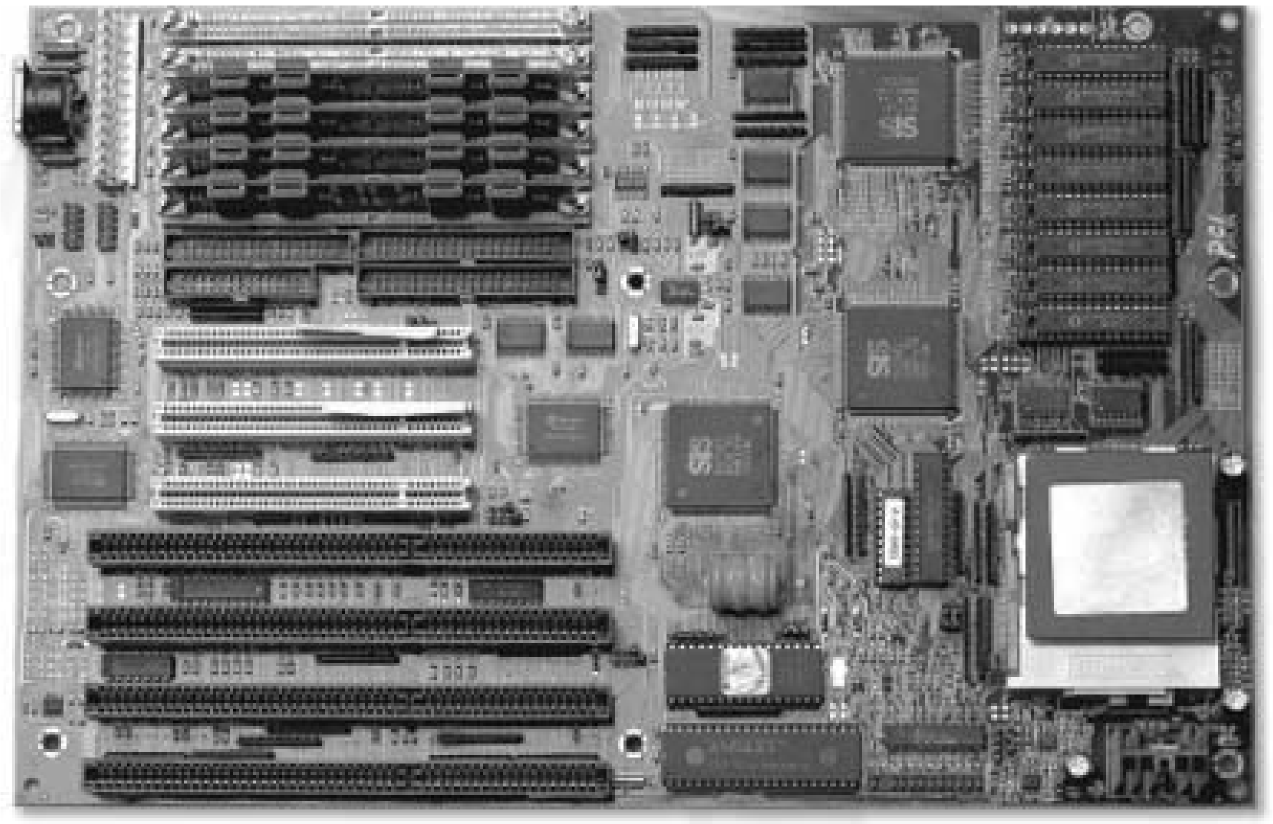

Figure 5 A processor assembled on a motherboard along with other circuit components

4 Representing data and instructions inside a computer

4.1 Switches

I have just indicated that a processor is made up of millions of electronic components manufactured as one very complex circuit. The majority of these components act as switches that can exist in one of only two states, either on or off. The states of certain switches tell the processor what instructions to carry out. Also when a processor is running a program it is altering the state of other switches, switching them on and off many, many times a second.

To represent more easily what is happening to the states of these switches, the ‘off state is often referred to as 0, and the ‘on’ state as 1.

Imagine eight switches in the following states:

on	off	off	off	on	on	on	off

The states of these switches can be written down concisely as the 8-digit code 10001110, where the digit on the extreme left represents the state of the leftmost switch and so on through to the digit on the extreme right representing the state of the rightmost switch.

If, for example, the state of these switches at any time represented an instruction for a processor to execute, then 10001110 would cause one particular instruction to be executed and 10100001 another. (These instructions can also be represented in shorthand, so a list of instructions doesn't have to be tediously written down as many 1s and 0s.)

The code 10001110 is made up of 8 digits. In computing terminology, because each digit can only take one of two values (either 1 or 0), each digit is referred to as a 'binary digit'. This is almost always abbreviated to **bit**. Therefore I can say 10001110 is an 8-bit code. As the code is in binary it is termed a **binary code**, so 10001110 is an 8-bit binary code.

Three switches in the following states would represent the 3-bit binary code 100:

on	off	off

Activity 4 (Exploratory)

Write down as many 2-bit binary codes as you can think of.

There are four possible 2-bit binary codes: 00, 01, 10 and 11.

This representation using 1s and 0s is very convenient. It makes it possible to write down what conditions exist inside the processor without having to deal with the complexities of the voltages and currents that exist to make the switches enter their on and off states. (If you could peer inside a processor you would not see 1s and 0s written down!)

Using binary codes is a very easy way to describe the state of the switches inside the processor, and allows people to represent what the electronic circuits that make up the processor are doing without having to understand how such circuits operate.

4.2 Representing data

But if all the data and computer instructions within a computer are represented by 1s and 0s, how can this limited set of conditions be used to represent, for instance, every letter of the alphabet that might be typed into a computer from a keyboard? Activity 4 showed that there are four possible combinations of 1s and 0s in a 2-bit binary code. So if you had only two bits available you could only represent four different letters, e.g. 'a' could be represented by 00, 'b' by 01, 'c' by 10 and 'd' by 11. This shows that a 2-bit binary code can only represent four items of data.

Activity 5 (Self assessment)

Write down all the possible combinations of a 3-bit binary code and state how many items of data three bits can represent.

Answer

The possible combinations of a 3-bit binary code are 000, 001, 010, 011, 100, 101, 110, 111. Hence the three bits can represent 8 items of data.

Interestingly, a pattern can be deduced for the relationship between the number of bits and the number of items they represent.

2 bits can represent $2 \times 2 = 2^2 = 4$ items

3 bits can represent $2 \times 2 \times 2 = 2^3 = 8$ items and following on with the same pattern

4 bits can represent $2 \times 2 \times 2 \times 2 = 2^4 = 16$ items.

Study note: If you are unsure of the use of the mathematical notation 2^2, 2^3 etc., you will find it helpful to refer to the Numeracy Resource below.

Click on the 'View document' link below to open the 'Numeracy resource'.

View document

A group of eight bits is called a **byte**, so an 8-bit binary code is 1 byte long, a 16-bit binary code is 2 bytes long and a 64-bit binary code is 8 bytes long.

Activity 6 (Self assessment)

1 How many bits are there in 4 bytes?

2 How many items of data can be represented by 1 byte?

Answer

1 Four bytes contain $4 \times 8 = 32$ bits.

2 Since one byte contains 8 bits, the number of items that can be represented by one byte is $2^8 = 256$. (Note that if you had to work out the number of items that eight bits could represent by writing down all possible combinations of 8 bits it would be very tedious and there would be a strong possibility of making an error. Using the calculation $2^8 = 256$ is much easier way of finding the answer, as would be $2^{16} = 65\,536$ for the number of combinations of a 16-bit binary code.)

In general, computers that perform more complex tasks at higher speeds use a larger number of bits to represent their data and instructions. The very simple central heating controller, which only has to do a limited amount of processing, may use an 8-bit representation. More powerful computers will use 16-, 32- or 64-bit representations.

When a computer is running a program a lot of data is being passed around the various elements within the system. The data received by the input subsystem(s) must be passed to the processor in a form it can use, and the processor in turn must present data to the output subsystem(s) in the required format. Even more fundamentally, the processor must be able to recognise each instruction within the program and execute it.

5 Examples of computers

5.1 The personal computer

Over the following screens you will look at three different examples of computers: a PC, which is obviously a computer, and a set of electronic kitchen scales and a digital camera, which are not so obviously computers. You will find that all three of these examples match with the functional block diagram of a computer given in Figure 3 in Section 2.3, although the tasks they have to perform mean that the individual components which perform the functions of the blocks within the diagram are quite different.

In Activities 2 and 3 you looked at how the components of the PC in Figure 2 could be related to the functional block diagram in Figure 3. Figure 6, which is a functional block diagram for the PC and shows the data flow between the components of the PC in Figure 2, should remind you of the outcome. Notice that in this diagram I have put in specific input and output devices and specific items of secondary memory instead of the generic items of Figure 3. In other words, Figure 3 is a *generic* diagram for any computer; Figure 6 is its *specific* form for the PC of Figure 2.

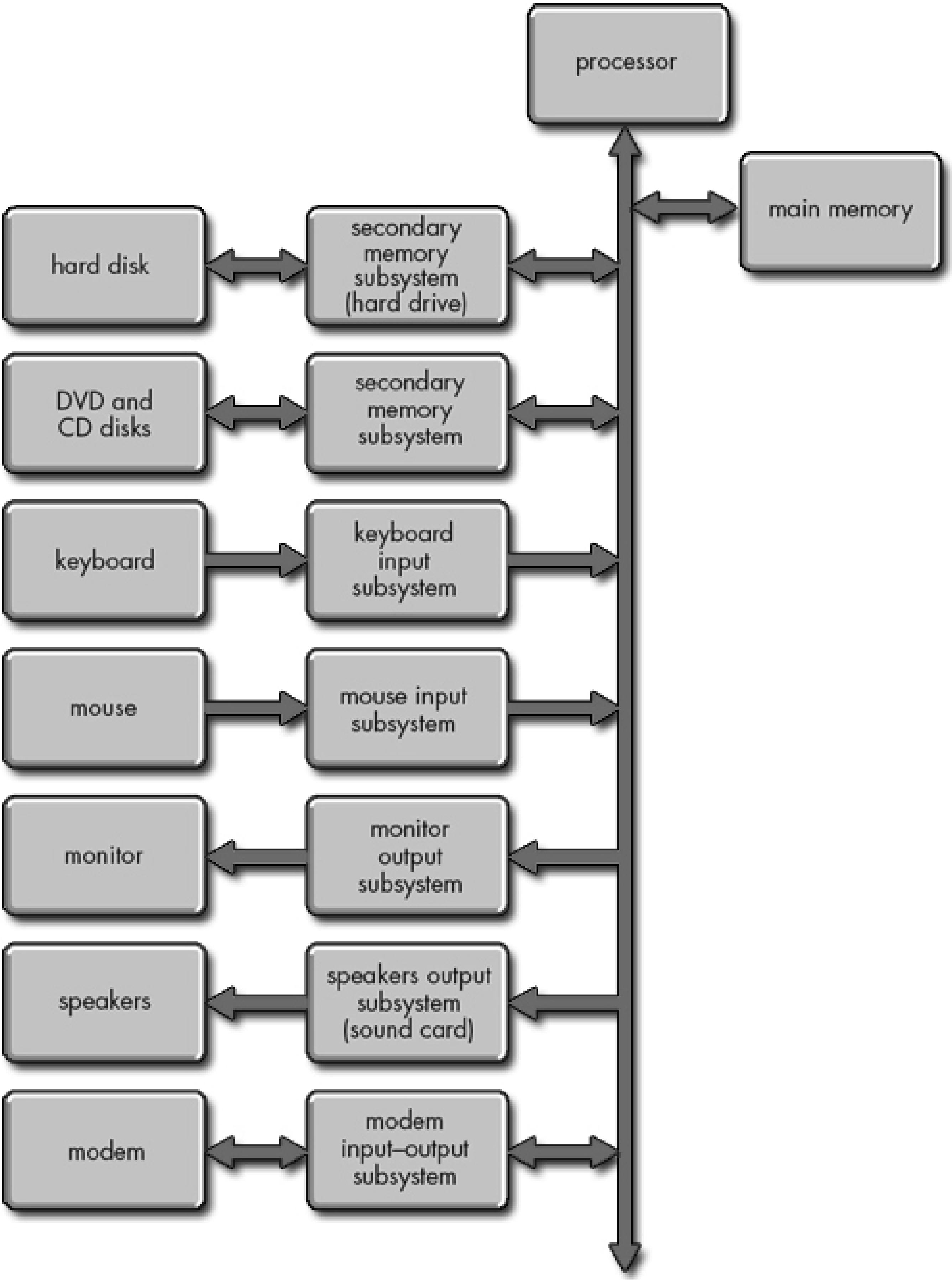

Figure 6 Functional block diagram for the PC in Figure 2

Other PCs may have some additional input devices such as a web cam, plus some additional output devices such as a printer.

The PC is a general-purpose computer. It can run different software programs at the user's request, and hence can be used for a variety of different applications. Typical examples are word processing, sending and receiving email, playing games, browsing the web, and sound and image recording and playback.

The following quote from the book *A Shortcut through Time, The Path to Quantum Computing* by George Johnson shows that even those long familiar with the concepts of how PCs work can still find them fascinating. (A register is a part of a processor and the term 'disk drive' is often used to describe either a floppy disk or hard disk.)

> With a modern PC we blithely double-click an icon on the desktop summoning a flow of data from the disk drive – the pattern of bits that configures thousands of little switches to act as a word processor or a web browser or an MP3 player – temporary little structures, virtual machines. They are allowed to exist only as long as they are needed. Then they are wiped away and replaced with other structures, all built from 1s and 0s.
>
> It is hard to believe sometimes how well this works. You can call up a movie trailer in a window and drag the image around the desktop, causing millions of bits to pour through the computer's hidden registers. It is overwhelming to try and imagine the precise coordination going on behind the screen. Ultimately though it all comes down to shuffling 1 s and 0s, flipping little switches on and off.
>
> (George Johnson, A Shortcut through Time, The Path to Quantum Computing)

Earlier you looked at how data can be represented by bits – two bits can represent four items, three bits eight items, four bits sixteen items, etc. This is fine if, for example, you want to represent a clearly defined set of data such as the letters of the alphabet and numbers. But how are the images of the 'movie trailer' in the above quote represented in your computer? How can you turn your image into items that can be represented by bits? As you will see from the next two examples, electronic kitchen scales and a digital camera, this issue of data representation exists in all computers.

5.2 Electronic kitchen scales

A set of electronic kitchen scales is shown in Figure 7. Their basic operation is relatively simple. When they are switched on and, for example, a 500-gram object is placed in the scalepan, the display shows the digits 500 and the letter g.

Figure 7 Set of scales showing a reading of 500 g

It might be possible to think of these electronic kitchen scales as a computer, in the sense that that they have hardware in the form of a processor, memory, input devices and subsystems, output devices and subsystems; and they have software in the form of programs. But they are not normally thought of in such terms because the fact that they are a computer is not of primary concern to the user – it seems more natural to think of them as kitchen scales. The term **embedded computer** is sometimes used when a computer is 'hidden' in this way. Objects like the kitchen scales are said to 'contain' a computer, rather than to 'be' a computer – the computer is thought of as being 'embedded' in the object.

For the electronic kitchen scales, a key input device is a sensor placed beneath the scalepan. This sensor measures how far the scalepan moves when an object is placed on it, and generates a signal to represent this change of position. The sensor's subsystem then converts this signal into binary coded data (that is, a pattern of 1s and 0s) that the processor can read and manipulate.

The seven-segment display on the scales is an output device. Its subsystem takes binary coded data from the processor and manipulates it into another binary form that will make the correct digits and letter appear on the display.

The processor in this system is performing a very simple task. Whenever the scales are switched on the program installed in the computer's main memory at manufacture runs. This program first instructs the processor to pick up the data placed on the computer bus by the input subsystem. The program then tells the processor to use this data, plus some data stored in main memory, to generate some further data. This new data is then taken from the bus by the output subsystem.

The computer is essentially taking a signal in one format from the sensor and translating it into another format which enables the display to show the correct digits and letter. It is important that it does this in a time-span acceptable to the user.

Just as Figure 6 is the specific form of the generic diagram in Figure 3 for the PC, so it is possible to create the specific form of Figure 3 for these electronic kitchen scales.

Activity 7 (Self assessment)

Using the information about the scales given above, create the specific form of Figure 3 for the kitchen scales. Note that these scales have no secondary memory.

Answer

The diagram is given in Figure 8, below.

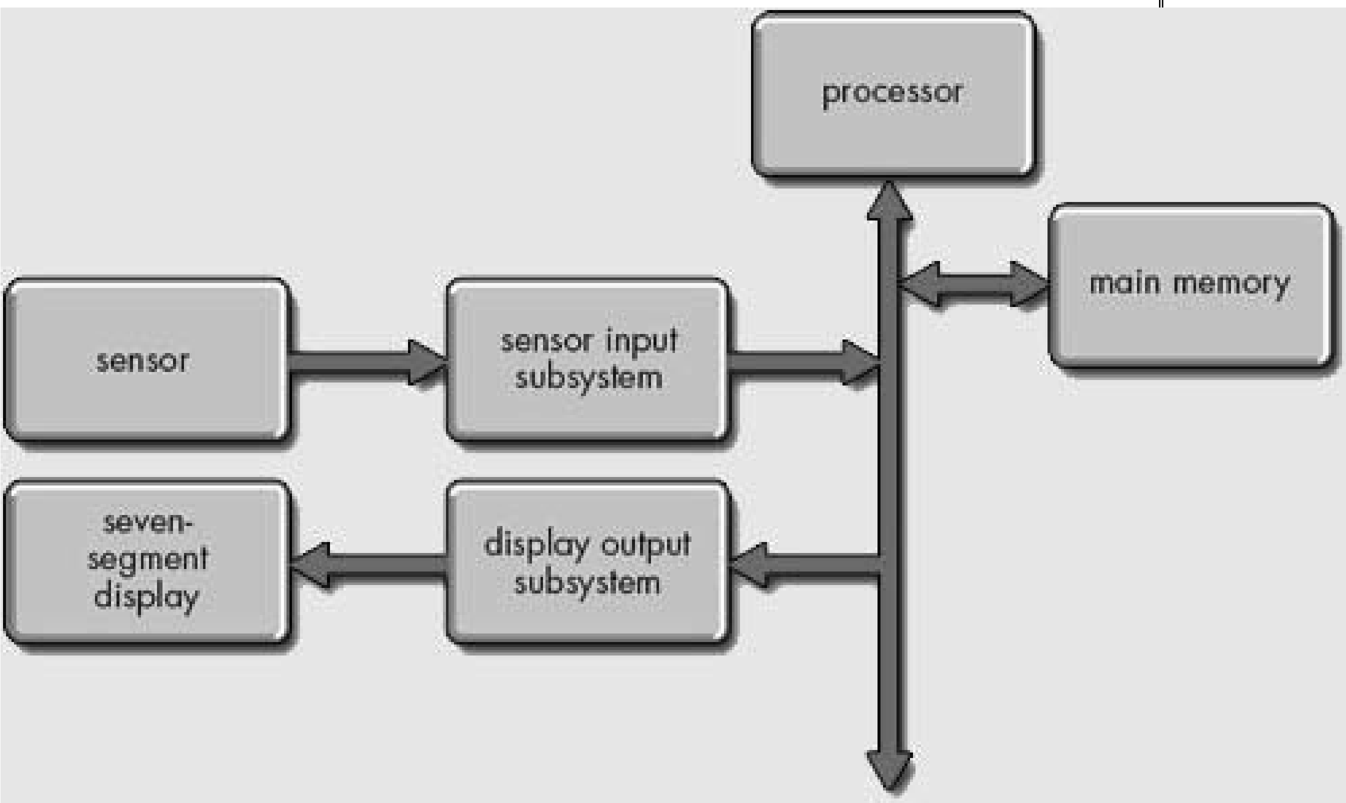

Figure 8

In products such as these electronic kitchen scales the capabilities of the processor can be used to implement additional features. In this case the scales have a count-down timer so they can be used as a kitchen timer, and they can measure in imperial units (pounds and ounces) as well as metric units. They also implement an add-and-weigh function which allows the user to set the scales' display to zero when there are some ingredients in the scalepan, making it possible to weigh the next ingredient without having to perform any mental arithmetic to add its weight to that of the ingredients already in the scalepan.

Activity 8 (Exploratory)

How might the input and output devices of the scales have to change if a countdown timer, a choice of measuring units and an add-and-weigh feature are all to be implemented? How would this change the diagram you drew for Activity 7?

The user would need some way of setting the timer, of telling the system whether measurements have to be displayed in imperial or metric, and of switching on and off the add-and-weigh feature. Input buttons would be needed for each of these tasks: to set the timer, change the system between metric and imperial and operate the add-and-weigh function.

A beeper would be needed, to sound when the timer has counted down to zero. The output display would have to have additional functionality; for instance, it would have to show ‘oz’ and ‘lb’ for ounces and pounds.

Additional input and output devices and their subsystems would have to be added to the diagram to represent the buttons used to set up the new features (inputs) and the beeper (output).

Figure 9 shows three photos of the scales’ display, each illustrating a different use. The top figure shows the display giving a reading in ounces; note that it displays fractions of an ounce. The middle figure shows the clock display; note that a colon is used in addition to the standard set of digits from 0 to 9. The bottom figure shows a weight displayed as a negative value. It may seem strange to have a ‘negative weight’, but it can occur when the add-and-weigh facility is used. Imagine that some ingredients are placed on the scalepan and the display reads 49 g. The user then invokes the add-and-weigh facility, so the display changes to 0 g. If the ingredients are then removed from the pan the display will read−49 g.

Figure 9 Three photos of the kitchen scales' display: (top) with the scales weighing in imperial units; (middle) with the timer function in operation; (bottom) negative values can be displayed for weights if the add-and-weigh facility is being used

To implement these additional features the scales' computer has to represent all the additional data that could be output on the display by predetermined codes consisting of 1s and 0s. It has to represent fractional data, negative numbers, a digital clock format and patterns to illuminate lb' and 'oz' as well as 'g'.

5.3 Digital camera

The last computer I am going to look at is the embedded computer in a digital camera.

Figure 10 shows a picture of a digital camera. The screen of the camera is displaying a picture that has previously been stored in a memory card within the camera. This memory card is not the camera computer's main memory, nor is it the secondary memory used to hold the computer's program; it is a form of removable secondary memory where the computer stores the images taken. Next to the camera in Figure 10 is an example of the memory card that this particular camera uses to store its images. The memory card can be unplugged from the camera and another memory card inserted.

Figure 10 Digital camera displaying image; a memory card is shown alongside

When the user presses the button to take a picture with a digital camera, its shutter opens, and the lens system focuses light from the image being photographed onto a device called a charge-coupled device or CCD. The CCD consists of a two-dimensional array of tiny light-sensitive cells that convert light into electrical charge. Figure 11 shows this array of cells and how the CCD is located behind the camera lens. The brighter the light that hits a cell, the greater the electrical charge that accumulates at that site. Once the camera shutter has closed, the information stored in the form of electrical charge at each cell is converted into a binary code and stored in the form of 1s and 0s in the camera's memory, and this forms the image captured by the camera. To collect colour information a system of colour filters is placed over the cells of the CCD.

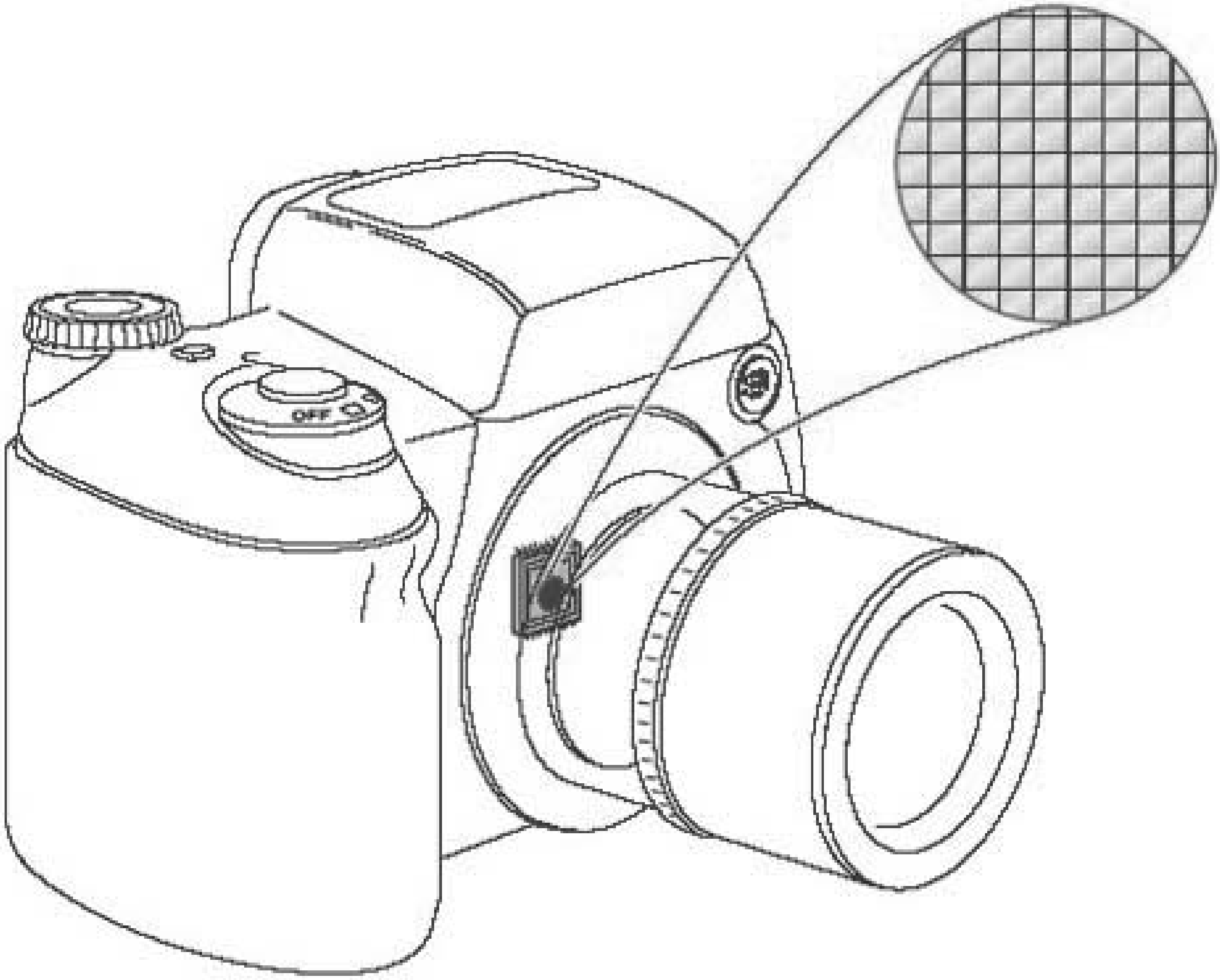

Figure 11 A representation of a CCD

This stored raw data representing the image is then processed. The colour is reconstructed and adjusted, and techniques are used to sharpen the fine detail. The result of this process is a picture ready to be stored as a file in the camera's secondary memory. To reduce the amount of stored data, the file is usually compressed – that is, the number of bits used to represent the image is reduced. In some cameras the user can select options to choose the type of compression carried out. The process of compression is described later in this block.

Figure 12 shows the actions that the digital camera performs when taking a picture. Note that this diagram is not a functional block diagram of the camera but shows the actions that must occur to take and store the picture, in the order in which they must happen. The digital camera shown in Figure 10 has some buttons that allow the user to set particular conditions when taking a picture. In addition to the button to take a picture, there are buttons to set the flash, control the preview of the stored images on the screen and set the zoom ratio. As there is a flash facility, there must also be a light-level meter incorporated into the camera; the level of light falling on the meter determines whether the flash will operate.