

An introduction to computers and computer systems



This item contains selected online content. It is for use alongside, not as a replacement for the module website, which is the primary study format and contains activities and resources that cannot be replicated in the printed versions.

About this free course

This free course is an adapted extract from the Open University course .

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 –

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

Copyright © 2020 The Open University

Intellectual property

Unless otherwise stated, this resource is released under the terms of the Creative Commons Licence v4.0 http://creativecommons.org/licenses/by-nc-sa/4.0/deed.en_GB. Within that The Open University interprets this licence in the following way:

www.open.edu/openlearn/about-openlearn/frequently-asked-questions-on-openlearn. Copyright and rights falling outside the terms of the Creative Commons Licence are retained or controlled by The Open University. Please read the full text before using any of the content.

We believe the primary barrier to accessing high-quality educational experiences is cost, which is why we aim to publish as much free content as possible under an open licence. If it proves difficult to release content under our preferred Creative Commons licence (e.g. because we can't afford or gain the clearances or find suitable alternatives), we will still release the materials for free under a personal end-user licence.

This is because the learning experience will always be the same high quality offering and that should always be seen as positive – even if at times the licensing is different to Creative Commons.

When using the content you must attribute us (The Open University) (the OU) and any identified author in accordance with the terms of the Creative Commons Licence.

The Acknowledgements section is used to list, amongst other things, third party (Proprietary), licensed content which is not subject to Creative Commons licensing. Proprietary content must be used (retained) intact and in context to the content at all times.

The Acknowledgements section is also used to bring to your attention any other Special Restrictions which may apply to the content. For example there may be times when the Creative Commons Non-Commercial Sharealike licence does not apply to any of the content even if owned by us (The Open University). In these instances, unless stated otherwise, the content may be used for personal and non-commercial use.

We have also identified as Proprietary other material included in the content which is not subject to Creative Commons Licence. These are OU logos, trading names and may extend to certain photographic and video images and sound recordings and any other material as may be brought to your attention.

Unauthorised use of any of the content may constitute a breach of the terms and conditions and/or intellectual property laws.

We reserve the right to alter, amend or bring to an end any terms and conditions provided here without notice.

All rights falling outside the terms of the Creative Commons licence are retained or controlled by The Open University.

Head of Intellectual Property, The Open University

Contents

Introduction	6
Introduction	6
Session 1: Computers and processors	9
Introduction	9
1 The first computers	10
2 The parts of a computer	12
3 The general purpose, programmable computer	13
4 Computers and data	14
5 Computers today	15
6 Summary of Session 1	17
Session 2: The components of a computer	19
Introduction	19
1 A Personal Computer	20
2 Processors	21
2.1 Multi-core processors	21
2.2 Processors in 'invisible' computers	21
3 Memory	22
3.1 Memory in 'invisible' computers	23
3.2 Memory beyond the computer	24
4 Peripheral devices	25
4.1 Input and output devices for physically impaired users	26
5 Connecting the devices	27
6 Computers as systems	30
7 Summary of Session 2	32
Session 3: Some facts about processors	34
Introduction	34
1 Processor statistics	35
2 What does a processor look like?	37
3 Moore's law	39
4 Summary of Session 3	40
Session 4: Representing data and instructions inside a computer	42
Introduction	42
1 Switches	43
2 Representing data: bits	45

3 Representing data: bytes	46
4 An alternative to binary?	47
5 Representing text in binary	48
6 Representing numbers in binary	49
7 Representing pictures and music in binary	50
8 Representing images in binary	51
9 Representing audio in binary	53
10 This session's quiz	54
11 Summary of Session 4	55

Session 5: Examples of computers 57

Introduction	57
1 The personal computer	58
2 Electronic kitchen scales	60
3 Digital camera	63
4 Summary of Session 5	66

Session 6: Computer programs 68

Introduction	68
1 Software	69
2 Operating systems	70
3 Using flowcharts to describe a task	71
4 Using flowcharts to describe a task 2	74
5 Tasks for a typing tutor	75
6 Developing an application program	77
7 The language of the web	78
8 Summary of Session 6	79

Session 7: Networks of computers 81

Introduction	81
1 Technologies	82
1.1 An example network	82
1.2 Routers and switches	83
1.3 Local Area network (LAN)	84
1.4 Wide Area Network (WAN)	85
1.5 Wireless networks	85
2 The internet	86
2.1 Out of this world	86
3 The Internet of Things	87
4 Summary of Session 7	88

Session 8: A look to the future 90

Introduction	90
1 Pervasive computing	91

1.1 Ubiquitous computing	91
1.2 Calm technology	92
1.3 Ethical implications	92
2 Green computing	93
2.1 Carbon emissions	94
2.2 Home working	95
3 This session's quiz	97
4 Summary of Session 8	98
5 Course summary	99
Next steps	100
Glossary	100
References	102
Acknowledgements	103

Introduction

Introduction

There is more to computers and processors than simply PCs. In fact computers are ubiquitous in everyday life. This free course, *An introduction to computers and computer systems*, challenges how we view computers through the examples of processors in kitchen scales and digital cameras, as well as examining the work of art that, at heart, is a computer. You will also explore how computers are connected together to achieve even more than when working alone.

This course lasts 16 hours and is comprised of eight sessions, which can be studied at your own pace. The eight sessions are linked to ensure a logical flow through the course. They are:

1. Computers and processors
2. The components of a computer
3. Some facts about processors
4. Representing data and instructions inside a computer
5. Examples of computers
6. Computer programs
7. Networks of computers
8. A look to the future

Each session should take you around 2 hours. There are a number of activities throughout the course where you are asked to note down your response. A text box is provided for you to do this, however if you would prefer to record your answers in another way that is fine. Your answers won't be visible to anyone else.

After studying this course, you should be able to:

- recognise the fundamental hardware components that make up a computer's hardware and the role of each of these components
- describe the difference between an operating system and an application program, and what each is used for in a computer
- understand something of the work involved in designing an application program
- appreciate some of the core networking technologies used to connect computers together
- reflect on some of the future directions for computers and computer systems.

Moving around the course

In the 'Summary' at the end of each session, you will find a link to the next session. If at any time you want to return to the start of the course, click on 'Full course description'. From here you can navigate to any part of the course.

It's also good practice, if you access a link from within a course page, to open it in a new window or tab. That way you can easily return to where you've come from without having to use the back button on your browser.

You can now go to [Session 1](#).

Session 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:

1. mobile phones
2. cars
3. washing machines
4. bar-code reading systems
5. central-heating controllers
6. microwave ovens
7. 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 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.

1 The first computers

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)

Or, as this ABC News report from 1974 asserts: 'One day, a computer will fit on a desk'. Watch the report at the following link:

[One day, a computer will fit on a desk](#)

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 square roots and compare two values for equality.

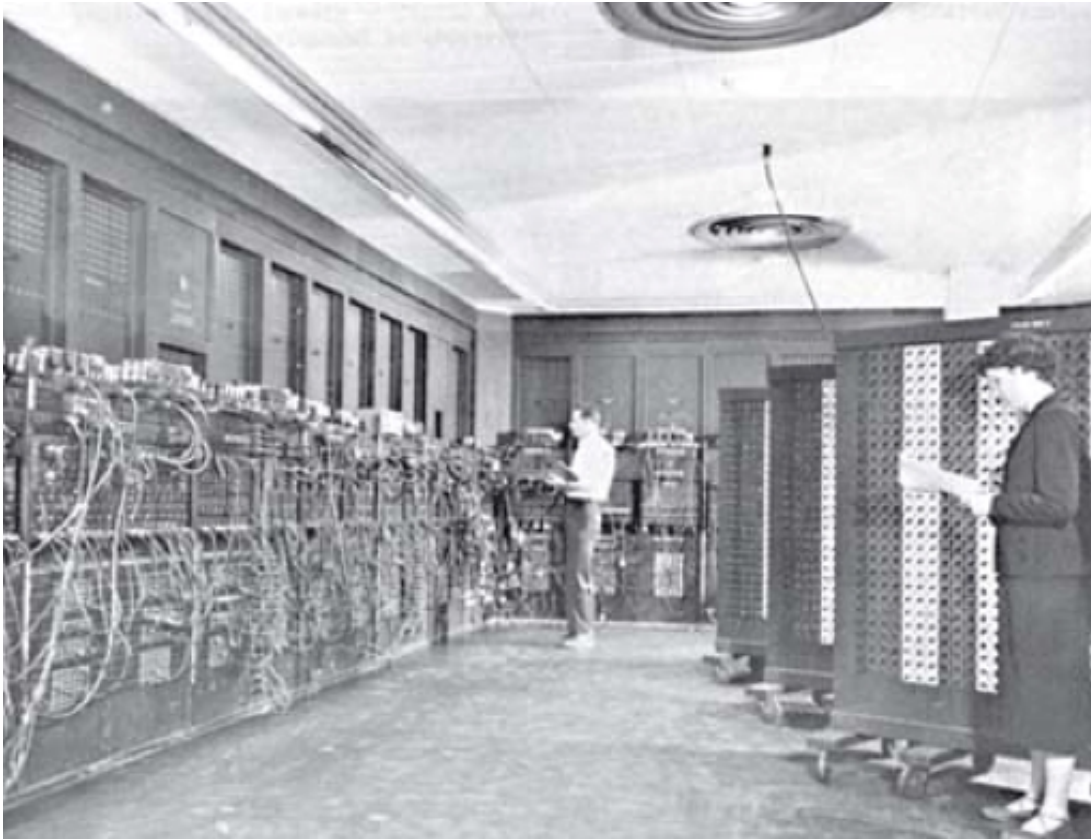


Figure 1 The ENIAC computer.

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 series of early programmable electronic computers were the Colossus machines created and used at Bletchley Park in the UK dedicated to breaking codes produced by the German cipher machines known as Lorenz machines used in the Second World War. The fundamental components of a computer have not changed since these first room-sized machines.

2 The parts of a computer

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. Many of the earliest computers read their input from paper tape adapted from existing ticker tape technology. Figure 2 shows such a computer, Colossus 10.

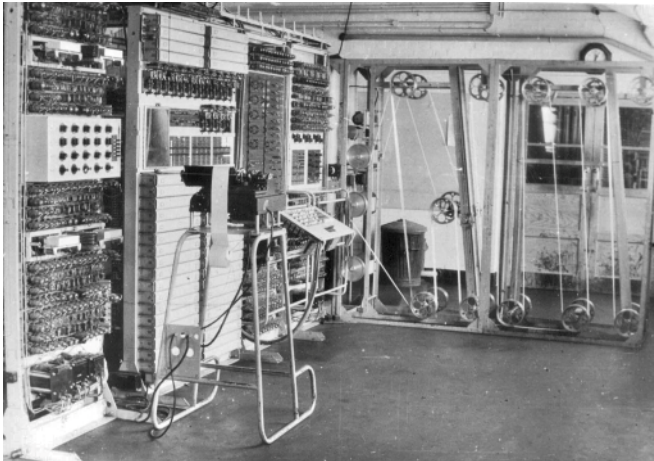


Figure 2 Colossus 10 in Block H of Bletchley Park, now part of The National Museum of Computing.

The large frame on the right of the picture with wheels attached is the input device. It guides paper tape with the stream of characters from an intercepted message punched into it. The tape was read into Colossus at 30 mph. Another existing technology, the electric typewriter, was used as an **output device** to print out the results from Colossus.

In the years since Colossus and ENIAC, many other technologies have been adapted or created to use as input and output devices. 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 of input devices. 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.

Sometimes the two types of device can be the same physical device. The electric typewriter was adapted first as an output device and later an input device. Today the touchscreen is widely used for both input and output.

3 The general purpose, programmable computer

The list of instructions the computer follows to process input and produce output is called a **computer program**. Computers are programmable.



For example, a QR code reader on a mobile phone app sends the name of a scanned bar-coded product and product details to the phone. The app has been programmed with a set of instructions that makes it:

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

What is meant by the term programmable? You can think of a program as a set of instructions, such as:

- add two numbers
- then divide the result by three.

The result will vary depending on which two numbers are inputted into the program, which is useful. For example, if you input the numbers 5 and 7, then the result given by the program is 4; if you input 15 and 6, then the output is 7.

More significantly, you can change the set of instructions to do something different, such as:

- subtract one value from the other
- then multiply the result by four.

This implies a level of versatility, since you can change the set of instructions (the program) in order to produce different results. It's this ability to change the instructions that makes a machine programmable.

```

11 4.9,3.1,1.5,0.1,setosa
12 5.4,3.7,1.5,0.2,setosa
13 4.8,3.4,1.6,0.2,setosa
14 4.8,3.0,1.4,0.1,setosa
15 4.3,3.0,1.1,0.1,setosa
16 5.8,4.0,1.2,0.2,setosa
17 5.7,4.4,1.5,0.4,setosa
18 5.4,3.9,1.3,0.4,setosa
19 5.1,3.5,1.4,0.3,setosa
20 5.7,3.8,1.7,0.3,setosa
21 5.1,3.8,1.5,0.3,setosa
22 5.4,3.4,1.7,0.2,setosa
23 5.1,3.7,1.5,0.4,setosa
24 4.6,3.6,1.0,0.2,setosa
25 5.1,3.3,1.7,0.5,setosa
26 4.8,3.4,1.9,0.2,setosa
27 5.0,3.0,1.6,0.2,setosa
28 5.0,3.4,1.6,0.4,setosa
29 5.2,3.5,1.5,0.2,setosa
30 5.2,3.4,1.4,0.2,setosa
31 4.7,3.2,1.6,0.2,setosa
32 4.8,3.1,1.6,0.2,setosa
33 5.4,3.4,1.5,0.4,setosa
34 5.2,4.1,1.5,0.1,setosa
35 5.5,4.2,1.4,0.2,setosa
36 4.9,3.1,1.5,0.1,setosa
37 5.0,3.2,1.2,0.2,setosa
38 5.5,3.5,1.3,0.2,setosa
39 4.9,3.1,1.5,0.1,setosa
40 4.4,3.0,1.2,0.2,setosa

```

The word ‘data’ has been used 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. In the examples in the previous sections, the data are the inputted numbers and those outputted by the program. The word data is actually plural (datum refers to an item of data), but is often used in the singular, e.g. ‘The data is ...’. You will find both uses in this course, depending on the context. You will find out more about the format of data a computer needs later in this course.

5 Computers today



Nowadays, a computer is just another item stocked in supermarkets. And as computers have become cheaper and smaller, they have been incorporated into a kaleidoscopic range of devices that bear no resemblance to what was once thought of as a computer. Powerful computers now sit at the heart of objects as diverse as smartphones and games consoles, cars and vacuum cleaners. The cost of computer power continues to decrease, making it possible to incorporate computer technologies into almost any object, no matter how small, cheap or disposable. And these **smart devices** are 'talking' to one another – not just within a single room or building, but across the world via the **internet**, using the **World Wide Web**. Thus, even as the computer vanishes from sight, it becomes vastly more powerful and ever-present – to use a term you'll become very familiar with, it is now ubiquitous.

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) and mobile phones help us communicate with each other, and also with other computers. Computer and telephone technologies have converged. 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 an email 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.

6 Summary of Session 1

In this session you have had a brief introduction to computers from their origins in large room-sized machines of eighty years to the pocket devices of today. You have also seen that computers work on data. In this course you will learn more about modern computers and how they process data, and more about the formatting of data so that computers can work with it as part of useful computer systems.

In the next session you will learn more about the components of a computer, its processor. You can now go to [Session 2](#).

Session 2: The components of a computer

Introduction

In this session, you will explore some of the many components, such as processors and memory within a computer, and attached devices (or peripherals as they are called) such as keyboards for input and screens for output, that go towards making a fully functional computer system.



1 A Personal Computer

Figure 1 shows an old advertisement for a laptop from about 2010. The main features of the computer are listed in this advert. One item on the list is 'Processor: AMD E450 1.65 GHz'. So this computer uses an AMD E450 processor, running at a speed of 1.65 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 (1.65 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 'Ram: 6gb DDR3'. These will be explained as the course progresses.)



The New K53U-Sx297v laptop.
Ideal for work or play, the perfect back to school Essential.

The Top Specs:
Long Battery Life.
A massive 500gb of storage.
6gb of Memory.
Windows 7 HP.

The Tech Spec:
Display: 15.6inch LED HD1080p
OS: Windows 7 Home Premium
Processor: AMD E450 1.65hz
Number of of cores: 2
Hard Drive: 500gb 5400Rpm
Ram: 6gb DDR3 (max 8gb)
Extras: Built in webcam, SD card reader, Office 2010.

Very light weight.

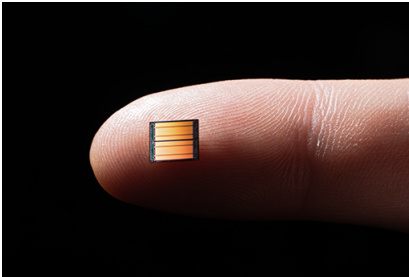
1 yr warranty including 3yrs free support.

Our Price £317.

next day delivery

Figure 1 An old advertisement for a personal computer.

2 Processors



You may remember that the quote from Danny Hillis in Session 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'.

2.1 Multi-core processors

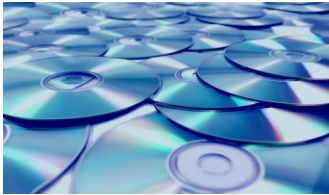
A multi-core processor is a 'processor' that contains two or more independent processors called cores. Each core performs the usual functions of loading data and instructions and executing the instructions on the data but instructions can be shared between each of the cores and run at the same time, increasing the overall speed of programs, provided that they are written in such a way to allow for this to happen.

You may think that four cores all working simultaneously would make a program run four times as fast. However, this is not the case. Each core requires its share of the data and instructions to be moved from the shared main memory into its local memory where the instructions are executed on the data. The program has to be written in such a way that a task can be split up into independent sub-tasks, each of which can be completed by a core, and then if necessary, reassembled into a final solution. This process of coordination is a small additional task for the computer to perform. Hence you do not get the full benefit of the extra cores.

2.2 Processors in 'invisible' computers

All computers, not just PCs, contain processors, so all those 'invisible' computers listed in Session 1 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 may be 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. Consequently, the PC's processor is likely to be physically larger and more costly. 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.

3 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.



Figure 2 Computer main memory modules.

This main memory is the 'main' place for storing instructions and data while a program is being executed. It is also called random access memory (RAM) from the way in which the processor accesses its contents. Its size is measured in bytes, or larger multiples of bytes such as gigabytes, a billion bytes, used in the laptop advert you saw earlier in this session. You will learn more about representing data in bits and bytes later in this course (in Session 4).

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 be downloaded from the web or installed from removable secondary memory. Removable secondary memory includes memory cards and DVDs, and older media such as floppy disks and CD-ROMs that are little used now. 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.



Figure 3 A hard disk.

Note that the term ‘hard disk’ has been used here, but in the advert in Figure 1 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 will be explained later.

Increasingly, PCs are using a solid-state drive for secondary memory. These drives use integrated circuits much like main memory and have the advantages of containing no moving parts and being much faster to access. However, they are a more expensive form of secondary memory, so some computers now have both solid-state drives and hard disks. The smaller solid-state drive is often used to contain programs so that the computer may access them quickly, while data is stored on the slower to access hard disk.

3.1 Memory in ‘invisible’ computers

Although programs and associated data are stored in secondary memory 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. Secondary memory devices usually make up the bulk of the memory in desktop, laptop and mainframe computers, and other devices such as mobile phones or tablets, but may be completely absent in **embedded computer systems**, such as control systems for appliances like washing machines and microwave ovens.



Figure 4 A central heating controller.

This means that while all computers have main memory, not all will have secondary memory. In an embedded computer system, such as the central-heating controller in Figure 4, the computer is ‘invisible’ and its software is already stored in the main memory when the computer system 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.

Activity 1

What other secondary memory device or devices are used by the PC in the advert shown in Figure 1? Make a note in the box below.

Provide your answer...

Discussion

The list of 'extras' shows that the laptop also uses SD cards as removable secondary memory.

3.2 Memory beyond the computer



So far you have looked at memory directly built into the computer, be it main memory or secondary memory. Though some secondary memory can be removed from the computer, the hardware to access the stored data is part of the computer. Think of DVDs and a DVD drive. This is very useful for offsite back-ups, copies, of your data.

There is another form of offsite data storage, in the cloud. Cloud providers such as Dropbox, Amazon Web Services, Microsoft and Google have built vast data centres to house thousands of special computers and data storage devices to store your data. Access to the data is made possible by the internet, which you will look at later in this course.

You might think a cloud as an anonymous, unknown location that is shared by many unrelated users. The location is owned, managed and operated by a business, or academic or government organisation, or some combination of these, and the hardware infrastructure physically exists on one or more of the premises owned by the cloud provider or one of their partners. This means your data is always offsite, and issues such as back-up and security of access, is managed for you by the cloud provider. This form of data storage is increasingly used as device using computers become smaller, such as mobile phones. Avoiding the need for local data storage and its hardware is an important advantage.

4 Peripheral devices



In Session 1 you saw some input and output devices, also known as peripheral devices (or peripherals for short) that are connected to the computer. Peripherals are used to load data and programs into the computer and to get the results out again. These devices include keyboards, screens, mice, printers, disk drives, sound cards, video cards and cameras.

Activity 2

List as many input and output peripheral devices that you can think of. State whether each is a device for input, output or both.

Provide your answer...

Discussion

Here are some common input/output devices:

Device	Input	Output
screen		X
speakers		X
printers		X
touchscreen	X	X
interactive whiteboard	X	X
CD/DVD drive	X	X
external hard drive	X	X
virtual reality glasses	X	X
digital camera	X	
webcam	X	
scanner	X	
keyboard	X	
mouse	X	
barcode reader	X	
joystick	X	

microphone	X
electronic stylus	X

As you saw in Session 1 early computers and most personal computers use keyboards for input and screens for output. However, those with limited manual dexterity might find it difficult to use a keyboard, and screens cannot be used by everyone with a visual impairment. You'll learn more about this in the next section.

4.1 Input and output devices for physically impaired users

An increasing amount of research has gone into making computing more accessible for people with disabilities. A user with limited movement may be able to use a joystick or a trackball to navigate a graphical user interface, and make input via programmable buttons. Ergonomic keyboards or software that converts a screen into a keyboard are now widely available.

Eye-tracker technology, which tracks the user's eye movements across a screen and controls input, has become steadily more affordable. Also available is a 'sip and puff' switch that measures air-pressure changes in a tube to convert 'sipping' (inhaling) and 'puffing' (exhaling) into signals that can be used to control a mouse, joystick or keyboard.

For visually impaired users, software such as voice recognition software and screen readers can be used for input and output, and braille readers are available for tactile output to be read by users.

In the next session you will be introduced to the concept of computer systems, to help you understand how these components actually work together.

5 Connecting the devices

So far, you have been introduced to the major components of a computer, namely a processor along with input and output devices, plus main and secondary memory. Now you will 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. 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 talking about input functions the term 'input device' is used 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' the data from the form the processor uses into information
- 'presenting' the information.

Again, when talking about output functions, the term 'output device' is used 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 was 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 5 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 5, 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 5.

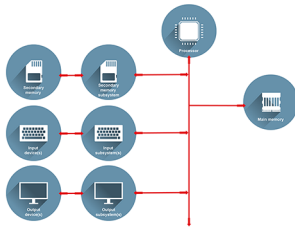


Figure 5 A functional block diagram of a computer which also shows the flow of data within the computer.

In Figure 5 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.

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, known as 'voltage high'.

(A couple of decades ago the high voltage was usually around 5 volts. When a predecessor to this course was written early in 2004, the high voltage was around 3 volts. Since then systems have used voltages near 2 volts and now even lower. Reducing the voltage is important for saving energy, crucial when the computer is battery powered such as a laptop or mobile phone. Saving energy is also important for the environment. At the end of this course in considering the future of computing you will look at green computing.)

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.

Figure 5 is an important diagram, and 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.

6 Computers as systems

As already mentioned, the functional blocks shown in Figure 5 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 more about computer hardware and software, including the processor and the programs it runs. 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.

Activity 3

For the PC shown in the advert in Figure 1 (in Section 1), write down how the following items of hardware relate to the functional blocks in Figure 5 (in the previous section). For simplicity, assume that items that provide input functionality relate to input devices, rather than input subsystems, and similarly items that provide output functionality relate to output devices rather than output subsystems.

- keyboard
- display
- 500 GB hard drive
- 6 GB DDR3 RAM
- speakers
- touchpad
- SD reader.

Provide your answer...

Discussion

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

To decide whether the 500 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 they probably meant the combination of the two.

The 6 GB DDR3 RAM relates to main memory.

The SD reader relates to secondary memory.

Activity 4

The laptop shown in Figure 1 contains a sound card and a network card. What problem arises if you try to relate these two items to the functional blocks of Figure 5?

Provide your answer...

Discussion

The two items can function both as input and as output devices.

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 that have been introduced, a word of caution. When you read books or other literature about computers you may find some of the terms 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 course you do need to make sure that you use the terms as defined here.

One term not used here that you might come across is **computer system** itself. Historically some people used the terms 'computer' and 'computer system' rather differently. The term computer was focused on the processor and computer system referred to the processor and its devices to make it a usable 'system'. 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' to cover the combination of a box with the processor and memory, and attached devices usually a screen, keyboard and mouse.

7 Summary of Session 2

In this session you have had an introduction to the components of a computer, starting with its processor, before moving on to look at some of the input and output devices that connect to it to make a useful computer system.

In the next session you will learn more about the heart of a computer, its processor.

You can now go to [Session 3](#).

Session 3: Some facts about processors

Introduction

In this session of the course you will look more closely at processors. As well as considering the fundamental question 'what is a processor?' you will explore:



1 Processor statistics

In this section you are going to find out a little more about one of the key components of a personal computer and so many other devices: the processor. It is the processor that manipulates data according to a list of instructions in a program.

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

Activity 1

Question 1

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

- ☐ 20 million
- ☐ 1 billion
- ☐ 40 million
- ☐ 125 million

Discussion

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 sold worldwide in 2020?

- ☐ 12 billion
- ☐ 123 million
- ☐ 25 billion
- ☐ 1 billion

Discussion

If the processor market was vast in 2000, consider how it has grown in the twenty years since then.

Question 3

Which of the numbers given below is closest to the number of Personal Computers sold worldwide in 2020?

- ☐ 250 million
- ☐ 25 million
- ☐ 1 million
- ☐ 1 billion

Discussion

In 2020, the same year that about 25 billion processors were sold, about 250 million Personal Computers were sold.

In other words, only one in a hundred processors were used in a Personal Computer. The other processors found their way in other devices.

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.

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 electronic circuits, around the size of a fingernail, which contain many millions of electronic components manufactured as one very complex circuit. Figure 1 shows how a processor manufactured as an integrated circuit is packaged so it can be used as a component in an electronic circuit. The processor is surrounded with pins to connect 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.

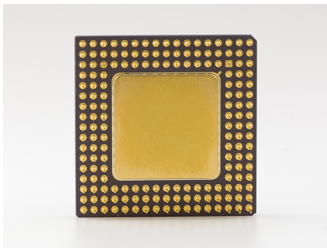


Figure 1 A processor surrounded by gold connection pins.

Figure 2 shows the packaged processor assembled with other components on the motherboard of a computer. A motherboard is the major circuit board inside a computer: it holds the processor, the computer bus, the main memory and many other vital components.

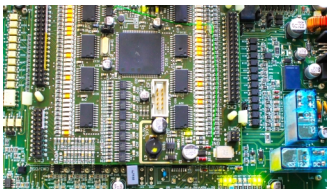


Figure 2 A processor assembled on a motherboard along with other circuit components.

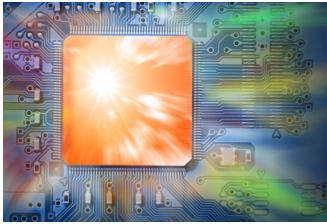
A processor is essentially a single integrated circuit that contains the central processing unit (CPU) of a computer. The CPU can be thought of as the command centre of the computer; amongst other functions, it interprets each program's instructions and organises the storage and retrieval of the data involved. Developing the CPU into a single component has considerable benefits, including reduced manufacturing and running costs and the potential to manufacture in very large numbers. The CPU can then be placed on a single circuit board, the motherboard, along with other key components such as memory. Bringing the components together allows the computer to work more efficiently as well as bringing down the cost of manufacture.

The processor runs at a set speed, its clock speed. This is measured in gigahertz (GHz). Hertz is the number of cycles (actions) per second, and the giga prefix multiplies that by a billion. This means that a processor rated at 2.2 GHz, completes 2,200,000,000 cycles per second. This clock is the speed that everything happens in your computer, such as performing a calculation and retrieving data from your hard-drives, all controlled and synchronized by your processor. Clock speed can be a very important measure in some situations, for example in scientific computing where there are many calculations for the

processor to perform. However, the processor speed will probably be less critical on your laptop, for example, where the processor will spend much time waiting for you to enter data via the keyboard, or to retrieve a page from the internet.

Processors have been getting faster, but the real growth in processor performance has been achieved by a different method. In the next section, you will look at a particular example of how more and more components have been brought closer and closer together to achieve greater overall computing performance.

3 Moore's law



Computers today are smaller, faster and cheaper than they have ever been. This statement could have been made at pretty much any time over the last 30 years and it would have been true; it's doubtful that any other product has maintained such rapid development over such a long period of time. But just how fast has computer development been? In 1965, Gordon Moore, the founder of the giant Intel Corporation, wrote in *Electronics* magazine that he expected the density of electronic components in an integrated circuit on a silicon chip to double every year for at least the next ten years.

Whether we look at the CPU or at memory, the fundamental component of the computer is the transistor. The incorporation of many transistors onto a single silicon chip, in the form of an integrated circuit, started a process of miniaturisation that continues to this day. More and more transistors are placed in a given space; as the distance between the transistors shrinks, so the speed of communication between them increases and the cost per transistor falls. When he made his prediction, Moore felt that he could see how the technologies needed to achieve it would evolve for the next five to ten years. In fact, the prediction was slightly out, as the density of components has doubled every two years rather than the year that was the rate in 1965, and it is this rate of growth that is now associated with Moore. Nevertheless, it is quite astonishing that development has continued at this rate for more than 50 years after the initial prediction. In fact, behind this lies a massive amount of investment to ensure that the trend continues. At some point, Moore's law, as it is called, moved from being a prediction to being the target for an industry. Intel's Core M processor, released in 2015 fifty years after the prediction was made, holds 1.3 billion transistors that each measure about 14 nanometers (nm) in size. To give you an idea of the scale: a flu virus is 20 nm in diameter.

4 Summary of Session 3

In this session you saw what a processor is, something of the vast market for them, and how they have developed over time.

Now that we have a processor and its components assembled together, in the next session you will learn something more about how the data is interpreted and processed to produce useful results.

You can now go to [Session 4](#).

Session 4: Representing data and instructions inside a computer

Introduction

In this session, you will explore how we can represent data and instructions in a format that processors can handle. You will look first at instructions for the processor, and then at the many forms of data from plain text to sounds that computers can process.



1 Switches

You have seen in the previous session 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**. You can see that there are eight bits in 10001110 and hence it is called an 8-bit code. As the code is in binary it is also 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 1

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

Provide your answer...

Discussion

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.

2 Representing data: bits



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 1 showed that there are four possible combinations of 1s and 0s in a 2-bit binary code. 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 2

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

Provide your answer...

Discussion

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 might find it helpful to refer to the following resource: [Using exponent notation](#).

You have seen that one binary digit of computer data is called a bit, but when we talk about computer data we often use the term byte. The next section takes you from bits to bytes.

3 Representing data: bytes

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 3

Answer the following questions:

1. How many bits are there in 4 bytes?
2. How many items of data can be represented by 1 byte?

Provide your answer...

Discussion

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 a 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.

4 An alternative to binary?

Some very early computers, such as the ENIAC you saw in Session 1, tried to represent data using our usual base-10 system. So 0 volts was used to represent the digit 0, 1 volt to represent the digit 1, and so on, all the way up to 9 volts to represent the digit 9. However, having a range of different values caused problems: voltage is not steady in a circuit, it varies as the circuit is switched on or off, and it also varies as the electricity flows through components. To cope with this, a lot of circuitry was needed just to distinguish between the different voltages, which took up a lot of space and generated a lot of heat.

The advantage of representing data in binary is that only two ranges of voltage need to be detected. The actual voltage values are defined in the specification of the electronic transistors used in a processor. In a particular transistor, any voltage between, say, 0 and 1.3 V ('low' voltages) might be interpreted as the digit 0, and any and any voltage above 1.7 V ('high' voltages) might be interpreted as the digit 1. In this case, the circuit would be designed to prevent voltages between 1.3 and 1.7 V. This gap means that if there are any small random dips or increases in the voltage (called noise), the two binary digits will still be distinguishable.



Figure 1 Representing 0s and 1s using high and low voltages.

This simple to implement and efficient method of representing binary data with high- and low-voltages is the basis for contemporary processors. However, we don't think in binary digits grouped into bytes. In the next section you will see how bytes can be rendered as something we humans understand.

5 Representing text in binary

Most modern systems for encoding text derive in part from **ASCII** (American Standard Code for Information Interchange, pronounced 'askee'), which was developed in 1963. In the original ASCII system, upper-case and lower-case letters, numbers, punctuation and other symbols and control codes (such as a carriage return, backspace and tab) were encoded in 7 bits. As computers based on multiples of 8 bits (or a byte) became more common, the encoding system became an 8-bit system, and so could be expanded to include more symbols.

When binary numbers were assigned to each character in the original ASCII system, careful thought was given to choosing sequences of values for the characters of the alphabet and numerals that would make it easy for a computer processor to perform common operations on them. (These encodings were preserved in the 8-bit system by simply padding out the leftmost bit with a 0.)

To illustrate, let's look at the 8-bit encoding of some of the lower-case and upper-case letters of the English alphabet, shown in Table 1.

Binary	Character	Binary	Character
0100 0001	A	0100 0001	a
0100 0010	B	0100 0010	b
0100 0011	C	0100 0011	c
0100 0100	D	0100 0100	d
0100 0101	E	0100 0101	e
0100 0110	F	0100 0110	f
0100 0111	G	0100 0111	g
0100 1000	H	0100 1000	h
0100 1001	I	0100 1001	i
0100 1010	J	0100 1010	j

Table 1

Notice that the ASCII values for corresponding upper-case and lower-case characters always differ by one bit, shown in blue. This means that converting from upper case to lower case (a very common manipulation of text) is simply a matter of 'flipping' one bit. The original ASCII codes are suitable for representing North American English, but do not allow for other languages that use Latin characters with diacritics, nor for languages that do not use a Latin alphabet at all. This was a major problem with the ever more widespread use of computers and processors. It took a long time for an acceptable international standard to emerge but since 2007, the standard encoding system for characters has been Unicode Transformation Format-8 (UTF-8) which uses a variable number of bytes (up to 6) to encode characters in use across the world. However, in order to maintain backward compatibility, the original 127 ASCII codes are preserved in UTF-8. In the next section, you will look at numbers.

6 Representing numbers in binary

Now looking at the encoding for numerals in Table 2, we can see how the characters 0–9 are represented in UTF-8.

Table 2 ASCII table for numerals

Binary code	Character
0011 0000	0
0011 0001	1
0011 0010	2
0011 0011	3
0011 0100	4
0011 0101	5
0011 0110	6
0011 0111	7
0011 1000	8
0011 1001	9

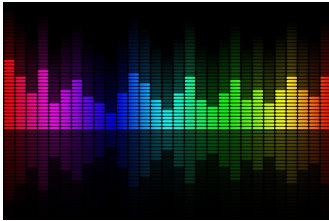
Note, this is the character representation of the number to display that number on a screen or to print in a document. It is not the actual representation that a processor will use when performing numerical calculations!

Consider the number '11'. To print this number we use two characters, a '1' followed by another '1'. However, this is a very inefficient way for a processor to handle numbers in a calculation. Instead, the number eleven can be simply encoded in a byte as 00001011. That is one byte, not two, and in a form that instructions such as add and subtract can be directly applied to the bits for efficient calculations. There are other special ways to encode numbers for processing. They are beyond the scope of this course.

Modern computer programming languages such as Python handle the conversion between the two representations of a number automatically. However, in many languages it is for the programmer to declare in the program whether a number is to be used say for printing on a report, or for calculating the values to be printed.

The logic of the program informs the processor how to handle each byte, whether it represents a character or a number, or even another data format altogether as you shall see in the next section.

7 Representing pictures and music in binary



You can see images and hear audio, but how does a processor do this?

Activity 4

Make a note of the digital media you already use.

Provide your answer...

Discussion

Perhaps you are a 'consumer' of media, and you like to watch television, listen to the radio, listen to music and look at pictures online. Do you also look at photographs and videos shared by my friends and colleagues on social networking sites?

Possibly you are also a media 'creator', and take photographs on your camera and share them electronically later. On short visits to places and events, many people use their smartphone to record some of the things they see and hear and post directly to social networking sites.

In using and creating digital media, we are often representing part of the real world in a digital form that can be stored, transmitted or manipulated using information and communication technology.

8 Representing images in binary

On a screen, images are generated by dividing the display into a large number of tiny units called pixels (from 'picture elements'). Each pixel can, in general, be displayed in a number of different colours and levels of light intensities or brightness. The greater the number of pixels used for a given size of display, the more detail can be displayed and the higher the quality of the image.

Now you'll consider an artificially small-scale example: a display using eight rows of sixteen pixels. In Figure 2 you can see how this system could display a 'triangle'.

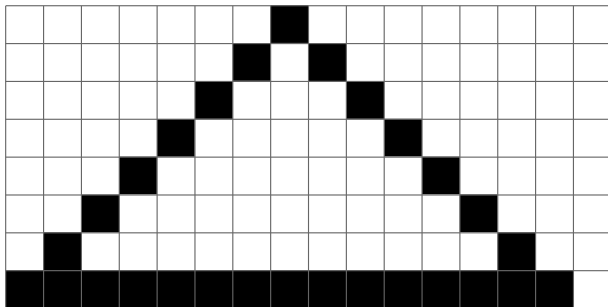


Figure 2 Black-on-white 'triangle' on a 16 x 8 pixel display.

Activity 5

What do you notice about the representation of the three sides of the 'triangle'?

Provide your answer...

Discussion

The horizontal side is a perfect straight line, since all the pixels are lined up along the row. The other sides have a stepped appearance, and only approximate a straight line. You may have noticed this effect on old mobile phone displays, which often used only a small number of pixels compared with, say, a computer display or the screen on a contemporary smartphone.

Next time you use your computer, look at a photograph and 'zoom in' as far as you can (usually an option under the View menu) to examine it in detail. The number of pixels in the 'zoomed in' section hasn't changed, but each pixel has expanded in the display so that it is possible to see each one as a discrete square. From normal viewing distances, each pixel merges with its neighbour so that we see a smooth image.

In other words, we have taken the smooth images we see around us, and broken them into tiny units that can be represented in binary to a processor, and when viewed appropriately, can still look like a smooth image to us.

For the simple black and white triangle, we can represent the state of each pixel with a bit, setting it to 0 for white and 1 for black. For more colours, we will need to use more bits at each pixel. If we use two bits at each pixel we can have four colours (remember $2 \times 2 = 2^2 = 4$) so that we could use 00 for white and 11 for black, and 01 for red and 10 for blue. As

we use more bits at each pixel we can add more information and show a greater variety of colours as well as other aspects of the picture such as how bright the pixel is.

Adding more detail to each pixel needs more bits to record that detail. Consider a screen that has a resolution of 1920 by 1080 pixels. That means there are 2 073 600 pixels on the screen when you look at it. If one byte is used to record the colour and so on of each pixel (remember one byte is eight bits), that means 16 588 800 bits of data are needed to describe the screen.

In some professional video applications, three bytes (24 bits) are used to store the data to be displayed at each pixel. That means 49 766 400 bits of data are needed to display one image measuring 1920 x 1080 pixels.

You may have come across the term 4K, which is used to describe an increasingly popular television and cinematography display resolution. 4K UHD is the dominant standard (there are several 4K standards!) and is four times the size of my monitor. In other words, the display is 3840 x 2160 pixels, or 8 294 400 pixels. Consider the number of bits needed to display an image of that size if each pixel has one byte to record its colour and additional detail such as brightness.

As file sizes increase so it is harder to process them (they simple take longer to process because there are more individual data elements to work on and more memory is needed to store this data while it is being processed). It becomes harder to use large files in other ways too. Some are obvious: consider the time taken to download an image or to stream a video over the internet; other perhaps less so: consider the size of your backup files.

There has been much research into making image files easier to handle producing a variety of standards depending on the application, such as jpeg (defined by the Joint Photographic Experts Group) which is particularly suited to photographs, gif (Graphics Interchange Format) which is particularly suited to simple graphics, and png (Portable Network Graphics) which is particularly suited to online viewing.

9 Representing audio in binary

Just as we see the world smoothly, not in steps, so too we hear the world smoothly. However, to represent sound in binary we apply the same technique and break sound up into tiny units we can represent in binary.

If you shout, hit a piano key or drop a plate, then you set particles of air vibrating – and any ears in the vicinity will interpret this tremor as sound. The sounds we hear generally consist of small rapid movements (changes, fluctuations) of the atmospheric air pressure that surrounds us. Sound can also be transmitted through other media, such as water, so not all sound consists of fluctuations in air pressure. However, this course will only consider sound in air.

A microphone is used to convert the changes in atmospheric pressure wave into an electrical signal with a voltage that varies in accordance with the pressure of the original signal. This electrical signal needs to be converted into a digital representation. However, there is no audio equivalent of a 'pixel' rather we talk of sampling rate. Associated with each sample is its depth (how many bits are used to record the data at that sample). For example, CD audio samples are taken 44 100 times a second (which you may see as 44.1 kHz), and each sample uses 16 bits for each stereo channel. Hence you will the quality of a recording determined by its sampling rate sampling and depth.

As with image files, audio files can quickly become very large as the sampling rate and depth increase. Similarly, there are many standards to encode and compress audio files to make them more manageable, such as MP3 (the third audio standard from the Moving Pictures Expert Group) and flac (an open source format from Xiph.Org Foundation), as well as many proprietary formats such as Microsoft's WMA (Windows Media Audio).

10 This session's quiz

You now have the opportunity to try a quiz in which you can test your learning of the course so far.

[Session 4 quiz](#)

Open the quiz in a new tab or window and come back here when you've finished.

11 Summary of Session 4

You'll end this session with a short activity.

Activity 6

In this session, you have looked at some examples of how data can be represented in binary. However, for processors to work, what else needs to be represented in binary?

Provide your answer...

Discussion

We need to represent the instructions that a processor applies to data.
You will learn more about this in Session 6.

In this session, you have seen how we can represent data in a format that processors can handle. You will come back to how processors handle data in Session 6, but before then, in the next session, you will see some different examples of 'computers'.

You can go to [Session 5](#).

Session 5: Examples of computers

Introduction

In this session, you will look at three different examples of computer systems: a PC, which is obviously a computer, and a set of electronic kitchen scales and a digital camera, which are not so obviously computers.



1 The personal computer

You will find that all three of the examples match with the functional block diagram of the laptop computer given in Figure 1 in Session 2, 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.

You looked at how the components of a computer could be related to the functional block diagram. Figure 1, which is a functional block diagram for the laptop, shows the data flow between the specific components of the laptop. Notice that in this diagram there are the specific input and output devices and items of secondary memory instead of the generic items seen previously in the functional block diagram in Session 2. That diagram is a generic for any computer; whereas Figure 1 below is its specific for the laptop.

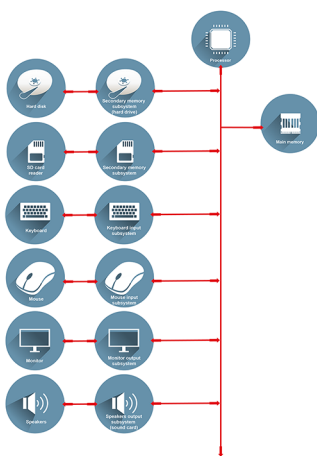


Figure 1 Functional block diagram for a laptop.

Other PCs may have some additional input devices such as a stylus, 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.

(Johnson, 2004)

In Session 4 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. You also saw how the images of the ‘movie trailer’ in the above quote can be represented in your computer, even though images are a more complex issue of data representation. 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.

2 Electronic kitchen scales

A set of electronic kitchen scales is shown in Figure 2. 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. Using a bowl on the scales allows for weighing many individual objects like the oats in Figure 2.



Figure 2 Set of scales showing a reading of 313 g.

It might be possible to think of these electronic kitchen scales as a computer, in the sense 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 ‘invisible’ 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 then 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.

Activity 1

Using the information about the scales given above, create a functional block diagram for the kitchen scales. Note that these scales have no secondary memory.

Discussion

Here is an example answer:

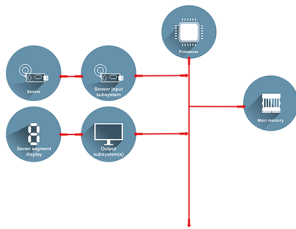


Figure 3 Functional block diagram for the kitchen scales.

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 2

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 1?

Provide your answer...

Discussion

The user would need some way of setting the timer, of telling the system whether measurements have to be displayed in metric or imperial, 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 when operating in imperial mode as well as 'g' for grams when operating in metric mode metric.

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 4 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 4 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'.

3 Digital camera

The last computer to look at in this session is the embedded computer in a digital camera. Figure 5 shows a picture of a digital camera. Inside the camera there is 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. The memory card can be removed from the camera and another memory card inserted.



Figure 5 A digital camera.

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 light sensitive sensor. Two forms of light sensitive sensors are in common use: a charge-coupled device or CCD and complementary metal-oxide-semiconductor or CMOS sensors.

The two types of device work in the same way. They consist of a two-dimensional array of tiny light-sensitive cells that convert light into electrical charge. Figure 6 shows this array of cells and how the sensor 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 sensor.

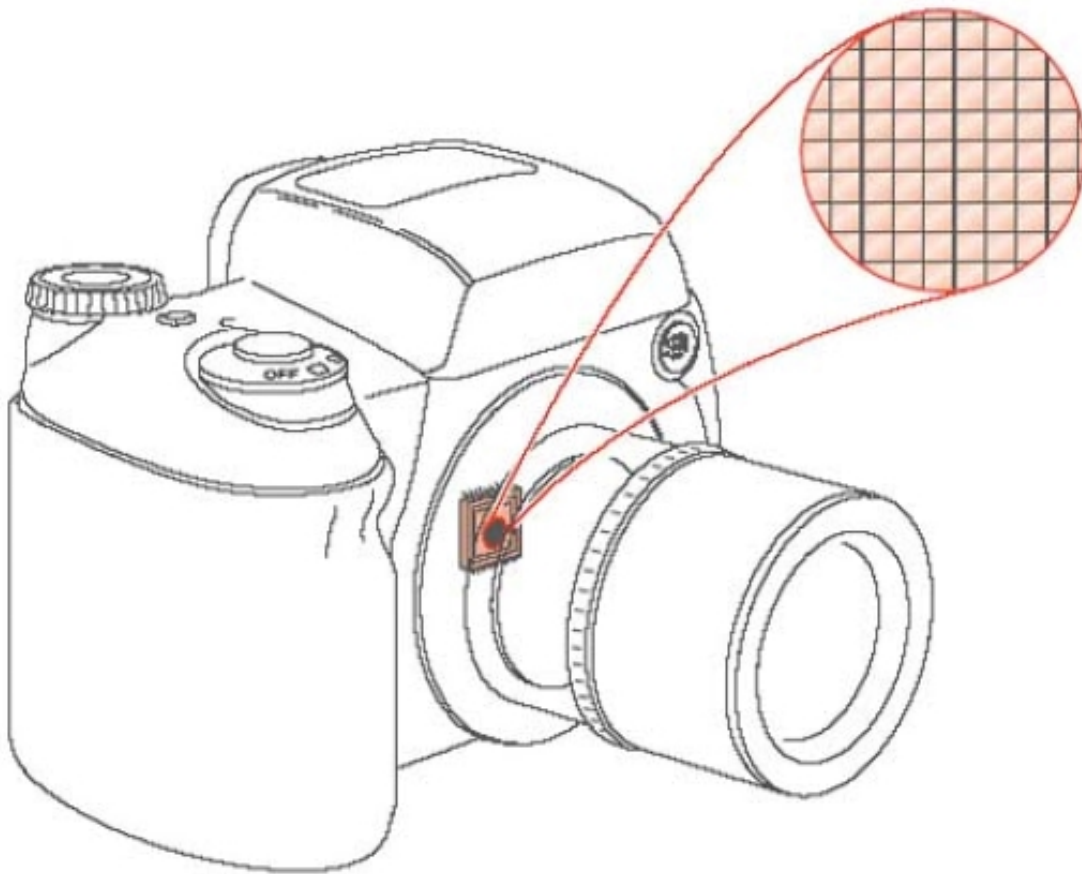


Figure 6 A representation of a light sensor array in a camera.

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.

In the example here, we follow the practice of most users and have the computer in the camera to do the image processing. Professional and serious enthusiast photographers will often take the raw image from the camera and process it in a far more powerful PC with a large monitor attached. This allows them greater control of the image processing process, with the ability to see fine detail as they make changes directly.

Figure 7 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 7 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 control the zoom lens and so on. 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.



Figure 7 Diagram showing, in order, the processes that occur when taking a picture with a digital camera.

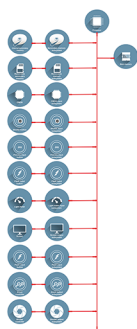
As with the PC and the electronic kitchen scales, a specific form of functional block diagram can be created for this digital camera.

Activity 3

Using the information about the digital camera given above, draw the specific functional block diagram for this camera.

Discussion

This computer has main memory and also two items of secondary memory: the removable memory card and the internal secondary memory. The input devices are the light sensor plus the buttons to take a picture, preview the stored images, set the flash and so on. The light meter is also an input device. The output devices are the camera's screen, the flash mechanism, the zoom and the control to open and shut the shutter. (If you are a camera enthusiast you may also have thought of the various controls for the shutter aperture, the focus etc., but as these are not explicitly mentioned them in the text they are not included in the answer.)



4 Summary of Session 5

You have seen that although the three products you have looked at are very different types of computer, they all embody the same basic functionality and a version of a functional block diagram can be drawn for each product to illustrate this.

One feature of the PC is the range of forms of secondary memory it can use, and also the variety of input and output devices which the user can choose. The kitchen scales' embedded computer is relatively simple with no secondary memory and relatively few input and output devices. The computer within the camera has a processor which needs to implement several complex processes to manipulate the image, has secondary memory and has many input and output devices.

Having considered the various forms data can take in Session 4 and the various forms computers can take in this session, you will next look at how we bring these together and control the processing of data by computers when you start to look at programs. That is the topic of the next session.

You can now go to [Session 6](#).

Session 6: Computer programs

Introduction

In this session, you will explore how a processor is given instructions to perform a defined function. The instructions are known as software. There are two broad classes of software: operating systems and applications, though these can be called by different names. You will be introduced to one technique that is used to organise the instructions for a computer to understand. Finally, you will look at some special purpose instructions targeted at web browsers, which will lead into the next session on the Internet.

```
130
131 def get_webpage(url: str,
132                 user_agent: str = 'Mozilla/5.0 (windows NT 10.0; win64; x64; rv:75.0) Gecko/20100101
133                 Firefox/75.0') -> str:
134     """Return text of web page at 'url'."""
135     time.sleep(random.randint(3, 5))
136     req = Request(url)
137     req.add_header('User-Agent', user_agent)
138     with urlopen(req) as f_web:
139         web_page = f_web.read().decode('utf-8')
140     return web_page
141
142 def read_local_webpage(folder: str, filename: str) -> str:
143     """Return local copy of webpage."""
144     f_lcl = Path(folder) / filename
145     return f_lcl.read_text(encoding='utf-8')
146
147 def set_folder(folder_name: str):
148     """Ensure folder_name exists."""
149     Path(folder_name).mkdir(parents=True, exist_ok=True)
150
151 def setup_log(subject: str):
152     """Create log file and format log entries."""
153     logging.basicConfig(
154         filename=f'../{subject}/{subject}.log',
155         format='{levelname:8s} {message}', style='{',
156         level=logging.INFO)
157
158
159
160
```

1 Software

The instructions are brought together in software. Software is a collection of the instructions that tell the computer how to work. In contrast, the physical hardware from which the system is built and actually performs the work is called hardware.

Software can be split into two categories, application software and operating systems.

Application software is the name given to programs which enable a computer to perform specific tasks. The program that processes the image in the digital camera is one example; a word processor running on a PC is another.

In computers that are running several application programs, the programs may well be sharing some of the computer's resources, such as its display or its hard disk. If this is the case then an **operating system** provides general-purpose software that controls the sharing of resources amongst the various programs, making sure that they are not competing for the same resource. The operating system on a PC makes it possible for, say, information about incoming email to appear on the screen whilst a word processor is running and a document is printing. The operating systems you are most likely to be familiar with in business and in schools is *Windows*®. There are however many other operating systems, such as macOS on Apple Macs, and iOS on Apple smartphones. The majority of other smartphones use Android as their operating system. Google developed of Android from UNIX, an operating system that is far older than Windows and very popular for back office functions that run at scale and require excellent reliability. For example, UNIX is used on nearly all internet servers.

Very simple computers, such as the one in the kitchen scales, have only one program running and consist of a modest set of resources. In such simple systems the distinction between the operating system and the application program is not clearly defined, and it is not customary to distinguish between them. This is typical of **embedded computers**, but then they are dedicated to deliver one task. In more complex computers an operating system becomes useful, and in something as complex as a PC it is a crucial component.

In the next section you will look a little more closely at operating systems. Then you will be introduced to how application programs are developed. Finally, you will see a little of a special form of application programming, and how to program for the internet.

2 Operating systems

You have already seen that operating systems organise the sharing of resources. But they do much more than this; they ensure the efficient running of a computer by:

- loading application programs from secondary memory into main memory and managing their execution
- supporting application programs by managing their use of the computer's resources
- managing the storage of programs and data in secondary memory
- accepting inputs from and supplying outputs to the user.

Next, you will examine each of these four aspects of an operating system in turn, using the PC as an example.

Unless an application program has been recently used and hence is already stored in main memory, the operating system will need to find the program in secondary memory, transfer it into main memory and arrange for the processor to execute it. If the user is running more than one application program, say a word processor and a drawing package, the operating system will need to manage execution of both in order to ensure that these two programs do not interfere with each other. When the user closes down an application program, the operating system has to manage this process and ensure that the computer can continue to operate normally.

Application programs make use of the computer's resources. For example, they send data to the display. Rather than the application program containing the instructions to perform tasks like this, they call on the operating system to perform the tasks on their behalf. This makes application programs easier to write.

In addition to organising the transfer of application programs from the secondary memory to the main memory, the operating system has to manage the process of storing application programs in secondary memory when the user first installs them. It also has to organise the storage of files that users create while they are running application programs. For example, with a word processor the operating system organises saving a newly created file to a folder specified by the user. On request, it also organises the retrieval of a previously saved file. Less obvious is the equally important task of storing and organising the temporary data generated by the word processor while it is running.

Accepting inputs from the user and generating outputs for the user are important functions of the operating system, and ones that can make a difference between a computer being easy to use or difficult. In the early days of PCs, users had to type text commands in order to get their computers to perform tasks, and all information from the computer came in the form of text. This was because the operating system of those days, DOS, did not have the 'graphical user interface' that everyone takes for granted today. There were no icons or menus on the screen, and pointing and clicking with a mouse was not an option. To a very large extent, it is the operating system, rather than the hardware, of a PC that influences how easy users find it to use.

In brief, the operating system in a PC not only controls the PC's resources but also hides many of the complexities of using a computer from the user, making the user's task easier. In the next section you will look at some of the programs making demands on the PC's resources, application programs.

3 Using flowcharts to describe a task



Application programs are designed to perform specific tasks. These tasks range from the relatively simple to the extremely complex. In this section you will look at what is involved in planning a program to perform some simple tasks.

There are many ways of writing an application program. However, one common starting point is to break down the overall task, or objective of the program, into smaller tasks. One technique to achieve this is to draw a flowchart, showing each of these smaller steps on the way to delivering the task.

In order to write a program, the task the program will perform has to be first written as a list of actions. The actions have to be given in an order that will ensure the task is carried out successfully.

Activity 1

Write down, in order, the list of actions you would have to carry out to boil some water in an electric kettle.

Provide your answer...

Discussion

You may have come up with the following list of actions:

- take lid off kettle
- turn on tap
- fill kettle with water
- monitor water level and turn off tap when correct
- put lid on kettle
- plug kettle in
- switch on kettle.

Your list may be different from this one. For example, you may pour the water into the spout without taking the lid off, or the method of supplying power to the kettle might not use a switch. This doesn't matter. What is important is that you can see how even a very simple task can be described as a series of actions, and that these actions must be given in a particular order for the task to be carried out successfully. In my answer, for example, it would be impossible to carry out the action to put the lid on the kettle if it hadn't been taken off earlier in the sequence of actions.

Consider a very simple set of electronic scales. These scales have an on/off switch, but no other input buttons, and a display to show the weight of the object in the scalepan in grams.

Activity 2

Write down, in order, the list of actions that the computer inside the scales has to carry out in order to show an object's weight on the display.

Provide your answer...

Discussion

The computer has to:

- accept data from the sensor that measures the displacement of the scalepan
- transform the data from the sensor into data for the display
- send the display-formatted data to the display.

The list of actions in the comment to Activity 2 can be shown diagrammatically in a type of diagram called a flowchart. Figure 1 shows how you could write this sequence of actions as a **flowchart**.



Figure 1 Flowchart of the tasks performed by simple electronic scales.

The shapes of the symbols used in flowcharts are significant. Flowcharts are a common language used to communicate processes and it is important to be consistent in their use. This flowchart uses three different symbols to show:

- the start/end points of the process
- data input or output
- a process to be carried out.

Figure 2 defines these symbols. It also defines an additional symbol that shows a decision being made, and another for showing connectors.

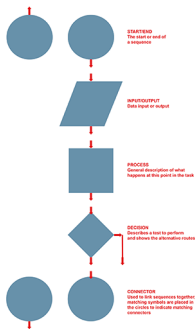


Figure 2 Flowchart symbols

In flowcharts, lines are used to connect symbols together, and arrows on these lines indicate the order in which tasks are carried out. You can see this in Figure 1.

The next example incorporates a decision box. It involves a slightly more complex set of electronic scales: they have one additional button on the front that allows the user to select whether the weight is displayed in grams or in pounds and ounces. A flowchart incorporating this choice of display format is shown in Figure 3. It uses the decision box to make a choice about which piece of the program will be run. There are two exit routes from the decision box; each route is called a branch. If the user has requested that the weight should be displayed in grams, the Yes branch is followed so that the sensor data can be transformed to gram format. If the user has requested that the weight should be displayed in pounds and ounces, the No branch is followed to transform the sensor data to pounds and ounces instead. Note that once the translation to the selected output format is complete the branches of the flowchart come together again and the 'send to display' part of the task is run regardless of which branch was taken earlier in the program.

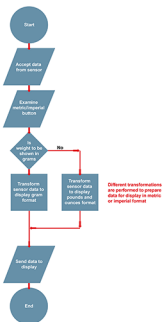


Figure 3 Flowchart for electronic scales with metric/imperial selector button

4 Using flowcharts to describe a task 2

Now consider what happens when you are weighing, for example, flour on a set of scales. You slowly add more flour to the scalepan until you reach the desired weight. As you do this the display constantly changes, showing the weight increasing as you add more flour. To do this, the scales' computer must repeatedly examine the input and update the display each time it does so. The flowcharts in Figures 1 and 3 do not implement this. They simply take one reading and need to be re-started to take another.

The flowchart can be changed so that the input is repeatedly examined and the output repeatedly displayed by means of what is called a loop. A loop allows a certain part of a flowchart to be carried out as many times as necessary depending on the results of a decision. In this instance, after every 'send to the display', the flowchart in Figure 4 shows a loop back to examine the sensor again, provided the scales have not been switched off. Note that a loop has to start as a branch from a decision box. Spend a few moments examining the loop in Figure 4; make sure that you understand how this flowchart differs from the one in Figure 3 and why it would enable the scales to display an increasing weight as, for example, flour was slowly added to the scalepan.



Figure 4 Flowchart for electronic scales incorporating a loop

Having seen how flowcharts can be useful to help understand the individual steps that make up a task, in the section, you will think about a more complicated example.

5 Tasks for a typing tutor

Now here's an example of a task that an application program called a typing tutor needs to perform. A typing tutor checks how accurately the user types certain groups of letters at the computer keyboard. For example, the group 'asdf' might have to be typed ten times as follows 'asdf asdf asdf ...'. As the user types at the keyboard, the program checks that the correct letters have been typed in each group, using space characters to define where a group ends. When the program has counted that ten groups have been entered it displays how many correct and how many incorrect groups have been typed.

The flowchart to describe this task is rather complex, so you will work through it step by step. One helpful step would be to write down a list of actions in the same manner as you did for boiling the kettle.

Activity 3

Starting with the two actions given below, write, in order, the list of actions that need to be carried out to complete the task of the typing tutor.

- Read in a group of letters from the keyboard
- If the letters are asdf then count this as a good entry and ...

Provide your answer...

Discussion

A suitable list of actions would be:

- Read in a group of letters from the keyboard
- If the letters are asdf then count this as a good entry and add 1 to the total of good entries
- If the letters are not asdf then count this as a bad entry and add 1 to the total of bad entries
- Add 1 to the number of groups received
- If 10 groups have been received then send the data about performance to the display
- If 10 groups have not been received then get another group of letters

Activity 4

Use the answer to Activity 3 to help you to produce a flowchart for implementing the task of the typing tutor. (Hint: you will not be able to use exactly the words shown above in some of the boxes of your flowchart. In particular, think carefully about how you will implement the sentences beginning with 'if' in the flowchart.)



Figure 5 Flowchart for Typing Tutor

The flowchart is given in Figure 5. Did you realise that the sentences that begin with ‘if’ need to be implemented as decision boxes? One of these starts two branches (depending on whether ‘asdf’ has been typed correctly) and the other a loop (to ensure that ten groups of letters are examined).

You have now seen several examples of how flowcharts can be used to describe tasks carried out by computers. Drawing such flowcharts is just one stage in the process of developing an application program. You will look at this in the next section.

6 Developing an application program



The process of developing any software starts with an analysis of the task or tasks to be performed by the computer, an analysis designed to tease out how the computer is to behave under every possible circumstance. The software development process may continue through the drawing of flowcharts like the ones you have seen, or the software developers may prefer other means of arriving at an understanding of each individual element of the task and how these elements fit together. In either case, however, the next step will be to write the computer program. This will take the various elements of the task and convert them into a program by use of a programming language, which is a structured language with a limited set of words and symbols and which can be used to tell a computer how to perform a task. Finally, the program must be 'debugged' (made free of errors) through extensive testing, and also documented to facilitate any future work on it.

After this there may be more forms of testing, from user evaluation of the 'user experience' of using the software to the more dramatic, literally pulling the plug on the computer part way through executing the program to see ensure the program fails gracefully. Important if your program is running a bank's teller machine and fails part way through a user withdrawing money; potentially vital if your program is monitoring a safety critical system.

There may be yet more steps before your program can be used, such as user training in complicated applications. The computer hardware and its programs are just the first steps towards delivering an effective system.

Before you finish this session, you are going to take a short look at a special form of application programming: programming for the web.

7 The language of the web



Making the web appear in your web browser may be thought of as a special form of application programming. Typically, modern web sites divide this programming into three parts: content, style and behaviour.

The content of a web page is put together using hypertext markup language (HTML), the language of the web, which your browser understands. A browser needs a set of codes to identify the various parts of the web page: the headings, lists, paragraphs and so on. These codes are called tags in HTML and they make it possible for a web page to be displayed in a browser. HTML tags are standardised so that they can be interpreted by different browsers. HTML describes the structure of a web page, but how about formatting the page?

Well-designed web page presentation is vital for readability and appeal. The appearance of a page is greatly simplified by the use of stylesheets. A stylesheet is code used to format a page in a desired way: the presentation information goes in special `<style>` tags, often as a separate page of code. Indeed, Cascading Stylesheets (CSS) can control the layout of multiple web pages all at once and thus save a lot of work. A style can be added to HTML elements in three ways:

- Inline – literally inline with the HTML, hence formatting and describing the page contents at the same time, all mixed together.
- Internal – by grouping all the formatting commands in to one part of the HTML document into a stylesheet. This simplifies applying the same style across a document, because you need only define it once.
- External – by using an external CSS file, to which each page can link. Now you can apply a common style to all the documents in your website.

Just as we can separate content (or structure) and formatting (or presentation) by putting the CSS in a separate file from the HTML web page, you can separate content from how the page behaves, how it responds to the user. Actions, such as validating a user email, is generally performed using the JavaScript programming language. You can use flowcharts to design a JavaScript program. Just as stylesheets can be linked to from HTML pages, so can JavaScript files. Hence, the one script can serve on many pages, which means you need only write the code to check for a valid email address once and then call it whenever you need it from any page in your website.

There are several other special application languages, such as Structured Query Language for working with databases. You will encounter these if you choose to take your study of computer programming further.

8 Summary of Session 6

In this session, you have looked at how a processor is given instructions. There are two broad categories of computer programs: application software that enables a computer to perform specific tasks, and operating systems that manage the sharing of resources amongst the various programs.

In flowcharts, you saw one common technique of capturing the breakdown of a large task into smaller steps that can be written as a computer program.

Finally, you were introduced to a couple of special purpose computer languages in HTML and CSS, that are used by web browsers.

In the next session, you will look some more at the internet.

You can now go to [Session 7](#).

Session 7: Networks of computers

Introduction

In this session, you will explore how we can progress beyond individual computers and join them together to form networks of computers. You will look at some of the technologies that make this possible, how computers have been linked together to form the internet, and how the internet itself is developing especially through wireless technology to become the ‘internet of things’.



1 Technologies

In the following short sub-sections, you will be introduced to network technologies. Firstly, you will be introduced to the building blocks that are used to create a network, and then at how various networks can be built from them ranging in size from a small one such as have in your own home to the global world wide web.

1.1 An example network

The internet provides the backbone for worldwide communication. You will look specifically at the internet and the technology that makes it work in Section 2. The term internet derives from the fact that the internet is itself made up of many other joined-together networks. But the internet is not the only player in providing communication. For example, there are the suppliers of broadband, the internet service providers, mobile phone operators, Wi-Fi access points and local area networks at work, to name most of the important ones. It is the joining together of all these different networks belonging to these service providers that allows us to access our workplace or browse the web seamlessly – that is, without being aware of the underlying networks that support the service.

Case study: A home office set up

Figure 1 shows one possible representation of the joining together of networks. In this example, a home network, a campus network and a branch office network are all joined together using the internet.

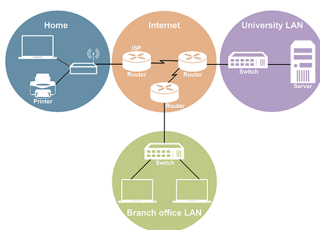


Figure 1 Home-to-office network.

This is a representation of my own home-to-workplace set-up. At home, I have a laptop and iPad which both connect to my home hub using Wi-Fi. My printer is plugged into the hub using an Ethernet port. The hub was provided to me as part of my contract with my broadband supplier and connects from my home to fibre-optic broadband. My broadband link will eventually get me to my internet service provider (ISP), where I will be connected to the internet through a router, called a gateway router in this location as it acts a gateway between two networks – in this case, the ISP and the internet. A connection is established across the internet to get me to another gateway router at the edge of the Open University network.

In this example, I wish to access a server at my office, which is shown at the bottom of the diagram. I have to pass through a hierarchy of network devices (routers and switches) to reach the server as the OU network supports over 3000 staff and many thousands of students. I would follow a similar pathway if I wanted to exchange emails with a colleague.

Connections can be made in a similar way to a branch office. At both the University and the branch office, the collection of networking equipment is joined together using a **Local Area Network (LAN)**. As an alternative to using the internet, the part in the middle of the diagram, which connects the university to the regional office, can be provided over privately leased links, when it then is referred to as a **Wide Area Network (WAN)**.

In summary, my connection from home to office (university) has been made possible by joining together my home network, a WAN and a LAN: a network of networks.

Now you'll look in more detail at these technologies.

1.2 Routers and switches

Routers and switches are specialised computers that handle communications over networks.

As a broad generalisation, routers connect together networks, switches connect together end devices.

As routers handle connections between networks they are commonly found in the internet, whereas switches handle connections in LANs. (LANs are explained in the next section.) They both have the same function of passing on your data to another device, but they do that function on different levels.

Sometimes the distinction is not always so clear. Your home hub is an example of a router. It manages the connection outside your house to a local cabinet, a roadside box, which is part of the larger telecommunications network.



Figure 2 A local cabinet.

Yet your home hub also connects devices within your home, such as a printer in the previous case study, and hence is also acting as a switch. In this case, the two types of connections are combined in the one physical box. As the main purpose of the box is to connect outside your home to the internet, it is generally referred to as a 'router'.

In large organisations however, when many thousands of devices need to be connected such as on the Open University's campus, the switches and routers are distinct devices. Indeed, there are so many devices to connect, local networks within the University, that there are rooms full of switches and routers stacked into cabinets as can be seen in Figure 3.



Figure 3 Plugged-in cables.

In Figure 3 the pink and green cables connect devices to the switches. The grey cables are connected to the routers. Typically, switches can connect many devices, while routers connect a few networks. This figure shows you the different number of ports available to plug in cables in the two types of device.

Let's look at the types of networks that switches and routers connect.

1.3 Local Area network (LAN)

A local area network (LAN) is a computer network that connects devices within a limited area such as a house, school, university or office. Ethernet and wi-fi are the two most common technologies used in LANs nowadays.

It is possible that an organisation is so big that it will have many LANs in its buildings.

Activity 1

Why might an organisation want more than one local area network?

Provide your answer...

Discussion

There are several possible answers. One very large network might not even be possible, because there are simply too many devices to connect. However, even if it is possible to connect many devices, large networks can be hard to manage. Typically networks will either match an organisation's structure with one LAN per department, or match the buildings physical structure with one LAN per floor. There may be particular requirements too so that the additional security for departments handling sensitive information is easier to apply or laboratories generating large amounts of test data do not overload the normal business network.

Devices have a network interface card that allows them to connect to a LAN. The term 'devices' can include computers as well as specialised input devices, such as scanners, and output devices such as printers.

1.4 Wide Area Network (WAN)

Organisations of all sorts – commercial as well as academic and governmental – have LANs on their premises, and many of them connect their LANs together to create an organisation-wide network referred to as a wide area network (WAN).

Connecting an organisation's LANs across many sites will involve leased telecommunication lines, in the same way as you lease a line to connect to the internet from your home hub router.

There are many related acronyms that have been devised over the years to describe various network configurations. One such example is Metropolitan Area Network (MAN), in which a city or district, has a network within its boundaries to provide connectivity. However, whatever acronym is used, they are only variations on either a LAN or a WAN. In the example of a MAN, that is a WAN, one usually operated by the local government of the city or district to enhance the business opportunities in its area.

You should be aware that when we say 'wires' when discussing networks, the 'wire' may not be wire at all! For the short distances involved in LANS, generally copper wire is used, but for WANS to meet the challenges of greater distance and the greater amounts of data they have to transmit, other technologies are used, such as optical fibres. These technologies are hidden from the user. One network technology, however, is not, and that's the absence of wires altogether.

1.5 Wireless networks

Wireless networks are increasingly taking over from wired networks. Similar connecting devices are still needed, it is only the data transmission that is changed.

The change to wireless communication has been driven by the development of the technology for mobile phone networks, and the evolution of the phone into the smartphone. A smartphone is a very portable computer, and that is why you can check your email or watch videos on your smartphone as easily as you can on a PC.

Sometimes wireless communication is used even where no mobility is required, because it is generally easier and quicker to install a wireless link than to install cables, and/or because there are specific obstacles that make it difficult to install a cable. Thinking back to the home hub mentioned earlier in this session, that is a wireless transmitter, as well as a switch connecting your devices into a home network, and it is also a router connecting your home network and devices to the internet; all without you have to recable your house.

The advantages of wireless communication are not confined to the home. In cities, for example, fixed microwave links are sometimes used for WANs and MANs, because they can be created by installing two antennas pointing at each other instead of having to dig up the streets in order to lay a cable.

2 The internet

The **internet** has probably had more impact on our daily lives than any other technology emerging from the information age. Most people, however young or old, will have heard of and used the internet, although it may mean different things to different people. For some, it is the means of accessing the web; for others, an essential part of their work; for some, an important tool for keeping in touch with friends or relatives; and for some, something not well understood and even to be feared.

The terms internet and **World Wide Web** are often used interchangeably. It is common to speak of 'going on the Internet' when using a web browser to view web pages. However, the World Wide Web or the Web is only one of a large number of Internet services.

The World Wide Web is a global collection of documents, images, multimedia, applications, and other resources, connected by hyperlinks, and each with a unique Uniform Resource Identifier (URI), which provide a global system of named references. URIs identify services, databases, documents and resources.

Hypertext Transfer Protocol (HTTP) is the main access protocol of the World Wide Web. Web services also use HTTP for communication between software systems for sharing and exchanging data. It is one of many protocols that can be used for communication on the Internet.

Other uses of the internet are email, data transfer and even telephony. Each requires their own standard to operate, such as File Transfer Protocol (FTP) for data transfer.

The Internet itself is a global network that comprises many voluntarily interconnected networks. It deliberately has no central server. This means the internet has no single point of failure, but will keep working should any of the connected networks or devices fail.

The underlying technology and main protocols are governed the Internet Engineering Task Force (IETF). This is an international non-profit organization that anyone may associate with by contributing technical expertise.

Other bodies ensure standards such as Internet Corporation for Assigned Names and Numbers (ICANN) which administers the principal name spaces of the Internet. ICANN is governed by an international board of directors drawn from relevant technical, business, academic, and other interested communities.

2.1 Out of this world

The internet is a global wide area network. However, that is not the limit of humanities ambitions.

The interplanetary internet is a network in space. It is intended to connect devices on other planets, orbiters and landers, to each other and back to Earth. Once on Earth, the data can be shared as any other data over the internet.

A major problem with interplanetary communication is the delay caused by interplanetary distances. Hence, a new set of protocols and technology that can tolerate these delays, as well as the increased rate of transmission errors and frequent disconnections, are required.

The next section brings you back closer to home.

3 The Internet of Things

Definitions of the **Internet of Things** (IoT) vary, but generally they express the idea of devices exchanging information with other devices, usually but not always over the internet. At its heart, the Internet of Things is about machine-to-machine communication.



Figure 4 Who needs cash when you have a credit card reader and the shop has a credit card reader that will talk to your bank and to their bank?

Some IoT applications are already well established. Credit-card readers in shops are very familiar. A more recent example is the so-called smart meter, such as an electricity meter that uploads its own readings to an electricity company, removing the need for a meter reader to call occasionally.

IoT is expected to become really innovative only when it is supported by automated, 'intelligent' processes. With the addition of automated intelligent processes, IoT could transform manufacturing, commerce and home life – according to some of its more zealous proponents. In fact, the term 'Internet of Things' is often shorthand for a new era of technological progress in which familiar objects and services, such as household utensils and appliances, cars, public transport, refuse collection, and so on, become 'smart'.

Whether the Internet of Things turns out to be as transformative as is sometimes claimed remains to be seen, but the concept of IoT has provoked a lot of interest in the worlds of research and business.

4 Summary of Session 7

In this session, you have explored some of the technologies that enable computers and computer systems to connect, and at the networks built on those technologies. This session concluded with a glimpse at two emerging technologies, the interplanetary internet and the Internet of Things.

The next and final session of this course, will look at some other future developments affecting the world of computing.

You can now go to [Session 8](#).

Session 8: A look to the future

Introduction

In this session, you will explore something of the future of computer systems. The future is a very wide topic, so you will focus on just two aspects. In the first you will look at the increasing use of computers in all manner of ways in our daily lives. In the second, you will look at the impact of computers on the world at large.



1 Pervasive computing

You'll begin by thinking about to what extent computers impact on your day-to-day life.

Activity 1

Before you go any further, think back to what you did yesterday and spend a few minutes making a brief list of any computers you interacted with.

Provide your answer...

Discussion

By now, you should be thinking of computers as more than the typical machine equipped with a monitor, keyboard, and so on. Your mobile phone is a computer that supports making phone calls, as well as many other functions. Your TV is a computer specialised for the display of audio and video programmes. Usually these programmes are broadcast, but perhaps your TV is a Smart TV and connected directly to the internet to stream programmes. You might think that not so different from the idea of a 'typical computer'.

If you went out, did you pay for any purchases with a cashless card? Or use a swipe card to enter a building or access a service.

If you stayed in, did you cook something in your microwave oven? Or wash clothes in your washing machine?

All the machines listed above are controlled by computers. There are computers *all around us* as we saw earlier in this course, they are embedded computers, and integrated into this world around us. In that sense, they are **pervasive computers**. They pervade everything we do.

Next you'll look some more at these pervasive computers.

1.1 Ubiquitous computing

The term ubiquitous computing describes the idea that computers are becoming pervasive – that is, they are integrated into the world around us. The term was coined by Mark Weiser at the Xerox Palo Alto Research Center (better known as Xerox PARC).

Ubiquitous computing names the third wave in computing, just now beginning. First were mainframes, each shared by lots of people. Now we are in the personal computing era, person and machine staring uneasily at each other across the desktop. Next comes ubiquitous computing, or the age of calm technology, when technology recedes into the background of our lives.

(Weiser, 1996)

In other words, we each find ourselves interacting with a number of computers, each of which may itself interact with other people and other computers. Often we are not even aware of where data is coming from or where it is stored – and more than this, we're often

unaware that we are interacting with computers at all. **Ubiquitous computing** is clearly already with us. It is in the background of our lives, or as Weiser says, it's **calm technology**.

1.2 Calm technology

Calm technology is really about technology not being intrusive, thus allowing us to relax (feel calm) in its presence. This doesn't mean the technology has to be invisible, but it shouldn't give (or ask for) more information than is necessary. However, if we don't want our daily lives to be constantly interrupted, then we have to be prepared to allow data exchange to take place – and this may have consequences, not least for data security. Some people would be very happy for technology to be almost invisible, but this does imply a certain loss of control; the technology has to make decisions that would otherwise have been yours.

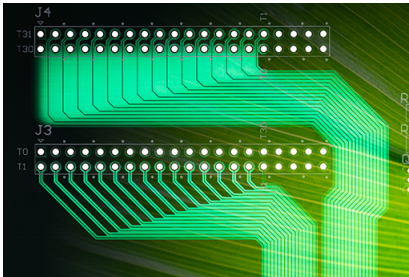
1.3 Ethical implications

As computers recede into the background of our lives, it is possible to forget about their existence. But think back to the opening activity in this session. Perhaps you left a digital trail of your movements and actions as you interacted with those computers. Where did you shop? Where did you go? How did you travel? Companies want to harvest that data to target their sales to you. Perhaps you give them explicit permission to do that, sometimes that permission is implicit just by choosing to use their service. Who else might access that data? And to what purpose?

The ethical aspects of computing and computing systems are far beyond this introductory course, but increasingly the question of how we use computer systems is becoming the focus of 'computer' research in place of examining computer technology itself.

The social consequences of computing technology on us is not the only concern about the increasing use of computer systems. In the next section, you'll look at another consequence, that on our world at large.

2 Green computing



Green computing is the environmentally responsible and eco-friendly use of computers and their resources. It is concerned with all aspects of computing: the design, manufacturing and disposing of hardware; software and hardware that is designed to be run efficiently to cut down on processing power; and working practices. Though there is a growing focus on the energy needed to power the ever-increasing number of computers. Some understanding of computing's energy consumption can be found in the 2017 report '[Clicking Clean: Who is Winning the Race to Build a Green Internet](#)' which was produced by *Greenpeace*. Here is their take on the comparative demands of different areas of the IT sector in 2012 and in 2017.

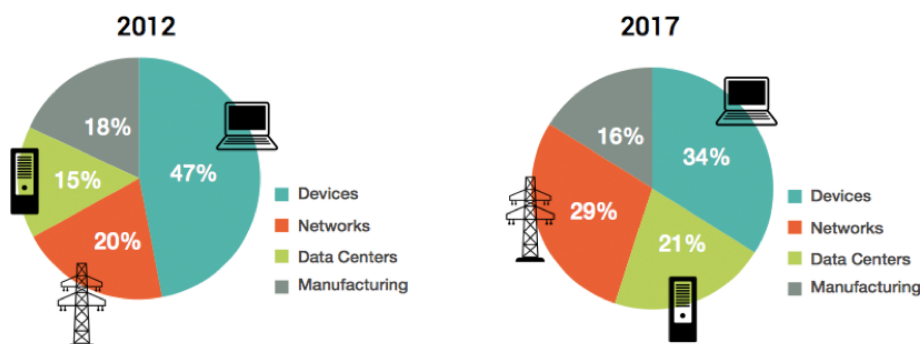


Figure 1 Main components of electricity consumption for the IT sector.

Figure 1 shows that within the IT sector, the fraction of energy consumed by individual devices is less in 2017 than 2012, as is the fraction of energy used to manufacture IT hardware. However, the percentage of energy consumed by data centres and networks is greater in 2017 than 2012. We can't be sure that this is as a result of the move to making more use of online services hosted in data centres, but it seems reasonable to assume that this is an important factor. The report also points out that the IT sector accounted for 7% of global electricity demand in 2012, but could be as high as 12% in 2017.

Activity 2

Find the current figures for the power consumption of computers and computer systems. Compare them with these historic figures.

Have the trends between 2012 and 2017 continued? Has the proportion of energy spent on manufacture continued to decline, while the proportion of energy consumed by data centres continued to increase?

Provide your answer...

Discussion

It is assumed that even if individual computers are more efficient, because there are more computers, the overall demand for energy has increased. Is this right?

Have the trends in changing energy consumption between sectors continued? Would you say that they are changing at the same rate, or faster, or slower?

In the following section, you will look more closely at one aspect of green computing.

2.1 Carbon emissions

A particular concern of green computing is the carbon footprint arising from the energy demands of computing. As you saw earlier, one of the claimed environmentally friendly benefits for cloud computing is that it reduces your carbon footprint because you only use the services when you need them. You can share the computing resources as you don't need them all of the time. While that may be true, it is also important to know what powers cloud computing's data centres.

Greenpeace produce a regular report that gives IT companies a mark for their commitment to green energy. Figure 2 shows the entry for Amazon in 2017, which records that they received a 'could do better' grade of C. This is not only based on the mix of energy sources they are using (17% renewable, 30% coal, 24% natural gas and 26% nuclear energy), but also on factors such as how transparent they are about their energy use, their commitment to renewable energy sources and their overall efficiency in terms of total energy consumption and reducing greenhouse gas emissions.



Figure 2 The proportion of different energy sources used by Amazon Web Services.

There are many other factors to take into account with data centres energy efficiency beyond how much energy they consume and where the energy comes from. You can read find out more for yourself about this topic, such as the importance of keeping the large number of servers and telecoms equipment cool. Here are two useful search terms that you may not think of: 'power usage effectiveness' and 'water usage effectiveness'.

Box 1

Power Usage Effectiveness (PUE) is the ratio of the total amount of energy used by a computer data centre facility to the energy delivered to computing equipment. A high PUE indicates that a large proportion of the power supplied to the data centre is being used for non-computational purposes, such as cooling and lighting, rather than being used directly to power the IT equipment.

Water Usage Effectiveness (WUE) is the ratio of the annual water consumption (in litres) to the energy delivered to computing equipment (in kWh). The higher the WUE value, the more water is needed in order to cool the IT equipment, and the less efficient the data centre is in terms of water usage.

Finally in this section, the discussion will be brought closer to home.

2.2 Home working



Please note that this course was written before sufficient research was published on the effects of homeworking due to the COVID-19 pandemic and enforced lockdowns.

An argument can be made that if your businesses' computer systems have been moved to the cloud then the individual workers can access those computer systems as easily from home as they can from the office, and that will reduce the businesses' carbon footprint.

Activity 3

While working from home can reduce a business' carbon footprint, does it really reduce the overall amount of carbon emissions? What do you think? What are the effects of working from home on carbon emissions?

Provide your answer...

Discussion

The Carbon Trust (2014) points out that it is not always true that working on the cloud from home reduces overall carbon emissions. Although the carbon footprint of the business may decrease, overall carbon emissions might increase because individuals working at home use more total energy in lighting and heating than the same number of people working in an office. This increase in emissions may be compensated for by the fact that these individuals are not travelling to the office, but as the graph in Figure 3 shows, the 'tipping point' depends on how a person commutes to the office, and how far.

Tipping points

Home working is only green for commuters who travel this far daily

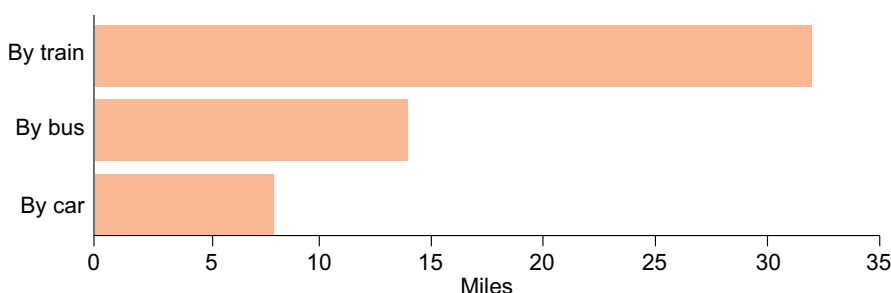


Figure 3 Home working does not always reduce carbon emissions according to the Carbon Trust.

Box 2

There is more to say on green computing and the changing nature of the services delivered by data centres in which more of their work is directed at storage and internet traffic than in computing power, and on the improving efficiency of all technology in data centres and its effects on power usage in the future. However, the latter paper was published in February 2020, just as COVID-19 spread around the world. With the dramatic changes in working practices and demands on data centres, previous assumptions about future trends are no longer valid. We await with interest future research that will follow up articles such as 'Microsoft, Google, Slack, Zoom et al struggling to deal with a spike in remote tools thanks to coronavirus' and 'We analysed electricity demand and found coronavirus has turned weekdays into weekends'. Until we are able to revise the course to include this future research, you may want to search for more articles on 'green computing' to bring yourself up to date with this important topic.

Resources

1. Cisco, 'Cisco Global Cloud Index: Forecast and methodology, 2016–2021 white paper' (Cisco, document 1513879861264127, 2018).
2. Masanet, E., Shehabi, A., Lei, N., Smith, S. and Koomey, J. 'Recalibrating global data center energy-use estimates', *Science*, 367 (6481), pp. 984–6.
3. [Microsoft, Google, Slack, Zoom et al struggling to deal with a spike in remote tools thanks to coronavirus](#)
4. [We analysed electricity demand and found coronavirus has turned weekdays into weekends](#)

3 This session's quiz

You now have the opportunity to try another quiz in which you can test your learning.

[Session 8 quiz](#)

Open the quiz in a new tab or window and come back here when you've finished.

4 Summary of Session 8

In this session, you have looked at two of the issues confronting our use of computers into the future. In the first you looked at pervasive computing and the increasing use of computers in our daily lives. In the second, you looked at green computing, and the demands that our use of computers make on resources.

You've now come to the end of this course. Thank you for coming this far. The next section is a short wrap up of the material you have covered

5 Course summary

Congratulations on completing the course. You should now be able to:

- visualise the room-filling early computers and appreciate how the technology has changed to achieve the computers of today
- consider how many computers there are in the world today, how they pervade our lives, and how computer systems come in many different forms
- understand how computers can be networked together to provide additional functionality
- identify the components of a computer, from its processor to its input and output devices
- understand how the world we live in with text, numbers, images and sound can be represented as data for a computer to process
- begin to see how instructions can be written for a computer to process data
- consider the energy implications of the increasing use of computers.

Next steps

If you have enjoyed this course you can find more free resources and courses on [OpenLearn](#).

Why not find out more about studying and gaining qualifications at The Open University? Visit [the OU prospectus](#) for more information.

At the OU, you can study for a qualification on the [BSc \(Honours\) Computing and IT](#), which includes the following introductory courses:

[Introduction to computing and information technology 1 \(TM111\)](#)

[Introduction to computing and information technology 2 \(TM112\)](#)

[Technologies in practice \(TM129\)](#)

Glossary

Computer

A machine that manipulates data following a list of instructions that have been programmed into it.

Computer program

The list of instructions the computer follows to process input and produce output.

Input device

A component that can function both as an input and as an output device.

Internet

A global network of connected networks.

Output device

Components that present data from a processor.

Smart devices

Devices that can communicate directly with other devices.

World Wide Web

A global collection of resources accessible over the internet.

Binary

A system limited to having just two values.

Computer bus

The internal data connections across the input and output subsystems and the secondary memory subsystem to the computer's processor and main memory.

Computer system

Formally a processor and its associated devices to make a usable 'system', but often the complete system, is referred to as a 'computer'.

Digital

In our context, it means systems using discrete rather than continuous values. In the larger context of computers in society, there is also the social and marketing use of 'digital', where it represents a way of engaging with people.

Embedded computer system

Specialised processors acting as controllers for devices such as washing machines and microwave ovens.

Hard disk

A form of secondary memory consisting of one or more rigid magnetic disks rotating about a central axle. Also known as a hard drive.

Hardware

The physical components of a computer or computer system.

Input-output device

A component that can function both as an input and as an output device.

Input-output subsystem

The components of a computer system responsible for the retrieving, transforming and presenting the data used by a processor, functioning as both an input subsystem and an output subsystem.

Input subsystem

The components of a computer system responsible for the entry of external data and its transformation into a form the processor can use.

Main memory

Where a processor stores instructions and associated data for execution.

Microprocessor

Was introduced when processors were first made on a single silicon chip, with the prefix 'micro' emphasising their small size

Output subsystem

The components of a computer system responsible for the transforming and presenting the data used by a processor into a form for use outside the computer system.

Secondary memory

Holds programs and data that will persist after the computer is switched off.

Secondary memory subsystem

The components of a computer system that prepare data for and retrieve data from secondary storage.

Software

The program, or collection of programs, that enable a computer to carry out its tasks.

ASCII

American Standard Code for Information Interchange, pronounced 'askee', a method of encoding text in binary digits.

binary code

The representation of an item of data using only two values

bit

A binary digit, which can be either one or zero.

byte

A group of eight bits.

Embedded computer

Computers that are part of another device dedicated to delivering one task.

Application software

The program that enables a computer to perform certain tasks.

Embedded computer

Computers that are part of another device dedicated to delivering one task.

Flowchart

A diagram showing a sequence of actions using specifically shaped symbols.

Operating system

Software that controls the resources and software using a computer.

Internet

A global network of connected networks.

Internet of things

Devices exchanging data directly with each other over the internet.

Local Area Network (LAN)

Connects devices within a limited in one network.

Wide Area Network (WAN)

Connects LANs into a larger network.

World Wide Web

A global collection of resources accessible over the internet.

Calm technology

When technology recedes into the background of our lives so that it does not impinge on us when we interact with it.

Green computing

Is the environmentally responsible and eco-friendly use of computers and their resources.

Pervasive computers

Embedded microprocessors in day-to-day objects, allowing them to communicate data, often used interchangeably with ubiquitous computing.

Ubiquitous computing

Or 'ubicomputing' describes when computing is anytime and everywhere, in contrast to desktop computing.

References

Johnson, G. (2004) *A Shortcut through Time, The Path to Quantum Computing*, London, Vintage Publishing.

Carbon Trust (2014) 'Homeworking: helping businesses cut costs and reduce their carbon footprint'. Available at:

<https://www.carbontrust.com/news-and-events/insights/could-homeworking-save-money-and-cut-carbon-emissions-for-your-business> (Accessed: 12 June 2021).

Weiser, M. (1996) 'Ubiquitous Computing'. Formerly available at: www.ubiq.com/hypertext/weiser/UbiHome.html (no longer available).

Acknowledgements

This free course was written by David King.

Except for third party materials and otherwise stated (see [terms and conditions](#)), this content is made available under a

[Creative Commons Attribution-NonCommercial-ShareAlike 4.0 Licence](#).

The material acknowledged below is Proprietary and used under licence (not subject to Creative Commons Licence). Grateful acknowledgement is made to the following sources for permission to reproduce material in this free course:

Every effort has been made to contact copyright owners. If any have been inadvertently overlooked, the publishers will be pleased to make the necessary arrangements at the first opportunity.

Introduction

Images

Course image: © New Africa; Shutterstock.com

Week 1

Images

Figure 1: © US Federal Government

Figure 2: © Good, Jack; Michie, Donald; Timms, Geoffrey (1945) *General Report on Tunny: With Emphasis on Statistical Methods*, UK Public Record Office HW 25/4 and HW 25/5

Section 3 figure: © geralt; Pixabay

Section 5 figure: © Oscar Wong; Getty Images

Week 2

Images

Introduction figure: © CreativeCore; Shutterstock.com

Section 2 figure: © MirageC; GettyImages

Section 3 figure: © carlos castilla; Shutterstock.com

Figure 2: © AddyTsl; Shutterstock.com

Figure 3: © patruflo; Shutterstock.com

Figure 4: © MartinPrescott; GettyImages

Section 3.2 figure: © ESB Professional; Shutterstock.com

Section 4 figure: © Daniel Krason; Shutterstock.com

Week 3

Images

Introduction figure: © Sashkin; Shutterstock.com

Figure 1: © ka_ru; GettyImages

Figure 2: © Anzay; Shutterstock.com

Section 3 figure: © photoinnovation; Shutterstock.com

Week 4

Images

Introduction figure: © Sashkin; Shutterstock.com

Section 2 figure: © extradeda; Shutterstock.com

Section 7 figure: © JAKKRIT SAELAO; Shutterstock.com

Week 5

Images

Introduction figure: © Den Rozhnovsky; Shutterstock.com

Figure 2: © Anton Belo; Shutterstock.com

Figure 5: © jeafish Ping; Shutterstock.com

Week 6

Images

Section 3 figure: © goffkein.pro; Shutterstock.com

Section 6 figure: © WOCinTech Chat; <https://creativecommons.org/licenses/by/2.0/>

Section 7 figure: © atm2003; Shutterstock.com

Week 7

Images

Introduction figure: © Jackal Pan; GettyImages

Figure 2: © Keith Ryall; Shutterstock.com

Figure 3: courtesy of Nicky Moss

Figure 4: © Javier Zayas Photography; GettyImages

Week 8

Images

Introduction figure: krisanapong detraphiphat; GettyImages

Section 2 figure: MirageC; GettyImages

Figure 1: Taken from: <https://green.googleblog.com/2013/08/the-latest-on-our-carbon-footprint.html>

Figure 2: 'Clicking Clean: A Guide to Building the Green Internet' (May 2015)
Greenpeace Inc.

Section 2.2 figure: fizkes; Shutterstock.com

Don't miss out

If reading this text has inspired you to learn more, you may be interested in joining the millions of people who discover our free learning resources and qualifications by visiting The Open University – www.open.edu/openlearn/free-courses.