Fundamentals of Computer Systems 2

Getting to know your unit

Assessment
You will be externally assessed by a written examination.

The fundamentals of computer systems should be known and understood by every computing professional. This knowledge of how and why computer components, and the data they use, perform in certain ways is a tool used in the many varied roles of the computing industry. Understanding how different parts of a computing system work together enables us to identify and solve problems. Computer programmers use their understanding of how computers operate to develop better software.

This unit will introduce you to the various systems in current use, including their component parts, how they run software apps and handle data, and the methods used to transmit data. It will also consider the impact of these on organisations and individuals.

How you will be assessed

This unit is assessed by a written examination set and marked by Pearson.

The supervised assessment period is a maximum of 1 hour 45 minutes. You will be assessed on your knowledge and understanding of how computer systems work, including the role of hardware and software, the way components of a system work together and how data in a system is used. The number of marks for the unit is 80.

Grade descriptors

To achieve a grade you are expected to demonstrate the following across the essential content of the unit. The principle of best fit will apply in awarding grades.

Level 3 Pass

You are able to apply knowledge and understanding of key computing concepts to a range of familiar vocational contexts.

You are able to use knowledge of computing to deconstruct problems within common situations and apply standard conventions to produce solutions with interpretation.

You can identify the impact of effective and ineffective computer systems and recommend ways in which a system can be developed and/or improved (using given structures and criteria).

Level 3 Distinction

You are able to analyse complex information, data and situations, within vocational contexts, in order to draw conclusions and make valid observations.

You are able to synthesise knowledge and understanding of computing to deconstruct problems, drawing on various sources of information to develop effective solutions with justification.

You are able to evaluate the effectiveness of computer systems to make justified recommendations on their development and future actions that can be taken.

Assessment criteria

AO1 Demonstrate knowledge and understanding of computing facts, terms, standards, concepts and processes

Command words: complete, draw, give, identify, name, state

Marks: ranges from 1 to 5 marks

AO2 Apply knowledge and understanding of computing facts, terms, standards, concepts and processes to real-life scenarios

Command words: calculate, complete, demonstrate, describe, draw, explain, produce

Marks: ranges from 1 to 5 marks

AO3 Select and use computing technologies and procedures to explore likely outcomes and find solutions to problems in context

Command words: calculate, demonstrate, develop, explain, produce

Marks: ranges from 1 to 6 marks

AO4 Analyse and evaluate data, information, technologies and procedures in order to recommend and justify solutions to computing problems

Command words: analyse, demonstrate, discuss, produce, write

Marks: ranges from 6 to 12 marks

AO5 Make connections between the application of technologies, procedures, outcomes and solutions to resolve computing problems

Command words: evaluate, produce, write

Marks: ranges from 6 to 12 marks

Getting started

There are many different types of computer systems in use today. What computing devices do you own or use? What are their best and worst features? How would you improve them? What will you eventually replace them with?

A Hardware and software

Every computer system comprises both hardware and software. Hardware is the components you can physical touch. Software is less tangible, and includes the operating system and programs that control the computing system as well as data stored on the system.

Computer hardware within a computer system

Computer hardware consists of anything in the computing system that is not software, that is, anything physical, including the computers, screens, keyboards, mice, printers, cabling and other mobile devices and servers.

This section will look at the types, features and implications of choosing particular hardware, including their internal components.

Types of computer system

There are a number of different types of computer system, which we will look at in this section.

Multi-functional devices

Multi-functional devices are those that perform multiple functions, such as printers that include a built-in scanner. The scanner is often on the top of the device so that it can also be used as a photocopier. Therefore this one device can print, photocopy and scan documents when attached to any computer connected to it over a network or through a cable. Such multi-functional devices often include fax circuits so they can send or receive images to or from another fax machine through a telephone line.

Mobile devices are also multi-functional devices because they often incorporate into the one device a phone, a camera and computer.

Personal computers

Personal computers (PCs) are widely used for work and by individuals at home. The work carried out on PCs includes anything useful to an organisation and normally requires the use of software programs such as word processors, spreadsheets, databases, programming environments or other software apps. PCs can be broadly divided into desktops and laptops. Desktop PCs are usually left in one place, either in an office or in a home office, and often have a cabled connection to a network. Laptops can be moved to different places within a wireless, WiFi network. Laptops are useful for individuals who want the flexibility to use their device anywhere in their home or while they are away from home.

Mobile devices

Mobile devices are multi-functional, as they not only consist of a computer, but often include a camera and, in the case of smartphones or smartwatches, they are also phones. Mobile devices are often small enough to fit into your hand. Tablets and smartphones are the most common mobile devices. There has been a lot of progress in the development of mobile devices and the latest products are almost as powerful as a PC and have good internet connection and a long battery life. The mobile devices that we now carry around in our pockets are more powerful than many PCs that were on sale just a few years ago!

Research

Ask your family and friends what mobile devices they have owned. Also find out in what year they used their first PC, when they first used printers, and how these devices and mobile devices have changed during their lifetimes.

Servers

Servers are computers that run a network operating system such as Microsoft® Windows® Server. They are used to control the network and the users logged into the system on workstations (which are also known as client computers). Servers are available in a variety of sizes to meet the needs of the smallest and largest organisations. They are powerful computers that usually have lots of memory and disc space and often have duplicated internal components to make them more reliable.

The purpose, features and uses of internal components

Internal components are the parts inside a hardware computer device. In this section, you will look at the features and uses of internal components in the different types of computer system.

Multi-functional devices

Multi-functional devices have an internal component called a processor to control them. The processor is a specialised computer that connects to a touchscreen and the network via a WiFi or wired network adapter. The touchscreen allows users to carry out functions such as photocopying and scanning and to change internal settings such as the network connection.

A network connection can be cabled into a **RJ45** port on the multi-functional device. Many smaller multi-functional devices also have WiFi so that they can make a wireless connection.

Key term

RJ45 – this is the common standard for network ethernet cable plugs and the socket or port that the cable uses to connect networked devices together.

The scanner mechanism in multi-functional devices is usually flatbed, often with a sheet feeder so that multiple documents can be scanned together without the need to place each one individually on the device.

The print mechanism of multi-functional devices usually uses inkjet, laser or solid ink technology. Inkjet technology is popular for use in homes and small offices. Laser and solid ink technologies are a good choice for businesses or organisations where there is a high demand for printing. Larger multi-functional devices include features for double-sided printing and stapling.

Link

See Output devices, in 'The hardware used in computer systems' for more detail on these technologies.

Personal computers

Desktop and laptop PCs have similar internal components, although it is usually much easier to change or upgrade these inside a desktop PC as a laptop has most of its functionality built into the motherboard, which cannot be changed.

Motherboard

The motherboard is the main circuit board that connects all the components plugged into it, including the processor, memory (RAM), backing storage (also referred to as secondary storage) and external components such as the keyboard, mouse, monitor, printer and network.

Processor

The processor (central processing unit or CPU) is a large chip that plugs into the motherboard. This is the component that understands and actions the program code, by working out any calculations, **logical operations** or **relational comparisons**. The CPU has one or more control units to action the code and arithmetic and logic units (ALU) for calculations and logic.

Link

See 'The concepts of microarchitecture' for more detail on how the processor (CPU) works.

Key terms

Logical operations – these are carried out between binary numbers using AND, OR, NAND, NOR, XOR and NOT logical operators.

Relational comparisons – these are carried out between numbers or text using operators such as less than (<), greater than (>), equal to (=), less than or equal to (<=), greater than or equal to (>=) and not equal to (<>).

The CPU is a powerful chip that does a lot of work and so it gets very hot. There has to be a method of removing the heat, which is usually a heat sink and fan. The heat sink sits on top of the CPU and is made from metal with a large surface area to conduct heat away and the fan blows air over the heat sink to make the heat dispersal more effective. Laptop CPUs usually have heat pipes instead of a heat sink as these are flatter. Without a way of removing heat, a modern CPU would get so hot during operation that it would melt away in around a minute.

Memory

When programs and documents are opened from backing storage they are placed in the memory (random access memory, RAM). The RAM operates much faster than the backing storage, enabling you to have multiple programs and documents active at any one time. The CPU needs the speed of the RAM because backing storage is far too slow for the CPU to use directly. Anything in the RAM disappears when the computer is switched off, which is why you must save your documents regularly when open, so that they are copied to the backing storage.

Backing storage

Backing storage is needed to store programs and data when the computer is switched off. The size or capacity of the backing storage represents how much storage space there is inside the device and is measured in gigabytes (GB) or terabytes (TB). There are a thousand gigabytes in a terabyte (see Fig 2.1). There are two main technologies used for backing storage – hard disc drives (HDD) and solid state drives (SSD).

- HDD: HDD is the older technology which works by spinning a disc round inside the device. Data is stored on invisible magnetic tracks (rings) on the disc using read/write heads which move to the track then copy data from the track to RAM (read) or record data from RAM to the track (write). Data is copied from the track to the RAM when a program/document is opened. Data is recorded from the RAM to the track when an updated or new document is saved.

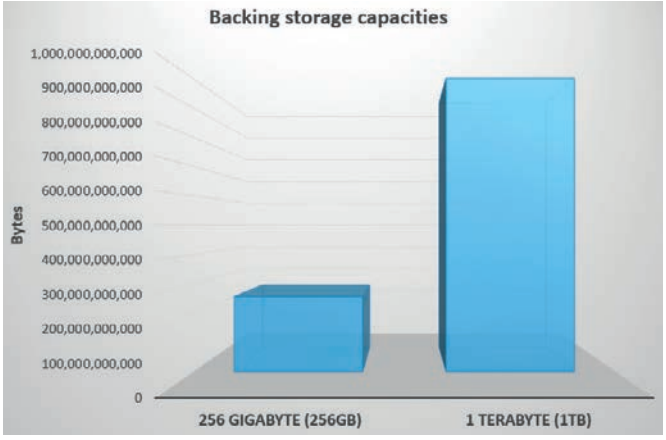


Figure: 2.1: Backing storage capacities

The performance of an HDD is influenced by the seek time, data transfer rate (DTR), spindle speed and cache size. These performance factors are explained in Table 2.1.

Key term

Cache – this is fast memory between a slow device and a quicker device that is used to prevent the quicker device from slowing down when the slower device is used. Processors contain cache, which is used for command queuing, to help program code run faster when commands are brought in from the slower RAM to the quicker CPU.

HDDs have RAM memory inside them that is used as cache to make the drive work more quickly by reading ahead or behind the current data so that the next data is already available (in cache) when needed. This technique allows speed matching of the slower to the quicker device. Write acceleration is also achieved as data can be quickly copied to cache then written to the slower disc.

The performance of an HDD can become poorer over time due to the read/write heads moving more between tracks as the data fragments. Fragmentation happens when files are deleted and the space is re-used for files which are of a different size and thus get saved to different parts of the disc. When this noticeably reduces performance the HDD needs to be defragmented (see Figure 2.2). SSDs do not need defragmentation as there are no read/write heads, so their performance is not slowed down by the need to move data to another part of the device.

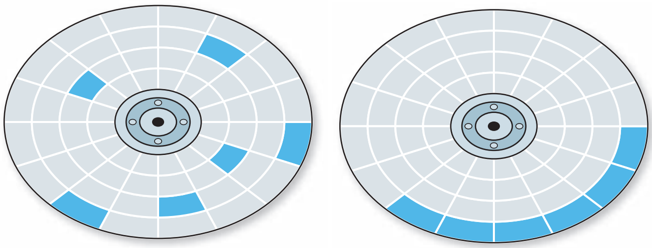


Figure 2.2: Before and after an HDD runs a 'defrag'

Table 2.1: HDD performance factors

Factor	Description	Units	Best performance
Seek time	The average time it takes the read/write heads to move to data in a track.	Milliseconds (ms)	Smaller seek time so finds the data quicker
Data transfer rate (DTR)	How quickly the data can travel to or from the HDD.	Megabytes per second (MBps) Gigabytes per second (GBps)	Larger DTR so transfers data faster
Spindle speed	How quickly the disc spins.	Revolutions per minute (rpm)	Larger spin speeds improve data transfer rates and seek times
Cache	Memory on the HDD to speed up data transfers.	Megabytes (MB)	Larger cache improves the speed on the HDD

- SSD: This newer technology uses electronic memory to store data, rather like the computer's RAM memory. SSDs use flash memory. This is non-volatile, which means it does not lose its content when power is removed (RAM memory on the other hand is volatile and loses its contents when power is switched off). Unlike the magnetic memory used in an HDD, an SSD has no moving parts, which has a number of benefits. The main benefit of using an SSD is that it is faster to access data since there is no wait for the disc to rotate or the head to move to the correct track as there is with an HDD. Data on SSDs can be directly read from any location and so they do not have a seek time, which gives them a much faster data access time than HDDs. This feature of SSDs means they do not suffer from the effects of data fragmentation either. With no moving parts, SSDs are shock proof and use less power than HDDs, making them more suitable for portable devices. The downside of SSDs is that they are more expensive than HDDs, and although prices have dropped since they were introduced, as of 2016 they are still around four times the price of a comparable HDD.

Both of these technologies use standard sizes of either 2.5 inches or 3.5 inches to allow easy fitting into PCs. The smaller size, 2.5 inches, is the usual size of drive used in laptops. The desktop connection between backing storage and the motherboard is usually through a **SATA** cable. The laptop connection is often the same, but the backing storage is plugged directly into the motherboard, rather than by means of a cable.

Key term

SATA – serial AT attachment (SATA) is the current standard for the cable connecting backing storage to the motherboard. This replaced the previous enhanced integrated drive electronics (EIDE) standard which was also known as ATA-2.

Discussion

With a peer or in a small group, discuss why an SSD is a better choice for a tablet than an HDD.

Mobile devices

Mobile devices differ from desktop computers in that size and battery life become important considerations. The main internal component of mobile devices such as smartphones is the **system on a chip** (SoC) which has the functionality of a motherboard and CPU. The RAM is usually separate to the SoC and **NAND flash memory** technology is used for the backing storage.

Key term

NAND flash memory – this kind of memory is used in SSDs, memory sticks and memory cards to hold documents, images and other files. This technology is non-volatile which means that it does not require power to store data. The name, NAND, is used to differentiate this technology from NOR flash memories. NAND and NOR are used to describe the algorithms used to store data rather than the type of logic gates used in their circuits.

System on a Chip – this is a single integrated circuit chip which combines all the functions of a computer. They are used in mobile systems because of their small size and low power consumption.

Other internal components inside mobile devices include the battery, long term evolution (LTE) modem, microphone, touchscreen display, sensors and camera.

Key term

Long term evolution (LTE) modem – mobile phones use long term evolution (LTE) modems to send and receive internet data.

Link

See 'The features and implications of embedded and mobile CPU architecture' for more about the internal components of mobile devices.

Servers

Server computers are often specialised versions of PCs with very large RAM, backing storage and powerful CPU(s). Many servers have redundant (duplicated) internal components so that, if one component fails, the redundant component can keep the system running. It is important that servers do not stop working, because they usually control and store data for many client computers across a network. Rack mounting is often used to keep a number of servers together in a cabinet.

Factors affecting the choice, use and performance of internal components

As a computing professional, you need to be able to recognise how the internal components of a computer system should be selected and the factors affecting this choice.

The performance of internal components is an important factor, but this has to be balanced with the user needs and cost. There is no point in paying for expensive and powerful hardware such as a gaming graphics card for a workstation that is only going to be used for basic office tasks.

Availability is another factor that should be considered. If you require a new computer system to be up and running now, in order that employees in a business or you at home can be productive, you will not want to wait for an internal component that is not expected to be on sale for another month. So internal components that are not available now would make poor choices for a computer needed this week. On the other hand, if you can wait, you might be able to get a higher specification component or the price of a specific component might come down.

Environmental concerns are increasingly important to consider as organisations recognise the need to 'do their bit' for the planet. Not only does considering environmental factors when choosing internal components help to improve the long term prospects of the planet, but can often save money and provide good marketing material for the manufacturer. Buying a PC with a more powerful CPU than is actually needed will use more power and cause it to get warmer than a CPU which is well matched to the intended workload, especially if this is from the most modern and up-to-date low-energy version of that processor. The reduction in electricity needed and air-conditioning costs for a large organisation using hundreds or thousands of these PCs could be significant.

The hardware used in computer systems

In this section, you will explore the hardware used in computer systems including input, output and storage devices.

Input devices

Input devices are anything used to control or get data into the computer system, so they may pass data to the computer but do not store data. The following are all examples of input devices that can be connected via a cable or wirelessly to a digital device.

- Keyboard – This can be connected to a computer, laptop, tablet or smart TV. Using the keys, users transmit instructions or input data into the digital device.
- Mouse – This is a pointing device that allows users to make selections on a computer screen.
- Webcam – This is a video camera that transmits moving images through an internet connection. Webcams make it possible to see friends and family on the other side of the world.
- Microphone – Used in conjunction with a webcam, a microphone allows us to transmit sounds across the internet or to give commands to the computer using our voices.
- Joystick – Similar to a computer mouse, a joystick allows users to make selections or navigate through a computer generated environment in a computer game.

Output devices

Output devices are used to get anything out of the computer system and provide information to the user from the computer. They perform a function using data or software stored in a computer. The following are all examples of output devices which can be connected via a cable or wirelessly to a digital device.

- Printers – Printers are used to reproduce text and images according to instructions they receive from any digital device. The printer receives data from the digital device, which it replicates on paper.
- Screens – A screen displays whatever software and/or data is being accessed on the digital device at any given time. They are crucial to human interaction with computers as, otherwise, we would not know what was stored on the digital device or be able to access the internet through it. A monitor is a peripheral device with a screen that is used in many computer systems.
- Projectors – These are similar to monitors as they are visual displays of whatever is being accessed on a digital device. Projectors are used to project the digital display onto a white or silver screen. They are mostly used to project films in cinemas (commercial and home) and for presentations in front of large groups of people (for example in schools or colleges) where there are too many people to gather around a small computer monitor.
- Speakers – Speakers convert digital audio signals into audible sound. They are used to listen to music or the soundtracks accompanying videos and enable users to hear people in other locations during web conferences.

Input/ouput devices

Some devices are both input and output devices.

- USB – this interface is used for a variety of purposes, including input devices such as mouse and keyboard, output devices like printers and storage devices like flash drives and external hard discs. USB has a number of versions, the current version being version 3. All the versions share the same physical connector and are backwards compatible, but the versions have progressively faster data transfer speeds with USB 3 being capable of up to 5 Gbit/s.
- Joysticks are generally input devices but you can get ones that have haptic feedback, which is a type of output. Haptic feedback is a tactile sensation that users feel in their hand in response to something happening in the software they are using. The most common use of haptic feedback is in computer games, as this enables users to feel sensations corresponding to things happening in the game, for example crashing into another car in a racing game.

Storage devices

Internal storage devices are components that are used to hold data and software for the normal running of the computer system.

- Hard disc drives – The HDD is an older technology, which is used where a lot of storage space is needed.
- Solid state drives – The SSD is the newer technology, which is used when the best performance is required (see the section on Backing storage for more detail).

External storage devices are peripheral devices used to hold data and software for the following reasons:

- to save space on your computer's hard drive
- to transfer files physically rather than over the internet
- to share files with another person or another digital device physically (if an internet connection is not available)
- to back up data in order to protect against damage or loss.

The following are examples of storage devices.

- External hard drives – These are used to back up data or to store large amounts of data. Digital devices such as laptops and tablets have limited storage space, so by storing data on an external hard drive you can increase your storage capacity.
- USB flash drives – These can store smaller amounts of data so that you can keep a copy of it or they can be used for temporary storage when moving data from one digital device to another.
- Server – This is a physical central location for storing and managing data between a number of networked computers.
- Cloud – This is a virtual server that is used for accessing and storing data via the internet.
- SD cards – SD (secure digital) cards provide high-capacity memory storage in a small space. They are most commonly used in small digital devices, particularly for storing still images.
- DVDs – DVDs (digital versatile discs) are a form of optical disc technology similar to CD-ROMs. They are most commonly used to store movies. Blu-ray discs are another form of optical storage designed for high definition video and have a higher capacity than DVDs.

Storage devices are often used as both input and output devices (known as I/O devices), meaning that you can copy or transfer the data from the device as well as onto it. Some storage devices will not allow you to reuse them, so you cannot change the data that is stored on them. These devices are read/write only.

Link

To learn more about how the features of hardware can affect their performance see:

- Table 2.1: HDD performance factors, also sections on backing storage, the purpose, features and uses of internal components and computer hardware in a computer system
- Table 2.2 Display performance factors, in this section
- Table 2.3 Printer performance factors, in this section
- Table 2.5: CPU performance factors, in 'Cache', in 'The concepts of microarchitecture'.

Discussion

With a peer or in a small group, identify as many types of input, output and storage device as you can.

How the features of hardware can affect its performance and the performance of a computer system

Most hardware components have features to improve the performance of both the component itself and the overall system. The overall performance of a system will be hampered by the slowest component part, so it is possible for one component to have an adverse effect on another. For example, a screen may not be able to display the high resolution produced by a high performance graphics card, so the performance of the graphics card would reduce down to the capabilities of the screen.

Display performance

Table 2.2 outlines the performance factors of display devices in computer systems.

Table 2.2: Display performance factors

Factor	Description	Units	Best performance
Refresh rate	How often the complete screen is displayed	Hertz (Hz). Most monitors have a 60Hz refresh rate.	A lower refresh rate will give you visible flicker and less smooth movement
Response time	A measure of how quickly each pixel of the display can change from one state to another (e.g. change of colour or brightness)	Milliseconds (ms)	Faster responses produce images with less trailing or ghosting which makes moving images seem blurred.
Brightness	The light output of a monitor	Candela per square metre, the larger the value, the brighter the monitor	The brighter the monitor the easier it will be to see in bright ambient lighting or sunlight.
Contrast	A measure of the difference between the light and dark areas of the image	A ratio, such as 600:1. The larger the first value, the higher contrast the monitor has	Higher contrast ratio will generally give you a higher quality image with the ability to distinguish between more colours

Printer performance

Table 2.3 outlines the performance factors of printers in computer systems.

Table 2.3: Printer performance factors

Factor	Description	Units	Best performance
Printing speed	How quickly pages can be printed	Pages per minute (ppm)	Larger number indicates a faster printer
First-out time	How long a delay before the first page starts printing	Seconds	Smaller number shows that the first page starts printing more quickly
Maximum printing resolution	How detailed the printed images can be	Dots per inch, often shown as across x down, e.g. 4800 x 1200	Larger numbers show a more detailed image
Monthly duty cycle	How many pages the printer can easily produce in a month	Pages	Larger number shows a more robust design

Factors affecting choice of hardware

There are many factors that determine our choice of hardware and these are discussed below.

User experience

User experience is the most important factor for many individuals and, to a lesser extent, for organisations. If the hardware does not give users a positive experience, then no one would choose it. Whether a user has a good user experience depends on whether the device meets their needs.

For example, an elderly person wanting to purchase their first mobile phone primarily for emergencies is likely to want a simple-to-use phone that makes and receives calls, with large keys, a clear screen and few, if any, other features. However, an individual looking to upgrade their smartphone will base their decisions on previous experience and may be looking for additional or enhanced features such as extended WiFi range, screen resolution and memory capacity. Other factors such as cost will also be a consideration for users and, while many sign up for monthly payment contracts, others may be wise to buy their phone on a pay-as-you-go contract.

There are number of aspects of user experience that will affect choice of hardware.

- **Ease of use** – Hardware that is easy to use or is more likely to save time is going to be preferred by most people, whether for personal or business use. Too many computer systems are overly complex and only a minor portion of those systems are used by most people.
- **Performance** – The level of desirable performance will vary depending on whether the technology is to be used for pleasure, business or academic research. The variety of functions that a device can perform is also a factor. For example, an elderly person may want a mobile phone for emergencies only so will not need particularly good performance and their choice will be based on a single criterion. However, a business may

base their choice on level of performance if it requires its employees to be contactable while on the move. Employees may also need to be able to access a central server for customer records, analyse financial data and produce complex reports. So businesses will make choices based on many criteria.

- **Availability** – Whether a hardware device is immediately available can influence our choices. We may make alternative selections if our first choice is out of stock. A business may buy on a large scale for multiple employees and cannot afford to wait for further supplies to become available. Many businesses require high availability server systems with redundant hardware so that single point failures do not bring the system down. Uninterruptable power supplies (UPS) may also be required so that power outages do not cause service loss.
- **Accessibility** – Accessibility or adaptability of technology may dictate hardware choices, especially if the user requires specialist devices such as touchscreens, ergonomic or light operated keyboards and mice, eye motion sensors, head motion trackers or sip-and-puff systems.

Research

Look at the different hardware and software systems than can be used by people with different disabilities.

User needs

Users will have different needs, which inform the criteria they use to make hardware choices. For example, your place of study requires computer systems that accommodate the needs of staff to use interactive whiteboards, printers and photocopiers. Computing students will use computers for their studies, and they are likely to need access to hardware such as printers and storage devices to save their assessments. Users also need to use their computers for different purposes. For example a user who needs a computer to do video editing will require a computer which has a powerful CPU, as this is a demanding application which also requires high capacity storage devices and a high performance graphics card. A different user who needs to run many web based applications at the same time might benefit from a PC with a multi-core CPU, lots of RAM memory and dual monitors, but may only require a hard disc of average size.

Specifications

The system **specification** will vary according to user needs. Software designers may require different types of system according to the tasks they are undertaking. For example, for programming, they might use fairly basic specification systems but they will probably want significant processing power and RAM, so they will require higher specification systems that provide these.

Key term

Specification – a description, often detailed, of the design of a hardware device. The specification gives details of the user interface that it uses, the processing power and amount of RAM, for example.

Compatibility

Compatibility means whether two or more things that need to connect and interact with one another do so properly. If they do not connect (physically in terms of cables or connectors) or interact with one another (work together), then they are incompatible. Decisions about choosing hardware for computer systems will be based on the needs of the whole organisation. Just as you might choose the same make for several devices out of personal preference and because you know they will work together, businesses are concerned about compatibility and whether the hardware components they choose will interact with each other correctly.

Cost

The cost of a computer system is very much a factor when making a choice about hardware. Businesses will allocate a budget for their computer systems. This budget is likely to include maintenance and upgrades. The cost varies according to the specifications and quality of manufacture of the hardware components. Cost is obviously an important factor for individuals and their budgets are usually much smaller than those of organisations.

Efficiency

The **efficiency** of a computer system must reflect user needs and is determined by the quality of the hardware components used. If a business is operating with low specification computer systems which under-perform, it will take employees longer to complete tasks and this will cost the business money. To achieve high efficiency, hardware needs to perform well with low power consumption.

Key term

Efficiency – a measurement of the useful output of a task or activity. A high efficiency means little wastage of time, materials or labour.

Implementation

Choosing hardware for a computing system can depend on the speed of its implementation. For example, individuals and organisations are likely to continue to use hardware which operates in the same way as their current hardware because this avoids unnecessary loss of time for training. Hardware for a new computing system will be chosen for performance and perhaps to save space, increase mobility and reduce weight.

The urgency for implementing computing systems will influence hardware choices.

- Timescales – The preferred computing system (and its component pieces of hardware) may not be readily available or the lead time for implementation may be too disruptive for the organisation or individual. Suppose that your printer stops working and you want to upgrade to one with greater performance but cannot get the same make because it is out of stock. In order to avoid a break in your ability to print documents, as you have an assignment deadline coming up, you may choose an alternative make or model because it is available straight away.
- Testing – It is hard, and possibly irresponsible, to choose a computer system without having tested it first. Depending on what it is needed for, you might want to test it for comfort and weight, especially if you travel a lot for work. Someone who uses keyboards a lot will probably want to try out the keyboard for touch, size and positioning of keys.
- Migration to new system(s) – Transferring data, software and files to a new computing system can be stressful as well as time consuming. The process of migration needs to be straightforward and reliable. Some suppliers include data migration as a service when purchasing a new computing system, and will transfer contacts, favourites and apps, for example, to another mobile phone for you.

Productivity

Businesses very often make a decision to upgrade, change or implement new hardware to increase productivity. Productivity in industry is a measurement based on the quantity of work carried out or products made in a given time period.

Security

Computer system security is of constant concern. Frequent news reports feature examples of major breaches in security due to hacking into personal and business records. The hardware we choose to use needs to be secure and have up-to-date protection against both physical and cyberattacks. Organisations will make their choices about IT systems according to the security precautions in place, especially where sensitive information is managed, such as police, health and financial records.

Data storage and recovery systems

Data is stored in a computer system on an HDD or SSD. Either of these technologies can be used as components in redundant array of independent discs (RAID) or network attached storage (NAS) systems. RAID can be used in standalone computers and both RAID and NAS can be used in networked systems.

RAID

RAID systems are used when more than one HDD or SSD are used inside the same computer. There are several RAID standards that use data redundancy to duplicate data across the drives, so improving speed and data security. Table 2.4 provides more detail about RAID standards.

PAUSE POINT Can you identify hardware products that can be used to improve security for a computer system? How do these products enhance security?

Hint Identify methods that can be used to restrict access to a computer system.

Extend Find some biometric products that can read user fingerprints or eye retinas.

Table 2.4: RAID standards

RAID	Description	Tolerance	Number of drives
0	Block level striping only, to spread read/writes across the drives, making a fast system.	None	2+
1	**Mirroring** only to create identical copies of data on the other drive(s).	All but one drive can fail	2+
2	Bit level striping with **hamming code error correction** to enable rebuilding data with spread read/writes.	One drive failure	3+
3	Byte level striping with **parity error correction** to enable rebuilding data with spread read/writes.	One drive failure	3+
4	Block level striping with parity error correction to enable rebuilding data with spread read/writes.	One drive failure	3+
5	Block level striping with parity error correction to enable rebuilding data with spread read/writes.	One drive failure	3+
6	Block level striping with double parity error correction to enable rebuilding data with spread read/writes.	Two drive failures	4+

Key terms

RAID level striping – RAID can be used to spread saving and reading data over the drives to improve speed. Conceptually, this is thought of as making stripes from the data across the drives. Block level striping divides the data into 512KB blocks with each stored onto one of the drives. Byte level striping uses the same technique, but divides the data into bytes. Similarly, bit level striping divides the bytes into bits for storage.

Mirroring – RAID uses mirroring to save two copies of the data, so, if a drive fails, a good copy of the data can be brought back from the mirror.

Parity error correction – This is an error-checking technique by which an extra bit is placed at the end of each byte to make the number of 1s in the byte (and parity bit) an even number. This is called even parity; a similar technique where the parity bit makes an odd number of ones is called odd parity. Double parity error checking is when RAID writes a second set of parity across the drives as an extra safeguard against data loss.

Hamming code error correction – This is used in RAID systems to utilise parity bits to rebuild corrupted data.

NAS (Network Attached Storage)

NAS systems consist of a box containing one or more backing storage drives that can connect to a network as a simple file server. They can be used as an alternative to **cloud storage**.

Key term

Cloud storage – a virtual space for storing data provided through an internet connection. Advantages of this technology include being able to access the same data from almost anywhere, at any time and on any device.

Computer software within a computer system

Software is any program or application (app) that can run on a computer system. In this section, you will look at the purpose, features and factors affecting the choice of operating system, utility software and application software.

Operating systems

In this section, you will learn about the types of operating system, the role of the kernel, how the operating system manages networking and security and factors affecting the choice and use of user interfaces and operating systems.

Types of operating system

Every computer has an operating system, that is, software which is there to connect other software to the hardware components and to let the user control the computer system. There are various types of operating system used to control different types of computer system.

Real-time operating systems

Real-time operating systems (RTOS) are used in every interactive computer system and, as the name suggests, they process data in current (real) time. If a system is not real-time, it has a batch operating system to allow for long jobs such as updating records to be run automatically. Real-time operating systems (RTOS) are used in specialised systems where the computer must respond instantly (in real time) to inputs. RTOS are, for example, used in in safety critical systems such as aircraft control and autopilot systems.

Single-user operating systems

Single-user operating systems are designed for computers used by one person at a time. These are usually multi-tasking so that the user can run several software applications at the same time. However, some are single-task systems which can only run one user application at any one time, such as on a simple mobile phone (unlike a smartphone, which runs several software applications at the same time and is therefore multi-tasking).

Multi-user operating systems

Multi-user operating systems are designed for large computers that are used by many people at the same time for jobs requiring a lot of processing power, such as weather forecasting. They enable several users to operate the same software applications at the same time, for example through a server, and are usually multi-tasking so that users can run several software applications at the same time.

The kernel

The kernel is the central or core part of an operating system and is used to control and manage system components and tasks. The role of the kernel includes the control and management of the following components and tasks.

- Program execution: To load software, allocate resources, run and terminate when finished.
- Interrupts: For components such as the mouse to request servicing by the CPU.
- Modes: To switch between user modes, to run apps, and system (kernel) mode, to run low-level code that the operating system needs for operations such as reading memory.
- Memory management: Needed to allocate RAM to running programs and to reuse RAM when programs end.
- Multi-tasking: To allocate enough CPU and peripheral time to every running program so that they all work concurrently.
- Disc access: To read from or to write to backing storage.
- File systems: To control how data is stored and organised on backing storage. Operating systems use a variety of different file systems which use different addressing methods (to allow the operating system to save and retrieve files). Simple file systems such as FAT32 are used for low capacity devices such as USB flash drives where security is not required. More sophisticated files systems such as NTFS can address larger devices and provide additional features such as user access rights, improved ability to recover from disc errors and encryption.
- Device drivers: To allow the operating system to communicate with hardware components, such as a graphics card, to best use all their features and performance.

Networking and security

Modern operating systems are expected to have a role in managing networking and security.

Networking for a home PC will be peer-to-peer, involving no servers, with basic sharing of the internet, printing and some folders. Security on a peer-to-peer network is likely to consist of a power-on password (or user recognition) for each computer, a WPA2 password for joining the WiFi and passwords on shared folders.

Networking for an organisation will be client-server, with servers controlling the resources that user accounts can access. Security on a client-server network is based upon authentication of users when they log in, usually by typing in their ID and password, and then upon the parts of the network they have rights to access. Systems running on the network often have separate authentication log-ins.

Factors affecting the choice and use of user interfaces

There are a number of factors affecting the choice of user interface for an operating system.

Most people will be familiar with the graphical user interface (GUI) provided by Microsoft® Windows. Other common GUIs are those provided by Apple's® OS X® and Linux®. The GUI makes the operating system easy to use, with icons and menus that can be selected using a mouse. This type of user interface is very effective for multi-tasking with each program (task) in a different window. The performance speed or response times from a GUI will be slower than other user interfaces due to the processing power needed to produce the graphics compared with simple text-based command line interface (CLI) or menu interfaces, but the high level of usability of GUIs means that they are now the most popular kind of user interface on computer systems.

The CLI was very popular on operating systems such as DOS or CP/M before GUIs became widespread. Commands such as DIR, to show what is on a drive (the directory) need to be typed into a CLI making them a little more difficult to use than a GUI, as you need to remember the commands, although help is available to list and explain the commands. Every modern GUI includes a CLI for technicians to use.

Menu-based user interfaces make the operating system simple to use and well suited to single-tasking operating systems such as some mobile phones and smart TVs. This interface can also be available on a modern GUI-run operating system, where, for example, it will show the boot-up options if there is a problem with the installation.

Android®, Apple® iOS and Windows® operating systems use touchscreens, particularly for mobile devices such as tablets, smartphones and some laptops.

Factors affecting the choice of operating system

There are many factors affecting the choice of operating system (OS), many of which are discussed in this section.

- Type of operating system: Most people will want to have access to a single-user multi-tasking operating system so that they can carry out more than one function at the same time, and such digital devices are more powerful and useful than a single task based system. PCs used by individuals and businesses will be single-user multi-tasking. Some businesses may also have access to a multi-user operating system for collaborative working, which will be accessed via a server or the cloud.
- Origin of operating systems: Individuals and businesses can choose between proprietary operating systems, such as Windows® and Apple® OS X®, and open-source operating systems, such as Linux®. Proprietary operating systems are generally more widely used and support most popular software. However, open-source operating systems have the advantage that users can contribute to the features and capabilities of the operating system and they are generally less restrictive.

Link

For more about open-source as opposed to proprietary operating systems, see the 'The principles and implications of open-source operating systems and software'.

- Compatibility: Depending on the operating system used, there may be differences in file systems and issues of compatibility. Apple® computers use the OS X which has a file system based upon Unix. Windows® uses the NTFS file system by default, although the older FAT32 is still an option. OS X has compatibility issues when it needs to write to a NTFS drive as this needs a third-party utility.
- Intended use: If you are planning to become a software developer, data engineer or an IT technician who solves network and security issues, you will need to interact directly with the operating system by using the command line. For a relatively new digital user or someone who has little interest in anything more than basic functions that are simple to use, save time, enhance daily living or who simply wants entertainment, then a WIMP style GUI operating system is likely to be the appropriate choice.
- Scope of customisation: Open-source operating systems are far more customisable than proprietary ones. Therefore, if an individual or a business needs to be able to customise their operating system, for example if they are in the software development business themselves, then they are more likely to choose an open-source operating system. Home and business PC users who just want to use office software, browse the internet, send emails and play games, are better off with a proprietary operating system which is designed for ease of use and to support the relevant apps.
- Performance: The level of performance can be important in the choice of operating system . The fastest performing operating systems make modest demands on the hardware, so tend to be from the Unix family or older versions of Windows®. Performance also comes from being able to support the latest hardware and software apps, so the most current versions of the operating system are more appropriate as they run on the best currently available hardware.
- Cost: The cost of an operating system will be a factor for many individuals and businesses. As well as the initial outlay for purchasing the operating system, there may also be a cost attached to technical support from the manufacturer. Individuals may wish to pay for support directly from the manufacturer, while larger businesses will use their own IT technicians. When a new operating system is purchased, the individuals using it may well have training needs which will need to be paid for, particularly if the new operating system is significantly different from the one they were using previously. The cost of a new operating system will also be greater if it is not compatible with the software and documents previously used by a business, because these too would need to be updated.

Use and performance of operating systems

The fundamental use of an operating system is to provide the interface between users and the software applications on the computer system. The operating system is the link between the people and the software via the hardware (keyboard, mouse and screen).

Organisations usually have networks with client-based operating systems which are accessed from individual workstations. Individuals log on from their workstations to the server operating system.

Newer versions of operating systems usually have more functionality, which, if the hardware remains the same, means that the performance of the operating system slows down. Therefore many organisations upgrade their hardware at the same time as running a major operating system upgrade to ensure that their users are able to keep up or increase their productivity.

The amount of data you want your computer system to process at any one time will affect the performance of the operating system. A computer system running multiple applications and performing numerous operations simultaneously, for example a computer system used to edit music and videos, will use considerable amounts of memory and processing power. In contrast, a computer system used entirely for administrative purposes, such as producing spreadsheets and word-processing documents, will use significantly less memory and processing power. In both examples, the operating system is single-user multi-tasking but the less memory required and the fewer applications running, the faster and more effectively an operating system can perform.

Utility software

Utility software runs alongside the operating system to carry out routine tasks to maintain and optimise the system operation. Utility software is there to add features and functionality to the operating system to help keep the computer running well. Some utilities are included with the operating system, such as defrag, while others can be purchased separately, such as anti-virus software.

The purpose, features and uses of utility software

The purpose of utility software is to undertake routine tasks. Routine tasks carried out by utility software include:

- security checks
- identifying and removing viruses
- cleaning up files
- software updates
- backing up data
- issuing warnings or alerts that updates or backups are required
- managing peripheral devices (such as cameras, printers and webcams).

Another use of utility software is to gather data to clean up the system and speed up performance by searching for junk data and purging it (removing it) to free up storage space. Utility software varies in the amount of memory it takes up according to its features.

Factors affecting the choice, use and performance of utility software

The choice of utility software is based on the needs of the system and whether a suitable utility is already present from the operating system installation.

Factors affecting the choice of utility software depend on what the user expects to get out of their device in terms of performance. If the device is mainly for browsing the internet or talking to friends using VOIP, it may not require software such as a backup utility. Users will choose utility software based on their needs. Most digital devices come with integrated software update processes, but it will be up to users whether they choose to add security and virus protection and to set up backup utility software.

Factors that can affect the performance of utility software include the amount of memory available on the digital device for utility software. Other factors are:

- bandwidth
- conflict with other utility software (such as multiple anti-virus applications running on one device)
- memory failures.

Application software

Application software (apps) enables users to interact with their computer systems to perform many different functions, including work and leisure activities. It is highly likely that you use several applications every day, such as Microsoft Word® and Excel®, as well as internet browsers or apps on mobile devices. Other examples of software applications are specialist business software used to manage stock control or accounting functions. They were all written by computer programmers. If you are studying computer programming, you may become familiar with programming software environments such as Windows® Visual Basic®, Logo and JavaScript.

The purpose, features and uses of application software

The purpose of application software is very wide ranging and includes software for work, entertainment, education, enterprise infrastructure, simulation, media development and product engineering.

The features of application software are focused on the expected usage. For example an app used to create presentations will have features such as transitions between slides when a slide show runs or a word processing app will have features to help document creation, such as different types of tab stops and control over margins.

Since the development of application software, business processes have been greatly enhanced. Application software enables businesses to cut costs, increase production and control quality through the use of automated digital processes and greater data processing power. There are also a number of businesses that benefit from the introduction of IT systems using application software that may not immediately occur to you. These include:

- farming – where software applications are used to manage harvesting, sorting, cleaning and packing produce
- textiles – software applications involved in the design and manufacturing processes in textile production.

Factors affecting the choice, use and performance of application software

For most organisations, the main factors affecting the choice of application software are compatibility with existing systems, cost and the user experience.

The usage of application software is largely determined by user needs. Performance is a minor factor in the choice of apps, as suitability is far more important and performance issues can usually be addressed by upgrading hardware.

One of the many reasons why Microsoft® products have been so popular worldwide is their versatility and usability (ease of use). Businesses needing general office applications mostly favour those which are familiar to the majority of users; in particular, Microsoft's Office® suite. When recruiting administrative staff, it is easiest to employ those with previously learned skills, such as how to use Microsoft Office® applications. Compatibility is also important. With so many people using Microsoft® Office, sending documents and spreadsheets using the Microsoft® Office file formats to other people is easy.

Apple Mac® products are most likely to be favoured by businesses in creative industries, such as advertising and graphic design, because the software that Apple® initially designed was more suitable for creative processes and, since then, they have built on that reputation. Microsoft® Office usually operates on PCs and does offer a version for Apple® Macs, although some Mac® users simply prefer the design of Apple® software.

The principles and implications of open-source operating systems and software

Operating systems and software are either open source or proprietary. Open-source operating systems are usually free, whereas proprietary operating systems are commercial and they cost money, either as a one-off payment or for a licence which can be renewed annually.

Behind all operating systems and application software there is programming code, known as the source code. You may have come across this if you have looked at the source code of a web page in **HTML**. There are two types of source code that are generally created – open source and proprietary.

Open source means that it is possible to edit and change the operating system (or application) software, as you have access to source code, whereas proprietary operating systems (and application software) are bought as-is, meaning they cannot be changed. An organisation with good in-house programmers would be able to modify an open-source operating system such as Linux®, to personalise it for that company. This means that not only can open source software be more closely customised to match an organisation's requirements, but it can also be much cheaper to implement. However, customising the software will require programmers who have the knowledge and skill to carry out the modifications and may also cause support issues since any problems that occur with the customised software will have to be dealt with by the organisation itself.

Key term

HTML – stands for hypertext markup language and is used to create web pages.

Link

For more on HTML, see *Unit 15: Website Development.*

For more about programming, see *Unit 1: Principles of Computer Science* and *Unit 4: Software Design and Development Project.*

Open source

Open-source code is a collaborative effort where more than one programmer contributes to the programming of the source code. It relies on peer review for testing its uses and functionality, and to fix any problems with it. Its main distinguishing feature is that the source code is open to use by anyone. Individuals can copy open-source operating systems and application software source code and, if they have the knowledge required, they can adapt it to their needs. Another distinguishing feature of open-source software is that it is usually free.

Link

For more about open source, see http://www.webopedia.com/DidYouKnow/Computer_Science/open_source.asp

One of the implications of using an open-source operating system, such as Linux®, and application software, such as OpenOffice®, is that it can be copied and modified by individuals with a malicious intent, such as expert programmers who do not share the same philosophy or responsibility to its integrity as the originators, or simply by inexperienced programmers. However, this risk can be mitigated by downloading from a reputable website.

On the positive side, open-source programming provides opportunities for individuals to be creative and share their knowledge and expertise globally and freely. It gives access to sophisticated software to people who would not be able to afford the proprietary software.

Proprietary

Proprietary operating systems and application software, unlike open source, have privately owned source code such as those produced by Adobe®, Apple®, Microsoft®, Pegasus® and Sage®. Proprietary OSs and apps will have been produced by teams of programmers who are employed by the company. Specifications and code of proprietary OSs and apps are kept secret within a company to avoid copies being made and to stop modifications being made by anyone other than the company's designated programmers. Proprietary operating systems and application software need to be licensed or purchased by individuals and businesses who can use the product, but cannot change the source code.

A major benefit of off-the-shelf proprietary software is that there is a lot of support available from other users, via the internet, training courses or books.

A further benefit of proprietary software for users is that the developers have control over the code so they are able to gather data on problems which occur and can use this information to correct programming issues. They are able to create code 'patches' that can then be delivered to registered users of the software.

One of the main implications of using proprietary software is that it can be costly, especially if the software is bespoke, that is, specially written for a particular client. Bespoke software often restricts further development to particular employed or contracted programmers.

Data processing

Data processing is the main use of computer systems for many organisations. Data can take many forms including customer records, stock information and financial accounts.

The use, features and implications of computer systems for data processing

You need to understand the ways in which computer systems are used for data processing and the implications of their use for both organisations and individuals.

Retail

Large supermarkets and other retailers use computer systems at the checkout to scan items sold and take payment, often registering points on a loyalty card. This gives the organisation useful information about stock levels and when to re-order goods and about the buying preferences of the customer. Customer information can be used to print vouchers that they can use on their next visit or to target mailshots and other sales promotions.

Supermarket systems consist of a large number of EPOS (Electronic Point of Sale) terminals at the checkouts connected to centralised server computers which record stock levels and automatically reorder items from suppliers when required.

Financial services

Banks have powerful computer systems to look after customer accounts and provide online banking and real-time information at cashpoints. These systems calculate interest and bank charges and interface with other systems to transfer money. Management use information from these systems to plan and market new products, such as savings accounts. Banks also have large distributed network systems with terminals located in branches connected to larger centralised server systems.

Farming

The farming industry is becoming increasingly automated. Livestock, such as chickens, are kept in barns that are automated to keep feeding and temperatures at optimum levels. Tractors use satellite-navigation systems to help control operations such as ploughing and seeding crops. These systems use sensors to detect environmental conditions such as temperature or geographical location and actuators to control equipment such as heaters or feeding machines. Usually the control software runs on an embedded system or laptop/desktop computer.

Manufacturing

Many manufacturing processes use computer controlled machines as they can perform tasks with a degree of accuracy and reliability that humans cannot match. Car manufacturing is heavily automated, with industrial robots used for tasks such as welding and paint spraying. Textile manufacturing uses automated looms and other machines to create cloth. Computer systems are used to program these systems and to control production according to the demands of the market. Computer controlled industrial machines generally use embedded systems or dedicated computers to control them. In a large manufacturing plant the control computers may be connected to central server systems to coordinate the manufacturing process

Engineering

The engineering industry uses sophisticated computer systems, based on high performance desktop computers running CAD (Computer Aided Design) software, to design and test components before manufacture. These designs can then be downloaded to computer-controlled machines to manufacture the physical product. Current advances in 3D printing make it possible to produce products anywhere without the need for shipping.

Science and medicine

Science and medicine are extensive users of computer systems, for example for exploration of computer models of theorems and for research into new drugs. Medical data can be used to identify the most effective treatments as well as poorly performing remedies. Doctors' surgeries typically use desktop systems connected to a small server to maintain patient records, while hospitals run a larger version of these database systems. Many medical instruments such as CT (computerised tomography) scanners use dedicated computer systems to create a detailed image of a person's internal organs from multiple X-ray images.

The role of hardware in collecting data

All data is entered into a computer system through hardware. Many hardware devices are specifically designed for data entry. These include keyboards, mice, touch screens and the various types of sensor.

The keyboard is still a very effective device for collecting data. Many data processing systems require paper-based information to be typed in so that the information becomes digital and suitable for further processing, such as sorting and arithmetic operations. Although a keyboard operator can make some typing errors, they can also add a lot of value to the activity as people can see problems in the source documents which an automated system might miss if the data was collected using a different method.

A mouse is not as versatile as the keyboard, but can still be effective at entering information, especially in completing on-screen forms using option buttons.

The touchscreen is a simple and effective data collection method that is often used in customer-facing systems such as bank cashpoints or automated check-in systems at GP surgeries.

There are many types of sensor used to collect data. One of the largest systems with sensors is the weather forecasting system used by the Met Office, which collects data from many thousands of sensors that measure rainfall, wind speed and direction, temperature, snow, atmospheric pressure, visibility, cloud and sunshine in the UK and across the globe.

Sensors that are used in computer-controlled environments could be as small as the central heating system in a small apartment or as large as a complex industrial process such as that used for refining oil. In both cases, the sensors collect a type of data that is then passed into the controlling computer system for processing. The output from these usually goes to relays which can switch electricity on or off to control hardware such as heaters.

Card readers are used in retail checkout systems to accept credit or debit cards to make a payment. The data from these transactions is sent to the card provider to record the sale. A loyalty card could be used to collect data for the retailer.

Many secure premises use swipe cards to control door entry. The data collected here records who used the door and when.

The role of software in collecting data

Software controls hardware in a computer system. Keyboards, mice and touch screens are likely to be the front end for software controlling a database when they are used to collect data. The data collection software, a spreadsheet or database, is used to collect, store and analyse the data. Sensors are likely to be part of a specialised system where software uses the data for a specific purpose such as creating a weather prediction model or controlling the temperature in an environment. Software can be used to indicate a problem with an input, for example showing a red light if a card swipe is not recognised.

Data processing functions

In this section, you will look at the main features of data processing.

Aggregation

Aggregation is a type of analysis where a large amount of data is processed into one or more summaries, which can then be used to create statistics or to provide data for another system. Data on potential customers could be packaged into aggregate reports which are then sold to businesses wanting to send out mailshots or run telephone-based marketing campaigns.

Analysis

Analysis is when some sort of meaning is extracted from data. There are many ways of analysing data such as exception reporting to identify poorly performing items in need of further attention. A supermarket could analyse sales of a new item and respond with a more prominent display or by withdrawing the item if sales do not meet expectations.

Conversion

Data processing can convert data from one form to another. A new system brought into an organisation may require a one-off conversion of data from the old system so that it can be used by the new, for example a payroll system. Conversions are often used in organisations to bring data held on a large central system, such as a selection of customer accounts, into a form that can be easily manipulated by the user, such as a spreadsheet.

Reporting

Reporting follows on from many data processing functions so that managers can understand the current state of their data or be made aware of any issues that have arisen during its processing, for example customers reaching their credit limit.

Sorting

Sorting is a basic method of data processing that is used to sequence data into **alphabetical or numerical order**. Sorting can involve multiple levels with the first level grouping identical items and the next level sorting items within each group.

Key terms

Alphabetical order – when data is placed in a sequence according to the order of letters in the alphabet, which means that anything starting with the letter B would be sorted into place after anything starting with the letter A and before anything starting with the letter C. Using alphabetical order, number 12 would be ordered after 1 and before 2.

Numerical order – when data is placed in a number sequence, usually ascending order, which means that 12 would be sorted into place after 11.

Research

What kind of systems are used for data collection and processing? What system does your school or college use? Do you have any family or friends who work in a large organisation? What kind of system do they use? What data is input and output? How is it processed?

Validation

Validation is used when data is entered to reject anything that is obviously wrong. Invalid data will not be allowed through validation. For example a product code could be two letters followed by four numbers so validation should reject anything which is not six characters in length and/or does not follow the pattern of two letters followed by four numbers.

The impact on individuals and organisations of using and storing data across multiple computer systems

In an ideal world, an organisation would be able to run all its various devices such as tablets, smartphones, laptops, desktops and servers integrated together into a single computer system. There have been real advances towards this goal in recent years, but using multiple computer systems are still the reality for many organisations. This section looks at some of the implications for individuals and organisations of using and storing data across multiple computer systems.

Access

Access to data is vital for staff who need to be able to work with customer accounts or other company information. This is easy on a standalone or networked computer but it can be difficult to synchronise data when multiple computer systems are in place which do not directly talk to each other.

Within a network, there might be an app that enables data from other apps to be updated overnight, so some data that is retrieved before the latest update could be a little out of date.

Discussion

With a peer or in a small group, discuss the benefits to businesses of synchronising data between mobile devices and an office computer network.

Cost

There could be a cost implication of storing data across multiple computer systems, due to the extra time needed to obtain information or from mistakes resulting from using out of date data on non-synchronised computer systems.

Implementation

Using data from another computer system could be via an update through a communication link or batch process. Sometimes the only way to use data from another system is to type it in using manual data entry, which is a time-consuming method that might result in human error.

PAUSE POINT What benefits are there to multi-level sorting of data? What real-life example can you find for data sorted in this way?

Hint Google 'multi-level sort Excel®' to find some examples of data sets that show ways of sorting.

Extend Why are the different sort orders useful for the data examples you found?

Tablet computers may be used by staff when visiting clients, with any new data synchronised through a WiFi network connection when they return to the office. Version control may be needed to ensure the best/latest version of data is always being used.

Productivity

Productivity will be reduced if data is stored across multiple computer systems, due to the time consumed in synchronising data, especially if this has to be done manually.

Security

Security might be improved with sensitive data on a very secure computer system, but, generally, using and storing data across multiple computer systems is less secure because there is more opportunity for hackers to access one of the systems. Security would be better in an integrated computer system as security can be controlled from one place and issues associated with securely transferring data between devices can be avoided. Storing data on mobile devices such as laptops or tablets may also create security issues, as these types of devices are more easily lost or stolen. Transferring data over an unsecured WiFi network to a mobile device being used out of the office might also present a security risk.

Backup and data recovery procedures

Backing up means to replace any data lost, for example, due to corruption or human error. The data is stored elsewhere (backed up) so that, if needed, it can replace the lost data. Backing up data is easier if it is all on a central server, as it can then be copied to the backup media in a single operation.

Backup procedures should be carried out on a regular basis, the frequency of which will depend upon the system requirements. A full backup might be made every week with incremental backups two or more times a day. A data recovery exercise should be carried out regularly to check that the data can be successfully restored to a system that has never been used for that data. This is important in case of a physical disaster, such as a fire or flood, affecting the location of the central computer server of an organisation.

Recovery procedures are also a lot simpler with one central server rather than lots of separate servers or networked computer systems.

Link

To learn more about backup and data recovery procedures, see the sections on 'Backing up data and IT disaster recovery plans', in 'Physical security' in *Unit 7: IT Systems Security and Encryption.*

Assessment practice 2.1

A02 A04 A05

A national chain of stores has an established IT system which uses barcodes on products to check levels of stock on shelves in the shop, storeroom and on the checkouts.

Management are very interested in the performance of the stores, so they use MIS to bring this information to them.

- Describe the hardware and software involved in this system.
- What data processing functions are likely to be involved in transforming the store stock and sales performance into useful management information?

The national chain of stores is considering mounting a scanner and tablet onto each of a number of trolleys to trial customer self-service.

- What would be the impact of this service?

Plan

- Do I have any existing knowledge of the task?
- How much time do I have to complete the task? How am I going to successfully plan my time and keep track of my progress?
- What aspects of the task do I think will take the most/least time? How will I balance these?

Do

- I can make connections between what I am reading/researching and the task, and can identify the important information.
- I am recording my own observations and thoughts.
- I am recording any problems I am experiencing and looking for ways/solutions to clarify queries.

Review

- I can explain how I would approach the hard elements differently next time (i.e. what I would do differently).
- I can describe what strategies I employed to cope with failure and improve self-esteem.
- I can draw links between this learning and prior learning.

B Computer architecture

Computer architecture is the design of how components are connected and work together in a computer system. You need to understand the implications of the current computer architecture models and the impact of relationships between component parts in these systems.

Approaches to computer architecture

In this section, you will learn about the different approaches to computer architecture, including the features and uses of different computer architecture models and factors affecting their choice. You will consider the implications of using different architecture models as well as emulation.

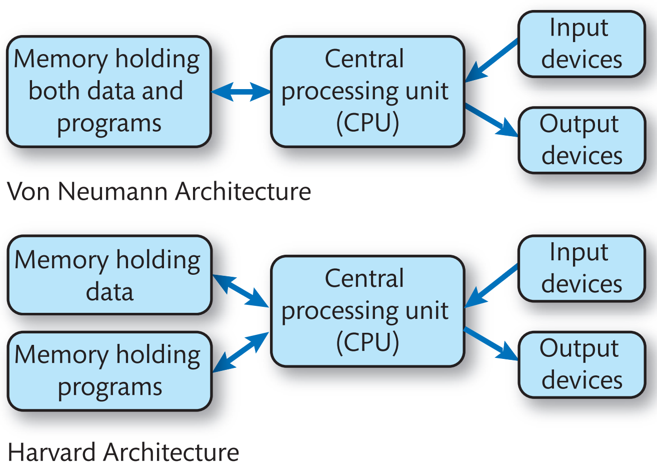


Figure 2.3: Computer Architectures

The features and characteristics of different computer architecture models

There are a number of different computer architecture models, each of which has its own features and characteristics.

Stored program model

The first computers did not use the stored program model, as programs were hard coded by setting switches. A technician needed to flip switches each time a new program was needed. Later systems used punched tape or cards to hold the programs and data. These have been superseded by modern computer systems.

The stored program model is normal in modern computers with the program(s) kept in electronic memory, which could be RAM or ROM. PCs, laptops and large systems use RAM, whereas ROM-based systems tend to be smaller and dedicated such as those used in a car engine management system.

Stored program model computer systems use either Von Neumann or Harvard architecture.

Von Neumann architecture

Most of the computers that you will have experience of using employ the Von Neumann architecture as they use the same RAM for both programs and data. The operating systems, apps and documents all share the same memory. Von Neumann architecture computers have a single processor that uses the **fetch-execute cycle** to run code from this memory.

Harvard architecture

The Harvard architecture, an alternative to Von Neumann, uses separate memory for programs and data. This architecture is used with embedded computer systems in devices such as personal video recorders (PVR) and **Arduino**.

Key terms

Fetch-execute cycle – this is the sequence repeated thousands of millions of times every second by a CPU to fetch the next instruction code from memory and then run that code.

Arduino – this is a small, cheap hardware device containing a CPU and input/output (I/O) ports, which can be used to explore programming and controlling devices.

Modern CPUs use Harvard architecture inside the chip to separate code and data into threads for faster processing, but the connection of the CPU to the motherboard and components will be Von Neumann.

Cluster computing

Supercomputers used by the Met Office to produce UK weather forecasting use the cluster computing architecture model. These computer systems have hundreds of thousands of processors and are capable of carrying out thousands of trillions of calculations every second. Simple home PC hardware can also be clustered together to provide a powerful system capable of much larger processing tasks than single machines.

PAUSE POINT Why do you think the Met Office needs so many parallel processors? Why would an ultra-quick single processor not be appropriate for this computer system?

Hint Find out how many input sensors from weather stations this system uses.

Extend What are the benefits to the Met Office of more processing power?

Uniform memory access and non-uniform memory access

Multi-processor computer systems, with more than one CPU, can share RAM between the processors. Uniform memory access (UMA) and non-uniform memory access (NUMA) are two techniques for sharing RAM between processors.

UMA is a technique for sharing RAM where each CPU can access any of the memory at the same speed. Individual CPUs can have their own cache memory allocated in the shared RAM. UMA systems share the same **memory bus**.

NUMA is an alternative technique for sharing RAM whereby each CPU can have some RAM allocated as local memory as well as sharing other (foreign) parts of the RAM with the other CPUs.

Key term

Memory bus – this is used to move data in a computer system between the CPU and the RAM.

Use and application of emulation

Processors are designed in families, for example the Intel CPUs found in modern computers have evolved from the 8086 processor from the late 1970s called the 'x86 family'. The 8088 version of this CPU was used in the original IBM® PC of 1981.

A family of processors uses the same **low-level program** commands (which can be expanded in later versions) and can usually run the same operating systems which provide the **API** needed for application software to run. An app will not usually run on a computer with an operating system from a different family.

Key terms

Low-level program – This is code written using instructions which map very closely to the CPU's actual instruction set. The CPU instructions are just binary numbers. In low level programming these are replaced by mnemonics. For example the CPU instruction for the add instruction might be 01000000 in a low level program this might be replaced by the mnemonic ADD.

API – this stands for application program interface, which is the place where code from an app can send calls (requests) to the operating system to make the hardware do something, such as opening a document.

Emulation occurs when software is run to provide an environment that is the same as that of a different processor family so that an operating system can run a program that was designed for a different processor. For example, a PS3 emulator could run on a Windows® PC so that the user can play a PS3 game on their PC. Emulation slows the computer system down as it adds an extra layer of software between the operating system and the hardware.

Link

For more about mnemonics and assemblers, see these key terms in 'The use and choice of instruction sets'.

Factors affecting the choice of different architecture models

The main factors affecting the choice of different architecture models are how suitable the architecture is and the cost of ownership.

Suitability of a computer architecture model is a complex mixture of the user experience, the apps currently available for the architecture and how well the platform integrates with the other systems run by an organisation.

Cost of ownership is calculated from the initial cost plus the ongoing expenses, which includes any licenses that require regular payment, and support costs.

Von Neumann with UMA is the default choice for a modern PC or laptop. This architecture performs well for a single user running one or more apps.

Harvard is the default choice for small dedicated devices as the code will be burnt into ROM with separate RAM memory for any changeable data that is entered.

Cluster computing with NUMA is a good choice where a massive amount of data from many sources needs to be processed quickly. The large number of CPUs can work in parallel to quickly process the data as it arrives.

UMA is a good choice for a computer with more than one CPU on the motherboard. Many networked server computers use multi-processor systems to provide good performance to the users accessing this system.

The impact of using different architecture models

The choice of architecture models for PCs and laptops is between the Apple® Mac family of hardware and the various manufacturers of PCs. The Apple® Mac family architecture has been developed during the last decade so that it is so similar to the PC that Windows® or Linux® can be installed onto a Mac® via the Boot Camp partition manager. It is, however, difficult to install OSX onto a PC as this operating system is locked down to Apple® hardware architecture, which limits hardware choice.

The impact from this choice is the operating system interface for the users, software and hardware compatibility. The Mac® OSX, Windows® and Linux® operating systems have different APIs, so need software apps written for these platforms. PCs have a wide variety of expansion hardware cards available, such as graphics, which are unlikely to work with Macs®.

Tablets and mobile phones have a close relationship between the hardware and the operating system, so can be classified as Android®, Apple® or Windows® devices. Android® is very popular for these devices with a wide range of apps available. The Apple iPhone® and iPad® families are also very popular, having a wide range of apps as well as offering good integration between these devices. Windows® phone devices have a much poorer choice of software apps but do offer the Windows® interface and some compatibility with Windows® apps.

The concepts of microarchitecture

Microarchitecture is the design of the very tiny circuits inside the CPU, motherboard and other computer components. Microarchitecture affects the choice of CPU for PCs and laptops and the choice of SoC for smartphones. In this section, you will explore the concepts of computer component microarchitecture.

Instruction cycles

An instruction cycle is when the CPU executes an instruction or command in low-level code. This is called a cycle because, when the execution is completed, the cycle restarts with the next instruction. Every CPU carries out many instruction cycles every second.

Execution speeds

The execution speed is the rate at which a CPU can process instructions. This section will look at the execution speeds in more detail.

Factors affecting execution speeds

The main factors affecting execution speeds are the clock speed and the design of the CPU. These and other factors are outlined here.

- Clock speed: The clock speed is how many instructions can be executed in a second. A typical processor can be rated at 4GHz, which means that it can execute 4,000 million instruction cycles every second.
- Design of the CPU: The design of the CPU circuits has a large impact on performance. For example, Intel designs include i3, i5, i7 processors. The performance of each of these processors at the same clock speed varies, with the i7 being the best and i3 being the slowest. The i7 has more design features compared with the i3, such as 8MB of cache (compared with 4MB) and turbo boost (so that the CPU can speed up for short periods).
- Cache memory: The cache memory in a CPU holds queues of instructions and data, ready for processing, to reduce the waiting time involved in fetching them from RAM and to help speed up writing to RAM. The greater the cache memory, the faster the execution speed.
- Number of processing cores: Multiple **cores** are able to execute code faster, for example a 4 core CPU can run four code threads at the same time, whereas a 2 core CPU can only run two at a time.
- Cooling: Cooling the CPU is important because if a processor gets too hot, it reduces the speed (throttles back) to generate less heat and so help cool itself down.
- Motherboard design: The design of the motherboard has a significant influence on execution speeds as this component connects the CPU to the RAM, video and backing storage.
- Amount of RAM: An inadequate amount of RAM can have a serious impact on execution speed as current code and data will need to be swapped out to backing storage. The trigger point for this **paging** depends upon the operating system and the apps and data open at any one time. For example, 4GB of RAM could easily be enough for one user but nowhere near enough for another user who needs several apps open at once or needs to work with large data sets.

Key terms

Core – the part of the CPU that is able to run code. Up until 2005, all processors had single cores as it was not possible to make the circuits small enough to fit two cores into one chip before that. 'Quad' (4) core processors are now common but processors with higher numbers of cores do exist.

Paging – this is a technique used by a PC or laptop to use the backing storage as an overflow area for RAM if the RAM fills up with programs and data, which would slow the system down. The technique is called paging because RAM is treated as a number of same-sized pages to make swapping in and out of disc easier.

Methods of increasing execution speed

The most effective method for increasing execution speed is to upgrade the CPU, motherboard, video system and RAM. A better CPU should increase the execution speed. The motherboard, video system and RAM move data in and out of the CPU, so a faster execution speed can be achieved through better components.

Some users like to over-clock their computer system. Over-clocking is when the system is set up to run the CPU at a faster speed that it has been rated for. There are risks in doing this because the faster speed will generate too much heat and may burn the CPU out or it may run unreliably, resulting in computer system crashes.

Implications of execution speeds

Faster execution speeds provide a better user experience as the system responds well and feels more responsive and satisfying to work with. A fast system encourages the user to explore new ideas and use more interesting software because they are not constantly waiting for the system to catch up with them.

Slow execution speeds can be quite frustrating to work with, because they involve waiting for simple tasks to complete such as the screen updating or loading documents.

More demanding software such as game development environments need fast execution speeds from the hardware used.

The use and choice of instruction sets

An instruction set is a range of low-level program code commands that are understood by the CPU. These instructions are called 'machine code' and are usually programmed using **mnemonics** through an **assembler**. Machine code is low-level code that has a direct influence on processing, which means that code can run very fast with the programmer being able to make the CPU do exactly what is required. The downside is that a lot of code is needed, as hundreds of machine code instructions are needed to run a program that would take a single high-level program command. Traditionally computers have complex instruction sets (CISC) with a large number of instructions, many of which are very rarely used in day-to-day programs. Reduced instruction set computers (RISC) only have those instructions which are commonly used, which makes for a simpler more efficient processor. Instructions sets also vary in the number of bits used for each instruction, with 8, 16, 32 and 64 bit instruction set computers in common use. In general the more bits used the larger the amount of memory that can be addressed by the instructions.

Key terms

Mnemonics – these are acronyms used in a low-level programming environment for producing machine code programs. Most machine code instructions are made from the opcode (command) and the operand (value or address the opcode is to use). Each mnemonic represents a single machine code opcode instruction, such as DEC which will decrement the operand (decrease by 1).

Assembler – this is a programming environment used for producing machine code programs. Use of low-level programming is quite rare as most code is written in high-level languages such as C#, which are a lot easier to use.

Pipelining

Pipelining is a technique used inside the CPU to help run code faster. Pipelining works by lining up code instructions inside the CPU so that several pipelines can run side by side at the same time. This is done to keep the CPU as busy as possible by processing the next code instruction as soon as the previous one is complete.

Pipelining breaks each instruction into four stages: fetch, decode, execute and write-back to RAM. As one instruction is being executed, the next is being decoded and the one after that is being fetched.

Cache

Cache is specialised RAM that helps to improve performance when a faster component needs to communicate with a slower part of the computer system. The cache is memory that is shared by both components with all data moving through the cache. Data sent from the faster component to cache is very quick, and the slower device can then take the data from cache at its own pace. This means that the faster component does not need to wait needlessly.

When the faster component needs data, more than is immediately wanted is brought from the slower device to cache so that, when the faster device needs more data, it is already in cache. For example, the CPU has cache memory inside the chip as this component is faster than the RAM (see Table 2.5).

PAUSE POINT Pipelining is a powerful technology that is built into modern processor CPU chips to help them run code more effectively. What is pipelining? How does it improve performance? What are the potential problems that need to be overcome in the design of pipelining inside these chips?

Hint Use the internet to research how pipelining works.

Extend Visit the Intel website to read their promotional materials for their current range of processors. What has Intel done to design their best use of pipelining?

Table 2.5: CPU performance factors

Factor	Description	Units	Best performance
Core design	Which family the CPU cores belong to	No units, based upon product names, e.g. i7	No performance figures
Number of cores	How many CPU core circuits are in the chip	Single, dual, quad, six, eight	A larger number of cores produces better performance
Clock speed	The number of processing cycles carried out every second	Gigahertz (GHz), number of 1,000,000,000 cycles in a second	A larger clock speed produces better performance
Manufacturing process	How tiny the circuits can be made	Nanometre (nm), 0.000000001 m	A smaller manufacturing process allows more circuits on the chip, which runs slightly faster as the parts are closer together
Cache	Fast memory in the CPU that holds instructions waiting to be processed	Megabyte (MB)	A larger cache gives a better performance as more code and data can be pre-loaded into the CPU

Registers

A register is a collection of circuits used to hold a byte of code in the CPU. When the CPU runs code, registers are used to hold the instructions and data and to help communicate with the RAM.

Link

For more about how registers are used, see 'Registers and register handling'.

Multi-processing and multi-threading

Being able to run different parts of the program code at the same time can obviously improve performance. Multi-processing and multi-threading are two ways of running different parts of the code at the same time.

- Multi-processing: This refers to hardware that is able to run more than one process at the same time. It could be several cores in the same CPU, several CPUs on the same motherboard or many CPUs in a larger system such as a supercomputer.
- Multi-threading: This is when the code in an app is run as several threads, each at the same time. For example, a modern word processor can have a thread to check for spelling and another to check grammar, while other threads respond to keyboard and mouse input. Multi-threading can also be applied to processor CPUs. If the current instruction cannot continue (perhaps due to waiting for data from the RAM) the CPU can switch execution to waiting processes which can continue.

The features and implications of embedded and mobile central processing unit (CPU) architecture

Embedded and mobile CPUs are used to control devices such as domestic appliances (washing machines, microwaves, etc.) and industrial machines such as a paint spraying robots, and to control systems such as a vehicle engine management system as well as smartphones and tablets. The architecture of the control for these systems is based upon a SoC, which means a system on a chip.

The SoC is a functioning computer system on a single chip. The CPU inside the chip is based upon circuits designed by manufacturers such as ARM, Zilog and Texas Instruments.

The connections between the SoC and the rest of the mobile device are a very important consideration for the designers of these devices who are looking to create easy control of specialised inputs and outputs through dedicated co-processors inside the SoC, such as 3D touch on a smartphone.

Most laptops use specialised mobile CPU architectures with processors that are designed to draw a small amount of power to maximise battery time between charges. Embedded CPUs used to control processes in machines require interfaces to sensors and actuators. For example an embedded system in a washing machine would need sensors to detect the water level and temperature and actuators to operate the water valve and heater.

The features and implications of microcomputer CPU architecture

A microcomputer is a computer system controlled by a microprocessor, which is a CPU chip. Although this is a broad definition that includes all laptops and many servers, microcomputer usually refers to a desktop PC system.

The main focus of microcomputer CPU architecture is the processor and motherboard, as these have the most impact on performance. The interfaces to backing storage, video and external devices can be bottlenecks which slow the system down, for example when you open files from backing storage.

The features and implications of server CPU architecture

For many businesses, server CPU architecture is a specialised adaptation of a microcomputer which can reliably control a network with many computer systems which need to be available 24/7.

The need for reliability can be met by incorporating redundant components. Redundancy is when a component in a computer system is duplicated. For example, a server can have two power supplies so that if one fails the other can take over. If a redundant component is a 'hot plug' it can be replaced without needing to turn the server off.

Servers also need a high enough speed and capacity to meet the needs of the many users connected to the computer system. Many servers have two CPUs on the motherboard with massive amounts of RAM and backing storage. RAID is common to maximise the speed of access to backing storage and to spread data across the drives, so a broken drive can be 'hot-plugged' for a replacement with no loss of data and without needing to turn the system off.

PAUSE POINT Many businesses use a server computer that is an enhanced version of a desktop PC. How is this different from a normal desktop PC? What features are likely to be present in a server computer?

Hint Close this book and brainstorm as many features that you can think of that would be useful in a server computer, for example lots of disc space.

Extend Visit the Dell UK website to research the features that they build into their more expensive servers. How does this list compare with your brainstormed list?

Registers and register handling

There are many registers inside the CPU of a computer system to hold the bytes of code and data when they have been fetched from the RAM for processing.

Types of register

There are a number of different types of register – general purpose registers and a variety of special registers – all of which you will look at in this section.

General purpose registers

General purpose registers are used in the CPU to hold data when code is executed. A low-level program could be written to loop through some data to count how many records match a search value, with one register being used to keep a count of how many matching records have been found and another register used to keep a count of how many iterations have been made.

Special registers

Special registers are inside the CPU and each is designed for a particular role during the processing of code.

- The accumulator is inside the ALU (arithmetic and logic unit) and is the part of the CPU where all the calculations and comparisons are made. This register usually holds one of the values to be used before the operation and always holds the result after the operation is complete.

- The instruction register (IR), also known as the current instruction register (CIR), is the place where each machine code instruction is held for decoding and execution by the control unit.
- When the CPU needs to read or write to memory, the address in RAM is placed in the memory address register (MAR) which is then sent by the address **bus** to the RAM so that the data bus can connect to that location.
- The memory data register (MDR), also known as the immediate access store, is inside the CPU at the end of the data bus. When the control unit sends a signal, data is copied from the MDR down the data bus to the RAM location specified by the MAR or from the RAM location to MDR.
- The role of the program counter is to hold the address of the next program instruction. Machine code instructions have different lengths, so when an instruction is fetched into the CPU, the program counter has the length of this code added to the address of the current instruction already held there, so it now points to the next. Before an instruction is fetched, the program counter is copied to the MAR so that the data bus can connect to that RAM location.

The function and purpose of general and special registers and their impact on the way in which computer systems perform

The function of the general registers is to be a resource for low-level programs and they fulfil the role of **variables** in **high-level programs**, allowing them to store data needed during processing.

The size of a register has an impact on the way in which a computer system performs. A 128 bit register is able to hold 16 bytes, while an 8 bit register can only hold 1 byte. Bigger registers need fewer clock cycles to process large calculations or to transfer data, because more bytes can be processed in the same clock cycle.

The role of interrupts in a computer system

Interrupts are used when a part of the computer system needs to get the attention of the CPU or operating system. A hardware interrupt could be a key being pressed or the click of a mouse. A software interrupt could be a process such as a wizard needing the user to make a decision.

After an interrupt occurs, it is handled while whatever operation is currently running is suspended. When the interrupt has been serviced, the suspended operation is resumed. Handling an interrupt involves copying the content of the registers out of the CPU. When the suspended operation resumes, the register contents are copied back into the CPU registers.

Key terms

Bus – the motherboard has parallel metal tracks to connect different parts, allowing each bit in a byte to travel along a track side by side when transferring data between components such as the CPU and RAM.

Variable – used by program code to hold a value as the program runs.

High-level program – this is code written within most programming environments. One command in a high-level language, for example pctResult.Image = Image.FromFile("../Pics/Correct.jpg"); would compile into many lines of binary machine code. High-level programming is much easier than low-level programming as much less needs to be typed and because the keywords are much easier to understand and use.

Assessment practice 2.2

A01 A02 A05

You have a part-time job in a small computer store as an assistant customer advisor. The owner has asked for some materials that the shop can use to help explain to customers how processors work.

Produce a booklet which describes how the fetch–execute cycle works in a CPU.

Include the special registers involved in these operations along with their roles in the fetch–execute cycle.

Plan
- How will I approach the task?
- Are there any areas I think I may struggle with?
- What resources do I need to complete the task? How can I get access to them?

Do
- I can question my own learning approach.
- Am I utilising all of the support available to me?
- What am I struggling with? Do I know how to overcome this?

Review
- I can draw links between this learning and prior learning.
- I can explain what skills I employed and which new ones I have developed.
- I can make informed choices based on reflection.

C How data is represented by computer systems

You need to understand the characteristics, concepts and implications of the methods of computer data representation that are used in modern computers, and these are covered in this section.

Number systems

All of the number systems that we use have a base number which is always shown as 10 in that number system. The position of each digit is always in columns, with each column determining the scale of any digit in that column.

The lowest column for a whole number in any number system will be the base number to the power of zero. As any number to the power of zero is one, this column is always the '1s' column and no scaling is needed for any digit in this column.

The other columns have different scaling, according to the number system, with the base number at a different power for each column. We are used to using base 10, our main numbering system, which has column scaling (from low to high) of 10^0 (1s column), 10^1 (10s column), 10^2 (100s column), 10^3 (1,000s column) and so on. The base 10 numbering system is called denary.

The number 289 in denary is made from 9 in the 1s column, 8 in the 10s column and 2 in the 100s column so this number is actually 9 + 80 + 200.

The use and interpretation of number systems used in computer systems

In this section, you will look at how number systems are used and interpreted in computer systems.

Units of digital data

All parts of a computer system use one of two values to hold or communicate data or code. The two values vary according to where they are, for example, RAM has electricity or no electricity, an HDD has tiny spots magnetised to north or south, an optical drive has tiny spots that reflect or absorb a laser beam, copper network cabling has tiny bursts of electricity or no electricity.

These two values are thought of as 1 or 0, sometimes true or false. This unit of digital data is called a bit (from **b**Inary dig**it**). A bit offers a choice of only two values, but, for most purposes, a wider choice of values is needed. The basic unit of digital data is the **V**, which is made from 8 bits, giving 256 different possible combinations.

Other units of digital data use multiples of approximately thousands. These are kilobyte (1,000 bytes), megabyte (1,000,000), gigabyte (1,000,000,000), terabyte (1,000,000,000,000) and petabyte (1,000,000,000,000,000). We round these off to multiples of 1,000s whereas the binary equivalents that are actually used by systems are 1024s.

Tip

The commonly used abbreviations for storing or using multiples of bytes are: kilobyte (KB), megabyte (MB), gigabyte (GB), terabyte (TB) and petabyte (PB).

Communications are usually measured as multiples of bits and are shown using lower case letters as follows: kilobit (kb), megabit (mb) and gigabit(gb).

Binary and hexadecimal

Everything inside a computer is logically a 1 or a 0 which makes it difficult for us to understand or communicate data, such as a byte made from 01000001.

Binary is the numbering system with base 2, with column scaling (from low to high) of 2^0 (1s column), 2^1 (2s column), 2^2 (4s column), 2^3 (8s column) and so on. There are only two possible digits in binary, which makes it a good match to the data actually inside a computer.

The earlier example of a byte made from 01000001 can be expressed as 65 in denary as there is just 1 in the 1 and 64 columns and 0 in the rest (1+64=65). Obviously, 65 is much easier to write, read and understand than 01000001 as well as taking up a lot less space on a screen or paper.

IT professionals who work with binary use the hexadecimal system, rather than the denary, as it offers a much better representation of binary. Hexadecimal is the numbering system with base 16, with column scaling (from low to high) of 16^0 (1s column), 16^1 (16s column), 16^2 (256s column), 16^3 (4096s column) and so on.

Hexadecimal looks complicated, but, in practice, only the 1s column is actually used for binary conversions. This column can contain any number between 0 and 15 (in denary). Only one digit can be in a column, so letters are used for values of 10 (A), 11 (B) and so on to 15 (F).

Four binary digits can hold a range of values from 0 (0000) to 15 (1111) which is an exact match for a single hexadecimal digit which can represent these as 0 (0000) to F (1111).

Any byte can be represented as a two-digit hexadecimal number with a range of possible values from 0 (denary), 00000000 (binary), 00 (hexadecimal) to 255 (denary), 11111111 (binary), FF (hexadecimal).

PAUSE POINT Binary is a numbering system with only two valid digits, 0 and 1. Any collection of binary digits can be converted into a denary or hexadecimal number by including column headings and then adding together every column heading with a 1 under it. What is a base number? Why do column headings work for numbering systems?

Hint Research the BBC Bitesize website on binary addition to check that you are able to add two binary numbers. Translate the numbers into denary (base 10) to check your answers.

Extend What would be the column headings for a base 8 numbering system? Convert two numbers into base 8 and then add them together. Translate your answer back to base 10 to check that it is correct.

To convert a byte to hexadecimal, divide it into two four-bit numbers (nibbles) and then convert each into a hexadecimal digit. The byte of 01000001 in the earlier example can be expressed as 41 in hexadecimal.

Discussion

A two-digit hexadecimal number can represent any byte consisting of 8 bits. With a peer or in a small group, discuss the maths behind converting the upper nibble into a hexadecimal digit in the 16s column.

Binary coded decimal (BCD)

Another method for using bytes to hold denary numbers is BCD. In this system, each nibble (half of the byte) is treated as a denary digit. Using BCD, 01000001 would be 41. 11111111 would be invalid, as 99 (10011001) is the largest valid number possible in a BCD byte.

The use of binary arithmetic (including BCD) to perform calculations

Binary is a numbering system, so can be used to perform calculations. Remember: you can always check your answers by converting the binary to denary to confirm that the result is the same.

Addition

Binary addition between two numbers is very easy and just like using denary. Start from the 1s column at the right of the number and add the pair of digits, carrying to the next column if necessary.

128	64	32	16	8	4	2	1	
0	0	0	0	1	1	1	1	**15**
0	0	1	0	0	1	1	0	**38**
						0	1	
					1			**Carry**

1 + 1 = 10
Zero goes in this column and 1 is carried over to the next column

1 + 0 = 1

128	64	32	16	8	4	2	1	
0	0	0	0	1	1	1	1	**15**
0	0	1	0	0	1	1	0	**38**
					1	0	1	
				1	1			**Carry**

Here we need to add 1 + 1 plus the 1 carried over
1 + 1 + 1 = 11
So 1 goes in this column and 1 is carried

128	64	32	16	8	4	2	1	
0	0	0	0	1	1	1	1	**15**
0	0	1	0	0	1	1	0	**38**
				0	1	0	1	
			1	1	1			**Carry**

1 has been carried over so in this column
1 + 0 + 1 = 10
So 0 goes in this column and 1 is carried

128	64	32	16	8	4	2	1	
0	0	0	0	1	1	1	1	**15**
0	0	1	0	0	1	1	0	**38**
0	0	1	1	0	1	0	1	**53**
			1	1	1			**Carry**

0 + 1 = 1

0 + 0 = 0 plus the carried over 1 makes this 1

Result is 53
32 + 16 + 4 + 1 = 53

Figure 2.4: Adding two 8 bit numbers: 00001111 and 00100110 – all addition shown is binary

Subtraction

Binary subtraction can be carried out using two's complement. The number to be subtracted (subtrahend) is converted into two's complement then added to the number it is to be taken away from (minuend). An example using denary would be 35 – 12 where the subtrahend (12) is converted into the negative (–12) then added to the minuend (35): 35 – 12 gives the same result as 35+ –12 with 23 the answer for both methods.

Key term

Two's complement – this method is used in binary arithmetic to represent negative numbers. To convert a binary number into two's complement, firstly invert the number (swap 1s for 0s and 0s to 1s) then add 1 to this inverted number to calculate the two's complement. The same operation (invert then +1) is used to convert a two's complement number back to binary.

Two's complement is similar to a negative number, but, when the calculation is made, a 1 should overflow from the high end (left) of the answer. This is expected and the 1 is ignored. If there is no overflow, the answer is negative, shown as two's complement, so needs to be converted back to binary.

128	64	32	16	8	4	2	1	
0	0	1	0	0	0	1	1	**35**
0	0	0	0	1	1	0	0	**12**

First step is to convert the subtrahend to twos complement (invert the bits and add 1)

128	64	32	16	8	4	2	1	
0	0	0	0	1	1	0	0	**12**
1	1	1	1	0	0	1	1	**invert**
0	0	0	0	0	0	0	1	**Add 1**
1	1	1	1	0	1	0	0	**2s complement**
				1	1	1		**Carry**

Now add the 2s complement number to the minuend

128	64	32	16	8	4	2	1	
0	0	1	0	0	0	1	1	**35**
1	1	1	1	0	1	0	0	**2s complement**
0	0	0	1	0	1	1	1	**23**
1	1							**Carry**

There is a carry bit on the last digit which is ignored

The result is 00010111 which in denary is 23

Figure 2.5a: Subtraction. In this example 12 (the subtrahend) is subtracted from 35 (minuend)

Multiplication

Binary multiplication is the same as with denary, but often easier to carry out. If denary is used to multiply 370 (multiplicand) by 12 (multiplier) you could multiply 370 by 10 then add this part-result to 370 multiplied by 2: 3700 + 740 = 4440.

To carry this out in binary, you have 0101110010 (multiplicand) by 1100 (multiplier). Each 0 in the multiplier can be disregarded as, when the multiplicand is multiplied by 0, the part-result will be 0. Each 1 in the multiplier needs to produce a part-result, all of which will be added together for the answer.

```
        0 1 0 1 1 1 0 0 1 0
X                   1 1 0 0
---------------------------
        0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0 0
    0 1 0 1 1 1 0 0 1 0
  0 1 0 1 1 1 0 0 1 0
---------------------------
  1 0 0 0 1 0 1 0 1 1 0 0 0
```

Where the multiplier digit is zero, all the results in the part result are zero

All the part results are added up to give the answer

Result is 1000101011000 Which is 4,440 in denary

Figure 2.5b: Multiplication

Division

For the calculation 162 ÷ 6 in denary using the 'bus stop' method, we write the dividend (162) under the bus stop and the divisor (6) before it. Starting at the left, the divisor cannot be divided into the first dividend digit (1), so we write a 0 above the bus stop for that column then move right. The divisor can be divided into 16 twice, so we write a 2 above the bus stop for that column and then carry out the division, which leaves a remainder of 4, so we move to the right to bring down the next digit from the dividend, giving 42. The divisor can be divided into 42 seven times, so we write a 7 above the bus stop for that column and then, as we have reached the lowest digit of the dividend, the calculation is complete (see Fig 2.6).

	0	2	7
6	1	6	2
	-1	2	
		4	2
		-4	2
			0

Figure 2.6: Denary division using the bus stop method

Binary division can be carried out using the bus stop method.

For the calculation 162 ÷ 6 we need both the dividend (162) and the divisor (6) in binary. Starting at the left, if the divisor cannot be divided into the dividend, write a 0 above the bus stop for that column then move right. If the divisor can be divided into the dividend, write a 1 above the bus stop for that column and then carry out the division and move right (see Fig 2.7).

			128	64	32	16	8	4	2	1	
			0	0	0	1	1	0	1	1	
1	1	0	1	0	1	0	0	0	1	0	162
Divisor (6)				-1	1	0					
				1	0	0	0				
					-1	1	0				
					0	1	0	0	1		
							-1	1	0		
							0	1	1	0	
								-1	1	0	
								0	0	0	

Figure 2.7: Binary division using the bus stop method

The use of binary to represent negative and floating point numbers

There are many methods used inside computer systems to represent numbers using the binary system, including negative and floating point numbers.

Negative binary numbers

Two's complement has been used to represent binary negative numbers since the earliest microprocessors, because it enabled subtraction to reuse other circuits already in the CPU for adding numbers and inversion using NOT gates.

Link

For more about NOT gates, see 'Boolean logic'.

Floating point binary numbers

The use of floating point binary numbers to represent very large and very small numbers is similar to the standard form (also known as scientific notation) that is used with denary numbers where the number is shortened to a **mantissa** with a single digit followed by a set number of decimal places and an **exponent** to remember how many places the decimal point moved.

Using standard form, 27,563,498,208 would be written as 2.76×10^{10} (this can also be written as 2.76E10) with 2.76 representing the number to two decimal places and $\times 10^{10}$ showing that the decimal point moved 10 times to between 2 and 7 in 2.76. Accuracy is lost with standard form because to convert 2.66×10^{10} back we would get 27,600,000,000 as there is no way of recovering the detail lost in rounding the mantissa to two decimal places.

A floating point binary number has the mantissa normalised to 0.1 (1.0 for two's complement negative numbers) with a set number of digits, and the exponent shows how many places the binary point has moved to normalise the mantissa.

Key terms

Mantissa and exponent – these are used to represent very large or very small numbers in a small space using floating point or scientific notation. The mantissa is the number part and the exponent shows how many places the point has moved, e.g. 23,467,334 in denary can be shown as 2.35×10^7, where 2.35 is the mantissa and 7 is the exponent.

Text representation

Writing text is a very common use for computer systems. Text representation standards are needed so that text and documents can be shared and viewed on the many different computer systems in use today.

The purpose and implications of using codes to represent character sets

Bytes are used in every computer system. These are made from 8 bits which can be arranged in 256 different combinations. The world of computing uses a common set of codes for the alphabet and numbers. Together these take up 62 (0–9, A–Z, a–z) of the 256 possible codes in a byte.

The 8 bit code for a character can be represented in denary or hexadecimal to give a sequence. The letter A is held in memory as 01000001 and can be represented as 65 in denary or 41 in hexadecimal. The letter B is 01000010 (66 or 42), the letter C is 01000011 (67 or 43) and so on.

The difference between upper and lower case characters is bit 6, set to 1 for lower case or 0 for upper case so the code for a is 01100001, 97 or 61, for b is 01100010, 98 or 62, for c is 01100011, 99 or 63 and so on.

This structure makes it easy for computer systems needing a user ID or word processors to recognise upper or lower case versions of a character by ignoring bit 6. Conversion between upper case and lower case is also easy by setting bit 6 to 0 or 1.

Discussion

Masking is used to convert an alphabetic character between upper and lower case using AND and OR operations between the character code and the mask. With a peer or in a small group, discuss the mask and logic operations needed to convert a character to upper case or to lower case.

Using codes to represent character sets offers a lot of possibilities for the font used for the character set. The appearance of numbers and alphabetic characters is easily changed using a different font. A font can also make use of the other codes for shapes or accented characters.

The features and uses of common character sets

In this section, you will explore the features and uses of two common character sets, ASCII and UNICODE.

ASCII

The ASCII (American Standard Code for Information Interchange) character set is used for the English language and is based on up to 7 bits with 128 different characters, which include control characters such as CR (carriage return), ESC (escape) and so on. Extended ASCII uses 8 bits.

UNICODE

ASCII has been in widespread use since the 1960s, but UNICODE has been the character set used by Windows® and most websites since the mid-1990s. The UNICODE character set uses between 1 and 4 bytes for each character, giving over a million possible characters, which makes this a good choice for systems using several languages.

UNICODE is compatible with ASCII because they share the same codes for numbers and the alphabet.

Image representation

Images are used in many ways by computer systems. The types of image used by computer systems include photographs, diagrams and animations. Bitmap/raster images are best used for photographs and vector images are best for diagrams.

How bitmap/raster image data is stored and represented within a computer system

Bitmap and raster are both terms used to describe images made from millions of dots (pixels), each of which has a colour. (There are no practical differences between the terms 'bitmap' and 'raster'). Each pixel is represented by a binary number. In a pure black and white image just a single binary digit would be sufficient to represent the two possible states of the pixel (black or white). For colour images large binary numbers are used to represent a range of colours. Using 8 bits to represent each pixel provides 256 possible colours. The number of bits used is called the colour depth, which is covered in the next section.

Images taken by a mobile phone, digital camera or through a scanner are raster images. Raster images are best used for photographs as their images are in the same form as captured by these devices.

Vector images are very different from rasters because they are made from coordinates that define the placing of objects that have properties such as line thickness and fill patterns, which are based on mathematical formulae. These images are better suited for drawings and diagrams than photographs.

Raster images can be converted into vector images although this is a time-consuming process and may even be impractical for some photographs due to the amount of fine detail.

The impact of image resolution on the way images are stored and represented

Image resolution is shown as two numbers, for example 1920×1080 which means 1080 rows of 1920 pixels. The bigger the number the more detailed is the image, which has more pixels and a larger file size when saved to backing storage.

Vector images typically have much smaller file sizes due to the limited amount of stored information for this type of image compared with the need to store the colour of every pixel in a raster image.

Printers and displays use different technologies to make images, but they all create dots on paper or other media. The most efficient use of storage space is made when the resolution of the image matches that of the output device. In a raster image, fine detail that cannot be displayed or printed wastes storage space.

Designers of web pages using images should consider the purpose and audience of the web page when deciding how high the resolution needs to be for the images they are using. There is no point in using a high resolution image which will then be scaled down by a web browser so that it fits onto the web page because it would slow down the loading of the page.

The quality of the image is not just a matter of the resolution, but also of how small and closely packed the pixels are. This is measured in dots per inch (dpi), with a higher number of dpi meaning that the quality of the image is higher.

The impact of sample/bit depth on the way that image data is stored and images are displayed

The bit depth means how many colours are sampled when the image is input. 256 colours use 1 byte for each pixel, which is also known as 8 bit colour depth. Other popular colour depths are 16 bit (64K colours), 24 bit (16.7 million colours) and 32 bit (16.7 million colours with one of the four bytes used for transparency).

There is a direct relationship between the colour depth, resolution and storage needs, because each of these aspects of the image affect the number of bytes used.

Vector image formulae are recalculated when the image is displayed or resized so there is no loss of quality at a different scaling. Changing the size of a raster will affect image quality. If the image is to be reduced, the software needs to remove pixels. If it is to be enlarged the software will have to 'make up' extra detail or show pixels as blocks of colour.

The effects of compression on image data

The size of storage space needed by a bitmap can be reduced through **compression**, which can be lossy or lossless.

- Lossy: Lossy compression loses some of the image detail when it is compressed, as with JPG image files. Data that is not needed, such as small differences between colours and any additional metadata, is removed. Therefore editing a JPG, saving and then re-editing it often reduces the image quality.
- Lossless: Lossless compression reduces the image size without any loss of quality, although the amount that the file size is reduced by is less than with lossy compression. For example, PNG image files use techniques to analyse and reduce the image by blocking pixels of the same colour together to store their colour, number of pixels and location in the image rather than having the colour repeated for each pixel.

Key term

Compression – is a technique used to make files smaller. Compression can be a feature of a file format or can be implemented by utility software such as WinZip.

Link

See the section on 'Types of compression'.

Assessment practice 2.3

A01 A02

You have been asked to prepare a presentation that can be used as a rolling display for a trade stand at a business fair to help promote a new range of storage devices.

The presentation is to be divided into three sections, each with an introduction slide followed by slides detailing the technical content.

Section 1: Binary representation of data

Explain how binary bits can be combined together into bytes, megabytes, gigabytes and terabytes with examples of where each could be found inside a computer system.

Section 2: Text representation of data

Explain how ASCII and UNICODE are used.

Section 3: Image representation

Explain how the quality of a digital image can have an impact upon the storage size.

Plan

- What am I being asked to do?
- Do I need clarification of anything?
- Do I have any existing knowledge for the task at hand?

Do

- Am I confident that I know what I am doing and that I know what it is I should be achieving?
- I can set milestones and evaluate my progress and success at these intervals.
- I can make connections between what I am reading/researching and the task, and identify the important information.

Review

- I can explain how I approached the task.
- I can explain which elements I found easiest.
- I can explain which elements I found hardest.

D How data is organised on computer systems

You need to recognise how data is used in computer systems. In this section, you will explore the characteristics and implications of methods of organising data in computer systems, and the impact this has on computer processes.

Data structures

There are many different ways in which computer systems can structure data into different sequences for processing.

The features, applications and implications of data types used within computer systems

In this section, you will look at the features, applications and implications of different data types.

Stack

A stack is a FILO (first in, last out) data structure where items are pushed onto the stack then popped off in the reverse order. This structure is used when a subroutine is called in code to remember where the code needs to return to when the subroutine completes.

Conceptually, a stack is like a pile of plates where plates are added or removed from the top of the pile. See a visual representation of a stack in Figure 2.8.

Discussion

With a peer discuss how a stack can be used when a spreadsheet works out a complex calculation.

Queue

A queue is a FIFO (first in, first out) data structure where items are added to the queue then used in the same order. This structure is used for batch processing where items are dealt with in order, for example when printing.

Conceptually, a queue is like a conveyor belt on a supermarket checkout. See a visual representation of a queue in Figure 2.8.

Discussion

With a peer discuss how queues can be used in a modern processor.

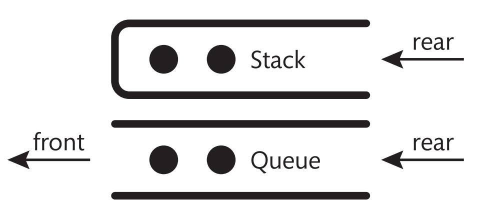


Figure 2.8: Stack vs queue

Array

An array is a type of variable used in programming where more than one item is to be held in the variable. The variable name is usually followed by subscript numbers inside brackets to define which item in the array is to be used. If you wanted to store the temperature on each day of the week you would define an array with 7 elements. Assuming the array was called Temperature, then Temperature(0) would hold Sunday's temperature, Temperature(1) would hold Monday's and Temperature(6) would hold Saturday's. There are many techniques that programmers use when coding with arrays, in particular, using iteration to loop through items in the array.

Arrays are stored in adjacent memory locations, so space is cleared and reserved for the whole array when the array is first declared in the program code.

Discussion

With a peer, discuss how an array can be used by a program to sort data.

List

A list is a data structure where an item links to another item. There are many types of list used in different ways, for example a database uses indices to define different sequences through a dataset.

Lists often use pointers in data items to point to the address of the next item in the list. An example of this technique is a binary tree which is used by database systems to store items of data as they are entered with links so that the items can be searched and brought back in an ordered sequence. For example, Fig 2.9 shows a binary tree where items were entered in this order: Jones, Mellor, Ali, Smith, McGill, Patel, Singh. Each item starts with both left and right pointers set to null (-1).

- Jones goes into memory location 1.
- Mellor goes into memory location 2.
- Mellor is compared with Jones; alphabetically more, so right null in Jones is set to 2 to point to Mellor.
- Ali goes into memory location 3.
- Ali is compared with Jones; alphabetically less, so left null in Jones is set to 3 to point to Ali.
- Smith goes into memory location 4.
- Smith is compared with Jones; alphabetically more, right not null in Jones, so pointer is followed to Mellor.
- Smith is compared with Mellor; alphabetically more, so right null in Mellor replaced by 4 to point to Smith.
- McGill goes into memory location 5.

- McGill is compared with Jones; alphabetically more, right not null in Jones, so pointer is followed to Mellor.
- McGill is compared with Mellor; alphabetically less, so left null in Mellor set to 5 to point to McGill.
- Patel goes into memory location 6.
- Patel is compared with Jones; alphabetically more, right not null in Jones, so pointer is followed to Mellor.
- Patel is compared with Mellor; alphabetically more, right not null in Mellor, so pointer is followed to Smith.
- Patel is compared with Smith; alphabetically less, so left null in Smith set to 6 to point to Patel.
- Singh goes into memory location 7.
- Singh is compared with Jones; alphabetically more, right not null in Jones, so pointer is followed to Mellor.
- Singh is compared with Mellor; alphabetically more, right not null in Mellor, so pointer is followed to Smith.
- Singh is compared with Smith; alphabetically less, left not null in Smith, so pointer is followed to Patel.
- Singh is compared with Patel; alphabetically more, so right null in Patel set to 7 to point to Singh.

Memory location	Left pointer	Items	Right pointer
1	3	Jones	2
2	5	Mellor	4
3	–1	Ali	–1
4	6	Smith	–1
5	–1	McGill	–1
6	–1	Patel	7
7	–1	Singh	–1

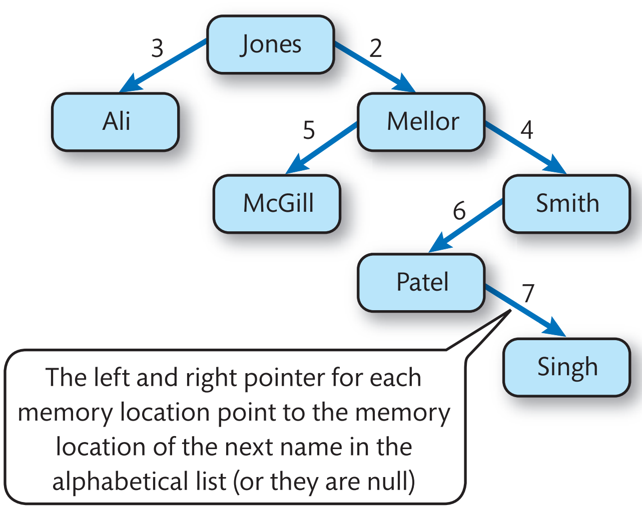


Figure 2.9: Binary tree

Conceptually, a list is a link to the next item in the list.

Discussion

With a peer discuss how a list can be used as a binary tree to structure storing and sorting data.

The use and application of data types in computer software

All computer software is created using some form of programming environment and language, such as C#. Almost all software uses variables to hold information or data when the code runs.

All programming languages allow the coder to define their variables so that each variable has a data type such as integer or string. Some programming languages also allow variables to be used with a default data type called a variant which is able to adapt to and hold any kind of data type. Variant data types carry a lot of performance-reducing overheads to cater for all the possible uses of a variable, most of which will not be needed for the variable.

Although variant data types are quicker to apply and cause fewer problems, it is much better practice to define a variable to hold the most appropriate data type for the following reasons.

- More accurate processing: Any places where the program tries to allow a wrong data type into a variable will be identified during testing, which helps to avoid errors during design or coding.
- Better use of memory: There will be no wasted space and memory will be optimised for that data type.
- Faster processing: There be no need for extra processing to examine data inside a variable before carrying out arithmetic or logical operations, which there would with a variant data type.

Data types

Programmers writing code use variables such as a name or quantity to hold data when the program is running. Best practice is to declare variables with a data type to maximise memory use and to check the expected type of data is input into the program. Variable types include:

- number when calculations are needed
- integers for whole numbers
- real for numbers with decimal places
- currency for money
- string for any mix of text and numbers
- Boolean for true or false.

The use and implications of data types in computer hardware

The different data types need to be stored and processed by computer hardware.

Storage requirements are directly linked to the data types. At the start of section C, you saw the different ways in which numbers can be stored in the memory. This also has an implication for the need for the ALU hardware circuits in the CPU to be able to correctly calculate with every possible type of data.

Indices and matrices

A matrix is a rectangular collection of numbers which can be used in mathematical operations. Matrices are used in many areas of computing including in the calculations needed for 'wire-framing', which forms the basis of virtual gaming landscapes. A 2D matrix has two rows which can be used to hold the (*x*, *y*) coordinates of a shape. These can be multiplied by other matrices to produce new (*x*, *y*) coordinates for operations such as rotation, mirroring, resizing, and so on. Indices are used to identify items in a matrix.

Matrix representation in computer systems

In this section, you will explore how matrices are represented in computer systems (including their relationship with arrays), mathematical operations using matrices and how to order a matrix.

Matrices and arrays

There are close relationships between matrices and arrays. Many programming languages store an array used by a programmer in their code as a matrix in the memory. Similarly, a programmer wishing to implement matrix techniques in their code would declare and use arrays to hold the data for such calculations.

An array can be single-dimensional (1D), two-dimensional (2D) or multi-dimensional, for example 3D. See Figure 2.10 for a visual representation of 1D, 2D and 3D arrays. There will be an equivalent number of subscripts to the dimension, so a 1D array has one subscript, 2D has two subscripts, 3D has three subscripts and so on.

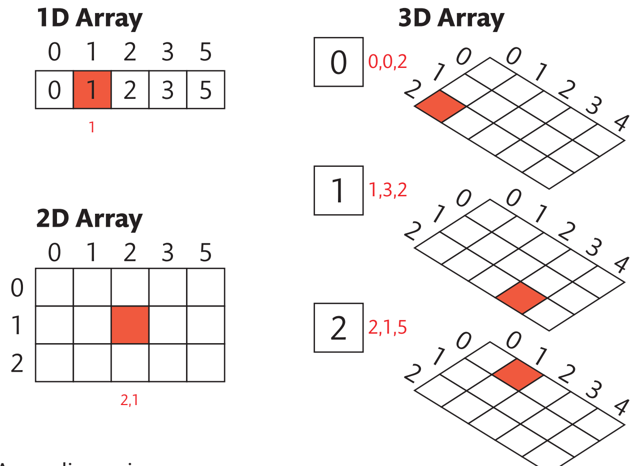


Figure 2.10: Array dimensions

Key term

Inversion – means flipping a value. In a Boolean inversion 0s are inverted into 1s and 1s are inverted into 0s. In a mathematical inversion a fraction is turned upside down so 2 would invert to $\frac{1}{2}$ and $\frac{1}{4}$ would invert to 4. This is also called finding the reciprocal of a number.

Mathematical operations

Mathematical operations include adding or subtracting matrices of the same size, multiplying matrices of different (or same) sizes and **inversion**. These mathematical operations can be used for manipulating shapes and solving simultaneous equations.

- Manipulating shapes is very easy using matrices. A matrix of (*x*, *y*) coordinates is multiplied by a 2×2 matrix with each pair of (*x*, *y*) coordinates producing a pair of transformed coordinates (see the worked example).

Worked Example

A shape is reflected in the *y*-axis using matrix multiplication. This example is shown in Fig 2.11 Reflection in the *y*-axis using matrix multiplication.

Step 1: Draw a shape on graph paper using straight lines with 5 corners. Write the (*x*, *y*) coordinates onto paper as a table with two rows

x	1	2	3	4	2
y	2	5	3	7	7

Step 2: Use matrix calculations to multiply your (*x*, *y*) coordinates by this 2×2 matrix

(a)	–1	0	(b)
(c)	0	1	(d)

Step 2a: Calculate the upper number in the answer. The formula for this is $ax + by$ with *a*, *b* shown above and (*x*, *y*) the first pair of coordinates in your table

$-1 \times 1 + 0 \times 2 \quad = \quad -1 + 0 \quad = -1$

Step 2b: Calculate the lower number in the answer. The formula for this is $cx + dy$

$0 \times 1 + 1 \times 2 \quad = \quad 0 + 2 \quad = 2$

Step 3: After repeating steps 2a, 2b for the rest of your (*x*, *y*) pairs you have the new (*x*, *y*) coordinates for your shape mirrored in the *y*-axis.

x	–1	–2	–3	–4	–2	–1
y	2	5	3	7	7	2

X	1	2	3	4	2	1
Y	2	5	3	7	7	2

Original XY coordinates

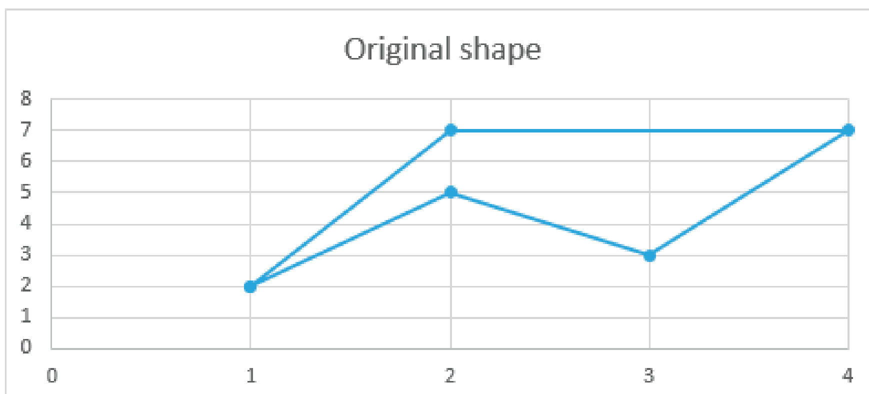
-1	0
0	1

2x2 matrix

Rotates 180 degrees

X	-1	-2	-3	-4	-2	-1
Y	2	5	3	7	7	2

Transformed XY coordinates



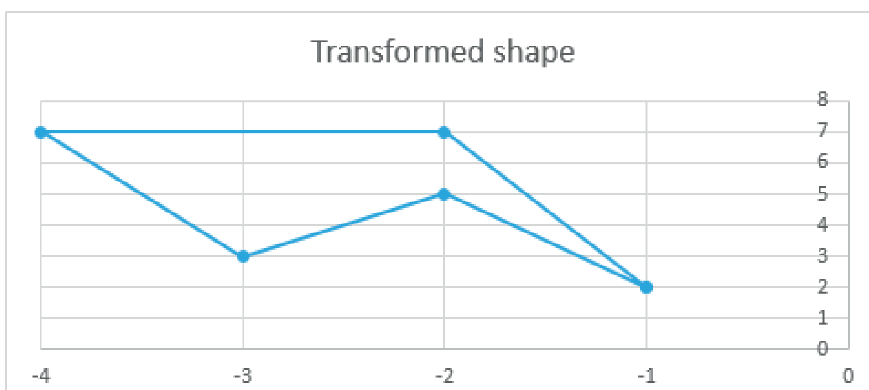


Fig 2.11: Reflection in the *y*-axis using matrix multiplication

X	1	2	3	4	2	1
Y	2	5	3	7	7	2

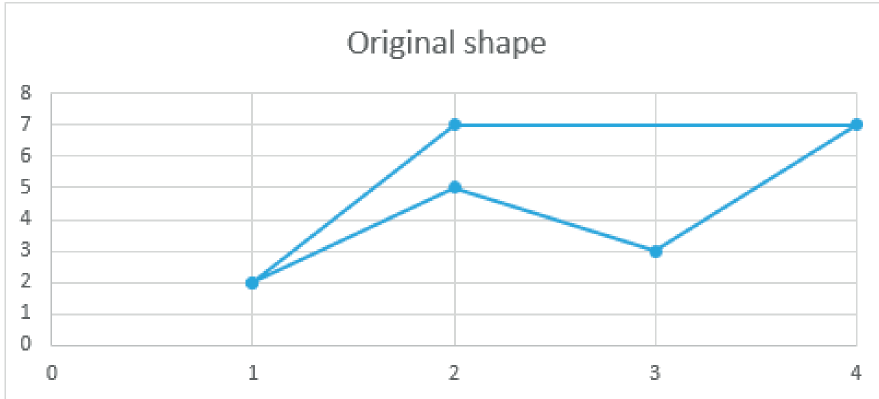
Original XY coordinates

-1	0
0	-1

2x2 matrix
Rotates 180 degrees

X	-1	-2	-3	-4	-2	-1
Y	-2	-5	-3	-7	-7	-2

Transformed XY coordinates



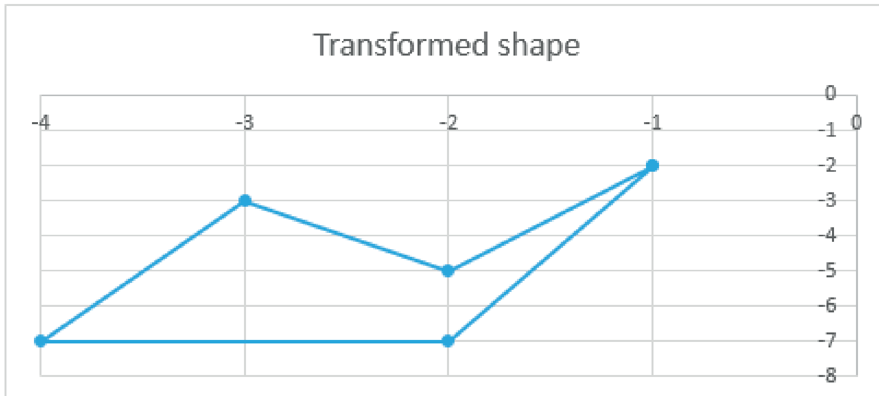


Fig 2.12: 180 degree rotation using matrix multiplication

Solving simultaneous equations is a little more complex (see the worked example).

Both of the equations terms need to be in the right sequence:

<number#1>X <+ or –> <number#2>Y = <number#3>

so, if one of the pair of simultaneous equations was $3x = 26 - 2y$ it would need to be manipulated into:

$3x + 2y = 26$

If the other of the pair of simultaneous equations was $28 = 10y - 2x$ it would need to be manipulated into:

$-2x + 10y = 28$

The simultaneous equations can now be represented as a matrix multiplication:

$$\begin{array}{l}(a)\\(c)\end{array}\begin{pmatrix}3 & 2\\-2 & 10\end{pmatrix}\quad\begin{array}{l}(b)\\(d)\end{array}\times\begin{pmatrix}x\\y\end{pmatrix}=\begin{pmatrix}26\\28\end{pmatrix}\quad\begin{array}{l}(ax+by=26)\\(cx+dy=28)\end{array}$$

This matrix can be thought of as simple algebra:

Matrix_2×2 × Matrix_1×2 = Matrix_Known

We are trying to find out the values of Matrix_1×2; the values of the other matrices are already known, so the algebraic formula needs to be manipulated so that Matrix_1×2 is the subject:

Matrix_1×2 = Matrix_Known / Matrix_2×2

To divide Matrix_Known by Matrix_2×2 we need to calculate the inversion of Matrix_2×2 which can then be multiplied by Matrix_Known (we want to divide a number by 4, we can multiply it by the inversion of 4 which is $\frac{1}{4}$).

Theory into practice

Draw a shape with 5 corners on graph paper using straight lines. Write the (*x*, *y*) coordinates onto paper then use matrix calculations to multiply these by the 2×2 matrix shown in Fig 2.12 to calculate the new (*x*, *y*) coordinates for your shape rotated by 180 degrees.

Plot your new (*x*, *y*) coordinates onto graph paper to check your calculations.

Use Excel® to repeat this exercise. Notice how the first (*x*, *y*) coordinates are repeated in the last column of the spreadsheet so Excel® draws the last line in your chart. You can select the cells where you want the new XY coordinates then use the MMULT() array formula to calculate them. The first array in your MMULT() formula is the 2×2 matrix, the second gives your (*x*, *y*) coordinates. The MMULT() formula completes with Ctrl+Shift+Enter.

Worked Example

A mobile phone shop has two offers that interest you, the MegaSaver (£30 per month then 20p per minute) and the SuperBundle (£50 per month then 10p per minute). The MegaSaver will be cheaper for a small number of calls and the SuperBundle cheaper for a large number of minutes. At which point does the SuperBundle become cheaper than the MegaSaver?

Step 1: Convert the deals into simultaneous equations

MegaSaver	Cost = £30 + (£0.20 × Minutes)	$C = 30 + 0.2M$
SuperBundle	Cost = £50 + (£0.10 × Minutes)	$C = 50 + 0.1M$

Step 2: Convert the simultaneous equations into this sequence $A + B$ = <number>

MegaSaver	Cost - (£0.20 × Minutes) = £30	$C - 0.2M = 30$
SuperBundle	Cost - (£0.10 × Minutes) = £50	$C - 0.1M = 50$

The calculations will work as shown above, but are easier with the decimals removed by multiplying the equations by 10:

MegaSaver	$10C - 2M = 300$
SuperBundle	$10C - M = 500$

Step 3: Convert the simultaneous equations into a matrix multiplication

$$\begin{matrix} 10 \times & -2 \\ 10 & -1 \end{matrix} = \begin{matrix} 300 \\ 500 \end{matrix}$$

Step 4: Invert the 2×2 matrix

Step 4a: Calculate $1/(ad - bc)$

$$\frac{1}{ad - bc} \qquad \frac{1}{-10 - (-20)} \qquad \frac{1}{-10 + 20} \qquad \frac{1}{10}$$

Step 4b: Swap a and d, change the signs of b and c

$$\begin{matrix} -1 & 2 \\ -10 & 10 \end{matrix}$$

Step 4c: Multiply each term by $1/(ad - bc)$ which is 0.1

$$\begin{matrix} -0.1 & 0.2 \\ -1 & 1 \end{matrix}$$

Step 5: Multiply the inverted 2×2 matrix by the 1×2 matrix

$$\begin{matrix} -0.1 & 0.2 \\ -1 & 1 \end{matrix} * \begin{matrix} 300 \\ 500 \end{matrix}$$

Step 5a: Calculate the upper number in the answer. The formula for this is $ax + by$

$-0.1 * 300 + 0.2 * 500 = -30 + 100 = 70$

Step 5b: Calculate the lower number in the answer. The formula for this is $cx + dy$

$-1 * 300 + 1 * 500 = -300 + 500 = 200$

Step 6: The calculated answers show that at the point where the SuperBundle becomes cheaper than the MegaSaver the cost (C) will be £70 and the number of minutes (M) is 200.

Discussion

With a peer or in a small group, discuss what transformations can be performed by multiplying different 2×2 matrices by a shape defined as a matrix of (*x*, *y*) coordinates.

Ordering a matrix

Memory locations are in a simple sequence and any memory usage to hold a matrix is applied to that sequence. The way in which a matrix can be held in memory will be either in row-major or column-major order.

With **row-major order,** the top row is held in the memory in the same sequence as the top row of the matrix, then the next row follows in the memory, and then any other rows follow afterwards. This is shown in the following example:

13	31	1	63	60
76	47	4	49	8

13	31	1	63	60	76	47	4	49	8

With **column-major order**, the first column is held in the memory, then the next column follows in the memory, and then any other columns follow afterwards. This is shown in the following example:

94	74	63	91
89	76	46	51
21	65	46	61

94	89	21	74	76	65	63	46	46	91	51	61

Assessment practice 2.4

A02 A03 A04

You are a member of a team that has been selected to produce two training sessions for students in their first year of programming at local centres.

The first session is to describe and explain four of the data structures commonly found in computer systems with an example of how each of the following could be used:

- stack
- queue
- array

The second session is to describe and explain how matrices can be used in computer systems to solve problems and to manipulate graphics.

Produce a set of handouts which:

- demonstrate how to solve simultaneous equations using matrices for a real-world problem
- use a spreadsheet model to demonstrate how shapes can be manipulated.

Plan

- What resources do I need to complete the task? How can I get access to them?
- How much time do I have to complete the task? How am I going to successfully plan my time and keep track of my progress?
- What aspects of the task do I think will take the most/least time? How will I balance these?

Do

- What strategies am I employing? Are these right for the task? Are they working? If not, what do I need to do to change this?
- I understand my thought process and why I have decided to approach the task in a particular way. I can explain this reasoning when asked.
- I can identify when I have gone wrong and adjust my thinking/approach to get myself back on course.

Review

- I can draw links between this learning and prior learning.
- I can explain what skills I have employed and which new ones I have developed.
- I can make informed choices based on reflection.

E How data is transmitted by computer systems

Communication is everywhere in the modern world, with data being transmitted between millions of computer systems 24/7. You need to understand the concepts, processes and implications of data transmission in and between computer systems, and these will be covered in this section.

Transmitting data

There are many methods used to transmit data in and between computer systems. In this section, you will explore the types of communication channel, methods of connecting devices and transmitting data and different types of transmission, along with the protocols used to control data transmission.

Types of communication channel

There are a number of different communication channels to choose from depending on the situation.

- **Simplex** can only be in one direction. Examples include a sensor sending temperature data into a control system or a television transmission.
- **Half-duplex** must be used in both directions, but only in one direction at a time. Examples include data in an old bus network topology or using walkie-talkie radios.
- **Full-duplex** uses both directions at the same time. Examples of this include current network cabling or voice telephone calls.
- **Point-to-point** is when there is direct communication between two computer systems.
- **Multi-drop** is when there is communication from one computer to several other computer systems.

Discussion

With a peer or in a small group, discuss how an organisation can make use of each type of communication channel.

Methods of connecting devices and transmitting data across and between computer systems

Devices can be connected using wired or wireless technologies in order to transmit data across and between computer systems.

Wired connections

Wired network connections could be either using a **USB** cable or a network. USB is usually used for short distance point-to-point transmissions, such as synchronising data between a tablet and a PC which might be standalone or attached to a network. Wired network connections are standard for workstations in organisations and are used for a wide variety of tasks including data entry.

Wireless connections

Wireless connections could be either using **Bluetooth** or WiFi technologies. Bluetooth is usually used for short distance point-to-point transmissions, such as synchronising data between a tablet and a PC which might be standalone, attached to a network or between mobile devices. WiFi connections are standard for laptops used in organisations where the user needs to log into the network. Once connected, a laptop can be used as a workstation.

Key terms

USB – a universal serial bus is a widespread socket or port found on computers, televisions, DJ decks, satellite-navigation systems and many other devices.

Bluetooth – this is a short-range wireless standard that can connect a mobile phone to a headset or transfer data such as music between devices.

The selection of connection methods to fulfil specified tasks and functions

For most organisations, the selection of connection method is a choice between WiFi or wired (cables) which could be either **copper** or **optical**.

A wireless WiFi connection is the obvious choice for tablets and laptops which need to connect to the main network. In some industries, it is common to use laptops with a wireless network connection both remotely and in the office so that it is easy to use laptops during meetings.

Cabled copper connections are a common choice for PC workstations as they are cheap, have good data throughput speed and are easy to install. Optical connections are common for **switch** to switch communications as their excellent speed justifies the extra cost but they are more difficult to install as they are harder to bend around corners.

Key terms

Copper – in this context, this is the name given to cabling using any metal to conduct electrical signals.

Optical – in this context, this is cabling that uses thin glass fibres to pass information using flashes of light. Optical cabling usually transmits data faster than copper cabling.

Switch – a box with a number of ports for network cables to plug into. The circuits inside the switch send any data they receive to the port holding the cable that connects to the device where the data is needed.

Key terms

Synchronised – computer systems each have clocks inside their circuits to provide timing pulses when moving data. A synchronised signal needs the clocks at each end of a data transmission to beat together.

Asynchronous – computer systems each have clocks inside their circuits to provide timing pulses when moving data. Asynchronous signals do not need the clocks at each end of a data transmission to beat together.

Asynchronous and synchronous data transmission

When computer systems communicate data, they send each byte of data as 1s and 0s (bits) which then get reconstructed back into bytes by the receiving system. Using a copper connection, a 1 could be a very quick electrical pulse of 5 volts with a 0 at no volts. Timing is very important, as a transmission of 111 would be a pulse of 5 volts lasting 3 times as long as a transmission of a single 1 (this is illustrated in Figure 2.13).

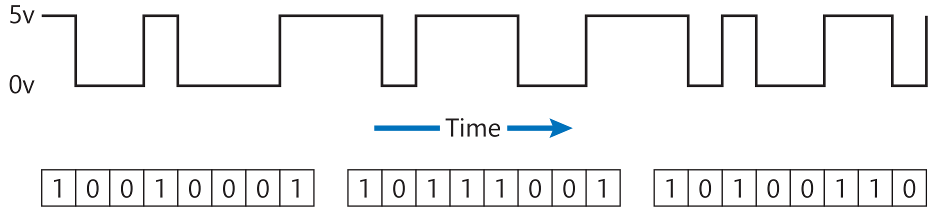


Figure 2.13: Data transmissions

With a synchronous data transmission, the device at each end of the connection has a **synchronised** clock. This means that the timings are easy to implement in components such as a computer motherboard.

Asynchronous data transmissions are very common in modern communications, as few systems need to synchronise their clocks when data is sent as packets. Each packet travels independently to the receiving system which is individually synchronised when it arrives (see Figure 2.15)

Parallel and serial transmission

Any computer data transmission will send data bytes between computer systems. A byte is made from 8 bits which can either be sent in parallel or serially. A parallel connection has a number of connections, side by side, usually a multiple of 8, so that the byte(s) can travel in a single transmission. With a serial connection, the bits in a byte travel after each other, one bit at a time.

A **parallel transmission** needs a connection for each bit. This would be a separate wire in a cable or a separate track in a circuit such as a motherboard. Parallel cables for connecting a printer to a PC were developed in the 1970s and were widely used until USB took over.

Serial transmission requires a simpler cable with just a pair of wires being needed to send a bit. In the days of parallel printer cables, a slow standard named RS232 was used. USB has been the standard since the late 1990s for serial transmissions and is now used everywhere. USB has been developed further into faster variants with USB 3.0 being the current standard.

Serial cables are a lot quicker than parallel cables. This is because there are timing issues in parallel cables because a complete byte must be received before the next can be transmitted, which is more difficult with longer cables as the bits arrive at slightly different times in the parallel wires.

PAUSE POINT

Why are parallel transmissions not used everywhere? Surely sending data byte(s) by byte(s) rather than bit by bit must be faster? What are the technical reasons behind the superiority of serial communications?

Hint Research Centronic printer cables and RS232 serial cables. Compare their technical specifications.

Extend Search for ‘why is parallel slower than serial’ to read around the subject. In particular, find out how hard disc connections in a PC have developed through the years.

Use of packets in transmitting data

Data transmitted between systems is usually divided into packets, which are individually routed to their destinations and are rebuilt into data by the receiving system. The structure of a data packet varies according to the protocols and network standards used for the transmission, but will normally include a header, data payload and footer/trailer (see Figure 2.14).

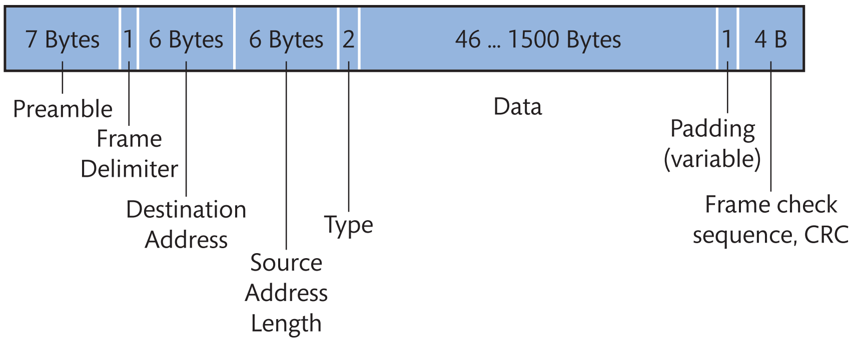


Figure 2.14: Data packet

The header section will include a preamble which states where the packet is going to (destination), where it came from (source), protocol and packet number (see Figure 2.15). The preamble is a regular 1010 so the receiving system can synchronise timing with the packet with a delimiter (11) at the end of the preamble to mark the start of the packet. The source address is needed so that the destination system can request a resend if a corruption occurs.

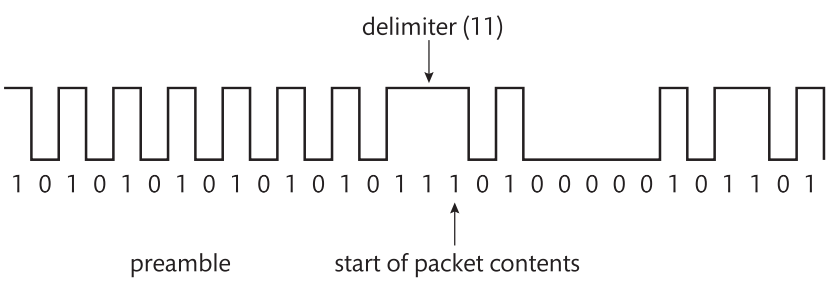


Figure 2.15: Data packet preamble

The data payload is by far the largest part of the packet. This is the data carried by the packet.

An error check is included at the end of the packet to identify any corruption that may have happened during transit. The source system uses the bytes in the packet to calculate a number which is placed at the end of the packet. The destination system carries out the same calculation on what it receives, and this is compared with the check value. If an error is detected, it triggers an automatic request for retransmission.

Packet switching

Packet switching is used in modern communication systems for the internet and telephone calls. Data is split up into packets rather than being sent in a continuous stream by the router. Not all the packets will necessarily follow the same route. At the receiving end packets are reassembled into their original order. Old telephones used circuit switching where a direct connection was opened up for each phone call, but these old mechanical exchanges have now been replaced with digital packet-switching systems. Packet switching is more efficient than circuit switching: it allows data to be sent by multiple routes and prevents large data transfers 'hogging' a circuit.

Protocols used to govern and control data transmission

Protocols are the rules agreed when computer systems transmit data. Without a protocol, the receiving computer would not be able to make any sense of the transmission.

The protocol determines the speed of transmission, error checking, the type of information, the packet structure, how data is forwarded and everything needed to successfully transfer data.

TCP/IP are commonly used protocols that allow you to surf the internet. The internet protocol (IP) enables a browser such as Internet Explorer to find a web page. The transmission control protocol (TCP) handles the movement of data between the web page and the browser.

The features, applications and implications of encryption

Encryption is used to protect sensitive data by scrambling the contents of a file so that it cannot be opened or read by unauthorised users. Encryption is a technique that is used to transmit and store sensitive information.

Simple encryption ciphers

Two simple encryption ciphers are the Caesar cipher and the Vigenère cipher.

Caesar cipher

The Caesar cipher shifts characters a number of places. For example, a start letter of J would mean that A is written as J, B as K and so on.

This method of communicating private information was used by the famous ancient Roman, Julius Caesar, to encrypt messages that he sent. Although this method of encryption is very simple, it was effective because the encrypted messages looked as though they were written in a foreign language.

Encrypting a message using a Caesar cipher can be done using a pair of wheels, where one of the wheels is moved so that the starting letter lines up with 'A' and the encrypted letters can now be read off.

Figure 2.16: The Caesar cipher works by shifting characters

A computer system using a Caesar cipher could use an encryption key, rather than a pair of wheels, to define how many times to shift each letter. For example, if the encryption key is 00000101 in binary, this would be 5 so 'B' would be encrypted to 'G', which is a shift of five characters.

Vigenère cipher

The Vigenère cipher uses a table to look up the ciphered characters.

This cipher dates back to 1553, but is named after Blaise de Vigenère who popularised it three hundred years later. Vigenère ciphers were used in many situations, such as the American Civil War.

This cipher is an extension to the Caesar cipher which uses a keyword to vary the shifting characters of the original message.

Encrypting a message using a Vigenère cipher can be done using a lookup table, as shown in Figure 2.17. If the keyword was BTEC, the first letter of the message would be looked up from the column of that letter in the B row, the second letter in the T row and so on. For example, MEETYOULATER with a key of BTEC would be NXIVZHWMTXGS.

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
2	B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
3	C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
4	D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
5	E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
6	F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
7	G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
8	H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
9	I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
10	J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
11	K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
12	L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
13	M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
14	N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
15	O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
16	P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
17	Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
18	R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
19	S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
20	T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
21	U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
22	V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
23	W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
24	X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
25	Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
26	Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Figure 2.17: The Vigenère cipher works using a lookup table

Encryption used in computer systems

Both Caesar and Vigenère ciphers are easily cracked by using language analysis techniques to find the letters which naturally occur more frequently, or even by a simple brute force method where the message is repeatedly output with characters shifted a place until a coherent message appears. For a simple Caesar cipher this would always be less than 26 repetitions.

Much more sophisticated techniques are used in computer-based encryption. There is a lot of processing available to the sending and receiving systems, so complex **algorithms** are easily implemented. These algorithms use one or more keys to calculate encrypted characters, rather than sampling by shifting along the alphabet.

Key term

Algorithm – a solution to a programming problem that is used to help plan the code that will be written to solve the problem.

Symmetric key encryption

The symmetric key encryption method uses the same key for encrypting the data as for decrypting. The algorithms using this method may transform the key before encryption to make the cipher more difficult to crack.

The major disadvantage of this method is that both the sender and receiver need the key.

Public key encryption

Public (also known as asymmetric) key encryption uses two keys: a public key and a private key. This technique is more popular than symmetric keys as the processing can happen automatically.

When an encrypted message is to be sent to a computer, the receiving system sends a public key to the sender. The sending system uses the public key to encrypt the message, but only the private key (known only to the receiver) can be used to decrypt the message.

Both the public and private keys are calculated from the same beginning in a way that cannot be reverse engineered to find one key from the other.

Public key encryption is used when you visit a secure (https) website to carry out an operation such as paying for products or services or looking at your bank account details.

Link

For more about encryption methods, see *Unit 7: IT Systems Security and Encryption*.

Types of compression

There are two types of compression, **lossy** and **lossless**. The main difference between the two types of compression is that lossy compression involves the loss of some detail in the file when it is then decompressed, whereas lossless compression involves no loss of detail so, when decompressed, you get back an exact version of the original.

Link

For more information on the two types of compression, see 'The effects of compression on image data'.

The applications and implications of data compression

Data compression is a technique that is used to reduce the size of files. This is useful as it reduces how much storage space is needed for files and also improves transmission speeds between devices because smaller files will travel through a communication link faster than larger files.

There are many methods of compressing data, but the main ways are through the choice of file type or by using a utility to compress files.

File type

The file type chosen can have a large impact on file size. Music and video files can be very large in their original form but can be reduced in size quite dramatically by saving them in an appropriate file format such as MP3 for music or MP4 for video (which are both lossy compressions). Using these smaller file sizes for music and video makes both storing and transmitting them much easier and quicker compared with a non-compressed format such as WAV for music or AVI for video.

Link

For more about image file types, see 'Image representation'.

Compression utility

Windows® comes with an included compression utility which produces a zip file from file(s) and/or folder(s). This is an effective method of compressing documents and is compatible with most other computer systems.

PAUSE POINT Symmetric key encryption is used a lot less than public key encryption. Why is this? How does public key encryption actually work?

Hint Reread the section in this unit on this subject, then carry out some internet research on the keys used by each of these encryption methods, and find out what the keys do.

Extend Public key encryption uses two keys, a public key and a private key. How are these related? Why is it not possible for another system that has been able to capture the public key to decrypt data sent out using that key?

Other compression utilities such as WinRAR are available, which has the added feature of password protection. If the compressed data is sent to another computer system, a compression utility that is compatible with the compressed file will be needed to be able to decompress and then use the data.

To compress or decompress data files requires processing power and memory, and so will take some time to complete. This is not usually an issue, but might become one on an old, low specification computer system.

Error detection

Error detection is crucial to finding any problems in the thousands of millions of 1s and 0s (the data) held in and transmitted through a computer system every day.

Methods used to detect errors in data transmission

There are a number of different methods of error detection which are used in different parts of the computer system. Data is often checked by more than one error detection method to reduce the number of errors.

Parity schemes

Parity schemes are used in the RAM to check the accuracy of each byte. A byte has 8 bits and using a parity scheme another bit is allocated to the byte to help check for errors. It does this by counting the number of bits set to 1 in the byte then using the parity bit to make the number of 1s an odd or even number.

An odd parity scheme would use the parity bit to make the number of 1s in the byte into an odd number. For example the parity bit for 01101110 would be 0 because there are 5 (an odd number) bits set to 1 already and for 01101111 would be 1 because there 6 (even number) bits set to 1 so another 1 makes it an odd number of 1s.

If 01101110 0 (odd parity) was transmitted and a corruption occurred, 01101010 0 could be received. As this has an even number of 1s (i.e. 4), the receiving device would detect the error and request a retransmission.

Similarly, an even parity scheme would use the parity bit to make the number of 1s in the byte into an even number, for example the parity bit for 01101110 would be 1 and for 01101111 would be 0. However, parity schemes are not very good at detecting multiple bit errors, since if two bits are received as 0s rather than 1s, the parity will remain correct, despite the error.

Checksum

Checksums are used at the end of a data transmission to check for errors. A calculation is carried out on the transmission before sending and the same calculation is worked out by the device that receives the data. Any difference in the checksum indicates an error so a request is sent back to the sending device requesting retransmission.

A simple checksum treats each byte in the data transmission as a number and adds them together.

Repetition schemes

Repetition schemes simply consist of transmitting the data more than once. Any difference in the repeated transmissions indicates an error.

Cyclic redundancy check (CRC)

CRC is a type of checksum placed at the end of a data packet, which uses a calculation based upon treating the packet as a number. This number has a division applied to it and the remainder from the division is used as the CRC.

Theory into practice

An ethernet frame (packet) typically has a size of 1542 bytes, with 4 bytes for the error checking. Use a spreadsheet to calculate the largest number that can be held in 4 bytes.

What are the advantages of using CRC over a simple checksum to calculate the error check?

The concepts, implications and applications of error detection

Every time data is transmitted or stored there is a risk of data **corruption**, so error detection is built into computer systems and protocols are set up to try to eliminate these problems. Generally this involves adding numbers which are mathematically related to the original data (such as a checksum or CRC) so they can been recalculated at the receiving end. However, calculating and recalculating these values adds to the processing overhead. These error detection schemes also decrease the real data transmission rate. They involve sending additional data which is not part of the actual data being transmitted: once at the receiving end, this error detection data is checked and discarded. Thus the overall efficiency of the data transmission system is reduced.

It is probably impossible to ever be certain of finding 100 per cent of any errors as there will always be a possibility that separate corruptions could 'correct' an error that has been detected.

Key term

Corruption – in the context of data, is when one or more bits get misunderstood by the receiving computer system during a data transmission.

PAUSE POINT The cause of a corruption in data varies for different media, as the corruption has to affect the signal inside the medium. For example, it could be caused by an electrical cable being placed close to a network cable. What are the possible causes of corruption for different media types?

Hint What is the signal used to transmit data in (a) cat 6 copper cable (b) fibre optical cable (c) WiFi and (d) laser point-to-point network connections?

Extend What would cause a corruption for each of these transmission methods? Which transmission method is the most reliable and why?

Error correction

Once errors have been detected they need to be corrected, and there are a number of error correction methods.

Commonly used error correction systems

This section looks at two commonly used error corrections systems – automatic repeat request and forward error correction.

Automatic repeat request (ARQ)

The ARQ method of responding to detected errors is to request a retransmission from the sending computer system. Any bad packets (those which contained errors) that were received are not used and are replaced by the resent data.

The protocol controlling data transmission may require the receiver to confirm that data packets have been successfully received, with the sender retransmitting any packets that have been identified as having errors or that have not been acknowledged by the receiver.

Forward error correction (FEC)

FEC is a technique whereby extra data is sent with the transmission so that the receiving system can rebuild bad data. There are sophisticated techniques used to implement FEC that involve repetition and extra parity bits.

Link

Look back at the section on 'Error detection' for more about parity and repetition schemes for error detection.

The concepts, implications and applications of error correction systems

Error correction systems are put in place to respond to any problems in the logical accuracy of transmitted data and to respond by retransmitting any bad data.

Error correction can only be as good as the detection systems used to identify errors in the first place.

Assessment practice 2.5

A01

Use a seven letter word to demonstrate how to use a Vigenère cipher to encode a word. Write a brief description of how you encrypted the word.

Produce a guide to methods used by computer systems to detect errors from data transmissions.

You should:

- explain how a computer system can use even parity checks
- describe and compare how ARQ and FEC are used in error correction systems.

Plan

- How will I approach the task?
- Are there any areas I think I may struggle with?
- How confident do I feel in my own abilities to complete this task?

Do

- I understand my thought process and why I have decided to approach the task in a particular way. I can explain this reasoning when asked.
- I can identify when I have gone wrong and adjust my thinking/approach to get myself back on course.
- What am I struggling with? Do I know how to overcome this?

Review

- I can explain how I approached the task.
- I can explain which elements I found easiest.
- I can explain which elements I found hardest.

F The use of logic and data flow in computer systems

All the circuits inside a computer system use digital logic. This section explains the uses, applications and interpretation of logical processes. Diagrams are used to represent the data flow and relationships in and between computer systems to help communicate their relationships and workings.

Boolean logic

Boolean logic is based on logic gates. These are tiny electronic circuits that can accept input(s) and produce an output according to the logic of the input(s). There are hundreds of millions of these circuits inside components such as the RAM, CPU and other chips used in a computer system.

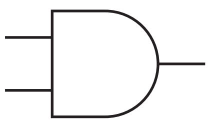
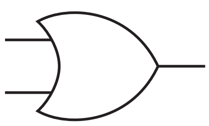
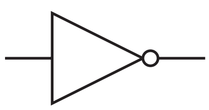
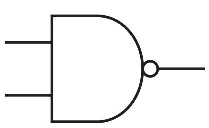
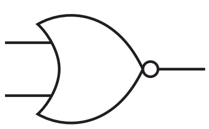
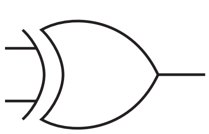
Datename	Symbol	Notification	Truth table
AND		$F = A \wedge B$ or $F = A.B$	A B AB 0 0 0 0 1 0 1 0 0 1 1 1
OR		$F = A+B$	A B A+B 0 0 0 0 1 1 1 0 1 1 1 1
NOT		$F = \overline{A}$	A F 0 1 1 0
NAND		$A = \overline{A.B}$	A B AB 0 0 1 0 1 1 1 0 1 1 1 0
NOR		$F = \overline{A+B}$	A B F 0 0 1 0 1 0 1 0 0 1 1 0
XOR		$F = A \oplus B$ $F = A \neq B$ $F = \overline{A}B + A\overline{B}$	A B F 0 0 0 0 1 1 1 0 1 1 1 0

Figure 2.18: Truth tables

Truth tables

The outputs from a logic gate can be shown using a truth table. All the possible input combinations are shown to the left of the truth table as 0s and 1s, arranged in a binary numerical sequence. An output of 0 or 1 is shown for each input combination (see Figure 2.18) according to the logic of the gate.

- AND gate has two or more inputs producing 1 as output when all inputs are 1.
- OR gate has two or more inputs producing 1 as output when any of the inputs are 1.
- NOT gate has one input which is inverted as the output.
- NAND gate (AND with NOT) has two or more inputs producing 0 as output when all inputs are 1.
- NOR gate (OR with NOT) has two or more inputs producing 1 as output when all inputs are 0.
- EXCLUSIVE OR (XOR, EOR) gate has two inputs producing 1 as output when the inputs are different.

The use, application and interpretation of Boolean logic to identify data flow and solve problems

The starting point for using Boolean logic to identify data flow and solve problems is to produce a truth table showing all the inputs into the circuit and all of the outputs. Every output where a 1 is needed has the input combination defined as a Boolean term. These Boolean term definitions can then be used to design the logical circuit.

Inside the CPU is the ALU (arithmetic and logic unit) where calculations are worked out. An example of a simple calculation is to add two numbers together. These numbers will be inside the ALU as binary bytes, simplified into 8 BITs, as shown in Figure 2.19.

	128	64	32	16	8	4	2	1	
	0	0	1	1	0	0	1	1	A (51)
+	0	0	0	1	0	1	1	1	B (23)
	0	1	0	0	1	0	1	0	Sum (84)
	0	0	1	0	0	1	1		Carry

Figure 2.19: Binary addition

Notice that the first calculation in the 1 column adds together the least significant digits from A and B with no carrying involved, as all the other calculations need to include the number carried from the previous pair of digits. The circuit used for this is called a **half adder (**see Figure 2.20: Half adder).

The truth table for the first calculation is shown in Table 2.6.

Table 2.6: Half adder truth table

A	B	Carry	Sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

The carry only has 1 when both A and B are 1, so can be defined as:

Carry = A AND B

The sum only has 1s when A and B are different, so can be defined as:

Carry = A XOR B

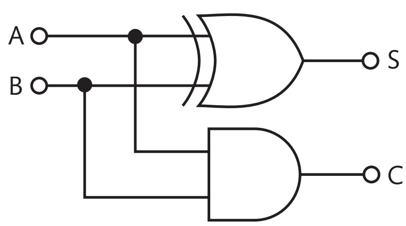


Figure 2.20: Half adder

Notice that the rest of the calculations in Figure 2.20 add together digits from A and B and the number carried from the previous pair of digits. The circuit used for this is called a **full adder**. The truth table for this is shown in Table 2.7.

Table 2.7: Full adder truth table

A	B	Carry in (Cin)	Carry out (Cout)	Sum
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

The carry out is 1 for several combinations of A and B that have carry in equal to 1, so needs to be defined as a term for each 1, where the terms have a logical OR relationship.

Each term uses AND between every input with any input at 0 inverted to make it into 1 for the AND operation to produce the 1 output.

Boolean algebra simplifies the writing of the expression by using a line over a term for NOT, a full stop for AND and a plus for OR:

Carry out = A.B.Cin + A.B.Cin + A.B.Cin + A.B.Cin

This expression needs to be simplified before a logical circuit can be designed to implement the carry out part of the half adder. One method is to use a **Karnaugh map** to reduce the terms.

A Karnaugh map is a grid with the possible inputs around the outside and outputs indicated using a symbol such as x. The inputs are arranged so that just one part of a term is different from the one next to it, as shown in Figure 2.21.

	A.B	$\overline{A}$.B	$\overline{A.B}$	A.$\overline{B}$
Cin	x	x		x
$\overline{C}$in	x			

Figure 2.21: Karnaugh map for Carry out

From the Karnaugh map, we can see in the first column that Cin makes no difference to A.B, so we now have:

Carry out = A.B + $\overline{A}$.B.Cin + A.$\overline{B}$.Cin

This can be simplified to:

Carry out = A.B + Cin ($\overline{A}$.B + A.$\overline{B}$)

Because $\overline{A}$.B + A.$\overline{B}$ is the Boolean expression for an exclusive OR gate, the carry out circuit needs to OR together the result from A AND B (A.B) with the result from Cin AND the exclusive OR of A with B.

Sum = $\overline{A.B}$.Cin + $\overline{A}$.B.$\overline{C}$in + A.$\overline{B}$.$\overline{C}$in + A.B.Cin

Sum = Cin($\overline{A.B}$ + A.B) + $\overline{C}$in($\overline{A}$.B. + A.$\overline{B}$)

Sum = Cin($\overline{A}\oplus\overline{B}$) + $\overline{C}$in(A$\oplus$B)

Sum = Cin$\oplus$A$\oplus$B

We now have the Boolean expressions needed for a full adder circuit (see Figure 2.22).

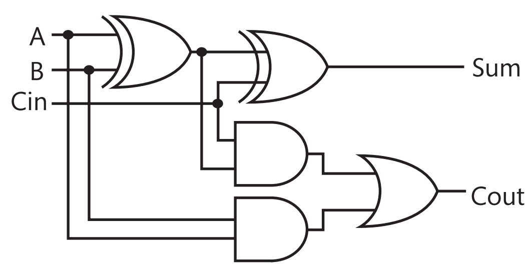


Figure 2.22: Full adder

The use, application and interpretation of Boolean logic to identify logical structures, represent data flow and solve problems

Hardware designers use Boolean logic to help them to identify logical structures which can then be made into circuits. In the early days of computing, these circuits were designed using drawing boards but, during the last few decades, computers have been used to make the design of tiny complex logical circuits possible.

Boolean logic can be used to represent data flow and solve problems with logical structures. The inputs into and outputs from a circuit can be defined in a truth table that has a column for each input and each output, and where every output is used to define a term in the Boolean expression for that column. As already seen, these expressions can be simplified to give the final solutions.

Flow charts and system diagrams

There are many ways in which diagrams are used to help explain how a computer system works.

Flow charts

Flow charts and other diagrams use arrows to represent data flow, where the direction is shown by the arrow. Flow charts also use a collection of shapes that are used to identify the functions of different parts of the system.

Link

Read the section on 'Flowchart and use of standard symbol conventions', in *Unit 4: Software Design and Development Project* to learn about the different shapes and arrows that are used in flow charts.

Flow charts are an important tool for both design and debugging as they can be used to trace through the expected paths that an algorithm should take with the help of a **dry run** table. This checks that variables are set and changed in the correct way at the right moments.

Key term

Dry run – this is a method used by code developers to carefully follow how a program will run, by keeping track of the contents of variables at every point and checking that the decisions are based upon the correct values so that the program follows the expected code branches.

System diagrams

System diagrams are less formal than flow charts. They use boxes to represent parts of the system and arrows to show the data flows. Flow charts are well suited to program code whereas system diagrams are good for showing how larger systems operate.

Solving problems

Flow charts and system diagrams are both used to explain how systems work and to help solve problems.

A diagram helps to visualise a system and to communicate its inner workings to the designers, developers and other stakeholders of a project. This helps them to understand the system and to discuss any part that is in need of development or repair.

Identified problems within a system can be resolved using these methods, as the circumstances leading up to the issue can be traced from the beginning through the diagram. This path through the system can be used to see where the issue arose, so helping to identify a solution to the problem.

PAUSE POINT What are the real benefits, if any, of using diagrams to help explain how a system works?

Hint Close the book and discuss with a peer how a diagram can communicate in ways that are difficult for words alone.

Extend A program flow chart has standard symbols to represent different types of code. What are the benefits of using standard shapes in a diagram? Who does this benefit the most?

Assessment practice 2.6

A01 A02 A05

Describe and explain the logic circuits needed to add together two 8-bit bytes in a processor. Include truth tables to define the inputs and outputs for each type of circuit used here.

Produce a guide to the symbols and describe the code represented by terminators, arrows, input/output, processes and decisions in a program flowchart.

Produce an example flowchart to illustrate a programming algorithm of your choice with a written description of the algorithm.

Plan
- What am I learning? Why is this important?
- How will I approach the task?
- Do I have any existing knowledge of the task?

Do
- Am I confident that I know what I am doing and that I know what it is I should be achieving?
- What strategies am I employing? Are these right for the task? Are they working? If not, what do I need to do to change this?
- I understand my thought process and why I have decided to approach the task in a particular way. I can explain this reasoning when asked.

Review
- I can explain how I would approach the hard elements differently next time (i.e. what I would do differently).
- I can explain where I learn best.
- I can explain what I have learned and why it is important.

THINK FUTURE

Tamas Diossy

IT Support Manager for a major bank

I've been working here for almost nine years. I got an internal promotion to my present position two years ago. My team consists of four technicians who support the IT system of this site of 400 users. The IT support we supply covers hardware, software, networking, purchasing recommendations and approvals. I like the scope and variety of the job, although I do miss programming as this skill is not required in my present role.

Any purchasing of new IT equipment or software needs to be approved by my department. We have a good understanding of the current marketplace and our current IT systems and can usually make one or two recommendations to any department needing new equipment or an upgrade to their system. We need to approve their decision as part of the sign-off process for the purchase. This process works well, we are pleased with the how the current IT system performs and we do our best to keep it that way by ensuring that new equipment is compatible with the current system, has a good performance rating and is good value for money.

Focusing your skills

Computer hardware within a computer system

Many job roles require an understanding of the current technologies and choices that can be made when purchasing new IT equipment as replacements or upgrades. Here are some suggestions for how you can improve your employability by improving your expertise and knowledge in this area.

- Understand the factors affecting the choice, use and performance of internal computer components. Be particularly aware of the need to get good value for money by choosing equipment that meets the needs of users without adding unnecessary extra functionality or performance, which adds to the cost.
- Be clear about how meeting needs includes providing an acceptable user experience, ease of use, high performance, availability, accessibility, compatibility, being within budget, availability within an acceptable timescale after testing/migration to a new system, productivity and security.
- Build up knowledge of the current options for data storage and recovery systems including RAID and NAS as well as announcements of new technologies and products.

The concepts of microarchitecture

Processors are one of the main components in a computer system. Knowing the concepts of microarchitecture will help you to understand and apply CPU specifications and features to help you to make an informed choice when purchasing a new processor and motherboard.

- Be clear about the factors affecting execution speeds, methods of achieving faster execution and implications of faster speeds. You also need to know the amount of processing power needed for different software applications.
- Find out the role of the cache inside a CPU. Research current processors to get to know how much cache is normal, small or impressive.
- Find out the role of cores inside a CPU and the benefits of more than one core. Research current processors to get to know how many cores are normal, small or impressive.
- Regularly use an independent website such as Tom's Hardware to see the results of testing hardware to build up your knowledge of how well different items of hardware perform and how much extra performance can be expected from the various performance features.

Getting ready for assessment

This section has been written to help you to do your best when you take the assessment test. Read through it carefully and ask your tutor if there is anything you are still not sure about.

About the test

The assessment test will last 1 hour and 45 minutes and there are a maximum of 80 marks available. There are 3 types of question in the test:

- annotated diagram questions
- short answer questions worth 1–2 marks
- a longer answer question worth up to 12 marks.

Remember that all the questions are compulsory and you should attempt to answer each one.

Each question will state how many marks are on offer for its completion. You should attempt all the questions.

Sitting the test

Arrive in good time so you are not in a panic.

The test will be carried out under supervised conditions.

- You may use a calculator.
- Internet access is not permitted.
- You must not bring anything into the supervised environment or take anything out without your tutor's knowledge and approval.

You will not be allowed to take your mobile phone into the exam.

Make sure that you arrive in good time for each test session and check that you have everything you need for the test ahead of time.

Listen to, and read carefully, any instructions you are given. Lots of marks are lost through not reading instructions properly and misunderstanding what you are being asked to do.

Organise your time, based on the marks available for each question. Set yourself a timetable for working through the test and then stick to it – do not spend ages on a short 1–5 mark question and then find you only have a few minutes for a longer 7–8 mark question.

Remember that you cannot lose marks for a wrong answer, but you cannot gain any marks for a blank space!

Ensure that you check all sides of the assessment task before starting work.

Ensure that you leave yourself enough time to check through your work. Proofread and correct any mistakes before handing in your work

Command words typically used in assessment

Most questions contain command words. Understanding what these words mean will help you understand what the question is asking you to do.

Command word	Definition – what it is asking you to do
Analyse	Identify several relevant facts of a topic, demonstrate how they are linked and then explain the importance of each, often in relation to the other facts.
Calculate	Apply some form of mathematical or computation process.
Complete	Complete a diagram or process. This can be applied to problems/solutions of varying complexity.
Demonstrate	Illustrate and explain how an identified computer system or process functions. This may require a piece of extended writing, drawing a diagram or a combination of the two.
Describe	Give an account of something or highlight a number of key features of a given topic. This may also be in relation to the stages of a process.
Develop	Provide a solution to a problem, typically using an existing system or structure that must be improved or refined.
Discuss	Investigate a problem or scenario, showing reasoning or argument.
Draw	Represent understanding through the use of a diagram or flow chart.
Evaluate	Review and synthesise information to provide a supported judgement about the topic or problem. Typically, a conclusion will be required.
Explain	Make a series of linked points and/or justify or expand on an identified point.
Identify	Assess factual information, typically when making use of given stimuli. Requires a single word or short sentence answer.
Produce	Provide a solution that applies established constructs to a given computing problem.
Write	Produce a solution, or a mechanism used as part of a solution, to a given computing problem.

Writing long answers

If you are writing a longer answer, try and plan before you start writing. Have a clear idea of the point your answer is making, and make sure that this comes across in everything you write, so that it is all focused on answering the question.

- Always make a plan for your answer before you start writing. Sketch this out so that you can refer to it throughout – remember to include an introduction and, if the command word is 'evaluate', a conclusion and think about the key points you want to mention in your answer. On this plan, set yourself some timeframes so that you make sure that you have time to cover everything you want to and, importantly, have time to write the conclusion!
- Try to keep your answer as focused on your key points as possible. If you find your answer drifting away from that main point, refer back to your plan.
- Make sure that you understand everything being asked of you in the activity instructions. It might help you to underline or highlight the key terms in the instructions so that you can be sure that your answer is clear and focused on exactly what you have been asked to do.

Sample answers

For some of the questions, you will be given some background information on which the questions are based.

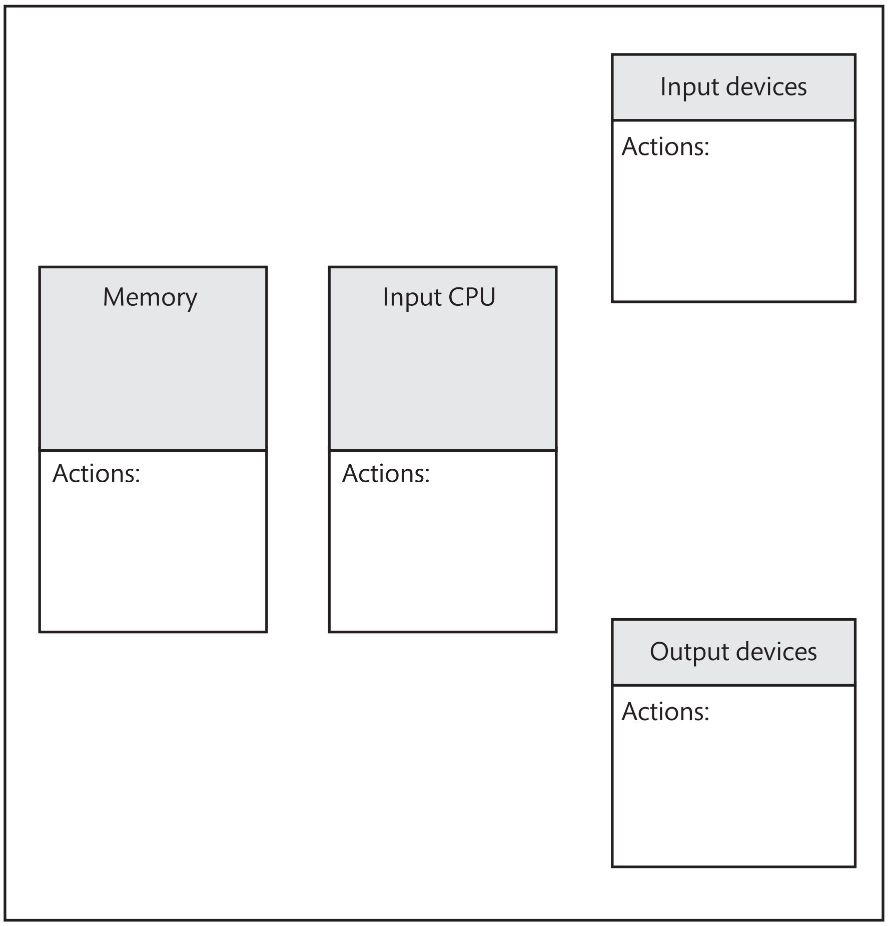
Look at the sample questions which follow and our tips on how to answer these well.

Answering annotated diagram questions

You need to be careful when annotating a diagram. There will be a small amount of space so carefully choose words which are suitable for the **context** of the question. Always read the question carefully and write the **most appropriate** answer.

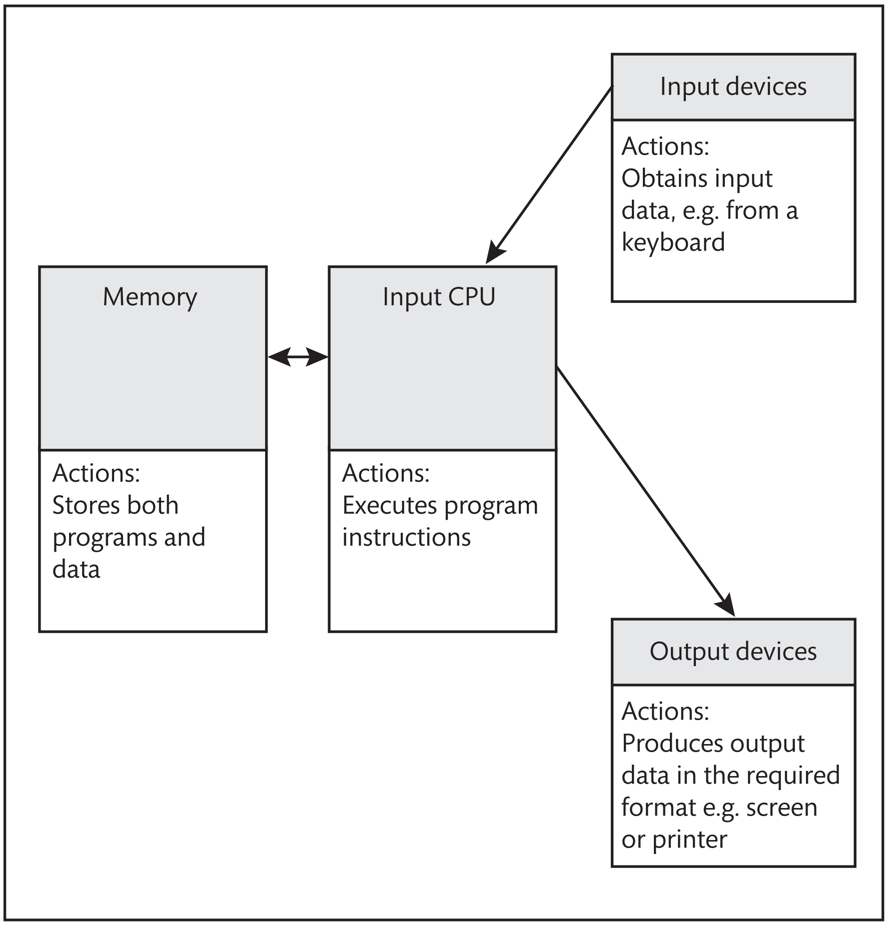
Worked example

Many computers use the Von Neumann architecture for their CPUs.
The main components of the Von Neumann architecture are show below.
Compete the diagram by describing the actions performed by each component and the direction(s) in which data flows:



To get full marks the data flow arrows must point in the correct direction.

Answer:



The examiner will accept any other relevant phrasing/wording alternatives to the model answer given here.

Answering short answer questions

- [ ] Read the question carefully.
- [ ] Highlight or underline key words.
- [ ] Note the number of marks available.

Make sure that you make the same number of statements as there are marks available. For example, a 2 mark question needs 2 statements.

Look carefully at how the question is set out to see how many points need to be included in your answer.

Worked example

State two types of operating system than can be used on a computer **[2]**

Answer: Real-time, multi-tasking.

This answer is enough to achieve the 2 marks. Other possible answers include single user and multi-user.

Answering extended answer questions

Worked example

Analyse how RAID discs can be used by server computers **[6]**

For a question using the word 'analyse', you must identify several relevant facts of the topic, demonstrate how they are linked and then explain the importance of each, in relation to the other facts.

Answer: Redundant Arrays of Independent Discs (RAID) is a disc storage technology that has two main benefits, improved performance and data security. Because the data is spread across at least two different discs this provides better performance because there are multiple drives reading and writing data rather than just one large drive. This is important for server computers because they usually support large number of users, all of which may be requesting data off the discs at the same time. Using RAID drives with multiple discs increases the number of simultaneous reads/write and therefore provides better performance for users.

Make sure you state not just how the RAID system works but also why it is important in a server computer environment

RAID also provides fault tolerance by spreading data across multiple disc drives using a technique called striping and by storing parity data across the drives. The means that if one drive in the RAID set fails the data that was on the drive can be calculated using the parity data. The most commonly used versions of RAID, RAID 5 and 6 allow a server computer to continue uninterrupted if a single drive fails. Once the failed drive is replaced the data on it is reconstructed. Fault tolerance is important for many servers because they often support business critical applications which would cause considerable business disruption if they were not available.There are no moving parts as the SSD uses flash storage which reduces the overall weight of the tablet making it more portable. There is no mechanical operation, which makes it silent and lets the SSD run without much heat generation.