FOR THE J277 SPECIFICATION

OCR GCSE (9–1)

COMPUTER SCIENCE

SECOND EDITION

GEORGE ROUSE, LORNE PEARCEY, GAVIN CRADDOCK

An OCR endorsed textbook

DYNAMIC LEARNING

OCR Oxford Cambridge and RSA

HODDER EDUCATION LEARN MORE

CONTENTS

How to use this book v

SECTION 1 COMPUTER SYSTEMS 1

1.1 System architecture 2
- **1.1.1** Architecture of the CPU 2
- **1.1.2** CPU performance 6
- **1.1.3** Embedded systems 7

1.2 Memory and storage 12
- **1.2.1** Primary storage – memory 12
- **1.2.2** Secondary storage 15
- **1.2.3** Units 19
- **1.2.4** Data storage 22
- **1.2.5** Compression 39

1.3 Computer networks, connections and protocols 52
- **1.3.1** Networks and topologies 52
- **1.3.2** Wired and wireless networks, protocols and layers 63

1.4 Network security 73
- **1.4.1** Threats to computer systems and networks 73
- **1.4.2** Identifying and preventing vulnerabilities 77

1.5 Systems software 83
- **1.5.1** Operating systems 83
- **1.5.2** Utility software 86

1.6 Ethical, legal, cultural and environmental impacts of digital technology 91
- **1.6.1** Ethical, legal, cultural and environmental impact 91

SECTION 2 COMPUTATIONAL THINKING, ALGORITHMS AND PROGRAMMING 105

2.1 Algorithms 106
- **2.1.1** Computational thinking 106
- **2.1.2** Designing, creating and refining algorithms 107
- **2.1.3** Sorting and searching algorithms 112

2.2 Programming fundamentals 125
- **2.2.1** Programming fundamentals 125
- **2.2.2** Data types 134
- **2.2.3** Additional programming techniques 136

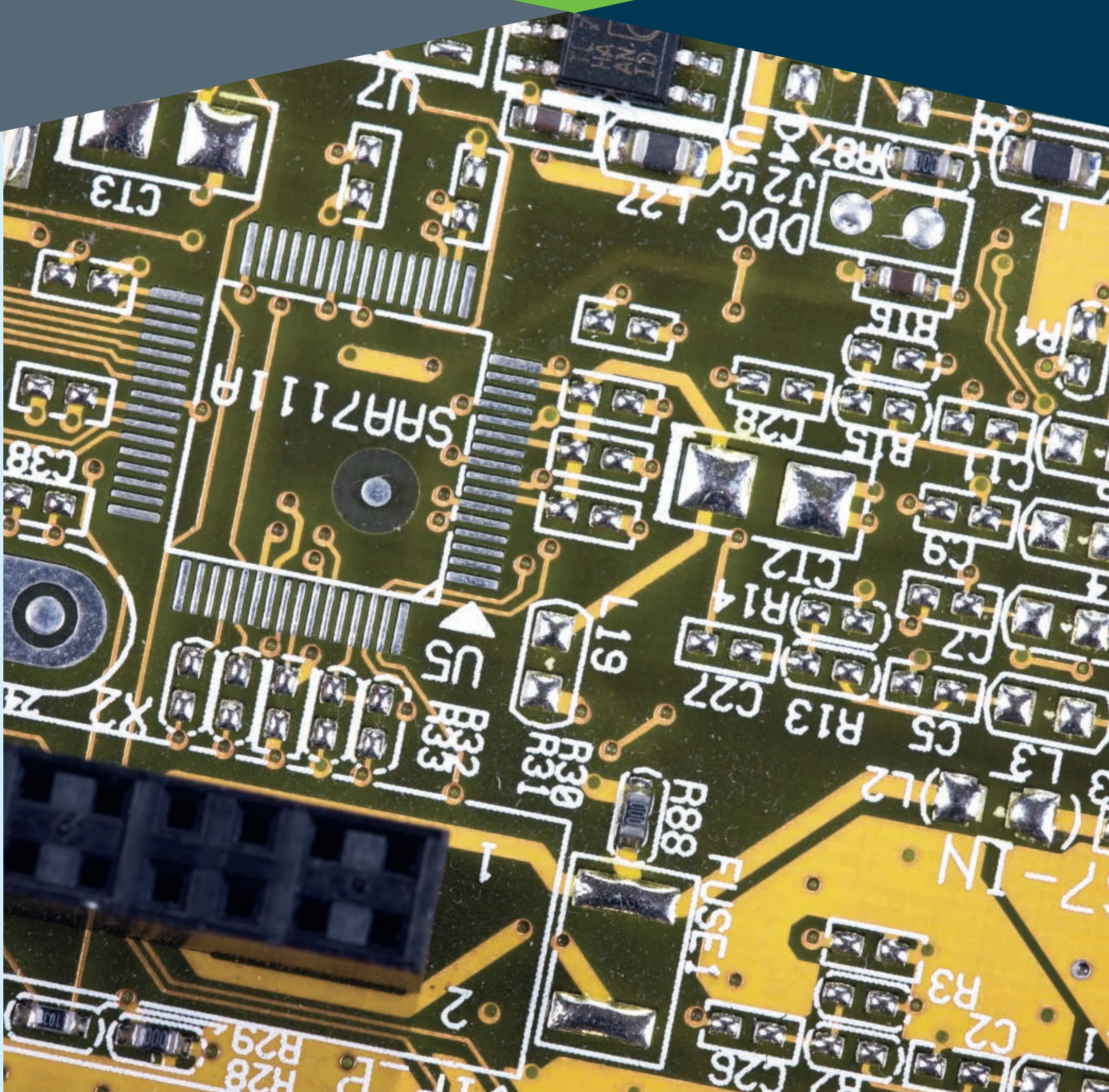
2.3 Producing robust programs 152
- **2.3.1** Defensive design 152
- **2.3.2** Testing 157

2.4 Boolean logic 164
- **2.4.1** Boolean logic 164

2.5 Programming languages and integrated development environments 174
2.5.1 Languages 174
2.5.2 The integrated development environment (IDE) 177
Appendix 182
Glossary 190
Knowledge check answers 195
Index 205
Acknowledgements 210

SECTION 1

COMPUTER SYSTEMS



1.1 SYSTEM ARCHITECTURE

CHAPTER INTRODUCTION

In this chapter you will learn about:

1.1.1 Architecture of the CPU

- The purpose of the CPU
- Common CPU components and their function
- Von Neumann architecture

1.1.2 CPU performance

- How common characteristics of CPUs affect their performance

1.1.3 Embedded systems

- The purpose and characteristics of embedded systems
- Examples of embedded systems

Figure 1.1.2 A CPU

A computer system consists of hardware and software working together to process data. Hardware is the name for the physical components that make up the computer system. Software is the name for the programs that provide instructions for the computer, telling it what to do, and is covered in Chapter 1.5.

A computer system receives information as an input, processes and stores that information, and then outputs the results of that processing.

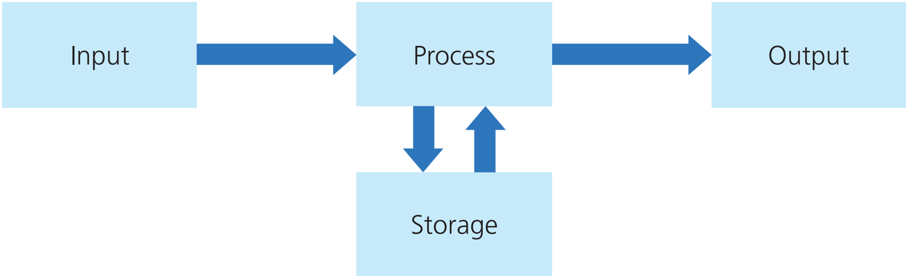


Figure 1.1.1 Input – process – output

Processing and storage is the job of the Central Processing Unit.

1.1.1 Architecture of the CPU

The purpose of the CPU

The **Central Processing Unit (CPU)** is made up of billions of transistors, which are like very small 'on–off' switches. The arrangement of transistors creates logic circuits that process data, carry out instructions and control the components of the computer.

The Fetch–Execute cycle

The processor continually:

- fetches instructions from memory
- decodes these instructions
- and then executes them.

This is called the **Fetch–Execute cycle**. There is more detail on this a little later in the chapter.

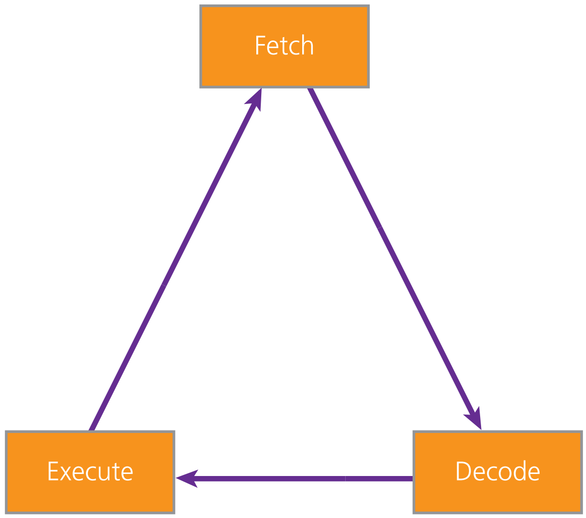


Figure 1.1.3 Fetch–Execute cycle

Common CPU components and their function

The CPU is made up from a number of components:

Arithmetic Logic Unit

The **Arithmetic Logic Unit (ALU)** is responsible for the following:

- arithmetic operations such as add, subtract, multiply and divide
- logical operations such as AND, OR and NOT, and the result of 'less than', 'greater than', 'equal to' comparisons
- binary shift operations (which are used for multiplication or division – see section 1.2.4).

The ALU carries out the calculations and logical decisions required by the program instructions that the CPU is processing.

Control unit

The purpose of the **control unit (CU)** is to co-ordinate the activity of the CPU.

It does this by:

- fetching then decoding instructions from memory
- sending out signals to control how data moves around the parts of the CPU and memory, in order to execute these instructions.

Cache memory

The purpose of **cache memory** is to provide temporary storage that the CPU can access very quickly.

Its role in the CPU is to store instructions and data that are used repeatedly or are likely to be required for the next CPU operation.

Registers

Registers are memory locations within the CPU that hold data temporarily and can be accessed very quickly.

Their role in the CPU is to accept, store and transfer data and instructions for immediate use by the CPU.

These registers are used during the Fetch–Execute cycle.

Von Neumann architecture

CPU architecture describes how the different components in the CPU are laid out and communicate with each other. The **Von Neumann architecture** describes a computer in which the data and instructions are stored in the same area of memory and are indistinguishable from each other. This means that the CPU has to decide what is an instruction and what is simply data.

Beyond the spec

The CPU uses a very low-level language called machine code (see Chapter 2.5). Machine code instructions are represented in binary. The value 1001 may be machine code for the arithmetic operation addition (ADD), but it is also the binary equivalent of the decimal number 9. Depending upon what the computer expects to find, it interprets the value 1001 as **either** an instruction to ADD **or** as the value 9.

Von Neumann architecture is the fundamental design concept behind all modern computer systems.

There are four important registers in a CPU with a Von Neumann architecture:

Program counter (PC): The program counter keeps track of the memory location (an address) for the next instruction. In many cases, the program counter is simply incremented to the next memory location at the Fetch stage of the Fetch–Execute cycle, to allow the program to be executed line by line. (Program instructions can, however, modify the value in the program counter to alter the flow of the program so that it continues from a new location.)

Memory data register (MDR): This register is used to store any data fetched from memory or any data that is to be transferred to and stored in memory.

Memory address register (MAR): This register stores the location in memory (known as an address) to be used by the MDR – that is, where the MDR needs to fetch data from or send data to.

Accumulator (ACC): This register either stores the results of any calculations made by the ALU, or stores the value of inputs and outputs to and from the CPU.

Figure 1.1.4 is a simplified diagram showing the layout of these components and how the CPU communicates with memory and input/output devices. Note that you do **not** need to know about buses for your exam.

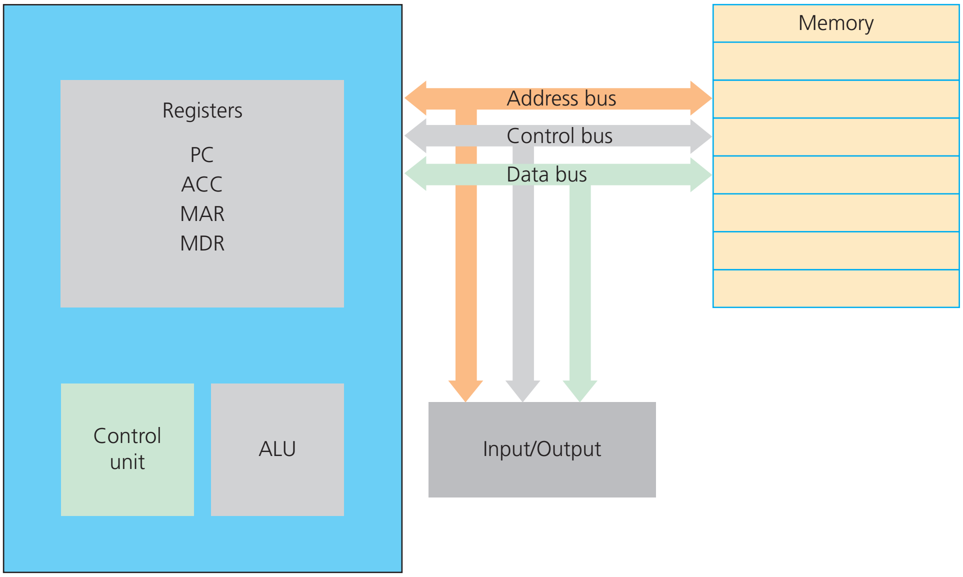


Figure 1.1.4 A CPU with Von Neumann architecture; the arrows represent the flow of data between components

Beyond the spec

To enable data and control signals to move around the CPU and memory, there are a number of buses. Buses are communication channels through which data can be moved.

There are three main buses inside the computer in relation to the CPU:

Data bus: This carries data between the CPU and memory.

Control bus: This carries control signals around the CPU and memory.

Address bus: This carries memory addresses for locations to be read from or written to.

Now that you know the names of the registers in the CPU, we can revisit the Fetch–Execute cycle so you can understand how it works in more detail.

Fetch

1. Each instruction in a computer program is stored in a particular location (or address) in memory. The address of the next instruction is copied from the **program counter** and placed in the **MAR**.
2. The MAR now contains a memory address. The **control unit** FETCHES the data that is stored at that address and copies it to the **MDR**.
3. The **program counter** is incremented to point to the next instruction to be processed in the program, ready for the next Fetch–Execute cycle.

Decode

4. The **MDR** now contains either data fetched from memory or an instruction. The **control unit** DECODES the instruction to see what to do.

Execute

5. The decoded instruction is EXECUTED. This might mean performing a calculation using the **ALU**, locating some data in memory, changing the **program counter** value – or something else.

Once the Execute part of the cycle is complete, the next Fetch–Execute cycle begins.

Knowledge check

1 Describe the purpose of the CPU in a computer.
2 Describe the Fetch–Execute cycle.
3 What is held in the memory address register (MAR)?
4 What is the purpose of the memory data register (MDR)?
5 What is the purpose of the program counter (PC) in the CPU?
6 State two arithmetic and two logical operations carried out by the Arithmetic Logic Unit (ALU).

1.1.2 CPU performance

How common characteristics of CPUs affect their performance

There are a number of factors that have an impact on the performance of a CPU.

Clock speed

The CPU is constantly fetching and executing instructions and the speed at which it does this is determined by an electronic clock. This clock uses a vibrating crystal that maintains a constant speed. Clock speeds are measured in hertz (Hz), which is the unit of frequency and means 'number of times per second' – see section 1.2.4. Typical modern computers work at speeds of up to 4 GHz (or 4 billion instructions per second). Each 'tick' of the clock represents one step in the Fetch–Execute cycle. The faster the **clock speed**, the more instructions that can be executed every second.

Size of cache memory

Cache memory is located between the main memory and the CPU. It is used to hold data that needs to be accessed very quickly. Accessing cache memory is much faster than accessing main memory (also known as random access memory (RAM) – see section 1.2.1).

The CPU control unit will first look in the cache for data or instructions, to see if they have already been copied from main memory. If they are not in the cache memory then the control unit will go to the main memory to locate them, and will then copy the data or instructions to cache and then to the CPU registers.

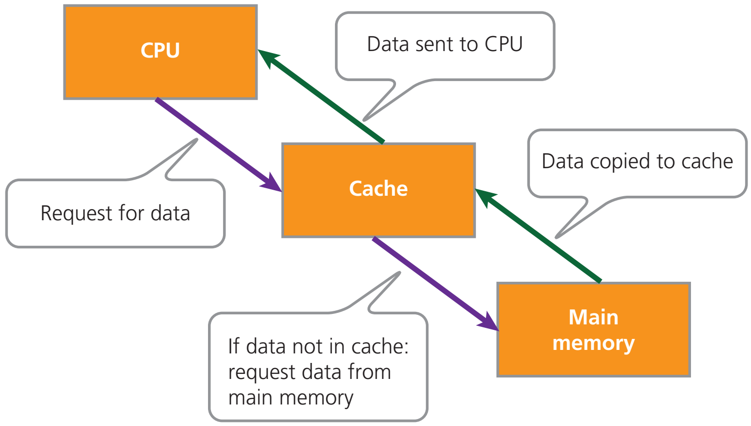


Figure 1.1.5 Cache memory is used to store data waiting to be processed

The more data that can be stored in cache memory rather than main memory, the faster and more efficient the process. Data that is likely to be required will be transferred to cache, ready to be used.

The larger the cache memory the more likely it is that the required data will already have been copied across from main memory. Cache memory is very expensive and while a mid-range laptop may have 8 GB of RAM, it is likely to have just a few KB of cache.

Number of processor cores

Another factor that can affect the performance of the CPU is the number of **processor cores**. Each core can fetch and execute instructions independently so a multiple core processor can handle several instructions at the same time. While these multiple cores can work on separate programs or parts of a program at the same time, this is only possible if the program has been written to take advantage of multiple cores. The task that the program is attempting must also be one that can be split up into subsections to take advantage of multiple cores.

Knowledge check

?

7 What is a meant by a quad core processor?
8 What is meant by 2.3 GHz when describing a CPU?
9 Describe how cache memory is used by the CPU.
10 Describe three characteristics of a CPU that affect its performance.

1.1.3 Embedded systems

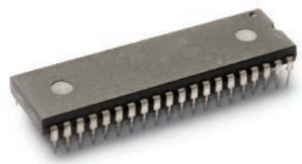


Figure 1.1.6 A microcontroller

An **embedded system** is a computer system that has a dedicated function as part of a larger device. The main components of a computer system are either manufactured onto a single chip (a microcontroller) or separate circuits for processing and memory are combined into a larger device.

Purpose and characteristics of embedded systems

When a computer device is required to perform a single or fixed range of tasks, it can be engineered to reduce its size and complexity in order to focus only on these tasks. Dedicated software will be programmed into the device to complete the necessary tasks and nothing else. The reduction of complexity of the hardware and the dedicated nature of the software will make the device more reliable and cost effective than using a general-purpose computer.

The embedded system will typically include some read-only memory (ROM) to store the dedicated program and some RAM (see section 1.2.1) to store user inputs and processor outputs. For example, in a washing machine, the ROM will store all of the data describing the separate washing cycles while the RAM will store the user's selected options (inputs) and the data used to display choices and progress of the washing cycle (outputs).

Embedded systems have the following characteristics:

- **low power** so they can operate effectively from a small power source such as in a digital camera
- **small in size** so they can fit into portable devices such as a personal fitness device

- **rugged** so that they can operate in harsh environments such as car engine management systems or in military applications
- **low cost**, making them suitable for use in mass-produced, low-cost devices such as microwave ovens
- **dedicated software** to complete a single task or limited range of tasks, such as in computer aided manufacture or control systems.

Examples of embedded systems

Embedded systems are found within common household devices such as washing machines, dishwashers, microwaves, set-top boxes, telephones, televisions, home security and control systems, and so on. Embedded systems are also widely used within larger and more complex systems such as car engine management, aeroplane avionics, computer-controlled manufacturing and military applications such as guidance systems.

Some embedded systems are connected to the internet via Wi-Fi to exchange data with third parties, for example water meters, energy smart meters and home security or heating monitoring systems.

Embedded systems are particularly useful for those with physical disabilities as they can make items more accessible. This can include voice control for gadgets in the home and systems that adapt motorised vehicles so they can be operated using limited physical movements.

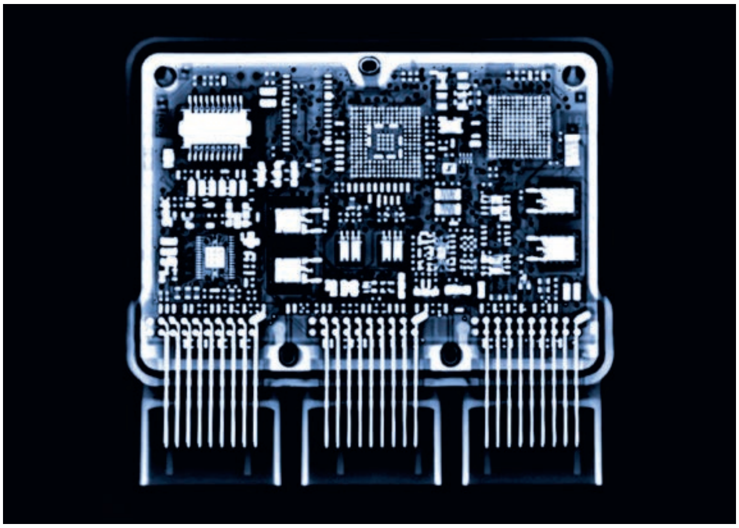


Figure 1.1.7 An x-ray image of an engine control unit in a motorcycle

Knowledge check

11 Explain why embedded systems have both RAM and ROM.
12 Identify one input and one output from the embedded system in a microwave oven.
13 Give two examples of systems that use embedded computer systems and explain why it is the most appropriate type of computer system to use in each case.

1.1 SYSTEM ARCHITECTURE

Important words

You will need to know and understand the following for the exam:

- Central Processing Unit (CPU)
- Fetch–Execute cycle
- Arithmetic Logic Unit (ALU)
- Control unit (CU)
- Cache memory
- Registers
- Von Neumann architecture
- Program counter (PC)
- Memory data register (MDR)
- Memory address register (MAR)
- Accumulator (ACC)
- Clock speed
- Processor cores
- Embedded system

1.1.1 Architecture of the CPU

Hardware is the term that describes the **physical components** of a computer.

The computer **inputs**, **processes** and **stores**, and **outputs** data.

The computer hardware works with the software to **process data**.

The hardware that processes the data is called the **CPU (Central Processing Unit)**.

Purpose of the CPU

The purpose of the CPU is to carry out a set of instructions that is contained in a computer program.

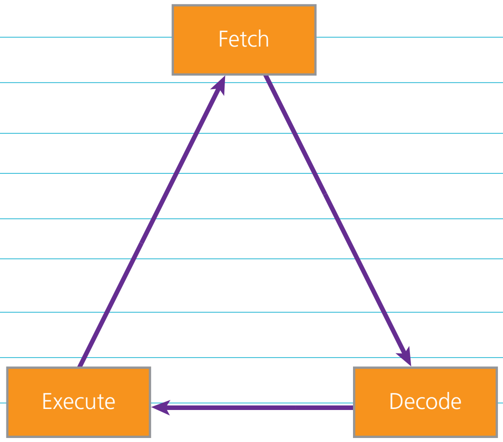
The CPU works at high speeds governed by the clock chip.

- The clock uses a vibrating crystal that maintains a constant speed.
- The clock chip operates at speeds of up to 4 GHz (or 4 billion instructions per second).

Fetch–Execute cycle

The CPU continually fetches, decodes and executes instructions.

- Fetch – an instruction in the form of data is retrieved from main memory.
- Decode – the CPU decodes the instruction.
- Execute – the CPU performs an action according to the instruction.



Fetch–Execute cycle

Common CPU components and their function

Arithmetic Logic Unit (ALU)

Carries out all arithmetic calculations and logical decisions, for example add, subtract, binary shifts to multiply and divide, and comparisons such as equal to or greater than.

Control unit (CU)

Decodes instructions and sends signals to control how data moves around the parts of the CPU and memory to execute these instructions.

Cache

Cache memory sits between the processor and main memory (RAM).

- The CPU looks in the cache for required data.
- If it is not there, it requests it from RAM.
- The data is moved into cache before being accessed by the CPU.

Registers

Memory locations within the CPU that hold data. A register may hold an instruction, a storage address or any kind of data. The data in registers can be accessed very quickly – even more quickly than cache memory.

Von Neumann architecture

CPU architecture refers to the internal logical structure and organisation of the computer hardware.

In Von Neumann architecture, data and instructions are stored in the same memory.

- **Program counter (PC)**: Stores the address of the next instruction to be processed.
- **Memory data register (MDR)**: Stores data fetched from memory or to be sent to memory.
- **Memory address register (MAR)**: Stores the address of the location in memory for data to be fetched from or sent to.
- **Accumulator (ACC)**: Stores the results of any calculations carried out by the ALU.

1.1.2 CPU performance

How common characteristics of CPUs affect their performance

Clock speed: The CPU works at high speeds governed by the clock chip. The **faster the clock** the **more instructions that can be completed per second**.

- The clock uses a vibrating crystal that maintains a constant speed.
- The clock chip operates at speeds of up to 4 GHz or 4 billion instructions per second.

Cache memory: Cache memory can be accessed very quickly by the CPU, so having more cache will provide the CPU with fast access to more data.

Number of cores: If a CPU has multiple cores it may be able to process more instructions simultaneously.

1.1.3 Embedded systems

Purpose and characteristics

- An **embedded system** is a computer system that has been designed for a **dedicated function as part of a bigger system**.
- Embedded systems are often **manufactured as a single chip**.
- The dedicated hardware and software make embedded systems **more robust and reliable** than general-purpose computers.

Characteristics of embedded systems include:

- Designed and engineered **to perform a limited set of tasks to reduce size and improve performance**.
- Programs are often **uploaded at the manufacturing stage**, directly to the device.
- There are often **very limited options to modify** these programs.
- **Low power consumption** to operate from a small power source.
- **Small in size** to fit in portable devices.
- **Rugged** so that they can operate in hostile environments.
- **Low cost** making them suitable for mass-produced products.

Examples of embedded systems

Embedded systems are found in most consumer products such as:

- washing machines
- microwave ovens
- home security systems
- home heating controls
- car engine management systems
- set-top boxes
- telephones
- televisions
- home security and control systems.

Extra resources

A free set of practice questions accompanies this section and is available online at: www.hoddereducation.co.uk/OCRGCSEComputerScience

These practice questions have not been produced either by OCR or by the OCR Principal Examiner. They are also not endorsed by OCR and have not been subject to any OCR quality assurance processes.

1.2 MEMORY AND STORAGE

CHAPTER INTRODUCTION

In this chapter you will learn about:

1.2.1 Primary storage–memory

- The need for primary storage
- The difference between RAM and ROM
- The purpose of ROM in a computer system
- The purpose of RAM in a computer system
- Virtual memory

1.2.2 Secondary storage

- The need for secondary storage
- Common types of storage
- The advantages and disadvantages of different storage devices

1.2.3 Units

- The units of data storage
- How data needs to be converted into a binary format to be processed by a computer
- Data capacity and calculation of data capacity requirements

1.2.4 Data storage

- Converting between denary, binary and hexadecimal
- How to add two 8-bit binary integers together and explain any overflow errors
- Binary shifts
- The use of binary codes to represent characters
- Character sets, the relationship between the number of bits per character and the number of characters in a character set
- How an image is represented as pixels and in binary
- Image metadata
- The effect of colour depth and resolution on image quality and size
- How sound can be sampled and stored in digital form
- The effect of sample rate, duration and bit depth sound quality and file size

1.2.5 Compression

- The need for compression
- Types of compression

For a computer system to be useful, it needs storage: storage for data and programs that are currently in use and storage for data and programs that can be accessed when required.

1.2.1 Primary storage – memory

The need for primary storage

A computer system needs primary storage for any data that it needs to access quickly. This includes the start-up instructions, the operating system, programs that are running and any associated data. There are two main types of primary storage: RAM and ROM.

Random access memory (RAM)

RAM is part of the main memory in a computer system, and a typical laptop will have around 8 GB of RAM available. RAM is required to hold the operating system, applications that are running and any associated data while the computer is on and in use.

When a program is loaded, it is copied from secondary storage, such as a hard disk drive (HDD), into RAM. (For an explanation of secondary storage see section 1.2.2.) Any data associated with the program will also be stored in RAM so that the CPU can access both the data and instructions.

Data is transferred into RAM from secondary storage for use by the CPU. With more RAM available more data and applications can be stored in it. Because RAM has fast data access times this leads to better performance of the system. In practice a system with more RAM can have more programs open at the same time without any noticeable decrease in performance

As noted in the previous chapter, once data and instructions are in RAM they are then transferred to cache memory, in order to further improve data access speeds for the CPU. Figure 1.2.1 illustrates how access to data and programs is improved by using RAM and cache together.

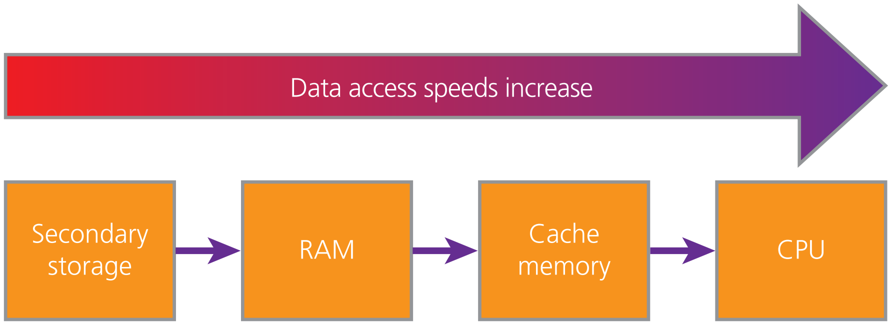


Figure 1.2.1 Data transfer speeds for secondary storage, main memory, cache and CPU

RAM is **volatile**, meaning it needs electrical power to operate. Any data stored in RAM is lost when the power is turned off.

RAM can be read from or written to by the computer. This means RAM is **read and write**.

Read-only memory (ROM)

Typically, in a computer system, **ROM** stores the instructions and data needed to get the system up and running and ready to load the operating system from secondary storage. This special program stored on ROM is called the Bootstrap Loader and we say the process 'boots' the computer – this means that it starts it from scratch.

Computers would not be so useful if they had to be switched on all the time. ROM is **non-volatile** memory, which means it does not require power to maintain its contents.

ROM is **read-only**. This means that the data stored in ROM is fixed and cannot be overwritten once it is created. This data is written to ROM either at the manufacturing stage or through a special process later on.

The difference between RAM and ROM

Table 1.2.1 A comparison of RAM and ROM

RAM	ROM
Volatile and needs power to maintain the content	Non-volatile and does not require power to maintain the content
Read and write – data can be read from and written to RAM by the computer	Read-only – the computer cannot overwrite its content
Holds the operating system and any programs and data currently in use by the computer	Holds the data and instructions required to start up (boot) the computer

Knowledge check

?

1 State two differences between RAM and ROM.
2 What is held in RAM while the computer is working?
3 What is held in ROM on the computer?

Virtual memory

It is not always possible to store all the data and instructions we need in RAM. If a computer is running complex programs processing large amounts of data – for example large images or video files – or lots of programs are open simultaneously, there may be insufficient RAM to hold them all. In this case the computer can allocate a section of secondary storage to temporarily act like RAM. It does this by selecting data in RAM that is not currently required by the CPU and moving it temporarily into secondary storage. Once that data is required by the CPU, it is moved back from secondary storage into RAM.

The area of secondary storage used to temporarily store data from RAM is called **virtual memory**.

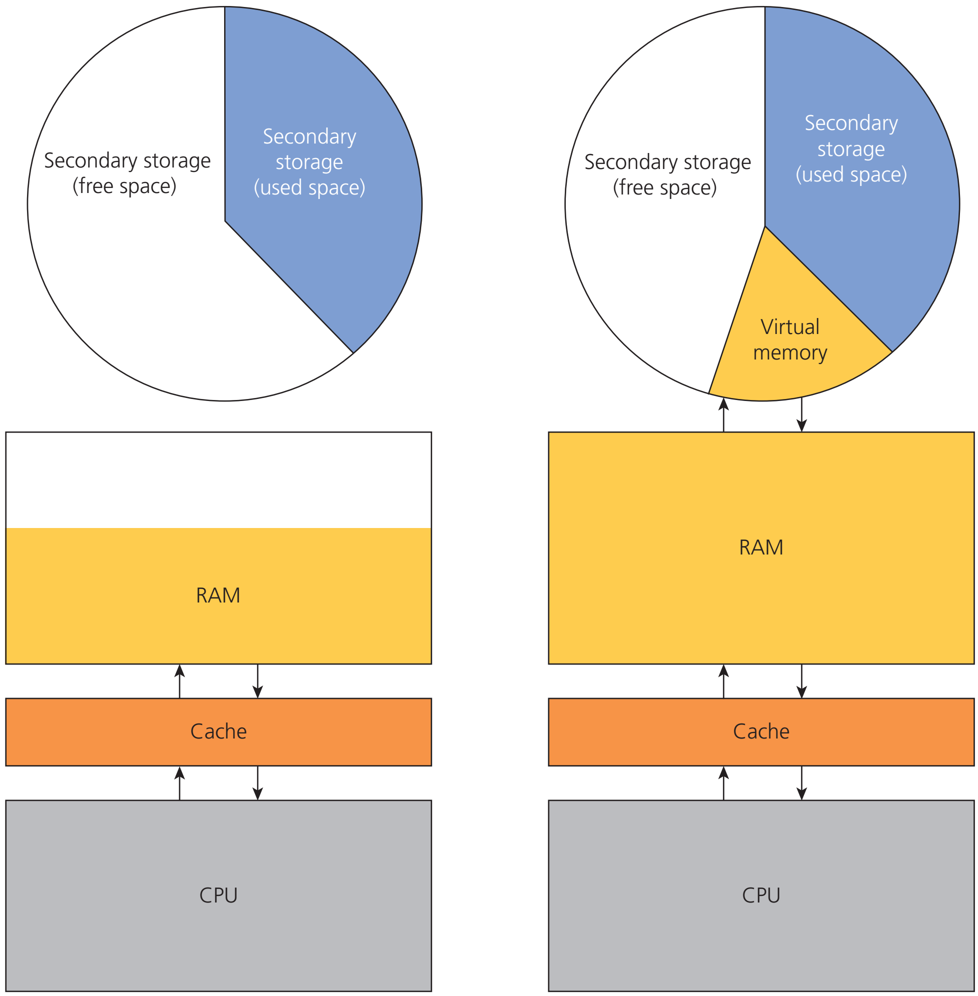


Figure 1.2.2 Virtual memory

In the left-hand diagram there is only one program running which only takes up some of the available RAM, and so the computer does not need to use virtual memory. In the right-hand diagram, more data and programs are open and the computer has run out of RAM. It allocates a section of secondary storage to act like RAM. This is represented by the orange section of secondary storage and is called virtual memory.

Using virtual memory will affect the performance of the computer system because there is a delay when transferring data from secondary storage back into RAM.

A computer with more RAM will need less virtual memory, reducing the number of data transfers between RAM and secondary storage and therefore delivering improved performance.

Knowledge check

4 What is virtual memory and why is it needed?
5 How might installing more RAM affect the use of virtual memory and how might this affect the performance of the computer?

1.2.2 Secondary storage

The need for secondary storage

Computer systems would be of little value if we lost all of our data and programs every time we switched them off. We need to store the operating system, data, images, programs, documents and various other files on our computers so that they are available the next time we switch on the computer. This kind of data requires a lot of space so we need a low-cost, high-capacity, read-and-write, non-volatile storage medium. This is known as **secondary storage**. Secondary storage needs to keep data safe and must be robust and reliable.

A number of different secondary storage media can be used for a computer system and the choice is based on several factors:

- Capacity: How much data does it need to hold?
- Speed: How quickly can the data be accessed?
  - Typically access times to secondary storage are very slow compared to primary storage (main memory).
- Portability: Does the data stored on the device need to be moved or transported?
  - If so the size, shape and weight of the medium is important.
- Durability: How robust is the medium?
  - Will it be damaged when moved around?
  - Will it be used in a hostile environment subject to shocks or extreme conditions?
- Reliability: Does it need to be used over and over again without failing?
- Cost: What is the cost per GB of data stored?
  - This is also an important factor where the media are being used to distribute data and will not be reused.
  - CDs for example only cost pennies but the cost per GB of storing data is higher than that for a magnetic hard disk drive.

Knowledge check

6 Why do we need secondary storage?
7 What is stored on secondary storage in a computer system?

Magnetic storage

Magnetic storage uses the principle of magnetism to store data.

Hard disk drives (HDDs) are magnetic and are the most common type of secondary storage.

Beyond the spec

HDDs are made of a stack of rigid disks (called platters) on a single spindle that rotates. Each platter is coated in a magnetic material, which is effectively made up of billions of separate tiny magnets that can either point 'north' or 'south'. Each bit of data (see section 1.2.3) is represented by these tiny magnets – north for '1' and south for '0'. A set of 'heads' moves across the platters, reading or writing data by sensing or changing the north/south alignment of the magnets.

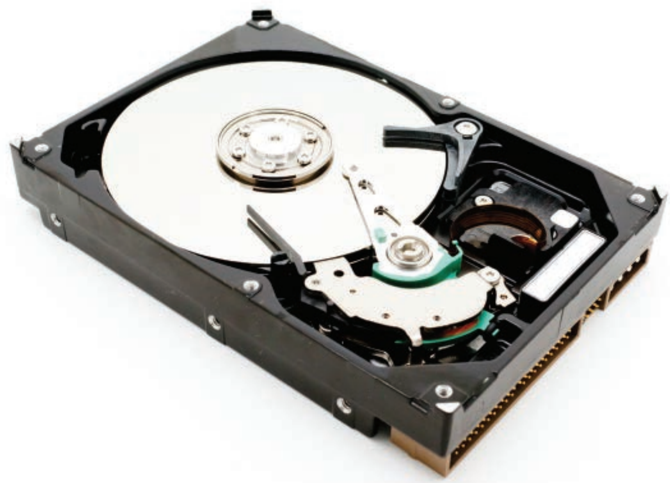


Figure 1.2.3 A hard disk drive showing the platters and heads

The magnetic hard disk is a reliable and cost-effective storage solution, providing high capacity at low cost. This makes them an ideal choice for the typical laptop or desktop computer. Internal and external hard disk drive capacities are currently measured in terabytes (a million megabytes – see section 1.2.3). Large hard disk drives are, however, less portable than solid-state drives or optical disks and are subject to damage if dropped or brought near to strong electric or magnetic fields.

Several drives can be combined in larger commercial systems to provide a significant amount of storage at a reasonable cost. At the other end of the scale, there are small portable hard disk drives that can easily be moved between computers.

Figure 1.2.4 An external hard drive connected to a laptop

Another form of magnetic storage is **magnetic tape**, which is still used as a cheap way to archive large amounts of data.

Tech term

Magnetic tape An old storage medium that lasts a long time and is very cheap but has very slow read/write times and thus is only really used for archiving.

Tech terms

Flash memory A method of storing data that is based on electronics.

Latency A delay before data can be transferred.

Fragmented Data stored in different physical locations across the disk.

Solid-state storage

Solid-state storage uses a technology called **flash memory** that has very fast data access times compared to HDDs, largely because there are no moving parts. Flash memory is, however, relatively expensive compared to hard disk drives and typically has lower capacity.

Solid-state storage is widely used for portable devices such as cameras (e.g. memory cards), and comes in a range of physical sizes and capacities to suit a wide range of applications. These portable devices, like USB pen drives, can be used to easily backup or transfer data between devices.

Solid-state flash memory is also used as the basis for **solid-state drives** (SSDs). SSDs have begun to replace magnetic hard disk drives (HDDs) because they have many advantages over them:

- In order to read data, magnetic HDDs have to line up the correct portion of the disk with the position of the read/write head. This means the magnetic disk has to rotate to the correct position and the head has to move across the disk. This in turn means there is a delay before data can be read or written. This delay is called **latency**. SSDs have lower latency times because there are no moving parts and access to the data does not require a platter to rotate or the read/write head to move. This improves access to the data and the performance of the device.
- The lack of moving parts means that SSDs have much lower power requirements and do not generate any heat or noise.
- HDDs can suffer from data being **fragmented** over the surface of the platters, producing very slow access speeds. This is not the case with SSDs.
- SSDs are significantly lighter, smaller and thinner than HDDs, making them particularly suitable for small, thin portable devices such as tablet computers or other portable devices.
- Since there are no moving parts, SSDs are not susceptible to problems caused by sudden movements, making them ideal in hostile environments or in portable devices.

Given the expense of SSDs, they are often combined with a magnetic disk drive to form a hybrid system. Frequently accessed data, such as the operating system, is stored on the SSD, while large, less frequently required data is stored on the magnetic disk. This provides the speed advantage of the SSD with the capacity advantage of the HDD at a reasonable cost compared to high-capacity SSDs.

Figure 1.2.5 Various solid-state devices

Beyond the spec

The surface of each disk is covered in billions of small indentations, known as **pits** and **lands**. When laser light is shone onto the surface of the disk it reflects off the pits and lands, with each pit or land representing a series of zeros, and the transition between pits and lands representing 1.

Tech terms

Pit A tiny indentation on a CD or DVD.

Land Areas on a CD or DVD where there are no pits.

Optical storage

Data can also be stored by using the properties of light. Typical **optical storage** media include CDs, DVDs and Blu-Ray disks. These are optical devices because they are written to and read from using laser light.

Some optical storage media are read-only but others can be written to, by creating pits on the surface of the disk using laser light.

For example, there are three main types of CD/DVD:

- **CD/DVD-ROM**: Read-only device with the data stored at the manufacturing stage.
- **CD/DVD-R**: Recordable media that can be written to once using a suitable drive.
- **CD/DVD-RW**: Rewriteable media that can be written to several times.

Typically, a CD will hold around 700 MB of data and cost pennies. They are used to distribute data and programs or make semi-permanent copies of data.

The DVD is very similar to the CD, but it has a larger capacity of 4.7–8.5 GB. This means that DVDs can store more data than CDs, such as standard resolution movies. A DVD has a faster access time than a CD and costs a little more, but is still only pennies.

Blu-Ray is similar but with significantly larger capacity (25–50 GB) and much faster access speeds. Blu-Ray disks can be used to store large amounts of data and the much higher access speed makes them particularly suitable for high-resolution movies and console games. They are slightly more expensive than DVDs but still reasonably inexpensive.

Table 1.2.2 Comparison of optical devices

Type	CD	DVD	Blu-Ray
Typical cost	18p	60p–80p	£1.80–£3.00
Capacity	700 MB	4.7 GB single layer 8.5 GB dual layer	25 GB single layer 50 GB dual layer

Choosing suitable secondary storage media

When choosing a suitable storage medium all of the following factors need to be considered.

- Capacity: How much data does it need to store?
- Speed: How quickly does the data need to be accessed?
- Portability: Does the device or medium need to be transported?
  - If so the size and weight are important.
- Durability: Will the device or medium be used in a hostile environment?
  - If so, the medium must be resistant to external shocks or extreme conditions.
- Reliability: Does it need to be used repeatedly without failing?
- Cost: What is the cost per unit of storage related to the value of the data?

Table 1.2.3 Capacity and cost of storage media

Media	Capacity	Typical cost	Cost per GB
Magnetic hard disk	Up to 15 TB	A 2 TB HDD costs around £60 and a 10 TB drive about £300	3p
SSD	250 MB up to 2 TB	£30 up to £300 for a 2 TB drive	15p
DVD	8.5 GB	80p	9p
Blu-Ray	50 GB	£3.00	6p
CD	700 MB	18p	23p

The cost per GB varies. All optical media are able to store reasonably small files for pennies per MB. For larger files, SSDs cost significantly more than HDDs for the same capacity. Larger storage requirements can be met more cost-effectively by a magnetic hard disk drive.

Speed

An SSD can transfer data at around 200–550 MB per second. A magnetic HDD transfers data at a much lower rate of 50–120 MB per second. An SSD will provide faster access to data. (For reference, data transfer times for RAM are around 12–20 GB/s.)

Table 1.2.4 Access speeds of storage media

Storage type	Data transfer rates (typical)
SSD	200–550 MB/s
Magnetic hard disk	50–120 MB/s
Blu-Ray disk	72 MB/s
DVD	1.32 MB/s
CD	0.146 MB/s

Table 1.2.5 Portability, durability and reliability

Media	Portability	Durability	Reliability
SSD	Small, with low power requirements Very portable	With no moving parts they are not subject to damage from sudden shocks.	The medium is reliable and will hold data safely for a very long time before failure.
Magnetic hard disk	With moving parts, higher power requirements than SSD. Available as external drives powered from a USB	Subject to damage from being dropped or from exposure to magnetic fields.	Ideal for medium term storage with a reliable life of 5–7 years. Motors and heads are subject to failure over time or from excessive use or mishandling.
CD **DVD** **Blu-Ray**	Light and small Very portable – can even be sent through the post	Reasonably robust and resistant to shocks Easily damaged by mishandling and scratches	CDs and DVDs will start to fail after 10 years; Blu-Ray will fail after 20 years.

Knowledge check

?

8 What are the advantages of solid-state storage over magnetic hard disks?
9 Why might a user choose magnetic hard disks over solid-state storage?
10 What is the most suitable medium for distributing high-definition video films?
11 Identify and describe three factors to consider when choosing secondary storage media and devices.

1.2.3 Units

The units of data storage

A computer uses electronic circuits etched onto computer chips to store data and instructions. These circuits contain electronic switches made from tiny transistors. Each switch can be in one of two states: **on** or **off**. The two states are represented by the numbers **1** or **0**. A computer uses combinations of these 1s and 0s to represent data and instructions.

Beyond the spec

In some sources, you will find people referring to one kilobyte as 1024 bytes, one megabyte as 1024 kilobytes and so on. However, there are different prefixes for this – for example, 1024 bytes is called 1 kibibyte. (1024 might seem like a strange number but it is used because it is a 'neater' number when written in binary.)

Key point

You should use multiples of 1000 in the exam. However, if you do use 1024 for calculations in the exam, you will not be penalised.

There is a number system that only uses the two values 1 and 0. It is called the **binary** number system and it is used to describe the on/off status of all the switches in a computer. One **bi**nary digi**t** is called a **bit**, and the symbol for this is **b**. Computers often group 8 bits together as one unit of data. These 8 bits together are called a **byte**, with the symbol **B**.

4 bits grouped together are called a **nibble**, which has no symbol.

In standard scientific notation the prefix 'kilo' means 1000 – for instance 1 kilometre is 1000 metres. We use kilo, and a whole set of other units, based on this scientific notation.

8 bits (b)	1 **byte (B)**
1000 B	1 **kilobyte (KB)**
1000 KB	1 **megabyte (MB)**
1000 MB	1 **gigabyte (GB)**
1000 GB	1 **terabyte (TB)**
1000 TB	1 **petabyte (PB)**

Worked example

For example, if we have a file that is 2.5 MB, what is that in bytes?

2.5 MB = 2.5 × 1000 × 1000 or 2 500 000 bytes

Knowledge check

12 How many kilobytes are there in 4.5 gigabytes?

13 How many gigabytes are there in 32 terabytes?

Data capacity calculations

As discussed, a computer works in binary because the billions of tiny switches that make up its circuits can only be on or off, which is represented by 1s and 0s. This means that all data needs to be converted into binary so that a computer can work with it. In section 1.2.4 we will describe how that conversion is done for sound, but for now we will look at typical file sizes for a range of file types, measured in bytes.

The choice of media to use is often governed by the number and type of file to be stored. The format used to store the file can also make a significant difference. For example, a document with just 250 characters saved as plain text, rich text or a Word document takes from 251 bytes up to 12 KB depending on how it is saved.

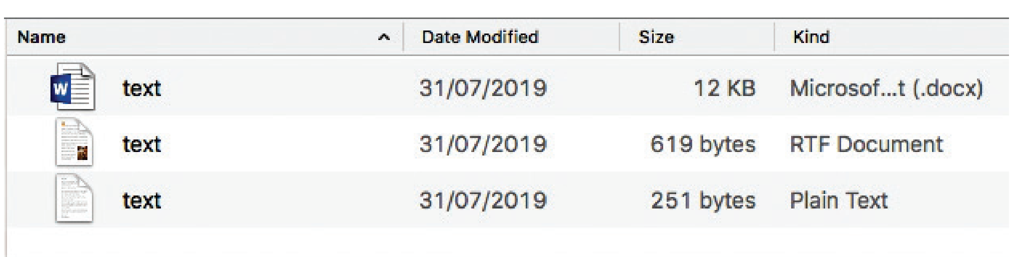


Figure 1.2.6 A 250-character file saved in different formats

FOR THE J277 SPECIFICATION

OCR GCSE (9–1)

COMPUTER SCIENCE

SECOND EDITION

Written by leading Computer Science teachers and endorsed by OCR, this brand-new textbook will guide students through the updated OCR GCSE Computer Science specification topic by topic, and provide them with standalone recap and review sections, worked examples and clear explanations of complex topics.

This Student Book:

- develops computational thinking skills in line with the new Practical Programming element of Component 02
- provides differentiated material with the 'beyond the spec' feature
- includes standalone recap and review sections at the end of each chapter
- provides definitions of technical terms, along with a glossary of words that will be needed for assessment.

George Rouse, **Lorne Pearcey** and **Gavin Craddock** are highly respected and widely published authors of resources.

Endorsed by OCR

- ✓ This title fully supports the specification
- ✓ It has passed OCR's rigorous quality assurance programme
- ✓ It is written by curriculum experts

ISBN 978-1-5104-8416-0

HODDER EDUCATION

t: 01235 827827

e: education@bookpoint.co.uk

w: hoddereducation.co.uk