COMPUTER SYSTEMS

National 5 Computing Science

Alison Cowie

Portobello High School, Edinburgh

Contents

[Data representation 3](#_Toc481695647)

[Using binary to represent data 3](#_Toc481695648)

[Integers 4](#_Toc481695649)

[Real numbers 5](#_Toc481695650)

[Characters 6](#_Toc481695651)

[Vector graphics 6](#_Toc481695652)

[Bitmapped graphics 6](#_Toc481695653)

[Computer structure 10](#_Toc481695654)

[Computer architecture 10](#_Toc481695655)

[Program instructions 12](#_Toc481695656)

[Translator software 12](#_Toc481695657)

[Environmental impact 13](#_Toc481695658)

[Security precautions 15](#_Toc481695659)

[Firewalls 15](#_Toc481695660)

[Encryption 15](#_Toc481695661)

## Data representation

### Using binary to represent data

The number 3862 in decimal means 3 x 103 + 8 x 102 + 6 x 101 + 2 x 100

The number 110101 in binary means 1 x 25 + 1 x 24 + 0 x 23 + 1 x 22 + 0 x 21 + 1 x 20

**Decimal** means numbers are written in base 10 using the digits 0 to 9

**Binary** means numbers are written in base 2 using the digits 0 and 1

For example, a binary piece of data could be 10111001 or 10000010100101000100010110010001

Almost all data within computer systems is held in binary form. This is often referred to as **digital data**

Computer systems use binary numbers to encode all information because

1. Electronically it is easier to represent just two values
2. It is easier to build electronic circuitry for carrying out calculations when only two values are involved
3. The significant advances in hardware mean that it is possible to store and process enormous quantities of digital information (information that is held in codes made up of 1s and 0s)

How does a computer system know what 10000010100101000100010110010001 represents? Is it a piece of text? Is it a number? Is it a piece of audio, video or graphic?

It depends largely on what the computer is *expecting*! Each piece of data will carry a code indicating what type of data it is

Each type of data is stored in different ways

### Integers

An integer is a number with no fractional part e.g. 76, -8945, 100006, -2

Storing only positive integers Range of values possible with n bits is 0 to 2n-1

How to turn an integer from decimal to binary

 e.g. What is 73 in binary? Put 1s and 0s in the right columns so that the numbers add up to 73

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |

 Answer: 73 in decimal = 01001001 in binary

How to turn an integer from binary to decimal

 e.g. What is 10011100 in decimal?

 Put the 1s and 0s under the column headings and add up the numbers where there are 1s

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |

 Answer: 10011100 in binary = 156 in decimal

### Real numbers

A real number is a number that has a fractional part e.g. 67.341, -0.0005, -100.3

The number is stored using *floating point representation* using a *mantissa* and an *exponent*

 e.g. 6527.802 = 0.6527802 x 104

 The decimal point has “floated” to the front of the number

 The 6527802 is called the *mantissa* and the 4 is called the *exponent*

The mantissa and the exponent would be stored as separate numbers. The computer recreates the number for calculations when it needs to

### Characters

**character** – any symbol or letter found on a keyboard. Each character has a unique numerical value inside the computer e.g. ‘A’ is 01000001 (=65)

**character set** **of a computer** – a list of all characters a computer can process and store. Different computers have different character sets

**control characters** – do not print on the screen in the normal way. They control certain operations of the computer such as ‘cursor up’, cursor down’, ‘clear screen’, ‘tab’, ‘return (enter)’, ‘delete’ etc.

**(EXTENDED ) ASCII**

* American Standard Code for Information Interchange
* An ASCII code is stored using one byte
* 28 = 256 therefore ASCII can represent 256 different characters
* In ASCII, codes 0 to 31 are used for control characters e.g. 0001001 is the tab key

**Aside:** There are other coding systems for characters. For example, Unicode is designed to represent all the characters from all of the world’s major languages. It was originally set up as a 16-bit code. 216 = 65536 different characters. Nowadays even more bits are used to give an even greater range of data

### Vector graphics

**Representing graphics using vector graphics**

* + - the graphic is made up of **objects** each having a set of **attributes**
		- **objects:** rectangle, ellipse, line, polygon **attributes:** co-ordinates (position), fill colour, line colour
		- e.g. a red rectangle. The *object* is a ‘rectangle, its *attribute* of fill colour has the value red
		- an example of a vector graphic package is **Draw**
		- the file size increases as the complexity of the graphic increases. The more objects that are added, the more data that must be stored

### Bitmapped graphics

**Representing graphics using bitmaps**



* **Pixel** - short for **pic**ture **el**ement
* **Bitmap** – a grid of bits used to represent the pixels of the graphic
* **Black and white bitmapped graphics**
	+ one bit per pixel (usually set to 0 for white, 1 for black)
	+ This means that the number of bits = number of pixels
* **Colour bitmapped graphics**
	+ - Several bits per pixel e.g. 10 bits per pixel means 1024 possible colours for each pixel
		- **True colour** = 24 bits per pixel (the limit at which the human eye can distinguish between colour)
* **Resolution** of a graphic
* determines the quality (accuracy) of the graphic
* the smaller the pixels, the higher the resolution and the better the quality
* **Screen resolution** e.g. 1024 x 768 (horizontal x vertical)
* **Capture resolution** e.g. digital camera with a total resolution of 8 megapixels
* **Storage requirements** – file size remains constant regardless of the complexity of the image since the whole screen is saved
* **Increasing the resolution** of a bitmapped graphic >> increased storage requirements

**Bit depth**

If more than one bit is used for each pixel, then shades of grey or even colour can be represented. For example, if two bits are used for each pixel then there are four possible combinations (00, 01, 10, 11) which can be used to represent the colour of a pixel (e.g. white, light grey, dark grey, black). The greater the number of bits used for each pixel, the greater the variety of colours)

***Definition*: Bit depth (colour depth) – the number of bits used to represent colour or shades of grey in a graphic**

How many bits would 128 colours require?

Ans: 128 = 27 so 7 bits required

What is the bit depth of a graphic allowing up to 1024 colours?

Ans: 1024 = 210 therefore bit depth = 10

A graphic has a bit depth of 6. How many colours does this allow? Ans: 26 = 64

**Comparison of bit mapped and vector graphics**

**EDITING**: When two shapes overlap in a bit mapped package such as *Paint* compared with a vector graphics package, the shape on top will “rub out” the shape underneath

**FILE SIZE**: The file size of a bit mapped graphic stays the same regardless of the amount of detail in the graphic because each pixel is stored regardless of its colour, whereas a vector graphic file size increases as objects are added

The description of each object making up a vector graphic is called its “set of attributes”

**RESOLUTION INDEPENDENCE**: This is when the resolution of the graphic on the screen does not affect the resolution of the printed graphic. This is a characteristic of **vector** **graphics** files because **the processor sends the file of object attributes** that represent the graphic to the printer and the printer prints the graphic off using its own resolution settings

When a **bit map** is printed out, the resolution of the printout is the same as the resolution of the bit map on the screen regardless of the resolution of the printer. This is because **the processor sends the bit map** as it is to the printer and it is printed as it is

**EDITING at pixel level**: Possible with a bit-mapped graphic

To edit an object in a vector graphics file, the object needs to be selected and then the values of its attributes (e.g. colour (attribute) red (value) to colour green

**The relationship between bit depth and file size**

As bit depth and resolution increases, the quality of the image will increase but the file size will too, meaning increased storage requirements and slower transmission times

***Definition*: True colour – where 24 bits per pixel are used to represent colours. This is considered to be enough to fool the human eye that it is seeing ‘true’ colour and gives a total range of 16777216 colours**

## Computer structure

### Computer architecture

Memory

RAM

ROM

Processor

Registers

Data bus

ALU

Control unit

Address bus

#### The processor (sometimes called the CPU or Central Processing Unit)

 **ALU – Arithmetic Logic Unit**

 Carries out calculations and makes decisions using logical operations

e.g. if a program needs to add 2.73 and 5.76, these would be brought into the ALU where they would be added together

e.g. if a program needs to compare the values of variables x and y, this would be done in the ALU

 **Control unit**

Controls all the parts of the processor and makes sure that program instructions are executed in the correct order

Makes sure that all events in the processor happen in the correct place and at the correct time

 **Registers**

 Registers are storage locations inside the processor chip that hold data while the processor is using it

**MAR** – Memory Address Register. This holds the *address* of the location in memory that is currently being accessed

 **MDR** – Memory Data Register. This holds the *contents* of the location currently being accessed

#### Memory

**RAM -** this is called **volatile** memory which means that it holds its data as long as the computer is switched on, but when the computer is switched off any data in RAM is lost

**ROM -** this is called **non-volatile** memory which means that it holds its data permanently even when the computer is switched off

N.B. Flash ROM can be reprogrammed whilst inside the computer and is now a very popular backing storage e.g. USB flash drives and digital camera memory cards. It is replacing hard disk drives etc.

#### Buses

Buses are sets of wires connecting parts of the computer together

 **Data bus**

 Bi-directional. This bus transfers data between main memory and the processor

 The number of wires making up the bus determines the *quantity of data* the bus can carry

 e.g. if the bus has 8 wires

 then the number of possible numbers it can carry is 28

 and the greatest number it can carry is 28 – 1 (assuming just positive numbers)

**Address bus**

Uni-directional. The address bus indicates which address in memory is being *read from* or *written to*

The number of wires making up the bus determines the *number of storage locations* the processor can access

e.g. if the bus has 32 wires then the number of possible addresses it can access is 232 and the range of addresses is from 0 to 232 - 1

### Program instructions

High level language instructions need to be translated into machine code instructions (in binary) in order for the processor to be able to execute them. This means that, in the end, even the most complicated programs like a computer game or an operating systems are just a huge collection of 1s and 0s !

Every processor has its own instruction set.

Portable software: A piece of software that can run on different computer systems i.e. it can be translated into the machine code of a variety of processors e.g. for an Intel processor, for an AMD processor etc.

A short machine code program might look like this

 Operator Operand *What the program means*

 01001000 00001100 *Load accumulator (in the ALU) with the contents of address 12*

 01110110 00000101 *Add to the accumulator the number 5*

 01011001 00001100 *Store the value of the accumulator in location with address 12*

Obviously, programming in binary machine code is time-consuming and hugely error-prone. Coding in machine code is usually only done in situations where the programmer needs to refer directly to the individual parts of the processor chip or refer directly to locations in main memory

### Translator software

High-level languages are **designed to be read by humans**. They try to use simple, common language to form instructions for the processor to carry out. For example, LiveCode

**on** placeFlags

 **repeat** with flagNumber = 1 to 4

 **set** the location of image id flagData[flagNumber]["imageID"] to flagData[flagNumber]["current position"] \* 100, 200

 **end** **repeat**

**end** placeFlags

Other examples of high level languages are C#, C++, Java, PHP, HTML, JavaScript, Perl, Ruby, Python, Scratch

However, before these instructions can be executed by the processor, they must be **translated into low-level** instructions. Each high-level instruction will translate into several low-level instructions

High-level language translators come in two forms:

#### Interpreters

The program (source code) is in memory along with the interpreter which **translates and executes each line of the program in turn**

**Advantage:** Often used during development time. It is easier to locate errors since the interpreter can highlight lines in the source code where the program fails

**Disadvantages:** The run (execution) is slower because of the simultaneous translation and because any repeated line will be re-translated. In addition, the interpreter software takes up memory

#### Compilers

The program (source code) is translated into **machine object code** that can be executed by the processor independently of the original source code. The only software that needs to be in main memory to run the program is the object code

**Advantages:** The run (execution) will be fast because the translation has been done separately. In addition, the original program source code is not available for people to view and possibly plagiarise.

**Disadvantage:** All errors need to have been corrected before source code can be compiled to run independently

## Environmental impact

A computing-related carbon footprint measures the amount of greenhouse gases emitted during the

* **production**
* **use**
* **disposal** of computer equipment.

Using the Internet consumes energy at the user end but also involves electricity usage and therefore carbon emissions by the companies that run the websites that users view

On the plus side, computers can help reduce carbon emissions by making equipment more efficient. An example of this would be modern engine-management systems on cars that control the car engine to make it more fuel efficient

Modern manufacturers of IT equipment are constantly looking for ways to reduce the energy consumption of devices from low–power-consumption processors to low-energy-rated LED monitors

With modern advances in networking, employees can telecommute, which means that they can work from home without having to physically commute by car, bus or train. Similarly, organisations can save money and the environment by holding meetings using teleconferencing. Teleconferencing involves people meeting together using webcams and microphones instead of having to leave their normal office and travel to another city or country to meet with colleagues or customers.

Manufacture

Chemicals/substances are used to manufacture computers:

*Problem:* Unsustainable – manufacturers will run out of lead, arsenic, antimony trioxide, selenium, cadmium, chromium, cobalt, mercury etc.

*Problem:* Many are toxic – there are health risks for factory workers

Energy sources are needed to power the factories that make computers:

*Problem:* Some energy sources: oil, gas, coal, nuclear are non-renewable

*Problem:* Energy production releases CO2 and other gases that contribute to global warming

Solutions:

Use non-toxic, biodegradable substances to make computers and renewable energy sources: solar, wind, wave, hydro-electric in a way that doesn’t impact the environment

Use

Energy is used to power computers

*Problem:* Some energy sources: oil, gas, coal are non-renewable. This is unsustainable

*Problem:* Energy production releases CO2 and other gases that contribute to global warming

Solutions:

Use low-power computing devices and make the computing device more energy efficient by adjusting the monitor and power-down settings. Computers can also be left on standby when not in use to cut power consumption

Disposal

What is WEEE?

WEEE is waste electronic and electrical equipment like TVs, fridges and computers. Many smaller items slip through the net and are not treated or recycled and get buried in landfill sites. This wastes resources and puts a major strain on the environment.

Stores that sell electrical equipment should provide facilities to take back WEEE. They must either offer in-store take-back or be a part of a distributor take-back scheme (and so tell customers where to take it). There should also be information in stores about the environmental impact of WEEE and what customers can do to help reduce the impact their purchases make on the environment.

Consumers should consider the environmental consequences before replacing mobile phones, computers or other gadgets. Could the equipment be upgraded or a refurbished product be bought instead?

Computer equipment can contain lead and other hazardous materials such as mercury and cadmium. Among the risks when exposed to these materials are lead poisoning, high blood pressure, iron-poor blood, liver disease, nerve and brain damage, and cancer. Because of the dangers involved with having these devices lying around in public landfills, it is important to dispose of equipment responsibly. Toxic components leach into the environment and this harms our ecosystem and, if recycling is done by low paid workers and/or in developing countries, there are usually increased health risks and environmental impact

Solutions:

*Recycle* responsibly

*Reuse* – give unwanted computing devices to charities as soon as you stop using them

*Reduce* – think before purchasing a new device

## Security precautions

### Firewalls

A firewall is a program or piece of hardware that helps stop hackers, viruses and worms from entering a computer over the internet. Some firewall software is free to download off the Internet e.g. ZoneAlarm

### Encryption

Encryption is the encoding of data held on the system. This means that if the network is hacked into the data are meaningless to the hacker because they do not have the security key to unlock or decipher the encrypted data.