3 Data Representation

3.13 Numbering Systems

- An **integer** is any whole number.
  $\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$
- A **natural number** is a whole number that is used for counting.
  $\mathbb{N} = \{0, 1, 2, \dots\}$
- A **rational number** is any value that can be expressed as a fraction.
  $\mathbb{Q}$
- An **irrational number** cannot be expressed as a fraction, and has an endless series of non-repeating digits.

  An irrational number cannot be correctly represented using a finite number of digits, therefore a **rounding error** will occur.

A **real number** is any natural, rational or irrational number. The **set of real numbers** $\mathbb{R}$ is defined as the set of all possible real world quantities.

Ordinal numbers describe the numerical position of objects - first, second, etc.

- Natural numbers are used for **counting**.
- Real numbers are used for **measurement**.
- Ordinal numbers are used as **pointers** to a particular element in a sequence, or to define the position of something in a list.

Number Bases

- **Denary** uses the digits 0 through 9 and has a base of 10.
- **Binary** uses only the digits 0 and 1 and has a base of 2.
- **Hexadecimal** uses digits 0-9 and letters A to F and has a base of 16.

The numbering base can be written as a subscript 11_{16}.

The hexadecimal system is used as the **shorthand for binary**, since

- It is simple to represent a byte in just two digits.
- **Fewer mistakes** are likely to be made in writing a hex number than a string of binary digits.
- It is easier for computer users to **remember a hex number** than a binary number.

Colour codes often use hexadecimal to represent the RBG values, as their are easier to remember than a 24-bit binary string.

3.14 Bits, Bytes and Binary

- A **bit** is the **fundamental unit of information** in the form of either a single 1 or 0.
- A **byte** is a set of eight bits.
- A **nibble** is a set of four bits.

The number of values that can be represented with n bits is 2^n.

A **kibibyte** KiB is 1024 bytes, whereas a **kilobyte** KB is 1000 bytes.

Character Sets

ASCII (Americal standard code for information interchange) is a code for **representing characters** on the keyboard.

- Uses 7 bits which form 128 different **bit combinations**.
- The first 32 codes represent **non-printing characters** used for control, such as backspace, enter, escape, etc.
- An 8-bit version **extended ASCII** was developed to include an additional 128 combinations.

By the 1980s, several coding systems had been introduced all around the world that were all **incompatible with one another**. A new 16-bit code called the **Unicode** (UTF-16) was introduced

- Allows for 65,536 different combinations that could represent alphabets from dozens of languages.
- The first 128 codes were the **same as ASCII** so compatibility was retained.
- A further version of Unicode called UTF-32 was developed to include just over a million character - more than enough to handle most of the characters from **all the languages**.

Unicode encodings take **more storage space** than ASCII, significantly **increasing file sizes** and transmission times.

Error Checking and Correction

Bits can change erroneously during transmission owing to **interference**.

- A **parity bit** is an additional bit used to check that the other bits transmitted are likely to be correct.
- **Majority voting** is a system that requires each bit to be sent three times.

If a bit value is flipped erroneously during transmission, the recipient computer would use the **majority rule** and assume that the two bits that have not been changed is correct.

Majority voting **triples the volume of data** that is sent.

- A **checksum** is a mathematical algorithm that is applied to a unit of data.

  1. The data in the block is used to **create a checksum** value, which is **transmitted with the block**.
  2. The same algorithm is applied to the block after transmission.
  3. If the **two checksums match**, the transmission is deemed successful.

     Otherwise, an error must have occurred during transmission, and the block should be **transmitted again**.

- A **check digit** is an additional digit at the end of a string designed to check for mistakes in an **input or transmission**.

  Printed books have a unique **ISBN** (International standard book number).

3.15 Binary Arithmetic and the Representation of Fractions

An **overflow error** occurs when a **carry from the most significant bit** requires a 9th bit, but only 8 bits are used to store the result of an addition.

Signed and Unsigned Binary Numbres

- An **unsigned representation** of binary number can only represented positive numbers.
- A **signed representation** can represent both positive and negative numbers.

Two's complement is a representation of signed binary number.

It works similar to numbers on an **analogue counter** - moving the wheel forward 1 will read 0001, back one the reading becomes 9999, which is interpreted as -1.

The range that can be represented by two's complement using n bits is given by

$$-2^{n-1} \ldots 2^{n} - 1$$

With 8 bits, the maximum range that can be represented as -128 (1000 0000) to 127 (0111 1111). The leftmost bit is used as a **sign bit** to indicate whether a number is negative.

To negate a binary number

1. Flip the bits.
2. Add one.

Binary subtraction can be done using the **negative two's complement number**, then adding the second number - the carry on the addition is ignored.

Fixed Point Binary Numbers

Fixed point binary numbers is a way to **represent fractions** in binary. A **binary point** is used to separate the **whole place values** from the **fractional part** on the number line.

Some fraction **cannot be represented** at all, as they will require an infinite number of bits to the right of the point. The number of fractional places would therefore be **truncated** and the number will not be accurately stored, causing **rounding errors**.

Two digits after the point and only represent 0, 1/4, 1/2, 3/4 and nothing in between.

3.16 Bitmapped Graphics

A bitmap image contains many **pixels** that make up the whole image.

- A pixel is the **smallest identifiable area** of an image.
- Each pixel is attributed a binary value which **represents a single colour**.

The **resolution** of an image determine the number of pixels within it. The greater the number of pixels it contains, the sharper the image, as the pixels must become smaller to fit its boundaries.

Resolution of an image can be expressed as

- **Width in pixels** × **height in pixels**.
- Pixels per inch - indicating the density of the pixels.

The number of bits per pixel is referred to as the **colour depth**. The number of bits determines the **number of combinations**, this determines the **number of colours** that a pixel can represent.

Metadata is data about data - details such as the image **width in pixels, height in pixels and colour depth**.

Vector Graphics

Vector images are made of **geometric shapes or objects** such as lines, curves, arcs and polygons. A vector file stores only the necessary details about each shape in order to **redraw the object** when the file loads.

E.g. the properties of an image of a circle.

- The **position** of its centre,
- Its **radius**,
- Fill and line colour,
- Line weight.

These properties are stored in a **drawing list** which specifies how to redraw the object.

Regardless of how large an image is draw, the image will **always be sharp**, and the amount of data required to store the image will not change.

Advantages of vector graphics

- Usually has a much smaller file size.
- Will **scale perfectly**.
- Used for logos, so the image will be sharp when printed on anything from a business card to a billboard.

Disadvantages of vector graphics

- Cannot easily replicate an image with continuous areas of changing colour.
- Individual pixels cannot be changed.