Higher Computing Science
Data Representation
Summary Notes

Storage of program instructions

Instructions are stored using the processor's own language called **machine code**. Different processors (or families of processors) use different machine code.

Positive integer storage

Every type of data in a computer system is stored using the binary number system, where 1 is represented by ON and 0 is represented by OFF.

Positive integers (whole numbers) can be represented as shown below.

Examples (8 bit)

128	64	32	16	8	4	2	1		
0	1	0	1	1	0	1	1	=	91
1	1	1	1	1	1	1	1	=	255
1	0	0	0	0	0	1	0	=	130

- The **range of numbers** that can be stored depends on the number of bits available in memory to store it.
- The highest number that can be stored in n bits is $\mathbf{2^n - 1}$
  (e.g. 8 bits can store values from 0 up to 2^8-1 = 256-1 = 255)

Negative integer storage

Positive and negative integers can be stored using the **two's complement** system.

Make the leftmost place value negative.

Examples (8 bit)

-128	64	32	16	8	4	2	1		
1	0	0	1	1	1	1	1	=	-97
0	0	1	1	1	0	1	0	=	58
1	1	1	0	0	1	0	0	=	-28

- The range of integers which can be stored in this way depends on the number of bits available.
  - **Example 1** – 8 bit two's complement - values can range from 10000000 (-128) to 01111111 (127)
  - **Example 2** – 4 bit two's complement – values can range from 1000 (-8) to 0111 (15)

Real number storage

Real numbers (numbers with a fractional part) are stored using the **floating point** system, which stores the **mantissa** and the **exponent**.

Example 1

Fixed Point	Signed bit (1 bit	Mantissa (15 bits)	Exponent (two's complement)	Notes
1000101.101	0	100010110100000	00000111	• Signed bit is 0 because the number is positive (no negative sign) • The decimal point is moved 7 places to the left so that it is in front of the first 1 therefore it is positive 7
Final answer	010001011010000000000111			

Example 2

Fixed Point	Signed bit	Mantissa	Exponent	Notes
-0.00010111	1	101110000000000	11111101	• Signed bit is 1 because the number is negative • The decimal point has moved 3 place to the right therefore the exponent is -3 (convert to 8 bit two's complement)
Final answer	110111000000000011111101			

Example 3

Floating Point Number Signed bit (1 bit	Signed bit	Mantissa (15 bits)	Exponent (two's complement)	Final answer
010001011010000000000111 0 100010110100000 00000111 (rewrite to separate signed bit, mantissa and exponent)	0 +ve number	0100010110100000000001 11 Remove extra 0's 1000101101	00000111 Convert to decimal = 7 Decimal point moves 7 steps from first 1	1000101.101

Example 4

Floating Point Number Signed bit (1 bit	Signed bit	Mantissa (15 bits)	Exponent (two's complement)	Final answer
110111000000000011110101 1 101110000000000 11110101	1 -ve number	101110000000000 10111	11111101 Convert to decimal = -3	-0.00010111

- Floating point representation is a very useful way of storing very large or very small numbers, although some accuracy may be lost depending on bits available for storage.
- Increasing the number of bits available for the mantissa means the **accuracy/precision** of the numbers improves, since more digits can be stored.
- Increasing the number of bits available for the exponent means the **range** of numbers that can be stored increases.

Character storage

Text is converted into binary using a code, where each character is represented by a binary number.

- **ASCII** code converts each character into an 8 bit binary number.
  This allows for 256 different characters (2^8=256).
- **Unicode** converts each character into a 16 bit binary number.
  This allows for 65536 different characters (2^{16}=65536). This allows representation for all possible characters from all the known alphabets in the world.
    - Advantage of Unicode over ASCII is that it can represent a larger range of characters
    - Disadvantage of Unicode over ASCII is that file sizes are larger as is uses 2 bytes per character compared to ASCII's 1 byte per character

Graphic storage – bitmapped

A bitmap image is stored as a sequence of binary numbers, each number representing the colour of a pixel.

- The number of bits used to store the colour of each pixel is known as the **bit depth**.
- RGB colour codes are created by using 8 bits for each of the three primary colours (red, green and blue) then combining them to produce a 24 bit colour code.
- **True colour** bitmapping uses 24 bits per pixel (16 777 216 colours).

Graphics storage – vector

Vector graphics store the attributes of every object in the image (e.g. circle: centre x, centre y, radius, fill colour, line thickness, etc).

Bitmapped	Vector
Manipulate at pixel level.	Manipulate at object level.
Typically a larger file size.	Typically a small file size, although it will increase as more objects are added to the image.
File size is affected by resolution of image.	
File sizes are fixed regardless of detail in graphic.	File size increases as more objects are stored
Becomes pixelated (blocky) when enlarged.	Can be enlarged without affecting quality (it is resolution independent).
Ideal for photos and realistic images.	Ideal for simple logos on websites, etc.

Compression

Compression is required to reduce the size of bit mapped images.

Lossy compression – quality of the graphic is reduced with compression
Lossless compression – quality of the graphic is unaffected by compression

Format	Compression Type	No of Colours	Transparency	Advantages	Disadvantages
Jpeg	lossy	2^{24} = 16777216	Yes	Small file size	Lossy compression – loss of quality
Gif	lossless	2^8 = 256 colours	Yes	Supports animation	Limited no of colours
Png	lossless	2^{32}=4294967296 colours	Yes	Large no of colours, good transparency support	