Higher Computing Science
Software Design and Development – Low Level operations
Summary Notes

Units of storage

The following units are used when referring to file sizes or storage capacity.

- Bit – a single binary digit, 1 or 0.
- Byte – 8 bits, e.g. 10010001
- Kilobyte (KB) – 1024 bytes
- Megabyte (MB) – 1024 KB
- Gigabyte (GB) – 1024 MB
- Terabyte (TB) – 1024 GB
- Petabyte (PB) – 1024 TB

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 **$2^n - 1$**
  (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. To represent a decimal number using two's complement simply make the leftmost place value (i.e. the most significant bit) 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.
  (e.g. for 8-bit two's complement, values can range from 10000000 (-128) to 01111111 (127)

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
In the number 0.10100001 x 2^{1011}, the mantissa is 0.10100001 and the exponent is 1011.

- 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 **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.

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/colour depth**.
- If the bit depth of the file is less than 24 then the actual **RGB code** may be stored in a separate colour palette, in order to reduce file size.
- 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).

- 2D vector graphics can be created using the **SVG** language while 3D vectors can be created using **VRML**.

Bitmapped	Vector
Manipulate at pixel level.	Manipulate at object level.
Typically a larger file size. File size is affected by resolution of image.	Typically a small file size, although it will increase as more objects are added to the image.
Becomes pixellated (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.

Video storage

Video is stored as a sequence of bitmapped images. The illusion of motion/movement is created when these are played back in quick succession.

- The **frame rate** of video is the number of invidual images that are displayed for every second of footage. The frame rate affects the quality of playback and the file size. A typical frame rate for a video is 25/30 fps (frames per second), although for smoother footage (especially for games) this may be much higher (50/60 fps).
- Other factors affecting quality are the **resolution** (number of pixels) and the **bit/colour depth** (number of colours). The principles here are exactly the same as when storing bitmapped graphics.

Audio storage

Sound is stored digitally by sampling the original analogue sound.

- **Sampling** means taking measurements of the signal. This process of digitising sound means that quality will be lost.
- **Sample frequency** is the number of times each second that the sound is sampled. Sample frequency is measured in Hertz (Hz). A frequency of 1Hz means one sample per second. The greater the sampling frequency, the better quality the sound but the larger the file size. Generally this is measured using **KHz** (1000s of hertz).
- **Sample/bit depth** refers to the number of bits available to record the sound level at each sample. The greater the sampling depth (i.e. the more bits available), the better quality the sound but the larger the file size.
- **CD quality** sound has a sample frequency of **44.1 KHz** and a sample depth of **16 bits** and is **stereo**.

Computer architecture - Processor

The CPU (Central Processing Unit) is the main processor in a computer.
It has 3 main components:

- **Control Unit** – manages the execution of program instructions and fetching from memory.
- **ALU (Arithmetic and Logic Unit)** – performs arithmetical operations (add, subtract, multiply, divide) and logical decisions (AND, OR, NOT).
- **Registers** – Temporary storage locations inside the processor, such as the Instruction Register, the Accumulator and the Program Counter.

Computer architecture - Memory

Memory is used to store programs while they are running. It is organised into millions of individual storage locations, each with a unique binary address.

- **RAM (Random Access Memory)** – can be written to, stores program which are in use, loses its contents when power is switched off.
- **SRAM (Static RAM)** is a type of RAM with fast access speeds, but expensive per MB, good for registers and cache memory.
- **DRAM (Dynamic RAM)** is a type of RAM with slower access speeds than SRAM but cheaper per MB, suitable for main memory.
- **ROM (Read Only Memory)** – can't be written to, retains contents even when power is switched off. Used to store the **bootstrap loader**, the program which loads up the OS when the computer is turned on.
- **Cache** – an area of fast access memory, physically close to the processor used to store frequently used instructions and therefore improve system performance.

Multi-level cache may be used – L1 (fastest but smallest) is checked first, then L2, then L3, etc.

Computer architecture - Buses

Buses are sets of wires which connect the processor and memory.

- **Data bus** – caries binary data to and from memory.
- **Address bus** – carries the memory location which data is to be written to or read from.
- **Control bus** – comprises several discrete lines, including:
    - **Read line** – used to signal a memory read operation.
    - **Write line** – used to signal a memory write operation.
    - **Clock line** – carries electronic pulses to CPU to keep components synchronised.
    - **Reset line** – carries a signal which returns processor to its original state.
    - **Interrupt line** – carries a signal to processor when a peripheral requires attention.

Computer architecture – Fetch-execute cycle

The way the CPU repeatedly fetches an instruction from memory, decodes it, and then executes it, is called the **fetch-execute cycle**.

Memory Read operation

1. CPU sets up the address bus with the required memory address by placing a value in the Memory Address Register.
2. The Control Unit activates the read line.
3. The contents of the storage location with that address are released on to the data bus and copied into the Memory Data Register.
4. The data is decoded and executed.

Memory Write operation

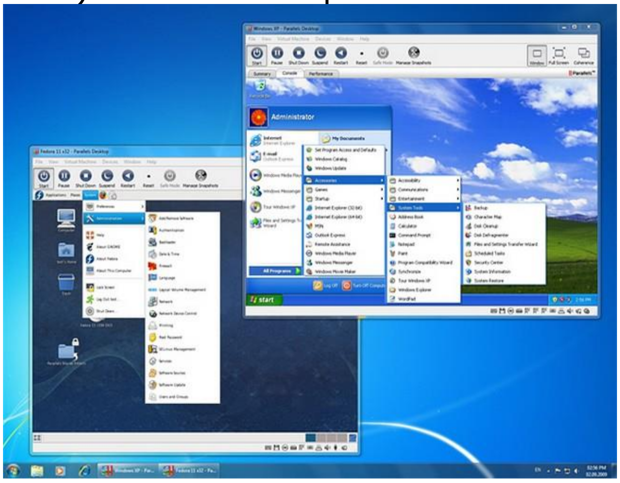
1. CPU sets up the address bus with the required memory address by placing a value in the Memory Address Register.
2. CPU sets up data bus with value to be written to memory, by placing the value in the Memory Data Register.
3. The Control Unit activates the write line.
4. The contents of the Memory Data Register are transferred to the appropriate storage location.

An easy way of recalling the order of operations for the fetch-execute cycle is simply by remembering to use the alphabet (i.e. ABCDE).

AB – **A**ddress **B**us pinpoints address in memory
C – **C**ontrol unit activates the read/write line.
D – instructions/data is passed along the **D**ata bus
E (read operation only) – instructions are **E**xecuted by the processor.

Virtual machines

A virtual machine is an emulation of the software used by a real computer system. One example of this would be to allow a modern computer system to run older software (e.g. an outdated operating system) that isn’t compatible.



- One physical computer is used to perform like other computers, so no need to purchase different hardware.
- Less space required for several computers.
- Fewer computers being powered so reduced energy consumption.
- A virtual machine does not run as efficiently as a real machine, since it cannot directly access hardware components.
- Running virtual machines can be processor and RAM intensive, resulting in poor performance.

Emulators

Emulation software imitates an item of computing hardware, such as a mobile phone or an obsolete computer.

- Game console emulators are a popular way of playing retro video games when you do not have access to the hardware.
- Emulation of mobile devices can be used to test software on other systems whilst it is being developed.
- Disc imaging software emulates a physical storage device such as magnetic tape.