**2.1.1 Fundamentals of computer systems**

**(a) define a computer system**

A device that can: receive, store, process and output data.

**(b) describe the importance of computer systems in the modern world**

They are everywhere and essential to our safety, comfort and wellbeing in the modern world

**(c ) explain the need for reliability in computer systems**

reliability is the probability that a computer will produce the correct outputs in a given time frame.

**(d) explain the need for adherence to suitable professional standards in the development, use and maintenance of computer systems**

standards allow for a variety of hardware or software devices to be compatible with each other.

**(e ) explain the importance of ethical, environmental and legal considerations when creating computer systems.**

Waste Electrical and Electronic Equipment (WEEE) – setup by the European Union.

The UK throw away 2 million tonnes of WEE each year.

Products stripped to raw materials, can be used again.

To avoid cost of recycling products send to developing countries.

Average home spends £45 - £80 a year from keeping items on stand bye.

5% of Europe’s annual energy bill is spend just on cooling computers.

40% total power consumption in legacy data centre.

88.9% power consumption recede in legacy data centres when using alternative methods.

**2.1.2 Computing hardware**

**The CPU:**

**(a) state the purpose of the CPU**

Computer brain, Responsible for all processing the computer does.

Arithmetic Logic Unit (ALU) – Logic circuits, Billions of electrical switches called transistors.

Control Unit (CU) – coordinates parts of the processor.

When processing the CPU – fetches, decodes and executes instruction.

counter – CPU uses program counter to keep track of instructions.

Decode – converts instruction into format understood by CPU.

Execute Step – could include: Loading from memory, storing in memory, preforming a calculation.

**(b) describe the function of the CPU as fetching and executing instructions stored in memory**

Fetch instruction from memory, decode instruction, execute instruction.

CPU contains registers – small storage locations.

Program counter – holds the address in RAM of next instruction to be executed.

After CPU has read program counter, it is updated with following instruction.

**(c ) explain how common characteristics of CPUs such as clock speed, cache size and number of cores affect their performance**

Clock speed – the number of times the fetch, decode, execute cycle can be performed per second.

Modern processors clock speed measured in GHz. 1GHz = 1 billion calculations per second.

Over clocking – increasing the clock speed of a processor, increasing calculations per second.

Overclocking increases change of: instability, system crashing, processor overheating.

Overclocking decrease the overall lifetime of a processor but makes it run faster (clock speed).

Cache found in CPU – small memory location, very fast, where data is stored temperedly.

It is like a waiting room for data before it is moved into RAM.

Web browser will sometimes “cache” webpages to increase time taken to load instruction. This form of cache is different to the CPU’s cache.

HDD cache – location where data is stored so it can be accessed quickly by applications. e.g. you web browser does not have to fully re-download webpage.

The more cores (individual processors) the CPU (referring to the entire silicon chip) has will increase the total overall clock speed.

e.g. a CPU with 4 cores running at 3.7GHz, will give an overall total clock speed of 14.8GHz.

compared to a CPU with 8 cores running at 3.2GHz, will give an overall total clock speed of 25.6GHz.

CPU clock speed can be improved by having more cores, or increasing individual core clock speeds.

Parallel processing is where multiple cores work on same program, at the same time.

Not all tasks befit from parallel processing.

**Binary logic:**

**(d) explain why data is represented in computer systems in binary form**

All data is stored in binary (base 2).

One 1 or one 0 is called a bit, 8 bits makes a byte.

Data is represented in binary form because it can be replicated using transistors in logic circuits. i.e. a transistor is either on or off, likewise 0 or 1 in binary.

**(e) understand and produce simple logic diagrams using the operations NOT, AND and OR,**

**(f) produce a truth table from a given logic diagram:**

AND:

A

B

Y

|  |  |  |
| --- | --- | --- |
| AND - Truth Table | | |
| Input A | Input B | Output Y |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

OR:

A

B

Y

|  |  |  |
| --- | --- | --- |
| OR – Truth Table | | |
| Input A | Input B | Output Y |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

NOT:

A

Y

|  |  |
| --- | --- |
| NOT – Truth Table | |
| Input A | Output Y |
| 0 | 1 |
| 1 | 0 |

**Memory:**

**(g) describe the difference between RAM and ROM**

Read Only Memory (RAM) is volatile, it can only store data when there is a constant electrical current flowing through it. All data will be lost when no electrical current is present.

Read Only Memory (ROM) is non-volatile, data is permanently stored even without a current.

**(h) explain the need for ROM in a computer system:**

Allows data to remain stored when the system is power off.

When the computer powers on, a set on instructions will need to be executed so that all the other components and hardware can be used, else the system will not be able to boot. This job is assigned to a ROM chip called the Basic Input Output System (BIOS).

There are types of ROM which allow you to write to them, through a process called flashing the firmware.

Bricked – when a devices BIOS is erased, resulting in the device being unable to boot.

**(i) describe the purpose of RAM in a computer system**

RAM – temporarily stores data until power is lost.

Data is stored unorder in a random form. All data can be assessed at the same speed.

Ram is needed for: Temporary storage, Loading of Programs, Memory Addresses.

When a program is closed, it’s data is deleted from RAM.

When a program is loaded, it is copied from Hard Drive to RAM, then travels to processor.

RAM contains no moving parts. Stores data electronically, works very fast.

RAM is faster than Secondary Storage. But slower than cache memory.

RAM has the capacity to store many values in an address.

An address specify which value is of interest to us.

8 bit addressing system, organises words into 1 byte.

16 bit addressing system, organises words into 2 bytes.

32 bit addressing system, organises words into 4 bytes.

e.g. in a 32 bit addressing system: first 32 bit word stored at address 0, second word at address 4 etc.

RAM makes reading, saving and modifying data at the address possible.

RAM is temporary storage for programs; programs can write data to RAM or read data from RAM.

Prevents programs from having to write to or read from slower permanent storage, e.g. HDD.

Dynamic RAM (DRAM) – each bit is stored in a separate transistor with an integrated circuit.

Static RAM (SRAM) – stores data in a 6 transistor circuit.

SRAM does not need re-charging as often as DRAM, and can hold content longer.

SRAM can contain values indefinitely as long as power is on.

SRAM is faster than DRAM.

SRAM uses less power that DRAM.

SRAM is more expensive that DRAM.

SRAM commonly used in cache memory.

**(j) explain how the amount of RAM in a personal computer affects the performance of the computer**

RAM provides temporary for programs and data in a computer system.

All data in a computer system is represented in binary.

1 byte = 8 bits, I kilobyte = 1000 bytes, 1 megabyte = 1000000 bytes.

Single Inline Memory Modules (SIMMS)

Double Inline Memory Modules (DIMMS)

The more RAM a computer has, less virtual memory is required.

Virtual memory is much slower than RAM.

Virtual memory is located on system drive, in the paging file.

The processors performance is determined on how quickly it can access memory.

Slow memory access creates a “bottle neck”, which slows down fetch, decode, execute cycle. Slows down processor as it has to wait for instructions to be fetched from memory.

Faster memory access, better performance.

The less RAM a computer has, the more virtual memory is requires, creating a bottles neck as secondary storage is very slow, which slows down the fetch decode execute cycle.

The operating system controls the CPU by loading a program into RAM.

CPU then reads and writes to memory.

The CPU can access each memory location using a unique address.

**(k) explain the need for virtual memory**

When the RAM is full, its contents overflow into virtual memory.

The operating system will always try to move pages which are in use into physical memory (RAM).

Swapping of pages between RAM and virtual memory takes time, which can result in a disk crash.

Using virtual memory is always going to be slower than physical memory alone, the more RAM you have the less virtual memory you are likely to need.

**(l) describe cache memory**

The overall speed at which data can travel is limited by the slowest part, this is called a bottle neck.

CPU works at very fast speed, RAM works much slower than CPU creating a bottle neck.

A bottle neck can be avoided by using static RAM. Static RAM is expensive.

CPU processes instruction, frequently used data is stored in cache, so the CPU does not have to wait for much slower main memory.

The larger the cache memory the slower it becomes.

The fastest and smallest cache memory level 1, is closest to the processor.

Level 1 cache – stores the most frequently used data.

Level 2 cache – next most frequently used data.