Computer Systems and Architecture Revision

*A combination of many connected parts all working together (interacting) to achieve a common goal is called a **Computer System.***

The purpose of a computer system is to take in data, store it, process it and output the result.

Computer systems involve interaction between:
- Hardware
- Software
- Data
- The User

Hardware	The physical components that make up a computer system e.g. CPU, monitor, motherboard
Software	The programs that run on a computer system and control the hardware
Application software	Designed to allow users complete a specific task e.g Excel
System software	Provides a platform for the application software to run on and manages the systems resources. e.g Operating systems like ios or Windows

Description of the systems involved in a computer system

Types of Computer Systems

General Purpose	Perform a wide range of tasks, e.g A laptop
Dedicated systems	Designed to perform a specific function e.g washing machine
Embedded systems	A system built into another system that has a specific function. e.g autopilot in a car.

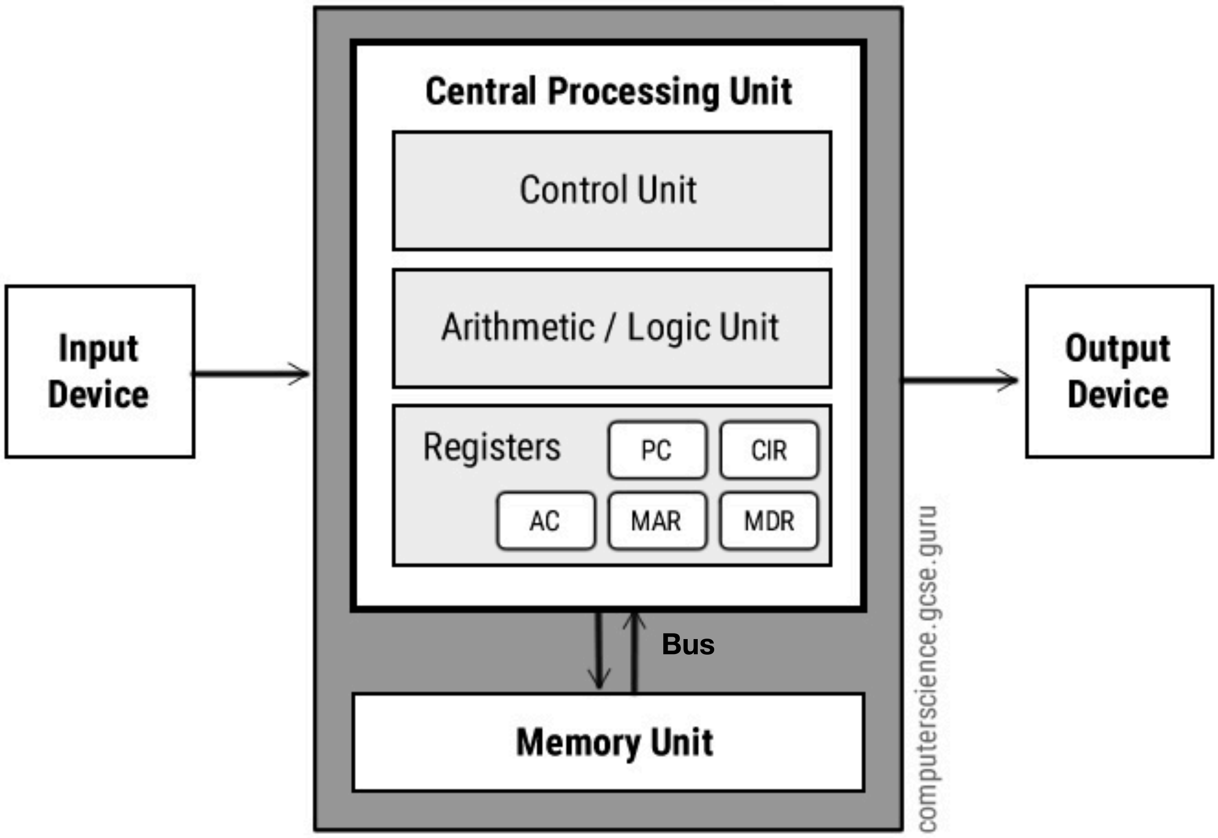
Examples of computer systems

Elements of Computer Systems

Many of todays computers share a common architecture, (think of a house, houses can have different layouts but they all contain common rooms, eg. bedroom, bathroom kitchen), similarly we have many different types of computer systems but the all will have common components. This architecture is known as the von Neumann architecture named after John von Neumann who first described it in 1945.

Von Neumann Architecture

- Memory
- Central Processing Unit
- Input and Output devices
- Computer bus



Typical Von Neumann Architecture

Memory

Primary storage	Secondary storage
Example : RAM (Random Access Memory)	Example: Hard Drive, USB
Data is lost when it looses power (volatile)	Data is not lost when power is lost (Non-volatile)
Temporary data storage	Larger capacity
Read only	Read and Writeable
Quick to respond	Slower than primary storage

RAM - Random Access Memory

- RAM is temporary memory, it holds data and instructions of a program that is currently being executed by the CPU.
- It is constantly being written to and read from.
- Needs a constant power supply to retain its contents.
- When computer is turned off everything stored in its RAM is lost.
- Easy to upgrade and relatively cheap.

ROM - Read Only memory

- ROM is non-volatile, its contents remain even when you turn off the computer.
- You cannot write over the contents once it has been created, it is read only.
- ROM usually programmed by a computers manufacturer.
- Used in BIOS loaded which is required at the start-up of the computer.

CPU

The Central Processing Unit (CPU) has a number of roles:

- Processes data and instructions coming from input and storage devices.
- Controls other components within the computer.
- Carry out computations on data by following instructions

The CPU is made up of the control unit, arithmetic/logic unit (ALU) and registers.

Control Unit: The control unit directs the operations of the CPU.

ALU: Performs all arithmetic and logic operations of the CPU. Receives instructions and data and operates on the data according to the instructions.

Registers: Temporary storage locations that can be accessed quickly when carrying out operations. Small memory, fast.

- Program Counter: Stores where the CPU is in a program sequence. Contains the memory location of the next instruction to be executed
- Accumulator: Stores the output of the ALU
- Memory Address Register: Stores memory addresses, a value that indicates a specific location in memory
- Memory Data Register: Holds data of an instruction temporarily

Clock: Generates a signal consisting of a continuous stream of electrical pulses at a stable frequency (GHz). Ensures all circuits operate together at the right timing. Each impulse activates a fetch-execute cycle.

Bus: Connects CPU and memory. Computers typically have more than one, a communication channel that carries data between internal components. USB = Universal Serial Bus

Inputs and Outputs: Apart from the memory, CPU and bus, all other parts of the von Neumann architecture are considered to be either an input or an output. This includes things like keyboads (input), Display devices (output), secondary storage (both input and output).

Fetch - Decode - Execute Cycle

As a CPU process instructions it runs through a continuous cycle over and over again. In this cycle the CPU retrieves a program instruction from memory and then executes that instruction before then repeating the cycle and getting the next instruction.

1. The address of the next instruction to be executed is copied from the program counter (PC) register to the memory address register (MAR).
2. The Program Counter increments so its points to the **next** instruction to be fetched.
3. The instruction in the MAR is copied to the Current Instruction Register (CIR).
4. The control Unit decodes the instruction in the Current Instruction Register to see what must be done next.
5. The instruction is executed

Factors Affecting CPU Performance

Factor	How it affects CPU Performance
Clock Speed	The faster the clock, the more fetch-execute cycles can take place and therefore more instructions processed
Number of Cores	Dual-core: 2 processor cores, 2 instructions executed simultaneously Quad-core: 4 processor cores, 4 instructions executed simultaneously.
Cache Memory Amount	Very fast memory (temporary) located in the CPU er capacity More cache memory allows CPU to store more instructions, reducing time lost waiting to retrieve data from RAM

Operating System

An operating system is type of system software. It controls the general operation of a computer and provides an easy way for us to interact with a computer and run application software.

The operating system performs several key functions:

- **Interface** - provides a user interface so it is easy to interact with the computer
- **Manages** the **CPU** - runs applications and executes and cancels processes
- **Multi-tasks** - allows multiple applications to run at the same time
- **Manages memory** - transfers programs into and out of memory, allocates free space between programs, and keeps track of memory usage
- **Manages peripherals** - opens, closes and writes to peripheral devices such as **storage** attached to the computer
- O**rganises** - creates a file system to organise files and directories
- **Security** - provides security through user accounts and passwords
- **Utilities** - provides tools for managing and organising **hardware**

The operating system is held in permanent storage, for example on a hard disk.

BIOS and OS startup sequence.

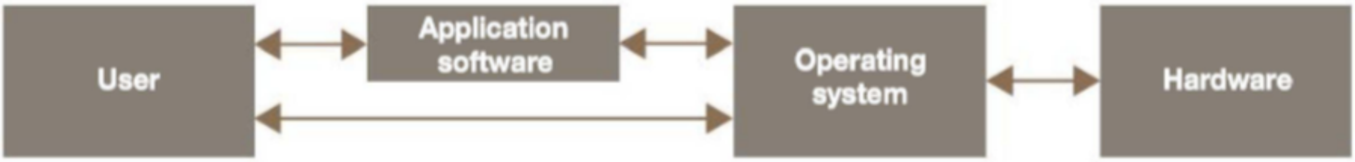
BIOS stands for Basic Input Output System. The BIOS is stored in the ROM. It contains all the basic code for controlling your computer hardware (such as keyboards, mice, monitors and hard drives). After the startup sequence is complete, and control has gone to the operating system, the BIOS does very little.

When you start up your computer, you may see a black screen displaying *"Press F2 for Setup"*. This is the BIOS. By pressing F2, you enter a setup screen where you can change where the BIOS loads the operating system from. The operating system is normally stored on the hard drive, but you can load an operating system from a USB drive or a CD instead.

Startup sequence

- The CPU starts and fetches instructions into RAM from the BIOS, which is stored in the ROM. A small program called the **loader** is held in the ROM. When a computer is switched on, the loader in the ROM sends instructions to load the operating system by copying it from storage into the RAM.
- The BIOS starts the monitor and keyboard, and does some basic checks to make sure the computer is working properly. For example, it will look for the RAM.
- The BIOS then starts the boot sequence. It will look for the operating system.
- If you don't change any of the settings, the BIOS will fetch the operating system from the hard drive and load it into the RAM.
- The BIOS then transfers control to the operating system.

Layer model of a computer system



Computer system layer model

Layered Structure is a type of computer system structure in which the different services of the system are split into various layers, where each layer has a specific well-defined task to perform.

Application software - The application layer contains all the application programs which perform specific tasks. They can write a text document e.g Word, draw a picture, or play a game. Every program you run on your computer's operating system is an application program that helps you accomplish a task or solve a certain problem. The user interacts directly with the application software and may get an output from that software e.g hear music or watch a video.

OS - The main task of an operating system (OS) is to manage computer resources. There are various computer systems that help users to communicate and interact with their computing system. Examples include Windows, Linux, Mac OS, and Android.

Hardware - This layer interacts with the system hardware and coordinates with all the peripheral devices used such as printer, mouse, keyboard, scanner etc. The hardware layer is the lowest layer in the layered operating system architecture.

Advantages :
There are several advantages to this layer design :

1. **Modularity :**
   This design promotes modularity as each layer performs only the tasks it is scheduled to perform.
2. **Easy debugging :**
   As the layers are seperate to each other so it is very easy to debug. Suppose an error occurs in the CPU scheduling layer, so the developer only has to search that particular layer to debug.
3. **Easy update :**
   A modification made in a particular layer will not affect the other layers.
4. **No direct access to hardware :**
   The hardware layer is the innermost layer present in the design. So a user can use the services of hardware but cannot directly modify or access it, unlike the Simple system in which the user had direct access to the hardware.
5. **Abstraction :**
   Every layer is concerned with its own functions. So the functions and implementations of the other layers are abstract to it.

Disadvantages :
Though this system has several advantages over the Monolithic and Simple design, there are also some disadvantages as follows.

1. **Complex and careful implementation :**
   As a layer can access the services of the layers below it, so the arrangement of the layers must be done carefully. i.e You couldn’t have the hardware layer before the OS layer.
2. **Slower in execution :**
   If a layer wants to interact with another layer, it sends a request that has to travel through all the layers present in between the two interacting layers. Thus it increases response time, unlike the Monolithic system which is faster than this. Thus an increase in the number of layers may lead to a very inefficient design.