**Von Neumann Architecture**

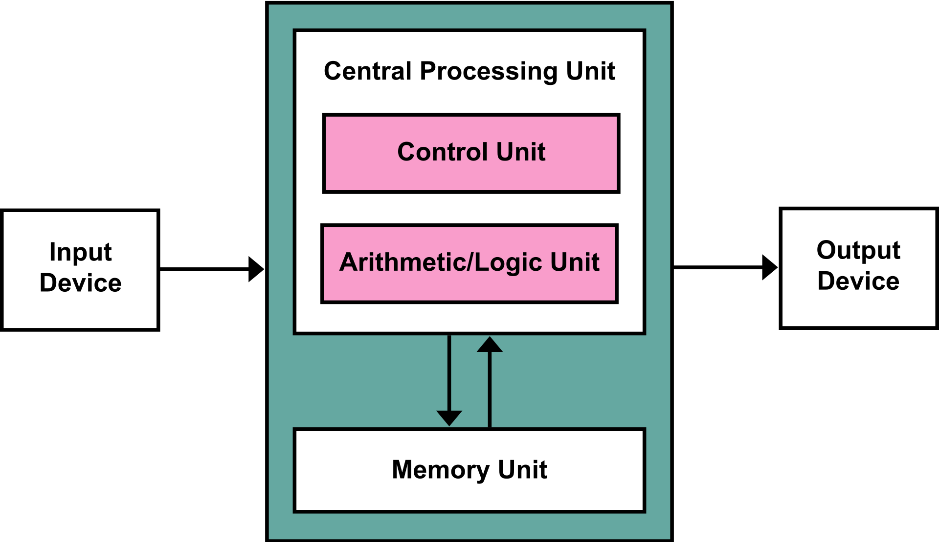
The Von Neumann architecture is a way of describing one design of how a computer works, particularly the CPU (Central Processing Unit), its inputs, and outputs.

It is formed, at a high level, of just four components – the **CPU**, the **memory**, the **inputs** and the **outputs.**

At a lower level, the CPU is split into two main components – the **CU (Control Unit)** and the **ALU (Arithmetic and Logic Unit)**.

* The CU is responsible for timing, control and management of other instructions and the flow of data in and out of the CPU.
* The ALU is responsible for doing arithmetic and conducting logical processing steps.
* Together, they form a working CPU that can fetch instructions from some input, execute them and return the result.

The diagram below shows the general idea:



However, we can go a bit deeper. This architecture is just an abstraction of how the CPU works. It gives no indication about the flow of instructions and data.

The CPU works using something called the **fetch, decode, execute** cycle.

For this to work, we need to use **registers**. A register is simply a special area on the CPU that can store a value.

**Fetch**

* First, we use the **PC** (**Program Counter**) register. This holds the address, in memory, of the next instruction to be executed.
* We copy this value into the **MAR (Memory Address Register).**
* We increment (add one) to the value of the PC register to point to the next instruction for execution.
* We copy the value in the MAR to the **MDR (Memory Data Register)**. This is a register that is a ‘holding area’ for data fetched from, or about to be written to, memory.
* Finally, we fetch the instruction from the MDR into the **CIR (Current Instruction Register).**

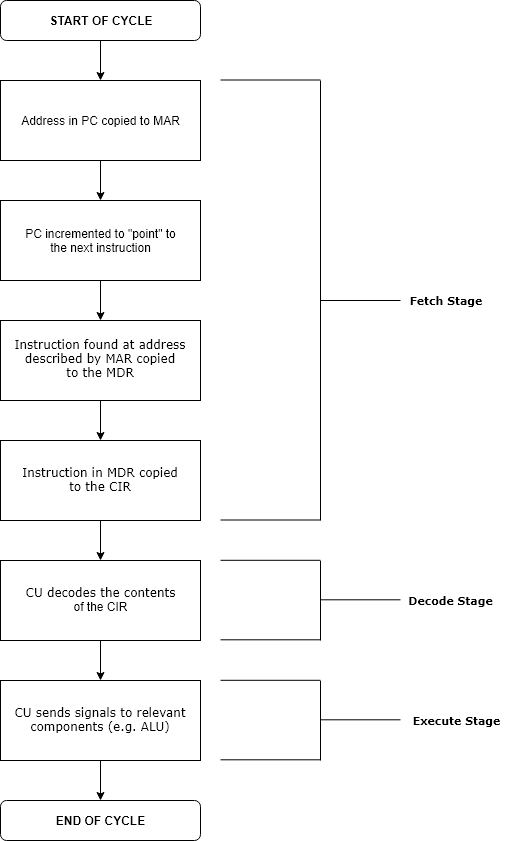
**Decode**

* The CU reads the contents of the CIR and decodes it – i.e. decompresses it, and turns it into an instruction that can be executed, such as ‘Add 1’ or ‘Move contents of address X to address Y’.

**Execute**

* The CU takes the decoded instruction and invokes other parts of the CPU to assist in executing it – i.e. the ALU will perform any calculations. Other types of component will be responsible for different operations – for example some CPUs have a **FLU** (Floating Point Unit) responsible for decimal operations.
* The CU then assembles the output from the other components and stores the output in the MBR ready to write back to memory.

This diagram summarises the F-D-E cycle:



The Von Neumann architecture is named after **John Von Neumann** (1903 – 1957).

There are other computer registers too, depending on the make and architecture of the CPU that are used for e.g. storing characters, storing the output of other operations, or extensions of existing registers.

Additionally, there are two main designs of Von Neumann architecture:

**Types of Design**

**RISC** – Reduced Instruction Set Computer – essentially, this is mostly hardwired. The CPU acts in a predefined way with dedicated electronic circuitry including hardwired logic gates. There is no need for control signals or microprogramming.

**CISC** – Complex Instruction Set Computer – These are the computers we have today, where we can program the contents of main memory and the registers on the CPU.