Syllabus Requirements:

3. Hardware

3.1Computer architecture

Candidates should be able to:

1- (a) Understand the role of the central processing unit (CPU) in a computer

Notes and guidance

The CPU processes instructions and data that are input into the computer so that the result can be output

(b) Understand what is meant by a microprocessor

Notes and guidance

A microprocessor is a type of integrated circuit on a single chip

2- (a) Understand the purpose of the components in a CPU, in a computer that has a Von Neumann architecture

- units: arithmetic logic unit (ALU) and control unit (CU)
- registers: program counter (PC), memory address register (MAR), memory data register (MDR), current instruction register (CIR) and accumulator (ACC)
- buses: address bus, data bus and control bus

(b) Describe the process of the fetch–decode–execute cycle including the role of each component in the process

- How instructions and data are fetched from random access memory (RAM) into the CPU, how they are processed using each component and how they are then executed
- Storing data and addresses into specific registers
- Using buses to transmit data, addresses and signals
- Using units to fetch, decode and execute data and instructions

3- Understand what is meant by a core, cache and clock in a CPU and explain how they can affect the performance of a CPU

- The number of cores, size of the cache and speed of the clock can affect the performance of a CPU

4- Understand the purpose and use of an instruction set for a CPU

- An instruction set is a list of all the commands that can be processed by a CPU and the commands are machine code

5- Describe the purpose and characteristics of an embedded system and identify devices in which they are commonly used

- An embedded system is used to perform a dedicated function, e.g. domestic appliances, cars, security systems, lighting systems or vending machines. This is different to a general purpose computer that is used to perform many different functions, e.g. a personal computer (PC) or a laptop

The purpose of the CPU

The CPU is the most important hardware component in a computer. It has two main functions:

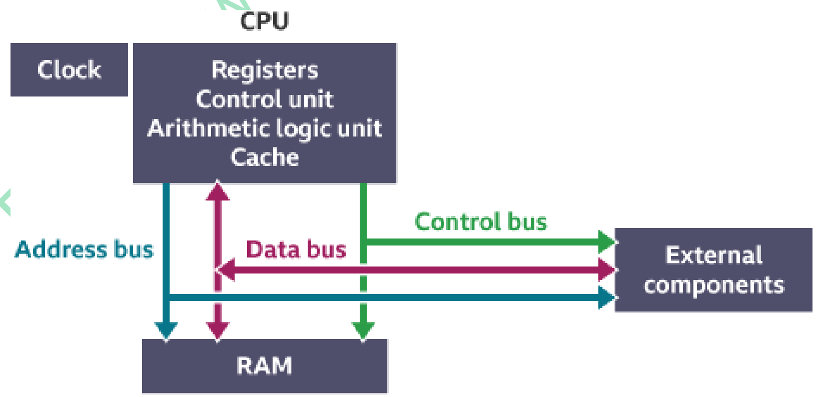
- to process data and instructions
- to control the rest of the computer system
- CPU is very often installed as an integrated circuit on a single microchip.
- The CPU has the responsibility for the execution or processing of all the instructions and data in a computer application.

All programs and data processing are run in the CPU and all hardware components are, to some extent, controlled by it.

Common CPU components

The central processing unit (CPU) consists of a number of components:

- control unit (CU)
- arithmetic logic unit (ALU)
- registers
- cache
- buses
- clock

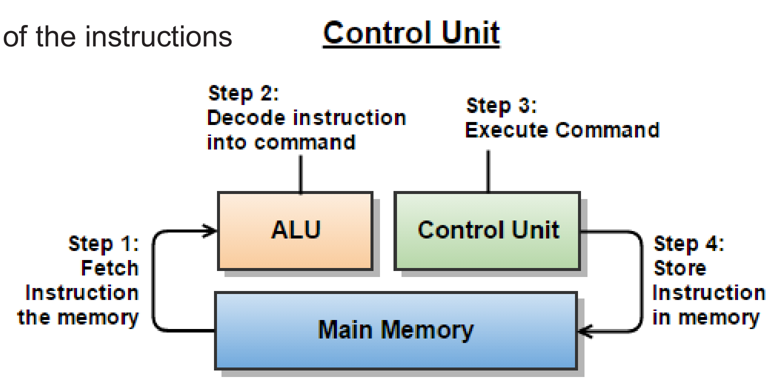


All components work together to allow processing and system control.

Control unit (CU)

The control unit provides several functions

- it fetches, decodes and manages the execution of the instructions
- issues control signals that control hardware
- moves data around the system



Arithmetic logic unit

The ALU performs arithmetic and logical operations or decisions. It is where calculations are done and where decisions are made based on logic.

Registers

Registers are small amounts of high speed memory contained within the CPU. They are used by the processor to store small amounts of data that are needed during processing, such as:

- the address of the next instruction to be executed
- the current instruction being decoded
- the results of calculations

Different processors have different numbers of registers for different purposes, but most have some, or all, of the following: We will study functionality of each register in Von Neumann Model.

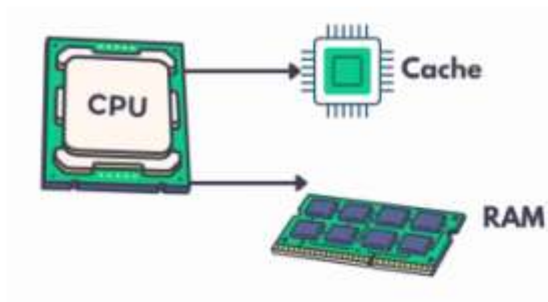
- program counter
- memory address register (MAR)
- memory data register (MDR)
- current instruction register (CIR)
- accumulator (ACC)

CPU REGISTERS	
REGISTER	SYMBOL
Program Counter	PC
Instruction Register	IR
Accumulator Register	AC
Memory Buffer Register	MBR
Memory Address Register	MAR

Cache

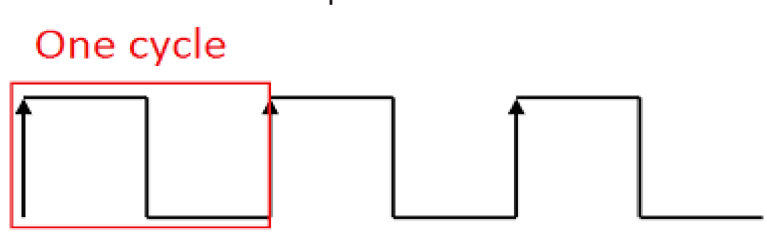
Cache is a small amount of high speed memory built directly within the processor. It is used to temporarily hold data and instructions that the processor is likely to reuse.

This allows for faster processing as the processor does not have to wait for the data and instructions to be fetched from the random-access memory (RAM).



Clock

The CPU contains a clock which is used to coordinate all of the computer's components.
The clock sends out a regular electrical pulse which synchronises - keeps in time - all the components.
The frequency of the pulses is known as the **clock speed.**



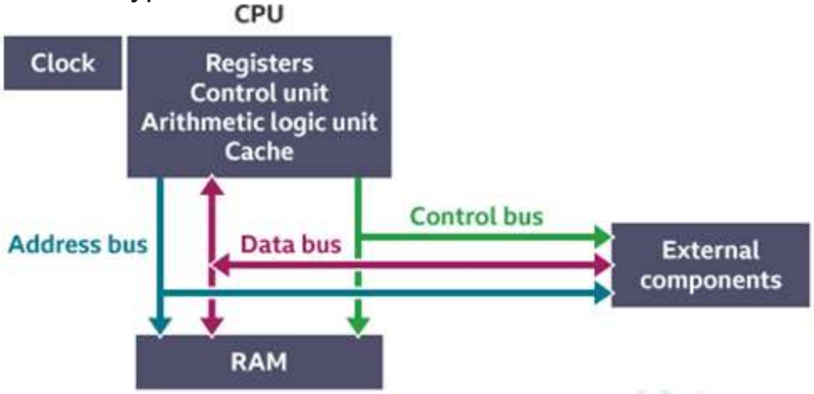
Clock speed is measured in hertz. The higher the clock speed, the greater the number of instructions performed in any given moment of time.

In the 1980s, processors commonly ran at a rate of between 3 megahertz (MHz) to 5 MHz, which is 3 million to 5 million pulses or cycles per second.

Today, processors commonly run at a rate of **3 gigahertz (GHz) to 5 GHz, which is 3 billion pulses or cycles per second**.

Buses

A bus is a high speed internal connection. Buses are used to send control signals and data between the processor and other components.
Three types of bus are used:



- **Address bus** - carries memory addresses from the processor to other components such as RAM and input/output devices.
- **Data bus** - carries the actual data between the processor and other components.
- **Control bus** - carries control signals from the processor to other components.

Von Neumann Model:

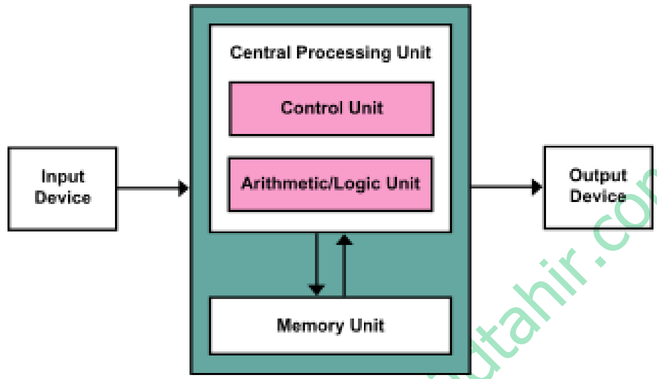
The earliest computers were not "programmable". They were designed to do specific tasks only. **Reprogramming** when it was possible at all was a tedious process, starting with flowcharts and paper notes, followed by detailed engineering designs, and then the often process of physically re-wiring and re-building the machine.

It could take three weeks to set up a program on ENIAC (a computer of 1940s) and get it working. **ENIAC (Electronic Numerical Integrator and Computer) was the first electronic general-purpose computer. It was Turing-complete, digital, and capable of being reprogrammed to solve "a large class of numerical problems**

The **von Neumann architecture**, also known as the **von Neumann model** and **Princeton Architecture**, is based on John **von Neumann's (**mathematician and physicist**)** research paper in **1945.**

Von Neumann architecture

Von Neumann architecture is the design upon which many general purpose computers are based. This architecture uses the stored program concept.The key elements of Von Neumann architecture are:



- data and instructions are both stored as binary digits
- data and instructions are both stored together in the **same RAM**
- instructions are fetched from memory one at a time and in order - serially
- the processor decodes and executes an instruction, before cycling around to fetch the next instruction
- the cycle continues until no more instructions are available

A processor based on Von Neumann architecture has five special registers which it uses for processing:

- program counter - holds the memory address of the next instruction to be fetched from main memory
- memory address register (MAR) - holds the address of the current instruction that is to be fetched from memory, or the address in memory to which data is to be transferred
- memory data register (MDR) - holds the contents found at the address held in the MAR, or data which is to be transferred to main memory
- current instruction register (CIR) - holds the instruction that is currently being decoded and executed
- accumulator (ACC) - holds the results of processing

The registers and key elements of the Von Neumann architecture all play a part in how an instruction is processed in the fetch-decode-execute cycle.
This described design architecture for an electronic digital computer with parts consisting

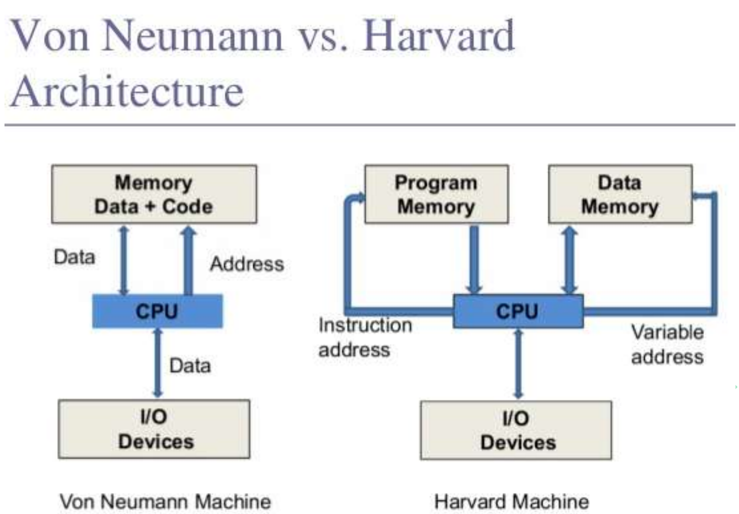
Central Processing Unit containing:
- Control Unit
- Arithmetic/Logic unit
- Processor registers,
- Memory to store data & instructions

Input / Output Mechanism
External Storage

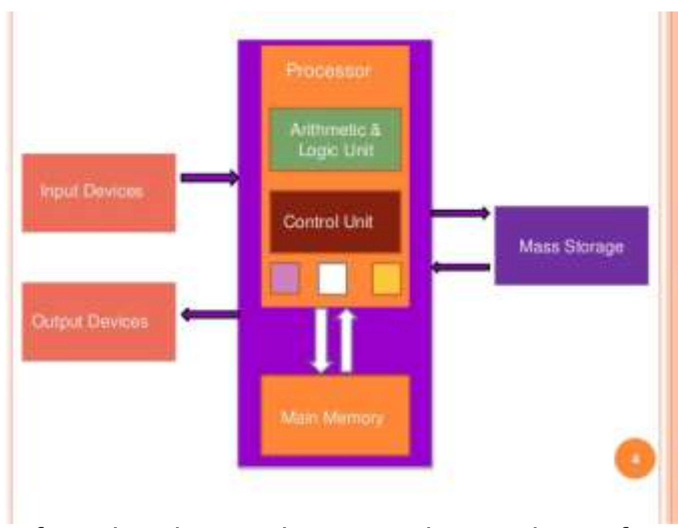
A stored-program digital computer: is one that keeps its programmed instructions, as well as its data, in read-write, random-access memory **(RAM)** So **John Von Neumann** introduced the idea of the **stored program Computer.**

Previously data and programs were stored in separate memories. Von Neumann realized that data and programs are indistinguishable and can, therefore, use the same memory The Von Neumann architecture uses a single processor which follows a linear sequence of fetch-decode-execute.

The Picture below shows difference between Von Neumann architecture and Harvard architecture (earliest computers)



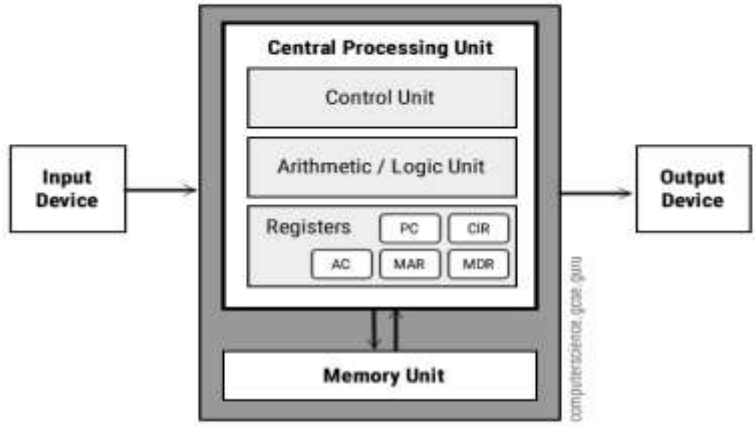
Features of a Von Neumann architecture



The illustration above shows the essential features of the Von Neumann or stored-program architecture.

Memory
The computer will have memory that can hold both data and also the program processing that data. In modern computers this memory is RAM.

Control Unit
The control unit will manage the process of moving data and program into and out of memory and also deal with carrying out (executing) program instructions - one at a time. This includes the idea of a 'register' to hold intermediate values. In the illustration above, the 'accumulator' is one such register.



The 'one-at-a-time' phrase means that the Von Neumann architecture is a **sequential processing machine**.

Input - Output

This architecture allows for the idea that a person needs to interact with the machine. Whatever values that is passed to and forth is stored once again in some internal registers.

Arithmetic Logic Unit

This part of the architecture is solely involved with carrying out calculations upon the data. All the usual Add, Multiply, Divide and Subtract calculations **will be available but also data comparisons such as '**Greater Than', 'Less Than', 'Equal To' **will be available.**

Registers:

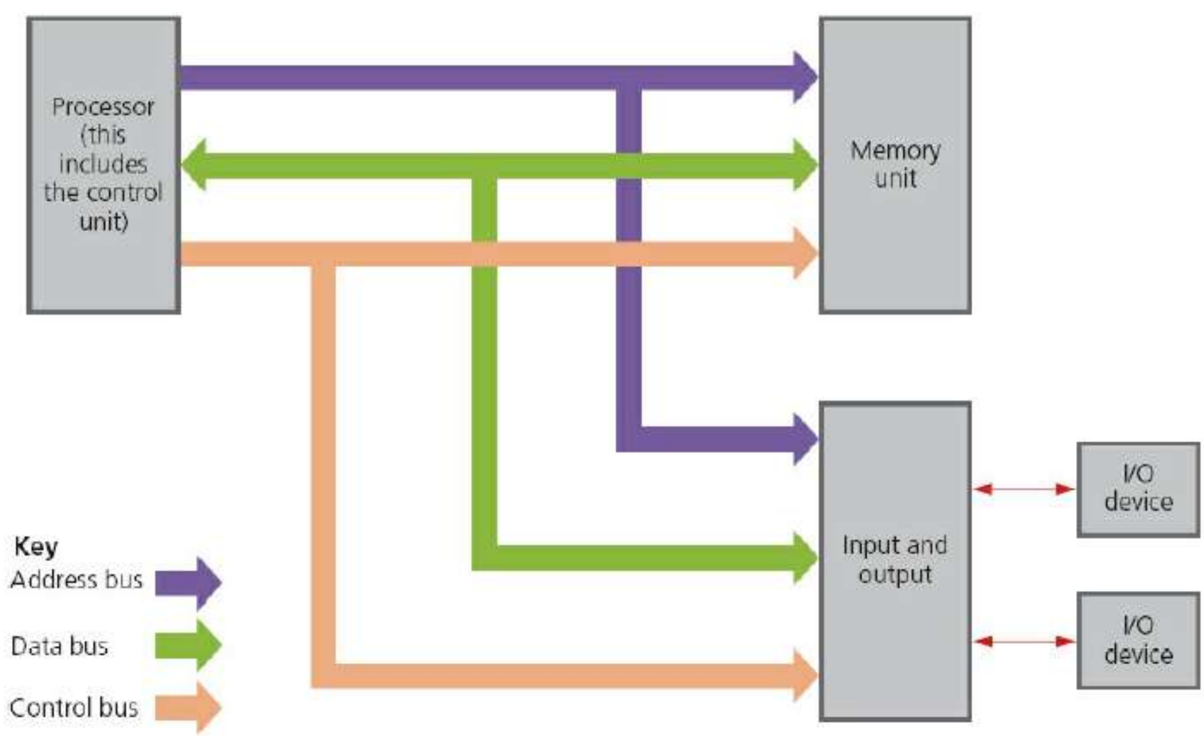
The Von Neumann architecture uses a single processor which follows a linear sequence of **fetch-decode-execute. In order to do this, the processor has to use some special registers,** which are **discrete memory locations with special purposes attached**. These are:

Register	Name/ Function
PC	**Program Counter**- keeps track of where to find the next instruction so that a copy of the instruction can be placed in the current instruction register
MAR	**Memory Address Register**- to hold the memory address that contains either the next piece of data or an instruction that is to be used.
MDR	**Memory Data Register:** acts like a buffer and holds anything that is copied from the memory ready for the processor to use it
CIR	**Current Instruction Register:** The current instruction register holds the instruction that is to be executed
ACC	**Accumulator:** is a special purpose register where data is worked on e.g I wanted to add 4 to 7, for example, I would fetch 4 from RAM and put 4 in the Accumulator. I would then get 7 from RAM and add that to whatever was in the Accumulator. I would then store the result briefly in the Accumulator before moving it back to somewhere in RAM to be used later. All calculations of any description are done using the Accumulator.

Bus

Notice the arrows between components? This implies that information should flow between various parts of the computer. In a modern computer built to the Von Neumann architecture, information passes back and forth along a 'bus'. There are buses to identify locations in memory - an 'address bus'

Type of bus	Description of bus	Data/signal direction
address bus	carries signals relating to addresses (see later) between the processor and the memory	unidirectional (signals travel in one direction only)
data bus	sends data between the processor, the memory unit and the input/output devices	bi-directional (data can travel in both directions)
control bus	carries signals relating to the control and coordination of all activities within the computer (examples include: the read and write functions)	unidirectional (signals travel in one direction only)

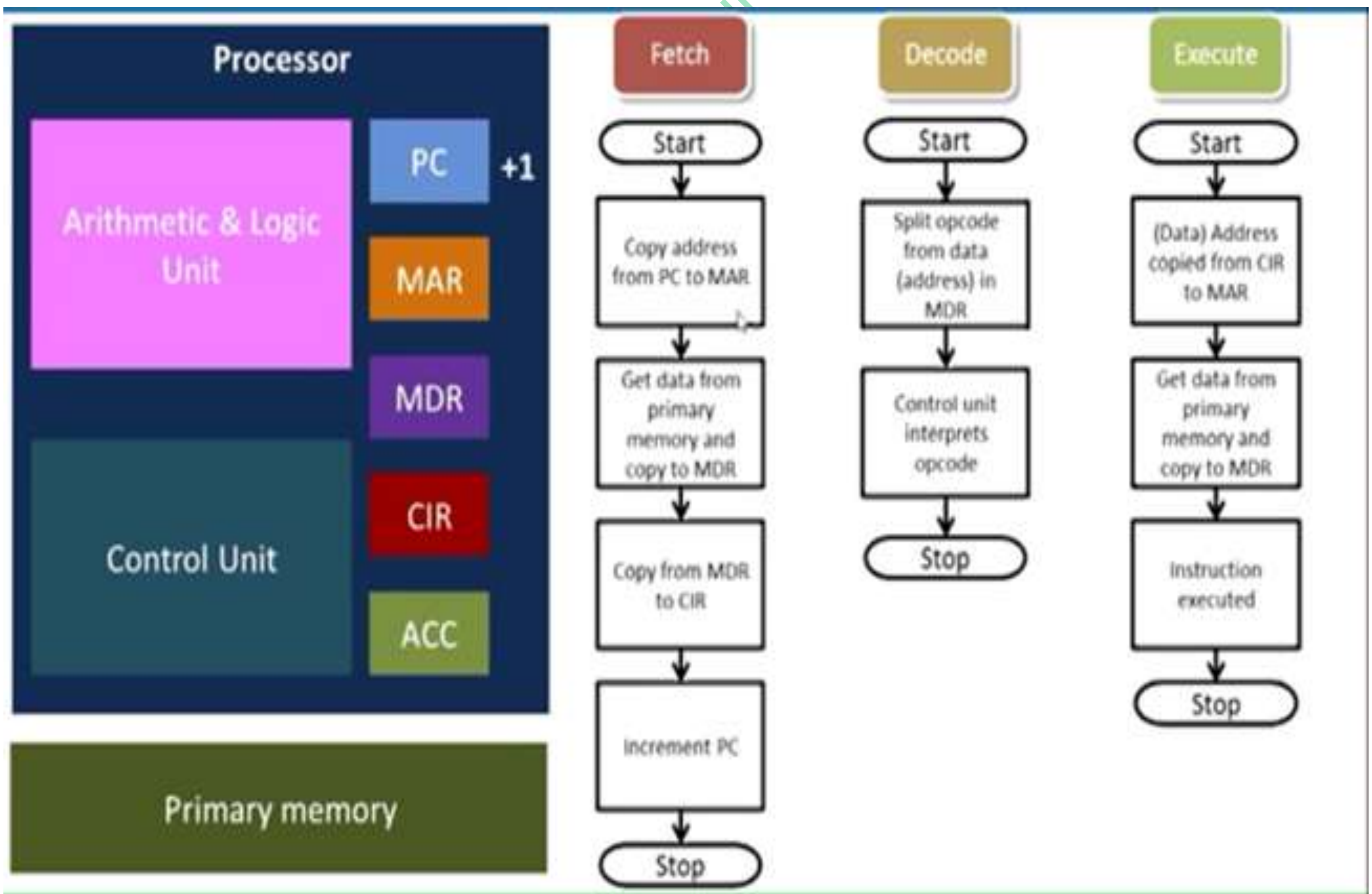


CPU and Fetch-Execute Cycle

The Fetch-Decode-Execute-Reset Cycle:

The following is an algorithm that shows the steps in the cycle. At the end the cycle is reset and the algorithm repeated.

1. **Load the address that is in the program counter (PC) into the memory address register (MAR).**
2. **Load the instruction that is in the memory address given by the MAR into the memory data register (MDR).**
3. **Load the instruction that is now in the MDR into the current instruction register (CIR).**
4. **Increment the PC by 1.**
5. **Decode the instruction that is in the CIR.**
6. **If the instruction is a jump instruction then**
    a. **Load the address part of the instruction into the PC**
    b. **Reset by going to step 1.**
7. **Execute the instruction.**
8. **Reset by going to step 1**



Factors affecting CPU performance:

Even though today's processors are tremendously fast, their performance can be affected by a number of factors:

- clock speed
- cache size
- number of cores

Clock speed:

Clock speed is the number of pulses the central processing unit's (CPU)'s clock generates per second. It is measured in hertz.
CPU clocks can sometimes be sped up slightly by the user. This process is known as overclocking. The more pulses per second, the more fetch-decode-execute cycles that can be performed and the greater the number of instructions that are processed in a given space of time. Overclocking can cause long-term damage to the CPU as it is working harder and producing more heat.

Cache size:

Cache is a small amount of high-speed random access memory (RAM) within the processor. It is used to temporarily hold data and instructions that the processor is likely to reuse. This allows for faster processing as the processor does not have to wait for the data and instructions to be fetched from RAM.
The bigger its cache, the less time a processor has to wait for instructions to be fetched.

Number of cores:

A processing unit within a CPU is known as a core. Each core is capable of fetching, decoding and executing its own instructions.
The more cores a CPU has, the greater the number of instructions it can process in a given space of time.

Embedded systems

As well as general purpose computers, there are other types of computer systems. The most common of these are known as embedded systems.
An embedded system is a small computer that forms part of a larger system, device or machine. It includes both hardware and software and its purpose is to control the device and to allow a user to interact with it. They tend to have one, or a limited number of tasks that they can perform.

Examples of embedded systems include:

- central heating systems
- engine management systems in vehicles
- domestic appliances, such as dishwashers and TVs

- digital watches
- electronic calculators
- GPS systems
- fitness trackers

Depending on the embedded system's purpose, they use input devices like sensors and switches to produce output, for example sounding warning buzzers and switching lights on.

Embedded devices are not usually programmable by a user - the programming is usually done beforehand by the manufacturer. However, it is often possible to upgrade the software on an embedded device. For example, fitness trackers are embedded systems, but the software can often be upgraded by connecting the device to a PC and installing the new software.

Embedded systems can have advantages over general purpose computers in that:

- Their limited number of functions means they are cheaper to design and build.
- They tend to require less power. Some devices run from batteries.
- They do not need much processing power. They can be built using cheaper, less powerful processors.

Past paper Question June 2015

7 (a) One of the key features of von Neumann computer architecture is the use of buses.
Three buses and three descriptions are shown below.

Draw a line to connect each bus to its correct description.

Bus	Description
address bus	this bus carries signals used to coordinate the computer's activities
control bus	this bi-directional bus is used to exchange data between processor, memory and input/ output devices
data bus	this uni-directional bus carries signals relating to memory addresses between processor and memory

[2]

(b) The seven stages in a von Neumann fetch-execute cycle are shown in the table below.
Put each stage in the correct sequence by writing the numbers 1 to 7 in the right hand column.

The first one has been done for you.

Stage	Sequence number
the instruction is then copied from the memory location contained in the MAR (memory address register) and is placed in the MDR (memory data register)	
the instruction is finally decoded and is then executed	
the PC (program counter) contains the address of the next instruction to be fetched	1
the entire instruction is then copied from the MDR (memory data register) and placed in the CIR (current instruction register)	
the address contained in the PC (program counter) is copied to the MAR (memory address register) via the address bus	
the address part of the instruction, if any, is placed in the MAR (memory address register)	
the value in the PC (program counter) is then incremented so that it points to the next instruction to be fetched	

[6]

Answers

(b)

description of stage	sequence number
the instruction is then copied from the memory location contained in the MAR (memory address register) and is placed in the MDR (memory data register)	3
the instruction is finally decoded and is then executed	7
the PC (program counter) contains the address of the next instruction to be fetched	*(1)*
the entire instruction is then copied from the MDR (memory data register) and placed in the CIR (current instruction register)	4
the address contained in the PC (program counter) is copied to the MAR (memory address register) via the address bus	2
the address part of the instruction is placed in the MAR (memory address register)	6
the value in the PC (program counter) is then incremented so that it points to the next instruction to be fetched	5*

The incrementation of the program counter can appear at any stage after 2. All other stages must be in the correct given order. [6]

Refrences:

https://www.bbc.co.uk/bitesize/guides/zkrr97h/revision/8
www.wikipedia.com
VCN – ICT Department 2013 Prepared by *Davis Rwatooro T.*