
1.1 Systems Architecture

1.1.1 Architecture of the CPU

This unit examines the most important part of a computer system, the CPU. It examines in more detail how modern CPUs actually follow instructions and how the performance of the CPU can be improved by tweaking the internal design of the actual CPU.

It also looks at how the CPU is internally organised and how different instructions move data around the CPU in different ways.

Embedded systems are an important use of CPUs that we often recognise as "smart" but not necessarily a computer system. This unit includes studying some of those embedded systems.

The purpose of the Central Processing Unit

The CPU (Central Processing Unit) is the part of a computer system that is commonly referred to as the "brains" of a computer. The CPU is also known as the processor or microprocessor.

The CPU is responsible for executing a sequence of stored instructions called a program. This program will take inputs from an input device, process the input in some way and output the results to an output device.

CPUs aren't only found in desktop or laptop computers, many electronic devices now rely on them for their operation. Mobile phones, DVD players and washing machines are examples of equipment that have a CPU.

The Function of the CPU

The function of the Central Processing Unit is to fetch, decode, and execute instructions.

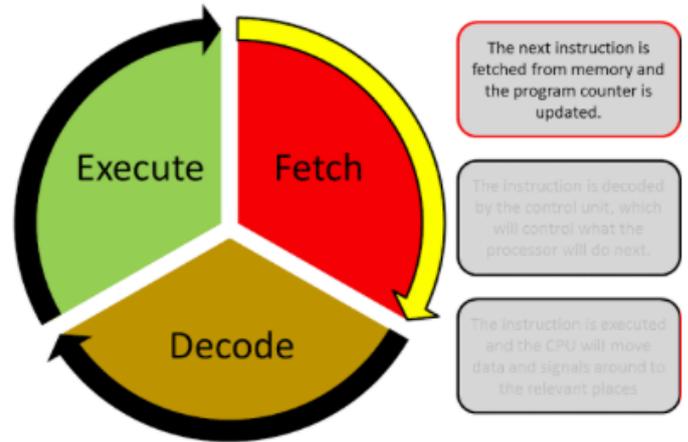
The program code instructions are stored in the systems RAM. They are **fetch**ed, one at a time, and retrieved along the data bus on the computer. The Control Unit **decodes** the instruction and sends data and signals around the CPU that carry out (or **execute**) the instruction.

This cycle repeats constantly. Even when the computer system seems to be doing nothing, it is actually running through this cycle, checking the keyboard, refreshing the display etc.

The Fetch Decode Execute Cycle Simplified

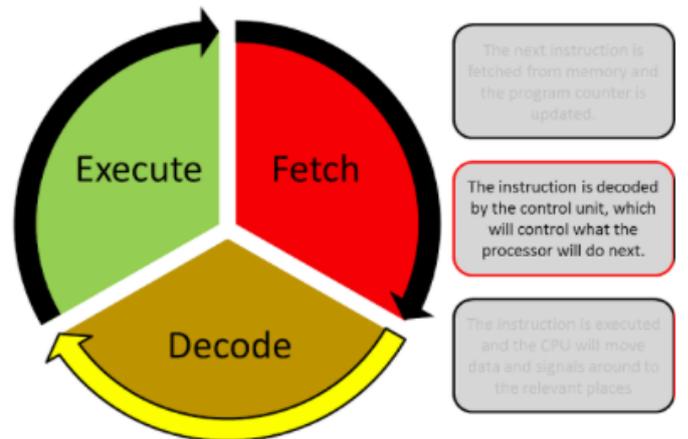
The Fetch Cycle

1. The PC value is copied to the MAR
2. PC is incremented
3. The Instruction at the MAR address is read from RAM



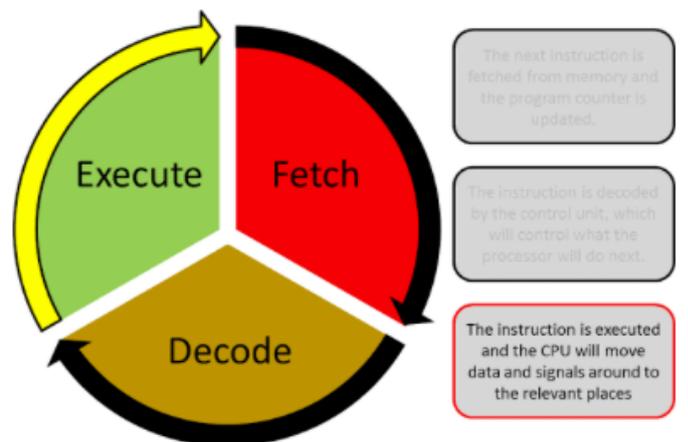
The Decode Cycle

1. The instruction in the MAR is decoded by the CU



The Execute Cycle

1. The data at the MAR address is read from RAM
2. The ALU performs any calculations required on the ACC
3. The value in the MDR is written to the address in the MAR.



John Von Neumann

John Von Neumann was a mathematician and physicist who studied how to design a computer and came up with a system that is used almost universally in modern computer systems. His designs include the concept of having a program that is stored and is executed when the computer runs. The program is stored in memory (often loaded from a hard drive), and the instructions are fetched, decoded and executed, and a single control unit controls the flow of program instructions and data around the system.

You will need to know how the Von Neumann architecture works in a CPU, and the names of the main components that it uses.

Instructions are firstly fetched from memory, and then decoded in the control unit. Once they have been decoded, they will be executed. You should memorise the phrase "The purpose of the CPU is to fetch, decode and execute instructions" as this question will appear frequently on exam papers.

You need to know about the following CPU components:

- ALU (Arithmetic Logic Unit)
- CU (Control Unit)
- Cache
- Registers

And you also need to know these specific Von Neumann architecture components:

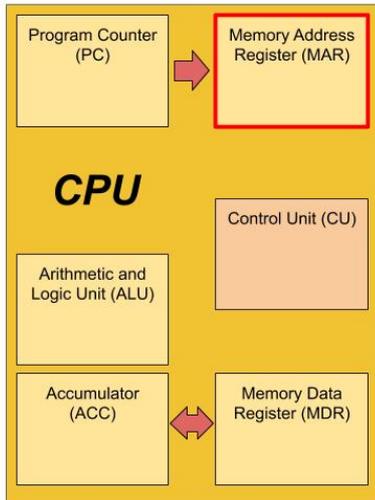
- MAR (Memory Address Register)
- MDR (Memory Data Register)
- Program Counter
- Accumulator



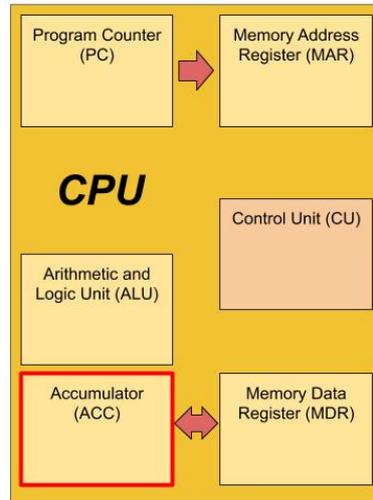
John Von Neumann (1903-1957) was an American mathematician and physicist who came up with the concept of self replicating organisms (a precursor to the discovery of DNA), worked on the design of the first nuclear bomb, and drew up the blueprint for the Von Neumann architecture which is used on almost all modern computer systems.

Common CPU Components and their Function

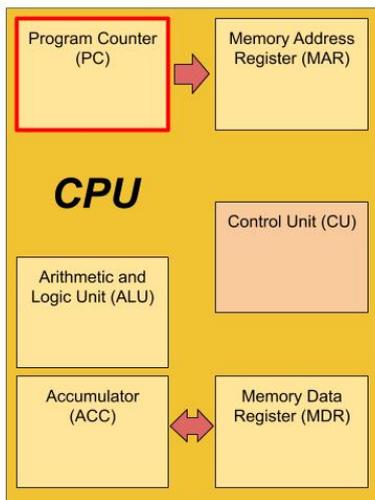
The Von Neumann architecture divides the internal parts of the CPU into specific sections. For your exam, you will need to be able to label a diagram and explain what each section does. The layout of the inside of a CPU diagram will vary depending where you take it from. Below is one possibility.



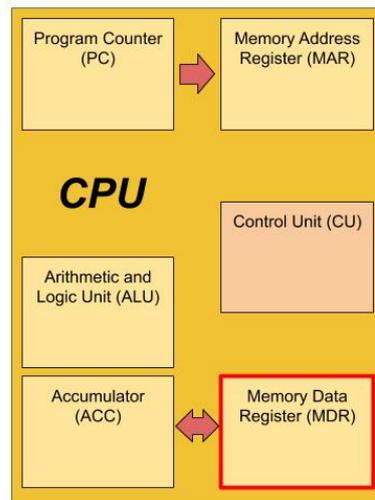
Memory Address Register
The memory address register (MAR) holds the address of where data or instructions will be fetched from or stored to.



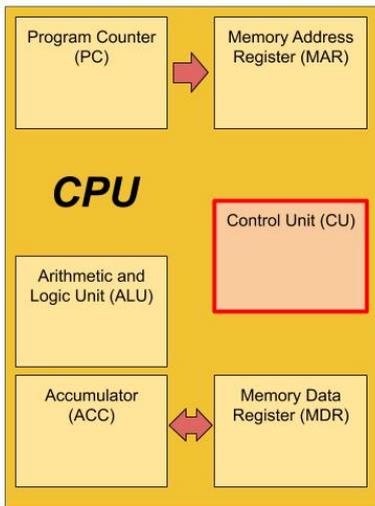
Accumulator
The accumulator holds the results of calculations and logic operations that the ALU has performed.



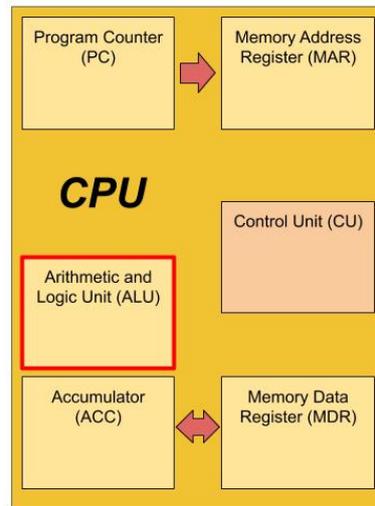
Program Counter
The program counter (PC) holds the address of the next instruction that will be fetched.



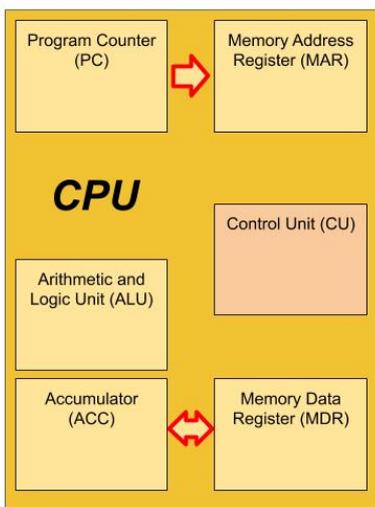
Memory Data Register
The MDR holds the data fetched from, or to be written to, the external memory.



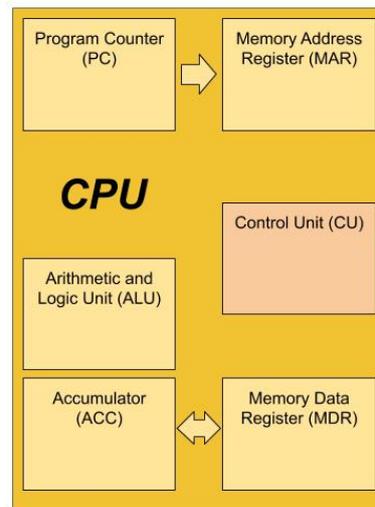
Control Unit
The control unit (CU) decodes instructions and sends signals to control how data moves around the CPU.



Arithmetic and Logic Unit
The arithmetic and logic unit (ALU) performs calculations and logic operations such as addition, subtraction, AND, OR etc.



Internal Buses
Data, instructions and control signals are moved around the inside of the CPU along internal buses.



Cache
This is some super fast access memory that can store frequently accessed instructions and data, without having to access the main computer memory. Cache memory makes the computer a lot faster!

The actual layout of the inside of a CPU can be presented in different ways (look at the Craig & Dave one) but they are showing the same thing. You need to be aware of what is what, so that you will not be confused despite slight differences in the way it is presented.

How the CPU Components perform the Fetch Decode Execute cycle.

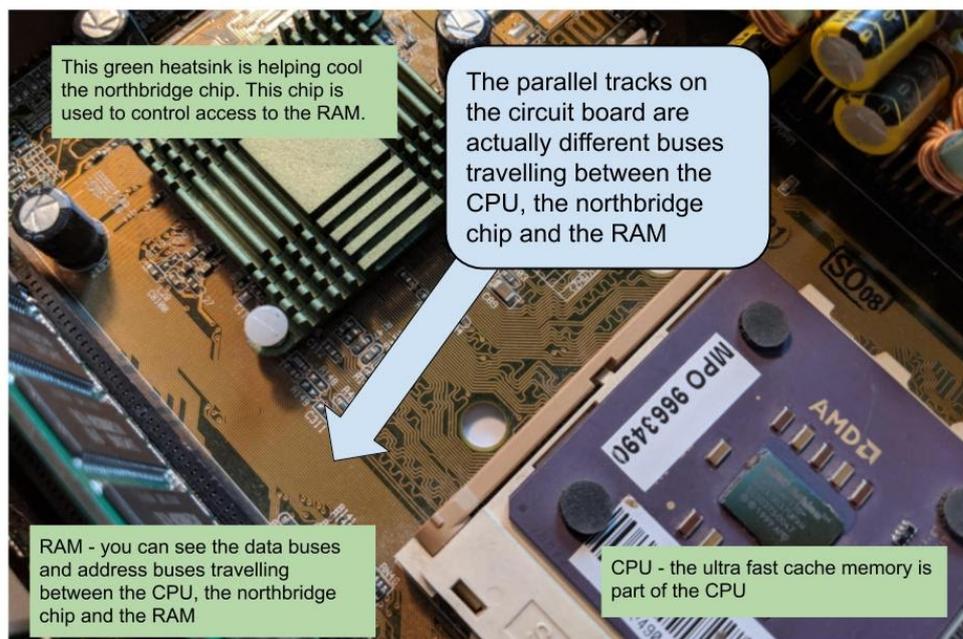
Appendix 1 of this chapter contains a very detailed description of the Fetch-Decode-Execute cycle. It is beyond the scope of GCSE to know all the details—unless you're aiming for the very top grades.

The Factors Affecting the Performance of a CPU

Cache

Cache memory is one of the key components of the CPU. Inside the CPU chip is an area of super fast memory not unlike RAM.

The Cache memory can rapidly load data and code from the RAM and deliver them along the internal buses at a much faster rate compared with the external address bus and data bus. Using a CPU with a larger amount of RAM will make the whole computer system run significantly more quickly.



In this picture you can actually see the bus lines on the motherboard which run between the CPU, the RAM and other sections on the system.

To improve the performance of a computer system, the CPU could be swapped for one that has more cache memory. This would make the computer faster because the cache memory can be accessed much faster than primary storage (RAM)

1. A CPU running at a faster clock rate will work more quickly than a slower one. CPU chips are designed to work at a certain rate and if they run too quickly they can be damaged, but most CPUs are capable of some overclocking.
2. If a CPU is swapped for a similar one with more cores, then the system will usually work much more quickly. Doubling the number of cores could potentially double

the speed of the computer, but the software must have been specially written to take advantage of multi-core processors.

3. CPUs with a large amount of cache will run the same code considerably more quickly than CPUs with no or little cache. This is because they can pre-load the code and data into the faster cache memory and re-use it rather than loading it from the slower RAM.

You will need to know these three factors for your exam, and you should also be able to explain the reason why improve the speed of a computer system.

The Clock

The computer system will continue to run through the FDE cycle all the time that the power is turned on. The CPU is very complex and there are often several things happening at once. To coordinate all of these different actions, an electronic clock makes everything run at the same time and in sync. The clock speed is measure in hertz. One hertz is one cycle per second - but the computer runs at gigahertz speeds - this is one billion cycles per second. It is difficult to image such fast speeds, in one billionth of a second, light only travels about 30 cm.

It is possible to change the clock speed, but this can only be done by a small amount. Increasing the clock speed is known as overclocking and most CPUs allow you to increase the clock speed by 10-20%. If you try to overclock the processor any more than that it may not work reliably, may run at an extreme temperature and will almost certainly void the manufacturers warranty. Temperature is the enemy of performance computers and they usually have a large and expensive cooling system to remove the heat as quickly as possible. The CPU will get very hot and may melt if the heatsink is removed.

To improve the performance of the CPU, the clock rate could be increased. This would make the whole fetch, decode, execute cycle faster. It is usually possible to use the same CPU when the clock rate is increased slightly, but the processor will generate more heat and may be unreliable if the clock rate is increased too much,

Number of Cores

Because computer systems are working at the very limits of what we understand to be the physical properties of computers, we need to look at other ways of making improvements in computer speed. The circuits inside the CPU are arranged on a grid

that is often just 14 nanometres across. Human fingernails grow about one nanometre a second, so 14 nanometres is tiny. Electrical signals behave in unusual ways at such a small scale, they cannot go around sharp corners and jump across the gaps which makes their behaviour erratic and unreliable.

The answer seems obvious! Instead of trying to make one CPU run through the fetch, decode, execute cycle at a faster and faster rate, we could just use two processors. This has recently led to some rapid development and a big boost in CPU performance. Reasonably priced CPU chips now have four or eight separate processors inside them, and are known as quad cored (4 CPUs) and octo-core (8 CPUs). High end servers take this even further and massively powerful super computers may have many hundreds of thousands of processors.

In 2018, the top ranked mobile phones use 8 core CPUs and some also have a graphics processor which may also be octo-core.

It is important to realise that using a CPU with more cores does not automatically make a computer run faster. The operating system and application software will need to be specially written to take advantage of the multi-core processor and not all tasks can be broken down into parallel tasks.

To improve the performance of the computer system, the CPU would be swapped for one that has more cores. This would mean that the computer system can carry out instructions at the same time, but only if the program had been written to make use of the extra cores. Modern operating systems can use many cores at once because they are written to make the maximum use of the CPUs features.

Embedded Systems

We often think of computer systems as the PCs, laptops, tablet computers and mobile phones which are all around us. There are actually many more computers than that! Embedded computer systems are built into larger appliances or devices such as washing machines, dishwashers and microwaves. These devices are not primarily intended to process data but perform another task instead.

The **purpose** of the embedded computer system is to monitor and control various components in the host device, for example, a washing machine is primarily intended to wash clothes, the embedded computer is there to help it keep track of washing programs, timings, water temperature and so on.

Many household devices are **examples** of embedded systems, but this does not mean that they all are. If it can be programmed or controlled with buttons that have a display then there is a good chance that a computer will be behind it and the device is an embedded system.

Embedded systems have software that controls the hardware connected to the system. This is a special kind of software known as firmware. The firmware can be updated, but this does not happen very often. Firmware is usually very basic but also very reliable.

These are embedded systems:	These are not embedded systems:
Washing machine	Toaster
Microwave	Conventional oven
Burglar alarm	Vacuum cleaner
Smart TV	Electric drill

In the above examples, all of the systems on the left are unlikely to receive a firmware update. The devices on the right do not have firmware because they do not have embedded systems.

This work is an adaptation by Mr S Fisher of an online resource by Mr P Waud (<https://sites.google.com/horburycomputing.co.uk/gcse-computing/>)

Original Work Copyright © 2017-2020 Mr P Waud
This adaptation Copyright © 2020 Mr S Fisher

This work is released under a Creative Commons Share and Share Alike Licence