1. Introduction

Computers today are all around us in one form or another. You are probably sitting in your classroom reading this text. Just have a think about all of the different computers that are nearby:

- Mobile phone
- Calculator
- Watch
- Interactive white board
- Projector
- Printer
- Desktop computer
- Laptop

As you can see, computers come in all shapes and sizes and are often built into other devices. However, all computers share some common features which we will cover in the next few pages.

What is a computer system

2. A computer system

A computer system takes a set of digital inputs, processes them and creates a set of outputs. This is done by a combination of hardware and software.
In other words, you put some data into a computer (**input**), something happens to that data (**processing**) and then the computer puts out a result (**output**).
In order to do this, the computer needs instructions to tell it what to do. The instructions are provided by a **program**

The diagram below shows you the idea of a computer system in its most basic form

A laptop or desktop computer are examples of computer systems.

Smart phones would also be another example as they have input, output and secondary storage devices.

What is a computer system

3. Computer input

As mentioned on the previous page, a computer system needs an input, in the form of data. Somehow the data has to get into the computer and this is done by using an **'input device.'**

For example, if you want to create an email, you might use a keyboard to type your message. As you press the key you will see the text appear on the screen. The keyboard is an 'input device' as it enables you to put data into the computer system.

KEY TERM: An input device is an example of **hardware**. Hardware is the word used to describe the physical parts of a computer system.

Examples of input devices

Input device	Task	The input data
Keyboard	Make a new paragraph in your document	You press the 'return' key, then the keyboard sends the 'return' digital code as input to the CPU. After processing, a paragraph break is added to the document.
Mouse	Save a document	You click on the save icon using your mouse. The input data from the mouse is sent to the microprocessor which causes the document to be saved.
Touch screen	Open an app on your phone.	The touch screen picks up your finger movements on the screen and produces input data for the computer system to act upon.
Microphone	Record your voice	The microphone system picks up sound and converts it into digital input data (via the sound card or chip).
Heat sensor	Keeps the house at a stable temperature.	The heat sensor detects the surrounding temperature and produces an input into the computer system.

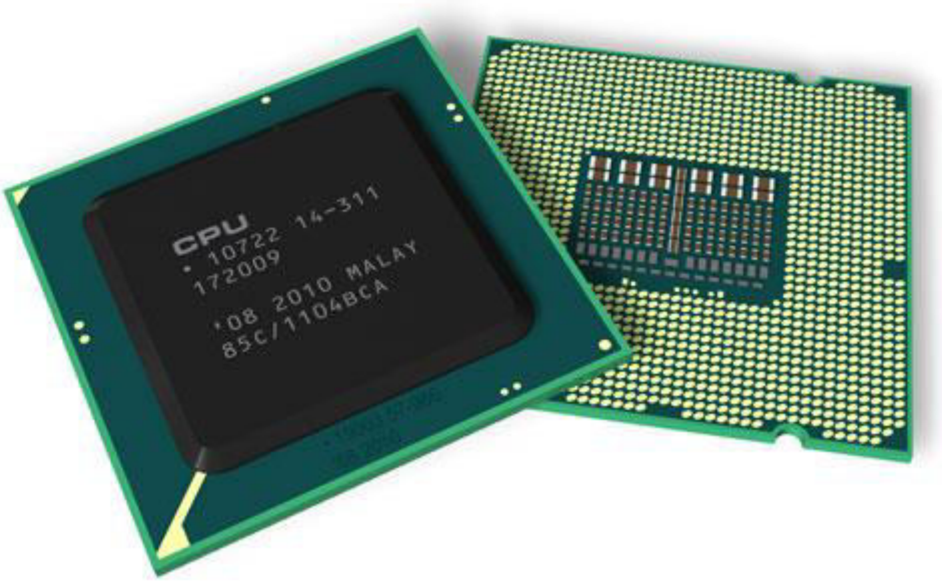
What is a computer system

4. Computer processing

The main purpose of a computer is to process data.

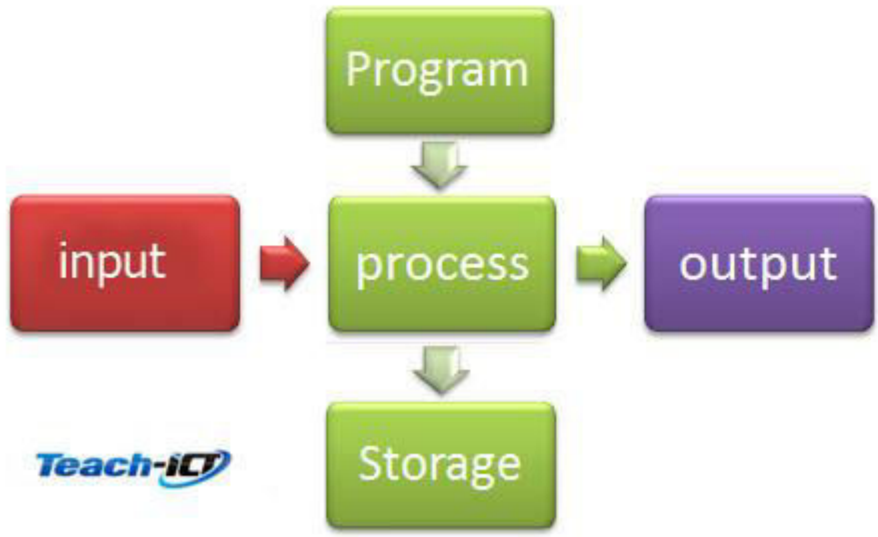
Once data has been input into the computer system something needs to happen to it i.e. it needs to be **processed**.
If you use your keyboard to type the letter A you would expect it to appear on your computer monitor. However, that doesn't magically happen, there is an in-between stage which takes the data you have input, it does some work on it (**process**) and then shows it on your computer screen as the letter A (output).
The processing stage takes place within the **CPU** (**Central Processing Unit**).
The CPU is a single electronic chip within the system.
Another term often used for the CPU is a **microprocessor**.

The CPU (or microprocessor) is the 'brain' of the computer. It takes input, processes it and produces output data.



What is a computer system

5. Software programs and processing



The CPU needs instructions to tell it what to do, usually many, many of them. These instructions are collected together to form a **computer program**. A typical application computer program will have thousands of instructions included.

For example with the letter 'A' data that you input from the keyboard, there needs to be an application program running to handle the input - maybe you are typing with an email client open, or any other application needing a keyboard input.

KEY TERM: **Software** is a **computer program** that contains the set of instructions needed by the CPU in order to perform a specific task.

There are two main types of software.

System software

These are programs that manage the operation of the computer. System software boots up the computer when you switch it on. System software tells the CPU how to communicate with other parts of the system such as the hard disk or DVD drive. Another task of system software is to load up the relevant application software when called for.

An 'operating system' such as Windows or Linux is an example of system software.

Application software

These are computer programs written for a specific purpose. For example, a word processing application, a music player or photo editor. There are thousands of application programs available.

What is a computer system

6. Computer output

The 'output' is what comes out of the computer after the processing stage is complete.

Let's go back to the email that you wanted to write. You typed the letter A using a keyboard (input) the CPU processed the data using a software program (processing). Now the next thing to happen is for that data to be given to you in a format that you can make use of (output).

For this to happen an **output device** will be used. In the case of your email the output device is probably a computer monitor. You type the letter 'A' and an 'A' appears on your screen inside the email application. But think of all the processing that has taken place between you hitting that key and it appearing on your screen.

There are many different output devices that can be used, some are listed in the table below.

Output device examples

Output device	**How it works**
Computer monitor	Presents output as an image on the screen. Note that 'image' is not just a photo - an image is literally what you see on a monitor that

Output device	How it works
	includes text etc
Printer	Prints out documents
Speakers	Outputs sound e.g. music
LED light	Lights up when an output is sent to it

Like input devices (page 3), output devices are also examples of computer **hardware**. This is because they are physical objects that you can see, touch and often pick up.

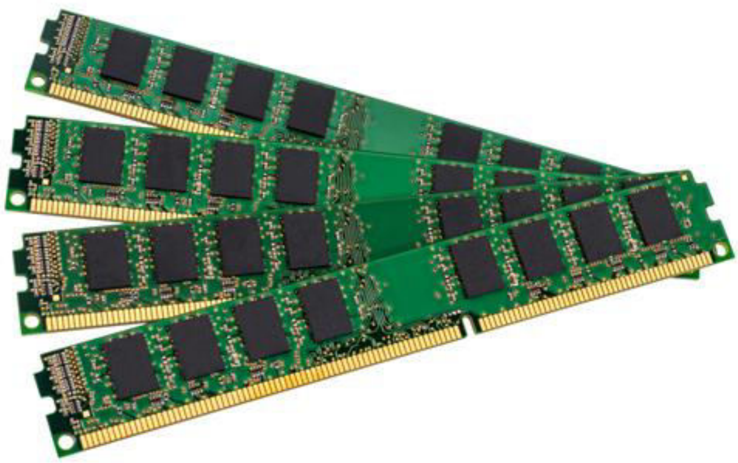
NOTE: in the exam if you are given the example of a computer system, for example a smart phone, make sure that the output device is actually part of that device, so display screen or inbuilt speaker would be fine but headphones would not as they are are an accessory.

7. Storage

On the previous page we explained how the processed data is usually sent to an output device. But this is not always the case - data does not always need to be output immediately, there are times when you want to store it.

The application you are running might allow you to store the data as a file. For example a data logger would store the incoming data.

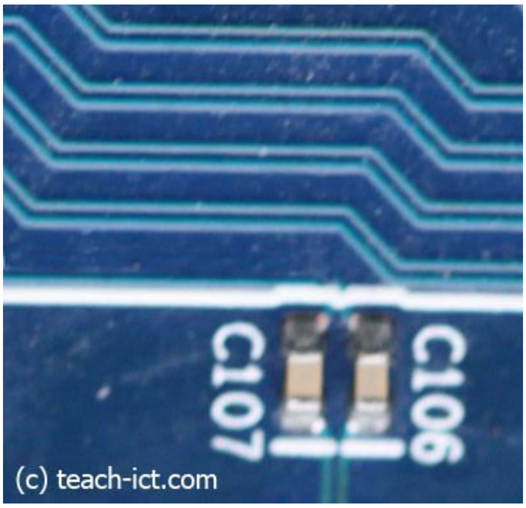
With a desktop computer data could be stored either in RAM or on the hard disk. A tablet or smart phone would use a secondary storage device such as solid state memory.



What is a computer system

8. Bus

So far we have talked about how data is input to a computer, how is it processed and then how it is either output or stored. But the next question is how does the data move around the system?



The answer is on a 'bus'.

A **bus** is a set of wires or tracks laid down on a printed circuit board. A bus is also exactly the same inside the CPU chip itself only much tinier.

In the photograph you can see an 8 bit bus.
The bus is the eight parallel tracks you see. Each track carries a single data bit.

Using the same logic, inside a 32 bit computer, the data bus will have 32 tracks or wires.

There are three main buses which are part of the computer system.

Data bus

This carries the actual data around the system. Each wire or track carries a single data bit.

Address bus

This carries information on where the data needs to go or come from i.e. the data address.

Control bus

This carries signals around the system to control various components such as the DVD drive, hard disk or graphics card.

What is a computer system

9. Summary

- A computer system is one that is able to take a set of inputs, process them and create a set of outputs. This is done by using a combination of hardware and software.
- An input provides data to the computer.
- Processing involves manipulating the input data.
- Output provides the processed data in a format that the user can understand and use.
- The physical parts of a computer are called hardware.
- The computer programs which provide the system with instructions are called software.
- A bus is a set of wires which carry data around the computer system.

1. Purpose of a CPU

The CPU is often described as the 'brain' of the computer (although that is not a technical term so do not write it in an exam). The CPU is an electronic chip called a microprocessor.

The CPU fetches instructions from memory (RAM), it decodes the instructions and then executes those instructions. The instructions are provided by a computer program.
The purpose of the CPU is to process data. It is where all the searching, sorting, calculating and decision making takes place in the computer.

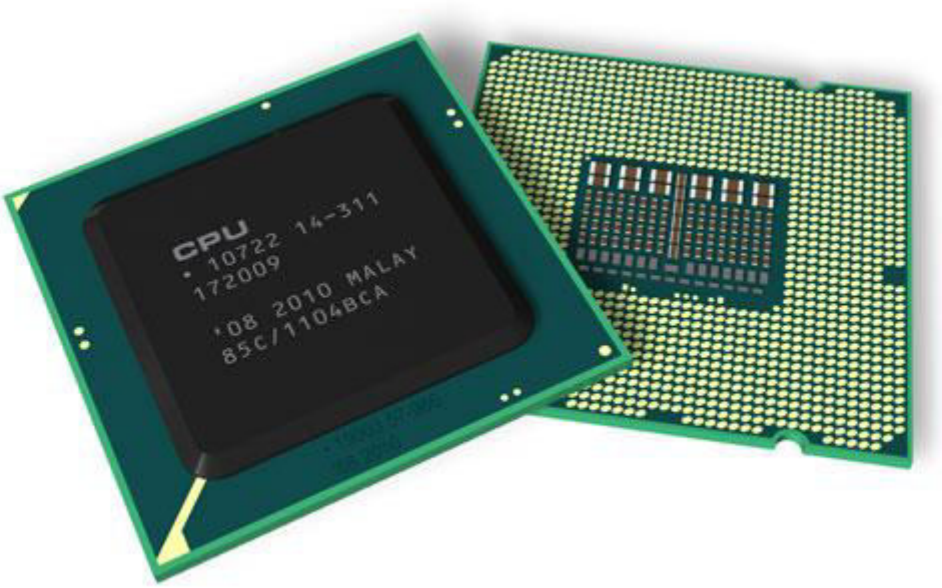
So no matter what you do on your computer, whether it be writing an essay, looking at your photographs or checking your emails, the CPU has dealt with all of the data and processing required to make these tasks possible.

Another task for the CPU is to control all the other component parts of the computer for example:

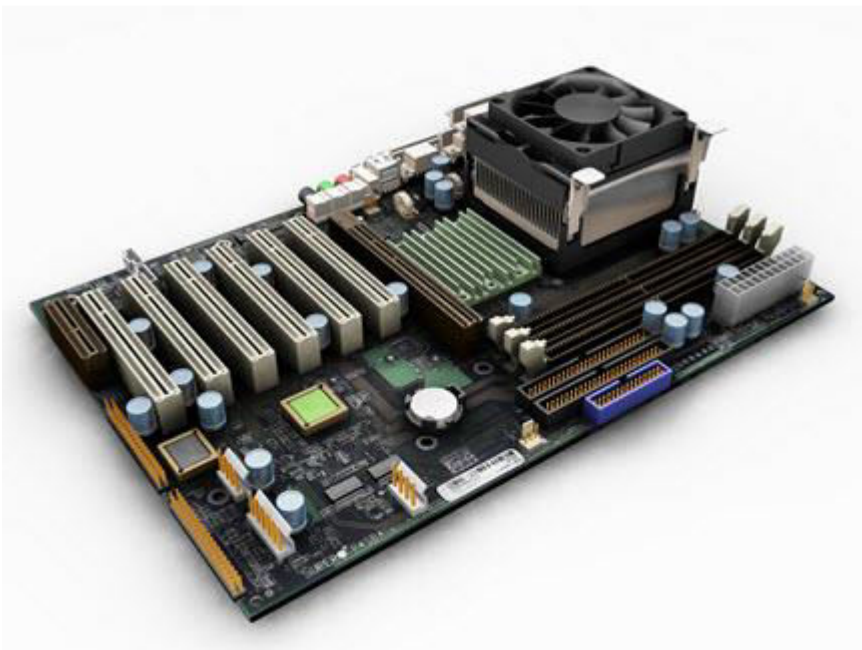
- Hard disk
- DVD drive
- Sound system
- Graphics system

2. An actual CPU

From the outside, the CPU looks like a single chip with lots of pins.



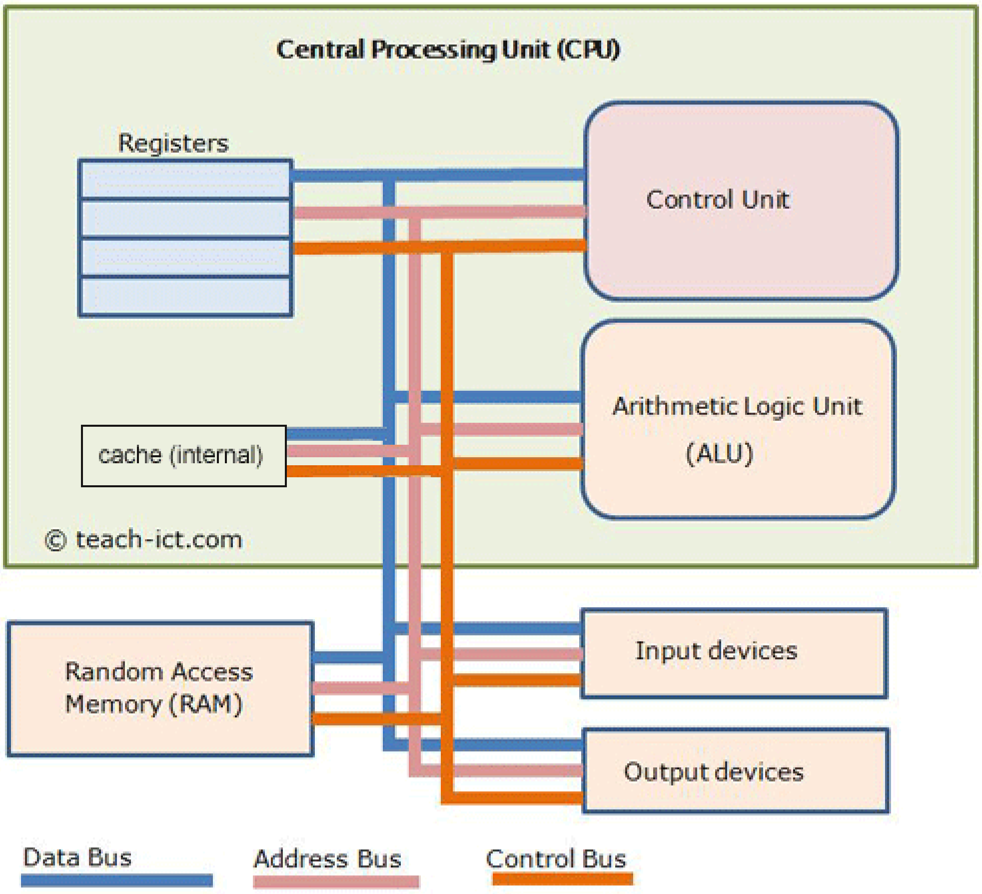
In a desk top computer, it is normally located on the 'motherboard'. This is a large printed circuit board where many of the main components of the computer are located. The photo below shows a motherboard with the CPU located beneath its large cooling fan.



3. Parts of a CPU

The CPU is a very complicated chip. It consists of billions of transistors which are tiny electronic switches which are either 'on' or 'off at any instant, depending on the program running at that instant'.

The diagram below shows the main parts of the CPU, along with its connections to external components such as RAM, input and output devices.



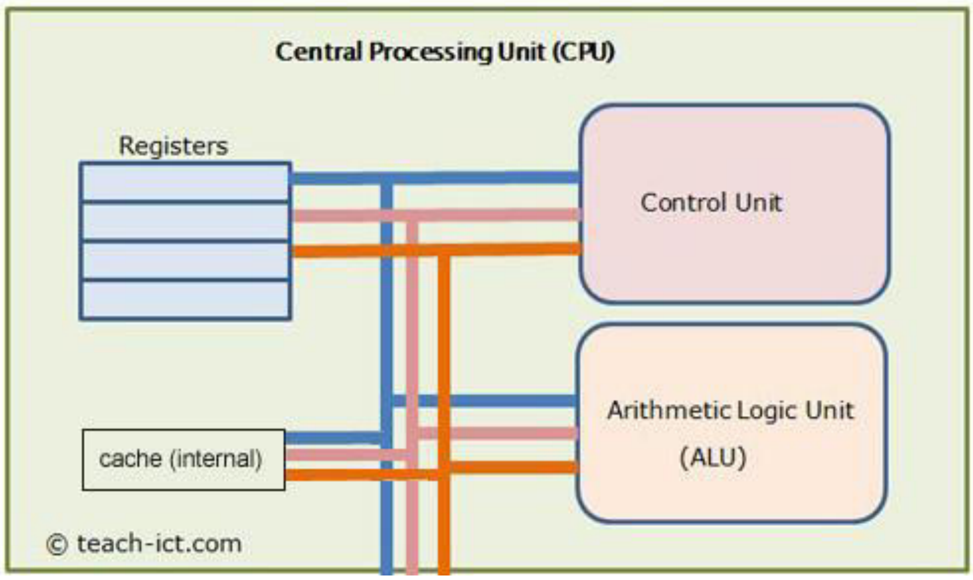
There are four main sections inside the CPU:

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

These are described in later pages. There are three main buses connecting all the parts together. These are the Data Bus, which carries the data, the Address Bus which carries information on the location of the data and the Control Bus, which handles commands to control devices such as the hard disk, the monitor and the graphics card.

Parts of a CPU

4. Control Unit

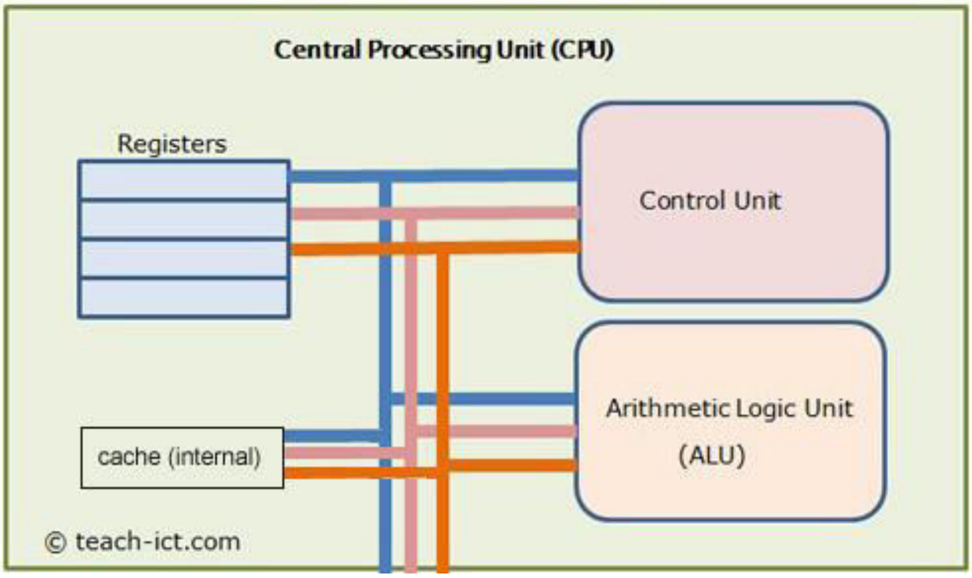


The control unit manages the tasks carried out in the CPU - it handles what data goes where, what is done with it and where the results need to go.

The control unit has three main jobs

1. It controls the way data moves around the CPU (timing signals)
2. It executes the instructions provided by the program (control signals)
3. It controls and monitors the flow of data between the CPU and other components such as input devices, memory, graphics card etc (signals to memory/devices)

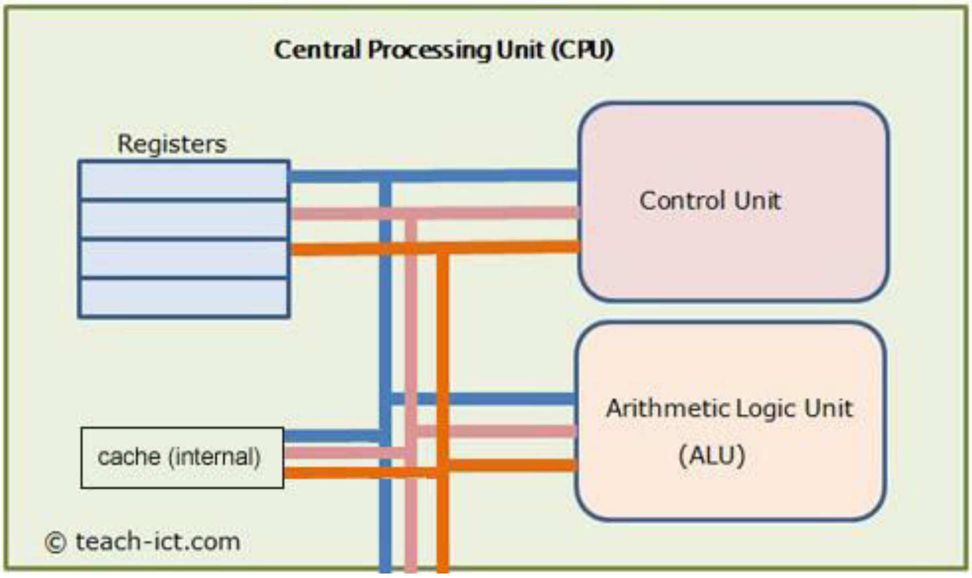
5. The ALU



The Arithmetic Logic Unit (ALU) handles the data processing inside the CPU. It consists of two types of processing

1. Arithmetic operation - it performs standard arithmetic operations such as addition, subtraction, multiplication etc
2. Logic operation - it deals with logic and comparisons such as 'is this value greater than that value' the answer is always either 'true' or 'false'.

6. Registers



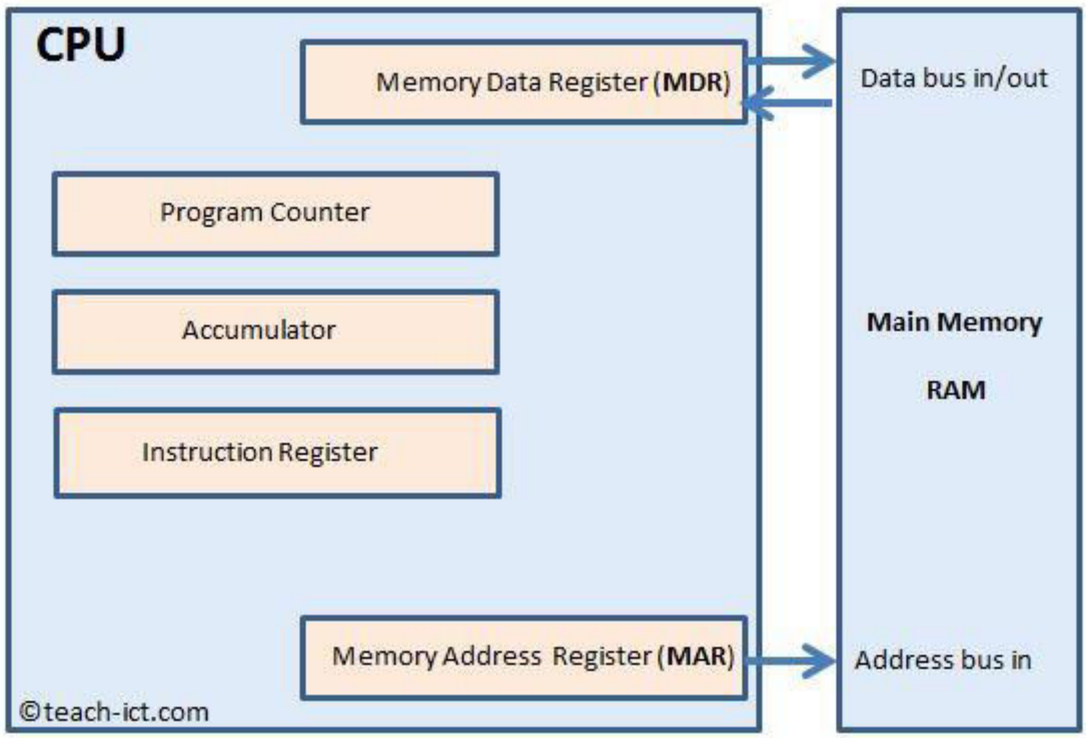
A 'register' is a type of memory inside the CPU. A register can hold a data value. For example an 8 bit CPU has a number of 8 bit registers, a 16 bit CPU has 16 bit registers and so on.

Registers are used to hold temporary data while a software program is running. As the CPU processes the data, the software program will shift the data in and out of the registers.

It is much faster to shift data to and from the registers rather than in and out of the Cache or RAM (Random Access Memory) and so this speeds up the processing time.

There are usually many registers available inside the CPU.

Some registers in the CPU have a specific purpose and so they have their own name. Here are some of them



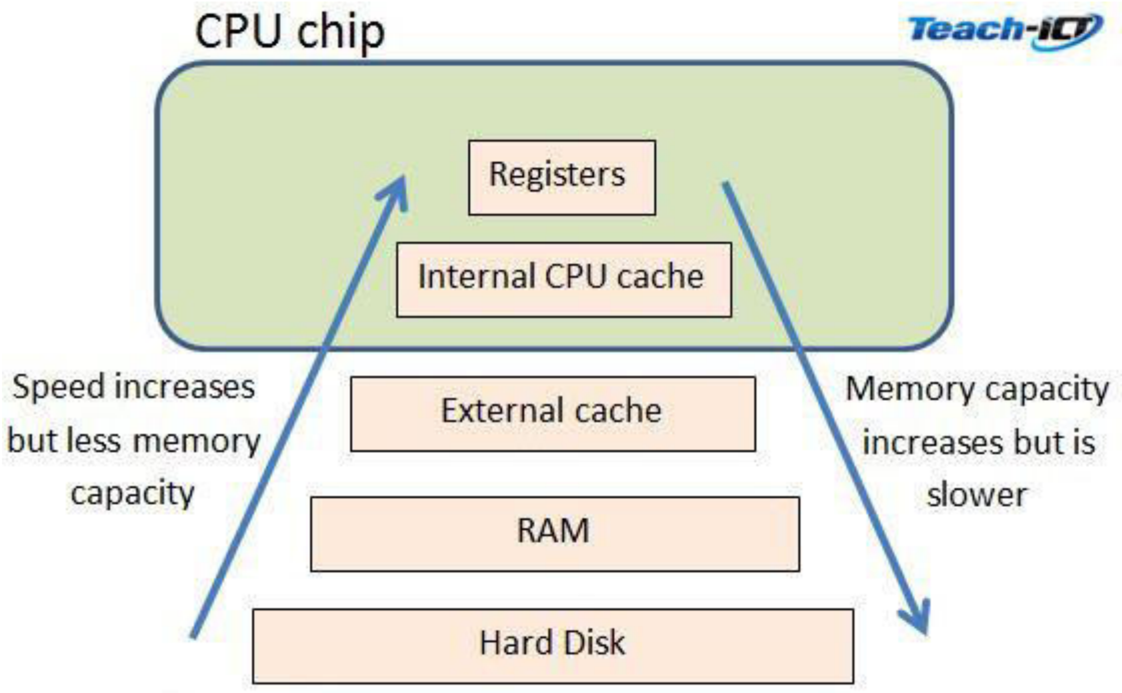
The MDR and MAR are used to move data and instructions to and from main memory. These are described more fully in the 'fetch-decode-execute' section. The Program Counter points to the next instruction. The Accumulator is where temporary data is stored as calculations are underway and the Instruction Register contains the next instruction to be run.

Parts of a CPU

7. Cache

In an ideal world, when the CPU needs data, it should be available instantly so as not to slow down the processing operation. But alas, it is not an ideal world - we need to have a system that can speed things along as fast as possible. This is the purpose of the cache.

A cache is a small, but extremely fast, type of RAM. It is often located inside the CPU chip itself. The diagram below explains why it is needed.



Data and programs currently in use are stored in RAM.

But RAM is comparatively slow to access when compared to the speed at which the registers work. So to help speed up the processing time, cache memory is used to store instructions or data that are either frequently used, have recently been used or are about to be used. This means that they don't have to be fetched directly from RAM.

There are different levels of cache. Level 1 (L1) cache is the fastest but the smallest size and resides inside the CPU chip. Level 2 (L2) cache is a bit slower but is larger, and so can hold more data, then Level 3 (L3) cache is slower still but can hold the most data. A cache though is always faster than the main RAM.

Parts of a CPU

8. Summary

- The purpose of a CPU is to process data and control other components within the computer system.
- The CPU is located on a 'motherboard'.
- The four main parts of a CPU are the Control Unit, Registers, Arithmetic Logic Unit (ALU) and the cache.

- The three main buses are Data bus, Address bus and the Control bus. These connect parts of the computer system.

- The cache is a very fast but small type of RAM.

- The CPU runs at the speed of its internal clock.

1. Introduction

In the early days of computing (1940s), computers were built to carry out a very specific task, for example breaking secret wartime codes. But if the computer then had to do another job, it literally had to be completely re-wired by hand. This could take weeks. There was no such thing as a 'software update' in those days!

Mr Von Neumann was a scientist who had an idea of how to create a computer that was far easier to change, this is known as the **Von Neumann architecture**.

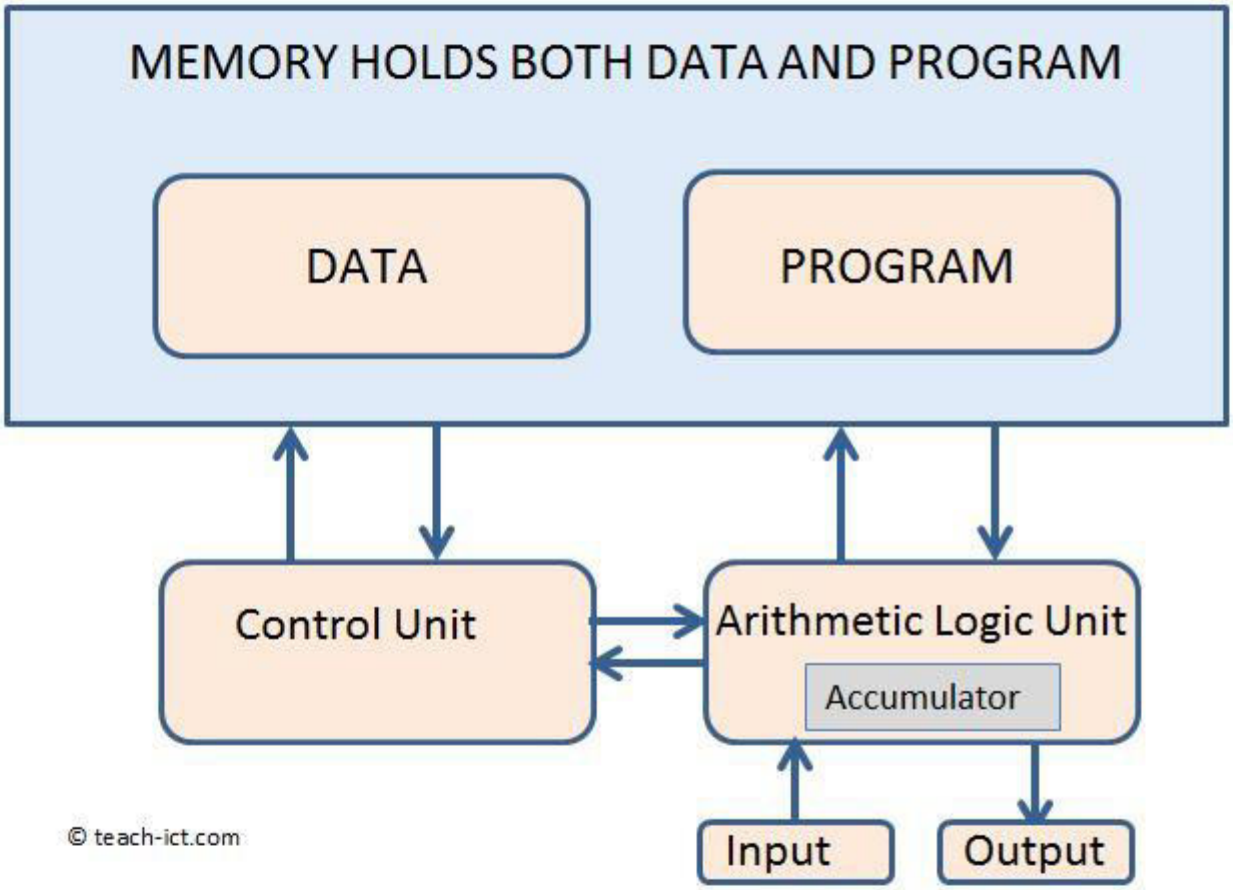
Von Neumann architecture is where both the data and the software that are currently being used are stored in computer memory (RAM).

It is also known as a **'stored program'** computer

With this architecture, the task a computer is working on can be changed by simply loading a different program into memory.

2. Von Neumann layout

A diagram of the Von Neumann architecture is shown below.



Note that a 'computer architecture' is not an actual computer - it is an ***idea*** of how a computer should be built.

The essential idea of the Von Neumann architecture is that both program and the data being used by that program are stored in a common memory block and they also share the same address and data bus.

What this means is that the operation of the computer can be changed immediately by simply running a different program in memory.

Modern computers have enough memory to support dozens of programs being held in memory but only one of them can execute instructions at any instant. A process scheduler handlies the switching between programs.

Von Neumann architecture

3. Features of the Von Neumann architecture

The Von Neumann architecture
There is enough memory available to hold both programs and data. This makes it easy to

The Von Neumann architecture
change task by loading a different program. In modern computers this memory is RAM.
There is a Control Unit responsible for handling the movement of instructions and data around the computer.
Data and instructions are carried on buses
There is an Arithmetic Logic Unit responsible for carrying out calculations and logic comparisons.
There is a special type of memory called registers which do specific jobs. The Accumulator is one such register. There are many other types of registers.
There is a way of inputting data into the computer and a way to output information.

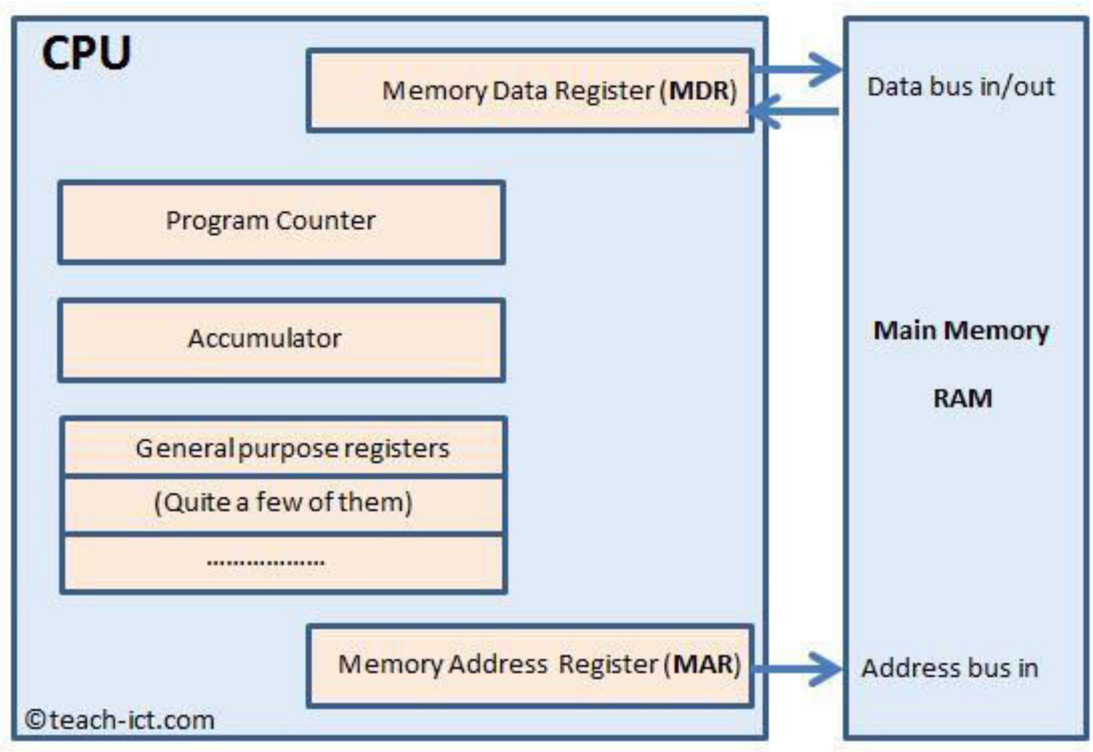
Most computers today are built with this idea in mind.

There are other architectures with which to build a computer but they are not part of this syllabus. If you are curious look up the Harvard computer architecture.

Von Neumann architecture

4. Registers in a CPU - MDR and MAR

The diagram below shows the main registers within a CPU.



Note that with the Von Neumann architecture, both data and programs, whilst in use, are located in main memory (RAM).

The main registers are shown above, these include two registers - the MAR and MDR that connect to the address and data bus respectively. There are also a number of specialised registers that are described in more detail later.

Von Neumann architecture

5. MDR and MAR

Memory Data Register.

In order for the CPU to fetch an instruction from main memory (RAM), the instruction is temporarily stored in a special register called the Memory Data Register (MDR).

Data (unlike instructions) goes both ways. Data can be fetched from main memory and brought to the CPU to be processed. After processing, it may be sent back to RAM to be stored (short term). All data must pass through the MDR whether it is on its way to the CPU or RAM.

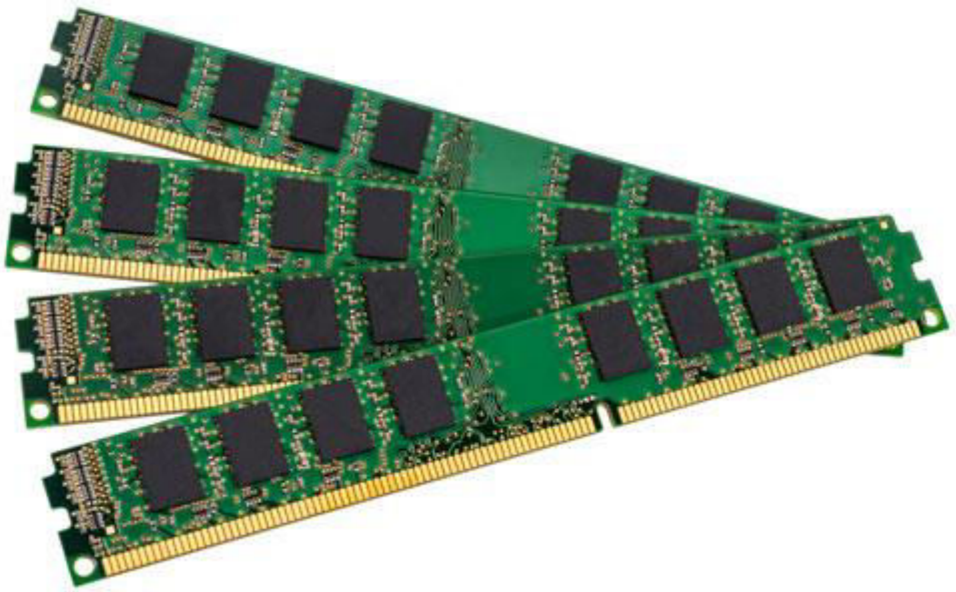
All data and instructions pass via the data bus.

Memory Address Register

Every instruction or data in main memory (RAM) is located at a specific location. This location has a unique address, just like a house has its own unique address.

The address of the data or instruction being accessed is temporarily stored in the Memory Address Register.

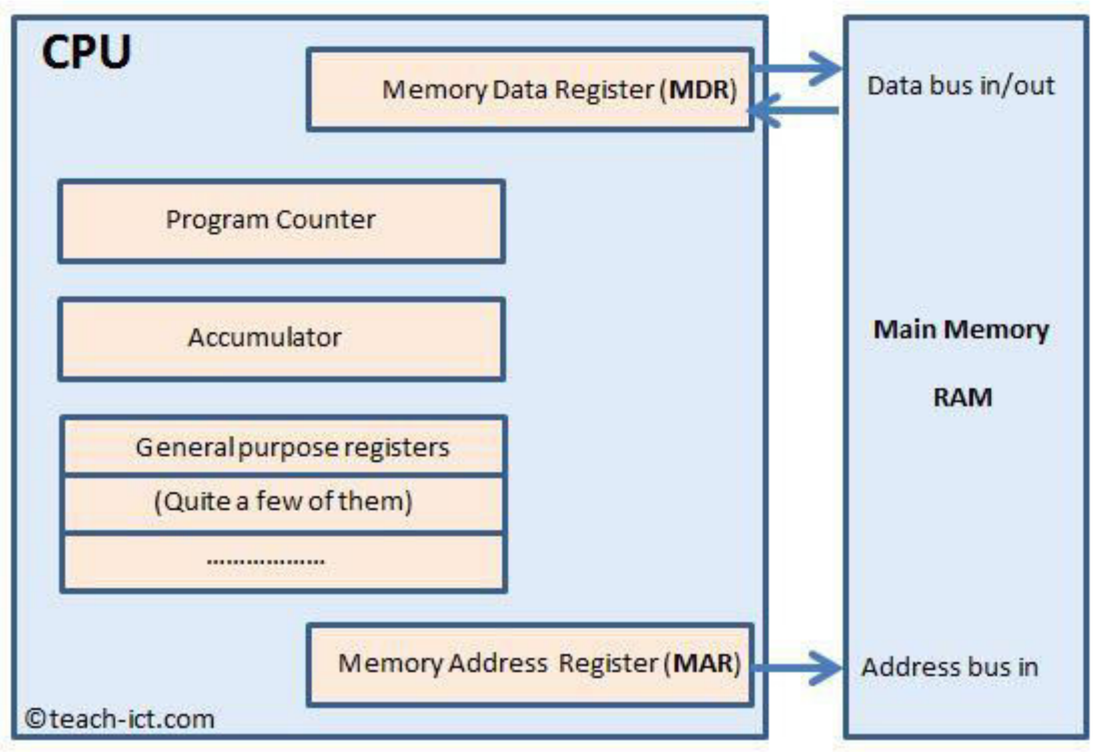
This address is passed to RAM by means of the address bus.



Von Neumann architecture

6. Registers in a CPU - Program Counter

The diagram below shows the main registers within a CPU.



Program Counter

All data and instructions in memory have a unique and specific address. As each instruction is processed, the software that is currently running, updates the program counter with the address of the next instruction to fetch.

The program counter then passes this next address to the memory address register as part of the standard fetch/decode/execute cycle (which is explained in its own section in the syllabus)

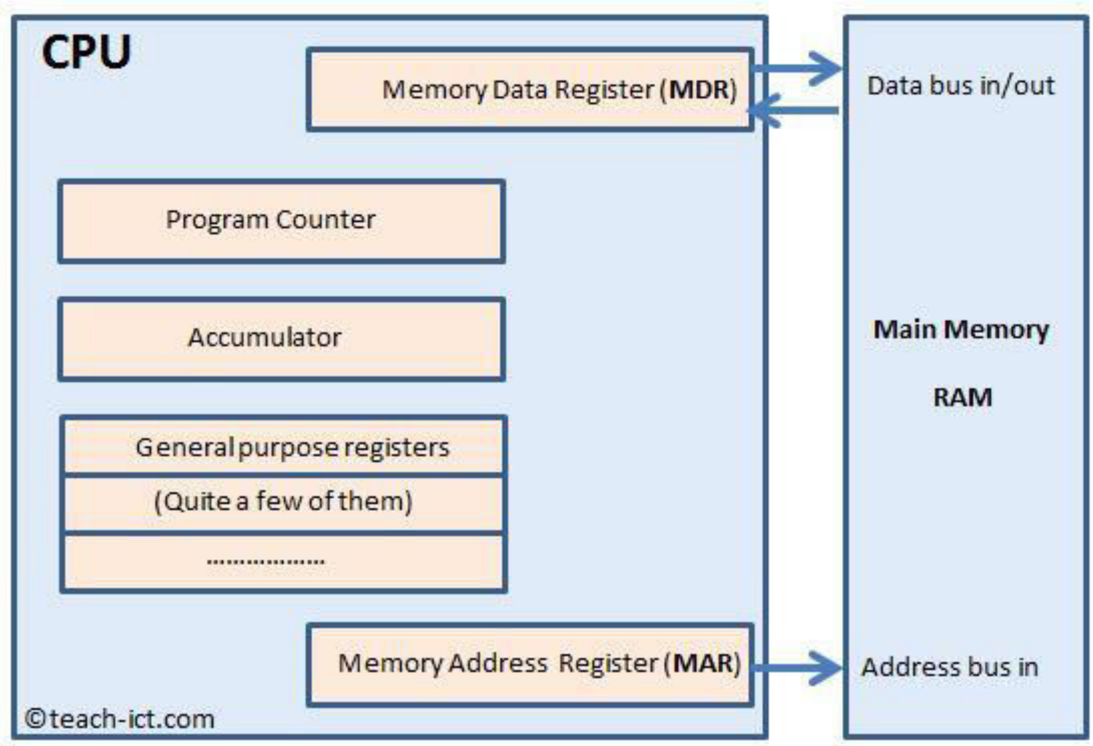
Normally the address of the next instruction will be only one location higher than the current one. Sometimes though, it is instructed to jump to a completely different address. This is called a branch instruction.

The software that is currently running controls the program counter.

Von Neumann architecture

7. Registers in a CPU - Accumulator

The diagram below shows the main registers within a CPU.



Accumulator

The Accumulator is used to temporarily store data whilst calculations are being performed within the Arithmetic Logic Unit (ALU)

You need at least one accumulator register to carry out calculations. But to make a programmer's life a bit easier, CPU manufacturers have included quite a few extra general purpose registers to store even more temporary data.

The Von Neumann architecture says that the 'accumulator' has to be available. The accumulator makes it much simpler to write software programs that handle data. It does this by storing temporary data while the program runs.

Also, having an accumulator means that the program can run much faster because the accumulator is a register and, as you already know, registers are the fastest type of memory. Without an accumulator, data would have to be stored in RAM which is much slower to access.

Von Neumann architecture

8. Summary

- Another name for the Von Neumann architecture is the stored program computer
- A computer architecture is not an actual computer but the idea of how a computer could be built.

- The Von Neumann architecture includes program and data loaded into memory

- Registers in a CPU includes the Memory Data Register (MDR) and the Memory Address Register (MAR)

- The Program Counter contains the the address of the next instruction

- The Accumulator holds temporary data whilst processing is underway.

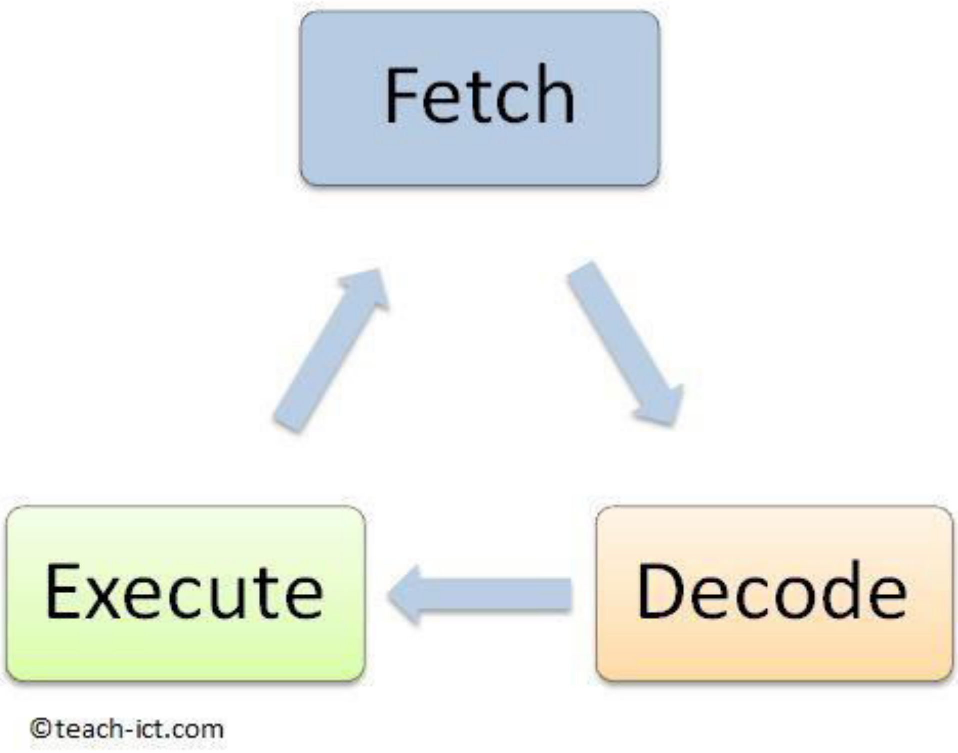
- Modern CPUs include a number of extra general purpose registers to make programs more efficient than having a single register needing to contain every temporary data item.

Fetch Execute Decode

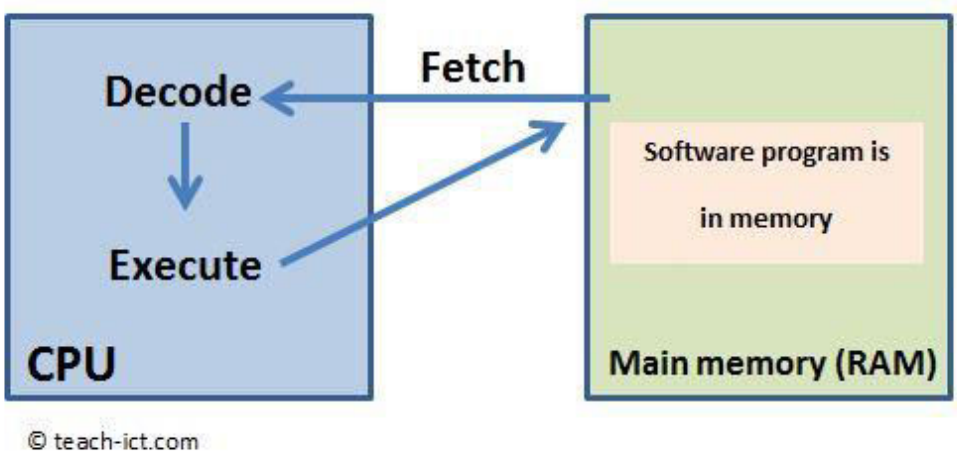
1. The Fetch-Decode-Execute cycle

The Fetch-Decode-Execute cycle is the basic method used by the CPU to carry out a single instruction contained in a computer program.

The diagram below shows how each step relate to one another. First there is a fetch operation followed by a decode operation and then an execute operation. Then another fetch begins and the cycle starts all over again.



The diagram below shows the software to be run is located in main memory (RAM). Usually the operating system loads it, ready for the CPU to use. The very first instruction in the program needs to be loaded into the CPU (fetch)

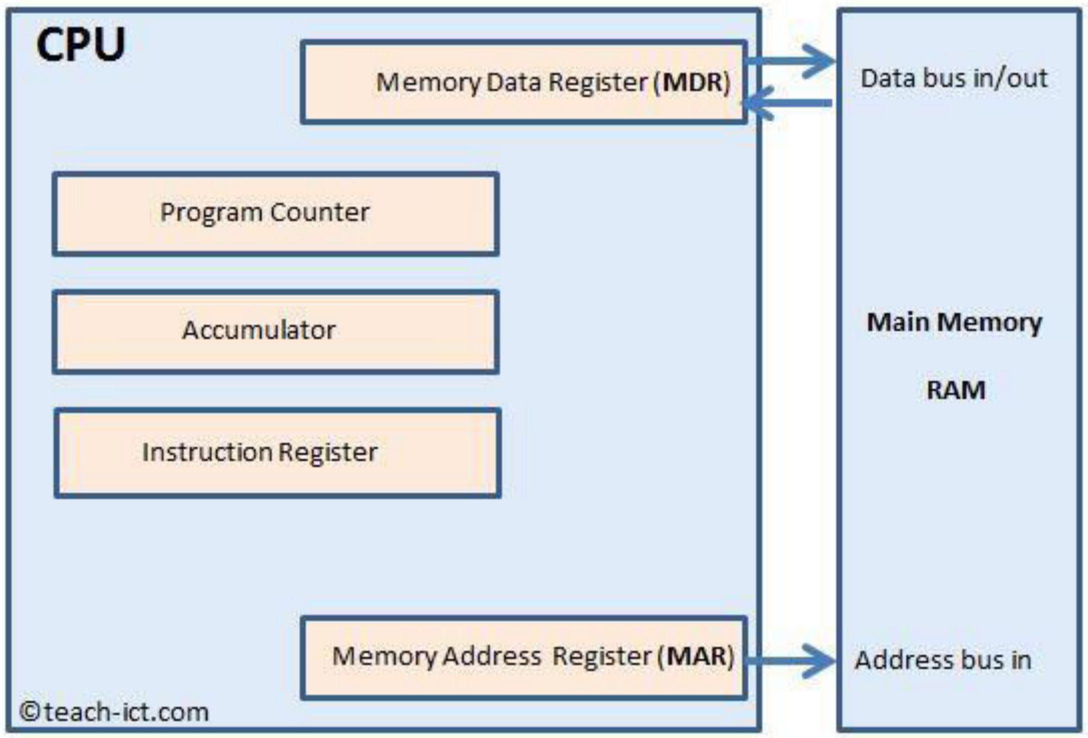


The fetch action loads an instruction into the CPU then the decode and execute stage take place inside the CPU. Once the first cycle is complete, the CPU fetches the next instruction.

Fetch Execute Decode

2. The Registers used

A number of registers inside the CPU are involved with the cycle. These are shown below



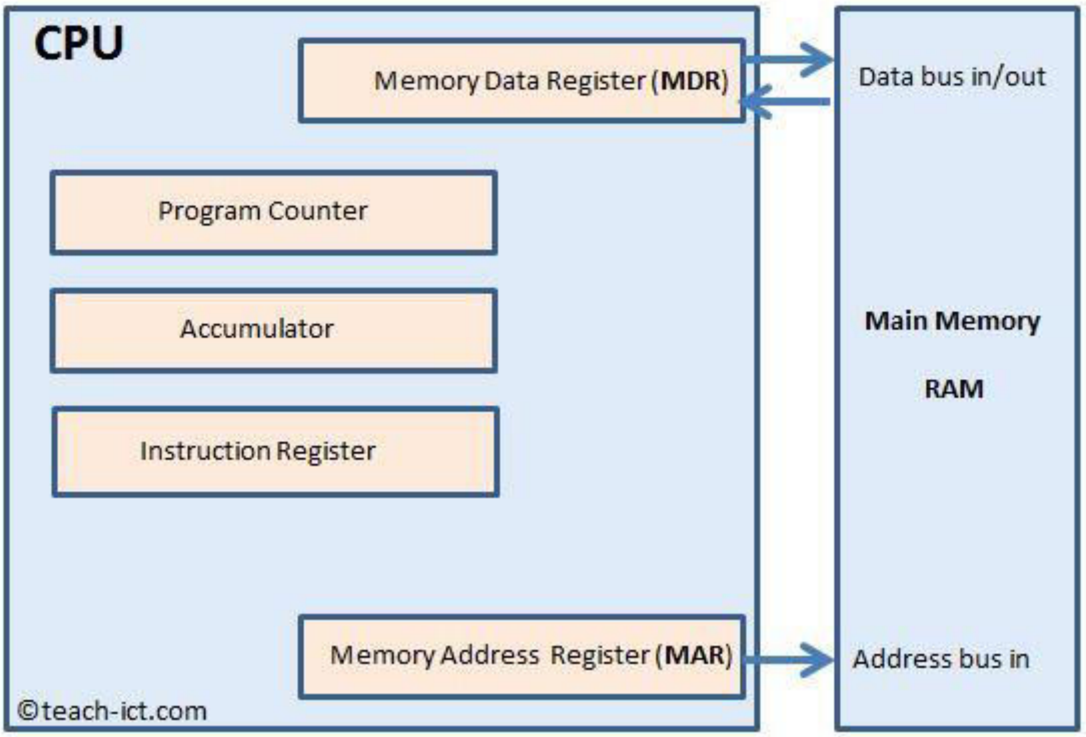
The registers are:-

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

We will next explain how these registers are used in each stage of the cycle.

Fetch Execute Decode

3. Fetch



'Fetch' is the first part of the cycle.

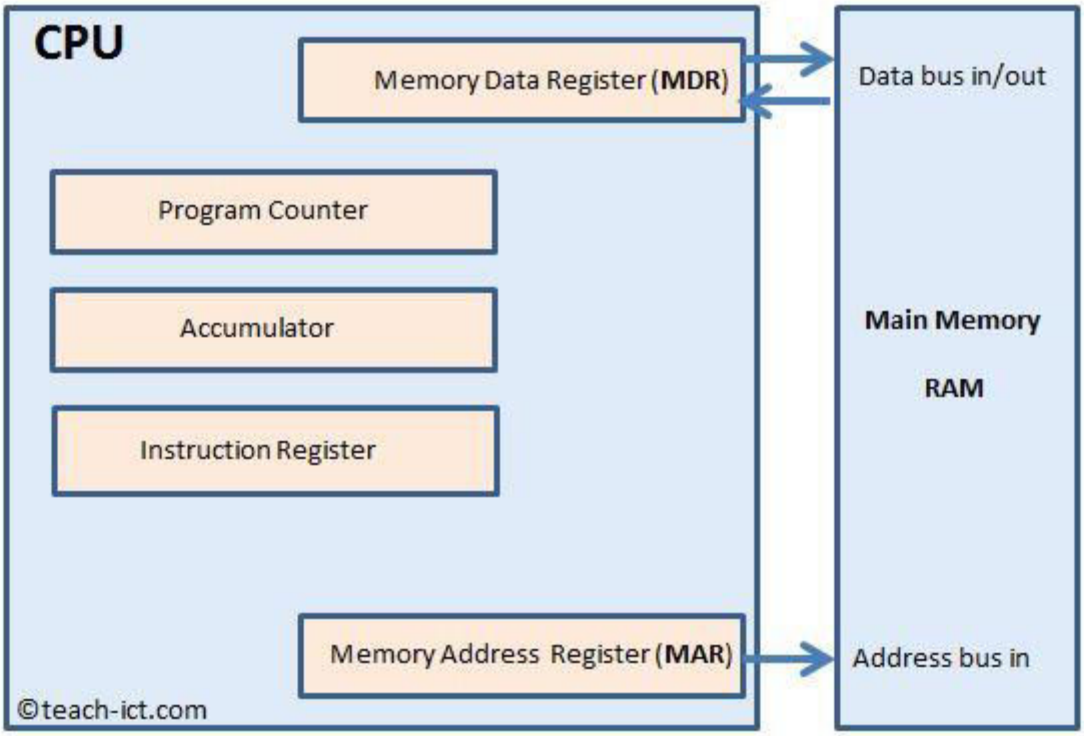
The actions are:

- The Program Counter is loaded with the location of the next instruction to be executed.
- This address is then loaded into the Memory address register (MAR) which is connected directly to the address bus. What this means is that the address bus is now pointing to the instruction in RAM
- This instruction is copied from RAM on to the data bus which then allows the Memory Data Register (MDR) to store a copy of it.

Now it is ready for the next phase which is 'decode'.

Fetch Execute Decode

4. Decode



Now that the instruction is available in the MDR, the next stage is called 'decode'. The purpose of the decode phase is to look at the content of the MDR and work out if it is a relevant instruction that the CPU can carry out.

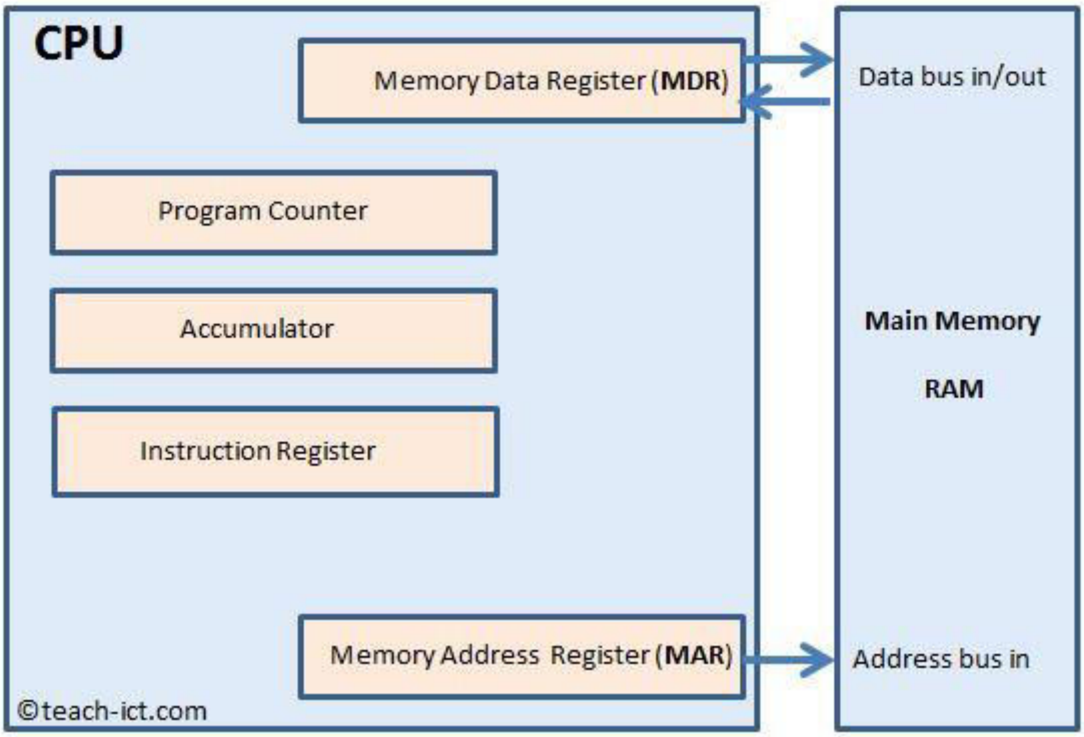
It does this by comparing the MDR item with the CPU's instruction set. If it matches, then the decode phase can prepare the CPU to actually do something with it.

The actions are:

- Check to see if the MDR contains a valid CPU instruction - if it is, load the instruction register with the instruction. If it is not valid, then ideally an error is flagged so that the program can deal with it.
- Set up the program counter with the address of the next instruction to be carried out.

Fetch Execute Decode

5. Execute



The final stage is called 'Execute' which means it actually carries out the instruction present in the instruction register.

The instruction often involves handling data in the accumulator. For example

'Increment accumulator' would increase the value in the accumulator by 1

'Add 5 to the accumulator' does exactly what it says

There could (most likely) be more than one register to handle temporary data in addition to the accumulator, but the basic fetch-decode-execute *idea* does not change, it simply involves more registers.

Once the execute phase is complete, the 'fetch' phase is visited once again. This is possible because the program counter was pre-loaded in the decode phase to point to the next instruction.

Fetch Execute Decode

7. Summary

- Software or program is initially loaded into main memory (RAM)
- A process called the Fetch-Decode-Execute cycle handles each instruction one at a time.
- The cycle also handles the loading of data from memory

- The first stage is 'Fetch' which loads an instruction or data from memory into the CPU
- The second stage is 'Decode'. The instruction is decoded by the Control Unit so the CPU can handle the last stage
- The third stage is 'Execute' which means carry out the instruction