Internal components of a computer

The **internal components** of a computer system consist of the hardware required to process data and to allow the processor to communicate with other devices such as secondary storage, display screens, and printers.

The main internal components of a computer system are:

- Processor (CPU)
- Main memory
- Input/output (I/O) controllers

These components are connected together by high-speed communication buses.

GCSE **Internal components**

Computer systems are all around us. For example, home heating systems, cars, games consoles, phones, watches, fitness trackers, and drones.

Computer systems

A computer system is identifiable as a **programmable device** that takes in **data**, processes it into useful **information**, and then **outputs** the information so it can be used.

What computer systems do you use and why do you use them?

Games console, for entertainment.
Watch, receive messages from social media accounts.
Phone, access the internet and apps for social reasons.

The **internal components** of a computer system consist of the **hardware** required to **store** and process **data**, and communicate with external devices.

The input/output (I/O) controller communicates with external devices, such as the input (e.g. keyboard, mouse), output (e.g. screen, printer), and external storage devices (e.g. USB stick). External devices that are portable are also called **peripheral devices** or **peripherals**.

Give two examples of each of the following:

- Inputs
- Outputs
- Storage

Inputs: Touch screen, microphone
Outputs: Projector, speaker

The main internal components of a computer system are:

Component	Description
Processor	Part of the computer that executes program instructions to process data, and handles main memory and input/output operations
Cache memory	Temporarily stores frequently used instructions
Main memory	Stores data and instructions while they are being processed
Input/outputs	Accesses other components such as the screen and secondary storage, which is used to permanently store data such as the operating system and user files

GCSE The processor

The processor is the part of the computer that processes data by **executing** programs. It also manages the rest of the hardware.

GCSE Main memory

Main memory is memory that can be accessed directly by the **processor**. The main memory consists of memory locations that store instructions or data. There are two types of main memory:

- **volatile memory**, which loses its contents when the power is removed
- **non-volatile memory**, which keeps its contents even without power

The term **main memory** is often used to mean **random access memory (RAM)**, which is the working memory that is used by the processor. RAM is a **volatile** (temporary) memory that has addressable locations. Each location can be accessed randomly, so any **instruction** or **data** can be placed in any location (overwriting whatever it previously contained).

This is different to the role of **read-only memory (ROM)**, a type of **non-volatile** (permanent memory), and also different from **secondary storage**, which the processor can't work with directly.

GCSE **Buses**

The different components of a computer system are connected together using **[glossary-inline:glossary-page-b[bus"buses"]**. A **bus** transfers **data** and **signals** between components inside a computer. For example, in order to execute **instructions** and **process** data, the **processor** needs to communicate with the **main memory** and with the **input and output devices**.

In this computer system, there are three main types of buses, collectively known as the **system bus**.

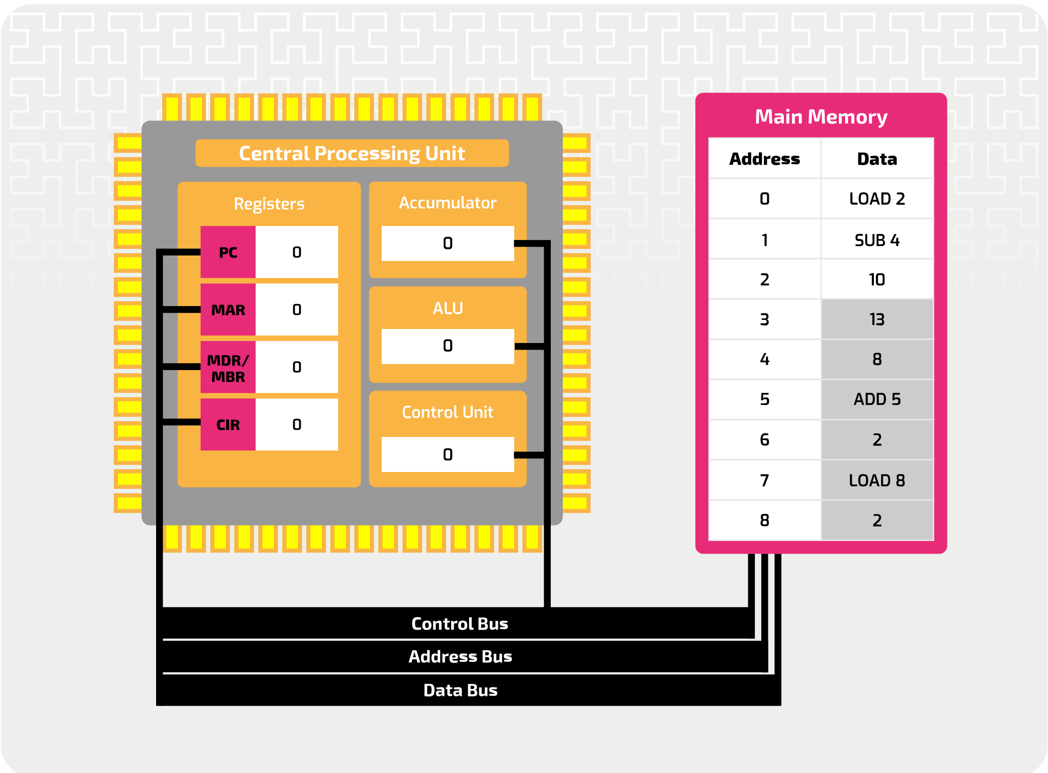
Bus	Purpose
Address	Carries address locations of stored data from the processor to memory and input/output devices
Data	Sends the data to and from the processor, memory, and input and output devices
Control	Carries signals that coordinate the operation of the components

The **address**, **data**, and **control bus** together are known as the **system bus**. The system bus is used to connect the processor (CPU) and main memory (RAM).

The following image illustrates an abstract model of the connections between input and output devices, the processor (CPU), and the main memory (RAM).

There are a few things to note:

- Each bus consists of a set of parallel — **not single** — lines
- Each connection of a component to a bus is made up of **parallel lines**
- Input, output, and storage devices communicate with the other components via **I/O controllers**
- The connection is only **one-way** from the I/O controller of the **input device** towards the **data bus**
- There is a **one-way** connection from the **data bus** towards the I/O controller of the **output device**
- There is a **one-way** connection from the **processor** towards the **address bus**
- There is a **one-way** connection from the **address bus** towards all the other components, i.e. the **main memory** and the **I/O controllers**



How components connect to the system bus to communicate and exchange data

A Level Internal components

The **internal** components of a computer system consist of the hardware required to store and process data, and communicate with **external**, peripheral devices.

The main internal components of a computer system are:

- Processor
- Main memory
- Input/output (I/O) controllers

These components are connected by the **system bus**, which is made up of the address bus, data bus, and control bus.

Other types of memory are crucial for the operation of a computer system. Secondary storage (e.g. SSD, HDD) provides permanent storage for programs and data. Secondary storage is discussed further in the hardware topic.

A Level The processor

The **processor**, sometimes referred to as the CPU (central processing unit), is the part of the computer that processes data by **executing program instructions**. At processor level, these will be **low-level instructions** in the form of **machine code** that the processor has been designed to handle, based on a specific processor instruction set.

For the processor to be able to execute a program, the program instructions need to be transferred from secondary storage into main memory from where they can be fetched, decoded, and executed. The data that needs to be processed is also loaded into main memory (from secondary storage) or provided by the input and output devices via the **I/O controllers**.

A Level Main memory

Main memory is memory that can be accessed directly by the processor. Each **memory location**, where instructions or data are stored as binary sequences, has a physical **address**, which is a number used to locate that memory location and access its contents.

There are two main types of main memory:

- RAM, which is the **working memory** that is used by the processor during the fetch-decode-execute cycle
- ROM, which is used in the boot process for the computer system

Main memory is distinct from secondary storage, which the processor can't work with directly. Secondary storage must be accessed through I/O controllers. You can read more about main memory on this concept page which also covers cache, registers, and virtual memory.

A Level

Input/output (I/O) controllers

All external (peripheral) devices are connected to the processor through **I/O controllers**. These provide the mechanism for:

- input data to be received for processing from input devices, such as keyboards
- the results of computation to be output from the system to output devices, such as display screens

Secondary storage devices are also connected to the processor through I/O controllers. Secondary storage, in contrast to main memory, cannot be accessed directly by the processor. It is not favourable for peripheral devices to be directly attached to the processor. Each device operates in a different way, at different speeds, and using different electronic signals. An I/O controller can be described as an **interface** between the core computer system and its peripherals.

I/O controllers provide a set of **addressable registers** that the processor (CPU) can access to communicate with the I/O devices. For example, the characters entered using a keyboard would be stored in an I/O controller that the processor (CPU) can access and process, such as displaying the characters on a screen.

A **physical port** (e.g. a USB port used to connect a keyboard to a computer) on the controller provides a connection for the peripheral. The peripheral device has a cable or connector to connect it to the port. The I/O controller circuitry facilitates the exchange of data in the form of electrical signals between the processor and the peripheral. The controller then translates the signals into the correct form. The peripheral device almost certainly operates at a much slower speed than the processor. The interface provides a buffer so that the disparate timing requirements can be satisfied.

Each peripheral device can have its own I/O controller or can share a hub controller (as is the case with USB). Large desktop computers usually have slots where additional specialist controllers can be added.

When a new peripheral is added, a **device driver** is also needed. This is a piece of software that provides a software interface for the peripheral device, which enables the operating systems to access the functions of the hardware without having to be configured specifically for every device that is used. Many devices are now 'plug and play', which means that the driver is either embedded or automatically downloaded when the hardware is connected.

A Level **Buses**

The components of a computer system are connected together using **buses**. A bus is a communication system that is used to transfer data between components.

- The **system bus** is a set of **parallel connections** that allow internal components to communicate with each other and exchange data.
- There are also **external buses** that are used to connect the peripherals to the processor. These can be **serial** or **parallel** connections.

Buses can be implemented in different ways, depending on their intended use, for example wires, electronic pathways, and tracks on printed circuits.

A model of the system bus

In Von Neumann architecture, the **system bus** is made up of:

- an **address** bus
- a **data** bus, and
- a **control** bus

The system bus is used to connect the processor (CPU), main memory, and the I/O controllers. The following diagram illustrates an abstract model of the connections.

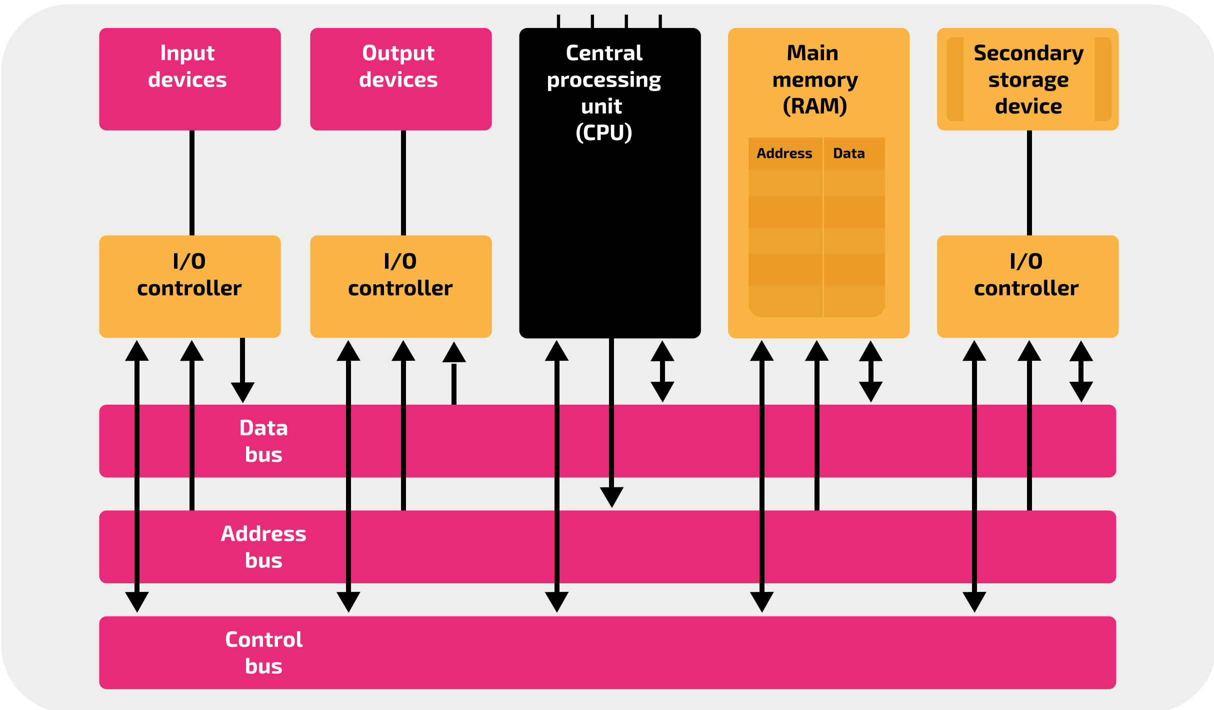


Figure 1: How components connect to the system bus to communicate and exchange data

There are a few things to note from the diagram in **Figure 1** that will help you identify components if you have an unlabelled or partially labelled diagram.

- Input, output, and secondary storage devices communicate with internal components via I/O controllers:
  - There is a one-way connection from the I/O controller of an input device towards the data bus
  - There is a one-way connection from the data bus towards I/O controller of an output device
  - There is a two-way connection from the data bus towards I/O controller of a secondary storage device
- There is a one-way connection from the processor towards the address bus (as only the processor can issue addresses)
- There is a one-way connection from the address bus towards all the other components, i.e. the main memory and the I/O controllers

- There is a two-way connection between main memory and the data bus
- There is a one-way connection between main memory and the address bus

Remember that each bus consists of a set of **parallel — not single — lines**.

A Level **Data bus**

The **data bus** is used to transfer data and instructions. The data bus is **bidirectional**, i.e. it allows a two-way connection between internal components of the system — allowing values to be written to or read from a location.

- Data is transferred to and from the processor
- Data is transferred to and from main memory
- Data is transferred to and from the I/O controllers
- Instructions are carried from main memory into the processor (to be decoded and executed)

The **width** of the data bus refers to its number of parallel lines. This determines **the number of bits that can be transferred in one operation**. For example, how many bits can be transferred in one go between the memory and the processor. It is typically a multiple of a byte (e.g. 8, 16, 32, or 64 bits).

- If the width of the address bus is 8 bits, then 2^8 bits can be transferred at one time
- If the width of the address bus is 16 bits, then 2^{16} bits can be transferred at one time

In general, if the width of the data bus is expressed as n bits, then 2^n bits can be transferred at one time. The amount of data that can be fetched at one time can affect the processor performance.

A Level **Address bus**

The **address bus** is used to specify the address of a memory location to either **read** (i.e. load) data from or **write** (i.e. store) data to that memory location.

There is a **one-way** connection from the processor to the address bus and a **one-way** connection from the address bus to the main memory and to the I/O controllers. This is because the address bus is a **unidirectional** bus, which allows the processor to establish a connection with an addressable 'unit', whether it's a memory location or an I/O controller.

The **width** of the address bus refers to its number of parallel lines, which determines **the number of bits that can be used to form an address** of a memory location. It is typically a multiple of a byte (e.g. 8, 16, 32, or 64 bits).

- If the width of the address bus is 8 bits, then there are 2^8 numbers that can be used to address memory locations
- If the width of the address bus is 16 bits, then there are 2^{16} numbers that can be used to address memory locations

In general, if the width of the address bus is expressed as n bits, then there are 2^n numbers that can be used to address memory locations.

Therefore, the width of the address bus determines the maximum number of addressable memory locations, i.e. the **maximum memory capacity of a computer system**. This is called the **address space** and it can affect processor performance.

A Level **Control bus**

The **control bus** is used to send **control signals** that manage and orchestrate the operations that take place inside a computer system. This includes exchanging **status signals** between the components of the computer system, and transmitting **clock signals** required for the coordination of operations. For example, a control signal can be used to:

- Request communication between two units
- Acknowledge a communication request
- Specify the type of data that is being transferred via the other buses (i.e. data values, instructions, or addresses)
- Synchronise the communication between the components using the clock pulses

The control bus is **bidirectional**, i.e. there are two-way connections between the components that the control bus connects.

Examples of control signals include:

- **Memory read:** Places data from a specific memory location (whose address is on the address bus) onto the data bus
- **Memory write:** Stores the data from the data bus onto a specific memory location (whose address is on the address bus)
- **Bus request:** Signifies that a component needs to access a bus
- **Bus grant:** Signifies that a component is informed that it can use the bus it requested access to
- **Bus busy:** Signifies that a bus is not available for use (as it is used by another component)
- **Interrupt request:** Signifies that an error or exception has occurred that requires the attention of the processor (CPU)
- **Clock signals:** The control bus supplies the components with clock pulses generated by the system clock