Learning Aims

- Difference between hardware and software
- Difference between internal and external components
- Name the different parts of a computer
- Describe the purpose of each component
- To understand the difference between an input and an output device

Key terms

Hardware
Software
Input-process-output
Components
Peripherals
Input device
output device

Motherboard
RAM
Hard drive
GPU
CPU
Network interface card

What is a computer?

- Computers are made up of hardware and software
- Hardware is the physical parts of the computer
- Software is the set of instructions that tells the computer what to do
- They work together to process data and carry out tasks

Today, we will be looking at the hardware components!

Hardware & Software

Computers are made up of hardware and software working together to process data and carry out instructions.

Hardware

- Hardware is the name given to the parts of the computer that you can physically touch.
- This also includes the internal parts of the computer like the hard drive, motherboard and CPU.

Software

- The term software relates to the sets of instructions that tell a computer what to do or how to perform a task.
- Software is basically the **programs** that are stored on a computer.

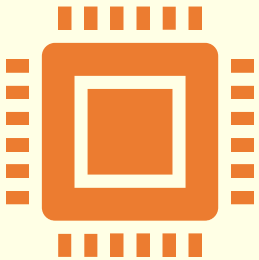
What is a computer?

1. Takes an input, usually from the user (a human)
2. Processes data based on the input
3. Outputs the data in a meaningful way

Note: They will only do what they have been programmed to do

Input	→	Process	→	Output
• Typing • Speaking • Send to print	→	• Getting key press • Recording voice • Store file in printer	→	• Put key on screen • Sending sound file • Print file on paper

Computer Systems



Internal Components (attached to the motherboard)

The processor

Main memory (RAM)

I/O controllers

Buses

Video card/ graphics card

External components (peripherals)

Keyboard, mouse, printer, disk drives

Output Devices - Video display / Printer

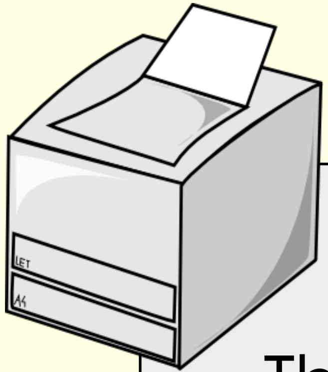
Secondary Storage eg. Flash drive

Input & Output Devices

Input Devices

An input device is a piece of hardware that is used to provide data to the computer; basically, it's a device that allows the user to get data into the computer.

How data enters the computer



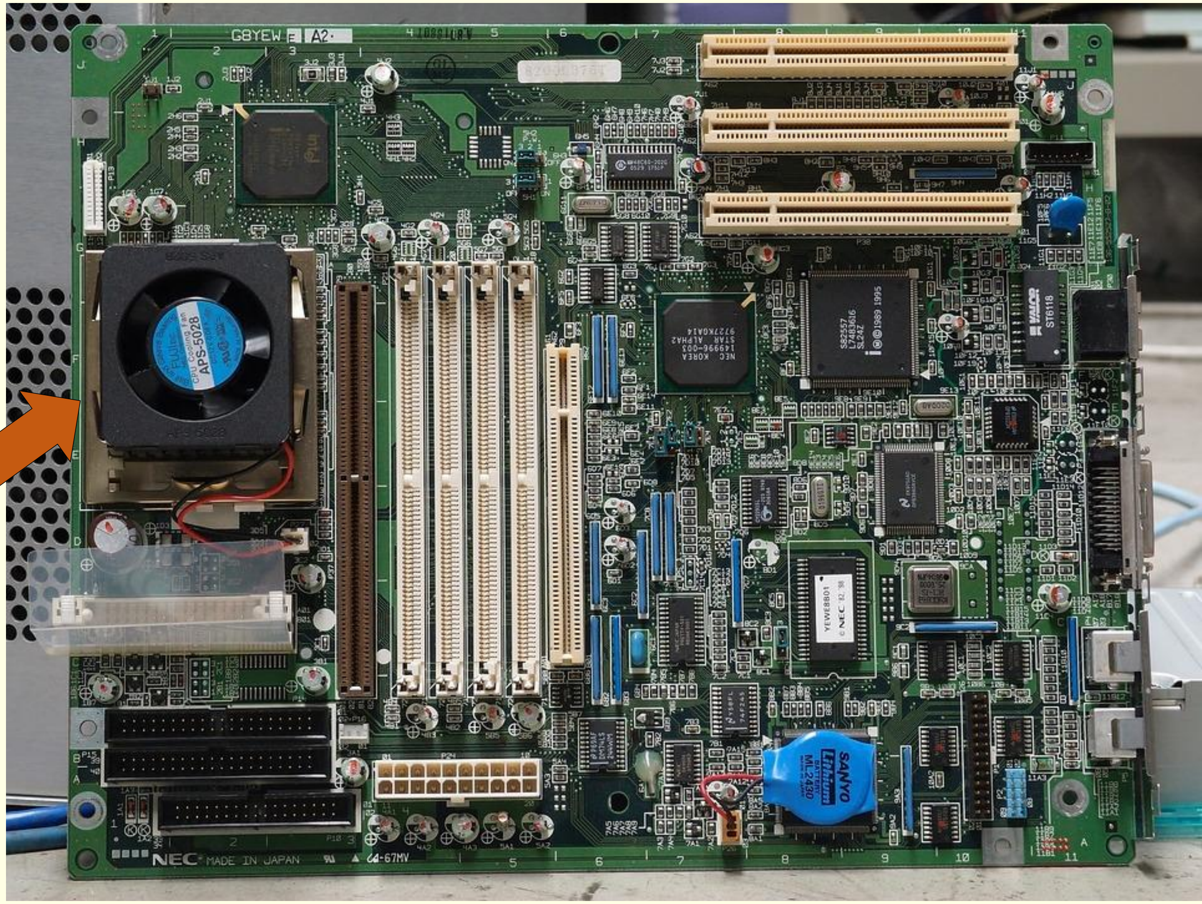
Output Devices

The opposite is true for output devices – these are pieces of hardware that allow data to come out of the system; they allow the user to see or hear the results of what has been processed in the system.

How data leaves the computer

Motherboard

- A circuit board
- Connects all other components together
- The CPU is underneath this fan and heat sink



Random Access Memory (RAM/ Main memory)

- Fast memory where your programs are held while the computer is running

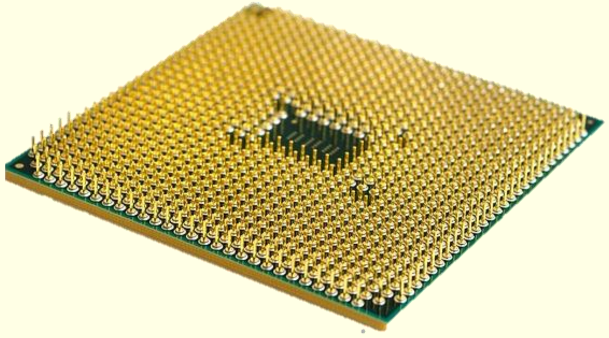
- Does retain data unless it has electricity running through it → "Volatile"

We have a lesson on memory later.

CPU

- Central Processing Unit
- Does almost every processing task, such as calculations
- Made of millions of transistors (tiny switches)

We'll be covering the CPU in detail next lesson



GPU

- Graphics Processing Unit or "Graphics Card"
- Has a processing unit, like a CPU, specialised for processing images
- Required for high-end gaming and media (such as photoshop)

Heat Sink

- Sits on top of your CPU to cool it down (by conducting the heat away from it)
- Often connected to a fan to help it get rid of the heat
- Without it your CPU could melt or set fire to your computer!

FAN

- Used for providing air flow to your computer
- Increased air flow means a cooler computer
- A cooler computer is a faster computer!

Hard drive

- Also known as a HDD or Hard Disk Drive

- A stack of magnetic disks used to store lots of data

- Where all your files are kept

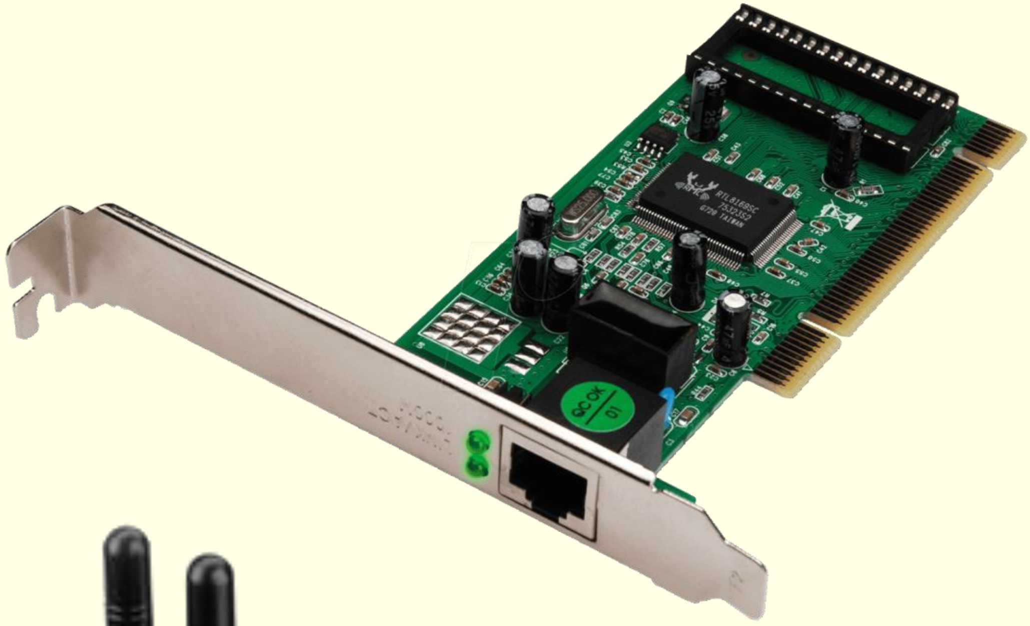
We'll have a lesson on storage, too.

Network Interface Card

- The interface between your computer and a network

- Used to connect your computer to the internet

- They may also have antenna for wireless support (WiFi)



PSU

- Power Supply Unit

- As the name suggests, it supplies power to the rest of the computer

- Very dangerous, do not mess with this, even when it is off!

Learning Aims

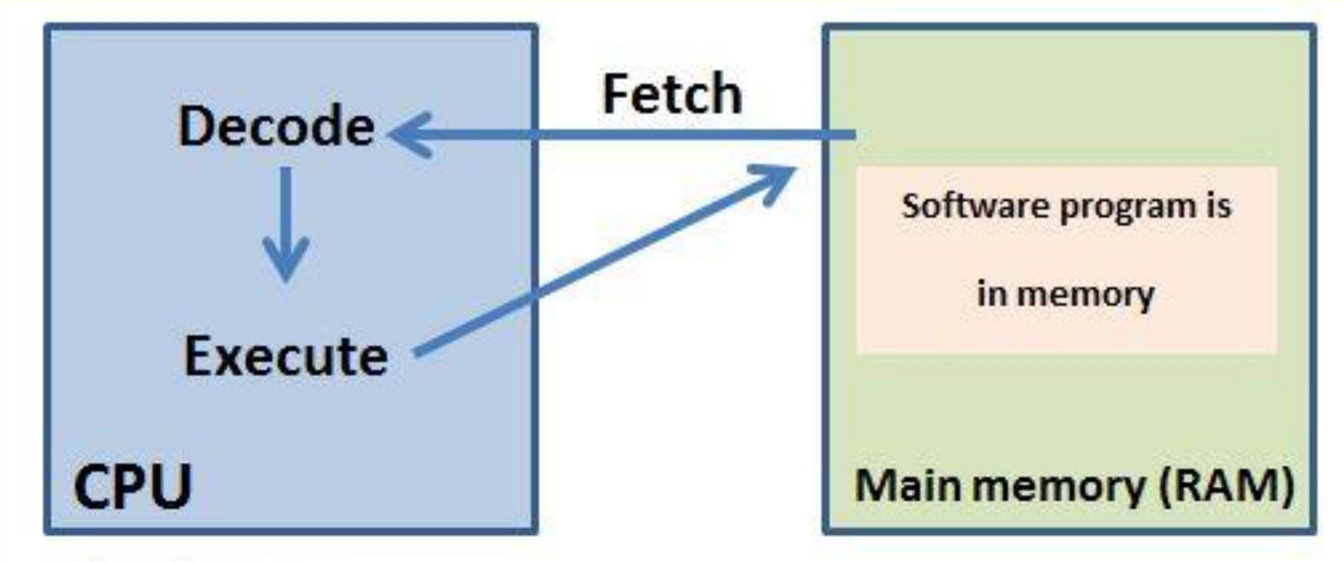
- Define what is meant by the 'stored program concept'
- Describe the hardware components used in the **Von Neumann architecture**
- Be able to describe the following components in the Central Processing Unit (CPU)
  - Arithmetic Logic Unit (ALU)
  - Control Unit (CU)
  - System Bus

Key terms

CPU
Arithmetic Logic Unit (ALU)
Control Unit (CU)
Registers
Main memory
Stored program concept
Data Bus
Address Bus
Control Bus

CPU: What does it do?

- The "brains" of the computer
- Processes data (CPU also known as "The Processor")
- Retrieves data/instructions from main memory (<u>Fetch</u>)
- Works out what to do with the instruction (<u>Decode</u>)
- Carries out instructions (<u>Execute</u>)

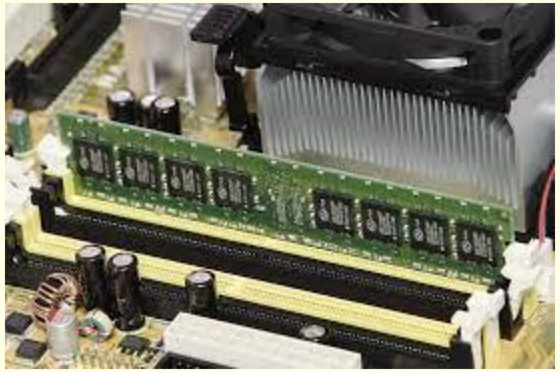


History of computers

mechanical devices

Electrical devices
Using programs on punched tape

Digital Computers
Programs and data stored in memory

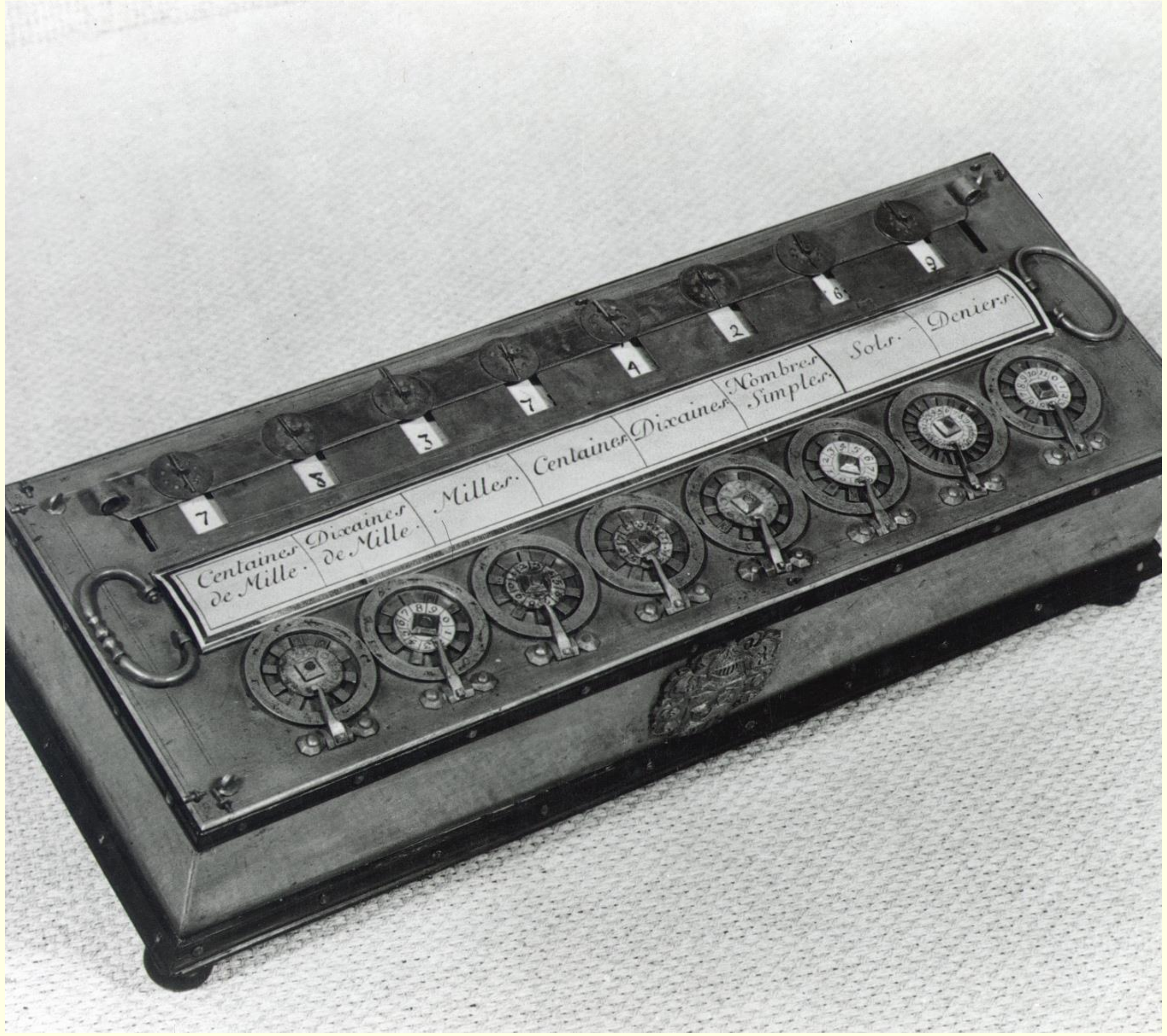


A little history

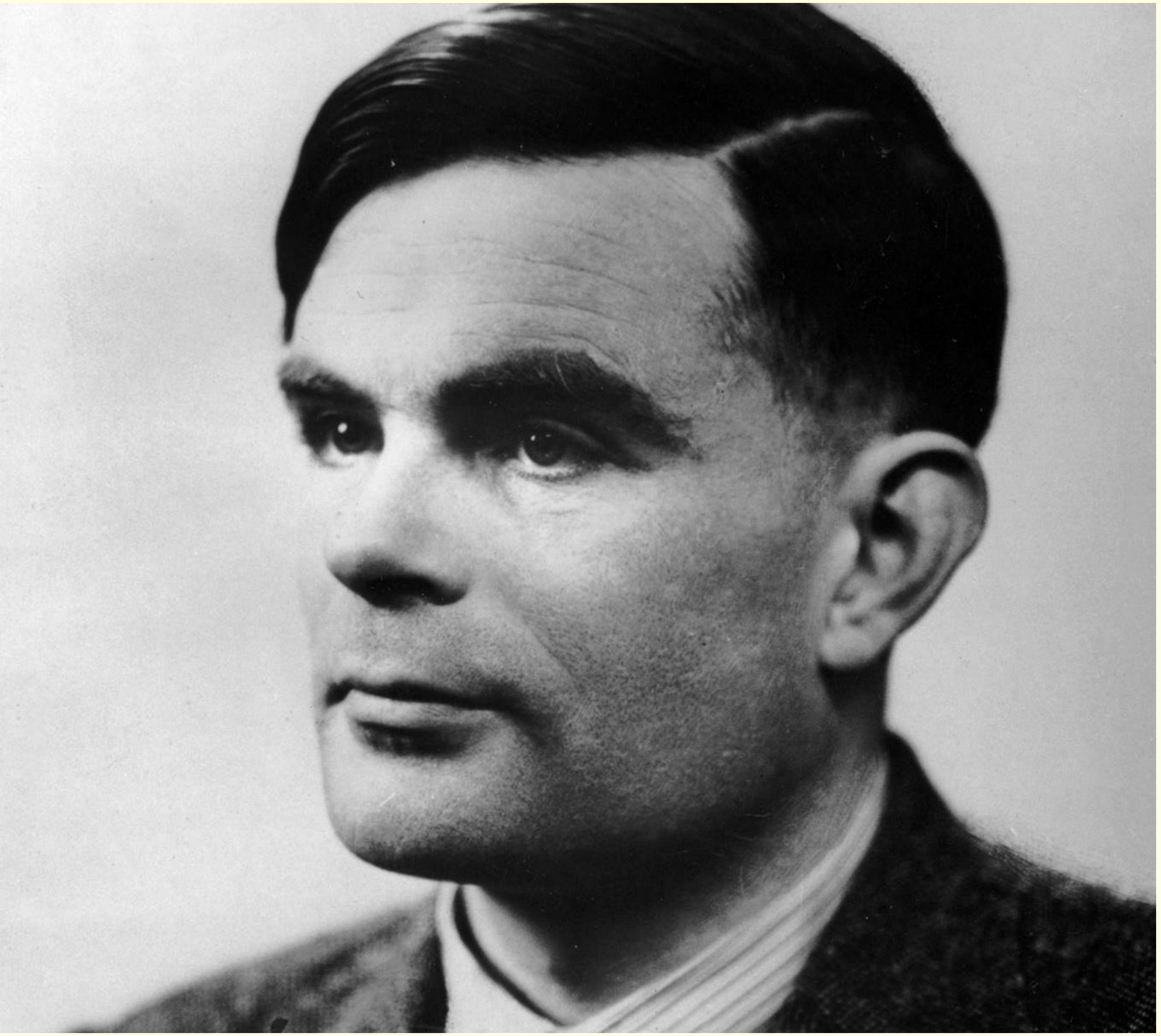
- As long ago as the 1640s, Mathematicians were creating **mechanical devices** that could perform mathematical operations. They had hundreds or even thousands of moving parts.

- These devices were created to fulfil a single purpose. If you wanted to change the purpose of such a device, it needed to be rebuilt or rewired.

- Pascal's calculator (shown here) was designed and built by the famous mathematician in 1642 (when he was 19 years old!) to help him with tax calculations. It performed multiplications and divisions.

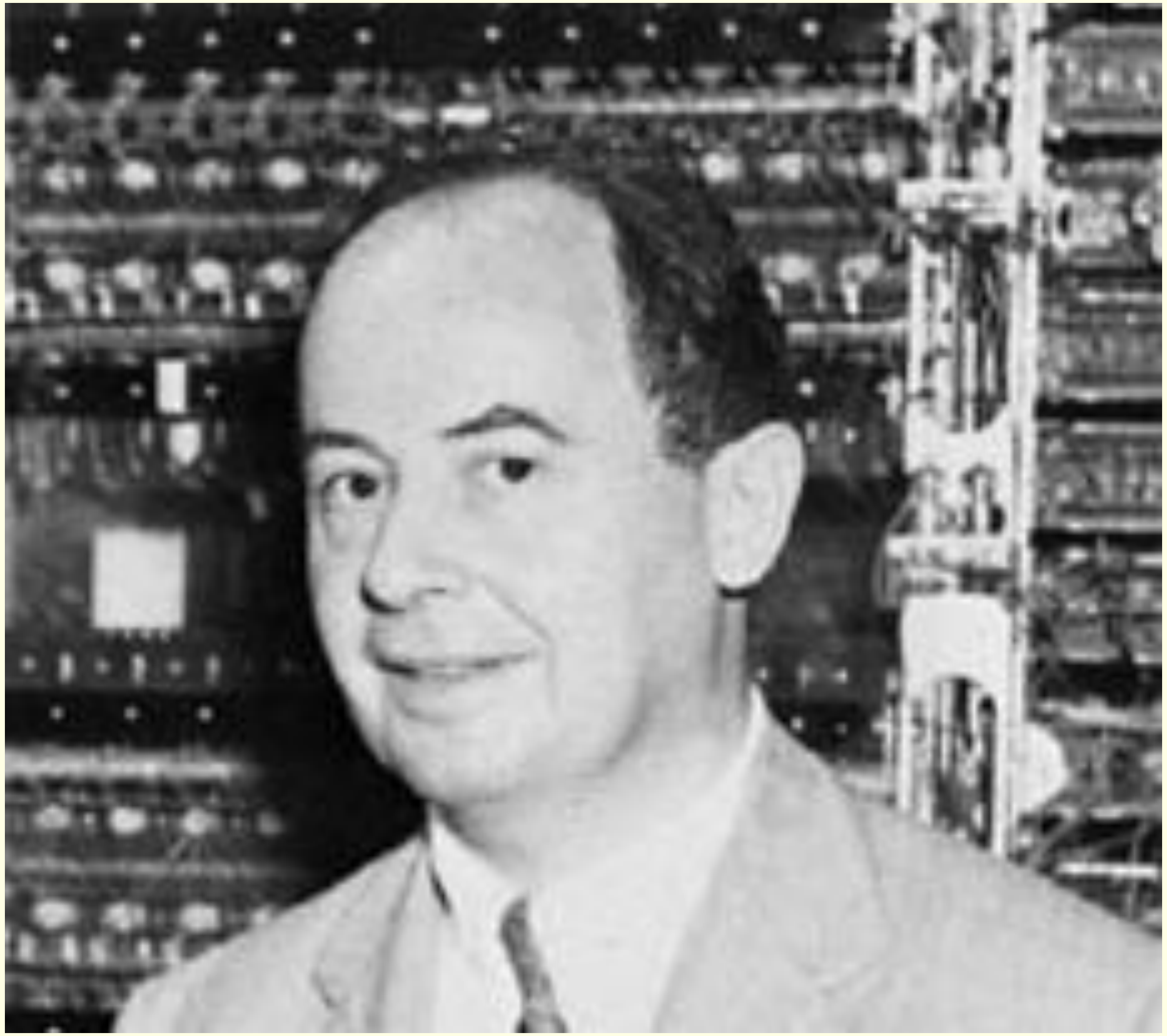


Programmable machines



- In 1936, Alan Turing proposed a device he called the 'Universal Computing Machine'.
- He proposed that a computing machine could be designed that could be 'programmed' by giving it instructions on punched tape.
- It was around this time that the work of engineer Tommy Flowers meant that computing devices were becoming electrical instead of mechanical machines.

Stored programs

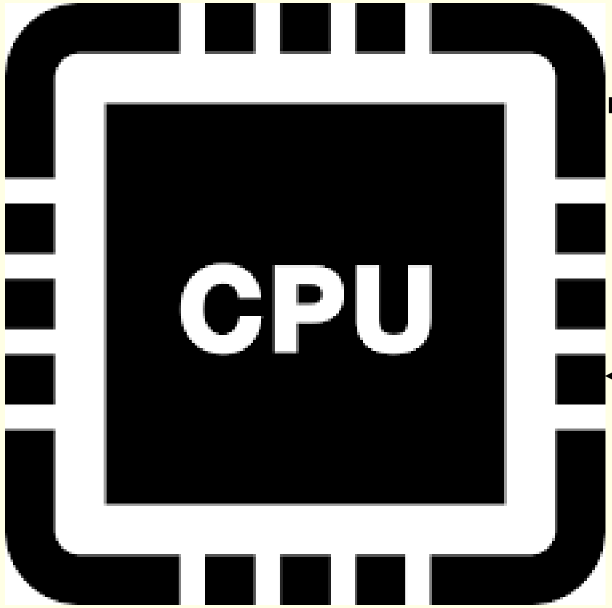


- The next step was to move programs from tape and store them electronically within a computer.
- In 1945, the mathematician and physicist John von Neumann, working at the University of Princeton USA, published a paper about the computer he and his team had designed and built.
- It was the first computer that used the basic component architecture we recognise in modern computers.
- **He identified that data and instructions could be stored in the same memory known as the stored program concept**

Stored Program Concept

Storing instructions and data in same main memory
Instructions are then fetched from memory, decoded and executed by the processor.

Storing Instructions and data



Memory Address	
0001	LOAD 0101
0010	ADD 0110
0011	STORE 0111
0100	STOP
0101	23
0110	12
0111	

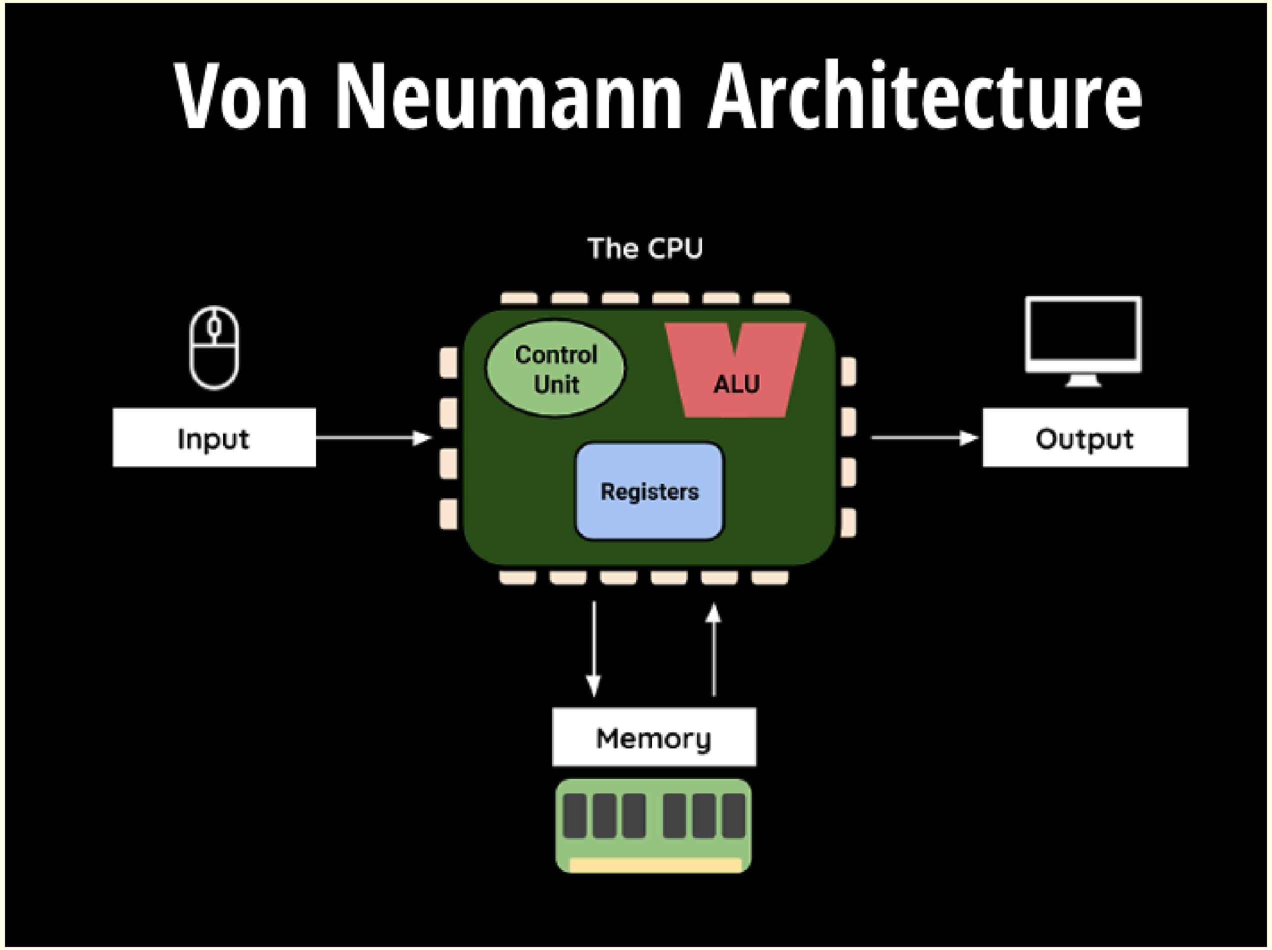
Main Memory

Inside the CPU

CPU stands for **central processing unit.**

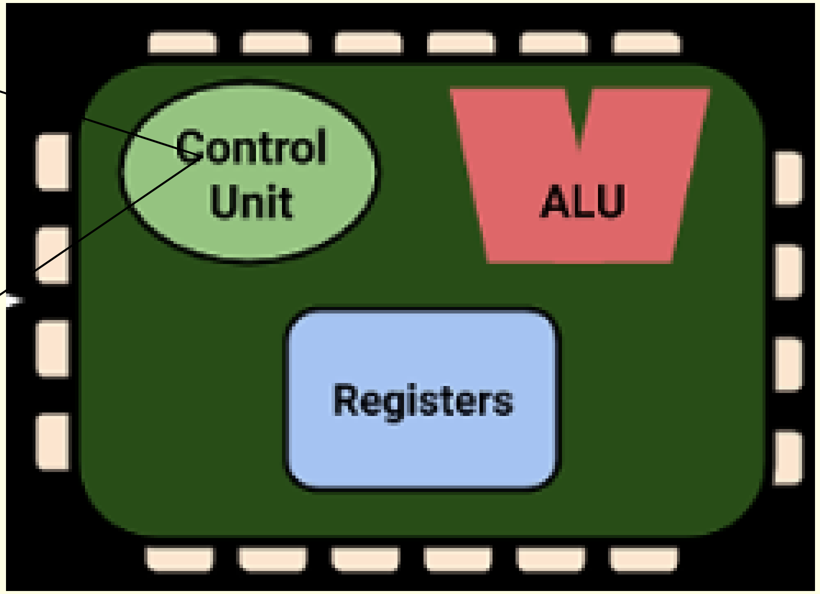
Inside of the CPU, there are a number of key components:

1. **Control unit**
2. **Arithmetic logic unit**
3. **Buses**
4. **Registers**
5. **Clock**

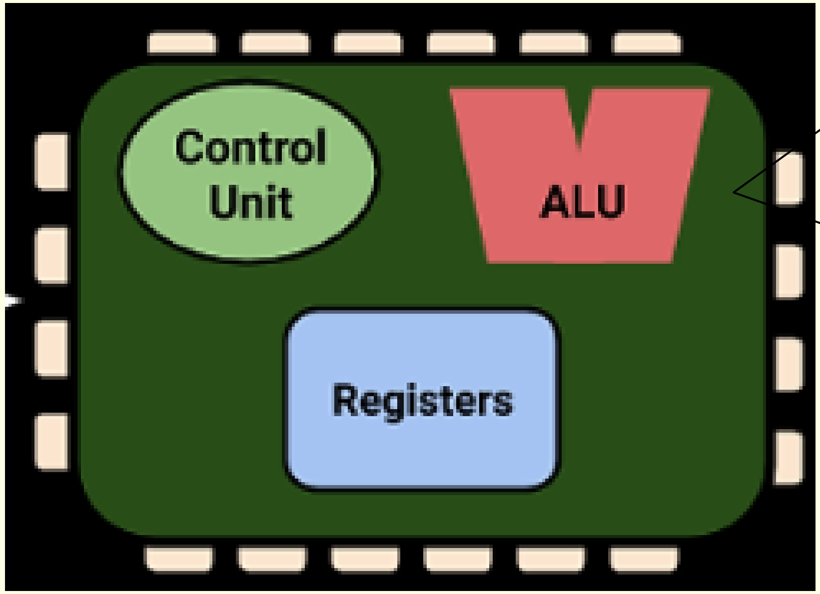


Control Unit

1. **Decodes instructions and manages the execution of instructions**
2. **The Control Unit (CU) manages the flow of data and instructions in the CPU.**
3. **It sends control signals to make sure the right parts of the computer do the right job at the right time.**



Arithmetic Logic Unit - ALU



Carries out arithmetic calculations

`5 + 6 = 11`

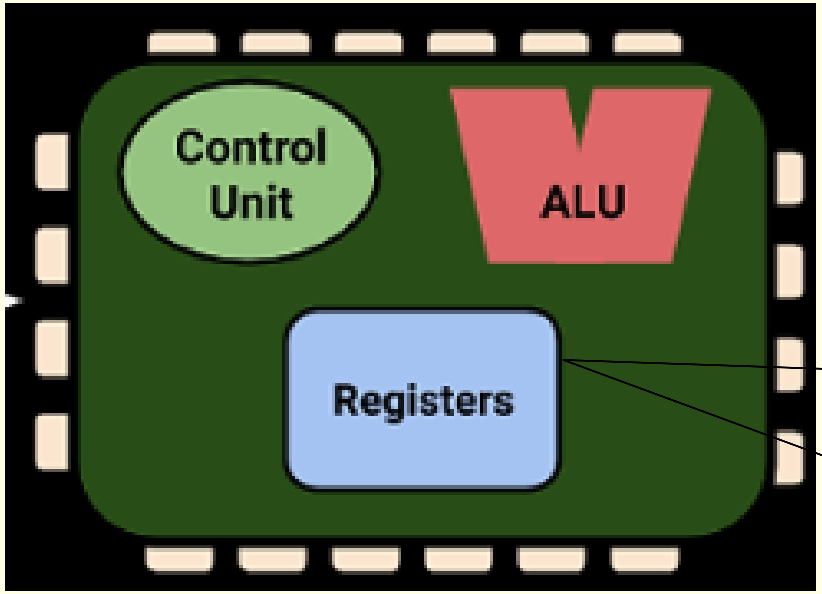
- Addition
- Subtraction
- Shifts (multiplication and division)

Carries out logical operations

- AND
- OR
- NOT
- Less Than <
- Greater Than >

`if x > y`

Registers

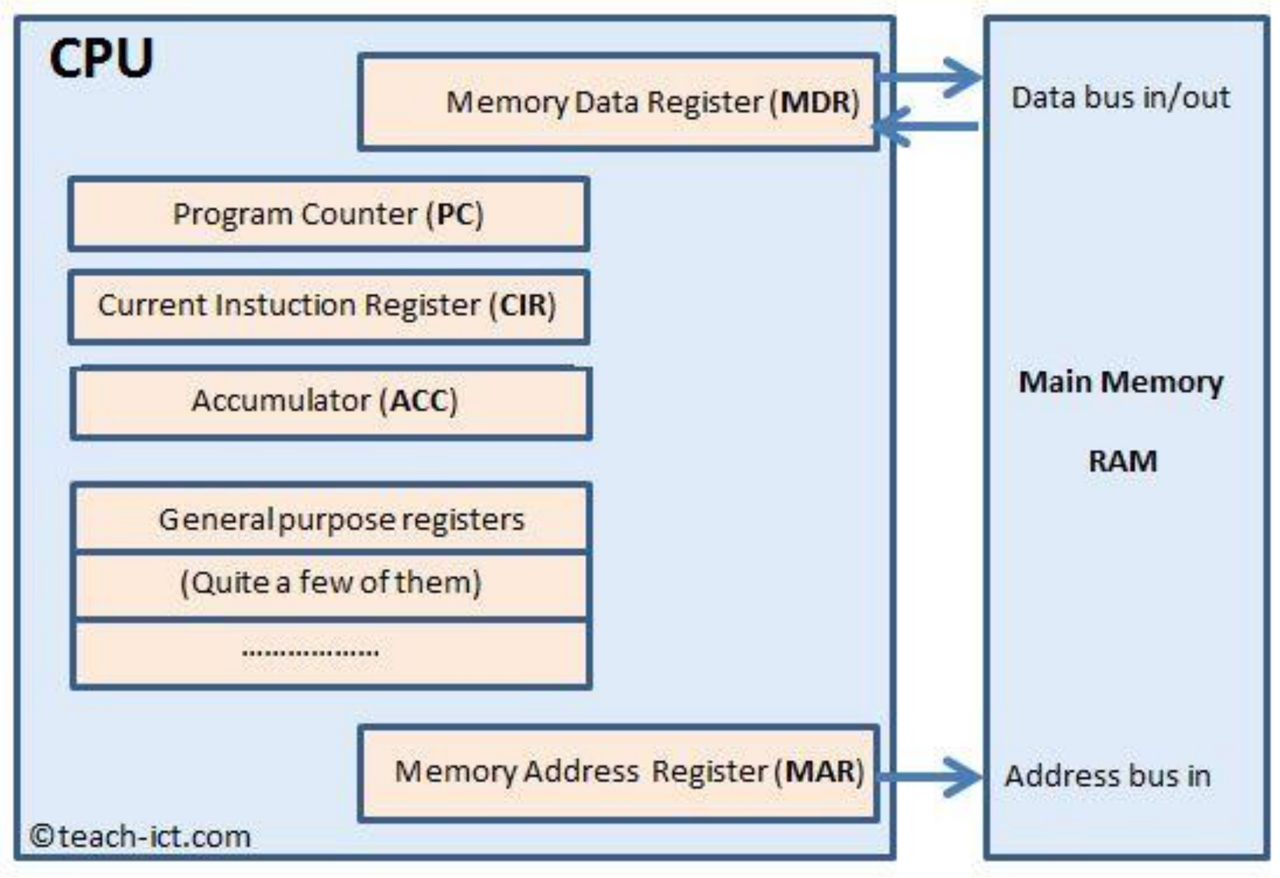


Registers

Registers are small areas of memory in the CPU

CPU Registers - Registers are small areas of memory in the CPU

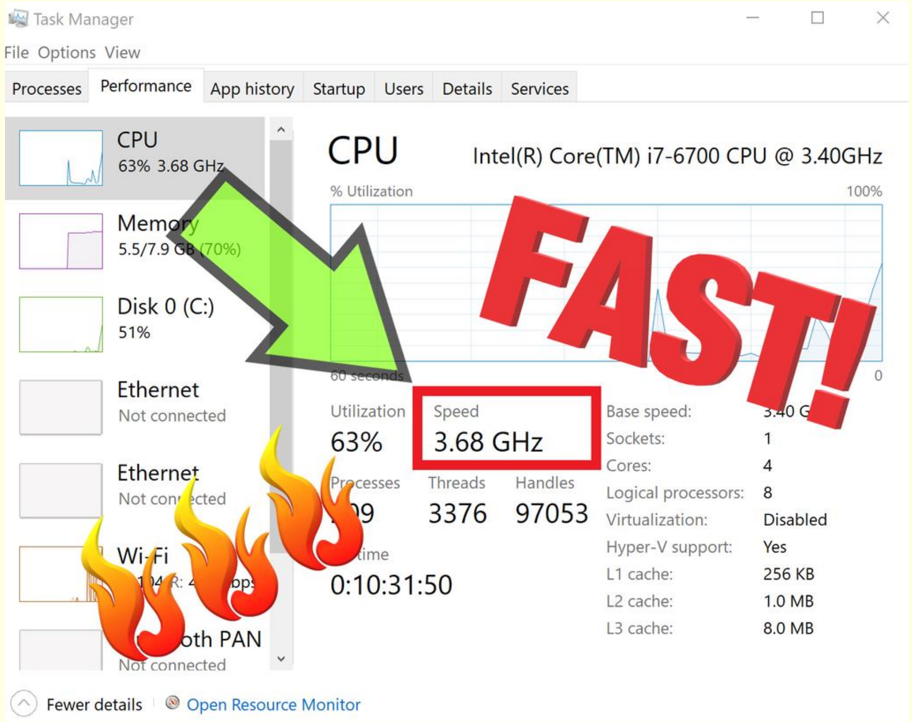
- **Program Counter** – keeps track of where the CPU is in the program. Points to the next instruction in the cycle.
- **Memory Address Register (MAR)** – holds the address of the instruction to be fetched.
- **Memory Data Register (MDR)** – stores the instruction about to be executed.
- **Accumulator** – stores the most recent result of processing.
- **Current instruction register (CIR)** - instruction to be executed



You do not need to learn the names or purposes of individual registers.

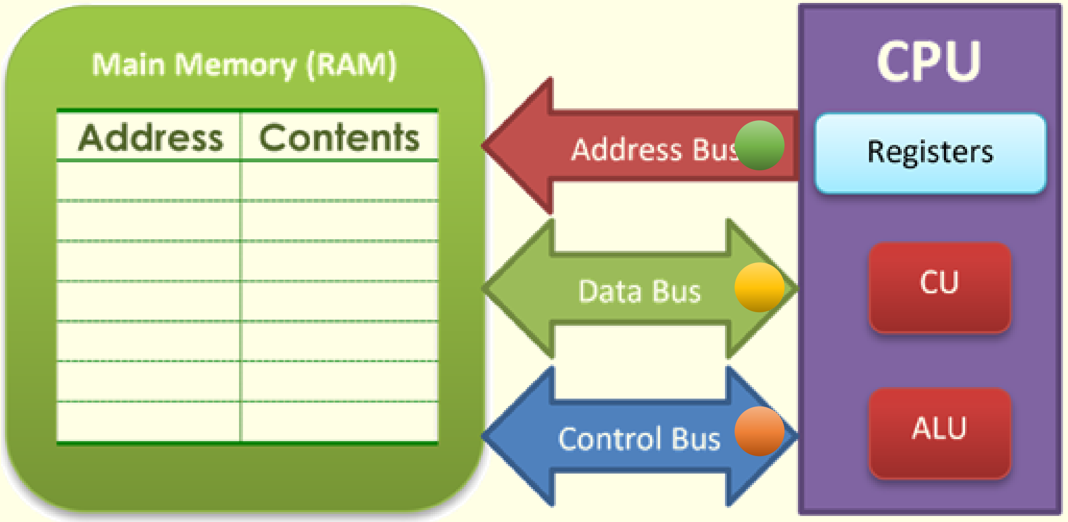
System Clock

- A clock determines the speed of the CPU.
- Regular electrical pulse which synchronises all the components.
- **The faster the clock the more instructions can be executed per second**
- The speed of the clock is measured in **Hertz** (Hz), which is the **amount of cycles per second**.
- **A clock speed of 500Hz means 500 cycles per second.**
- Current computers have clock speeds of **3GHz**, which means **3-billion cycles per second.**
- Each 'tick' means that one part of the fetch-decode-execute cycle can be carried out.



System Bus

- **address bus**
  - Used to send a memory address uni-directional
- **data bus**
  - Used to send instructions and data
  - bi-directional
- **control bus**
  - Used to send command signals to different components and receive status messages of devices.
  - bi-directional

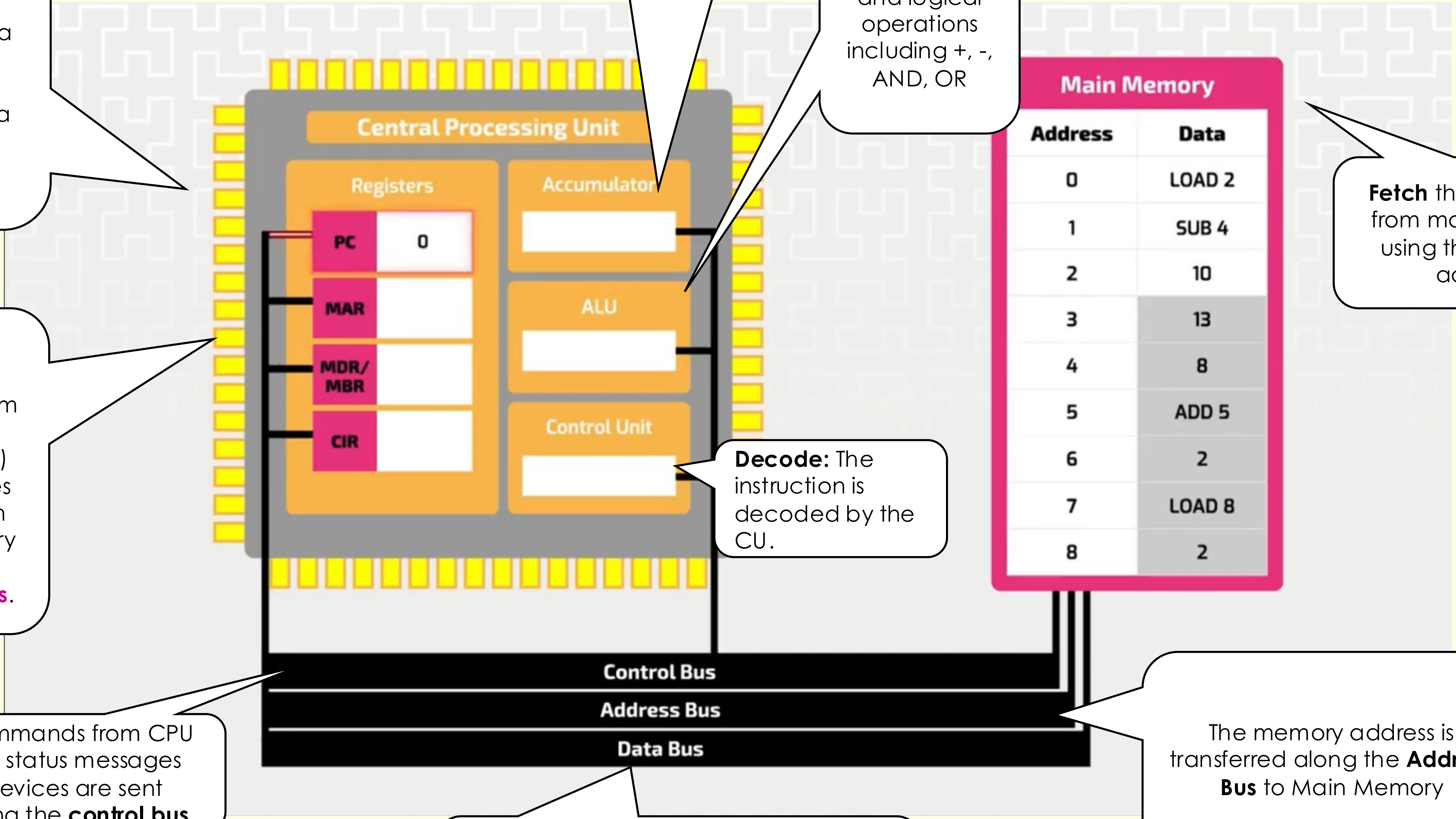


Execute: Then the next steps will be performed by the processor, for example: perform a calculation, load data from main memory, store data into main memory, output/input data.

Results of calculations are placed into the Accumulator

Performs arithmetic and logical operations including +, -, AND, OR

Fetch the instruction from main. memory using the memory address



CPU fetches data and instructions from the main memory (RAM) and then stores them in its own temporary, very fast memory called **registers**.

Decode: The instruction is decoded by the CU.

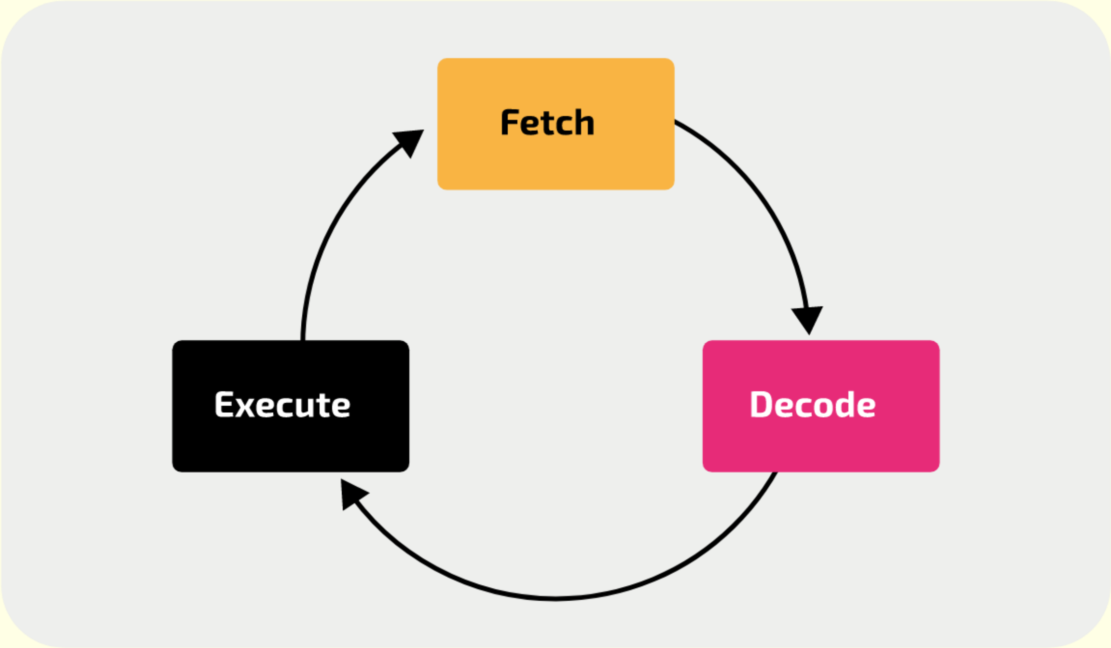
Commands from CPU and status messages of devices are sent along the **control bus**

The memory address is transferred along the **Address Bus** to Main Memory

The instruction/data is transferred back to the processor along the **data bus**

Learning Aims

- Be able to describe the **FDE cycle**



Fetch next instruction from memory
Decode instruction
Execute the instruction

Key Terms

Fetch
Decode
Execute
Data
Instructions
Memory Address
Bus Width

Recap - Fill in the gaps

The manages the execution of instructions.

The carries out calculations

are small areas of memory in the CPU

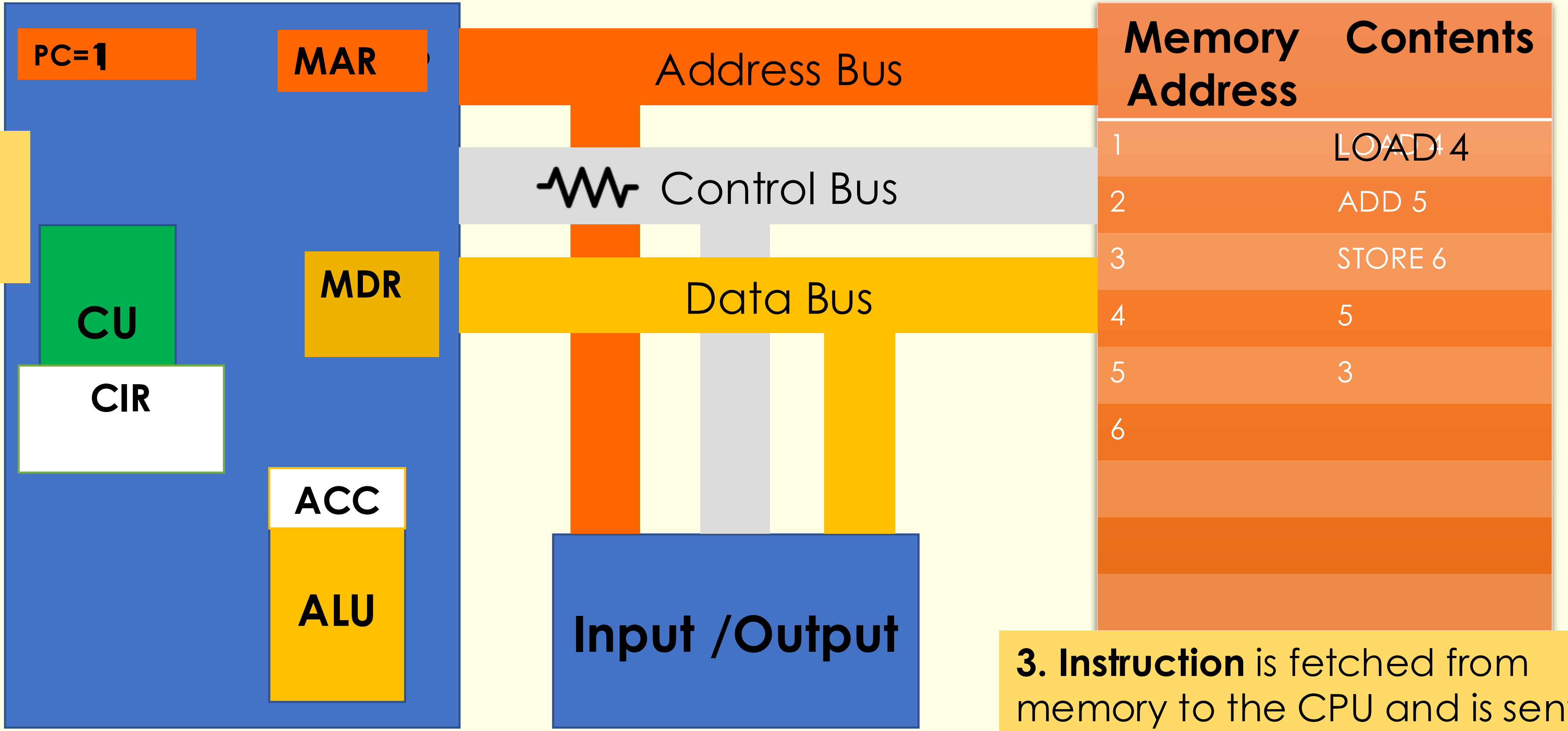
Instructions and data are from main memory

Fetching **Load 4** instruction

1. Address of the next instruction to be fetched is sent from the CPU to main memory across the **address bus.**

2. Control unit sends a signal along the **control bus** to start a **read** operation.

4. Increase **program counter** by 1



3. Instruction is fetched from memory to the CPU and is sent along the **data bus**

Instruction Set

- Every processor has a unique instruction set.

- The instruction set is the full list of operations that a microprocessor chip can carry out.

Example of an instruction set:

Mnemonic	Action
LDA	Loads a value from a memory address
STA	Stores a value in a memory address
ADD	Adds the value held in a memory address to the value held in the accumulator
SUB	Subtracts from the accumulator the value held in a memory address
MOV	Moves the contents of one memory address to another

Simple program loaded into memory

Memory Address	Contents
1	LOAD 4
2	ADD 5
3	STORE 6
4	5
5	3
6	

Simple program loaded into memory to add 2 numbers together

1. LOAD Data from Memory Address 4 (5)
2. ADD Data from Memory Address 5 (3 + 5)
3. STORE Result into Memory Address 6 (8)

The control unit also sends command signals

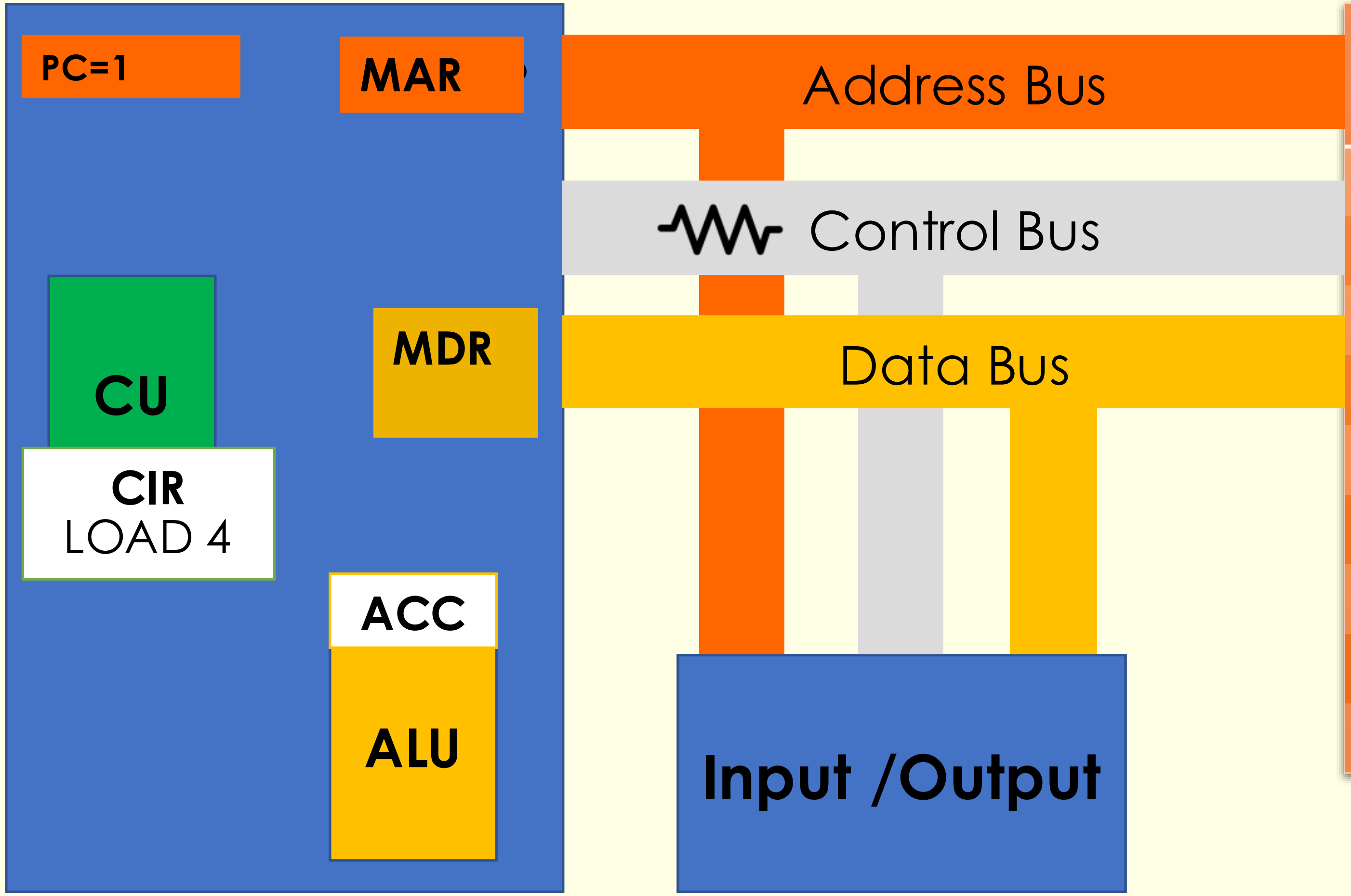
For example:

Memory Read: causes data from the addressed location to be placed on the data bus

Memory Write: causes data on the data bus to be written into the addressed location

Decode Load 4

5. The instruction is decoded by the **Control Unit**.
The CPU is designed to understand a specific set of commands. These are called the '**<u>instruction set</u>**' of the CPU.



Memory Address	Contents
1	LOAD 4
2	ADD 5
3	STORE 6
4	5
5	3
6	

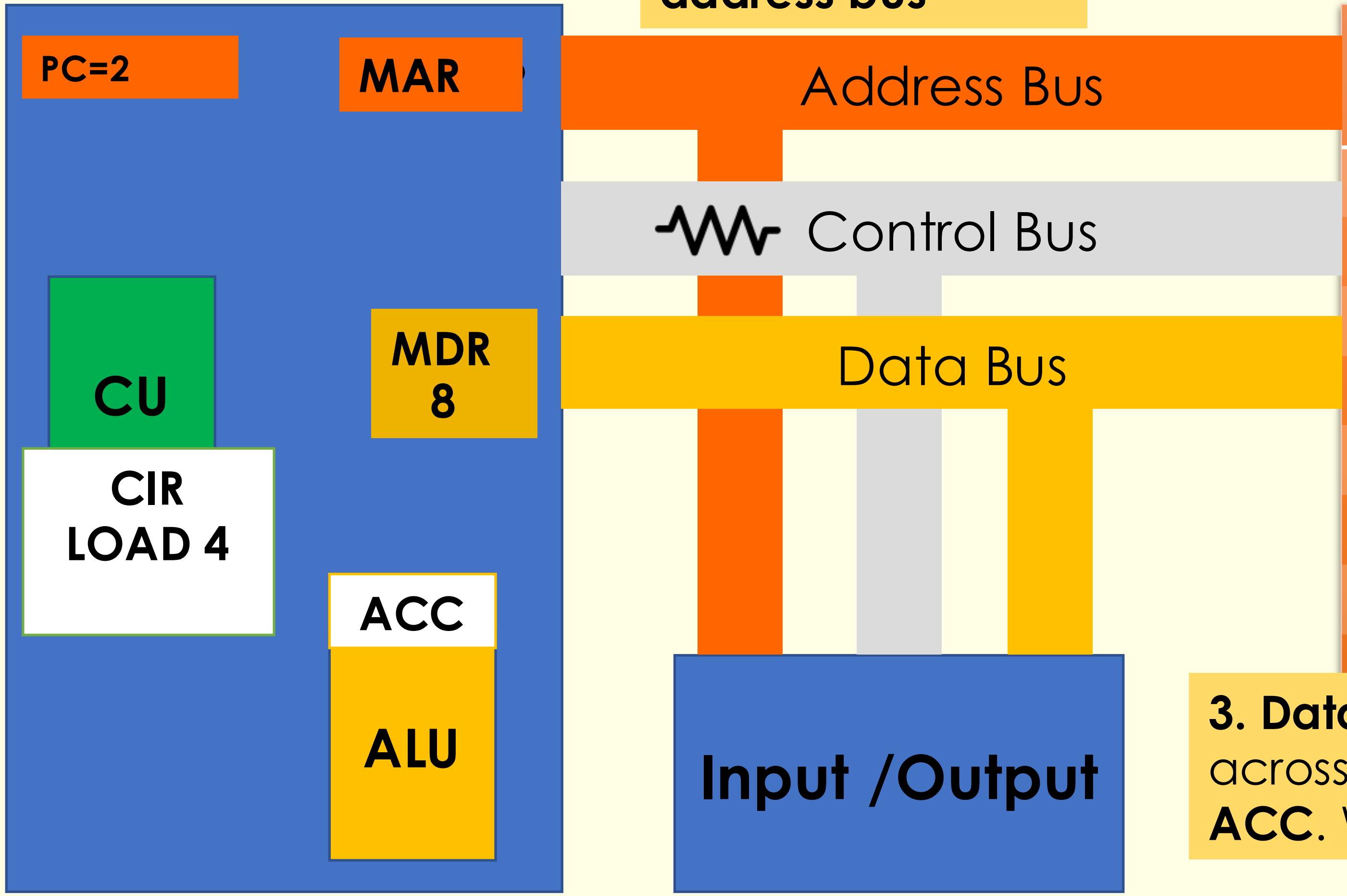
Executing **LOAD 4** instruction

1. Control unit sends a signal on the **control bus** to start a **READ** operation

2. Address 4 is placed on the **address bus**

6 The instruction is executed
This could be:

- **Load data** from memory
- **Write data** to memory
- Do a **calculation** or logic operation using the ALU
- Change the address in the PC
- Halt the program



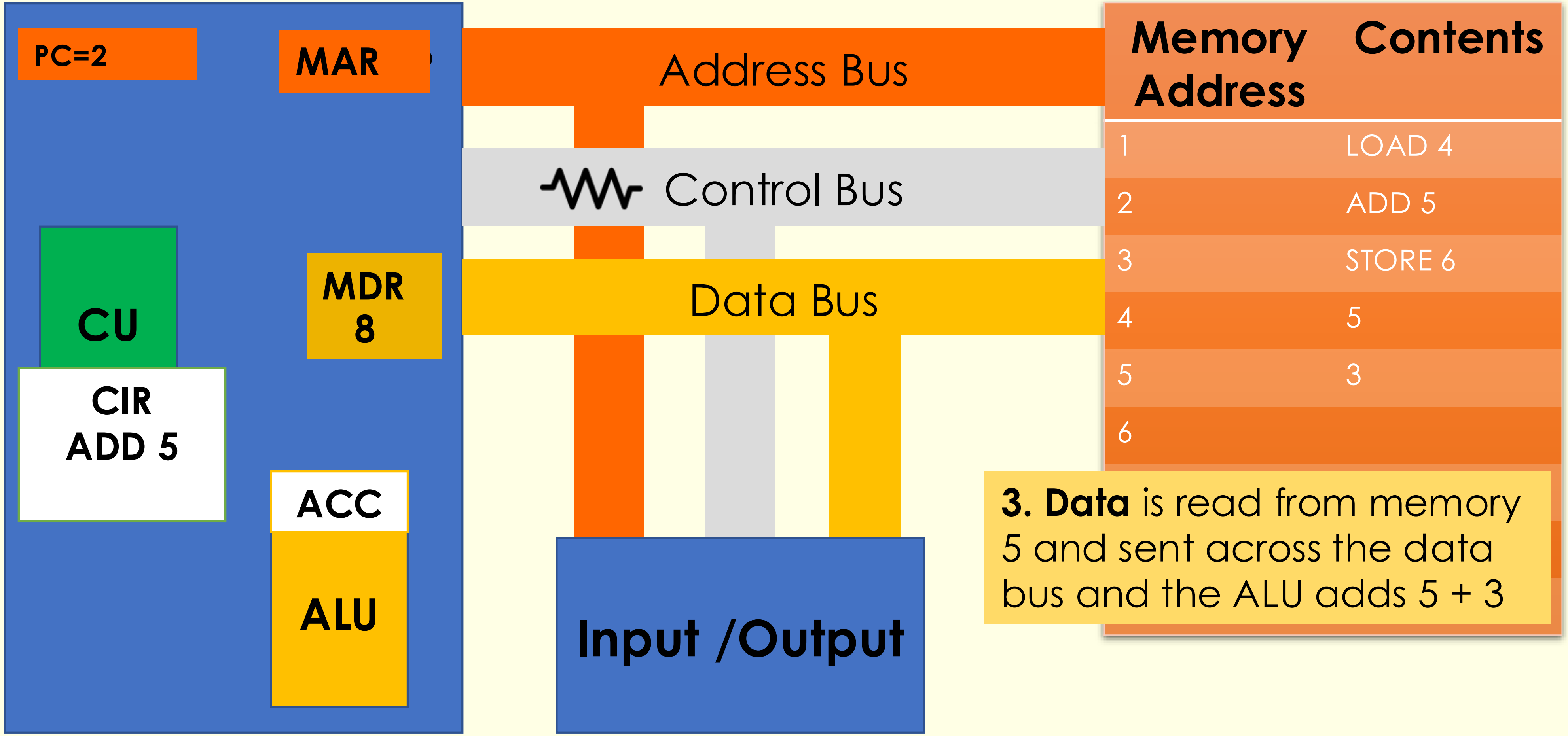
Memory Address	Contents
1	LOAD 4
2	ADD 5
3	STORE 6
4	5
5	3
6	

3. Data is read from memory and is sent across the data bus and placed on the **ACC**. Waiting to be used by the ALU

Executing **ADD 5** instruction

1. Control unit sends a signal on the **control bus** to start a **READ** operation

2. Address 5 is placed on the **address bus**



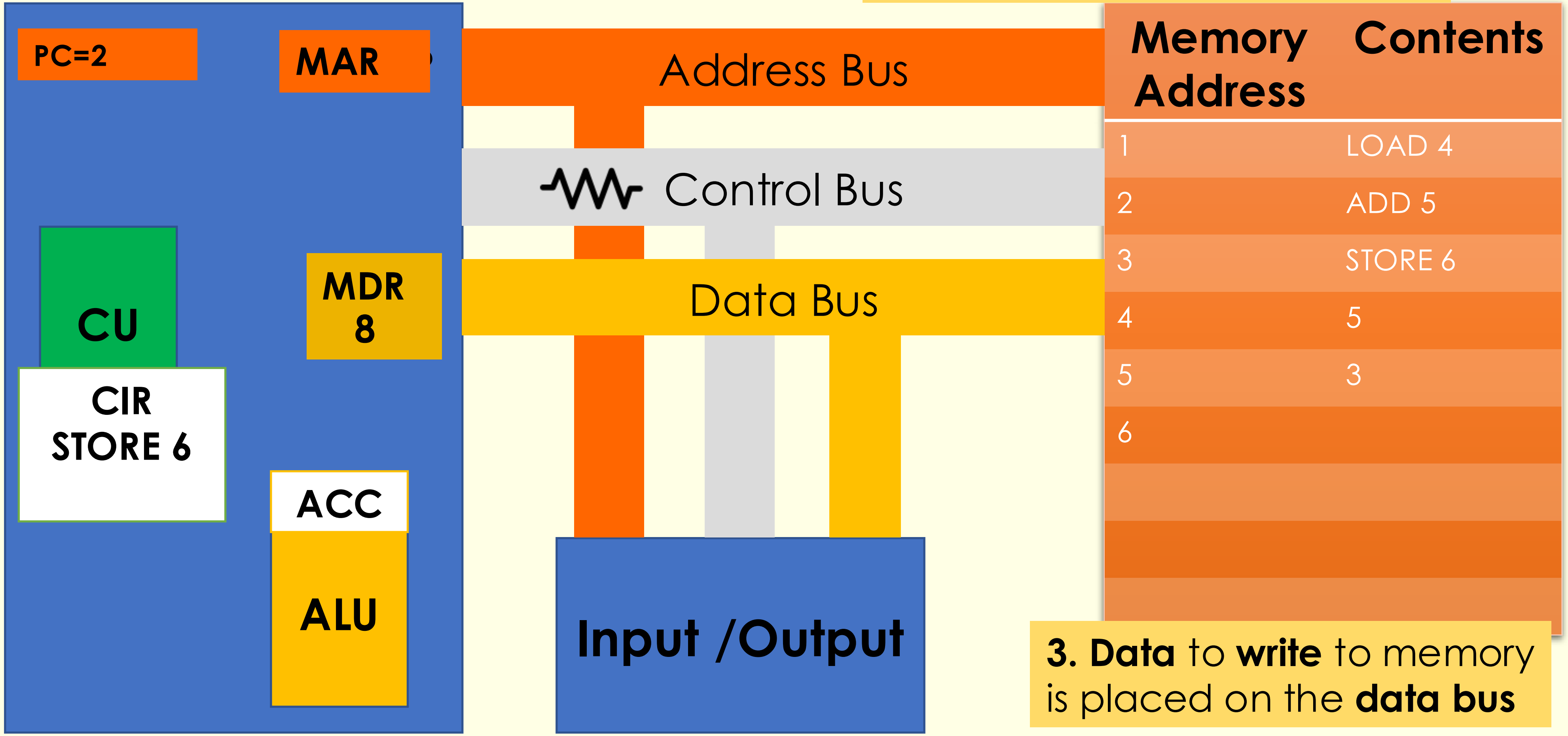
3. Data is read from memory 5 and sent across the data bus and the ALU adds 5 + 3

Executing **STORE 6** instruction

STORE 5 is an instruction used to write data to memory.

1. Control unit sends a signal on the **control bus** to start a **write** operation

2. Address of memory (to store data) is placed on the **address bus**



3. Data to **write** to memory is placed on the **data bus**

FDE

Fetch Instruction

1. Memory address is copied from a CPU register(Memory Address Register) and sent along the **address bus** to main memory
2. Instructions are fetched from the memory address and sent along the **data bus** to the CPU.
3. Instruction is copied into registers (Memory data register and moved to Current Address Register)
4. Increase program counter by 1

Decode Instruction

5. The instruction is decoded by the **Control Unit**.
The CPU is designed to understand a specific set of commands. These are called the '**<u>instruction set</u>**' of the CPU.

Execute Instruction

6. The instruction is executed
This could be:

- Load data from memory
- Write data to memory
- Do a calculation or logic operation using the ALU
- Change the address in the PC
- Halt the program

Clock

Clock sends electrical signals at regular intervals so that operations can be synchronized

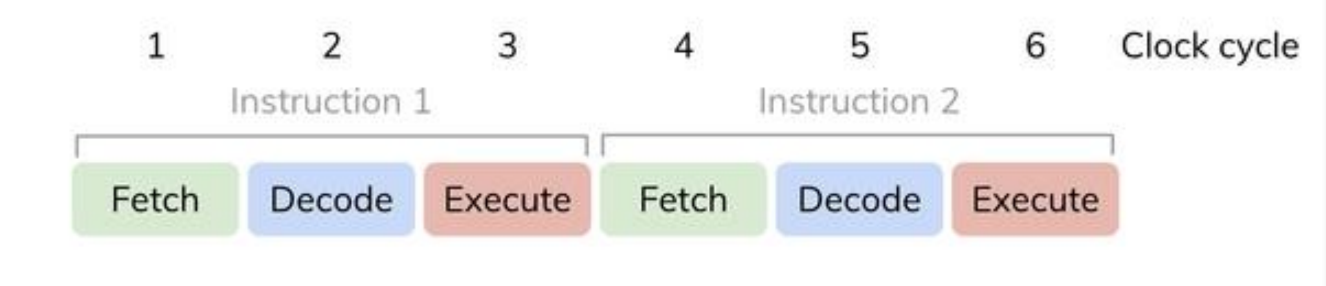
The **clock cycle** is a single tick of the computer's clock

Each clock cycle triggers the CPU to perform a basic operation, such as fetching an instruction, decoding it, or executing it.

The speed of a CPU is often measured in clock cycles per second, known as **clock speed** or frequency, typically measured in gigahertz (GHz).

A higher clock speed means the CPU can process more instructions in a given time, leading to faster performance.

A 3GHZ – 3 billion F-D-E cycles per second



Recap System Bus

Buses

- **address bus**
  - The address of memory (holding instruction) is placed on the address bus
  - **uni-directional**
- **data bus**
  - Contents (instruction/data) of the memory are placed on the data bus
  - **bi-directional**
- **control bus**
  - Control unit sends a signal on the control bus to start an operation and communicate with other devices within the computer.
  - Receive status messages of devices
  - **bi-directional**

Address Bus Width

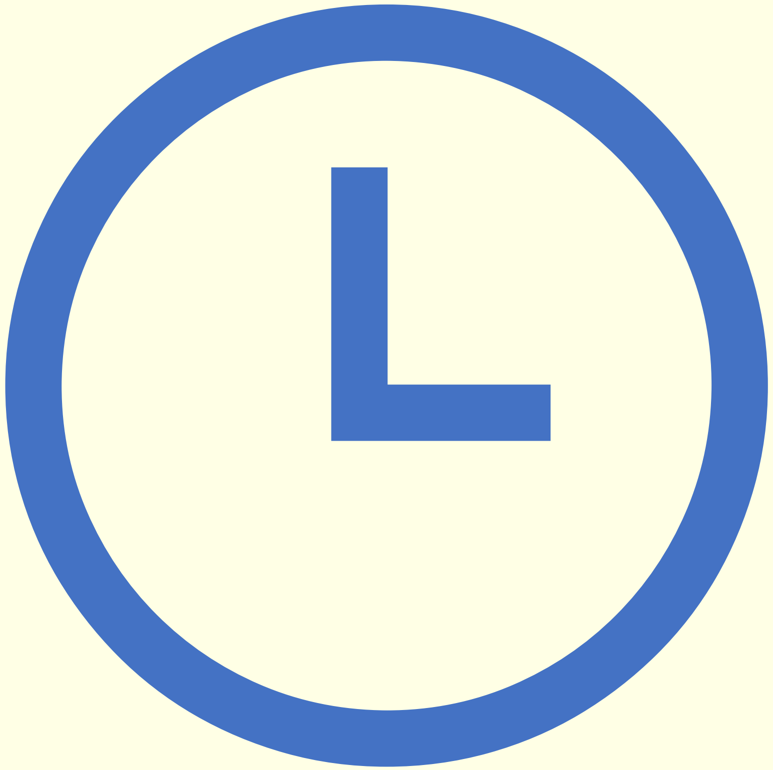
- Bus Width – number of connections on a bus
- Each connection represents a binary digit (0 or 1)
- A greater bus width means larger number values can be communicated

For example:

8 bit address bus	=	00000000 (0) to 11111111(255)	=	256 memory addresses 2^8
32 bit address bus	=	00000000 00000000 00000000 to 11111111 11111111 11111111	=	4 GiB memory addresses 2^{32} memory addresses

Learning Aims

- Describe the role of the **clock** and explain how the speed of the clock impacts on performance.
- Describe the term **pipelining** and how it relates to CPU performance

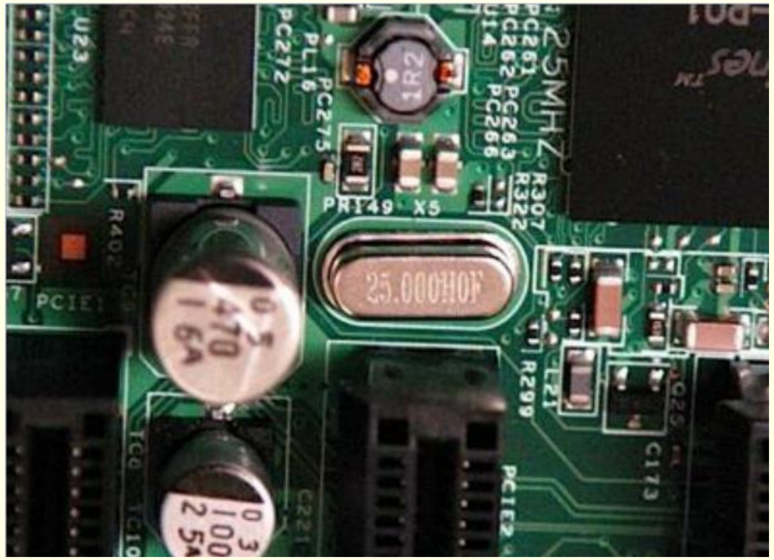


System Clock

- Using a quartz crystal, the clock in a computer breathes life into the microprocessor by feeding it a constant flow of pulses.

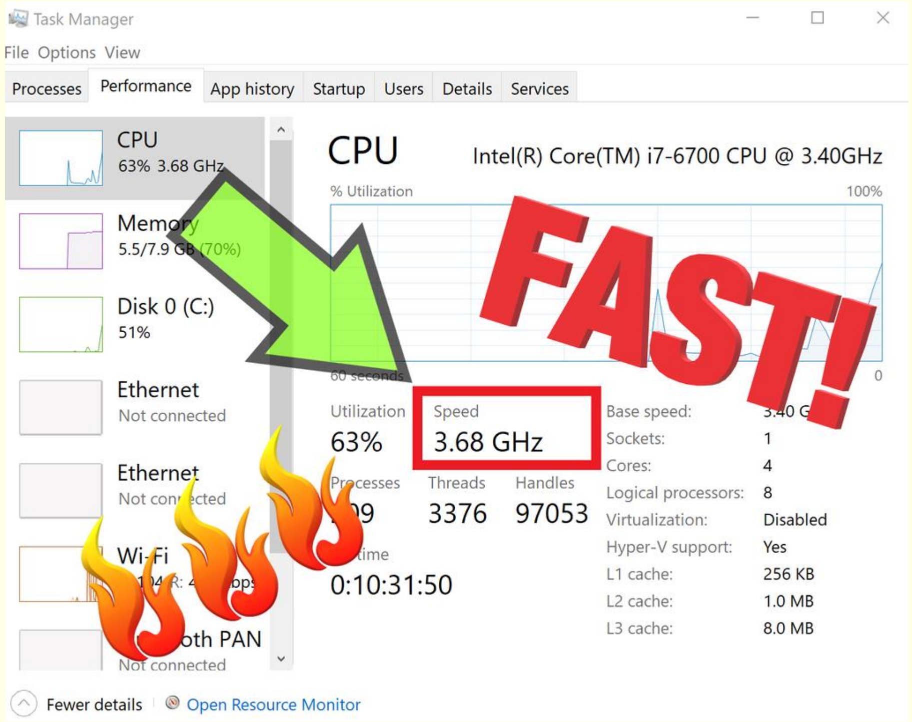
- The clock rate of a CPU is normally determined by the frequency of an oscillator crystal.

- In other words, the **frequency of electronic signals per second**



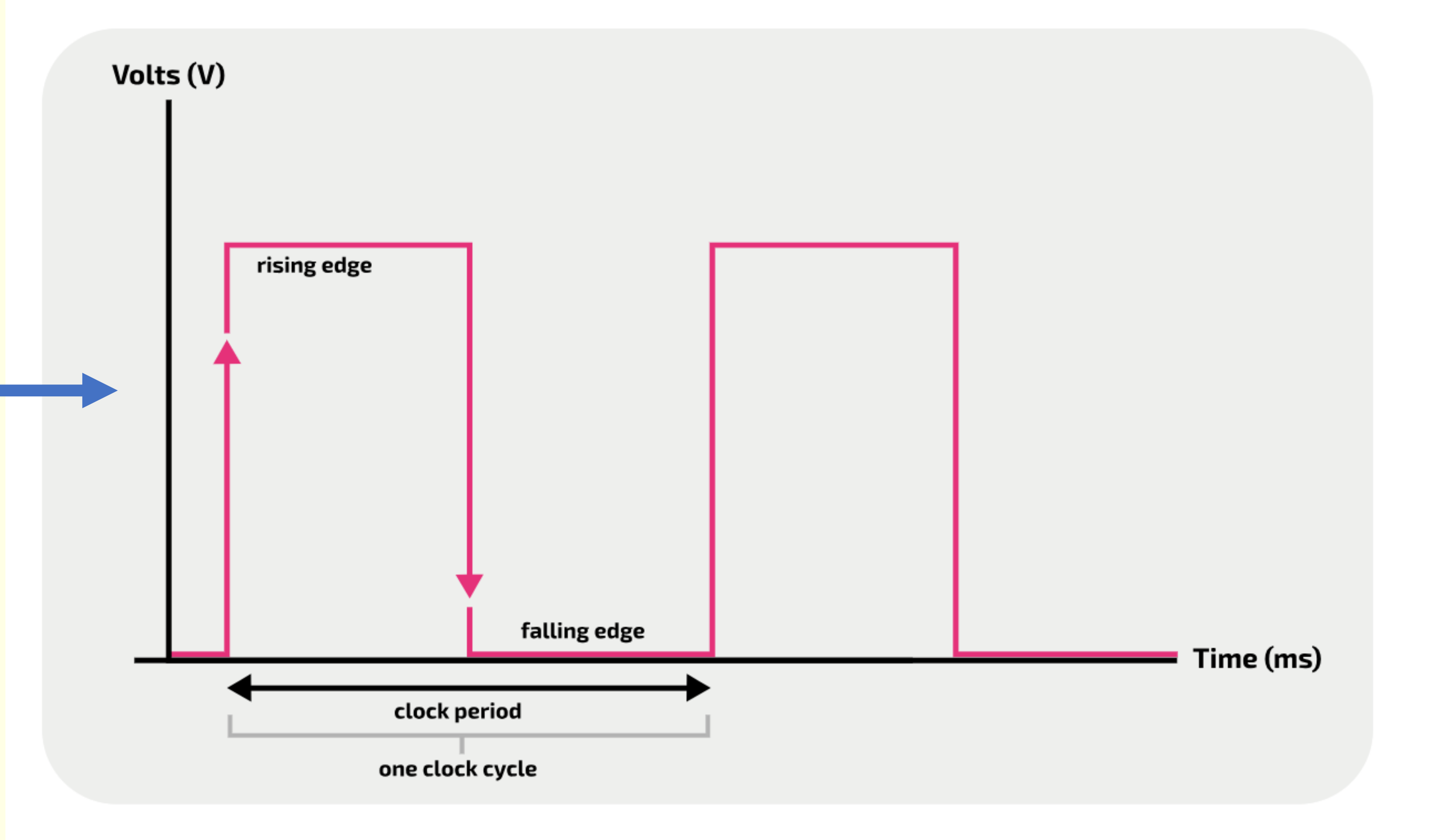
Clock

A clock determines the speed of the CPU.
The faster the clock the more instructions can be executed per second
The speed of the clock is measured in **Hertz** (Hz), which is the **amount of cycles per second.**
A clock speed of 500Hz means 500 cycles per second.
Current computers have clock speeds of **3GHz**, which means **3-billion cycles per second.**
Each 'tick' means that one part of the fetch-decode-execute cycle can be carried out.



- The **system clock** — also simply referred to as the **clock** — generates regular clock pulses by emitting a signal that continuously switches between a low (or '0') and a high (or '1') state.

On the rising edge a FDE task is carried out



The time taken between two sequential rising edges is called a **clock cycle** or a **clock period**. The **clock speed** is measured by the number of clock cycles in one second — 1 clock cycle per second is 1Hz.

Calculating Clock Speed

Herz	Measurement
1 hertz (Hz)	1 clock cycle per second
1 million Hz	megahertz (MHz)
1,000 MHz	gigahertz (GHz)

Some modern-day processors have a clock cycle of over 3GHz! Remember that 1GHz is equal to 1,000,000,000Hz — so 3GHz is 3,000,000,000Hz or 3,000,000,000 clock cycles in one second!

What does Synchronise mean?

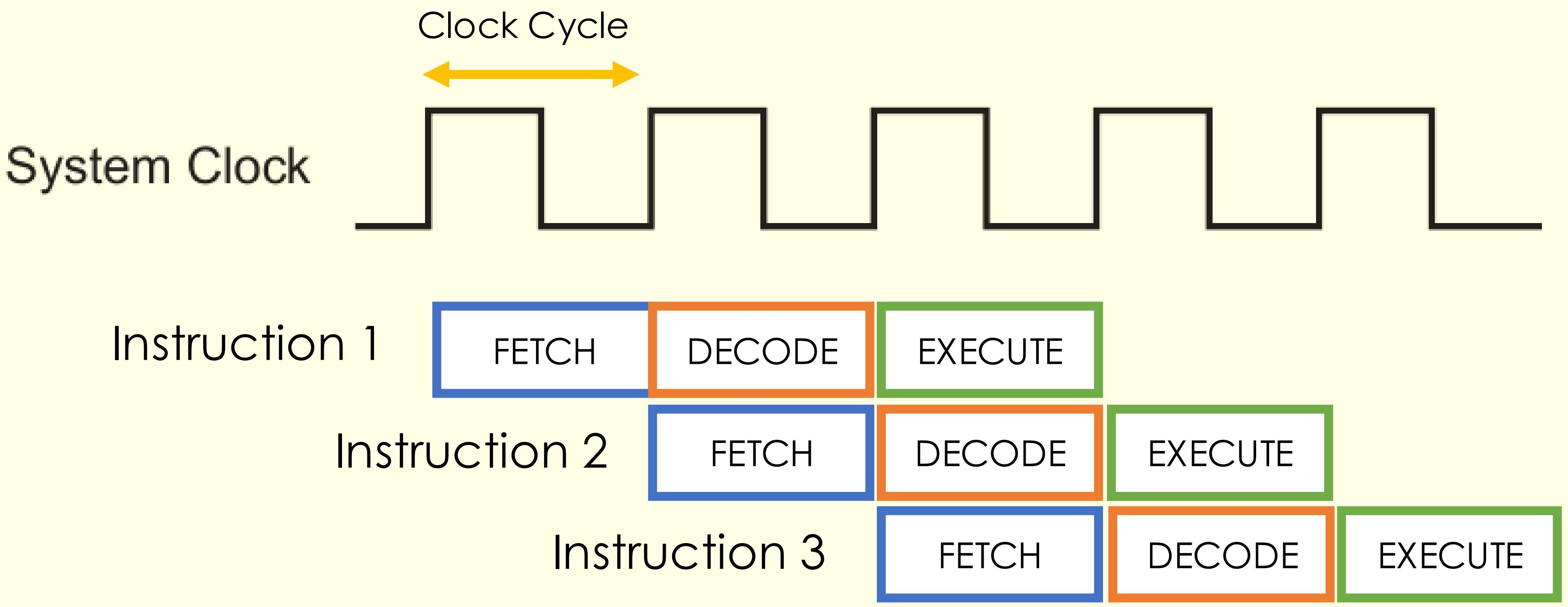
Occur at the same time

The **clock signal** is used to **synchronise** the **operations** of the processor components.

What is computer operations?

A given task carried out by the CPU during the Fetch Decode Execute cycle

FDE Clock Cycle



When instructions are processed at the same time is known as pipelining

Pipelining Video

FDE Clock Cycle

What would happen if the clock failed?

- Then the **operations** of the processor components inside the processor would **not** be **synchronised** and it would not be able to process any data.
- The computer would not work.

Pipelining

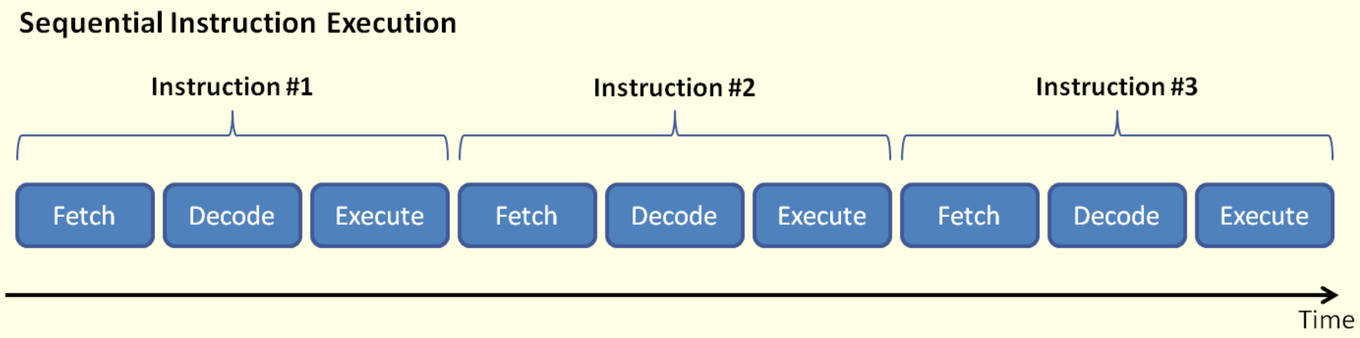
Each instruction that is carried out has several stages (fetch-decode-execute).

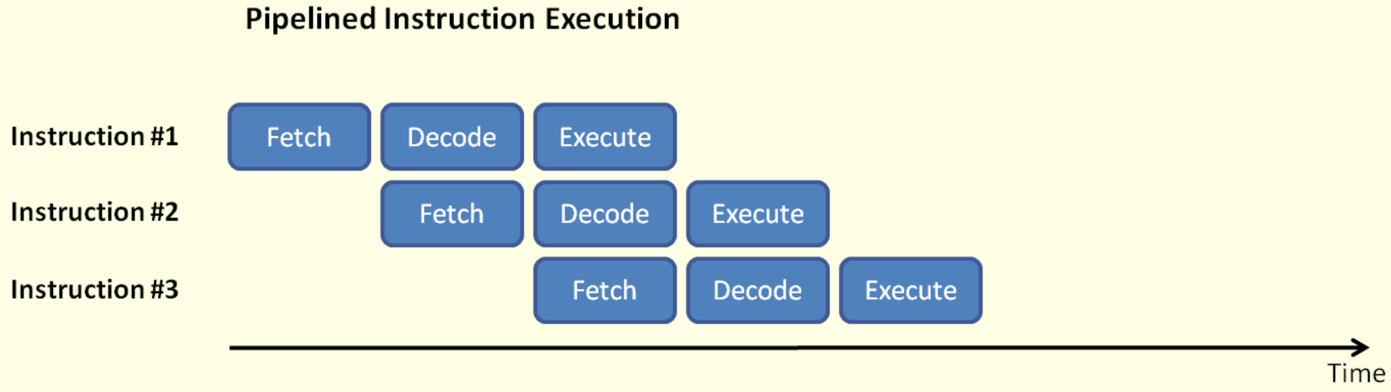
Each component of the CPU has a different job to do in this cycle.

Sometimes, one component has nothing to do whilst it is waiting for the component before it in the sequence to complete its task.

This increases the overall processing time needed.

Pipelining is a method of keeping all the components busy so as to improve overall CPU performance.





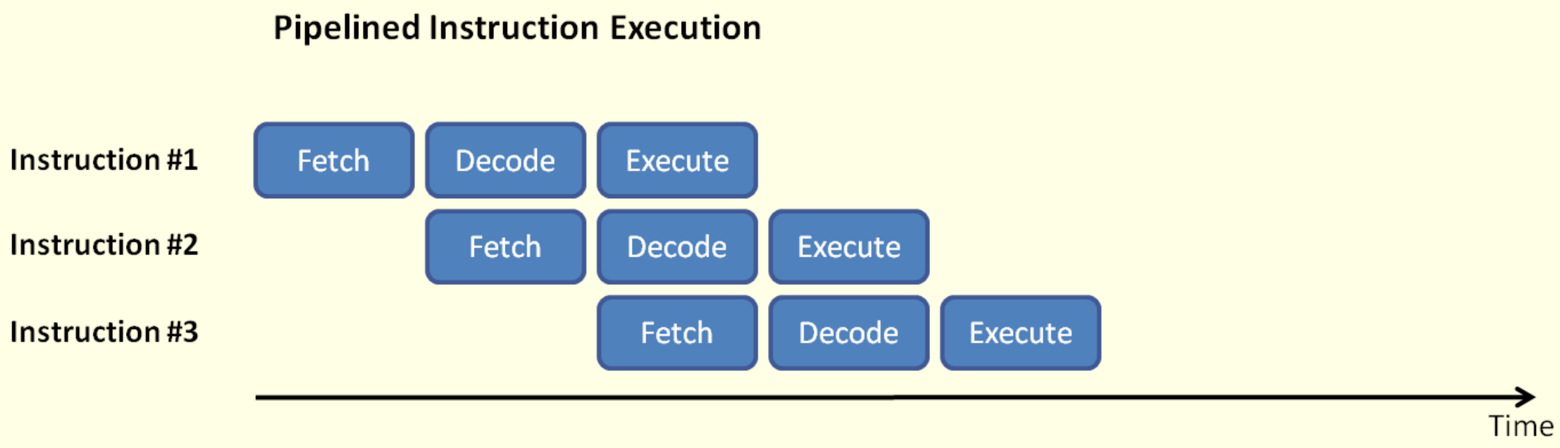
Pipelining

Pipelining attempts to keep every component of the CPU busy at all times.

Instructions flow through the CPU in stages.

This means that an instruction can be completed every clock cycle. This increases CPU **throughput** allowing overall performance to be optimised.

A CPU is called ‘fully pipelined’ if an instruction can be fetched every clock cycle.



Embedded systems

GCSE Computer Science

Lesson Objectives

Identify

Identify several embedded systems

Explain

Explain the difference between an embedded system and a general-purpose computer

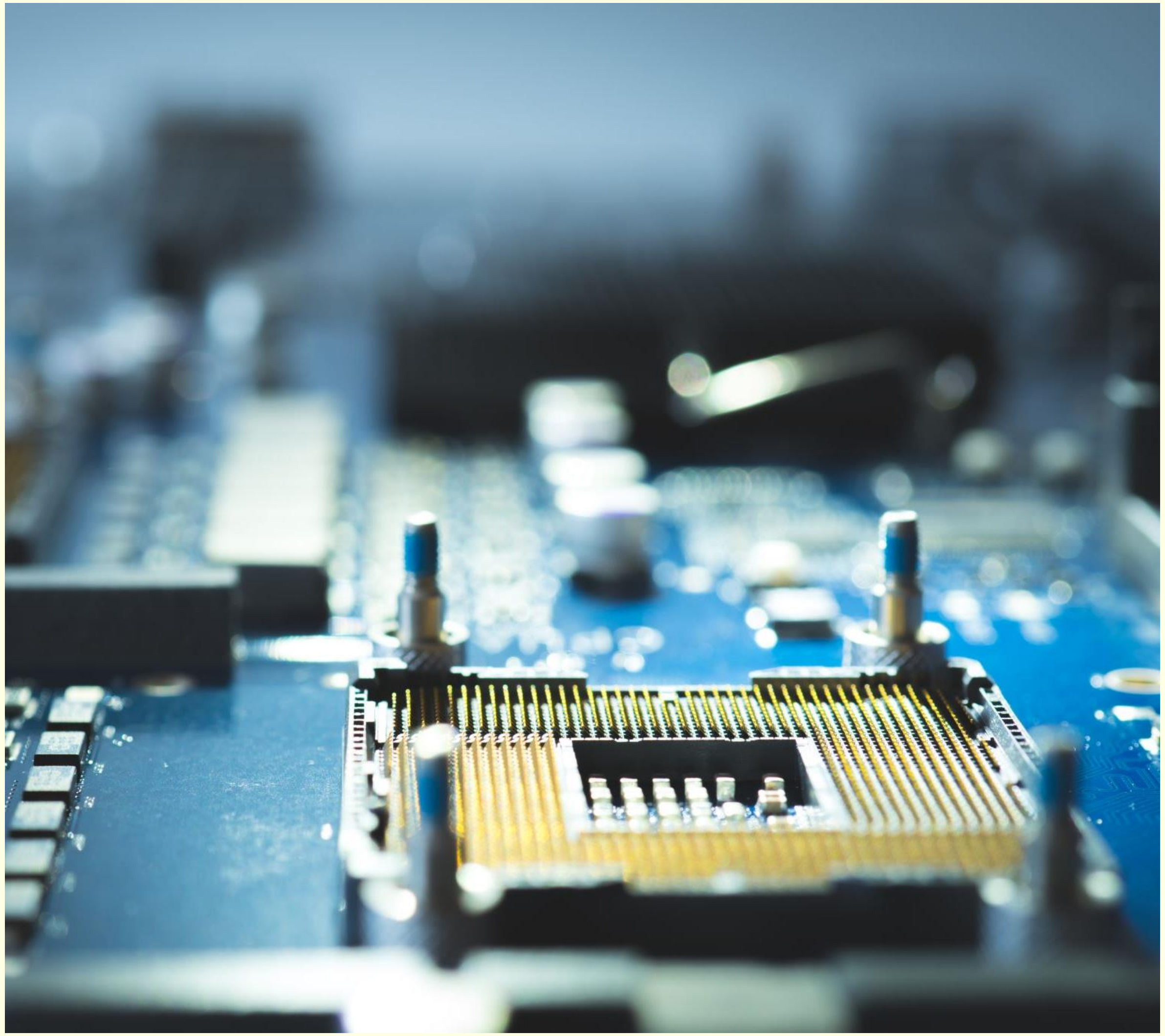
Identify

Identify the benefits of using an embedded system

Computer Systems

- Previously, you have learned what makes up a computer.
- What are the components required for a device to be considered a computer?

- Processor
- Memory
- Input and output.



General-purpose computers

- There are two general categories of computers.

- General-purpose computers are the ones that you are most familiar with.
- You would recognise them as computers.

- Devices that could be considered general purpose include desktop PCs, laptops, smartphones and tablets.

Computers that don't look like computers

The vast majority of devices that could be considered a computer don't look anything like one.


Embedded Systems
Industrial Robots
GPS
GPS Receivers
Digital Cameras
DVD Players
Wireless Routers
MP3 Players
Set top Boxes
Gaming Consoles
Photocopiers
Microwave Ovens

Embedded systems

- An embedded computer is usually a fairly simple computer, often on a single printed circuit board.
- It is usually responsible for a dedicated task.

- Traffic lights are an example. The controller is usually out of sight but it has parts that:
- switch the lights
- keep track of time in each state
- receive input from the pedestrian button
- receive input from the road sensors, etc.

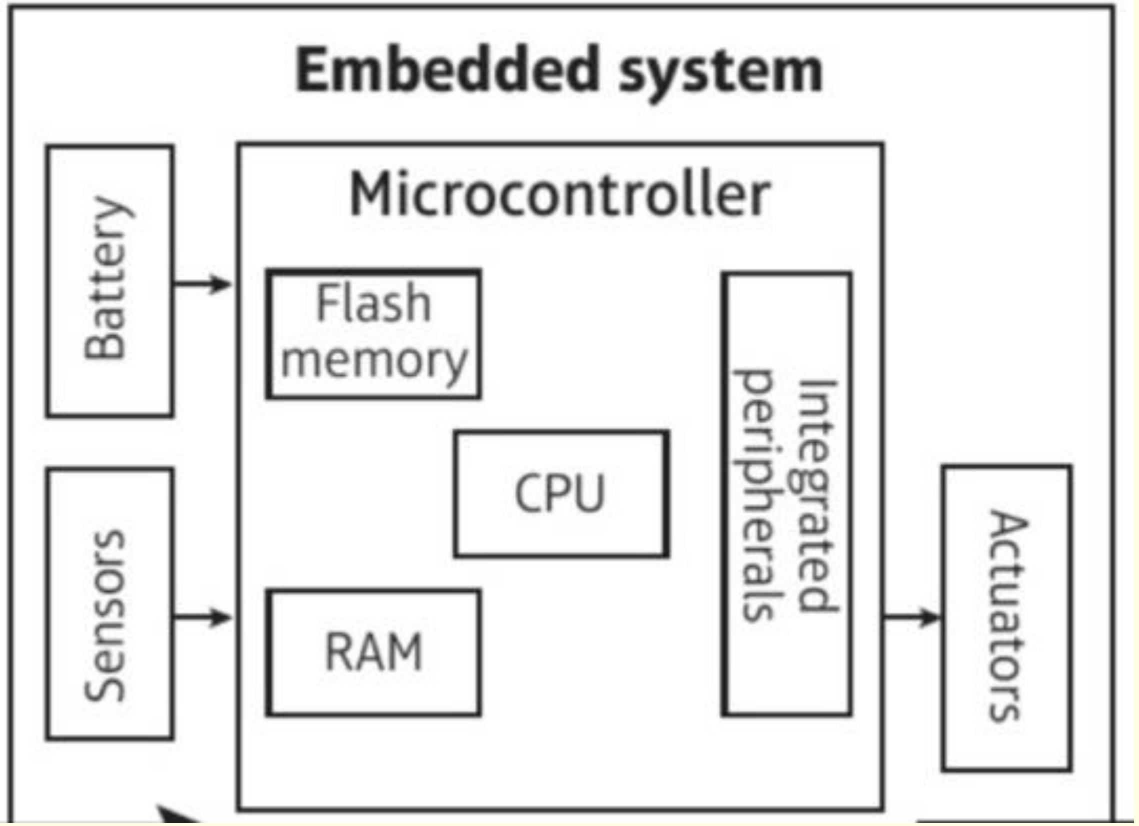
Microcontroller

All the components are all on a single chip – known as a microcontroller.

Senor readings (temperature sensor reading) are input into the system and stored in RAM.

CPU interprets and acts on the data according to the program instructions stored in flash memory.

A signal is sent to an output device, such as a actuator (switch cooling unit on and off).



Characteristics of an embedded system

- Task specific (single purpose)
- Low power consumption
- Small physical size
- Low maintenance
- Low cost
- Often real-time or responsive to their environment.

embedded within a washing machine. What are its inputs and outputs?

Inputs

- Heat setting
- Time setting
- Spin setting
- Wash-type selection

Control Unit

Specific functions are:

- Temperature control
- Time control
- Spin speed
- Drainage control
- Additional settings for garment type (sportswear, baby clothes, woollen, etc.)
- Door lock
- Child lock settings

Outputs

- Display settings
- Lights for selected options
- Warning and completion beeps

Software in an embedded system

- All computing devices need some kind of program to control them.

- Generally, embedded computers have a small storage device to store their simple programs. As most embedded computers are single-purpose, their programs are short and simple, relying only on the fixed inputs and outputs of the system.

- This means that programs written for embedded systems are often very small and they do not need to be modified once installed on the device. This is often known as **firmware**.

Hardware in an embedded system

GPS

Touch screen

Pedometer

Heart rate
monitor

Motherboard

Vibration generator

Accelerometer

Bluetooth
receiver/transmitter

NFC chip

Battery

Using an embedded system

Advantages:

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

Disadvantages:

- **Difficult to program** and requires specialist knowledge
- If a problem occurs with the system it is **difficult to repair**
- Difficult to upgrade
- If any problem occurs then you need to reset settings
- Troubleshooting is difficult