

## Lesson

## 4

## Architecture & Operations of CPU & Main Storage

A computer system is composed of both internal and external components. We are going to look at the computer's internal components and how programs and data are stored.

The internal components are contained in the **Central Processing Unit (CPU)**.

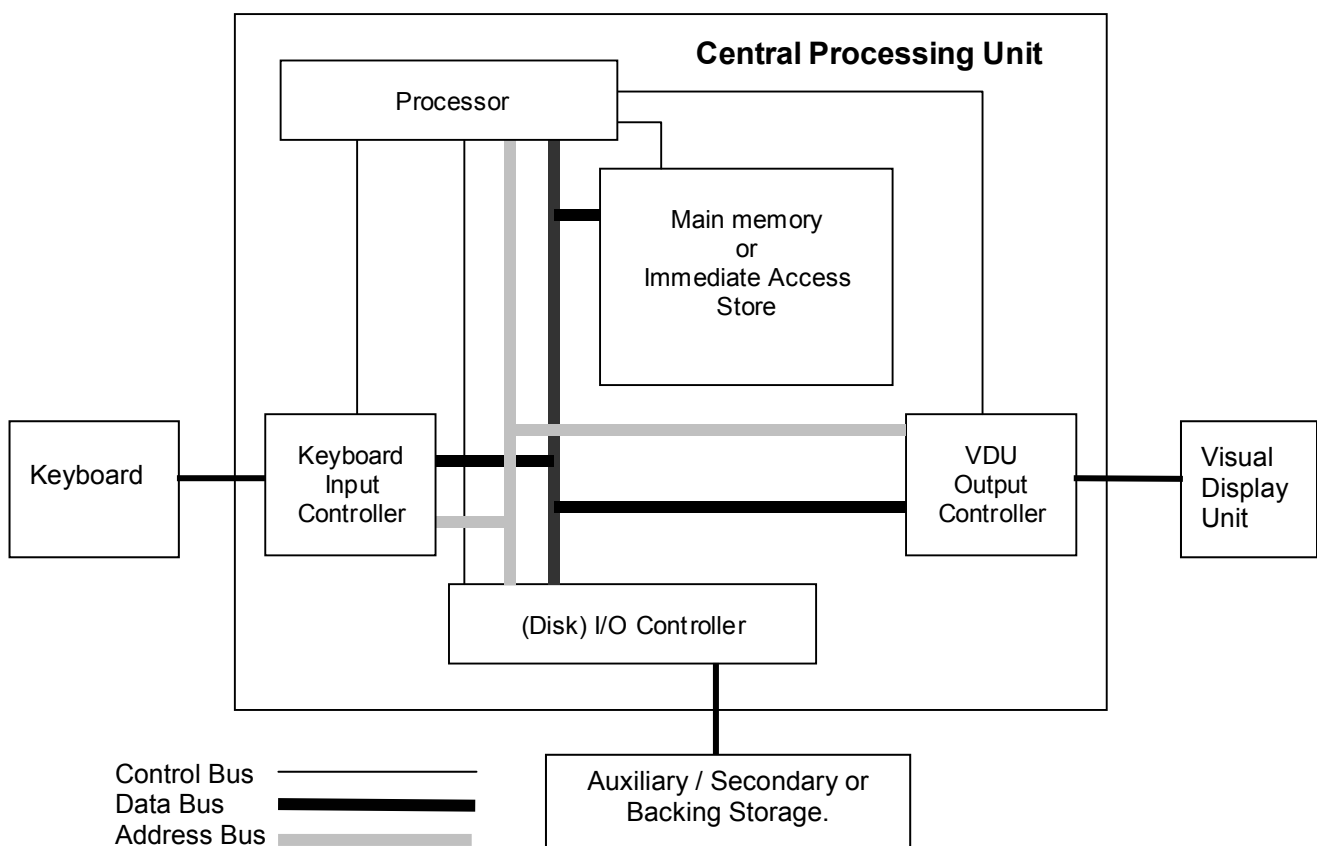
The term CPU is sometimes used to mean the actual processor unit which carries out the fetching, decoding and executing of instructions.

Other times it is used in a broader sense as the unit which houses the

- The processor;
- Main memory;
- IO controllers, some of which may be input only, some output only, some both input and output;
- Busses

The **external components** are also known as **peripherals** and include input, output and storage devices such as keyboard, mouse, printer and disk drives.

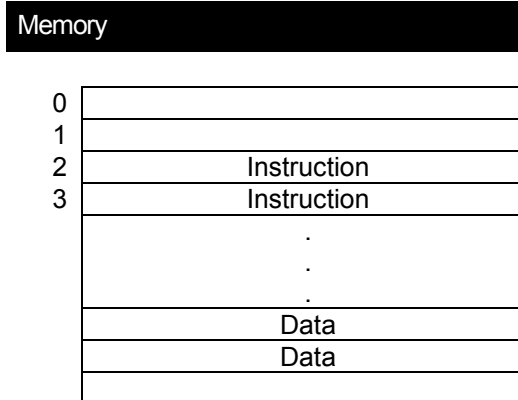
The processor receives and transmits data from and to the processor through a part of an I/O controller called an **I/O port**.



The internal and external components of a computer

## Memory and the stored program concept

Computers as we know them were first built in the 1940s, and two of the early pioneers were Alan Turing and John von Neumann. Each of them separately came up with the concept of a machine that would hold in a single store (main memory) both the instructions (program) and the data on which the instructions were to be carried out. Virtually all computers today are built on this principle, and so the general structure as shown below is sometimes referred to as the von Neumann machine.



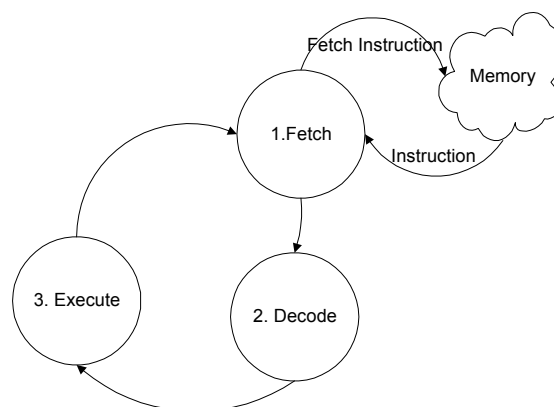
## The stored program concept

### The Processor

The processor contains the control unit and the arithmetic/logic unit (ALU).

The control unit coordinates and controls all the operations carried out by the computer. It operates by repeating three operations:

- Fetch - cause the next instruction to be fetched from main memory;
- Decode - translate the program instruction into commands that the computer can process
- Execute - cause the instruction to be executed;



The ALU can perform two sorts of operations on data.

1. **Arithmetic** operations include addition, subtraction, multiplication and division.
2. **Logical** operations consist of comparing one data item with another to determine whether the first data item is smaller than, equal to or greater than the second data item.

## Buses

A **bus** is a set of parallel wires connecting two or more components of the computer. The CPU is connected to main memory by three separate **buses**. When the CPU wishes to access a particular memory location, it sends this address to memory on the **address bus**.

The data in that location is then returned to the CPU on the **data bus**. Control signals are sent along the **control bus**.

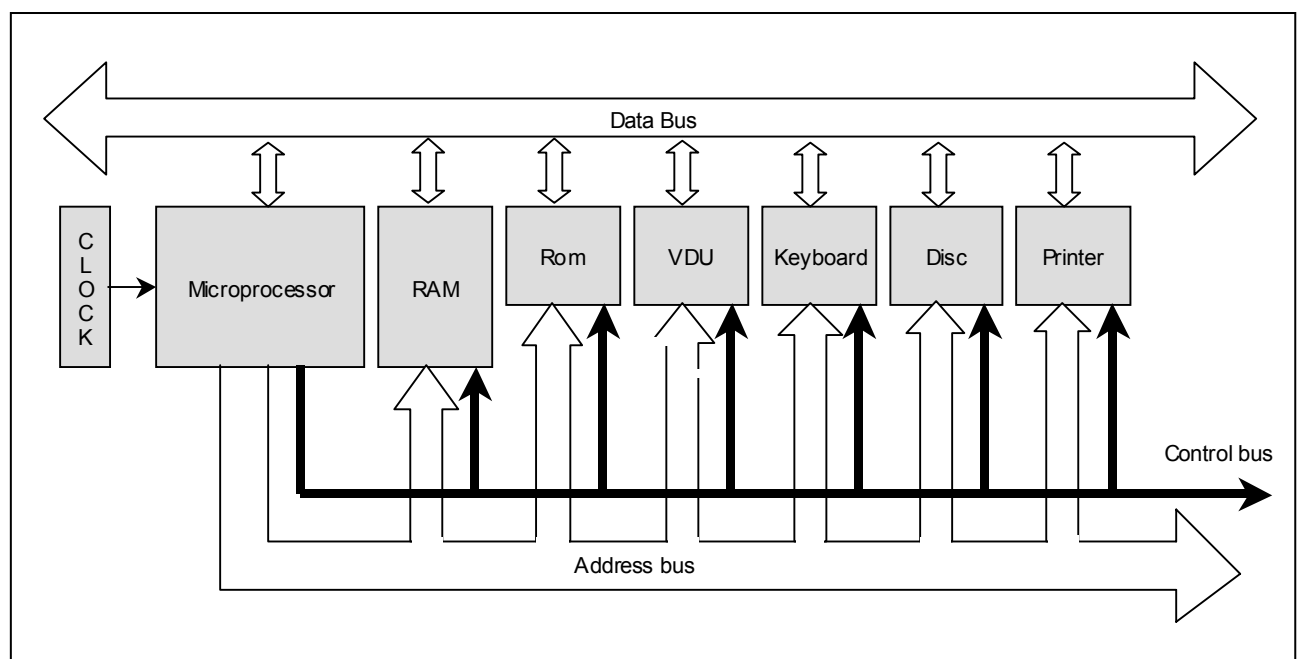
In the diagram below you can see that data, address and control buses connect the processor, memory and I/O controllers.

These are all **system buses**.

Each bus is a shared transmission medium, so that only one device can transmit along a bus at any one time.

Data and control signals travel in both directions between the processor, memory and I/O controllers.

Addresses on the other hand, travel only one way along the address bus: the processor sends the address of an instruction, or of data to be stored or retrieved, to memory or to an I/O controller (see diagram below.)



**Direction of transmission along the buses**

## Control bus

The control bus is a bi-directional bus meaning that signals can be carried in both directions. The data and address buses are **shared** by all components of the system.

Control lines must therefore be provided to ensure that **access to and use of the data and address buses** by the different components of the system does not lead to conflict.

The purpose of the control bus is to transmit command, timing and specific status information between system components. Timing signals indicate the validity of data and address information.

Command signals specify operations to be performed. Specific status signals indicate the state of a data transfer request, or the status of a request by a component to gain control of the system bus. Typical control lines include:

- *Memory Write*: causes data on the data bus to be written into the addressed location.
- *Memory Read*: causes data from the addressed location to be placed on the data bus.
- *I/O Write*: causes data on the data bus to be output to the addressed I/O port.
- *I/O Read*: causes data from the addressed I/O port to be placed on the data bus.
- *Transfer ACK*: indicates that data have been accepted from or placed on the data bus.
- *Bus Request*: indicates that a component needs to gain control of the system bus.
- *Bus Grant*: indicates that a requesting component has been granted control of the system bus.
- *Interrupt Request* indicates that an interrupt is pending.
- *Interrupt ACK*: acknowledges that the pending interrupt has been recognised.
- *Clock*: used to synchronise operations.
- *Reset*: initialises all components.

## Data bus

The data bus, typically consisting of 8, 16, or 32 separate lines provides a bi-directional path for moving data and instructions between system components.

***The width of the data bus is a key factor determining overall system performance.***

For example, if the data bus is 8 bits wide, and each instruction is 16 bits long, then the processor must access the main memory twice during each instruction cycle.

## Address bus

When the processor wishes to read a word (say 8, 16 or 32 bits) of data from memory, it first puts address of the desired word on the address bus.

**The width of the address bus determines the maxim possible memory capacity of the system.**

For example, if the address bus consisted of only 8 lines, then the maximum address it could transmit would be (in binary) 11111111 or 255 - giving a maxim memory capacity of 256 (including address 0).

A realistic minimum bus width would be 20 lines giving a memory capacity of  $2^{20}$ , that is 1Mb. The address bus is also used to address I/O ports during input/output operations.

No of address lines, m	Maximum no of addressable cells	Maximum no of addressable cells expressed as a power of two, $2^m$
1	2	$2^1$
2	4	$2^2$
3	8	$2^3$
4	16	$2^4$
8	256	$2^8$
16	65536	$2^{16}$
20	1048576	$2^{20}$
24	16777216	$2^{24}$

## Word size

The word size of a computer is the number of bits that the CPU can process simultaneously, as opposed to the bus size, which determines how many bits are transmitted together.

Processors can have 8-, 16-, or 64-bit word sizes (or even larger), and the word size will be one of the factors that determines the speed of the computer.

The faster, more powerful PCs have 32-bit processors, and most mainframes have 32-bit words.

Supercomputers may use a 64-bit or 128-bit word size.

## I/O Controllers

Peripheral devices cannot be connected directly to the processor.

Each peripheral operates in a different way and it would not be sensible to design processors to directly control every possible peripheral. Otherwise, the invention of a new type of peripheral would require the processor to be redesigned.

Instead, the processor controls and communicates with a peripheral device through an **I/O or device controller**.

I/O controllers are available which can operate both input and output transfers of bits, for example, floppy disk controller.

Other controllers operate in one direction only, either as an input controller, for example, keyboard controller or as output controller, for example, vdu controller.

The controller is an electronic circuit board consisting of three parts:

- an interface that allows connection of the controller to the system or I/O bus;
- a set of data, command and status registers;
- an interface that enables connection of the controller to the cable connecting the device to computer.

An interface is a standardised form of connection defining such things as signals, number of connecting pins/sockets and voltage levels that appear at the interface. An example is an RS232 interface which enables serial transmission of data between a computer and a serially connected printer. The printer also contains an RS232 interface so that both ends of the connection are compatible with each other.

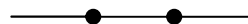
## Binary System

Digital computers use the binary system for representing data of all types — numbers, characters, sound, pictures and so on.

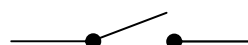
A binary system uses just 2 symbols to represent all information. The symbols could be anything like + and -, or 0 and 1.

The great advantage of the binary system is that the digits 1 and 0 can be represented by electrical circuits that can exist in one of two states — current is either flowing or not flowing, and a circuit is either closed or open, on or off.

A closed circuit allowing current to flow represents 1



An open circuit represents 0



### Bits and Bytes

A binary digit (1 or 0) is known as a bit short for Binary digit.

In most computers today, bits are grouped together in 8-bit bytes.

A byte can hold  $2^8$  different combinations of 0s and 1s, which means that, for example, 256 different characters can be represented.

**One byte holds one character**

### The ASCII code

Over the years, different computer designers have used different sets of codes for representing characters, which has led to great difficulty in transferring information from one computer to another.

Most personal computers (PCs) nowadays use the ASCII code (American Standard Code for Information Interchange), but many mainframe computers use a code called EBCDIC (Extended Binary Coded Decimal Interchange Code — pronounced EB-SEE-DICK).

ASCII originally used a 7-bit code. The 128 different combinations that can be represented in 7 bits is plenty to allow for all the letters, numbers and special symbols. Later, the eighth bit was also used, which allowed an extra 128 characters to be represented. The extra 128 combinations are used for symbols such as ç, è, ü, ©, ®, etc.

The first 32 ASCII codes are used for simple communications protocols, not characters. For example ACK stands for 'acknowledge' and would be sent by a device to acknowledge receipt of data or Communication signal.

### Representing numbers

Using ASCII, each character has a corresponding code, so that if for example the 'A' key on the keyboard is pressed, the code '01000001' will be sent to the CPU. If the key '1' is pressed, the code '00110001' will be sent to the CPU. To print the number '123', the codes for 1, 2 and 3 would be sent to the printer.

This is fine for input and output, but useless for arithmetic. There is no easy way of adding two numbers held in this way, and furthermore they occupy a great deal of space. Numbers, which are to be used in calculations, are therefore held in a different format as **binary numbers**.

Before we look at the binary system, it is helpful to examine how our ordinary decimal or **denary** numbers system works. Consider for example the number '134'. These three digits represent one hundred, three tens and four ones.

100	10	1
1	3	4

This represents  $100 + 30 + 4 = 134$

As we move from right to left each digit is worth ten times as much as the previous one. We probably a base 10 number system because we have ten fingers, but essentially there is no reason why some base such as 8 or 16 could not be used.

In the binary system, as we move from right to left each digit is worth twice as much as the previous Thus the binary number 1000110 can be set out under column headings as follows:

128	64	32	16	8	4	2	1
1	0	0	0	0	1	1	0

This represents  $128 + 4 + 2 = 134$

Obviously, using only one byte (8 bits) to hold a number places a severe restriction on the size of number the computer can hold. Therefore two or four consecutive bytes are commonly used to store numbers.

### **Memory addressing**

The memory of a computer can be thought of as a series of boxes, each containing 8 bits (1 byte), each with its own unique address, counting from zero upwards. The memory capacity of a computer is measured in thousand-byte units called kilobytes, megabytes or gigabytes.

These measures can be abbreviated to Kb, Mb and Gb. These are all powers of 2; thus although 1Kb often thought of as being 1,000 bytes, **it is actually 1024 bytes**.

You will find it useful to memorise certain powers of 2.

$$2^{10} \text{ bytes} = 1024 = 1\text{Kb}$$

$$2^{20} \text{ bytes} = 1024 \times 1024 = 1\text{Mb}$$

$$2^{30} \text{ bytes} = 1024 \times 1024 \times 1024 = 1\text{Gb}$$

**Exercises**

1. What is meant by each of the following terms?
  - (i) bit (1)
  - (ii) byte (1)
  - (iii) word (1)
  
2. A microprocessor *data bus* has 16 lines and its address bus has 24 lines. What is the maximum memory capacity that can be connected to the microprocessor? (2)
  
3. (a) What is meant by the term **bus** in the context of computer hardware? (1)  
(b) There are three types of external bus connecting the processor to main memory.  
Briefly describe a purpose of each ? (3)
  
4. (a) Briefly describe the roles of the data bus and the address bus within the Central Processing Unit(CPU). (3)  
(b) (i) State one benefit of increasing the width of the data bus.  
(ii) State one benefit of increasing the width of the address bus. (2)
  
5. A memory location in a certain microcomputer contains 8 bits.
  - (i) How many different data patterns can be stored in one memory location? (1)
  - (ii) What is meant by the term “word” in this context? (1)