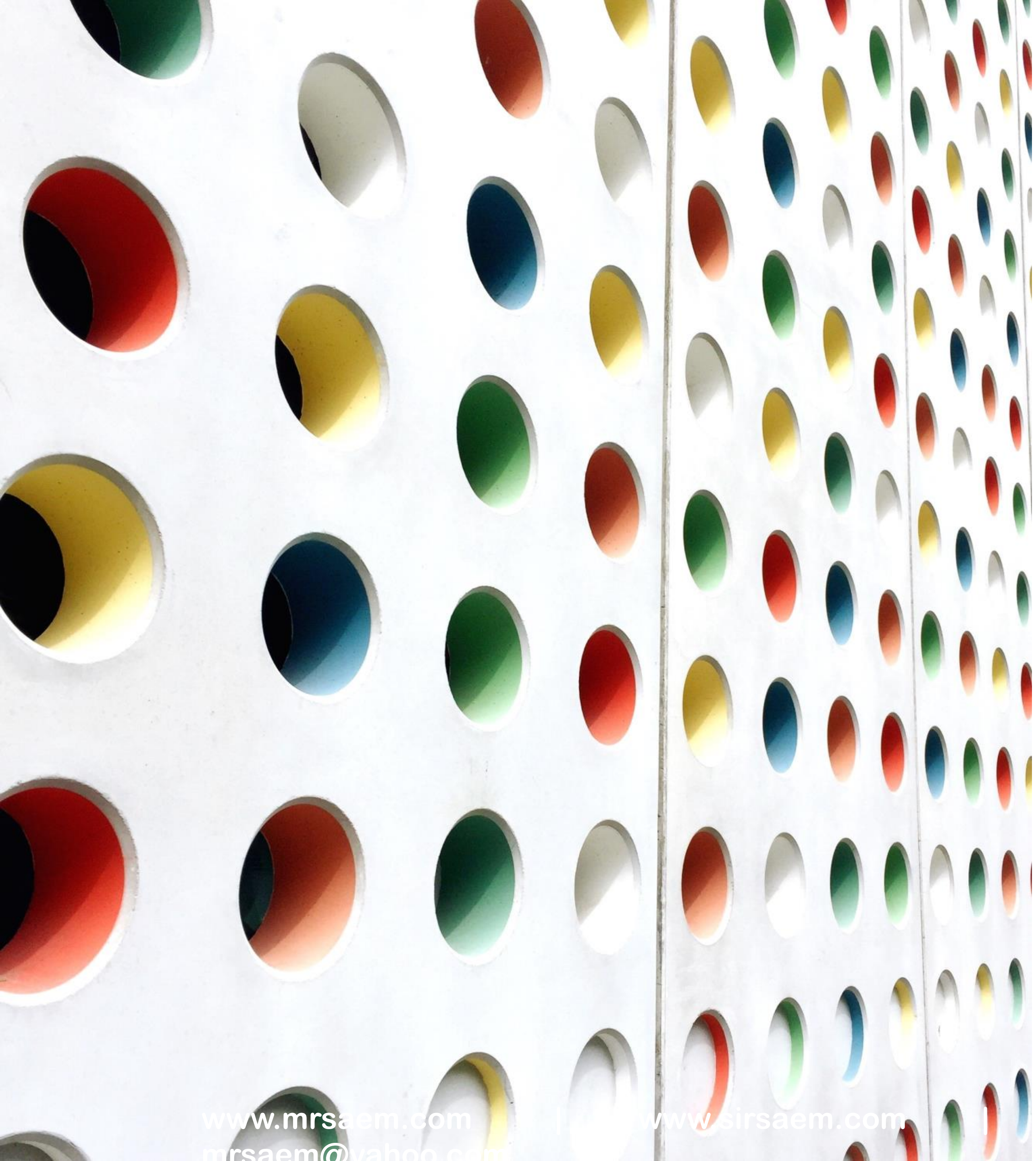


Chapter 5:

Processor Fundamentals

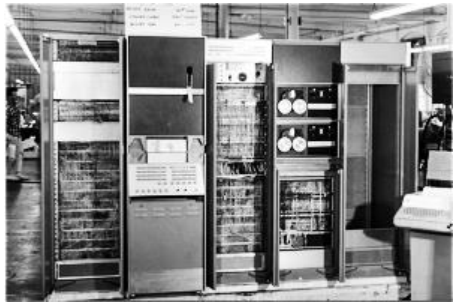
CS With Mr Saem

Designing Computers

In the mid-1940s, John Von Neumann developed the concept of the stored program computer. It has been the basis of computer architecture for many years.

The main, previously unavailable, features of the Von Neumann architecture were

- **a central processing unit (CPU or processor)**
- **a processor able to access the memory directly**
- **computer memories that could store programs as well as data**
- **stored programs made up of instructions that could be executed in sequential order.**

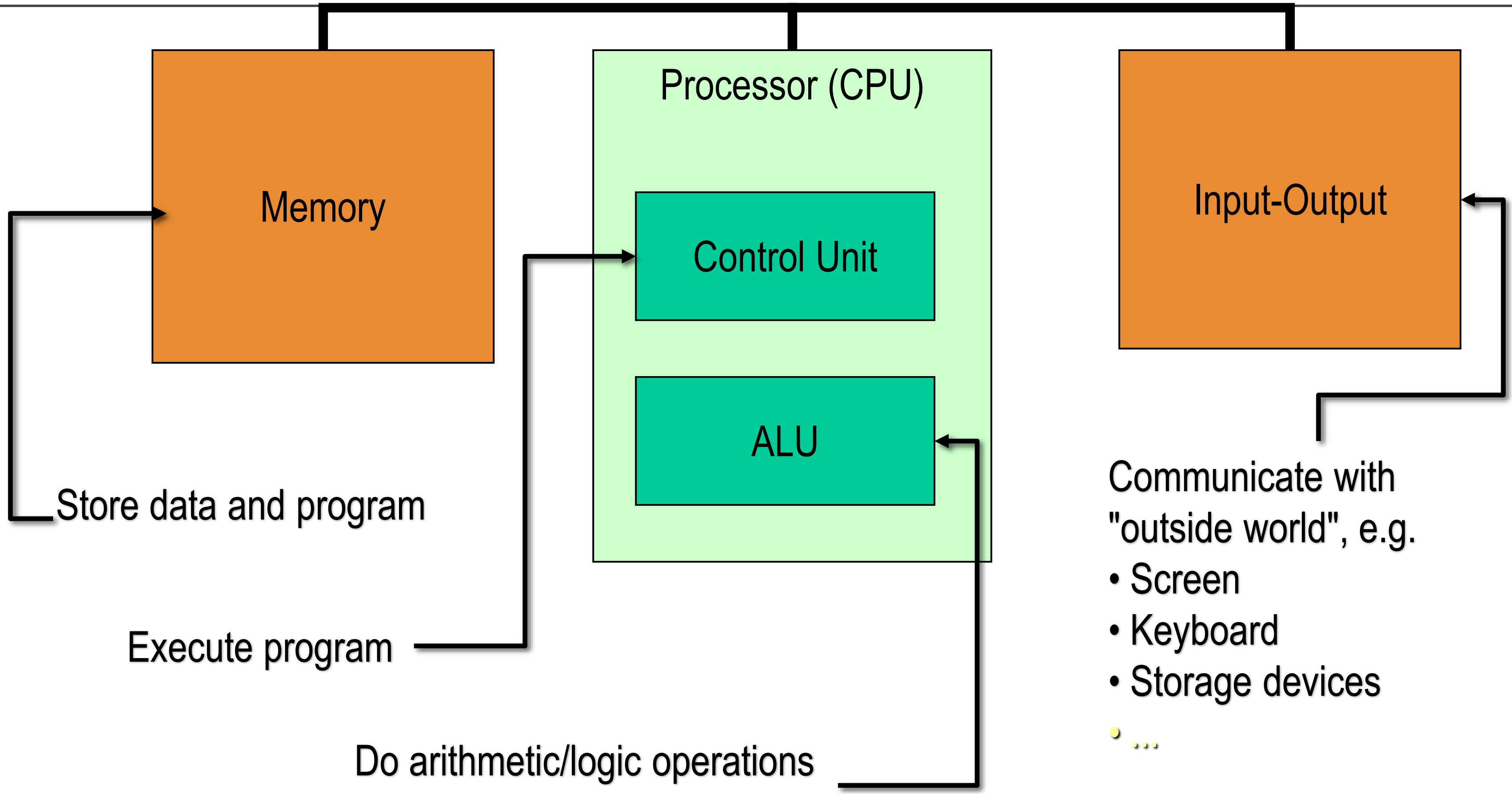


The Von Neumann Architecture

Model for designing and building computers, based on the following three characteristics:

1) The computer consists of four main sub-systems:
   - **Memory**
   - **ALU (Arithmetic/Logic Unit)**
   - **Control Unit**
   - **Input/Output System (I/O)**
2) Program is stored in memory during execution.
3) Program instructions are executed sequentially.

The Von Neumann Architecture



Store data and program

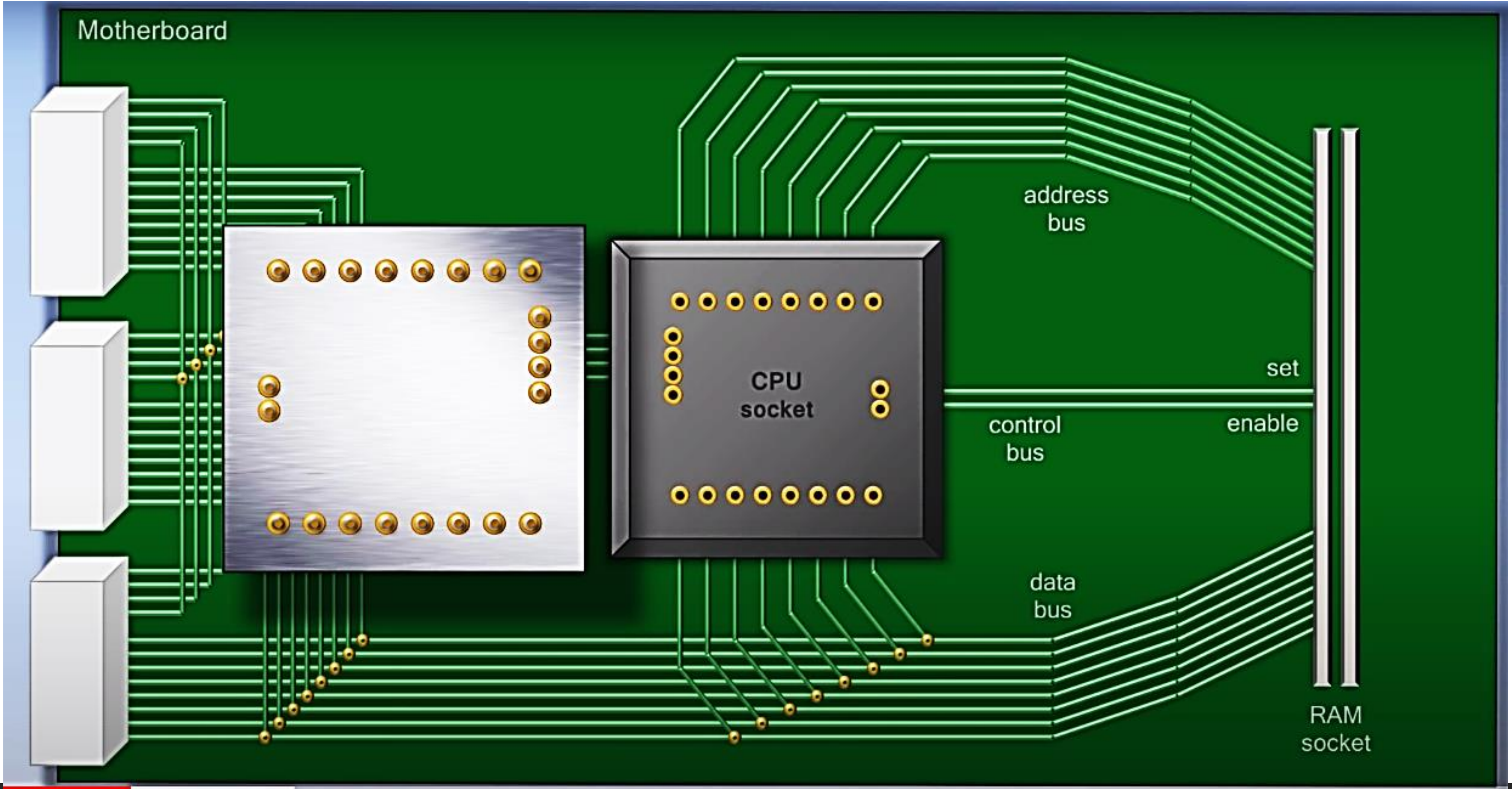
Execute program

Do arithmetic/logic operations requested by program

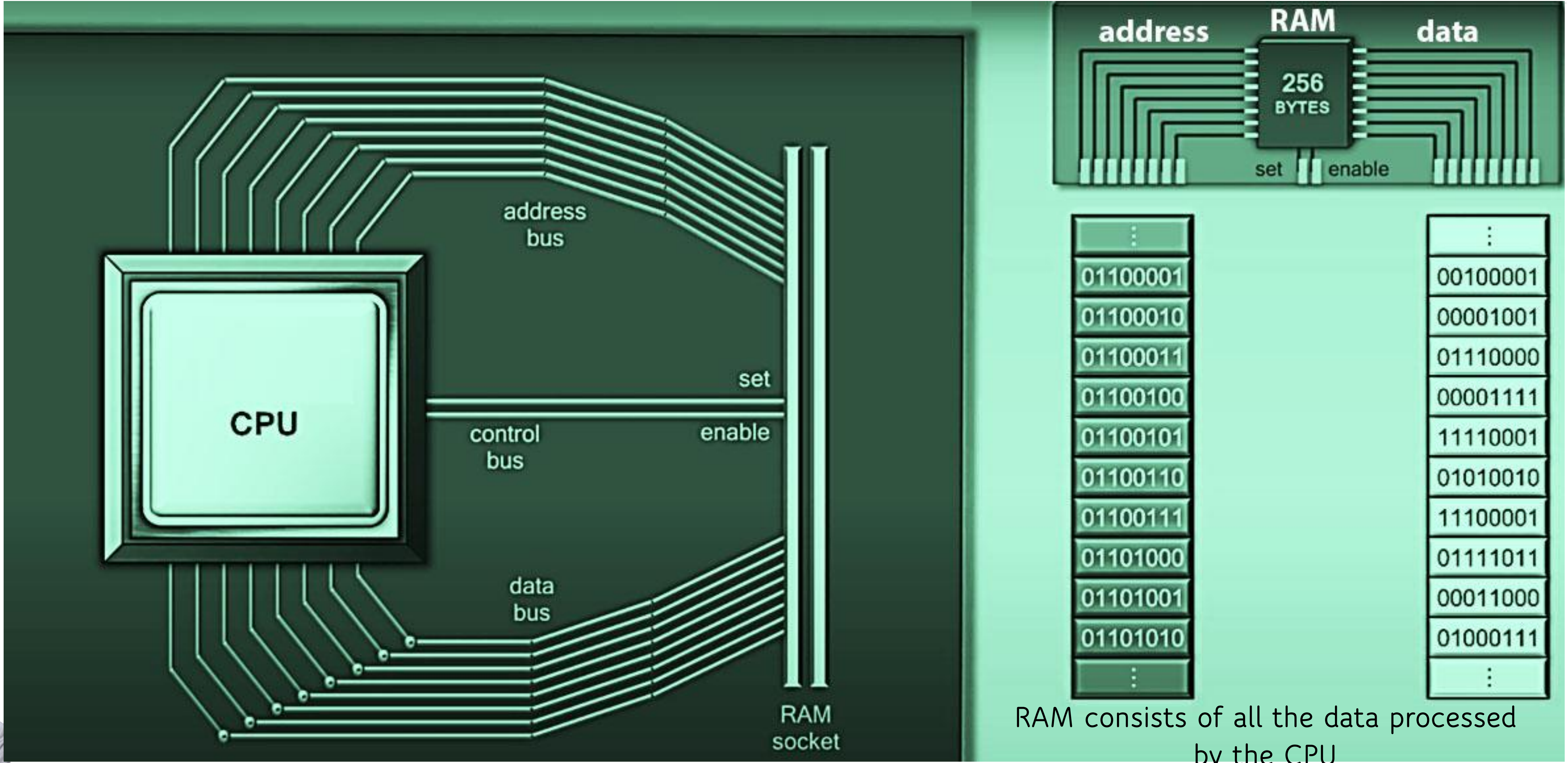
Communicate with "outside world", e.g.

- Screen
- Keyboard
- Storage devices
- ...

MotherBoard



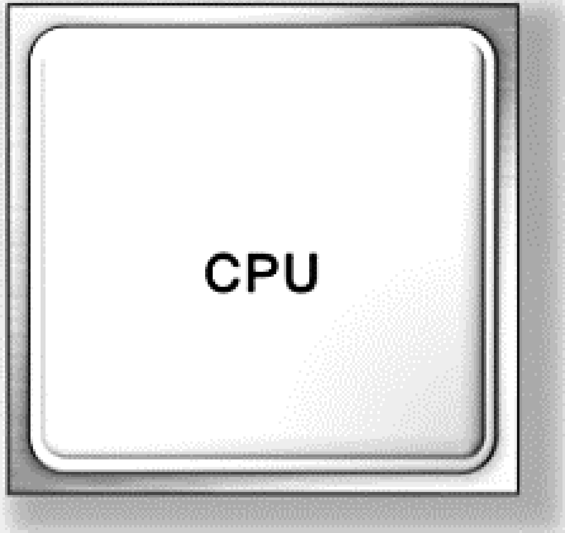
CPU-BUS-RAM



address	data
⋮	⋮
01100001	00100001
01100010	00001001
01100011	01110000
01100100	00001111
01100101	11110001
01100110	01010010
01100111	11100001
01101000	01111011
01101001	00011000
01101010	01000111
⋮	⋮

RAM consists of all the data processed by the CPU

Instruction Set



LOAD a number from RAM into the CPU

ADD two numbers together

STORE a number from the CPU back out to RAM

COMPARE one number with another

JUMP IF *Condition* to another address in RAM

JUMP to another address in RAM

OUTPUT to a device such as a monitor

INPUT from a device such as a keyboard

⋮	⋮
01100001	Instruction
01100010	Number
01100011	Instruction
01100100	Address
01100101	Instruction
01100110	Instruction
01100111	Address
01101000	Instruction
01101001	Address
01101010	Letter
⋮	⋮

RAM – Random Access Memory

RAM is temporary storage space that can be accessed very quickly.

Physically, RAM is a chip or series of chips on which the data is stored electronically.

- **It is made up of millions of cells, each of which has its own unique address.**
- **Each cell can contain either an instruction or some data.**
- **The cells can be accessed as they are needed by the processor, by referencing the address.**
- **That is they can be accessed randomly, hence the name.**
- **Because they are electronic they are able to be accessed quickly.**

However, RAM is volatile, which means that when you turn your computer off, all of the contents of RAM are lost. Whenever a program is run on your computer, the entire program or parts of it are loaded into RAM. The more memory you have, the more applications you can have loaded at any one time. For example, if you load a spreadsheet file, both the file and the spreadsheet application are stored in RAM. When you are creating formulae, these are stored temporarily in

Components of the CPU

Memory or Storage Unit

This unit can store instructions, data, and intermediate results. This unit supplies information to other units of the computer when needed. It is also known as internal storage unit or the main memory or the primary storage or Random Access Memory (RAM).

- **Its size affects speed, power, and capability. Primary memory and secondary memory are two types of memories in the computer.**

Functions of the memory unit are −

- **It stores all the data and the instructions required for processing.**
- **It stores intermediate results of processing.**
- **It stores the final results of processing before these results are released to an output device.**
- **All inputs and outputs are transmitted through the main memory.**

Components of the CPU

Control Unit

This unit controls the operations of all parts of the computer but does not carry out any actual data processing operations.

Functions of this unit are −

- **It is responsible for controlling the transfer of data and instructions among other units of a computer.**
- **It manages and coordinates all the units of the computer.**
- **It obtains the instructions from the memory, interprets them, and directs the operation of the computer.**
- **It communicates with Input/Output devices for transfer of data or results from storage.**
- **It does not process or store data.**

Components of the CPU

ALU (Arithmetic Logic Unit)

This unit consists of two subsections namely,

Arithmetic Section

Logic Section

Arithmetic Section

Function of arithmetic section is to perform arithmetic operations like addition, subtraction, multiplication, and division. All complex operations are done by making repetitive use of the above operations.

Logic Section

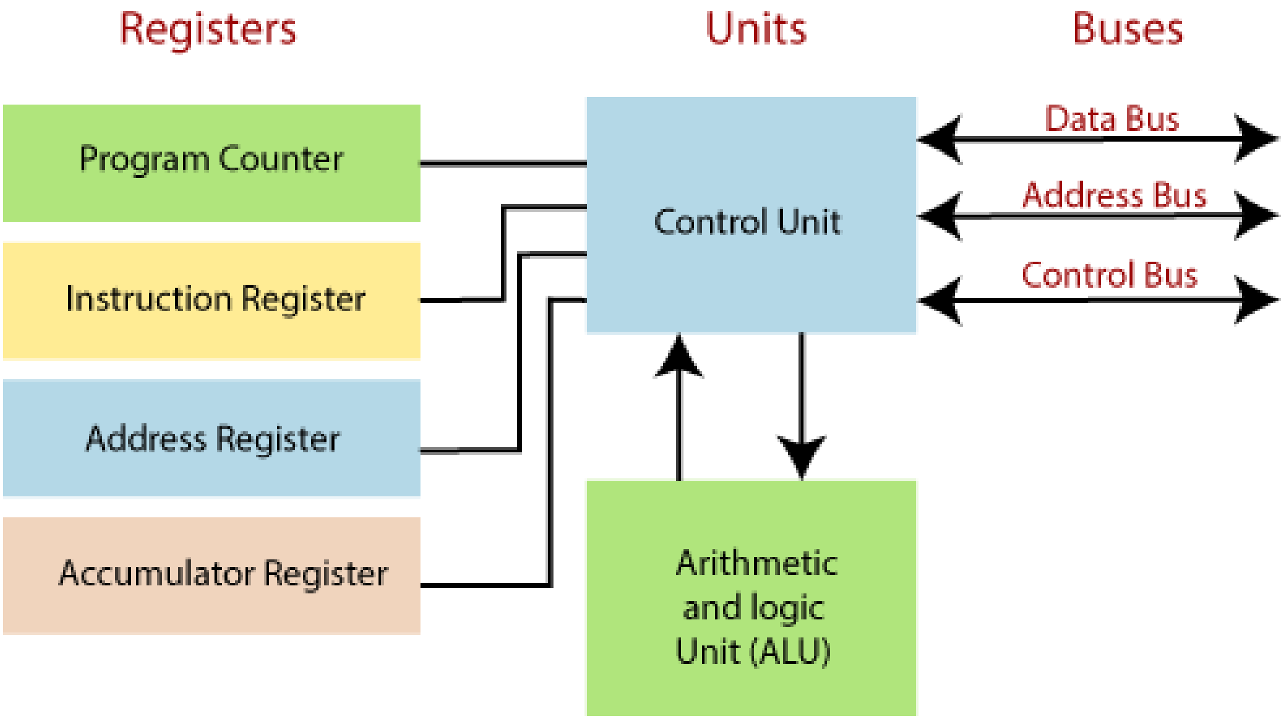
Function of logic section is to perform logic operations such as comparing, selecting, matching, and merging of data.

CPU Registers

A register is a temporary storage area built into a CPU.

Some registers are used internally and cannot be accessed outside the processor, while others are user-accessible. Most modern CPU architectures include both types of registers.

The Central Processing Unit (CPU)



CPU Registers

MAR	Memory Address Register	Holds the memory location of data that needs to be accessed
MDR	Memory Data Register	Holds data that is being transferred to or from memory
AC	Accumulator	Where intermediate arithmetic and logic results are stored
PC	Program Counter	Contains the address of the next instruction to be executed
CIR	Current Instruction Register	Contains the current instruction during processing
IR [IX]	Index register	Stores a value; only used for indexed addressing
SR	Status register	Contains individual bits that are either set or cleared

1. **Memory Address Register (MAR):** This register holds the address of memory where CPU wants to read or write data. When CPU wants to store some data in the memory or reads the data from the memory, it places the address of the required memory location in the MAR

2. **Program Counter (PC):** Program Counter register is also known as Instruction Pointer Register. This register is used to store the address of the next instruction to be fetched for execution. When the instruction is fetched, the value of instruction pointer IP is incremented. Thus this register always points or holds the address of next instruction to be fetched.

3 **Instruction Register (IR):** Once an instruction is fetched from main memory, it is stored in the Instruction Register. The control unit takes instruction from this register, decodes and executes it by sending signals to the appropriate component of computer to carry out the task.

4. **The Memory Data Register (MDR)** holds data that is being transferred to or from memory.

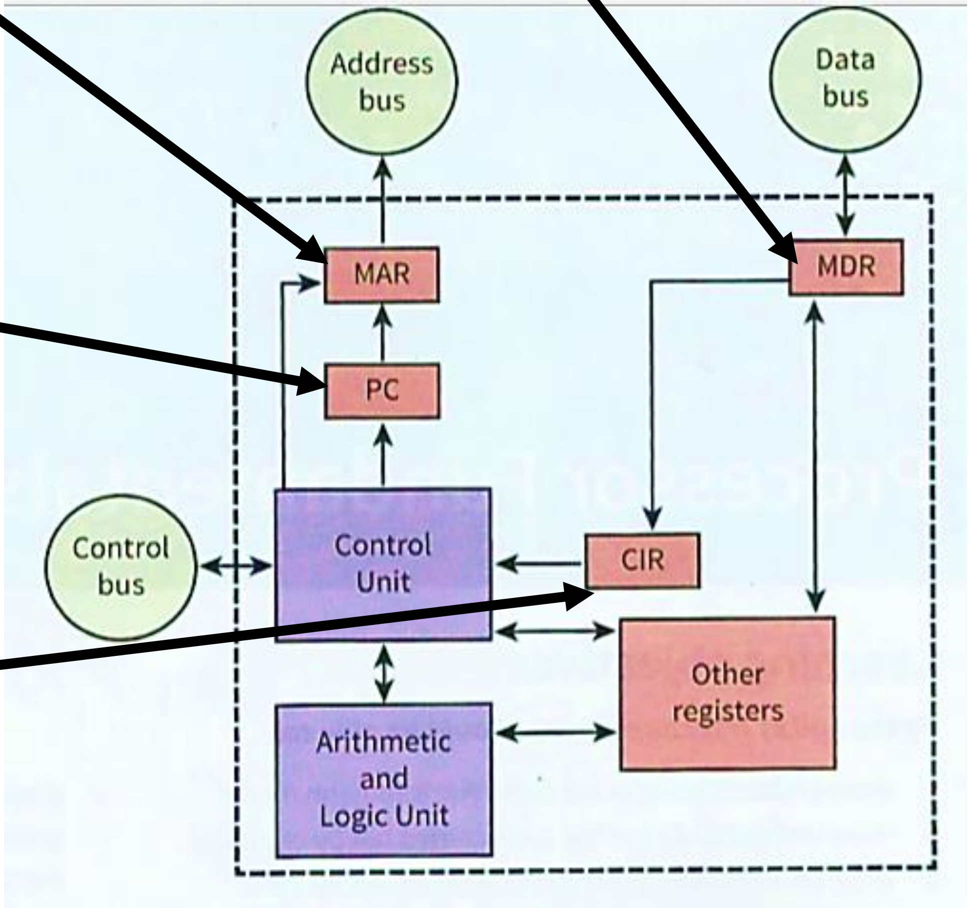
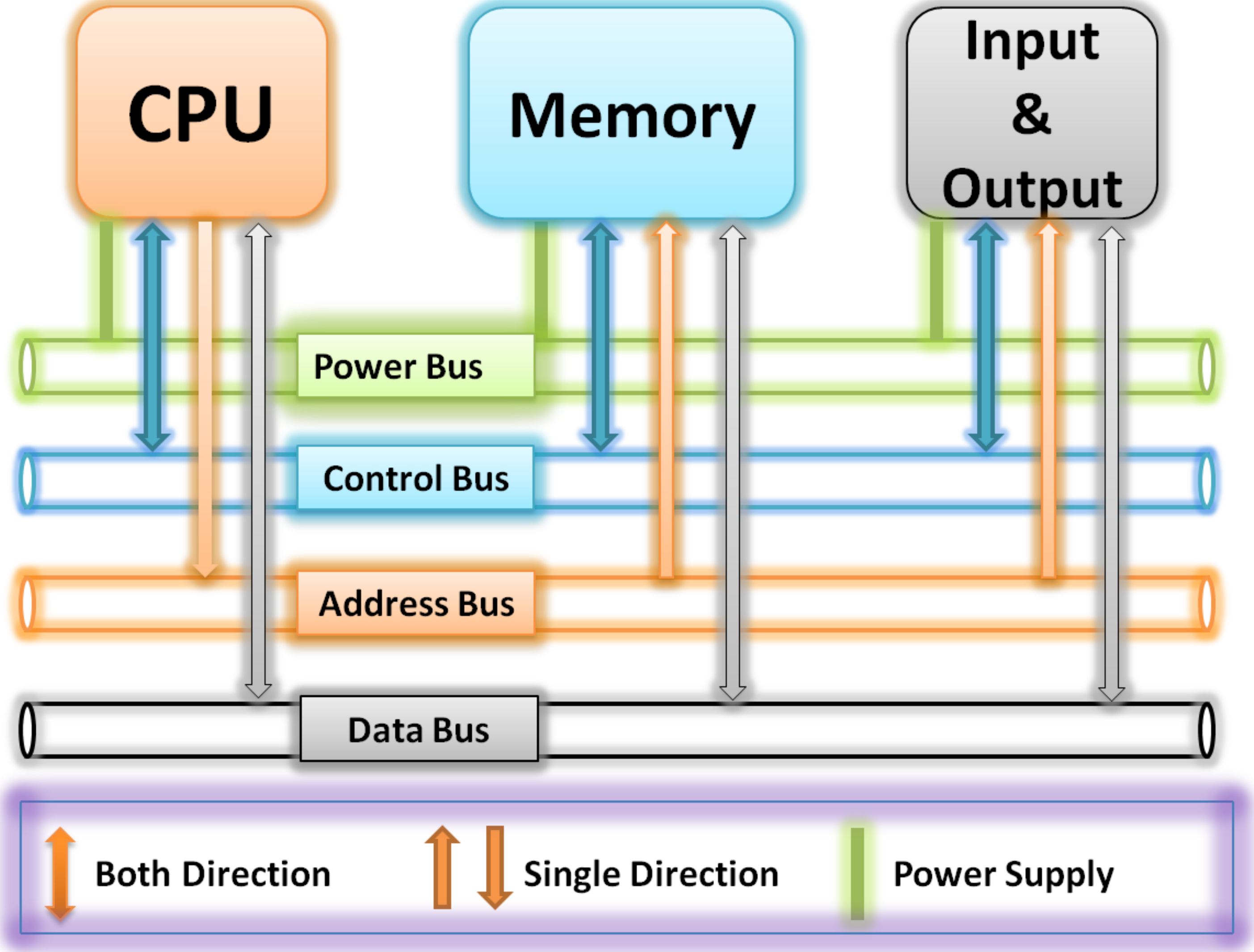


Figure 5.01 A schematic diagram of the architecture of a simple CPU

Buses


CPU
Memory
Input & Output
Power Bus
Control Bus
Address Bus
Data Bus
Both Direction
Single Direction
Power Supply

Buses

Buses are the means by which data is transmitted from one part of a computer to another, connecting all major internal components to the CPU and memory.

A standard CPU system bus is comprised of a **control bus, data bus and address bus.**

Address Bus	Carries the addresses of data (but not the data) between the processor and memory. The address bus is a 'one-way street'. It can only be used to send an address to a memory controller. It cannot be used to carry an address from the memory controller back to the CPU. Address. The width (i.e. the number of wires) determines the number of memory locations the CPU can address. A 32 bit address bus has 32 parallel wires each switched on (1) or off (0). Every time a wire is added to the width of the address bus, the address range doubles. The width of a bus is important. **The wider the bus, the more memory locations which can be directly addressed at any given time; for example, a bus of width 16 bits can address 2 16 (65 536) memory locations, whereas a bus width of 32 bits allows 4 294 967 296 memory locations to be simultaneously addressed**

Data Bus	Carries data between the processor, the memory unit and the input/output devices. It is bi-directional i.e. two-way. The number of wires determines the quantity of data that the bus can carry so increasing the number of wires in the data bus increases the quantity of data it can carry. A typical 32-bit data bus can carry 32-bits of data or instructions at a time
Control Bus	Carries control signals/commands from the CPU (and status signals from other devices) in order to control and coordinate all the activities within the computer. It is made up of discrete wires each with a specific function. Read and Write signals are initiated to fetch – execute instructions in memory.

The address bus is connected to the MAR; the data bus to the MOR; and the control bus to the control unit

The system bus allows data flow between the CPU, the memory, and input or output (1/0) devices as shown in the schematic diagram in Figure 5.02.

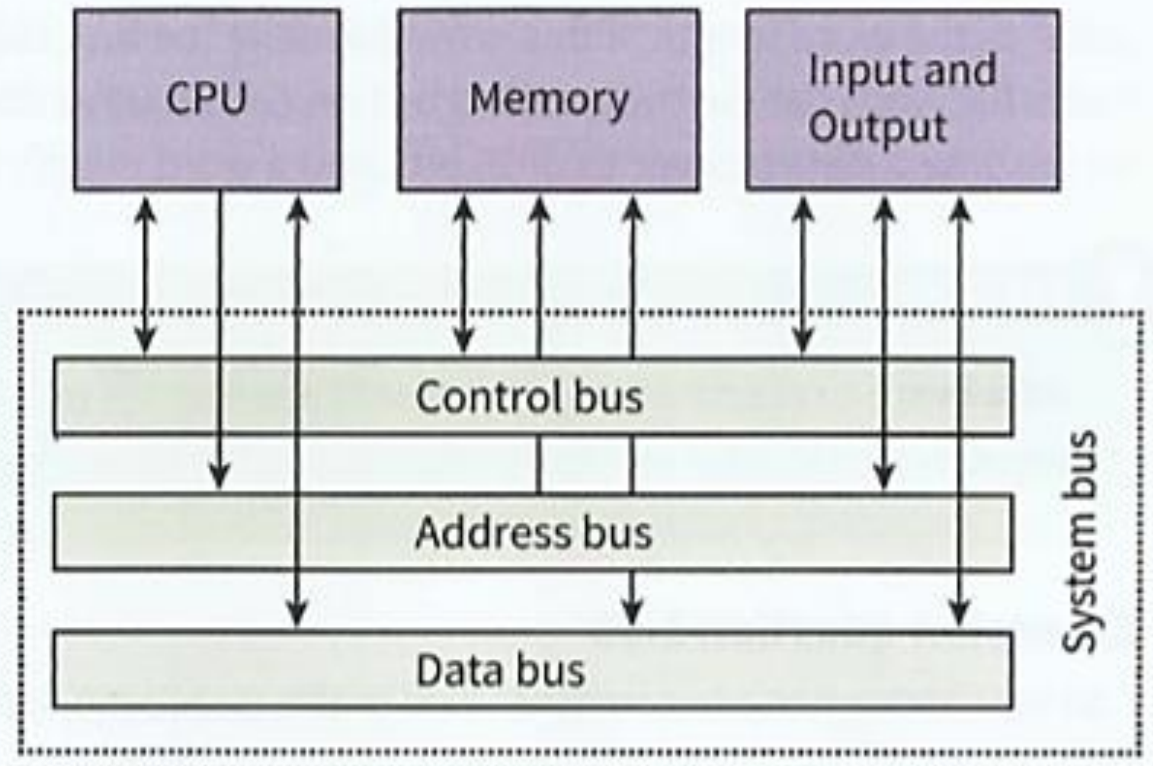


Figure 5.02 A schematic diagram of the system bus

There are four registers that are used by the processor as part of the fetch-execute cycle:

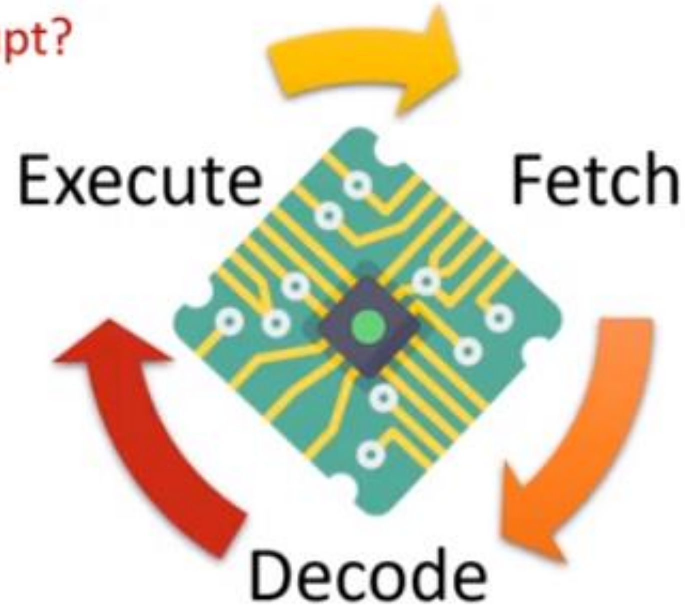
O The Current Instruction Register (CIR) stores the instruction that is currently being executed by the processor.

O The Program Counter (PC) stores the memory location of the next instruction that will be needed by the processor.

O The Memory Buffer Register (MBR), also known as the Memory Data Register (MDR), holds the data that has just been read from or is about to be written to main memory.

O The Memory Address Register (MAR) stores the memory location where data in the MBR is about to be written to or read from.

What is an interrupt?



Execute

Fetch

Decode

Keyboard

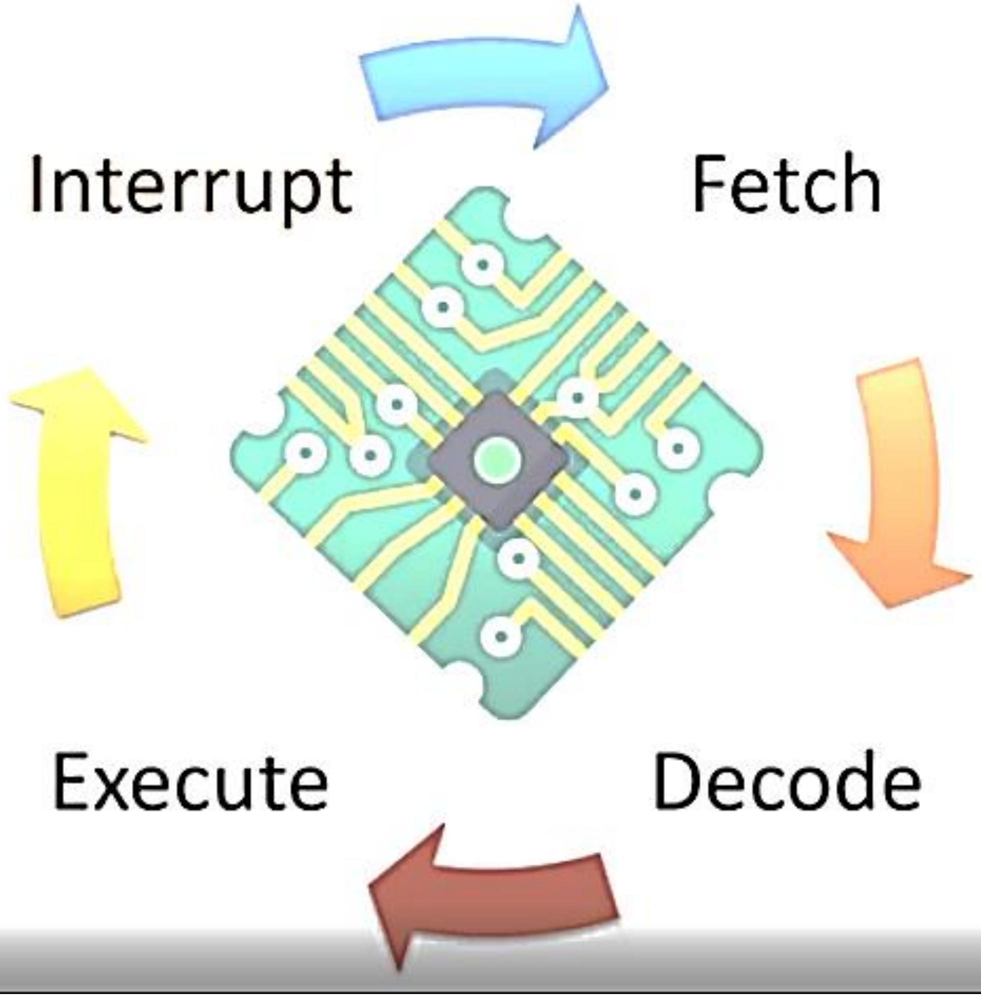
Printer

Hard disk controller

RAM

- A computer system executes programs.
- The processor fetches, decodes and executes instructions over and over.
- While this is occurring, other devices and applications may require the processor's attention.
- They need a way to signal to the processor that they require attention – that is what we call an **interrupt**.

What is an interrupt?



Interrupt

Fetch

Decode

Execute

- We have evolved our understanding of the fetch-decode-execute cycle to add an additional step:
    1. Fetch
    2. Decode
    3. Execute
    4. Check for new interrupts