COA MODULE 7

TAXONOMY OF PARALLEL ARCHITECTURES

SISD

SIMD

MISD

MIMD

VON NEUMAN ARCHITECHTURE

Historically there have been 2 types of Computers:

Fixed Program Computers – Their function is very specific and they couldn't be programmed, e.g. Calculators.

Stored Program Computers – These can be programmed to carry out many different tasks, applications are stored on them, hence the name.

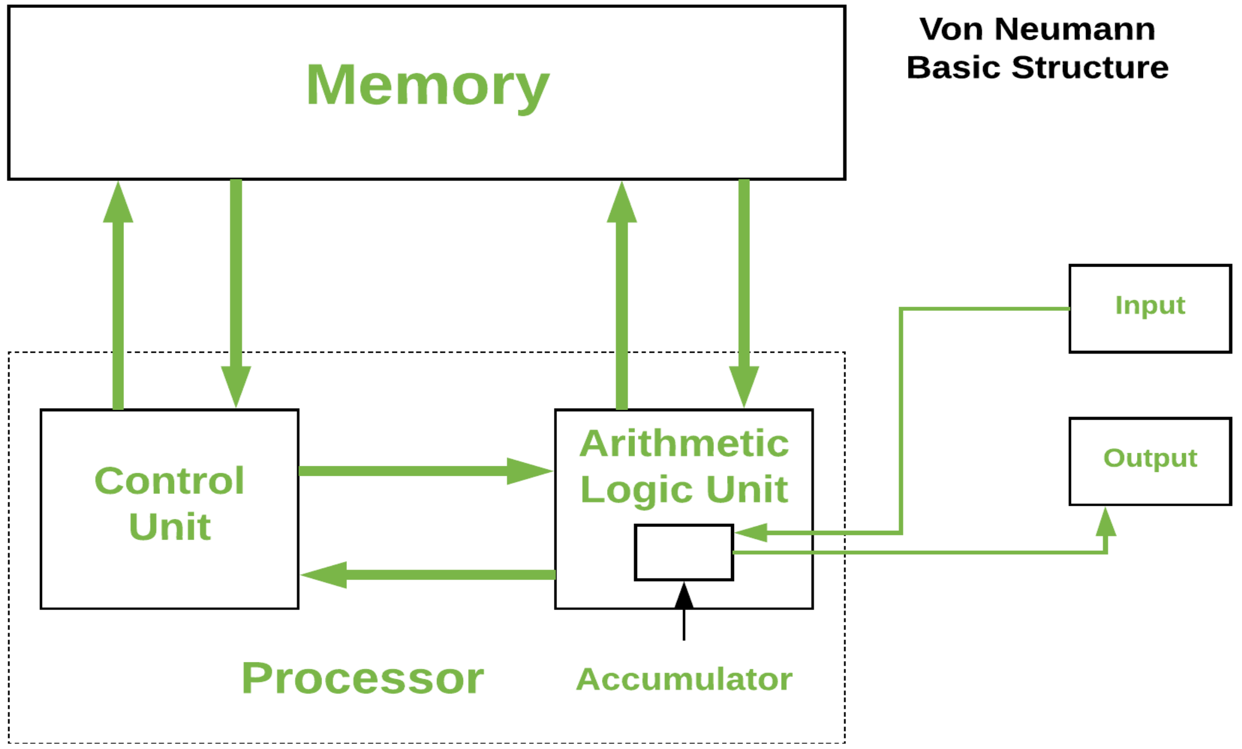
BASIC DETAILS

The modern computers are based on a stored-program concept introduced by John Von Neumann.

In this stored-program concept, programs and data are stored in a separate storage unit called memories and are treated the same.

This novel idea meant that a computer built with this architecture would be much easier to reprogram.

BASIC ARCHITECTURE



IT IS ALSO KNOWN AS **IAS** COMPUTER AND IS HAVING THREE BASIC UNITS

The Central Processing Unit (CPU)

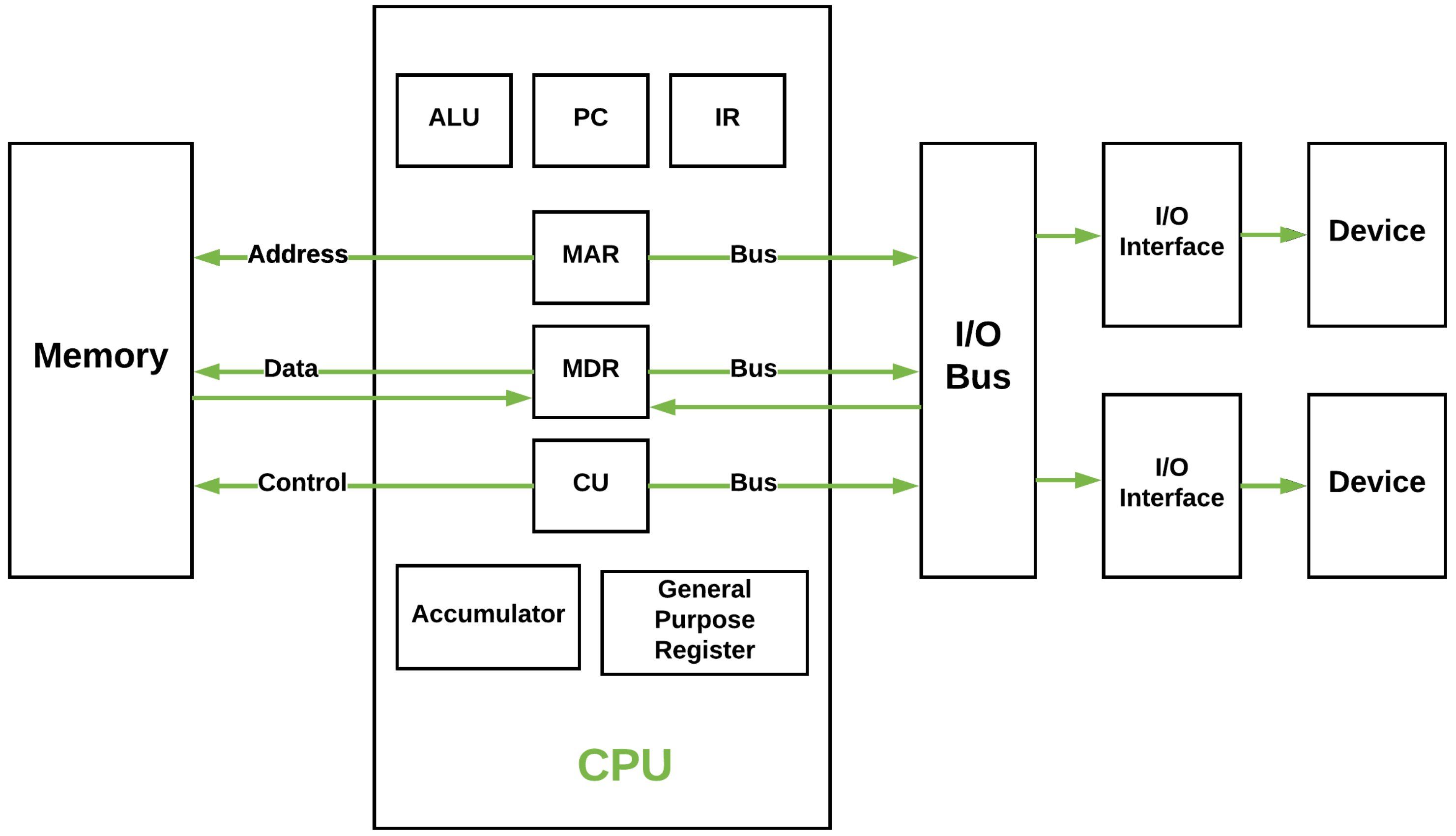
The Main Memory Unit

The Input /Output Device

Control Unit –A control unit (CU) handles all processor control signals. It directs all input and output flow, fetches code for instructions and controlling how data moves around the system.

Arithmetic and Logic Unit (ALU) –The arithmetic logic unit is that part of the CPU that handles all the calculations the CPU may need, e.g. Addition, Subtraction, Comparisons. It performs Logical Operations, Bit Shifting Operations, and Arithmetic Operation.

FIGURE – BASIC CPU STRUCTURE, ILLUSTRATING ALU



DETAILS

Main Memory Unit (Registers) –Accumulator: Stores the results of calculations made by ALU.

Program Counter (PC): Keeps track of the memory location of the next instructions to be dealt with. The PC then passes this next address to Memory Address Register (MAR).

Memory Address Register (MAR): It stores the memory locations of instructions that need to be fetched from memory or stored into memory.

Memory Data Register (MDR): It stores instructions fetched from memory or any data that is to be transferred to, and stored in, memory.

Current Instruction Register (CIR): It stores the most recently fetched instructions while it is waiting to be coded and executed.

Instruction Buffer Register (IBR): The instruction that is not to be executed immediately is placed in the instruction buffer register IBR.

BASIC DETAILS

Input / Output Devices – Program or data is read into main memory from the *input device* or secondary storage under the control of CPU input instruction. O*utput devices* are used to output the information from a computer. If some results are evaluated by computer and it is stored in the computer, then with the help of output devices, we can present it to the user.

Buses – Data is transmitted from one part of a computer to another, connecting all major internal components to the CPU and memory, by the means of Buses. Types:

- **Data Bus:** It carries data among the memory unit, the I/O devices, and the processor.
- **Address Bus:** It carries the address of data (not the actual data) between memory and processor.
- **Control Bus:** It carries control commands from the CPU (and status signals from other devices) in order to control and coordinate all the activities within the computer.

VON NEUMANN BOTTLENECK

Whatever we do to enhance performance, we cannot get away from the fact that instructions can only be done one at a time and can only be carried out sequentially.

Both of these factors hold back the competence of the CPU. This is commonly referred to as the ‘Von Neumann bottleneck’.

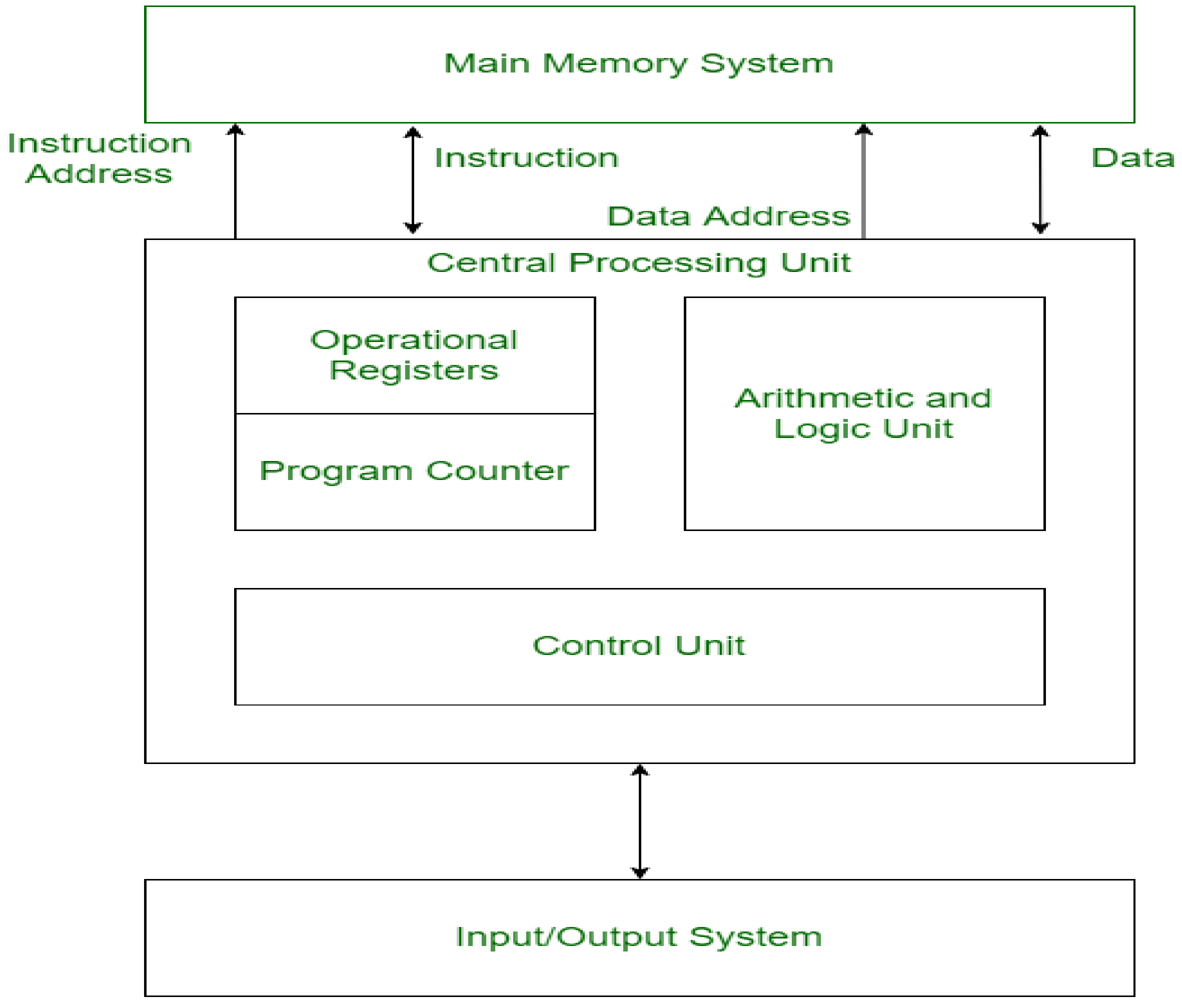
We can provide a Von Neumann processor with more cache, more RAM, or faster components but if original gains are to be made in CPU performance then an influential inspection needs to take place of CPU configuration.

This architecture is very important and is used in our PCs and even in Super Computers.

HAVARD ARCHITECTURE

Harvard Architecture is the digital computer architecture whose design is based on the concept where there are separate storage and separate buses (signal path) for instruction and data.

It was basically developed to overcome the bottleneck of Von Neumann Architecture.



Harvard Architecture

DIFFERENCE BETWEEN VON NEUMANN AND HARVARD ARCHITECTURE

VON NEUMANN ARCHITECTURE	HARVARD ARCHITECTURE
It is ancient computer architecture based on stored program computer concept.	It is modern computer architecture based on Harvard Mark I relay based model.
Same physical memory address is used for instructions and data.	Separate physical memory address is used for instructions and data.
There is common bus for data and instruction transfer.	Separate buses are used for transferring data and instruction.
Two clock cycles are required to execute single instruction.	An instruction is executed in a single cycle.
It is cheaper in cost.	It is costly than van neumann architecture.
CPU can not access instructions and read/write at the same time.	CPU can access instructions and read/write at the same time.
It is used in personal computers and small computers.	It is used in micro controllers and signal processing.

MULTICOMPUTERS

Shared Memory

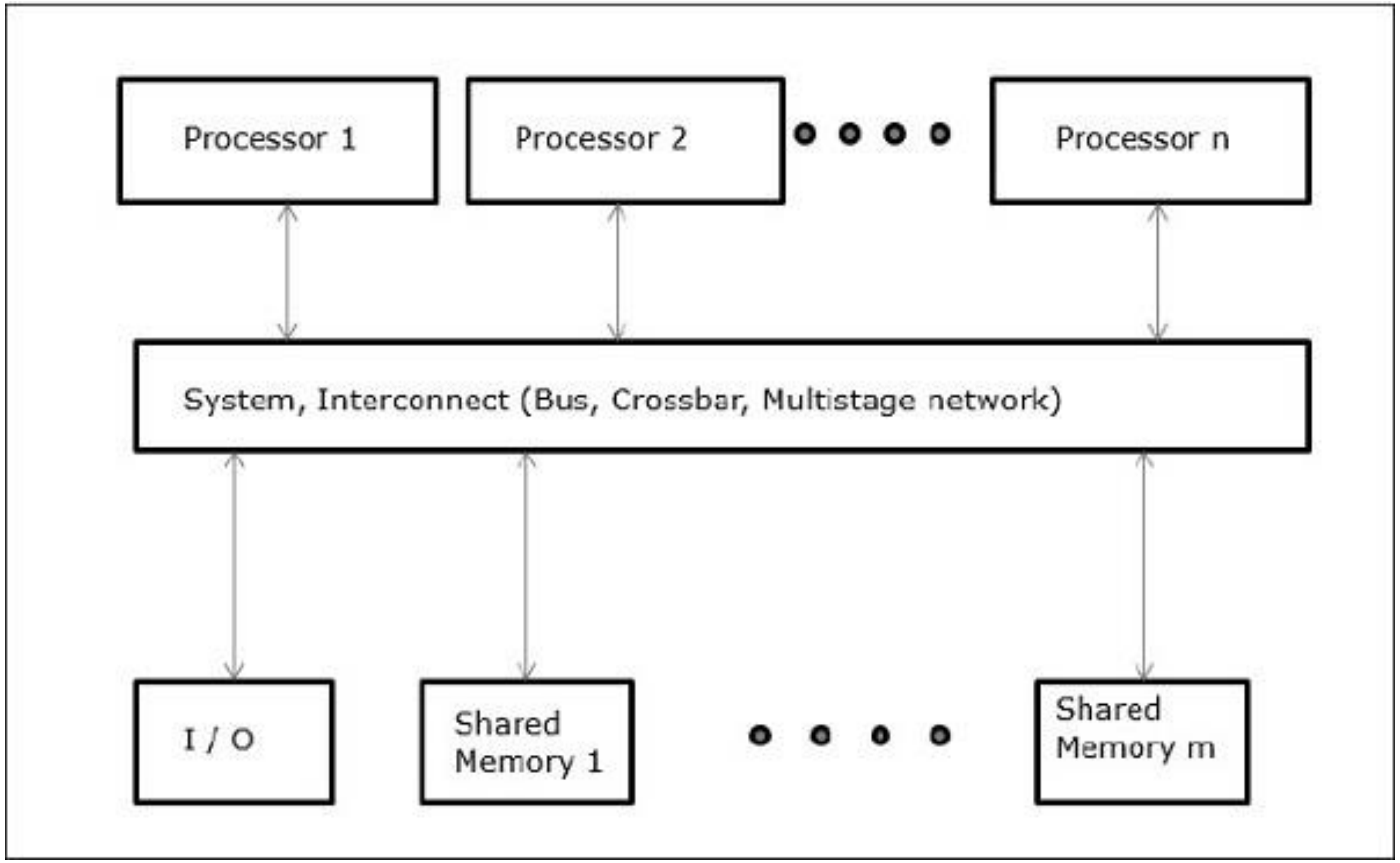
Distributed Memory

UMA (UNIFORM MEMORY ACCESS)

In this model, all the processors share the physical memory uniformly. All the processors have equal access time to all the memory words. Each processor may have a private cache memory. Same rule is followed for peripheral devices.

When all the processors have equal access to all the peripheral devices, the system is called a **symmetric multiprocessor**. When only one or a few processors can access the peripheral devices, the system is called an **asymmetric multiprocessor**.

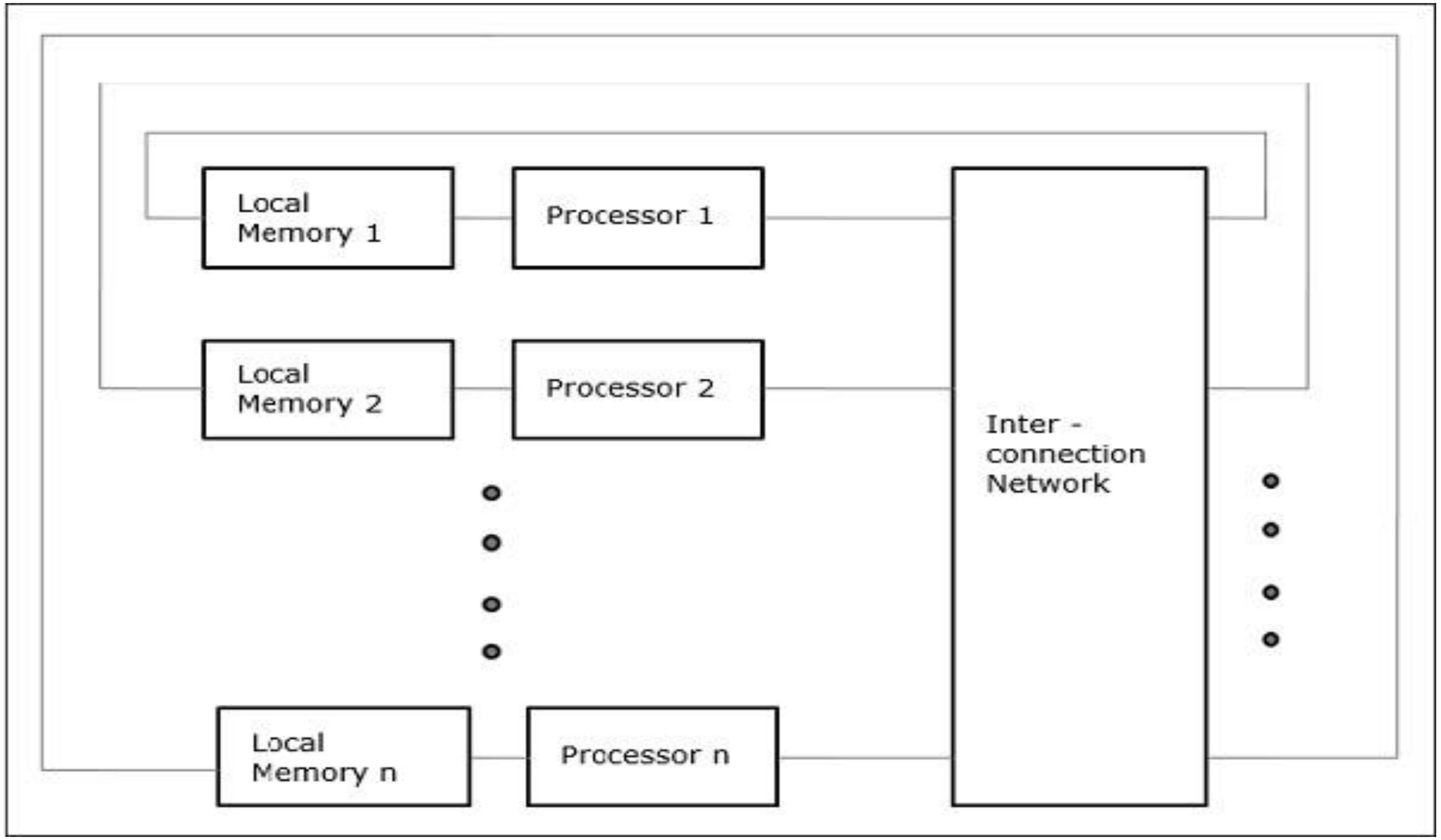
UMA


Processor 1
Processor 2
Processor n
System, Interconnect (Bus, Crossbar, Multistage network)
I / O
Shared Memory 1
Shared Memory m

NON-UNIFORM MEMORY ACCESS (NUMA)

In NUMA multiprocessor model, the access time varies with the location of the memory word. Here, the shared memory is physically distributed among all the processors, called local memories. The collection of all local memories forms a global address space which can be accessed by all the processors.

NUMA

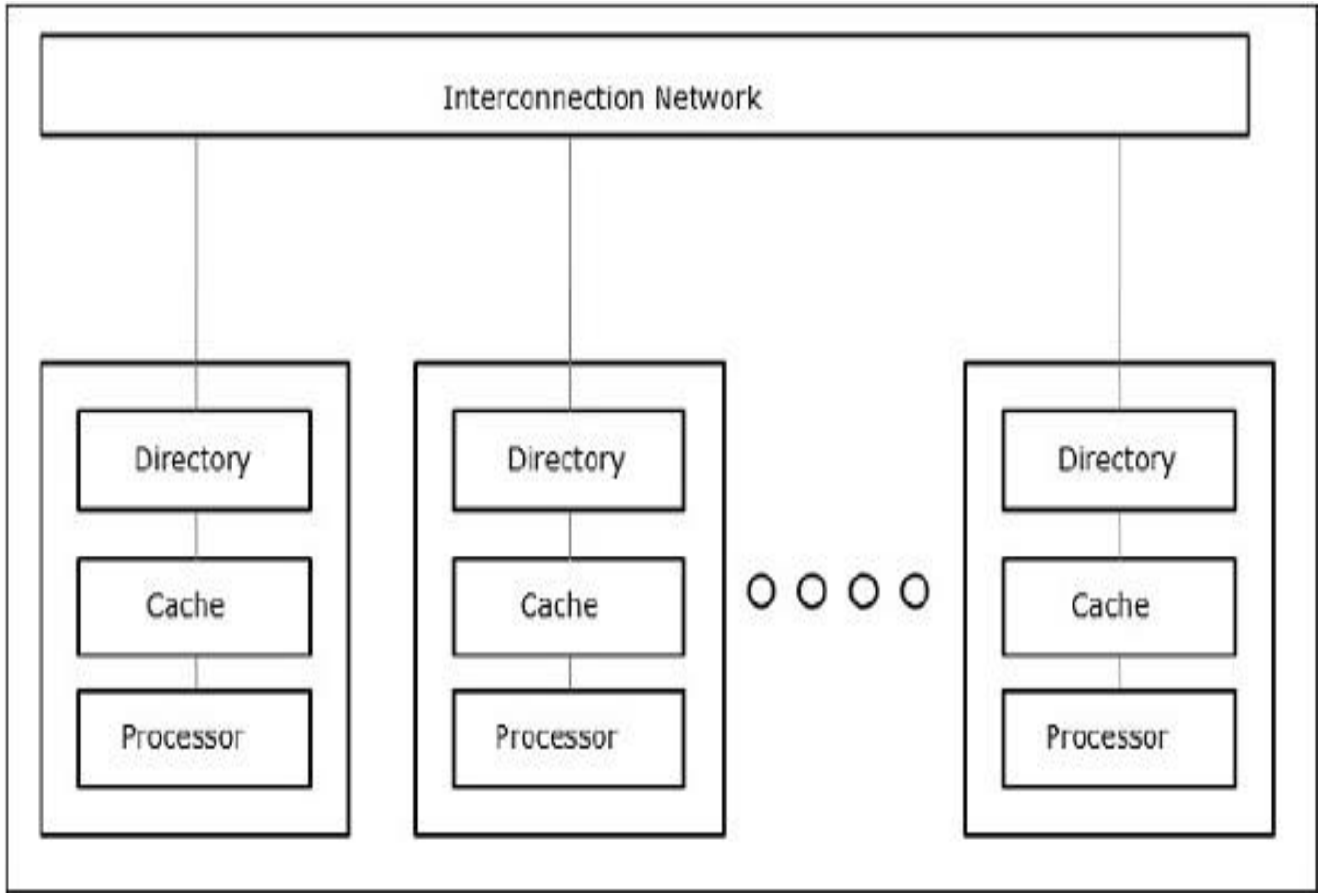

Local Memory 1
Processor 1
Local Memory 2
Processor 2
Inter - connection Network
Local Memory n
Processor n

CACHE ONLY MEMORY ARCHITECTURE (COMA)

The COMA model is a special case of the NUMA model.

Here, all the distributed main memories are converted to cache memories.

COMA

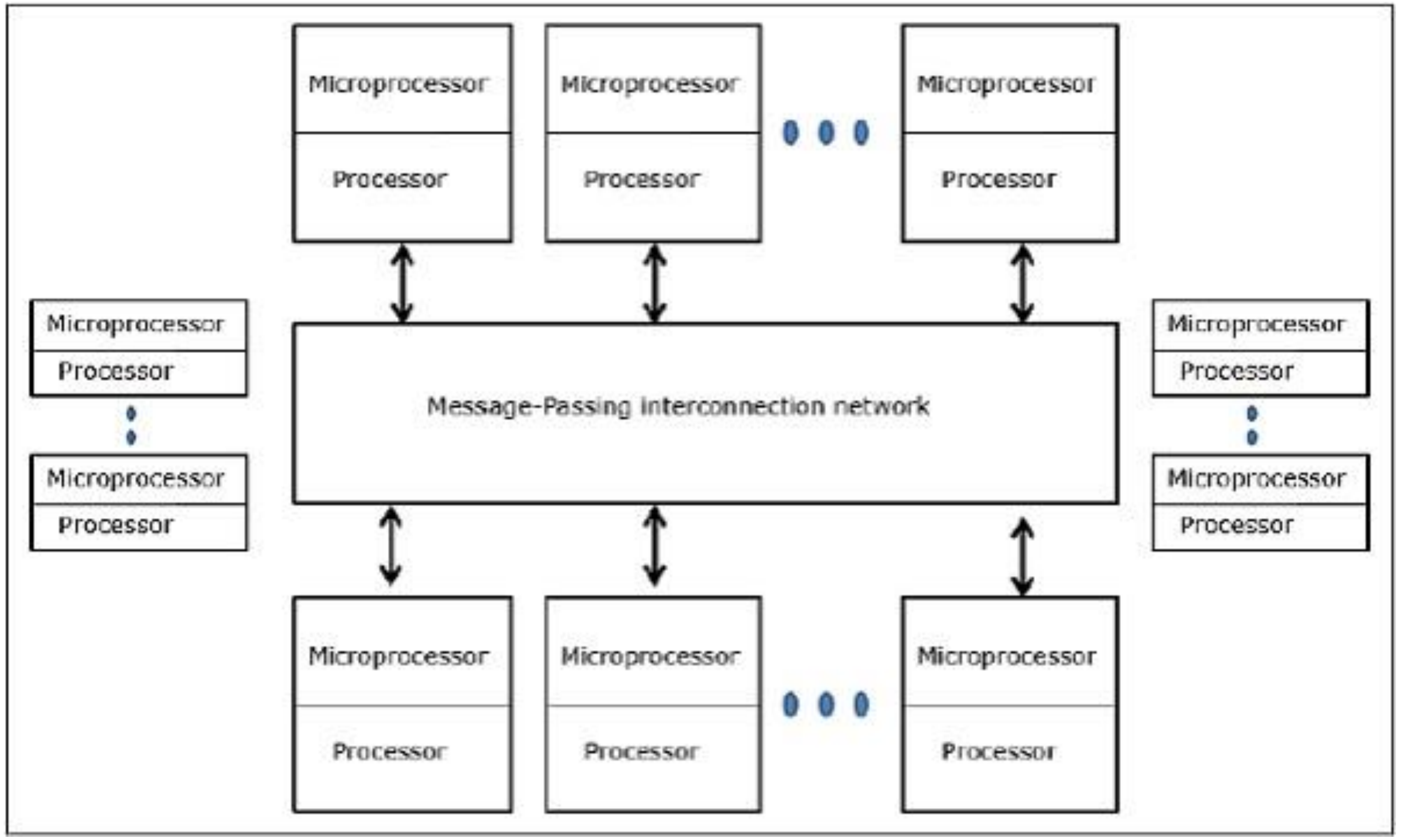

Interconnection Network
Directory
Cache
Processor
Directory
Cache
Processor
Directory
Cache
Processor

DISTRIBUTED - MEMORY MULTICOMPUTERS

A distributed memory multicomputer system consists of multiple computers, known as nodes, inter-connected by message passing network. Each node acts as an autonomous computer having a processor, a local memory and sometimes I/O devices. In this case, all local memories are private and are accessible only to the local processors.

This is why, the traditional machines are called **no-remote-memory-access** (**NORMA**) machines.

DISTRIBUTED MEMORY ARCHITECTURE



THANK YOU!!