Unit 6
Multiprocessors

Characteristics of Multiprocessor, Structure of Multiprocessor- Inter-processor Arbitration, Inter-Processor Communication and Synchronization. Memory in Multiprocessor System, Concept of Pipelining, Vector Processing, Array Processing, RISC and CISC, Study of Multicore Processor – Intel, AMD.

Multiprocessor:

Multiprocessor system is the use of two or more central processing units (CPUs) within a single computer system. The term also refers to the ability of a system to support more than one processor or the ability to allocate tasks between them. There are many variations on this basic theme, and the definition of multiprocessing can vary with context, mostly as a function of how CPUs are defined (multiple cores on one die, multiple dies in one package, multiple packages in one system unit, etc.).

According to some on-line dictionaries, a multiprocessor is a computer system having two or more processing units (multiple processors) each sharing main memory and peripherals, in order to simultaneously process programs.

Characteristics of Multiprocessor:

- A Multiprocessor system implies the existence of multiple CPUs, although usually there will be one or more IOPs as well.
- A Multiprocessor system is controlled by one operating system that provides interaction between processors and all the components of the system cooperate in the solution of a problem.
- Microprocessors take very little physical space and are very inexpensive brings about the feasibility of interconnecting a large number of microprocessors into one computer system.

Advantages of Multiprocessor Systems are:

- Increased Reliability because of redundancy in processors.
- Increased throughput because of execution.
- Multiple jobs in parallel.
- Portions of the same job in parallel.

A single job can be divided into independent tasks, either manually by the programmer, or by the compiler, which finds the portions of the program that are data independent, and can be executed in parallel. The multiprocessors are further classified into two groups depending on the way their memory is organized. The processors with shared memory are called tightly coupled or shared memory processors. The information in these processors is shared through the common memory. Each of the processors can also have their local memory too. The other class of multiprocessors is loosely coupled or distributed memory multi-processors. In this, each processor has their own private memory, and they share information with each other through interconnection switching scheme or message passing.

The principal characteristic of a multiprocessor is its ability to share a set of main memory and some I/O devices. This sharing is possible through some physical connections between them called the interconnection structures.

Structure of Multiprocessor:

Time Shared Common Bus:

A system common bus multiprocessor system consists of a number of processors connected through path to a memory unit.

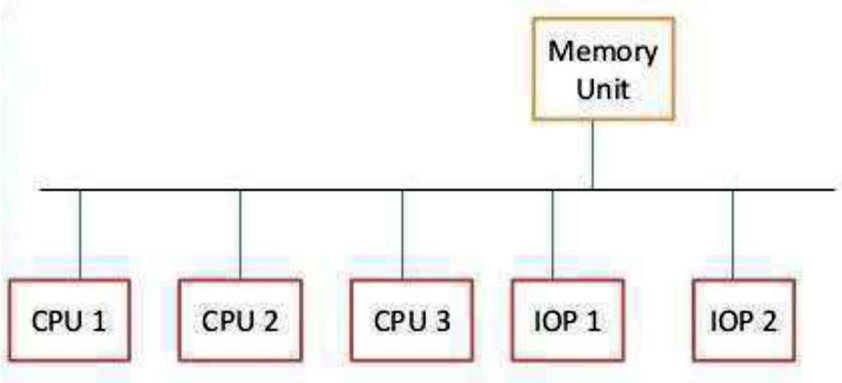

Figure 1: Time Share Common Bus

Hierarchical Bus Systems:

A hierarchical bus system consists of a hierarchy of buses connecting various systems and sub-systems/components in a computer. Each bus is made up of a number of signal, control, and power lines. Different buses like local buses, backplane buses and I/O buses are used to perform different interconnection functions.

Local buses are the buses implemented on the printed-circuit boards. A backplane bus is a printed circuit on which many connectors are used to plug in functional boards. Buses which connect input/output devices to a computer system are known as I/O buses.

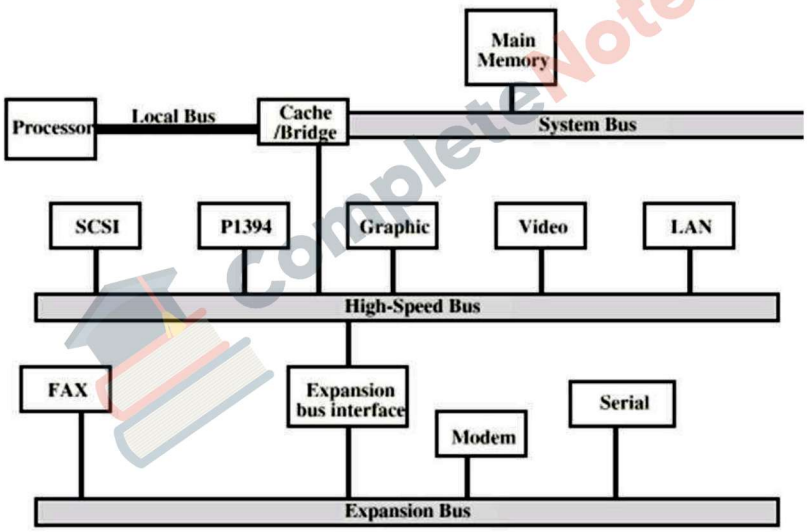

Figure 2: Hierarchical Bus System

Crossbar switch and Multiport Memory:

Switched networks give dynamic interconnections among the inputs and outputs. Small or medium size systems mostly use crossbar networks. Multistage networks can be expanded to the larger systems, if the increased latency problem can be solved.

Both crossbar switch and multiport memory organization is a single-stage network. Though a single stage network is cheaper to build, but multiple passes may be needed to establish certain connections. A multistage network has more than one stage of switch boxes. These networks should be able to connect any input to any output.

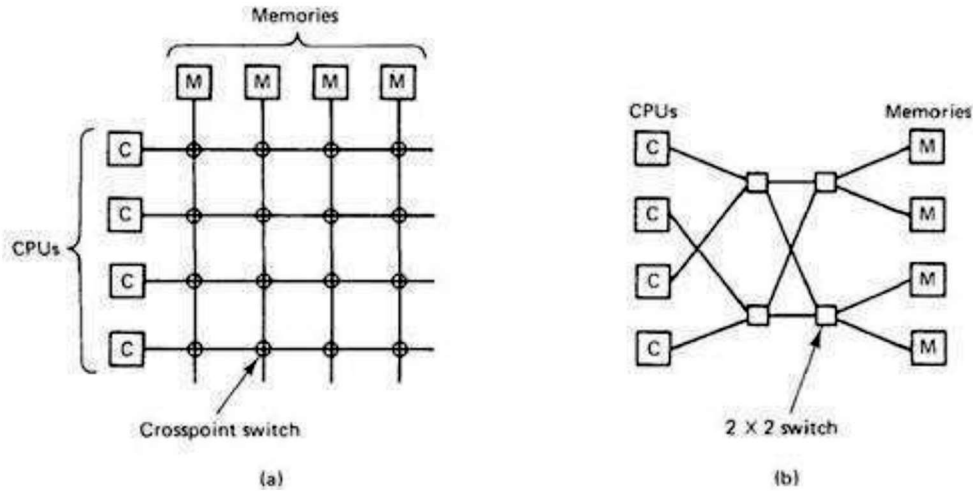

Figure 3: Cross Bar Switch

Multistage and Combining Networks:

Multistage networks or multistage interconnection networks are a class of high-speed computer networks which is mainly composed of processing elements on one end of the network and memory elements on the other end, connected by switching elements.

These networks are applied to build larger multiprocessor systems. This includes Omega Network, Butterfly Network and many more.

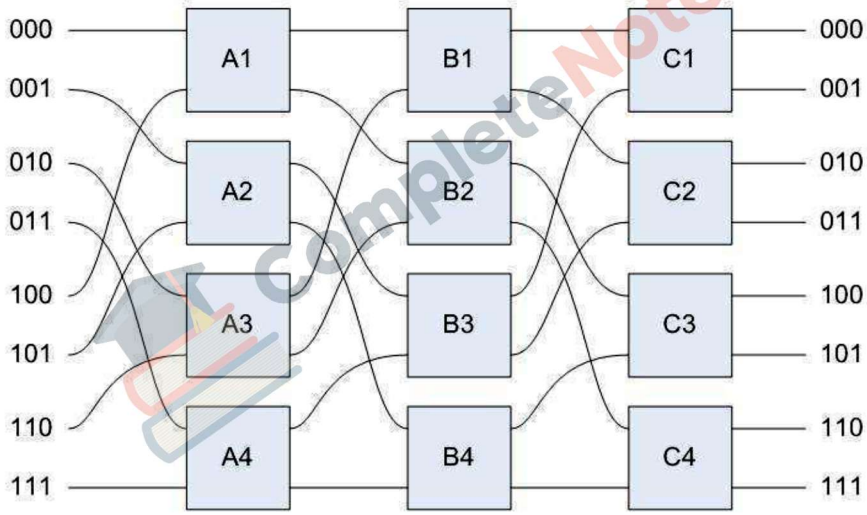

Figure 4: Multistage and Combining Networks

Multi-computers:

Multi-computers are distributed memory MIMD architectures. The following diagram shows a conceptual Multicomputer.

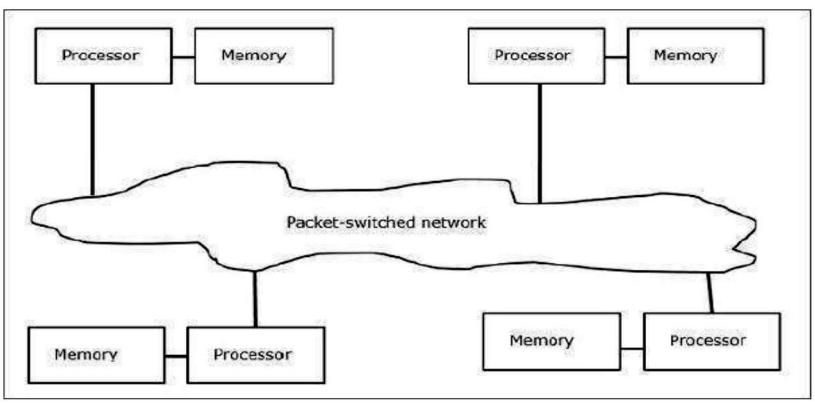

Figure 5: Multi-Computers Interconnection

Multi-computers are message-passing machines which apply packet switching method to exchange data. Here, each processor has a private memory, but no global address space as a processor can access only its own local memory. So, communication is not transparent: here programmers have to explicitly put communication primitives in their code.

Having no globally accessible memory is a drawback of multi-computers. This can be solved by using the following two schemes –

Virtual Shared Memory (VSM)

Shared Virtual Memory (SVM)

In these schemes, the application programmer assumes a big shared memory which is globally addressable. If required, the memory references made by applications are translated into the message-passing paradigm.

Virtual Shared Memory (VSM)

VSM is a hardware implementation. So, the virtual memory system of the Operating System is transparently implemented on top of VSM. So, the operating system thinks it is running on a machine with a shared memory.

Shared Virtual Memory (SVM)

SVM is a software implementation at the Operating System level with hardware support from the Memory Management Unit (MMU) of the processor. Here, the unit of sharing is Operating System memory pages.

If a processor addresses a particular memory location, the MMU determines whether the memory page associated with the memory access is in the local memory or not. If the page is not in the memory, in a normal computer system it is swapped in from the disk by the Operating System. But, in SVM, the Operating System fetches the page from the remote node which owns that particular page.

Inter-processor Arbitration:

Computer system needs buses to facilitate the transfer of information between its various components. For example, even in a uni-processor system, if the CPU has to access a memory location, it sends the address of the memory location on the address bus. This address activates a memory chip. The CPU then sends a red signal through the control bus, in the response of which the memory puts the data on the address bus. This address activates a memory chip. The CPU then sends a read signal through the control bus, in the response of which the memory puts the data on the data bus. Similarly, in a multiprocessor system, if any processor has to read a memory location from the shared areas, it follows the similar routine.

There are buses that transfer data between the CPUs and memory. These are called memory buses. An I/O bus is used to transfer data to and from input and output devices. A bus that connects major components in a multiprocessor system, such as CPUs, I/Os, and memory is called system bus. A processor, in a multiprocessor system, requests the access of a component through the system bus. In case there is no processor accessing the bus at that time, it is given then control of the bus immediately. If there is a second processor utilizing the bus, then this processor has to wait for the bus to be freed. If at any time, there is request for the services of the bus by more than one processor, then the arbitration is performed to resolve the conflict. A bus controller is placed between the local bus and the system bus to handle this.

Inter-Processor Communication and Synchronization:

In computer science, inter-process communication or inter process communication (IPC) refers specifically to the mechanisms an operating system provides to allow the processes to manage shared data. Typically, applications can use IPC, categorized as clients and servers, where the client requests data and the server responds to client requests. Many applications are both clients and servers, as commonly seen in distributed computing. Methods for doing IPC are divided into categories which vary based on software

requirements, such as performance and modularity requirements, and system circumstances, such as network bandwidth and latency.

In order to cooperate concurrently executing processes must communicate and synchronize. Inter process communication is based on the use of shared variables (variables that can be referenced by more than one process) or message passing.

Process Synchronization:

Process Synchronization means sharing system resources by processes in such a way that, Concurrent access to shared data is handled thereby minimizing the chance of inconsistent data. Maintaining data consistency demands mechanisms to ensure synchronized execution of cooperating processes. Process Synchronization was introduced to handle problems that arose while multiple process executions. Synchronization is often necessary when processes communicate. To make this concept clearer, consider the batch operating system again. A shared buffer is used for communication between the leader process and the executor process. These processes must be synchronized so that, for example, the executor process never attempts to read data from the input if the buffer is empty.

Depending on the solution, an IPC mechanism may provide synchronization or leave it up to processes and threads to communicate amongst themselves (e.g. via shared memory).

While synchronization will include some information (e.g. whether or not the lock is enabled, a count of processes waiting, etc.) it is not primarily an information-passing communication mechanism per se.

Examples of synchronization primitives are:

- Semaphore
- Spinlock
- Barrier
- Mutual exclusion:

Memory in Multiprocessor System:

In computer hardware, shared memory refers to a (typically large) block of random access memory (RAM) that can be accessed by several different central processing units (CPUs) in a multiprocessor computer system.

Shared-Memory Multicomputer:

Three most common shared memory multiprocessors models are –

1. Uniform Memory Access (UMA):

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.

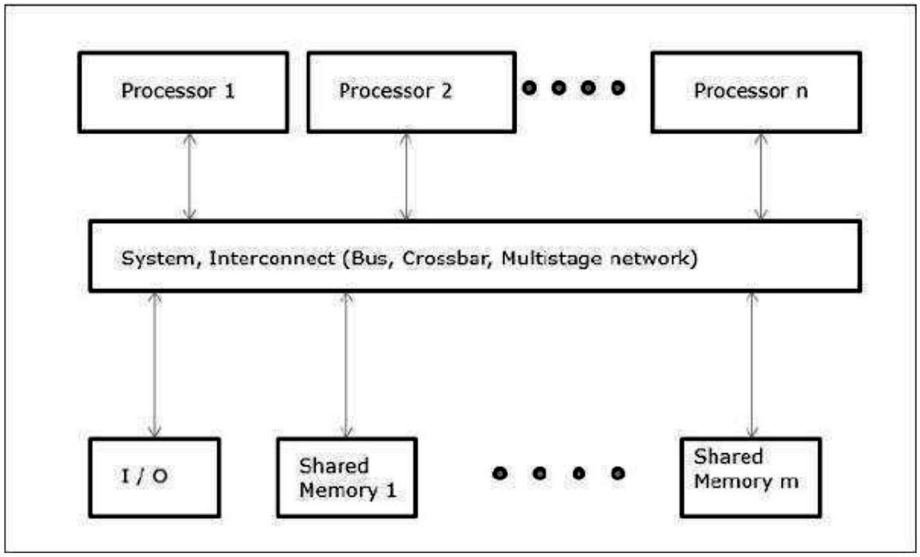

Figure 6: Uniform Memory Access

2. 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.

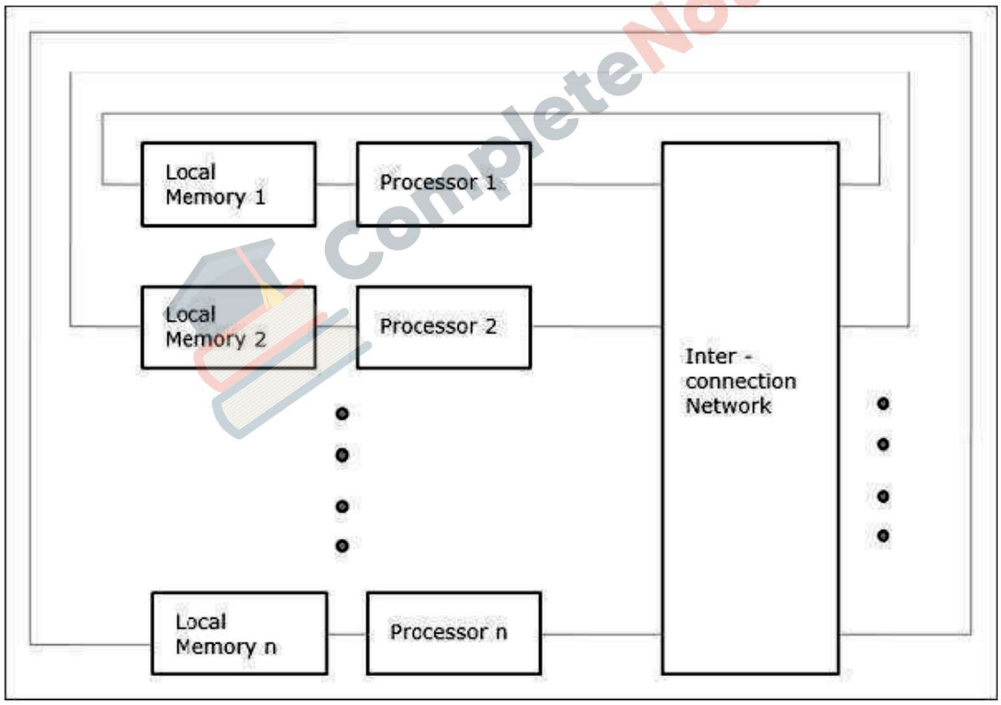

Figure 7: Non-uniform Memory Access

3. 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.

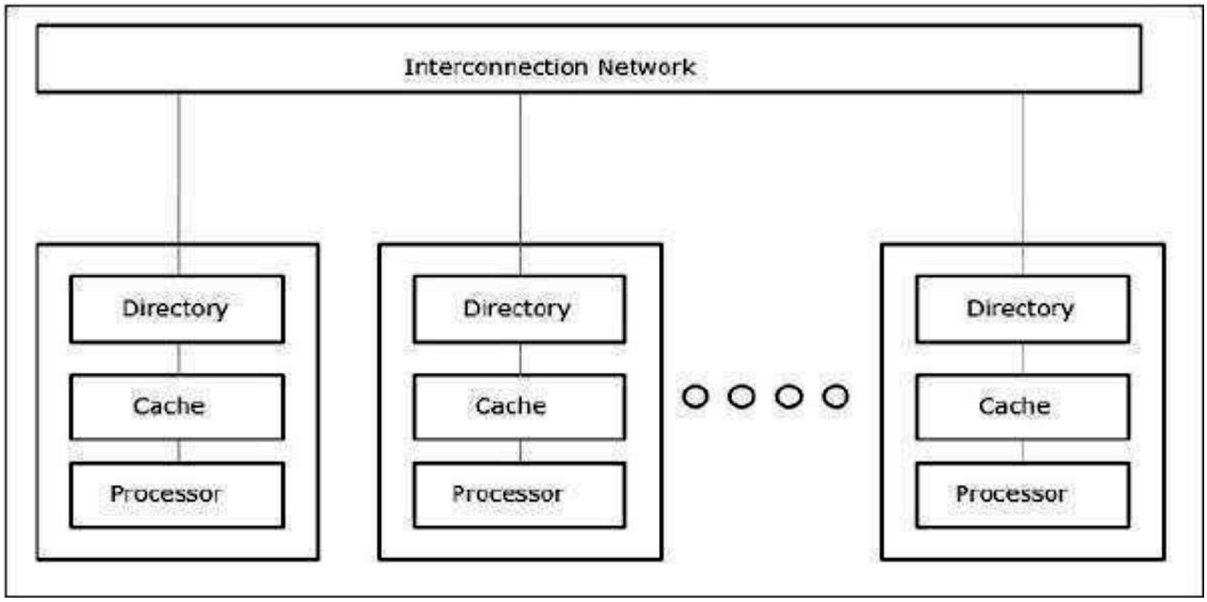

Figure 8: Cache only Memory Architecture

Concept of Pipeline:

- Pipelining is the process of accumulating instruction from the processor through a pipeline. It allows storing and executing instructions in an orderly process. It is also known as pipeline processing.
- Pipelining is a technique where multiple instructions are overlapped during execution. Pipeline is divided into stages and these stages are connected with one another to form a pipe like structure. Instructions enter from one end and exit from another end.
- Pipelining increases the overall instruction throughput.
- In pipeline system, each segment consists of an input register followed by a combinational circuit. The register is used to hold data and combinational circuit performs operations on it. The output of combinational circuit is applied to the input register of the next segment.

Types of Pipeline:

It is divided into 2 categories:

1. Arithmetic Pipeline:

Arithmetic pipelines are usually found in most of the computers. They are used for floating point operations, multiplication of fixed point numbers etc. For example: The input to the Floating Point Adder pipeline is:

X = A*2^a

Y = B*2^b

Here A and B are mantissas (significant digit of floating point numbers), while a and b are exponents.

The floating point addition and subtraction is done in 4 parts:

1. Compare the exponents.
2. Align the mantissas.
3. Add or subtract mantissas
4. Produce the result.

Registers are used for storing the intermediate results between the above operations.

2. Instruction Pipeline:

In this a stream of instructions can be executed by overlapping fetch, decode and execute phases of an instruction cycle. This type of technique is used to increase the throughput of the computer system.

An instruction pipeline reads instruction from the memory while previous instructions are being executed in other segments of the pipeline. Thus we can execute multiple instructions simultaneously. The pipeline will be more efficient if the instruction cycle is divided into segments of equal duration.

Pipeline Conflicts:

There are some factors that cause the pipeline to deviate its normal performance. Some of these factors are given below:

1. Timing Variations:

All stages cannot take same amount of time. This problem generally occurs in instruction processing where different instructions have different operand requirements and thus different processing time.

2. Data Hazards:

When several instructions are in partial execution, and if they reference same data then the problem arises. We must ensure that next instruction does not attempt to access data before the current instruction, because this will lead to incorrect results.

3. Branching:

In order to fetch and execute the next instruction, we must know what that instruction is. If the present instruction is a conditional branch, and its result will lead us to the next instruction, then the next instruction may not be known until the current one is processed.

4. Interrupts: Interrupts set unwanted instruction into the instruction stream. Interrupts effect the execution of instruction.

5. Data Dependency:

It arises when an instruction depends upon the result of a previous instruction but this result is not yet available.

Advantages of Pipelining

1. The cycle time of the processor is reduced.
2. It increases the throughput of the system
3. It makes the system reliable.
4. Disadvantages of Pipelining
5. The design of pipelined processor is complex and costly to manufacture.
6. The instruction latency is more.

Vector Processing:

There is a class of computational problems that are beyond the capabilities of a conventional computer. These problems require vast number of computations on multiple data items that will take a conventional computer (with scalar processor) days or even weeks to complete. Such complex instructions, which operate on multiple data at the same time, requires a better way of instruction execution, which was achieved by Vector processors.

Scalar CPUs can manipulate one or two data items at a time, which is not very efficient. Also, simple instructions like ADD A to B, and store into C are not practically efficient. Addresses are used to point to the memory location where the data to be operated will be found, which leads to added overhead of data lookup. So until the data is found, the CPU would be sitting ideal, which is a big performance issue.

Hence, the concept of **Instruction Pipeline** comes into picture, in which the instruction passes through several sub-units in turn. These sub-units perform various independent functions, for example: the first one decodes the instruction, the second sub-unit fetches the data and the third sub-unit performs the math itself. Therefore, while the data is fetched for one instruction, CPU does not sit idle; it rather works on decoding the next instruction set, ending up working like an assembly line.

Vector processor, not only use Instruction pipeline, but it also pipelines the data, working on multiple data at the same time.

In vector processor a single instruction, can ask for multiple data operations, which saves time, as instruction is decoded once, and then it keeps on operating on different data items.

Applications of Vector Processors:

Computer with vector processing capabilities are in demand in specialized applications. The following are some areas where vector processing is used:

1. Petroleum exploration.
2. Medical diagnosis.
3. Data analysis.
4. Weather forecasting.
5. Aerodynamics and space flight simulations.
6. Image processing.
7. Artificial intelligence.

Array Processors:

Array processors are also known as multiprocessors or vector processors. They perform computations on large arrays of data. Thus, they are used to improve the performance of the computer.

Why use the Array Processor:

- An array processor increases the overall instruction processing speed.
- As most of the Array processors operate asynchronously from the host CPU, hence it improves the overall capacity of the system.
- Array Processors has its own local memory, hence providing extra memory for systems with low memory.

There are basically two types of array processors:

Attached Array Processors:

An attached array processor is a processor which is attached to a general purpose computer and its purpose is to enhance and improve the performance of that computer in numerical computational tasks. It achieves high performance by means of parallel processing with multiple functional units.

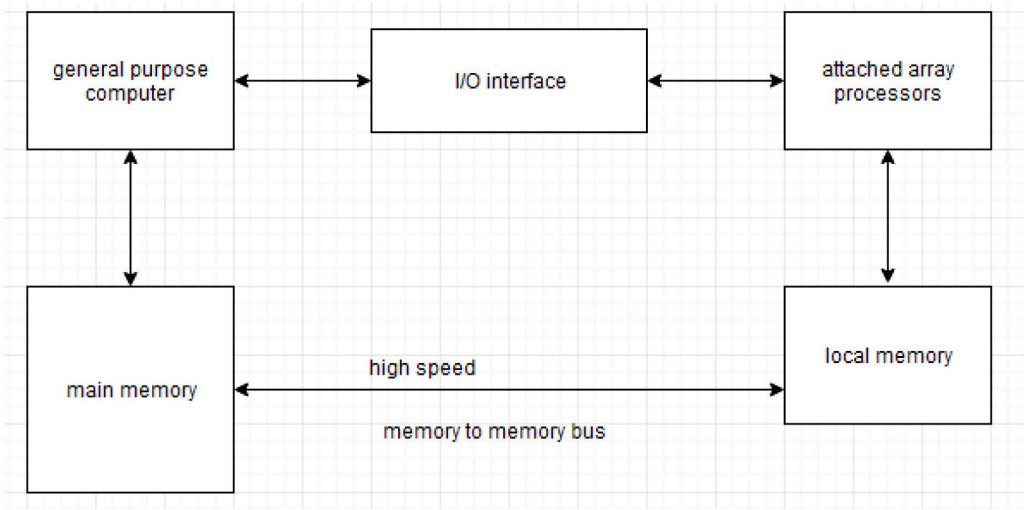

Figure 9: Attached Array Processors

SIMD Array Processors:

SIMD is the organization of a single computer containing multiple processors operating in parallel. The processing units are made to operate under the control of a common control unit, thus providing a single instruction stream and multiple data streams.

A general block diagram of an array processor is shown below. It contains a set of identical processing elements (PE's), each of which is having a local memory M. Each processor element includes an **ALU**and **registers**. The master control unit controls all the operations of the processor elements. It also decodes the instructions and determines how the instruction is to be executed.

The main memory is used for storing the program. The control unit is responsible for fetching the instructions. Vector instructions are send to all PE's simultaneously and results are returned to the memory.

The best known SIMD array processor is the **ILLIAC IV** computer developed by the **Burroughs corps**. SIMD processors are highly specialized computers. They are only suitable for numerical problems that can be expressed in vector or matrix form and they are not suitable for other types of computations.

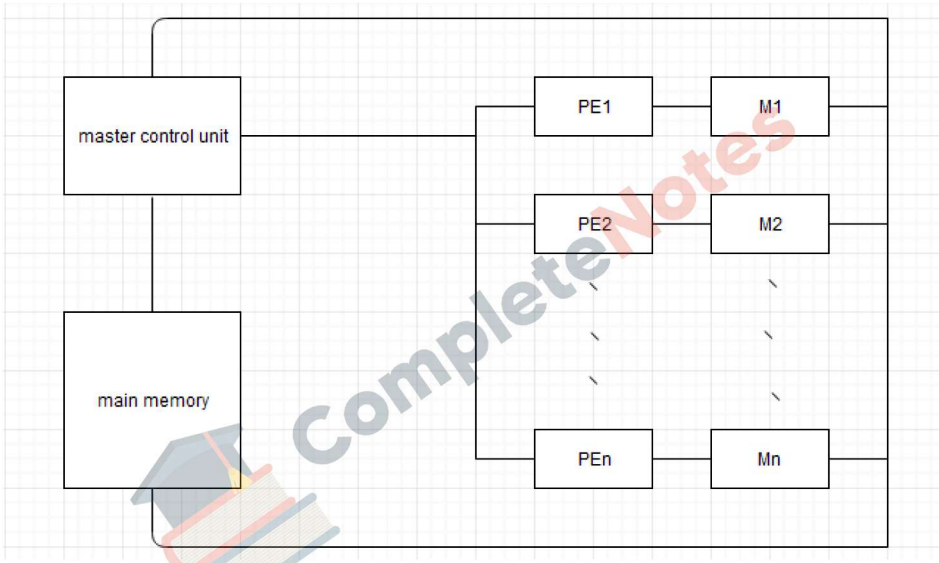

Figure 10: SIMD Array Processors

Type of Multicore Processors:

Ideally, a dual core processor is nearly twice as powerful as a single core processor. In practice, performance gains are said to be about fifty percent: a dual core processor is likely to be about one-and-a-half times as powerful as a single core processor.

Multi-core processing is a growing industry trend as single-core processors rapidly reach the physical limits of possible complexity and speed. Most current systems are multi-core. Systems with a large number of processor cores -- tens or hundreds -- are sometimes referred to as many-core or massively multi-core systems.

Inter-Process Arbitration:

Computer system needs buses to facilitate the transfer of information between its various components. For example, even in a uniprocessor system, if the CPU has to access a memory location, it sends the address of the memory location on the address bus. This address activates a memory chip. The CPU then sends a red signal through the control bus, in the response of which the memory puts the data on the address bus. This address activates a memory chip. The CPU then sends a read signal through the control bus, in the response of which the memory puts the data on the data bus. Similarly, in a multiprocessor system, if any processor has to read a memory location from the shared areas, it follows the similar

routine.

There are buses that transfer data between the CPUs and memory. These are called memory buses. An I/O bus is used to transfer data to and from input and output devices. A bus that connects major components in a multiprocessor system, such as CPUs, I/O's, and memory is called system bus. A processor, in a multiprocessor system, requests the access of a component through the system bus. In case there is no processor accessing the bus at that time, it is given then control of the bus immediately. If there is a second processor utilizing the bus, then this processor has to wait for the bus to be freed. If at any time, there is request for the services of the bus by more than one processor, then the arbitration is performed to resolve the conflict. A bus controller is placed between the local bus and the system bus to handle this.

RISC Processor:

- It is known as Reduced Instruction Set Computer. It is a type of microprocessor that has a limited number of instructions.
- They can execute their instructions very fast because instructions are very small and simple.
- RISC chips require fewer transistors which make them cheaper to design and produce.
- In RISC, the instruction set contains simple and basic instructions from which more complex instruction can be produced.
- Most instructions complete in one cycle, which allows the processor to handle many instructions at the same time.
- In this instructions are register based and data transfer takes place from register to register.

CISC Processor:

- It is known as Complex Instruction Set Computer.
- It was first developed by Intel.
- It contains large number of complex instructions.
- In this instructions are not register based.
- Instructions cannot be completed in one machine cycle.
- Data transfer is from memory to memory.
- Micro programmed control unit is found in CISC.
- Also they have variable instruction formats.

Table 1: Comparison between RISC and CISC Processor

Architectural Characteristics	Complex Instruction Set Computer(CISC)	Reduced Instruction Set Computer(RISC)
Instruction size and format	Large set of instructions with variable formats (16-64 bits per instruction).	Small set of instructions with fixed format (32 bit).
Data transfer	Memory to memory.	Register to register.
CPU control	Most micro coded using control memory (ROM) but modern CISC use hardwired control.	Mostly hardwired without control memory.
Instruction type	Not register based instructions.	Register based instructions.
Memory access	More memory access.	Less memory access.
Clocks	Includes multi-clocks.	Includes single clock.
Instruction Nature	Instructions are complex.	Instructions are reduced and simple.

Study of Multicore Processor – Intel, AMD:

A multi-core processor is a single computing component with two or more independent actual processing units (called "cores"), which are the units that read and execute program instructions. The instructions are ordinary CPU instructions such as add, move data, and branch, but the multiple cores can run multiple instructions at the same time, increasing overall speed for programs amenable to parallel computing

- Intel and AMD have the same design philosophy but different approaches in their micro architecture and implementation.
- AMD technology uses more cores than Intel, but Intel uses Hyper-threading technology to augment the multi-core technology.
- AMD uses Hyper-Transport technology to connect one processor to another and Non-Uniform Memory Access to (NUMA) to access memory. Intel on the other hand usesQuick Path Interconnect technology to connect processor to one another and Memory controller Hub for memory access.
- AMD supports virtualization using Rapid Virtualization Indexing and Direct Connect architecture. While Intel virtualization technology is Virtual Machine Monitor.
- AMD ranked higher in virtualization support than Intel. Moreover, the Quick Path Interconnect in Intel Proliant server have self-healing links and clock failover, hence their technology focuses more on data security while AMD Proliant servers focuses more on power management.

Note: For More Details you may refer:

https://en.wikipedia.org/wiki/Advanced_Micro_Devices

https://www.slideshare.net/anaghvj/intel-core-i7-processors