IOSR Journal of Engineering (IOSR JEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
PP 77-83

Research Paper on Basic Parallel Processing

[1]Ms. RupaYashwantaNagpure, [2]Prof Sandhya Dahake
[1]*Department of Information Technology, MCA Sem 5, GHRIIT, Nagpur*
[2]*Department of Information Technology, GHRIIT, Nagpur*

Abstract: *In computers, parallel processing is the processing of program instructions by dividing them among multiple processor with the objective of running a program in less time.*
The next improvement was multiprogramming. In a multiprogramming system, multiple programs submitted by users were each allowed to use the processor for a short time. To users it appeared that all of the programs were executing at the same time. Problems of resource contention first across in these systems. Explicit requests for resources led to the problem of the deadlock. Competition for resources on machines with no tie-breaking instructions lead to the critical section routine.
Vector processing was another attempt to increase performance by doing more than one thing at a time.
Keywords: *Parallel Processing, multiprogramming, SISD, SIMD, MISD, MIMD*

I. Introduction

Parallel Processing Systems are designed to speed up the execution of programs by dividing the program into multiple fragments and processing these fragments simultaneously. Such systems are multiprocessor systems also known as tightly coupled systems. Parallel systems deal with the simultaneous use of multiple computer resources that can include a single computer with multiple processors, a number of computers connected by a network to form a parallel processing cluster or a combination of both.

Parallel computing is an evolution of serial computing where the jobs are broken into discrete parts that can be executed concurrently. Each part is further broken down to a series of instructions. Instructions from each part execute simultaneously on different CPUs.

Parallel systems are more difficult to program than computers with a single processor because the architecture of parallel computers varies accordingly and the processes of multiple CPUs must be coordinated and synchronized. Several models for connecting processors and memory modules exist, and each topology requires a different programming model. The three models that are most commonly used in building parallel computers include synchronous processors each with its own memory, asynchronous processors each with its own memory and asynchronous processors with a common, shared memory. Flynn has classified the computer systems based on parallelism in the instructions and in the data streams. These are:

1.Single instruction stream, single data stream (SISD).
2.Single instruction stream, multiple data stream (SIMD).
3.Multiple instruction streams, single data stream (MISD).
4.Multiple instruction stream, multiple data stream (MIMD).

The above classification of parallel computing system is focused in terms of two independent factors: the number of data streams that can be simultaneously processed, and the number of instruction streams that can be simultaneously processed. Here 'instruction stream' we mean an algorithm that instructs the computer what to do whereas 'data stream' (i.e. input to an algorithm) we mean the data that are being operated upon.
Even though Flynn has classified the computer systems into four types based on parallelism but only two of them are relevant to parallel computers. These are SIMD and MIMD computers.
SIMD computers are consisting of ‘n' processing units receiving a single stream of instruction from a central control unit and each processing unit operates on a different piece of data. Most SIMD computers operate synchronously using a single global dock. The block diagram of SIMD computer is shown below:

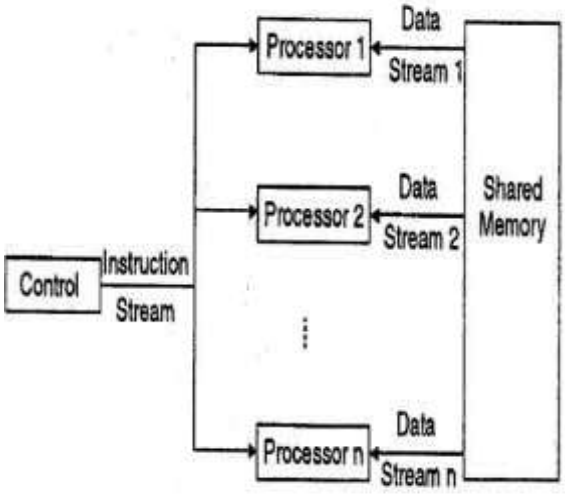

Fig.1 The block diagram of SIMD computer is

MIMD computers are consisting of 'n' processing units; each with its own stream of instruction and each processing unit operate on unit operates on a different piece of data. MIMD is the most powerful computer system that covers the range of multiprocessor systems. The block diagram of MIMD computer is shown.

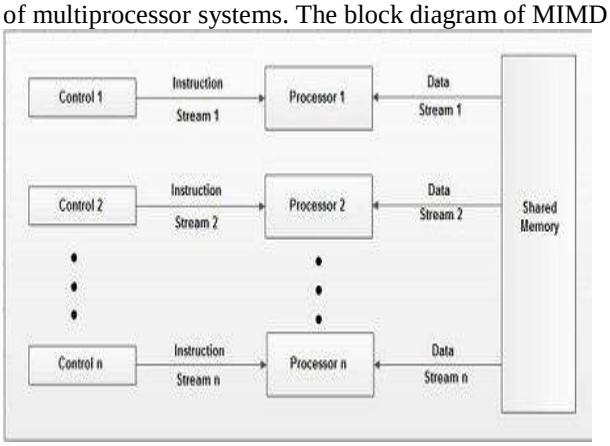

Fig.2 The block diagram of MIMD computer

The SIMD systems are easier to program because it deals with single thread of execution. On the hand, the MIMD machines are more efficient because you can utilize the full machine power.

Parallel operating system are primarily concerned with managing the resources of parallel machines. A parallel computer is a set of processors that are able to work cooperatively to solve a computational problem. So, a parallel computer may be a supercomputer with hundreds or thousands of processors or may be a network of workstations.

A few years ago, parallel computers could be found only in research laboratories and they were used mainly for computation intensive applications like numerical simulations of complex systems. Today, there are a lot of parallel computers available in the market; used to execute both data intensive applications in commerce and computation intensive applications in science and engineering.

Today, new applications arise and demand faster computers. Commercial applications are the most used on parallel computers. A computer that runs such an application; should be able to process large amount of data in sophisticated ways. These applications include graphics, virtual reality, and decision support, parallel databases, medicine diagnosis and so on. We can say with no doubt that commercial applications will define future parallel computers architecture but scientific applications will remain important users of parallel computing technology.

Concurrency becomes a fundamental requirement for algorithms and programs. A program has to be able to use a variable number of processors and also has to be able to run on multiple processors computer architecture. According to Tanenbaum, a distributed system is a set of independent computers that appear to the user like a single one. So, the computers have to be independent and the software has to hide individual computers to the users. MIMD computers and workstations connected through LAN and WAN are examples of

distributed systems. The main difference between parallel systems and distributed systems is the way in which these systems are used. A parallel system uses a set of processing units to solve a single problem A distributed system is used by many users together.

How parallel processing work?

In general, parallel processing means that at least two microprocessors handle parts of an overall task. The concept is pretty simple: A computer scientist divides a complex problem into component parts using special software specifically designed for the task. He or she then assigns each component part to a dedicated processor. Each processor solves its part of the overall computational problem. The software reassembles the data to reach the end conclusion of the original complex problem.

It's a high-tech way of saying that it's easier to get work done if you can share the load. You could divide the load up among different processors housed in the same computer, or you could network several computers together and divide the load up among all of them. There are several ways to achieve the same goal.

Parallel Computer Architectures

Shared memory: all processors can access the same memory

Uniform memory access (UMA):-identical processors-equal access and access times to memory

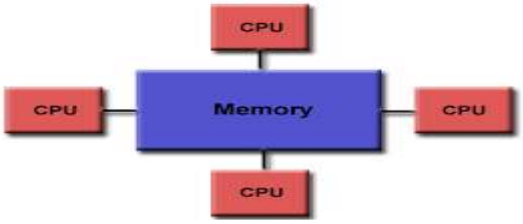

Fig.3 Parallel Computer Architectures

Serial computation

Traditionally software has been written for serial computation:- run on a single computer-instructions are run one after another-only one instruction executed at a time.

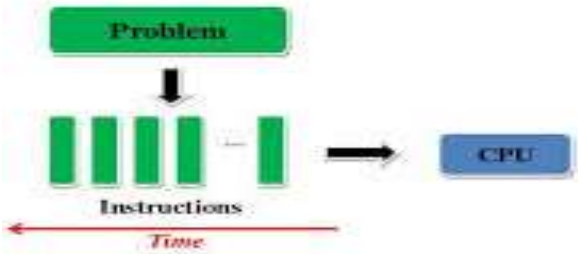

Fig.4 serial Computation

Von Neumann Architecture

John von Neumann first authored the general requirements for an electronic computer in 1945

Data and instructions are stored in memory

Control unit fetches instructions/data from memory, decodes the instructions and then sequentially coordinates operations to accomplish the programmed task.

Arithmetic Unit performs basic arithmetic operations

Input / Output is the interface to the human operator

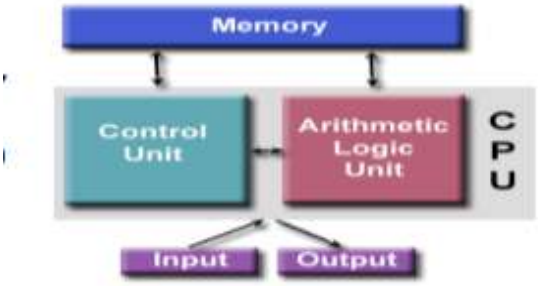

Fig.5 Von Neumann Architecture

Parallel Computing

Simultaneous use of multiple compute sources to solve a single problem

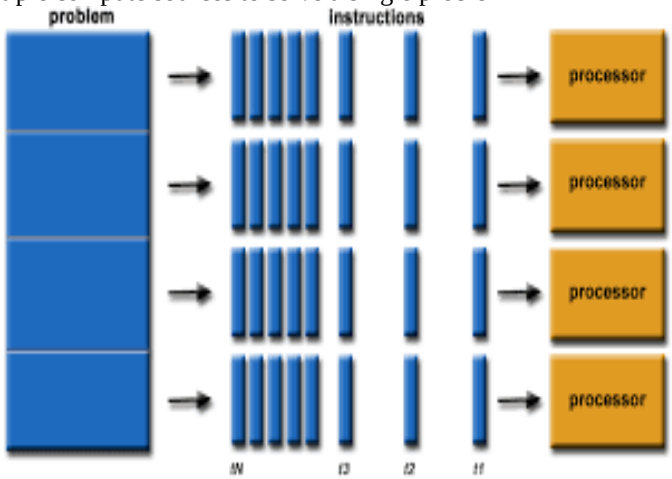

Fig.6 Parallel Computing

Concepts of Parallel Processing

1. Single instruction stream, single data stream (SISD)

Is a computer architecture in which a singleuni-core processor, executes a single instruction stream, to operate on data stored in a singlememory.

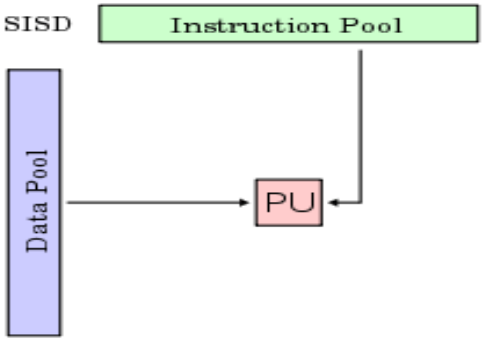

Fig.1.1SISD

Examples: older generation main frames, work stations, PCs

Fig.1.1.a. example of SISD

2. Single instruction stream, multiple data stream (SIMD).

A type of parallel computer. All processing units execute the same instruction at any given clock cycle Each processing unit can operate on a different data element. Two varieties: Processor Arrays and Vector Pipelines. Most modern computers, particularly those with graphics processor units (GPUs) employ SIMD instructions and execution units.

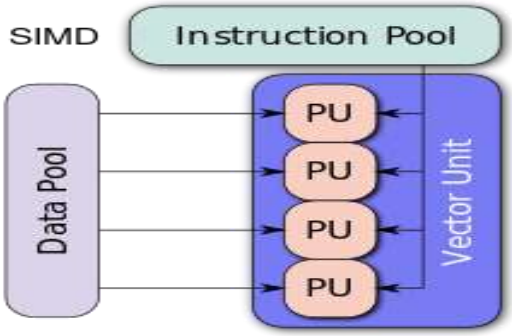

Fig.2.1 SIMD

3. Multiple instruction, single data (MISD)

A single data stream is fed into multiple processing units. Each processing unit operates on the data independently via independent instruction streams.
where many functional units perform different operations on the same data.

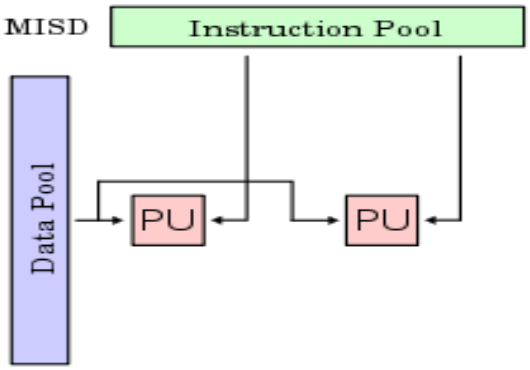

Fig.3.1 MISD

Few actual examples : Carnegie-Mellon C.mmp computer (1971).

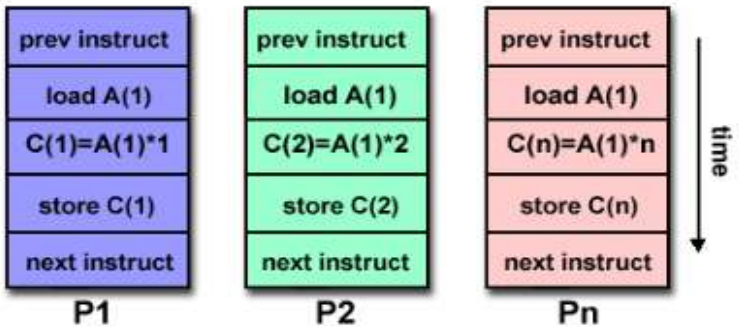

Fig.3.2 example of MISD

4. Multiple instruction, multiple data (MIMD)

Currently, most common type of parallel computer. Every processor may be executing a different instruction stream. Every processor may be working with a different data stream. Execution can be synchronous or asynchronous, deterministic or non-deterministic. MIMD machines can be of either shared memory or distributed memory categories.

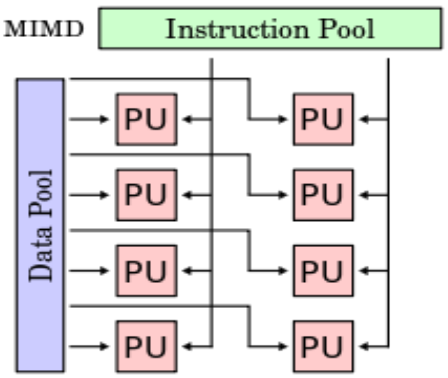

Fig.4.1 MIMD

Examples: most current supercomputers, networked parallel computer clusters and "grids", multi-processor SMP computers, multi-core PCs.
Note: many MIMD architectures also include SIMD execution sub-components

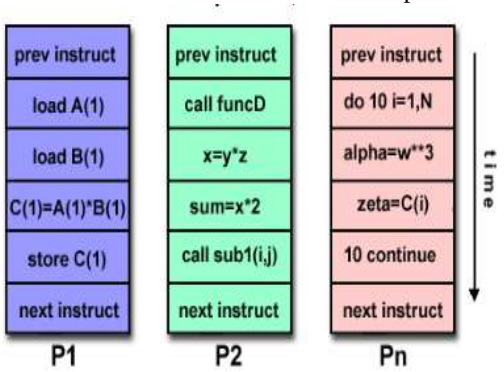

Fig.4.1.2 Example of MIMD

Parallel programming models

- Shared memory
- Threads
- Message Passing
- Data Parallel
- Hybrid

All of these can be implemented on any architecture.

- **Shared memory :-**

Tasks share a common address space, which they read and write asynchronously. Various mechanisms such as locks / semaphores may be used to control access to the shared memory.

Advantage:
No need to explicitly communicate of data between tasks. simplified programming
Disadvantages:
Two known disadvantages are: scalability beyond thirty-two processors is difficult, and the shared memory model is less flexible than the distributed memory model.

Need to take care when managing memory, avoid synchronization conflicts. Harder to control data locality

- **Threads**

A thread can be considered as a subroutine in the main program. Threads communicate with each other through the global memory. Commonly associated with shared memory architectures and operating systems.

- **Message Passing**

A set of tasks that use their own local memory during computation. Data exchange through sending and receiving messages. Data transfer usually requires cooperative operations to be performed by each process. For example, a send operation must have a matching receive operation. MPI (released in 1994) .MPI-2 (released in 1996)

- **Data Parallel**

The data parallel model demonstrates the following characteristics: -
Most of the parallel work performs operations on a data set, organized into a common structure, such as an array.
A set of tasks works collectively on the same data structure, with each task working on a different partition.
Tasks perform the same operation on their partition.
On shared memory architectures, all tasks may have access to the data structure through global memory. On distributed memory architectures the data structure is split up and resides as "chunks" in the local memory of each task.

- **Hybrid**

Combines various models, e.g. MPI/Open MPSingle Program Multiple Data (SPMD)-A single program is executed by all tasks simultaneously. Multiple Program Multiple Data (MPMD)-An MPMD application has multiple executables. Each task can execute the same or different program as other tasks.

References

[1]. https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&uact=8&ved=2ahUKEwjGwbH0xdbhAhUKcCsKHVT4Be0QFjABegQIABAB&url=https%3A%2F%2Fcomputing.llnl.gov%2Ftutorials%2Fparallel_comp%2F&usg=AOvVaw13yn9AyxPEwKPIU2Z2QVNW
[2]. https://www.google.com/search?client=firefox-b-d&q=evolution+of+parallel+computers
[3]. https://computer.howstuffworks.com/parallel-processing.htm
[4]. https://www.sciencedirect.com/science/article/pii/S096007790000223X