HIGH PERFORMANCE COMPUTER SYSTEMS

(HPCS)

compiled by

Dr.N.Gopalakrishna Kini

Professor, Dept. of CSE

MIT, Manipal

INTRODUCTION TO PARALLEL PROCESSING

compiled by
Dr.N.Gopalakrishna Kini
Professor, Dept. of CSE
MIT, Manipal

Architectural classification Schemes:

(Flynn's classification) : -

Flynn's classification scheme is based on the notion of a stream of information. Flynn considers the organization of a computer system by the number of instruction and data items.

Two types of information flow into a processor: instructions and data.
The sequence of instruction read from memory constitutes an **instruction stream (IS).**
The operations performed on the data in the processor constitute a **data steam (DS).**

According to Flynn's classification, either of the instruction or data streams can be single or multiple and is classified into the following four distinct categories:

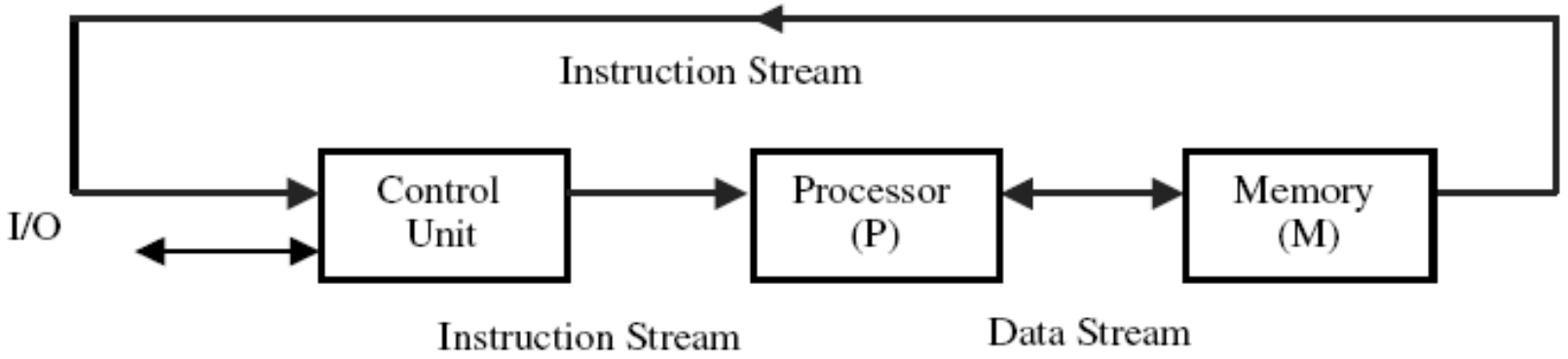
i) single-instruction single-data streams (SISD);

ii) single-instruction multiple-data streams (SIMD);

iii) multiple-instruction single-data streams (MISD);

iv) multiple-instruction multiple-data streams (MIMD).

i) Single-instruction single-data streams (SISD)

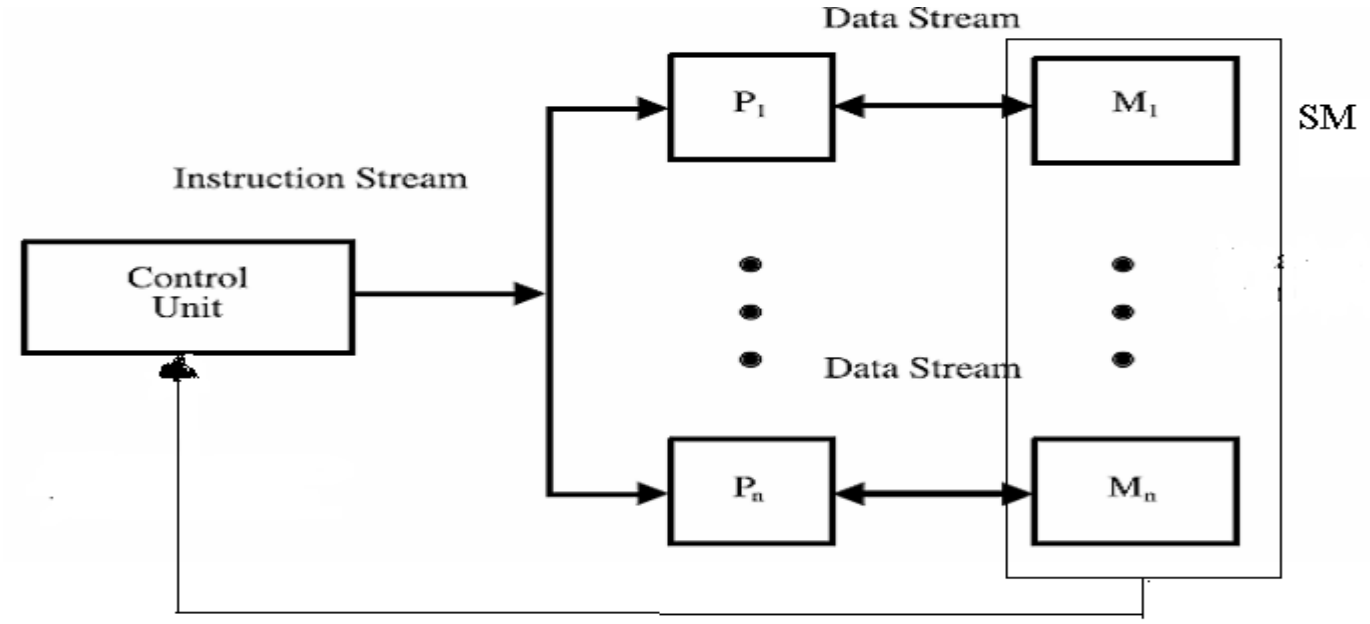


Instructions are executed sequentially but may be overlapped in their execution stages (Pipelining).

Most SISD computers are pipelined. An SISD computer may have more than one functional unit in it.

All functional units are under the control of single control unit.

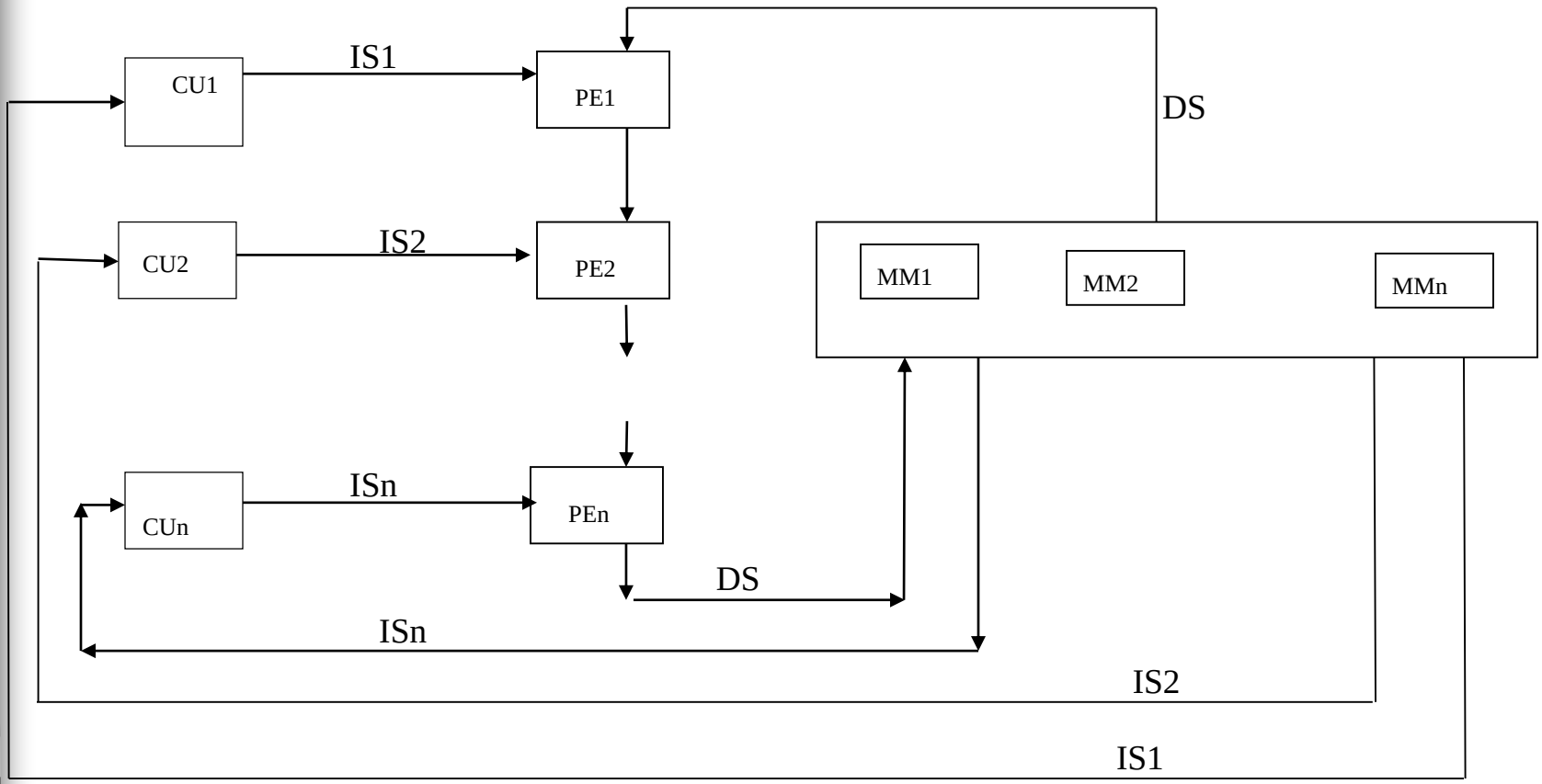
ii) Single-instruction multiple-data streams (SIMD)



This classification corresponds to array processors. There are multiple PEs supervised by the same control unit.

All PEs receive the same insruction broadcast from the control unit but operate on different data sets from distinct data streams. The shared memory subsystem may contain multiple modules.

iii) multiple-instruction single-data streams(MISD)

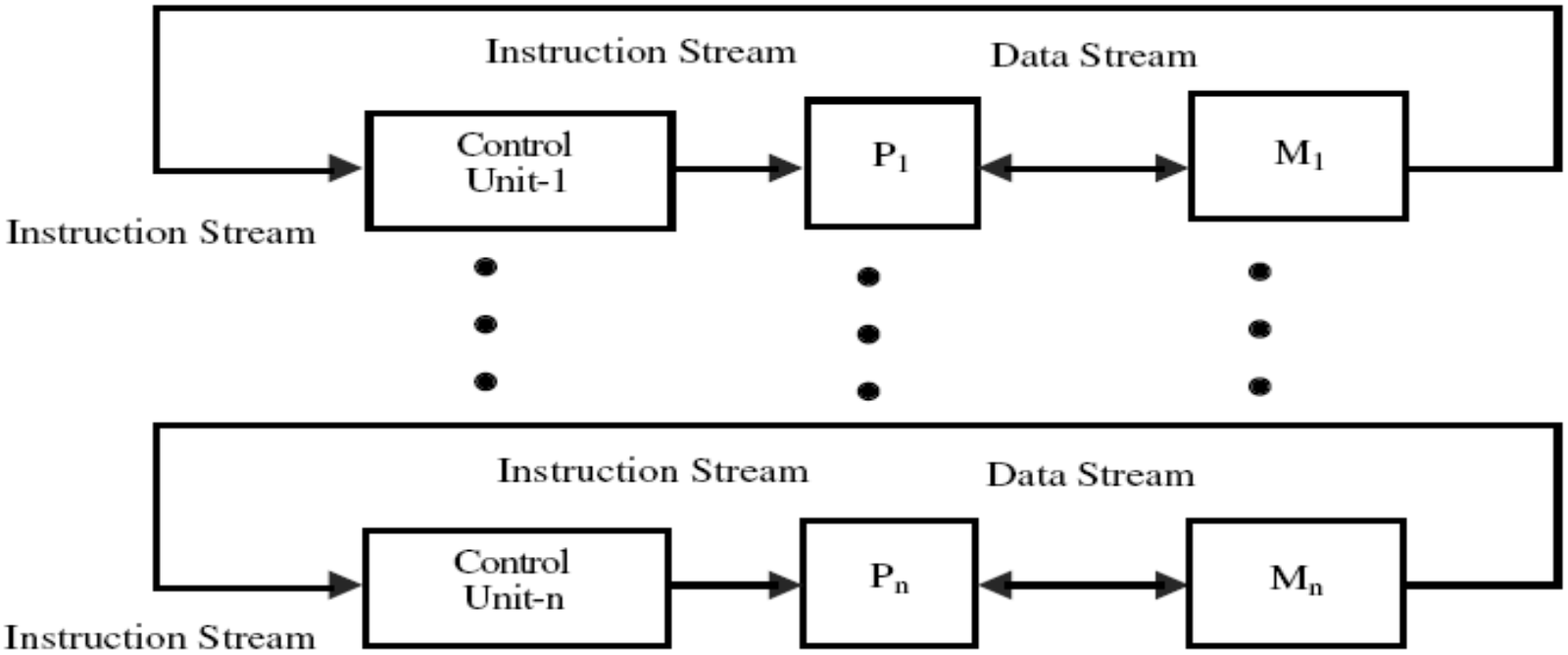


There are n PEs, each receiving distinct instructions operating over the same data stream and its derivatives.

The result of the one PE become the input of the next PE in the micropipe.

This sturcture has only theoretical study but impractical.

iv) Multiple-instruction multiple-data streams(MIMD



Most multiprocessor systems can be classified in this category.

MIMD computer implies interaction among n PEs because all memory streams are derived from same data space shared by all PEs.

If n data streams are derived from disjointed subspaces of shared memory then we have multiple SISD (MIMD) operation which is nothing but a set of n independent SISD uniprocessor systems.

If the degree of interaction among the processors is high then MIMD computer is said to be tightly coupled else loosely coupled.

Most commercial MIMD computers are loosely coupled.

What is Parallel Processing?

It is an efficient form of information processing which exploits the concurrency of execution.

Types of Concurrent Events:

There are 3 types of concurrent events:-

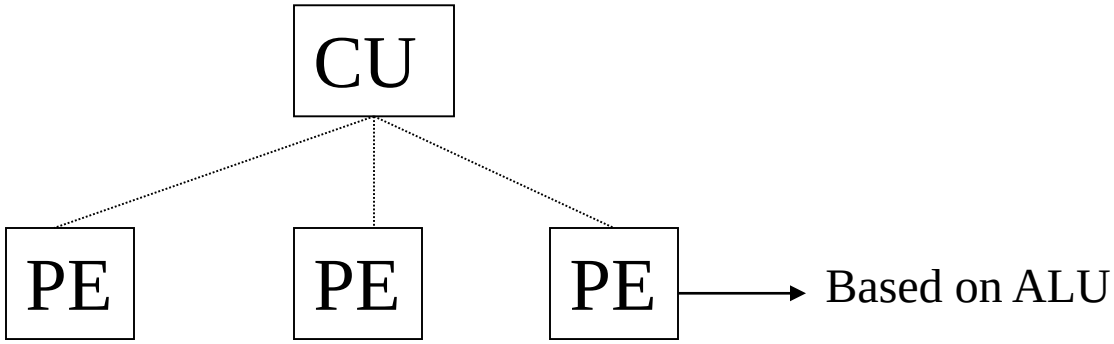
1. **Parallel Event or Synchronous Event :-**

(Type of concurrency is parallelism)

It may occur in multiple resources during the same time interval.

Example

Array/Vector Processors



2. Simultaneous Event or Asynchronous Event :-

(Type of concurrency is simultaneity)

It may occur in multiple resources during the same time instant.

Example

Multiprocessing System

3. Pipelined Event or Overlapped Event :-

It may occur in overlapped spans.

Example

Pipelined Processor

Programmatic Levels of Parallel Processing

Above said three terms implies parallel processing 4 programmatic levels:-

1. Job / Program Level
2. Task / Procedure Level
3. Interinstruction Level
4. Intrainstruction Level

1. Job / Program Level :-

The highest level of parallel processing is conducted among multiple jobs or programs through multiprogramming, time sharing and multiprocessing.

It requires the development of parallel processable algorithms.

The implementation of parallel algorithms depends on the efficient allocation of limited hardware and software resources

to multiple programs being used to solve a large computational problem.

Example: Weather forecasting , oil exploration etc.

2. **Task / Procedure Level** :-

The next highest level of parallel processing is conducted among procedure/tasks within the same program.

This involves the decomposition of the program into multiple tasks or modules.
(for simultaneous execution)

3. Inter instruction level:

The third level of parallel processing is to exploit concurrency among multiple instructions so that they can be executed simultaneously.

Data dependency analysis is often performed to reveal parallellism among instructions. Vectorization may be desired for scalar operations within *DO* loops.(data dependency analysis).

4. Intra instruction Level :-

Intra instruction level exploits faster and concurrent operations within each instruction e.g. use of carry look ahead and carry save address instead of ripple carry address.

Note:

1. Hardware role increases from high to low levels whereas software role increases from low to high levels.
2. As highest job level is conducted algorithmically, lowest level is implemented directly by hardware means.
3. As hardware cost declines and software cost increases , more and more hardware method are replacing the conventional software approaches.

Conclusion :-

Parallel Processing is a combined field of studies which requires a broad knowledge of and experience with all aspects of algorithms, languages, hardware, software and computing alternatives.

Parallel Processing in Uniprocessor Systems

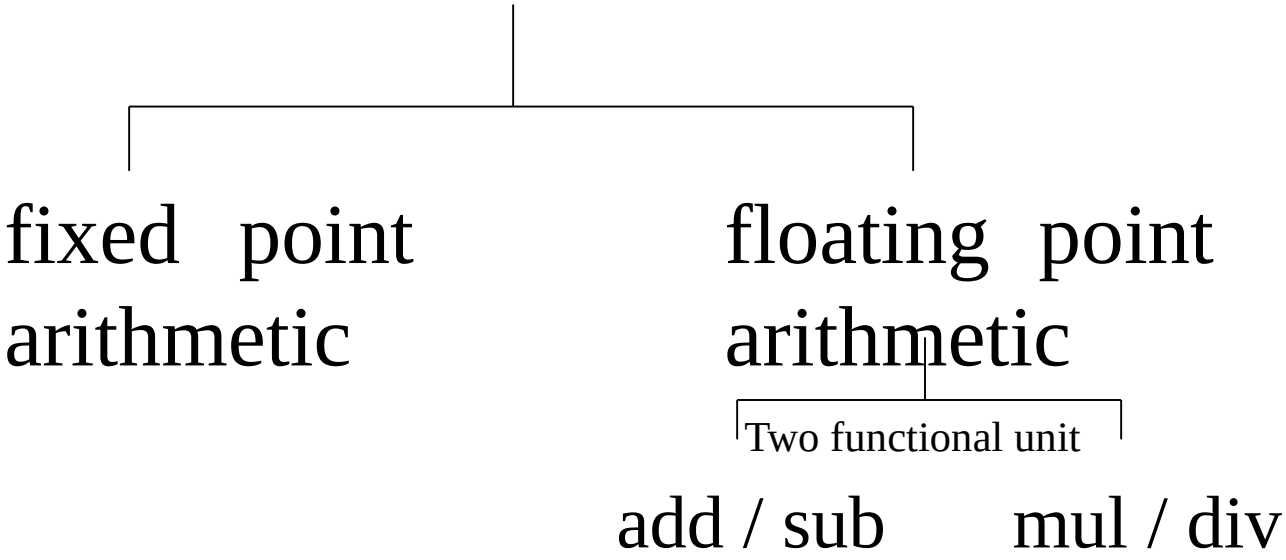
A number of **parallel processing mechanisms** have been developed in uniprocessor computers. We identify them in six categories which are described below.

1. Multiplicity of Functional Units :-

Different ALU functions can be distributed to multiple & specialized functional units which can operate in parallel.

Another example of multifunction uniprocessor is the following. It has two parallel execution units.

IBM 360 / 91



This machine is highly pipelined , multifunctional scientific uniprocessor system.

2. **Parallelism & Pipelining within the CPU** :-

Use of carry-lookahead & carry-save address instead of ripple-carry adders built within the ALUs. The use of multiple functional units is a form of parallelism within the CPU.

Various phases of instruction execution is now pipelined including IF, ID, OF, EX and store result. To facilitate overlapped instruction executions through the pipe, insruction prefetch and data buffering techniques have been developed.

3. Overlapping CPU & I/O Operations :-

DMA is conducted on a *cycle-stealing* basis.

- CDC-6600 has 10 I/O processors of I/O multiprocessing.
- Simultaneous I/O operations & CPU computations can be achieved using separate I/O controllers, channels.

4. Use of hierarchical Memory System :-

A hierarchal memory system can be used to close up the speed gap between the CPU &

& main memory because CPU is 1000 times faster than memory access.

5. **Balancing of Subsystem Bandwidth** :-

Consider the relation

$$t_m < t_m < t_d$$

Bandwidth of a System :-

Bandwidth of a system is defined as the number of operations performed per unit time.

Bandwidth of a memory :-

The memory bandwidth is the number of words

accessed per unit time. It is represented by B_m. If **'W'** is the total number of words accessed per memory cycle t_m then

Mem B.W

$$B_m = \frac{W}{t_m} \quad \text{(words / sec)}$$

Practically, utilized memory BW B_m^u is usually lower than B_m . i.e. $B_m^u \leq B_m$.

$$B_m^u = \frac{B_m}{\sqrt{M}}$$

where M= no. of interleaved memory modules in the memory system

Processor Bandwidth :-

B_p :- maximum CPU computation rate.

B_p^u :- utilized processor bandwidth or the no. of output results per second.

$$B_p^u = \frac{R_w}{T_p} \quad \text{(word result)}$$

R_w :- no of word results.

T_p :- Total CPU time to generate R_w results.

B_d :- Bandwidth of devices. (which is assumed as provided by the vendor).

The following relationships have been observed between the bandwidths of the major subsystems in a high performance uniprocessor.

$$B_m \geq B_m^u \geq B_p \geq B_p^u \geq B_d$$

Due to the unbalanced speeds we need to match the processing power of the three subsystem, we need to match the processing power of the three subsystems.

Two major approaches are described below :-

1. Bandwidth balancing b/w CPU & memory :-

Using fast cache having cache access time $t_c = t_p$.

A block of memory word is moved from main memory to cache, so that immediate instrns/ data can be available most of the time from the cache . Cache serves as a data/ instrn buffer.

2. Bandwidth balancing b/w memory & I/O :-

is done by using I/O channels in b/n them. The I/O channels perform buffering and multiplexing functions to transfer the data from multiple disks into main memory by cycle stealing of CPU. In ideal case, we wish to achieve totally balanced system, in which the entire memory BW matches the BW sum of the processor and I/O devices.

i.e. $B_p^u + B_d = B_m^u$

6a. **Multiprogramming** :-

As we know that some computer programs are CPU bound & some are I/O bound.Whenever a Process **P1** is tied up with I/O operations the system scheduler can switch the CPU to process **P2**.This allows simultaneous execution of several programs in the system.This interleaving of CPU & I/O operations among several programs in a uniprocessor system is called *multiprogramming*, so the total execution time is reduced.

6b. Time sharing :-

In multiprogramming, sometimes a high priority program may occupy the CPU for too long to allow others to share. This problem can be overcome by using a *time-sharing* operating system.The concept extends from multiprogram –ming by assigning fixed or variable time slices to multiple programs. In other words, equal opportunities are given to all programs competing for the use of CPU.

Time sharing is particularly effective when applied to a computer system connected to many interactive terminals.

Parallel computer Architecture:

Parallel computers are those that support parallel processing. Parallel computers are divided into three architectural configurations:

i) Pipelined Computers

ii) Array Processors

iii) Multiprocessor systems

System contains two or more processors of approximately same capabilities.

All processors share access to common set of memory modules, I/O channels and peripheral devices.

The entire system must be controlled by a single operating system providing interaction between processors and their programs at various levels.

Besides the shared memory and I/O devices each processor has its own local memory and private devices.

Interprocessor communication is done through the shared memory or through an interrupt network.

Feng's Classification of Parallel computers (Serial vs. parallel processing)

Handler's Classification (Parallelism vs. pipelining)

What are the applications of Parallel Processing?