Multiprocessor Architecture

PURPLE PETAL EDU HUB

Introduction

A computer systems that include more than one processor are called *multiprocessor systems* or *parallel systems.*

In a multiprocessing system, two or more independent processors are linked together in a coordinated manner. In such systems, instructions from different and independent programs can be processed at the same time, by different processors. The need of multiprocessor system was felt when simultaneous applications were executed, which required high-speed processing. The speed of processing enhanced drastically when these kinds of systems were used. Independent processors in such a system are connected by a high-speed system bus, with each processor having its own cache to improve the performance. Multiprocessor systems have the advantage of redundancy (fault-tolerance); if one processor fails, then the system keeps on functioning, though at a slower speed.

Why Choose a Multiprocessor?

- A single CPU can only go so fast, use more than one CPU to improve performance
- Multiple users
- Multiple applications
- Multi-tasking within an application
- Responsiveness and/or throughput
- Share hardware between CPUs

Processor symmetry

In a **multiprocessing** system, all CPUs may be equal, or some may be reserved for special purposes. A combination of hardware and operating-system software design considerations determine the symmetry (or lack thereof) in a given system. For example, hardware or software considerations may require that only one CPU respond to all hardware interrupts, whereas all other work in the system may be distributed equally among CPUs; or execution of kernel-mode code may be restricted to only one processor (either a specific processor, or only one processor at a time), whereas user-mode code may be executed in any combination of processors. Multiprocessing systems are often easier to design if such restrictions are imposed, but they tend to be less efficient than systems in which all CPUs are utilized.

Instruction and data streams

In multiprocessing, the processors can be used to execute a single sequence of instructions in multiple contexts (single-instruction, multiple-data or SIMD, often used in vector processing), multiple sequences of instructions in a single context (multiple-instruction, single-data or MISD, used for redundancy in fail-safe systems and sometimes applied to describe pipelined processors or hyper-threading), or multiple sequences of instructions in multiple contexts (multiple-instruction, multiple-data or MIMD).

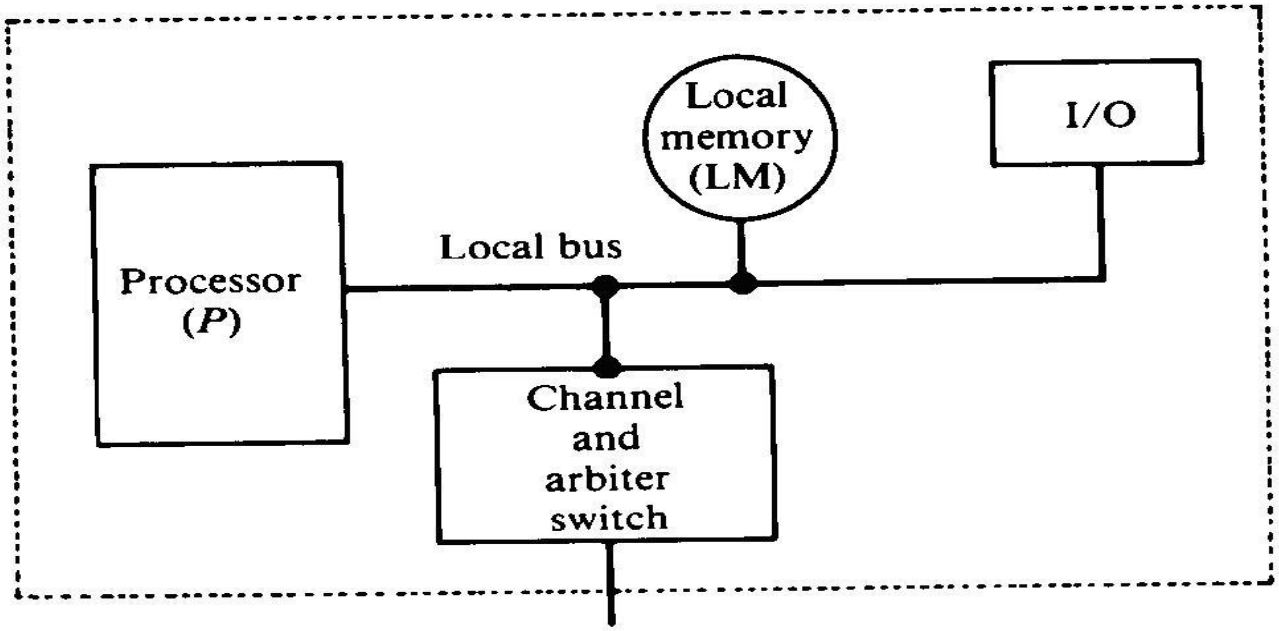
Processor Coupling

The coupling between the processor and memory module or processor and I/O devices , is called processor coupling.

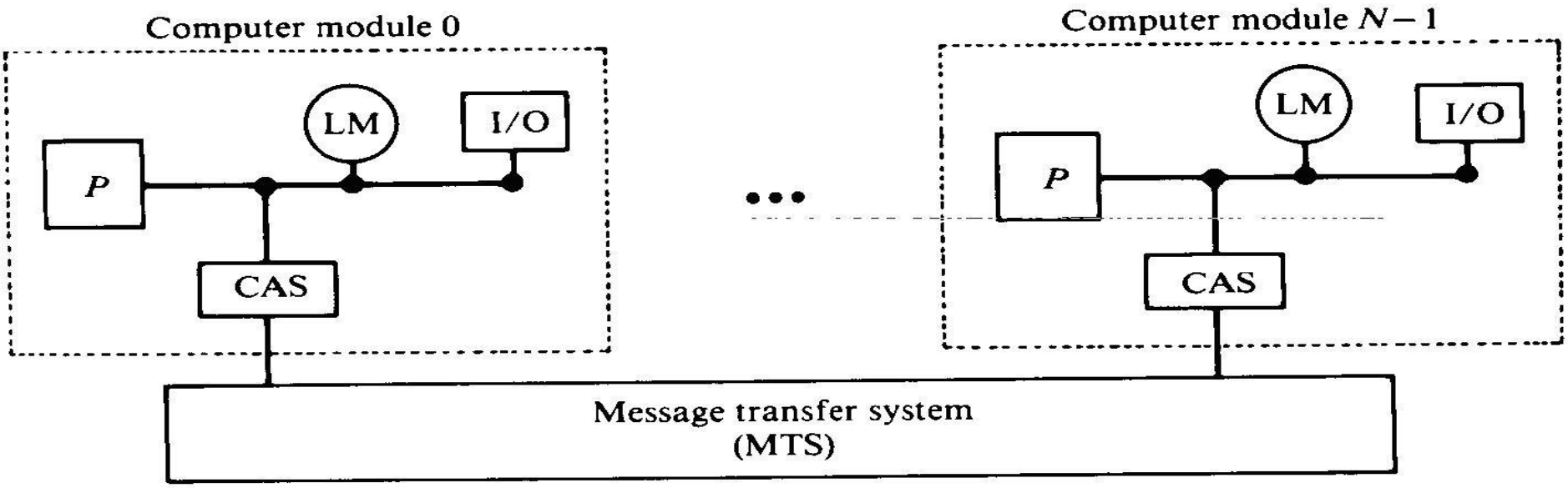
Processor Coupling can be of two types:

1. Loosely Coupled
2. Tightly Coupled

Loosely-coupled multiprocessor systems (often referred to as clusters) are based on multiple standalone single or dual processor commodity computers interconnected via a high speed communication system (Gigabit Ethernet is common). In computing and systems design a **loosely coupled** system is one in which each of its components has, or makes use of, little or no knowledge of the definitions of other separate components.

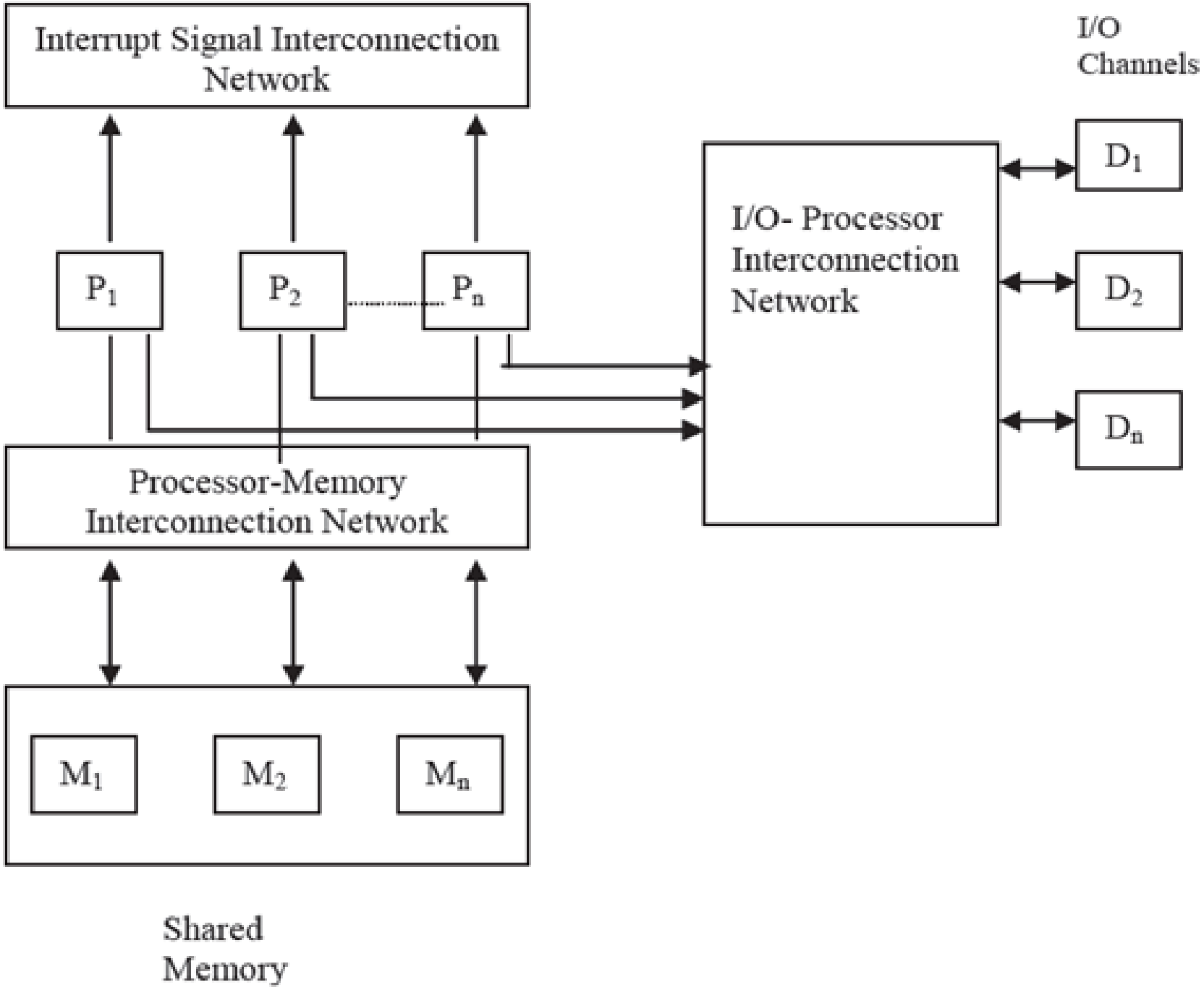


(*a*) A computer module



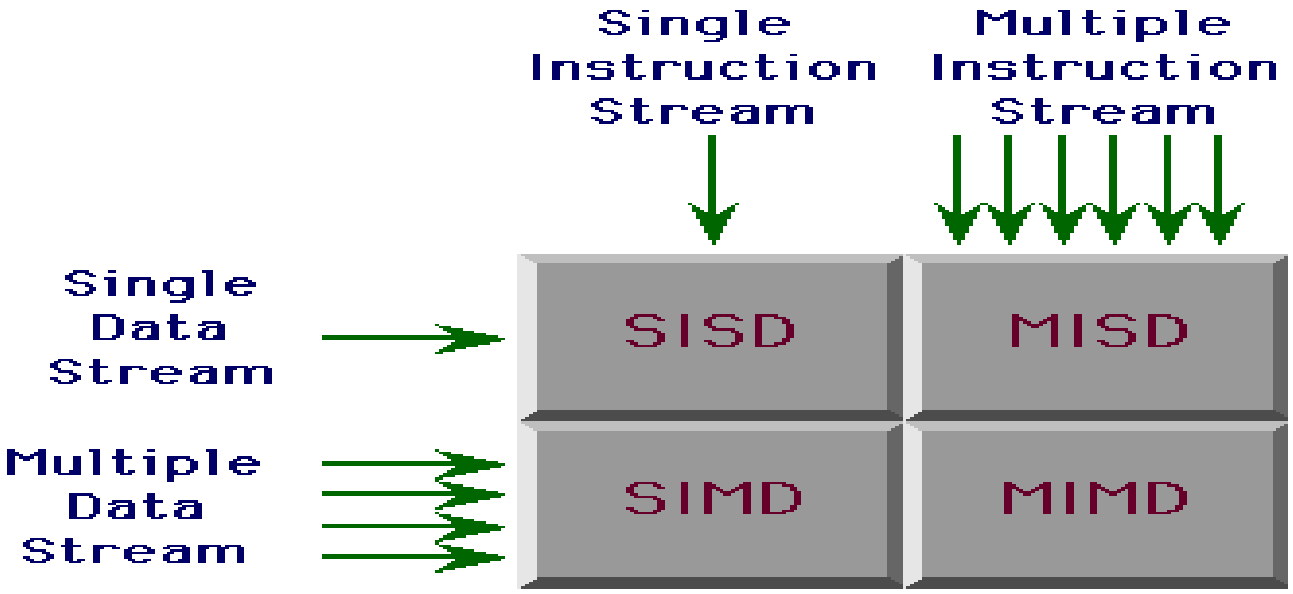
Tightly-coupled systems perform better and are physically smaller than loosely-coupled systems, but have historically required greater initial investments and may depreciate rapidly; nodes in a loosely-coupled system are usually inexpensive commodity computers and can be recycled as independent machines upon retirement from the cluster.

Power consumption is also a consideration. Tightly-coupled systems tend to be much more energy efficient than clusters. This is because considerable economy can be realized by designing components to work together from the beginning in tightly-coupled systems, whereas loosely-coupled systems use components that were not necessarily intended specifically for use in such systems.


Interrupt Signal Interconnection Network
I/O Channels
I/O- Processor Interconnection Network
D_1
D_2
D_n
P_1
P_2
P_n
Processor-Memory Interconnection Network
M_1
M_2
M_n
Shared Memory

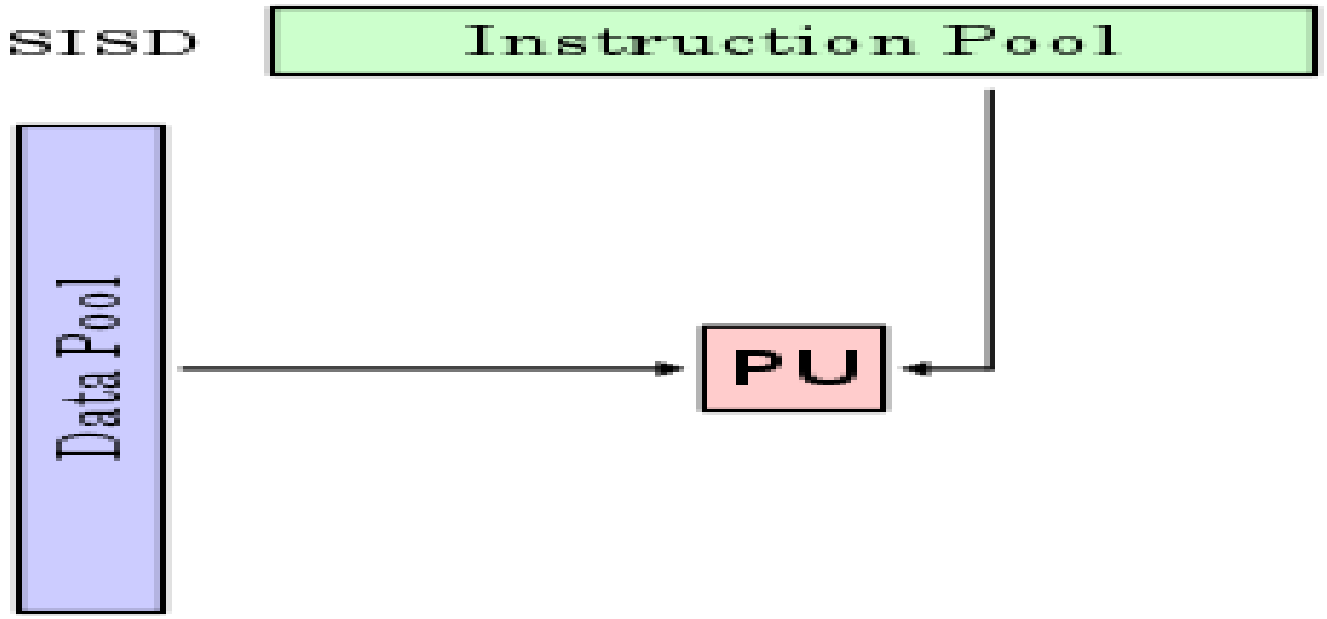
Types of Multiprocessor Systems:

Multiprocessor systems may be classified into types: single instruction stream, single data stream (SISD); single instruction stream, multiple data stream (SIMD); multiple instruction stream, single data stream (MISD): and multiple instruction stream, multiple data stream (MIMD). Systems in the MISD category are rarely built. The other three architectures may be distinguished simply by the differences in their respective instruction cycles:



SISD (Single Instruction Single Data)

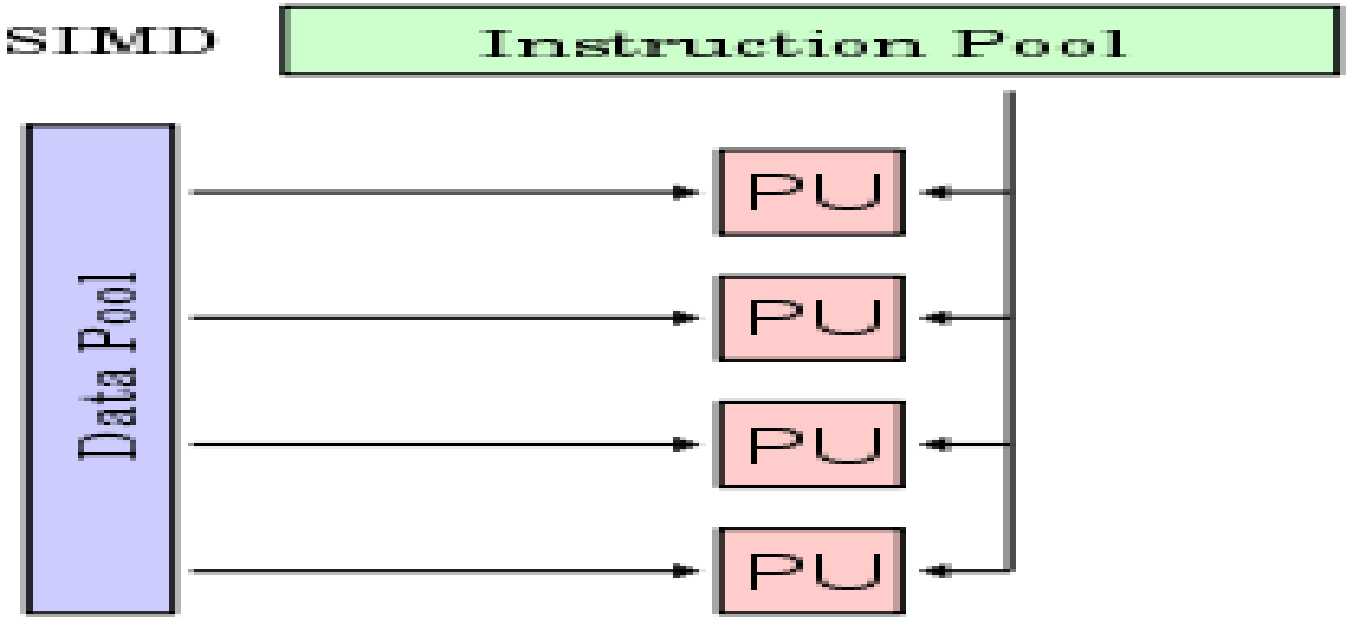
In a single instruction stream, single data stream computer one processor sequentially processes instructions, each instruction processes one data item.



SIMD (Single Instruction Multiple Data)

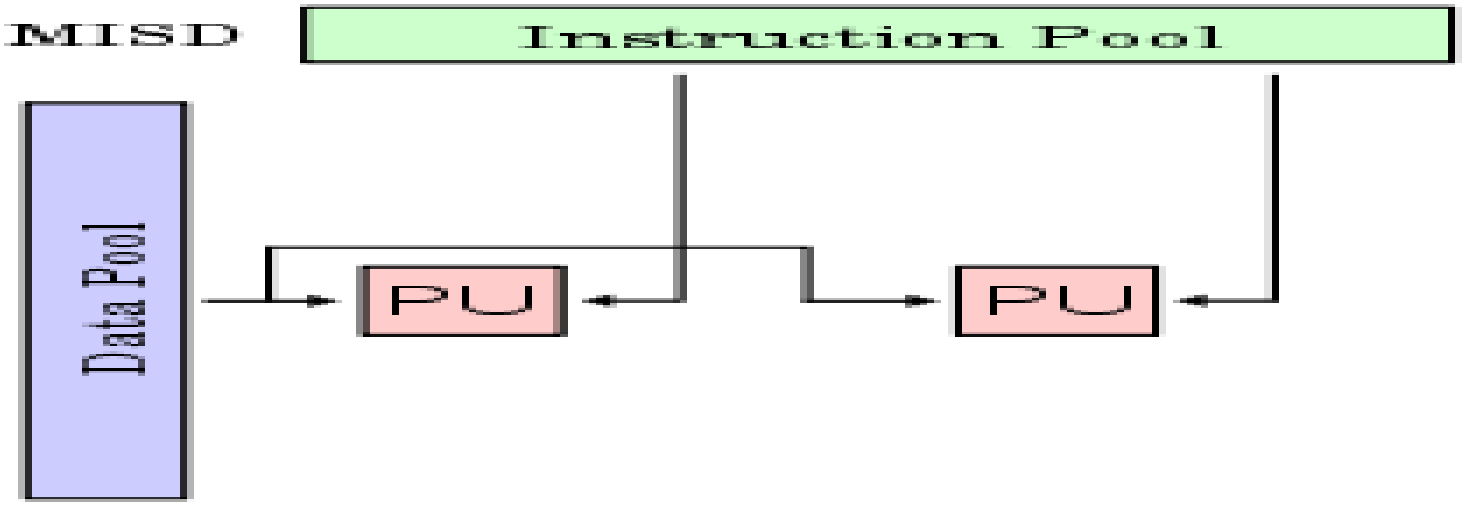
In a single instruction stream, multiple data stream computer one processor handles a stream of instructions, each one of which can perform calculations in parallel on multiple data locations.

SIMD multiprocessing is well suited to parallel or vector processing, in which a very large set of data can be divided into parts that are individually subjected to identical but independent operations. A single instruction stream directs the operation of multiple processing units to perform the same manipulations simultaneously on potentially large amounts of data.



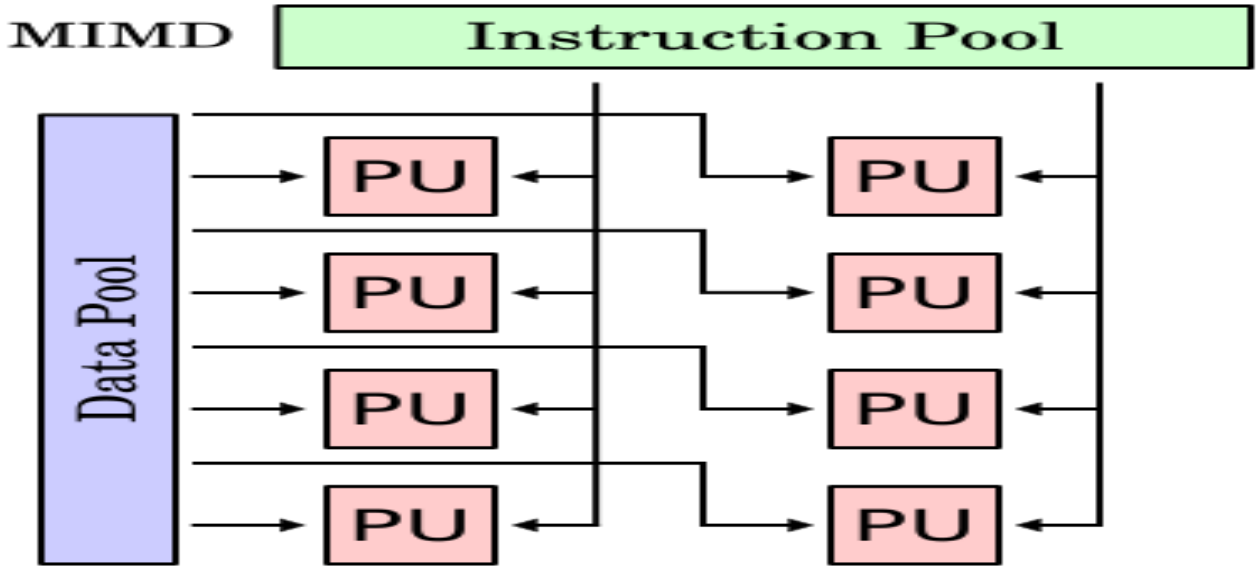
MISD (Multiple Instruction Single Data)

MISD multiprocessing offers mainly the advantage of redundancy, since multiple processing units perform the same tasks on the same data, reducing the chances of incorrect results if one of the units fails. MISD architectures may involve comparisons between processing units to detect failures. Apart from the redundant and fail-safe character of this type of multiprocessing, it has few advantages, and it is very expensive. It does not improve performance. It can be implemented in a way that is transparent to software. It is used in array processors and is implemented in fault tolerant machines.



MIMD (Multiple Instruction Multiple Data)

In an **MIMD** architecture, several instruction cycles may be active at any given time, each independently fetching instructions and operands into multiple processing units and operating on them in a concurrent fashion. This category includes multiple processor systems in which each processor has its own program control, rather than sharing a single control unit.



Conclusion

- Parallel processing is a future technique for higher performance and effectiveness for multiprogrammed workloads.
- MPs combine the difficulties of building complex hardware systems and complex software systems.
- Communication, memory, affinity and throughputs presents an important influence on the systems costs and performances