CSE3009 - Parallel and Distributed Computing

Course Type: LTP Credits: 4

Prepared by

Dr Komarasamy G

Senior Associate Professor

School of Computing Science and Engineering

VIT Bhopal University

Unit-1

Parallelism Fundamentals – Key Concepts and Challenges – Overview of Parallel computing – Flynn's Taxonomy – Multi-Core Processors – Shared vs Distributed memory.

Performance of Parallel Computers, Performance Metrics for Processors, Parallel Programming Models, Parallel Algorithms.

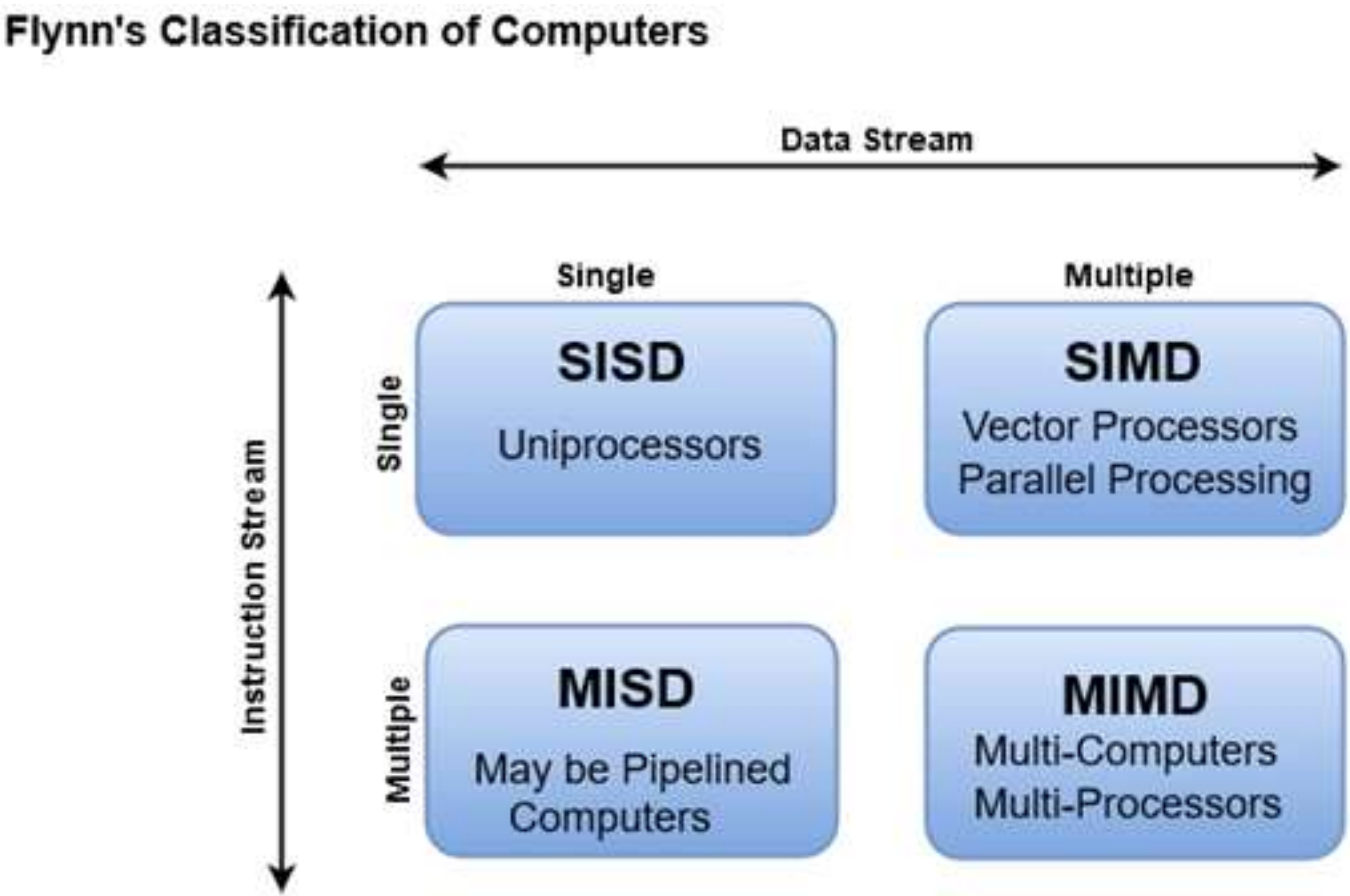
Flynn's Taxonomy

- **Parallel computing** is computing where the jobs are broken into discrete parts that can be executed **concurrently.**
- **Each part is further broken down into a series of instructions.**
- **Instructions from each piece execute simultaneously on different CPUs.**
- The breaking up of different parts of a task among multiple processors will help to reduce the amount of time to run a program.
- 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.

Flynn's Taxonomy

- 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. The difficult problem of parallel processing is portability.
- An Instruction Stream is a sequence of instructions that are read from memory.
- **Data Stream is the operations performed on the data in the processor.**
- **Flynn's taxonomy is a classification scheme for computer architectures proposed by Michael Flynn in 1966.**
- The taxonomy is based on the **number of instruction streams and data streams** that can be processed simultaneously by a computer architecture.

Flynn's Taxonomy



Flynn's Taxonomy

- **Single Instruction Single Data (SISD):** In a SISD architecture, there is a single processor that executes a single instruction stream and operates on a single data stream. This is the simplest type of computer architecture and is used in most traditional computers.
- **Single Instruction Multiple Data (SIMD):** In a SIMD architecture, there is a single processor that executes the same instruction on multiple data streams in parallel. This type of architecture is used in applications such as image and signal processing.
- **Multiple Instruction Single Data (MISD):** In a MISD architecture, multiple processors execute different instructions on the same data stream. This type of architecture is not commonly used in practice, as it is difficult to find applications that can be decomposed into independent instruction streams.

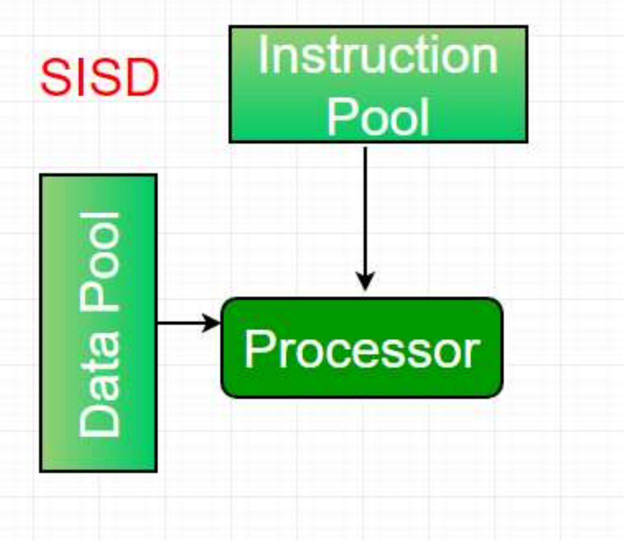
Flynn's Taxonomy

- **Multiple Instruction Multiple Data (MIMD):** In a MIMD architecture, multiple processors execute different instructions on different data streams.
- This type of architecture is used in distributed computing, parallel processing, and other high-performance computing applications.

Flynn's Taxonomy

- **Single-instruction, single-data (SISD) systems**
- An SISD computing system is a uniprocessor machine that is capable of executing a **single instruction, operating on a single data stream.** In SISD, machine instructions are processed in a sequential manner and computers adopting this model are popularly called sequential computers. Most conventional computers have SISD architecture.

All the instructions and data to be processed have to be stored in primary memory. The speed of the processing element in the SISD model is limited(dependent) by the rate at which the computer can transfer information internally. Dominant representative SISD systems are **IBM PC, and workstations.**



Flynn’s Taxonomy

- **Single-instruction, single-data (SISD) systems**

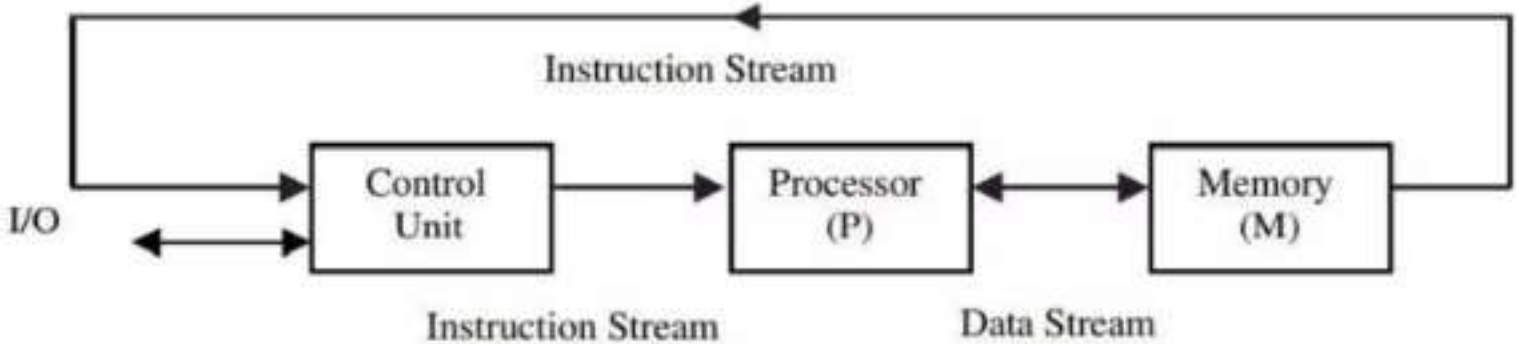
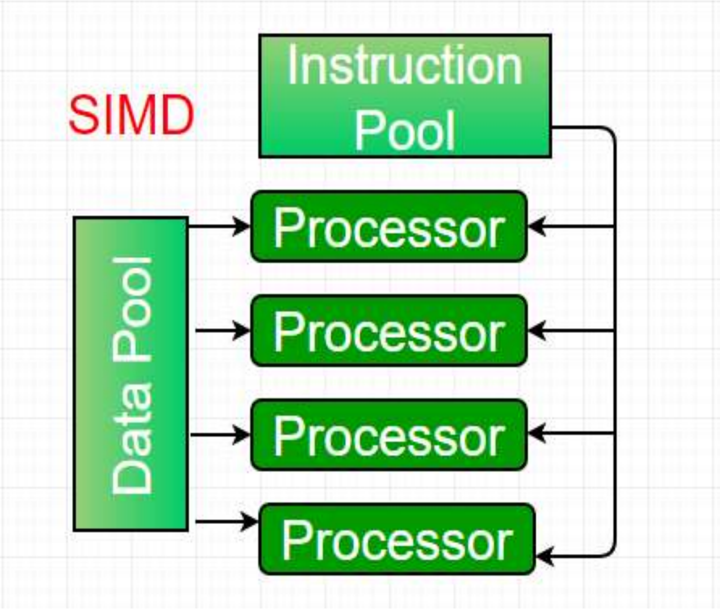


Figure 1.1 SISD architecture.

Flynn's Taxonomy

- **Single-instruction, multiple-data (SIMD) systems**
- An SIMD system is a multiprocessor machine capable of executing the **same instruction on all the CPUs but operating on different data streams.** Machines based on a SIMD model are well suited to scientific computing since they involve lots of vector and matrix operations.

So that the information can be passed to all the processing elements (PEs) organized data elements of vectors can be divided into multiple sets(N-sets for N PE systems) and each PE can process one data set.



Dominant representative SIMD systems are Cray's vector processing machines.

Flynn’s Taxonomy

- **Single-instruction, multiple-data (SIMD) systems**

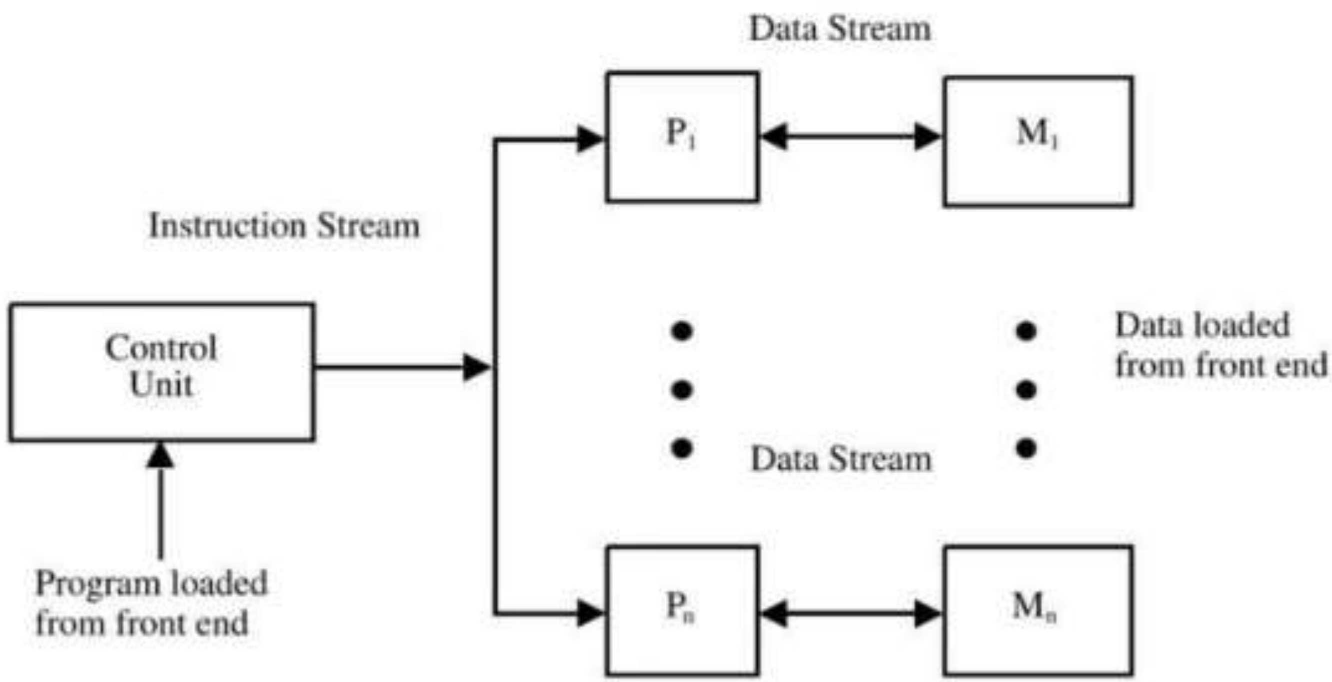


Figure 1.2 SIMD architecture.

Flynn’s Taxonomy

- **Single-instruction, multiple-data (SIMD) systems**

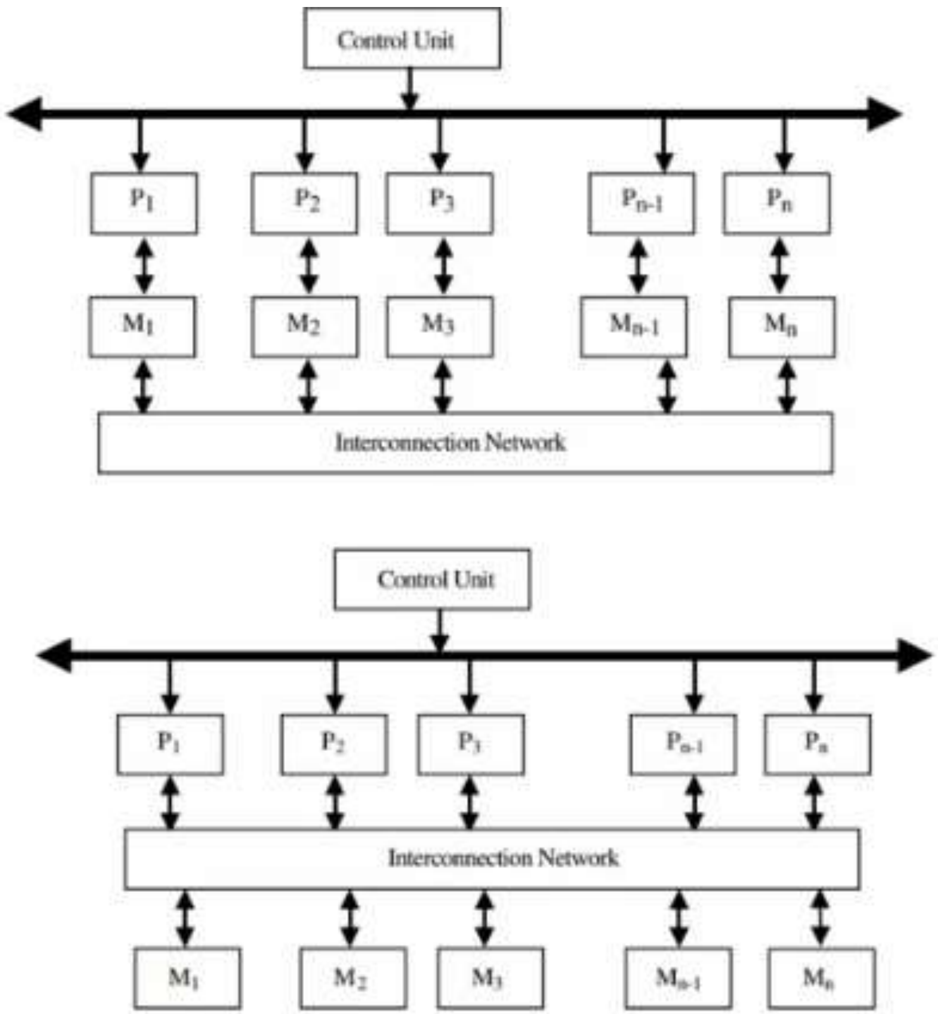
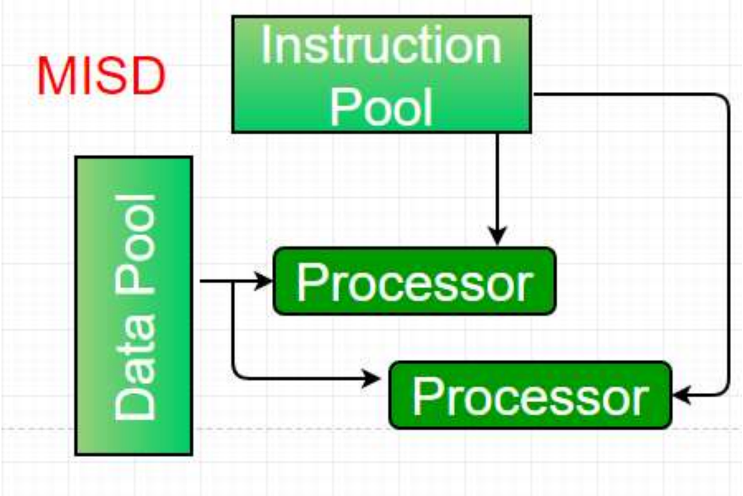


Figure 1.5 Two SIMD schemes.

Flynn's Taxonomy

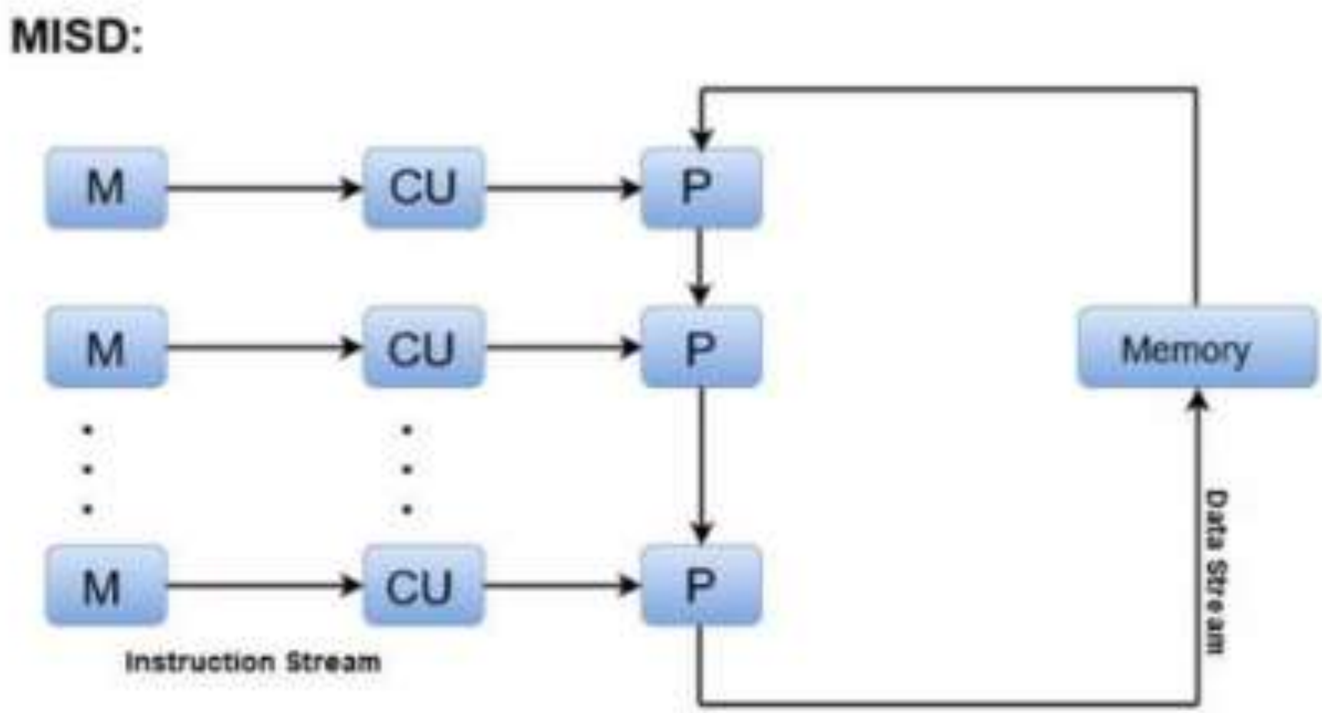
- **Multiple-instruction, single-data (MISD) systems**
- An MISD computing system is a multiprocessor machine **capable of executing different instructions on different PEs but all of them operate on the same dataset.**



Example Z = sin(x)+cos(x)+tan(x) The system performs different operations on the same data set.

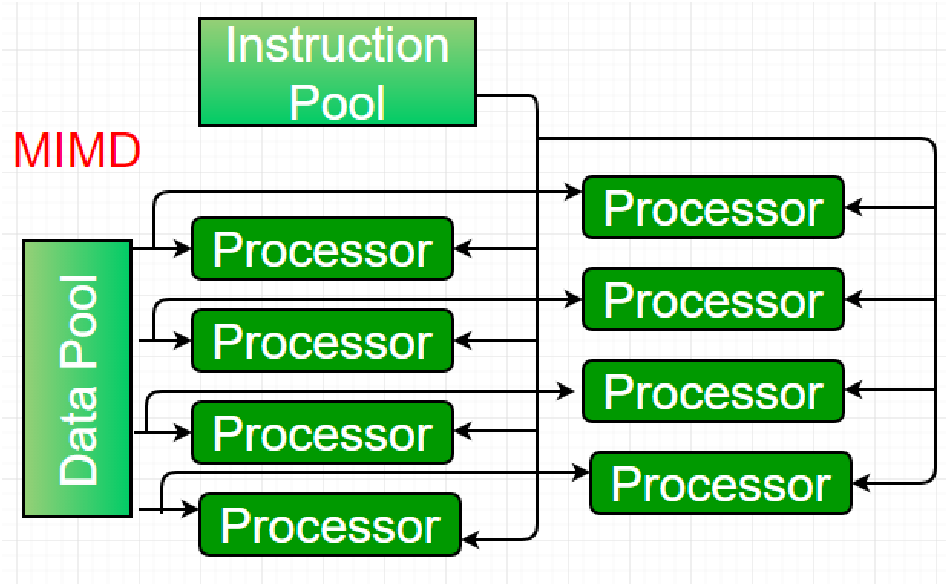
Machines built using the MISD model are not useful in most applications, a few machines are built, but none of them are available commercially.

- **Multiple-instruction, single-data (MISD) systems**



Flynn's Taxonomy

- **Multiple-instruction, multiple-data (MIMD) systems**
- An MIMD system is a multiprocessor machine that is **capable of executing multiple instructions on multiple data sets.** Each PE in the MIMD model has separate instruction and data streams; therefore machines built using this model are capable of any application. Unlike SIMD and MISD machines, PEs in MIMD machines work asynchronously.



- **Multiple-instruction, multiple-data (MIMD) systems**

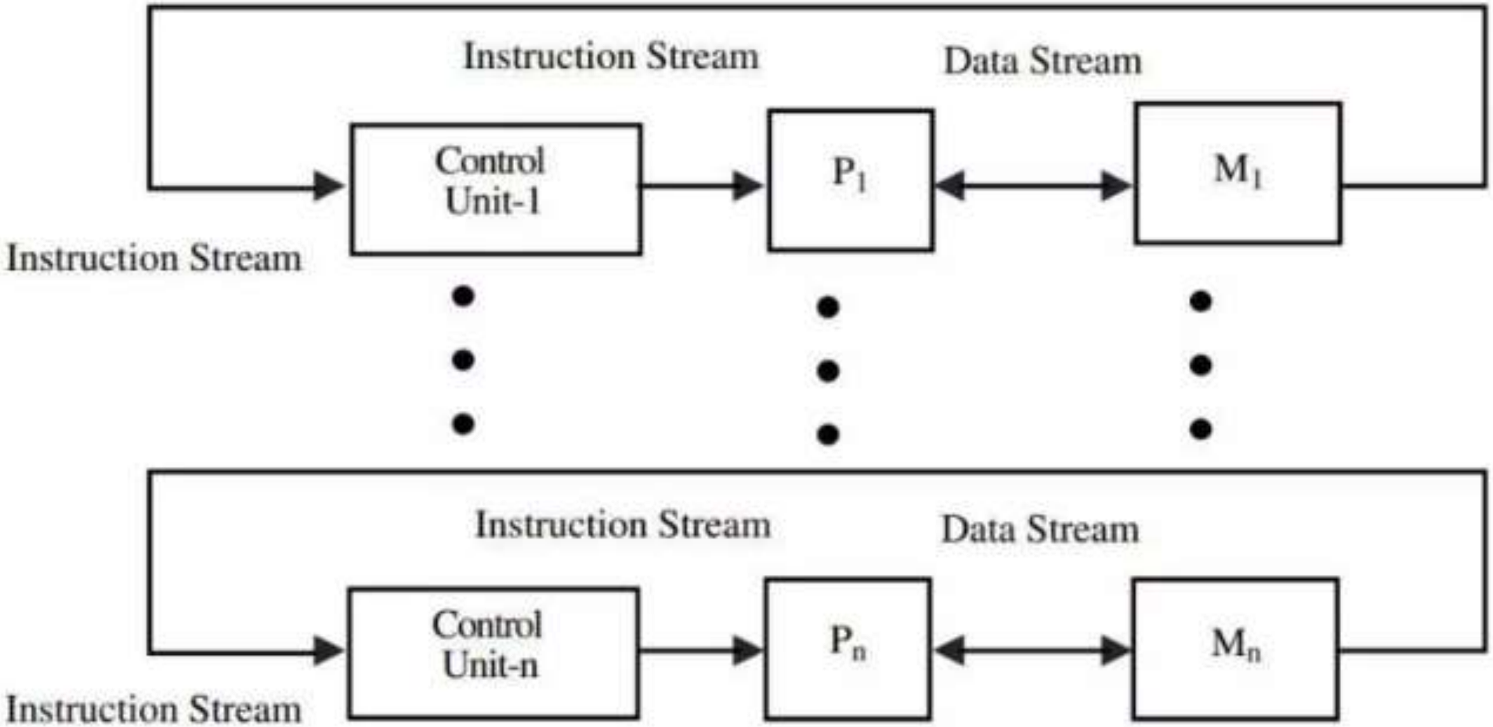


Figure 1.3 MIMD architecture.

Flynn's Taxonomy

- MIMD machines are broadly categorized into **shared-memory MIMD** and **distributed-memory MIMD** based on the way PEs are coupled to the main memory.
- In the **shared memory MIMD** model (tightly coupled multiprocessor systems), all the PEs are connected to a single global memory and they all have access to it.
- The communication between PEs in this model takes place through the shared memory, modification of the data stored in the global memory by one PE is visible to all other PEs.
- The dominant representative shared memory MIMD systems are Silicon Graphics machines and Sun/IBM's SMP (Symmetric Multi-Processing).

Flynn's Taxonomy

- Different types of computer architectures that fall under **Flynn's taxonomy have their own advantages and disadvantages.**
- **SISD architecture:** This is the simplest and most common type of computer architecture. It is easy to program and debug and can handle a wide range of applications. However, it does not offer significant performance gains over traditional computing systems.
- **SIMD architecture:** This type of architecture is highly parallel and can offer significant performance gains for applications that can be parallelized. However, it requires specialized hardware and software and is not well-suited for applications that cannot be parallelized.
- **MISD architecture:** This type of architecture is not commonly used in practice, as it is difficult to find applications that can be decomposed into independent instruction streams.

Flynn’s Taxonomy

- **MIMD architecture:** This type of architecture is highly parallel and can offer significant performance gains for applications that can be parallelized.
- It is well-suited for distributed computing, parallel processing, and other high-performance computing applications.
- However, it requires specialized hardware and software and can be challenging to program and debug.

Flynn's Taxonomy

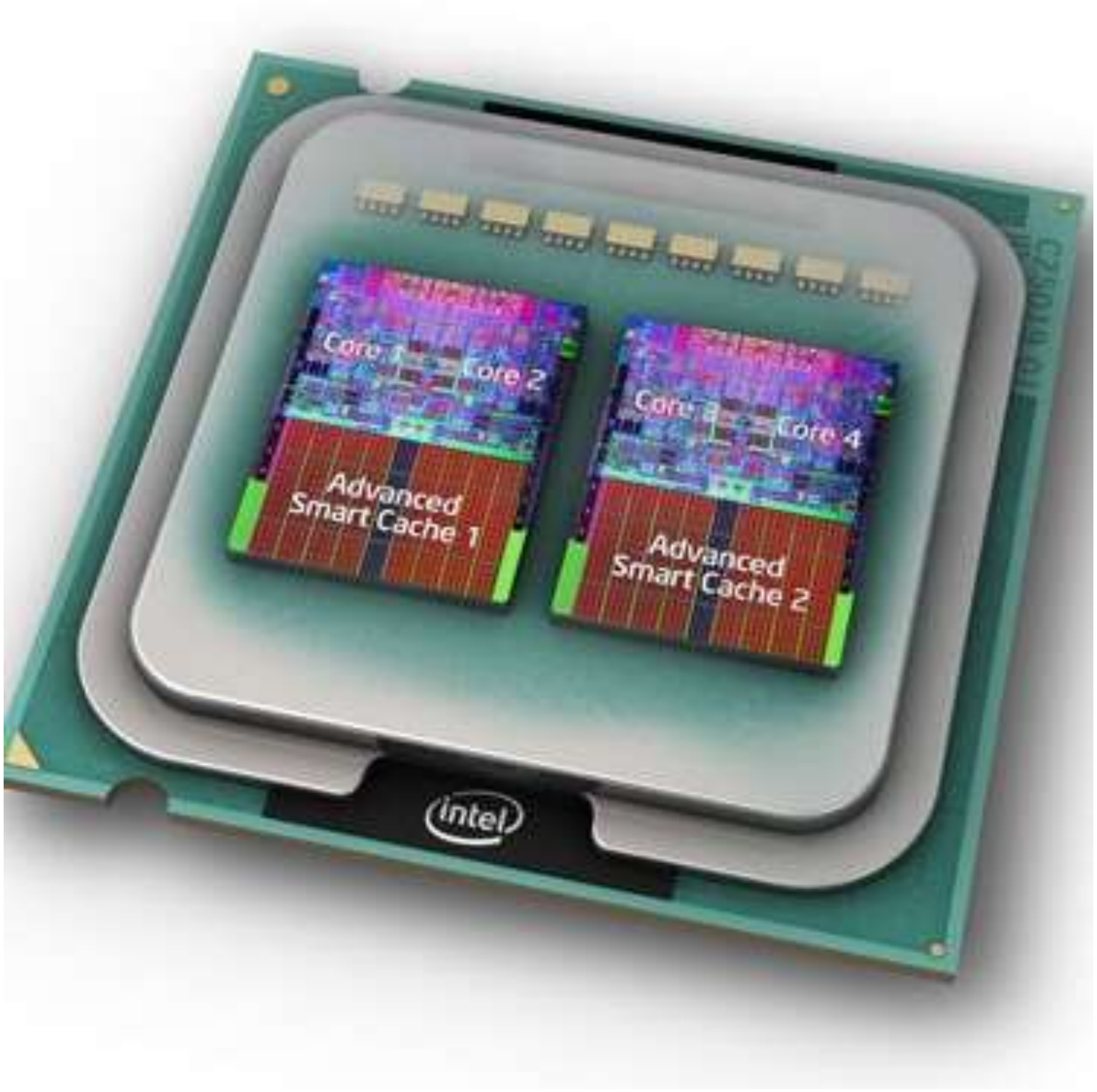
- **Features of Flynn's taxonomy include:**
- **Concurrency:** Flynn's taxonomy provides a way to classify computer architectures based on their concurrency, which refers to the number of tasks that can be executed simultaneously.
- **Performance:** Different types of architectures have different performance characteristics, and Flynn's taxonomy provides a way to compare their performance based on the number of concurrent instructions and data streams.
- **Parallelism:** Flynn's taxonomy highlights the importance of parallelism in computer architecture and provides a framework for designing and analyzing parallel processing systems.

Multi Core Processors

- A multi-core processor is an **integrated circuit with two or more processors connected to it for faster simultaneous processing** of several tasks, reduced power consumption, and for greater performance. Generally, it is made up of two or more processors that read and execute program instructions.
- In other words, on a **single chip, a multi-core processor comprises numerous processing units, or "Cores," each of which has the potential to do distinct tasks**.
- For instance, if you are performing many tasks at once, such as watching a movie and using WhatsApp, one core will handle activities like watching a movie while the other handles other responsibilities like WhatsApp.

- https://www.youtube.com/watch?v=Pr5yosuGZDc

Multi Core Processors

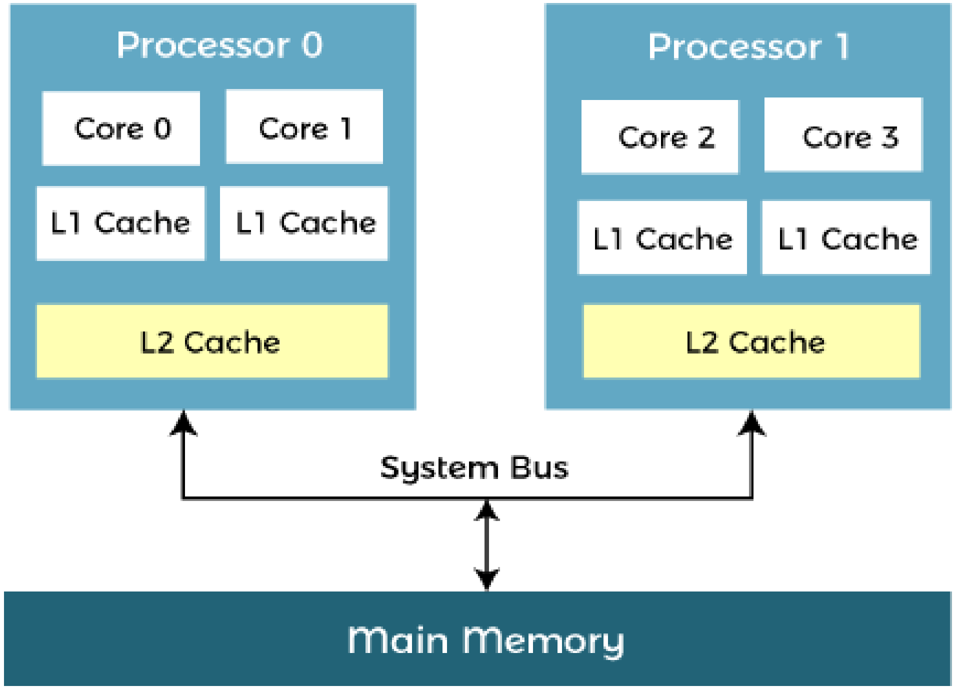


Multi Core Processors

- A dual-core configuration is comparable to having **several different processors installed** on the same computer, but the connection between them is faster because the two CPUs are plugged into the same socket.
- **Several instructions in parallel may be executed by individual cores**, boosting the speed of software built to make use of the architecture's unique features.
- As compared to a single-core processor, a dual-core processor usually is twice as powerful in ideal circumstances. In actuality, performance gains of around 50% are expected: a dual-core CPU is roughly 1.5 times as powerful as a single-core processor.

Architecture of Multicore Processor

- A multi-core processor's design enables the communication between all available cores, and they **divide and assign all processing duties appropriately**. The processed data from each core is transmitted back to the computer's main board (Motherboard) via a single common gateway once all of the processing operations have been finished. This method beats a single-core CPU in terms of total performance.



Architecture of Multicore Processor

- **Advantages of Multi-Core Processor**
- **Performance**
- A multi-core CPU, by nature, can do more work as compared to a single-core processor. The spacing between the cores of an integrated circuit allows for faster clock rates. As a result, the signals do not need to travel a large distance to reach their target and are also persistent. When compared to using a separate processor, the speeds are far quicker.
- **Reliability**
- In multi-core CPUs, the software is always assigned to different cores. When one piece of software fails, the others remain unaffected. Whenever a defect arises, it affects only one core. As a result, multi-core CPUs are better able to resist faults.
- **Software Interactions**
- Even if the software is running on multiple cores, it will communicate with one another. Spatial and temporal isolation is a process that a multi-core processor goes through. Core threads are never delayed as a result of these processes.

Architecture of Multicore Processor

- **Multitasking**
- An operating system can use a multi-core CPU to run two or more processes at the same time, even if many programmes may be executed at the same time. A photoshop application, for example, can be used to perform two jobs at once.
- **Power Consumption**
- Multitasking with a multi-core CPU, on the other hand, requires less power. Only the part of the CPU that generates heat will be used. The power consumption is eventually minimized, resulting in less battery utilization. Some operating systems, on the other hand, need more resources as compared to others.
- **Obsolescence Avoidance**
- Architects can avoid technology obsolescence and increase maintainability by using multicore CPUs. Chipmakers are using the most recent technological advancements in their multicore CPUs. Single-core chips are becoming increasingly difficult to come by as the number of cores increases.

Architecture of Multicore Processor

- **Multicore processors may increase (but do not guarantee) geographical and temporal isolation when compared to single-core systems.**
- Software on one core is less likely to impact software on the other if both cores are executing on the same single-core.
- This decoupling happens due to geographical and temporal isolation (threads on one core are not delayed by threads on another core).
- With the help of limiting the impact of errors to a single core, multicore processing can increase robustness.
- When executing mixed-criticality programmes separately, this enhanced isolation is very important (safety-critical, mission-critical, and security-critical).

More details

- https://www.javatpoint.com/what-is-a-multicore-processor
- https://www.techtarget.com/searchdatacenter/definition/multi-core-processor