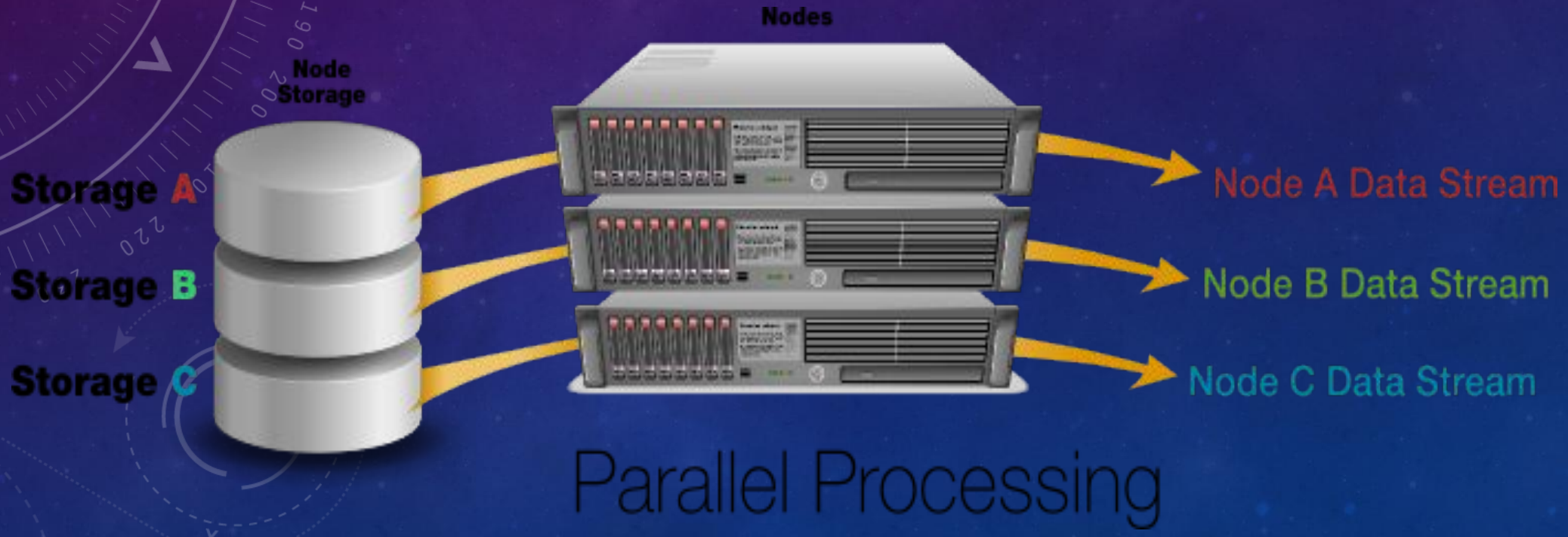
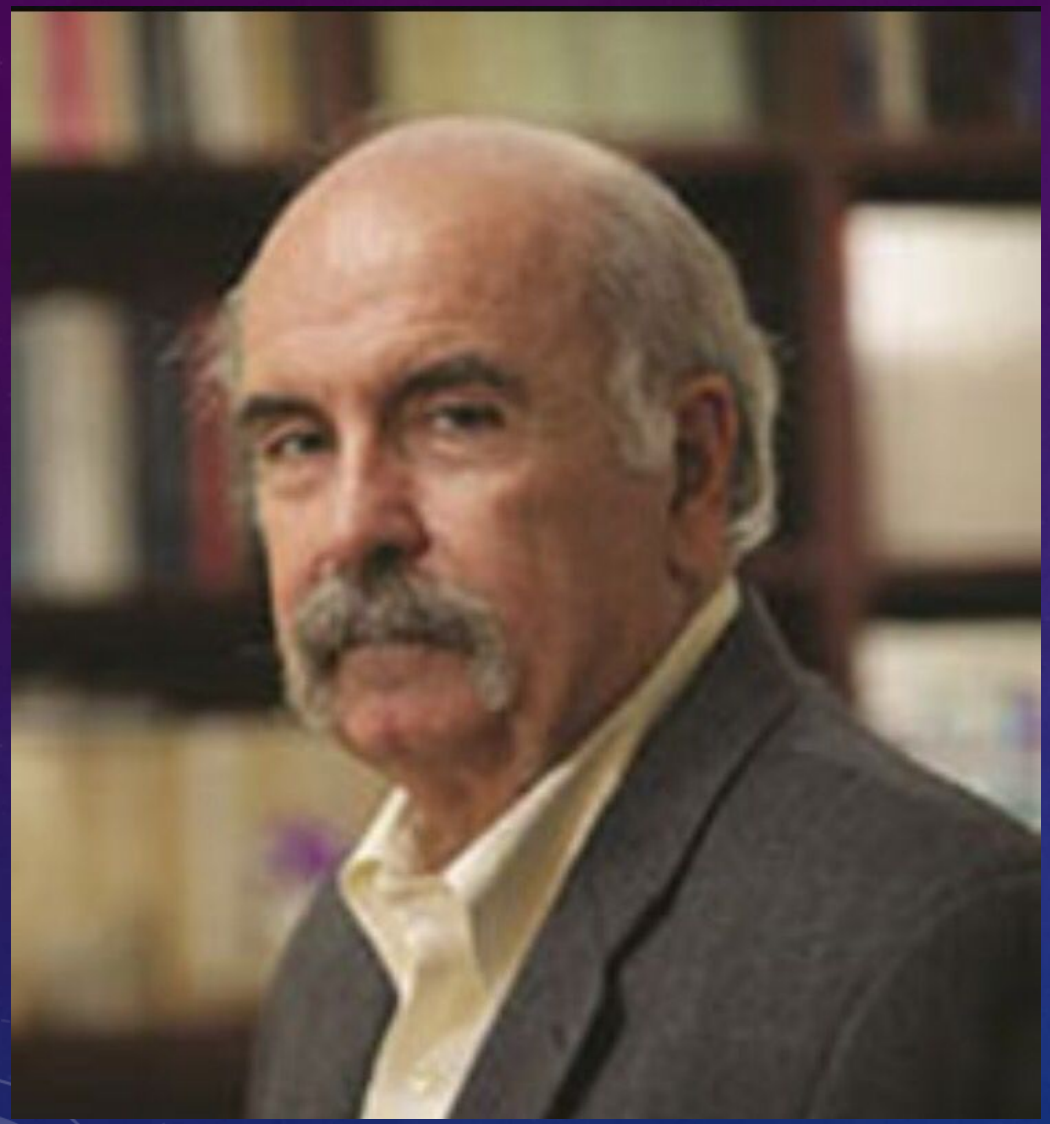
PARALLEL PROCESSING OF DATA AND THEIR RESTORATION



IN COMPUTERS, PARALLEL PROCESSING IS THE PROCESSING OF PROGRAM INSTRUCTIONS BY DIVIDING THEM AMONG MULTIPLE PROCESSORS WITH THE OBJECTIVE OF RUNNING A PROGRAM IN LESS TIME. IN THE EARLIEST COMPUTERS, ONLY ONE PROGRAM RAN AT A TIME. A COMPUTATION-INTENSIVE PROGRAM THAT TOOK ONE HOUR TO RUN AND A TAPE COPYING PROGRAM THAT TOOK ONE HOUR TO RUN WOULD TAKE A TOTAL OF TWO HOURS TO RUN.

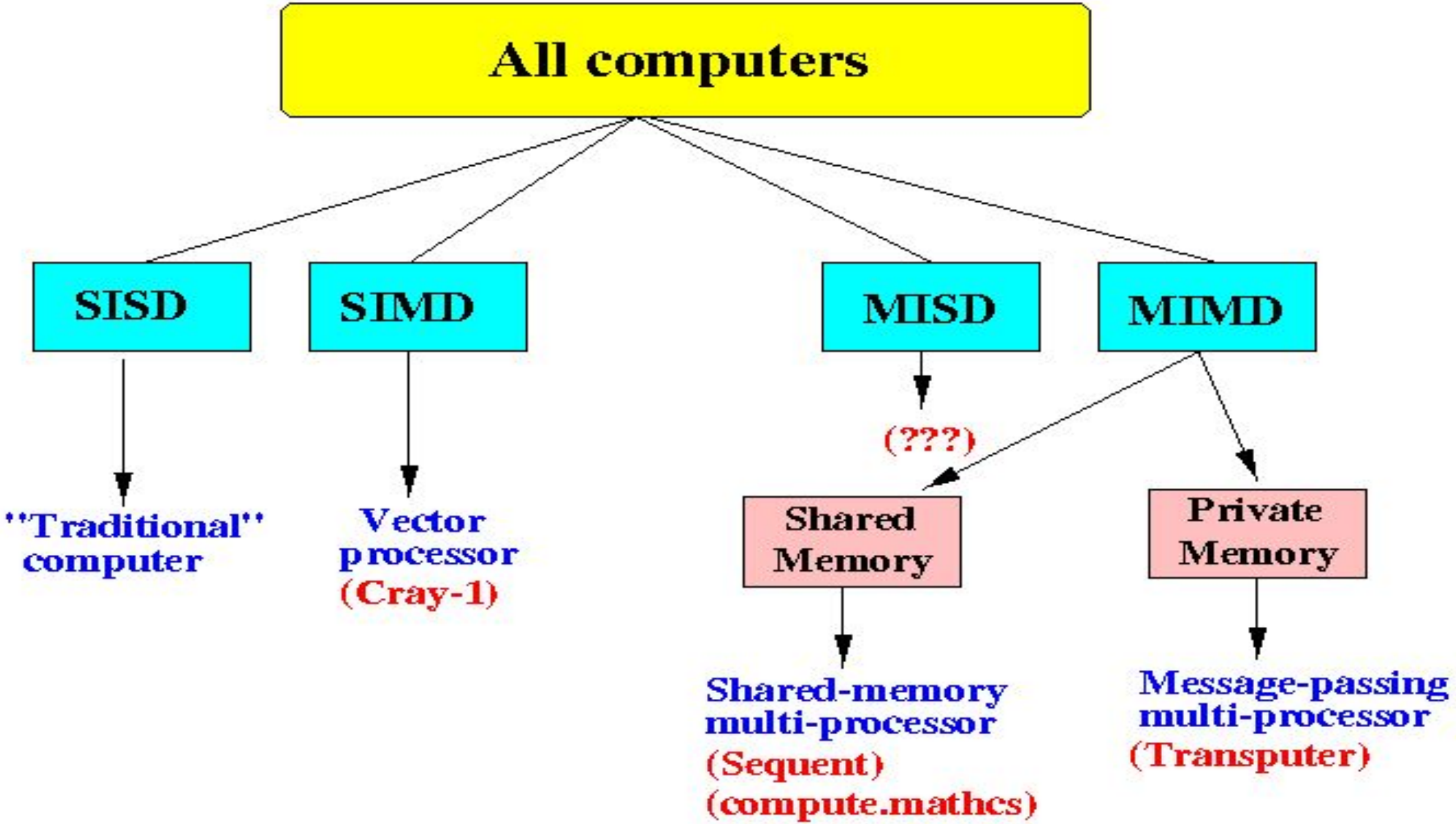
ARCHITECTURE OF HIGH PERFORMANCE SYSTEMS IS QUITE WIDE, BECAUSE UNDER THE ARCHITECTURE WE CAN UNDERSTAND THE WAY OF PARALLEL DATA PROCESSING USED IN THE SYSTEM AND ORGANIZATION OF MEMORY, AND THE TOPOLOGY OF CONNECTIONS BETWEEN PROCESSORS, AND EXECUTION SYSTEM ARITHMETIC OPERATIONS.



ATTEMPTS TO SYSTEMATIZE ALL THE VARIETY OF ARCHITECTURES WAS FIRST MADE IN THE LATE 60'S AND CONTINUES TO THIS DAY. IN 1966 M. FLYNN (FLYNN) WAS OFFERED AN EXTREMELY CONVENIENT APPROACH TO THE CLASSIFICATION OF ARCHITECTURES OF COMPUTING SYSTEMS. IT WAS BASED ON THE CONCEPT OF FLOW, WHICH REFERS TO A SEQUENCE OF ELEMENTS, COMMANDS, OR DATA PROCESSED BY THE PROCESSOR.

Parallel Computing

- SISD : single instruction stream, single data stream.
- SIMD : single instruction stream, multiple data stream.
- MISD : multiple instruction stream, single data stream.
- MIMD : multiple instruction stream, multiple data stream.


All computers
SISD
SIMD
MISD
MIMD
"Traditional" computer
Vector processor (Cray-1)
(???)
Shared Memory
Private Memory
Shared-memory multi-processor (Sequent) (compute.mathcs)
Message-passing multi-processor (Transputer)

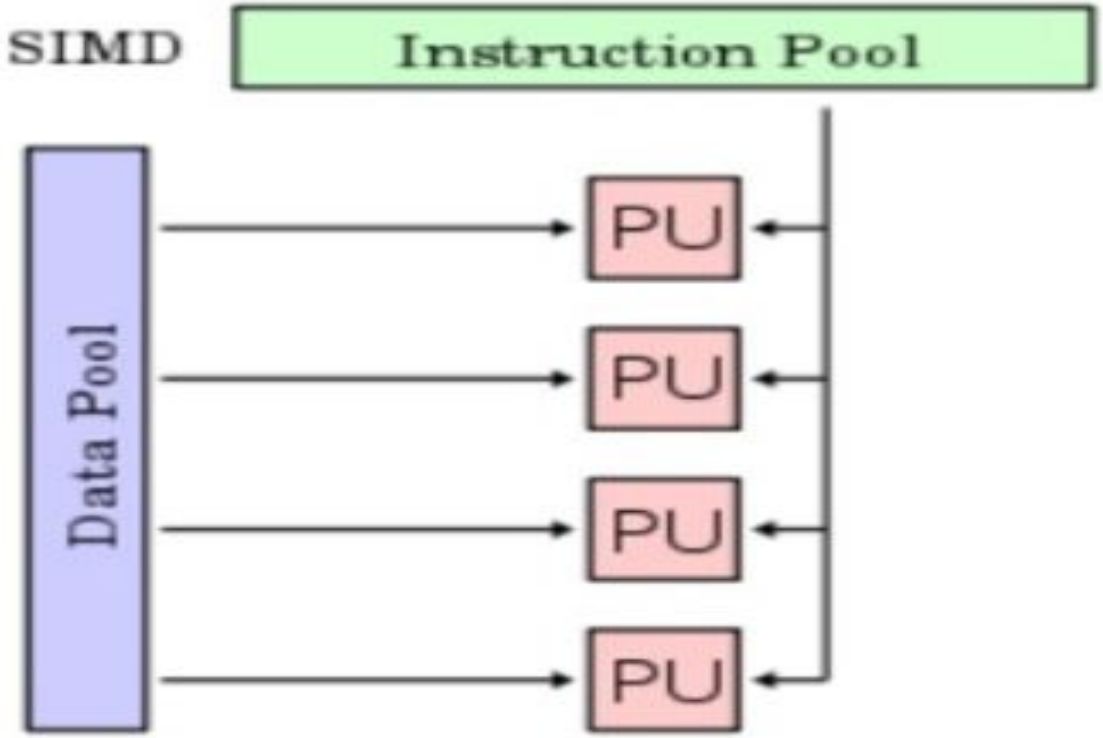
SIMD

Single Instruction Multiple Data

SIMD

- **Single Instruction Multiple Data (SIMD)**, is a class of parallel computers in Flynn's taxonomy.

- In computing, **SIMD** is a technique employed to Achieve data level parallelism.



SISD

SINGLE INSTRUCTION SINGLE DATA

SISD

- This is the oldest style of computer architecture, and still one of the most important: all personal computers fit within this category
- *Single instruction* refers to the fact that there is only one instruction stream being acted on by the CPU during any one clock tick;
- *single data* means, analogously, that one and only one data stream is being employed as input during any one clock tick.

CHARACTERISTICS OF SISD

- **Serial Instructions** are executed one after the other, in lock-step; this type of sequential execution is commonly called *serial*, as opposed to *parallel*, in which multiple instructions may be processed simultaneously.

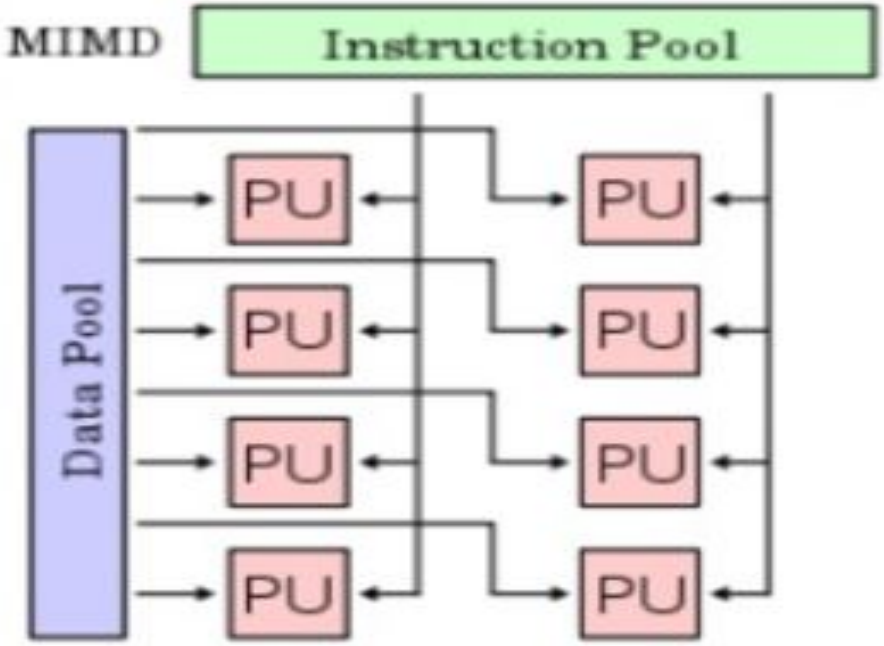
- **Deterministic** Because each instruction has a unique place in the execution stream, and thus a unique time during which it and it alone is being processed, the entire execution is said to be *deterministic*, meaning that you (can potentially) know exactly what is happening at all times, and, ideally, you can exactly recreate the process, step by step, at any later time.

- **Examples:**
  - All personal computers,
  - All single-instruction-unit-CPU workstations,
  - Mini-computers, and
  - Mainframes.

MIMD

Multiple Instruction Multiple Data

MIMD

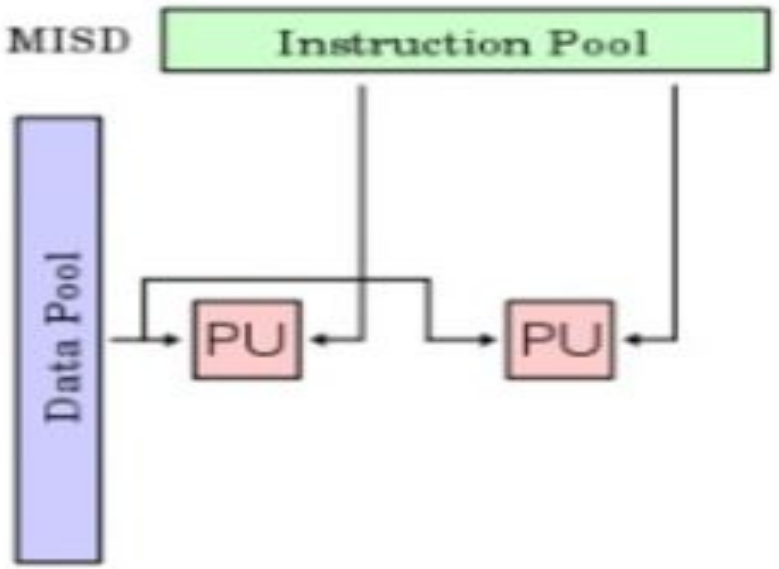


- In computing, **MIMD** is a technique employed to achieve parallelism.
- Machines using MIMD have a number of processors that function asynchronously and independently.
- At any time, different processors may be executing different instructions on different pieces of data.
- MIMD architectures may be used in a number of application areas such as computer-aided design/computer-aided manufacturing, simulation, modeling, and as communication switches.

MISD

MULTIPLE INSTRUCTION SINGLE DATA

MISD



- In computing, **MISD** is a type of parallel computing architecture where many functional units perform different operations on the same data.
- Pipeline architectures belong to this type.
- Fault-tolerant computers executing the same instructions redundantly in order to detect and mask errors, in a manner known as task replication, may be considered to belong to this type.
- Not many instances of this architecture exist, as MIMD and SIMD are often more appropriate for common data parallel techniques.

PARALLEL COMPUTING

PARALLEL COMPUTING

- **Parallel computing** is a form of computation in which many calculations are carried out simultaneously, operating on the principle that large problems can often be divided into smaller ones, which are then solved concurrently ("in parallel").

- There are several different forms of parallel computing:
  - bit-level,
  - instruction level,
  - data, and
  - task parallelism.

Parallelism has been employed for many years, mainly in high-performance computing.

As power consumption by computers has become a concern in recent years, parallel computing has become the dominant issue in computer architecture, mainly in the form of multicore processors.

PARALLEL COMPUTING

- Computer Software is written for serial computation. To solve a problem, an algorithm is constructed and implemented as a serial stream of instructions. Only one instruction may execute at a time—after that instruction is finished, the next is executed.

- Parallel computing, on the other hand, uses multiple processing elements simultaneously to solve a problem.

- This is accomplished by breaking the problem into independent parts so that each processing element can execute its part of the algorithm simultaneously with the others.

- The processing elements can be diverse and include resources such as a single computer with multiple processors, several networked computers, specialized hardware or any combination of the above.

TYPES OF PARALLELISM

- **Bit-level parallelism:**
  - From the advent of VLSI in the 1970s until about 1986, speed-up in computer architecture was driven by doubling computer word size— the amount of information the processor can manipulate per cycle. Increasing the word size reduces the number of instructions the processor must execute to perform an operation on variables whose sizes are greater than the length of the word.
- **Instruction-level parallelism:**
  - A computer program is, a stream of instructions executed by a processor. These instructions can be re-ordered and combined into groups which are then executed in parallel without changing the result of the program. This is known as instruction-level parallelism.
- **Data parallelism:**
  - Data parallelism is parallelism inherent in program loops, which focuses on distributing the data across different computing nodes to be processed in parallel.
- **Task parallelism**:
  - Task parallelism is the characteristic of a parallel program that "entirely different calculations can be performed on either the same or different sets of datal This contrasts with data parallelism, where the same calculation is performed on the same or different sets of data.

THANK YOU