



UNIT 6

COMPUTER ORGANIZATION

Dr. Ankita Markare

UNIT 6

- An Overview of Parallel Processing
- Organization of Multiprocessor
 - Flynn's Classification
 - System Topologies
- Multiprocessors
 - Symmetric Multiprocessors
 - Asymmetric Multiprocessors
- Pipelining
 - Construction
 - Systolic Arrays
- Vector Processors



Parallel Processing

Dr. Ankita Markare

An Overview of Parallel Processing

What is parallel processing?

- Parallel processing is a method to improve computer system performance by executing two or more instructions simultaneously.

The goals of parallel processing.

- One goal is to reduce the “wall-clock” time or the amount of real time that you need to wait for a problem to be solved.
- Another goal is to solve bigger problems that might not fit in the limited memory of a single CPU.

An Analogy of Parallelism



The task of ordering a shuffled deck of cards by suit and then by rank can be done faster if the task is carried out by two or more people. By splitting up the decks and performing the instructions simultaneously, then at the end combining the partial solutions you have performed parallel processing.

Parallelism in Uniprocessor Systems

- It is possible to achieve parallelism with a uniprocessor system.
 - Some examples are the instruction pipeline, arithmetic pipeline, I/O processor.
- Note that a system that performs different operations on the same instruction is not considered parallel.
- Only if the system processes two different instructions simultaneously can it be considered parallel.

Parallelism in a Uniprocessor System

A reconfigurable arithmetic pipeline is an example of parallelism in a uniprocessor system.

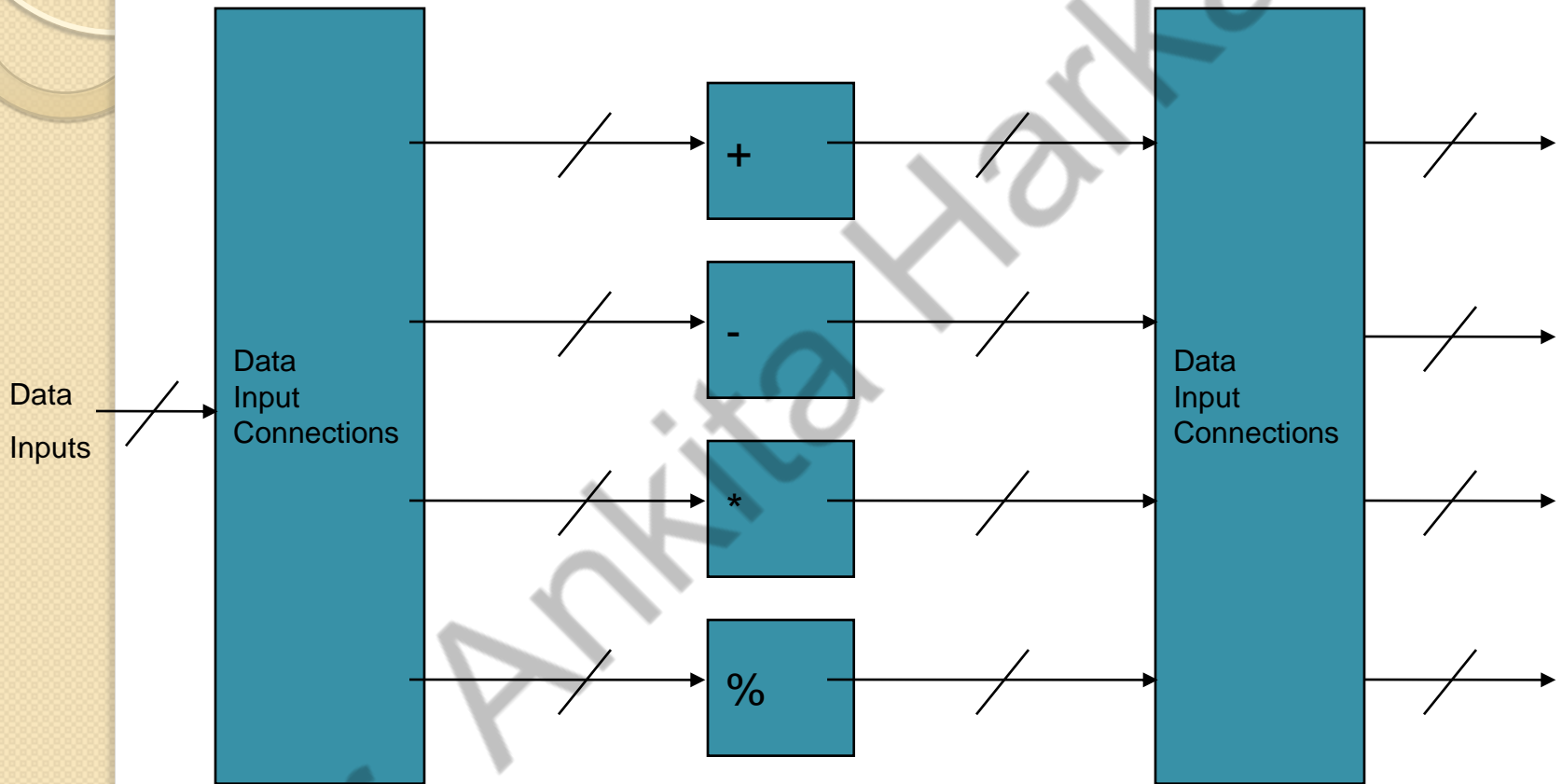
Each stage of a reconfigurable arithmetic pipeline has a multiplexer at its input. The multiplexer may pass input data, or the data output from other stages, to the stage inputs. The control unit of the CPU sets the select signals of the multiplexer to control the flow of data, thus configuring the pipeline.

Vector Arithmetic Unit

A vector arithmetic unit contains multiple functional units that perform addition, subtraction, and other functions. The control unit routes input values to the different functional units to allow the CPU to execute multiple instructions simultaneously.

For the operations $A \leftarrow B + C$ and $D \leftarrow E - F$, the CPU would route B and C to an adder and then route E and F to a subtractor for simultaneous execution.

A Vectored Arithmetic Unit



$A \leftarrow B + C$

$D \leftarrow E - F$



Organization of Multiprocessors

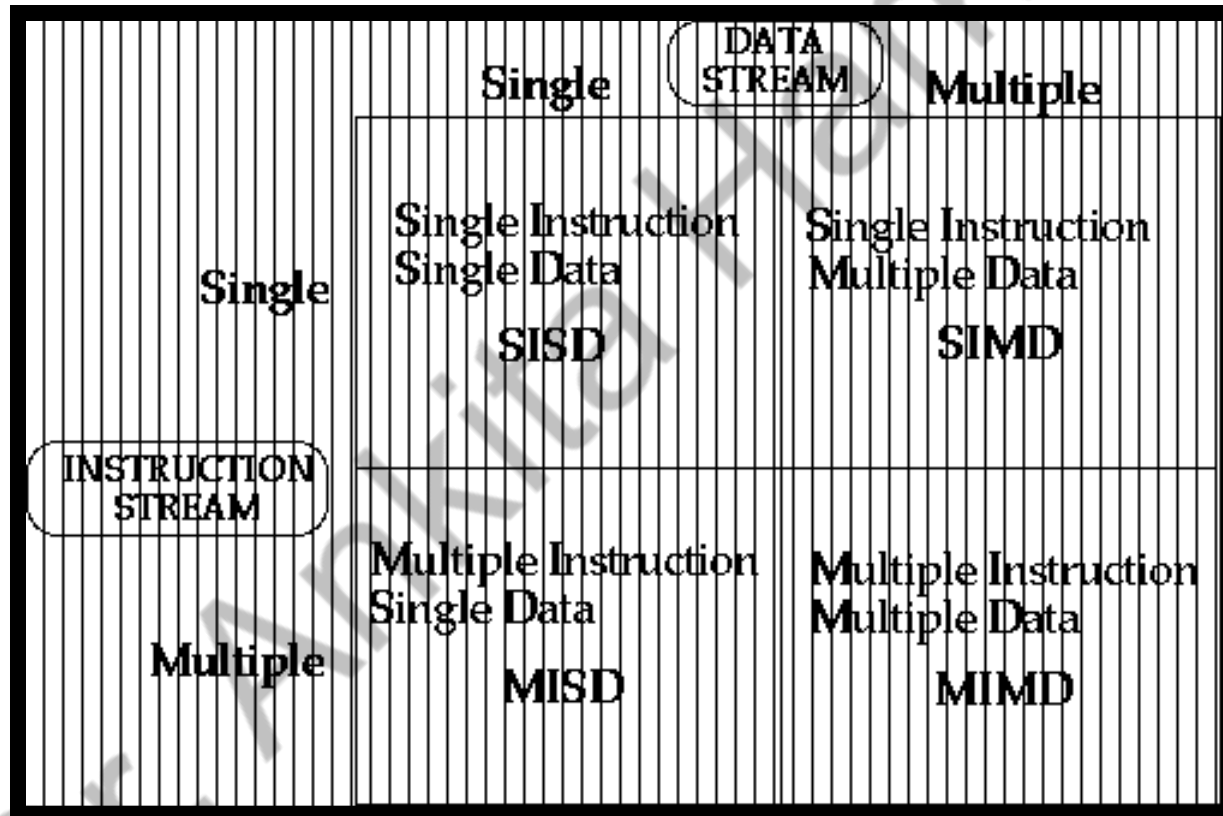
Dr. Ankita Markare

Organization of Multiprocessor Systems

- Flynn's Classification
 - Was proposed by researcher Michael J. Flynn in 1966.
 - It is the most commonly accepted taxonomy of computer organization.
 - In this classification, computers are classified by whether it processes a single instruction at a time or multiple instructions simultaneously, and whether it operates on one or multiple data sets.

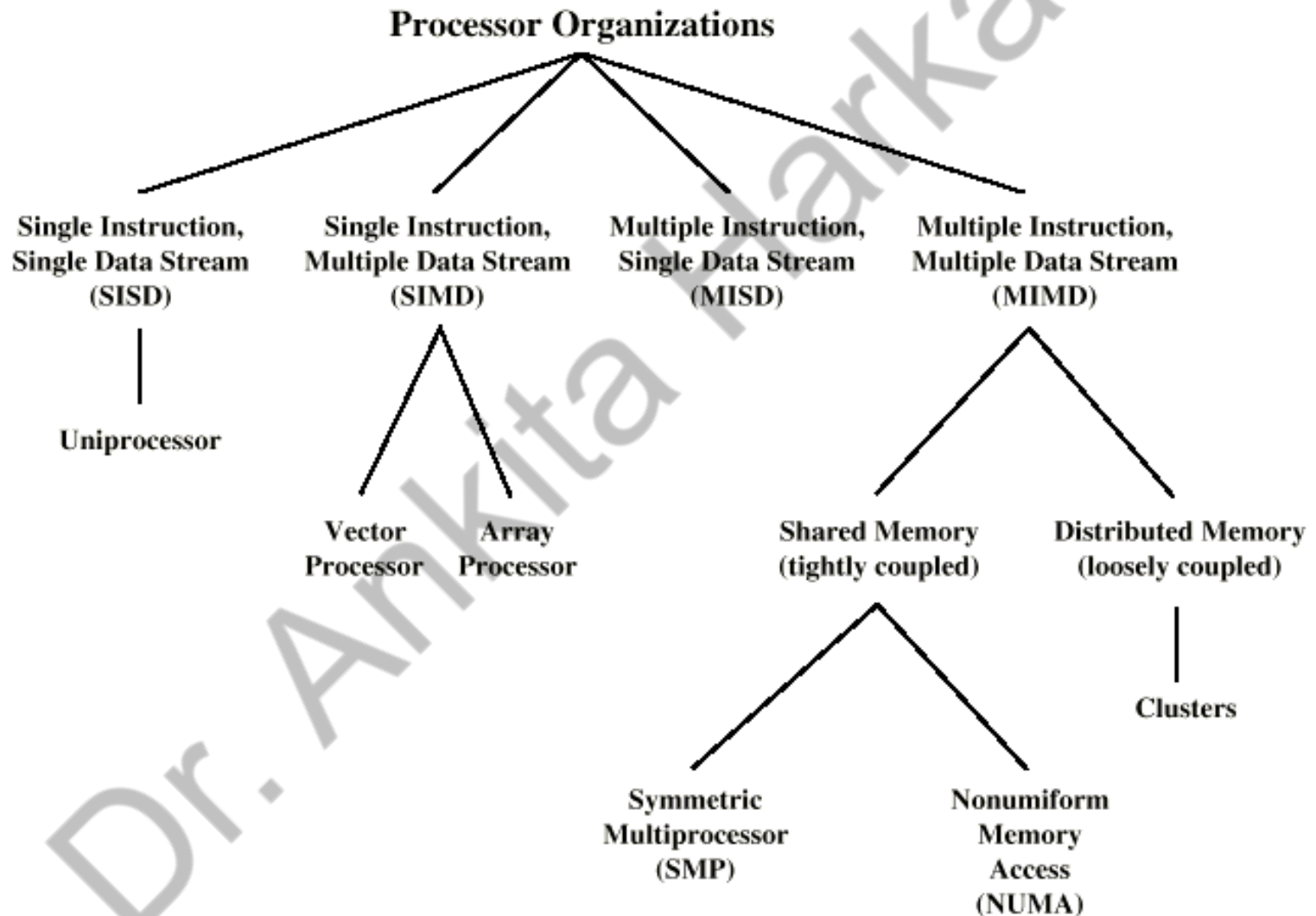
Taxonomy of Computer Architectures

Simple Diagrammatic Representation



4 categories of Flynn's classification of multiprocessor systems by their instruction and data streams

Taxonomy of Parallel Processor Architectures

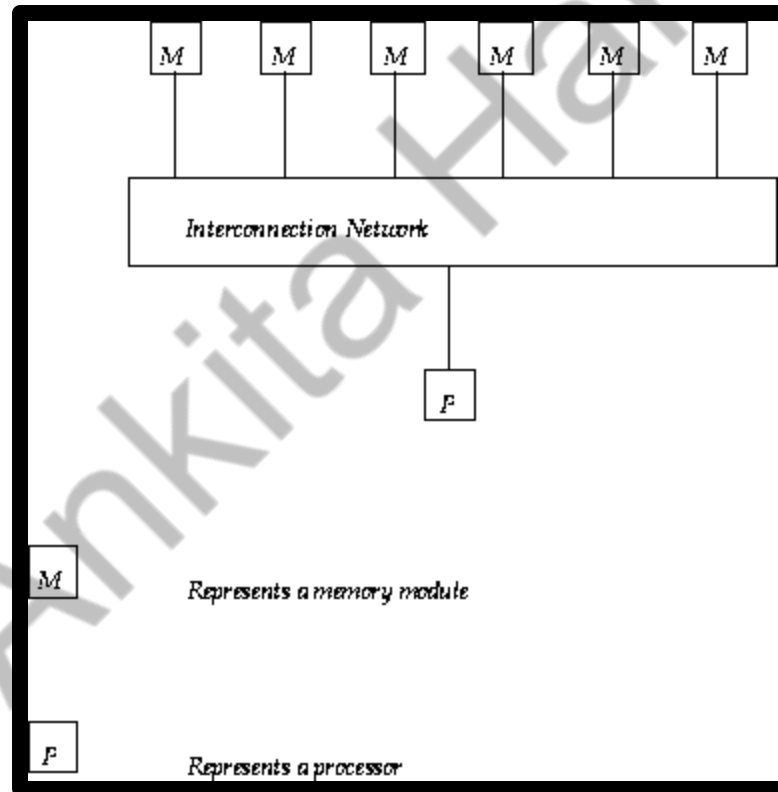


Single Instruction, Single Data (SISD)

- SISD machines executes a single instruction on individual data values using a single processor.
- Based on traditional Von Neumann uniprocessor architecture, instructions are executed sequentially or serially, one step after the next.
- Until most recently, most computers are of SISD type.

SISD

Simple Diagrammatic Representation

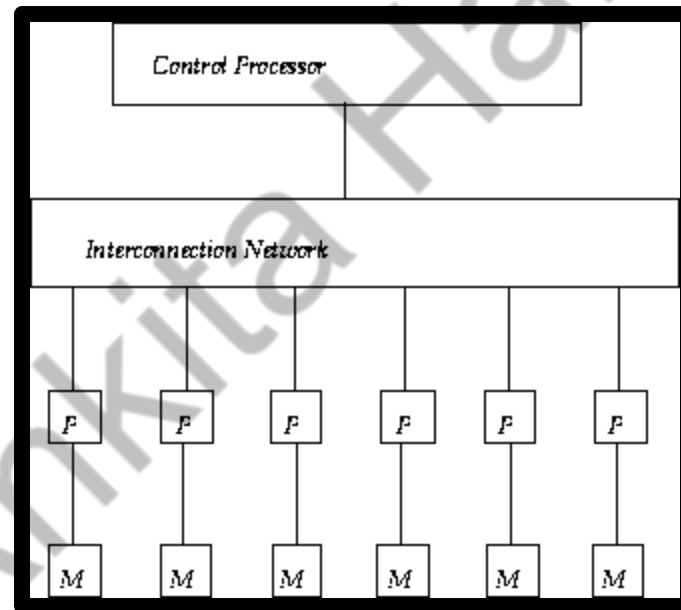


Single Instruction, Multiple Data (SIMD)

- An SIMD machine executes a single instruction on multiple data values simultaneously using many processors.
- Since there is only one instruction, each processor does not have to fetch and decode each instruction. Instead, a single control unit does the fetch and decoding for all processors.
- SIMD architectures include array processors.

SIMD

Simple Diagrammatic Representation

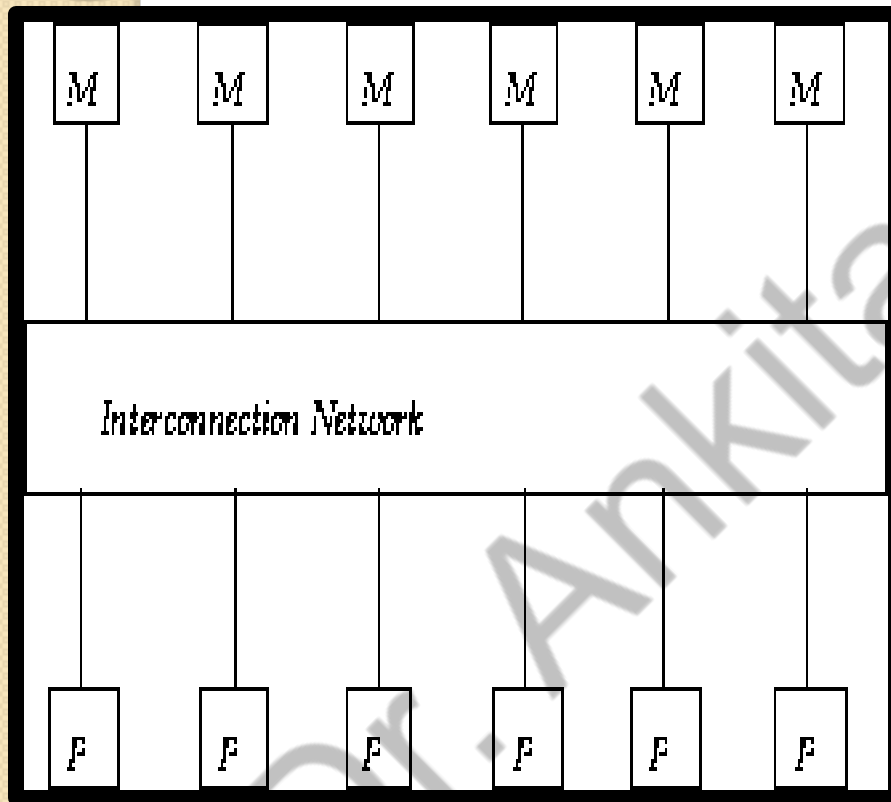


Multiple Instruction, Multiple Data (MIMD)

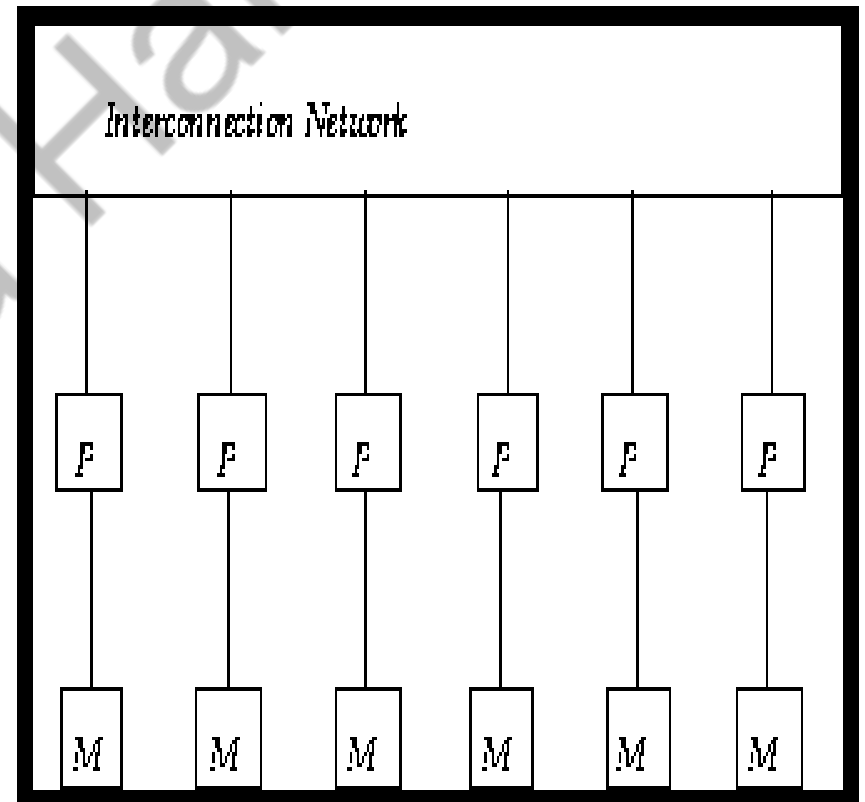
- MIMD machines are usually referred to as multiprocessors or multicomputers.
- It may execute multiple instructions simultaneously, contrary to SIMD machines.
- Each processor must include its own control unit that will assign to the processors parts of a task or a separate task.
- It has two subclasses: Shared memory and distributed memory

MIMD

Simple Diagrammatic Representation
(Shared Memory)



Simple Diagrammatic Representation
(Distributed Memory)



Multiple Instruction, Single Data (MISD)

- This category does not actually exist. This category was included in the taxonomy for the sake of completeness.



Multiprocessors

Dr. Ankita Harkare

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

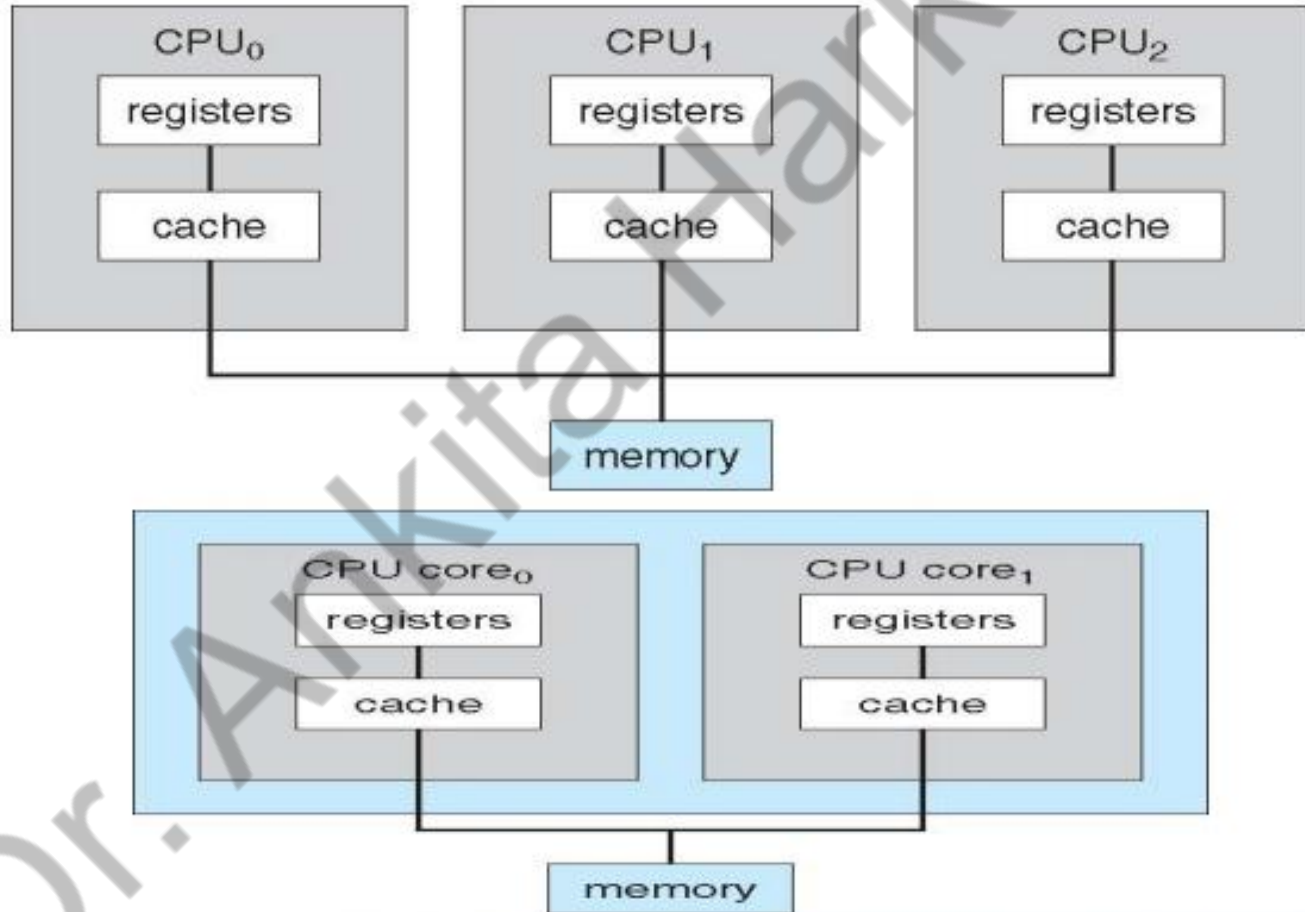
Multiprocessor Symmetry

- 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.
- Systems that treat all CPUs equally are called symmetric multiprocessing (**SMP**) systems.
- If all CPUs are not equal, system resources may be divided in a number of ways, including asymmetric multiprocessing (**ASMP**), non-uniform memory access (**NUMA**) multiprocessing, and clustered multiprocessing.

Multiprocessor Systems

- Parallel Systems / Tightly coupled systems
- More than one processor in close communication, sharing the computer bus, the clock sometimes memory and IO devices.
- Advantages
 - Increased throughput
 - Economy of scale
 - Increased reliability

Multiprocessor Architecture



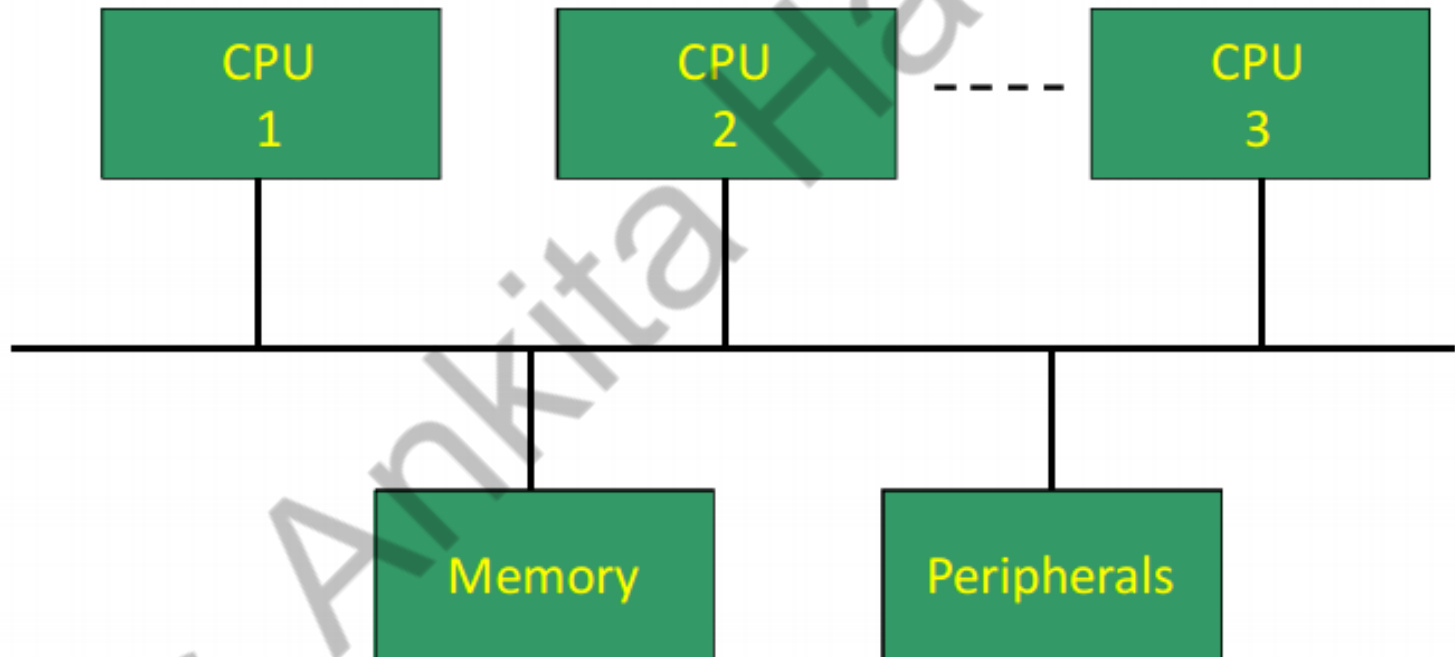
Types of Multiprocessor Systems

- **Asymmetric Multiprocessing**
 - master processor schedules and allocates work to slave processors.
- **Symmetric Multiprocessing (SMP)**
 - Each processor runs an identical copy of the operating system.
 - Typically each processor does self-scheduling from the pool of available process.
 - Most modern operating systems support SMP.

Symmetric Multiprocessing (SMP)

- Each processor can perform the same functions and share same main memory and I/O facilities (symmetric).
- The OS schedule processes/threads across all the processors (real parallelism).
- Existence of multiple processors is transparent to the user.
- Incremental growth: just add another CPU!
- Robustness: a single CPU failure does not halt the system, only the performance is reduced.

Symmetric Multiprocessing



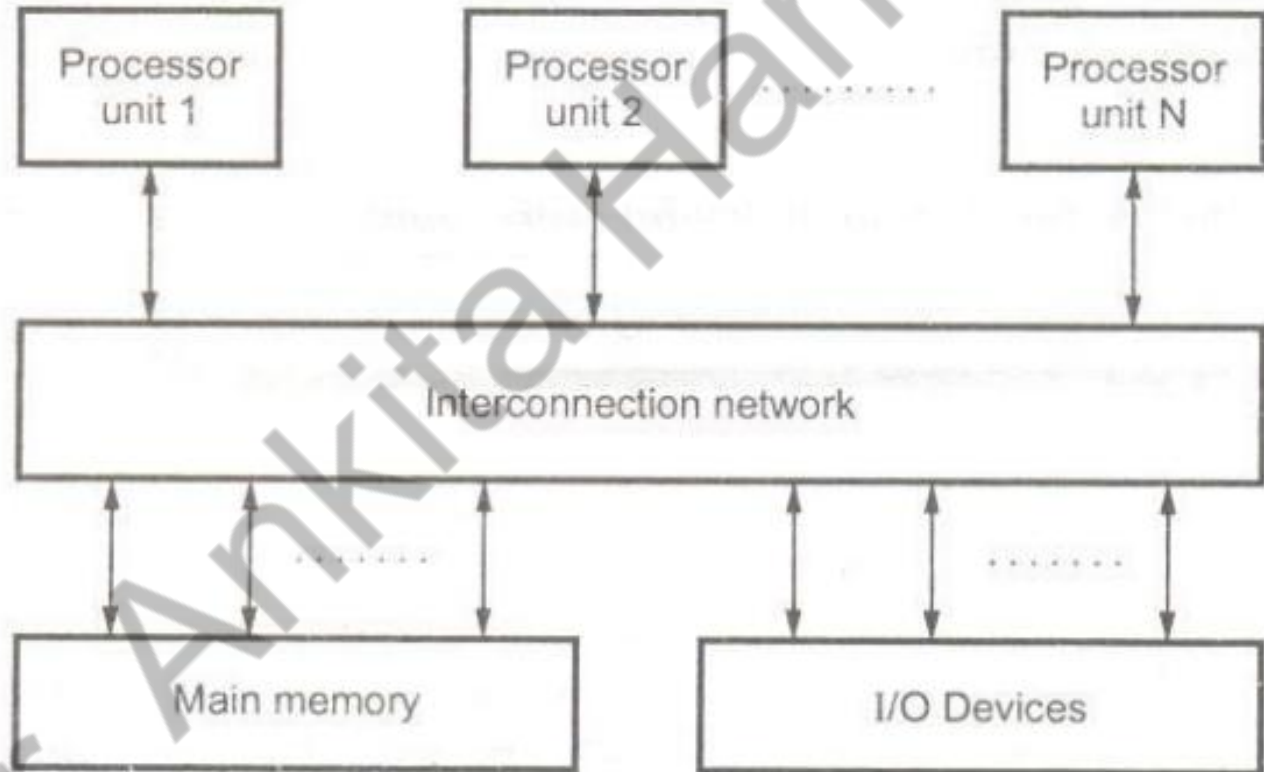
Characteristics

- Two or more similar processes are employed in a stand-alone system.
- Processors share same memory and I/O facilities.
- Processors accesses memory and I/O in approx. equal amount of time.
- Processors are capable of performing the same functions.
- The OS controls the interaction between processors and their programs at different levels.

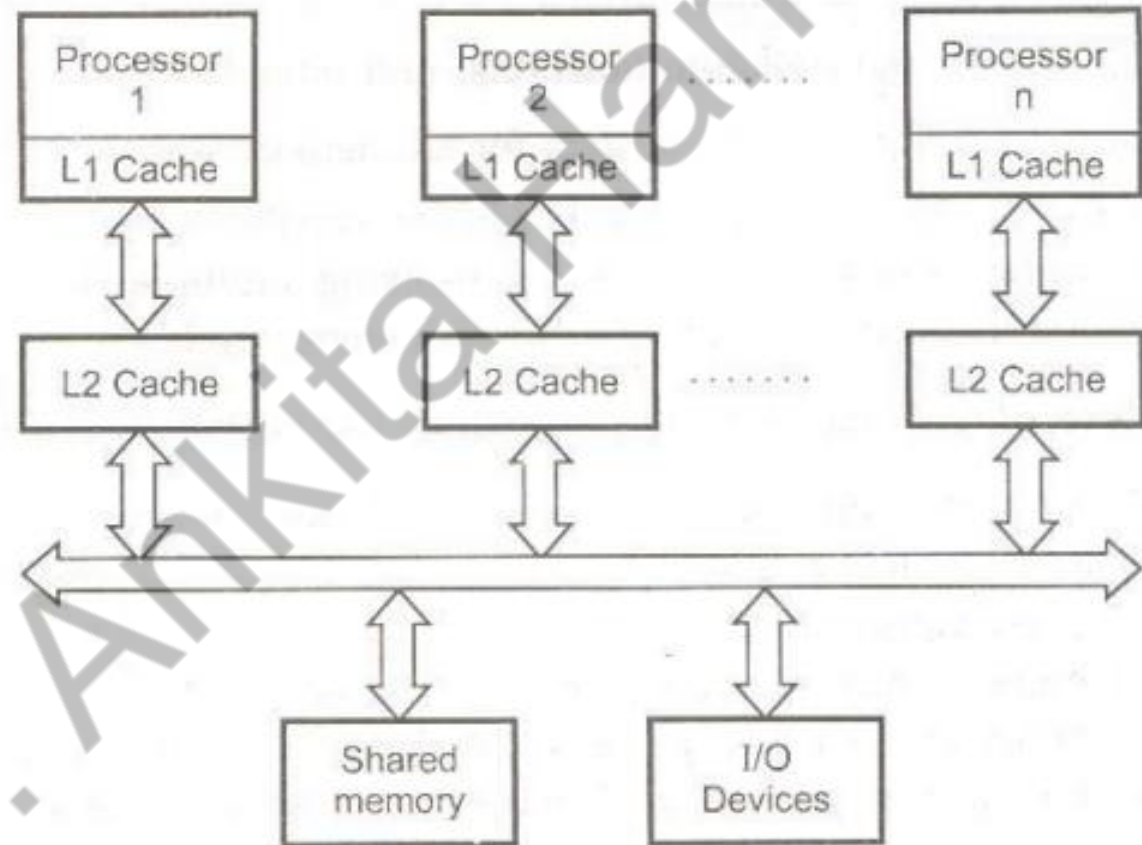
SMP Advantages

- Performance
 - If some work can be done in parallel
- Availability
 - Since all processors can perform the same functions, failure of a single processor does not halt the system
- Incremental growth
 - User can enhance performance by adding additional processors
- Scaling
 - Vendors can offer range of products based on number of processors

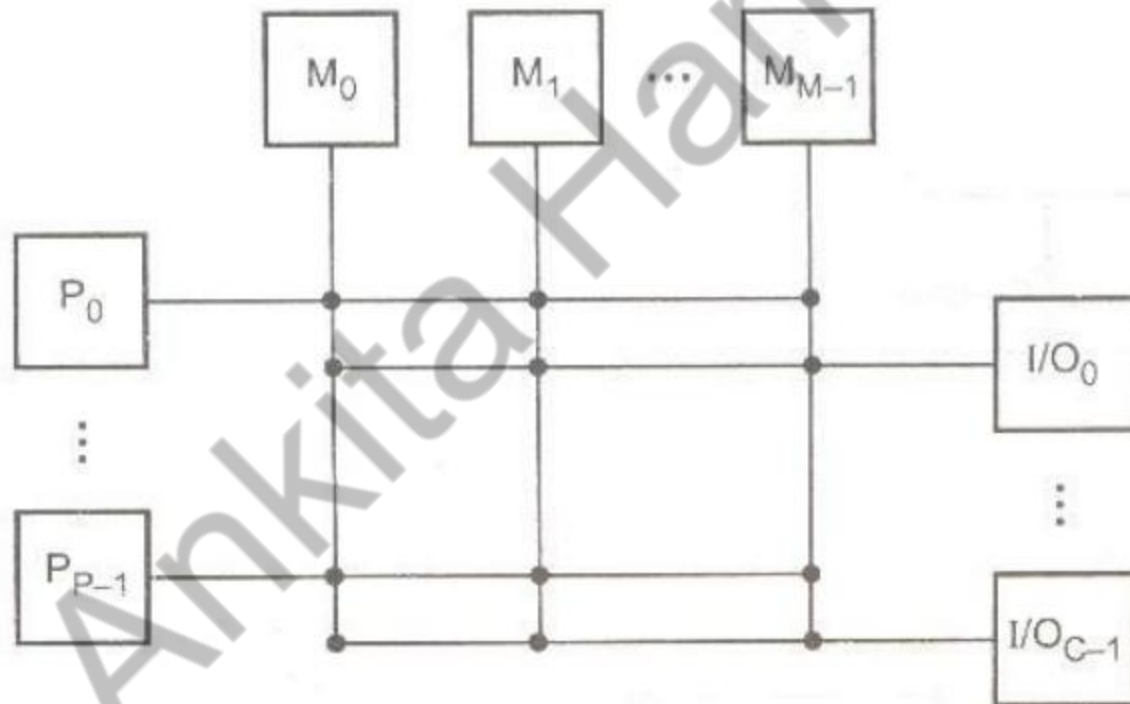
General architecture



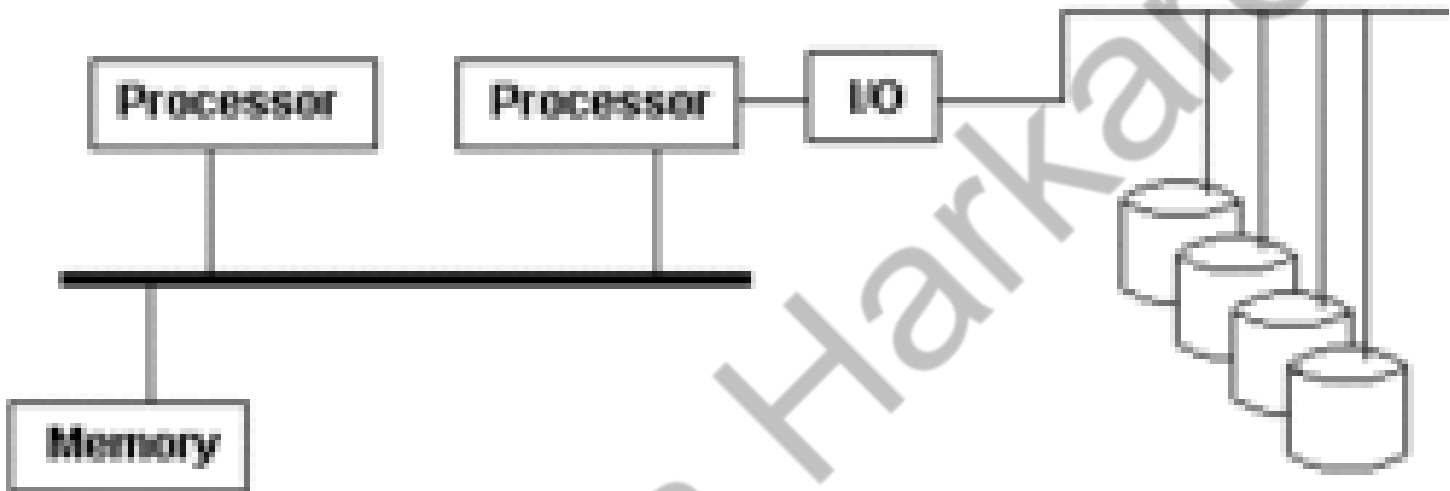
SMP with time shared bus




Crossbar switch system



Asymmetric multiprocessing (AMP) was a software stopgap for handling multiple CPUs before symmetric multiprocessing (SMP) was available. It has also been used to provide less expensive options on systems where SMP was available. In an asymmetric multiprocessing system, not all CPUs are treated equally; for example, a system might only allow (either at the hardware or operating system level) one CPU to execute operating system code or might only allow one CPU to perform I/O operations. Other AMP systems would allow any CPU to execute operating system code and perform I/O operations, so that they were symmetric with regard to processor roles, but attached some or all peripherals to particular CPUs, so that they were asymmetric with regard to peripheral attachment.



Dr. Ankita Harkare



Multiprocessor OS design considerations

- Simultaneous concurrent processes.
- Scheduling
- Synchronization
- Memory management
- Reliability and fault tolerance

Pipelining

Dr. Ankita Harkare

Characterize Pipelines

- Hardware or software implementation – pipelining can be implemented in either software or hardware.
- Large or Small Scale – Stations in a pipeline can range from simplistic to powerful, and a pipeline can range in length from short to long.
- Synchronous or asynchronous flow – A synchronous pipeline operates like an assembly line: at a given time, each station is processing some amount of information. A asynchronous pipeline, allow a station to forward information at any time.
- Buffered or unbuffered flow – One stage of pipeline sends data directly to another one or a buffer is place between each pairs of stages.
- Finite Chunks or Continuous Bit Streams – The digital information that passes though a pipeline can consist of a sequence or small data items or an arbitrarily long bit stream.
- Automatic Data Feed Or Manual Data Feed – Some implementations of pipelines use a separate mechanism to move information, and other implementations require each stage to participate in moving information.

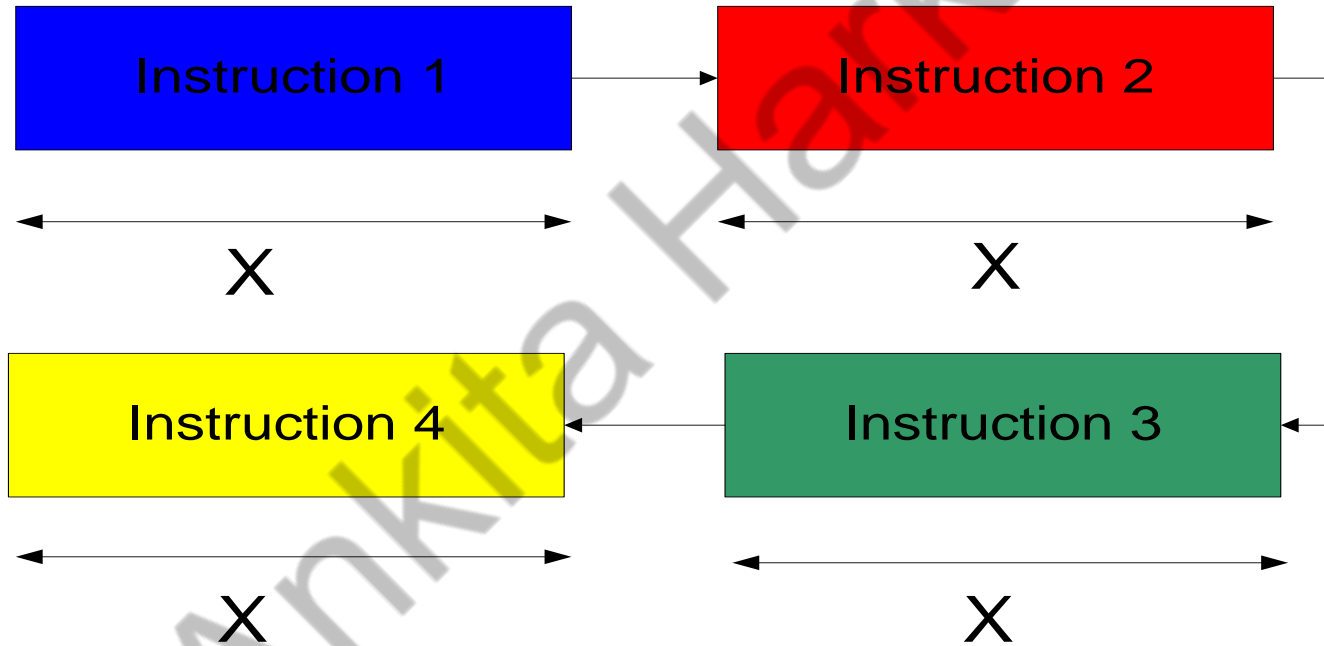
What is Pipelining

- A technique used in advanced microprocessors where the microprocessor begins executing a second instruction before the first has been completed.
- A Pipeline is a series of stages, where some work is done at each stage. The work is not finished until it has passed through all stages.
- With pipelining, the computer architecture allows the next instructions to be fetched while the processor is performing arithmetic operations, holding them in a buffer close to the processor until each instruction operation can be performed.

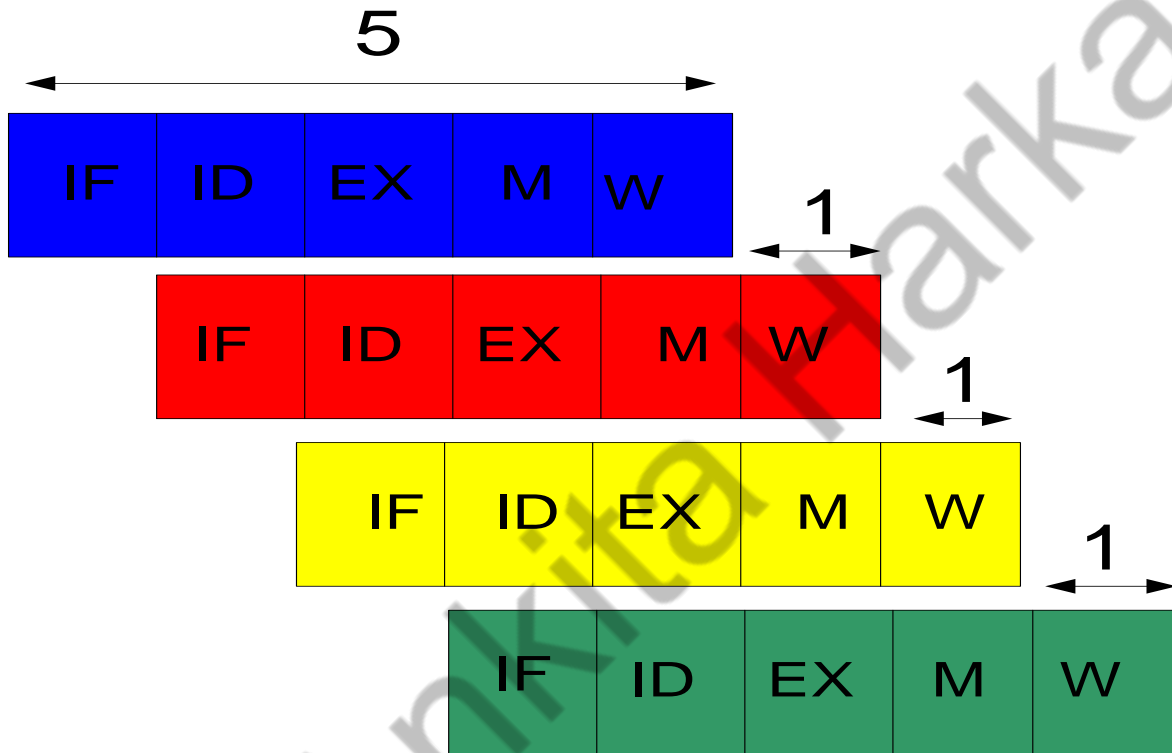
How Pipelines Works

- The pipeline is divided into segments and each segment can execute its operation concurrently with the other segments. Once a segment completes an operation, it passes the result to the next segment in the pipeline and fetches the next operations from the preceding segment.

Example



Four sample instructions, executed linearly



Four Pipelined Instructions

Instructions Fetch

- The instruction Fetch (IF) stage is responsible for obtaining the requested instruction from memory. The instruction and the program counter (which is incremented to the next instruction) are stored in the IF/ID pipeline register as temporary storage so that may be used in the next stage at the start of the next clock cycle.

Instruction Decode

- The Instruction Decode (ID) stage is responsible for decoding the instruction and sending out the various control lines to the other parts of the processor. The instruction is sent to the control unit where it is decoded and the registers are fetched from the register file.

Execution

- The Execution (EX) stage is where any calculations are performed. The main component in this stage is the ALU. The ALU is made up of arithmetic, logic and capabilities.

Memory and IO

- The Memory and IO (MEM) stage is responsible for storing and loading values to and from memory. It also responsible for input or output from the processor. If the current instruction is not of Memory or IO type than the result from the ALU is passed through to the write back stage.

Write Back

- The Write Back (WB) stage is responsible for writing the result of a calculation, memory access or input into the register file.

Operation Timings

- Estimated timings for each of the stages:

Instruction Fetch	2ns
Instruction Decode	1ns
Execution	2ns
Memory and IO	2ns
Write Back	1ns

Advantages/Disadvantages

Advantages:

- More efficient use of processor
- Quicker time of execution of large number of instructions

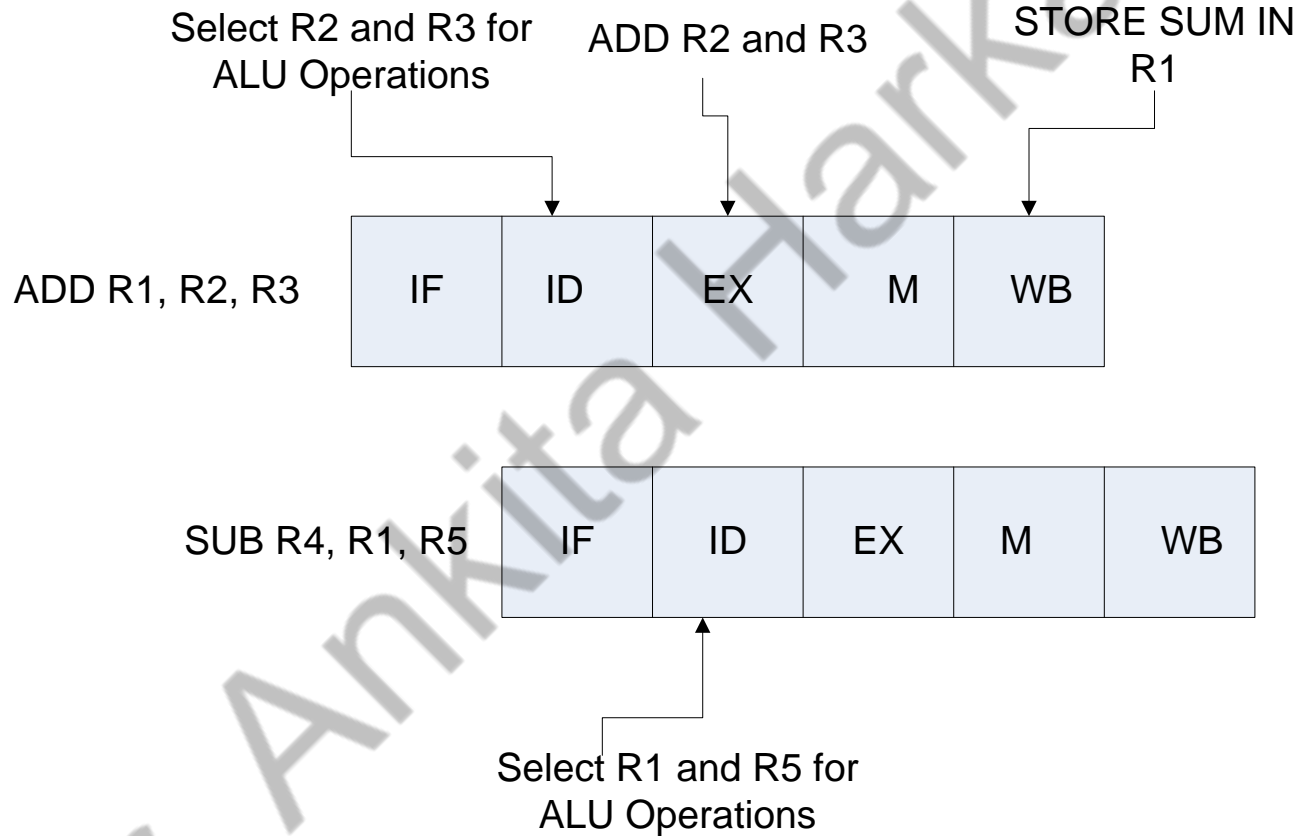
Disadvantages:

- Pipelining involves adding hardware to the chip
- Inability to continuously run the pipeline at full speed because of pipeline hazards which disrupt the smooth execution of the pipeline.

Pipeline Hazards

- Data Hazards – an instruction uses the result of the previous instruction. A hazard occurs exactly when an instruction tries to read a register in its ID stage that an earlier instruction intends to write in its WB stage.
- Control Hazards – the location of an instruction depends on previous instruction
- Structural Hazards – two instructions need to access the same resource

Data Hazards



Stalling

- Stalling involves halting the flow of instructions until the required result is ready to be used. However stalling wastes processor time by doing nothing while waiting for the result.

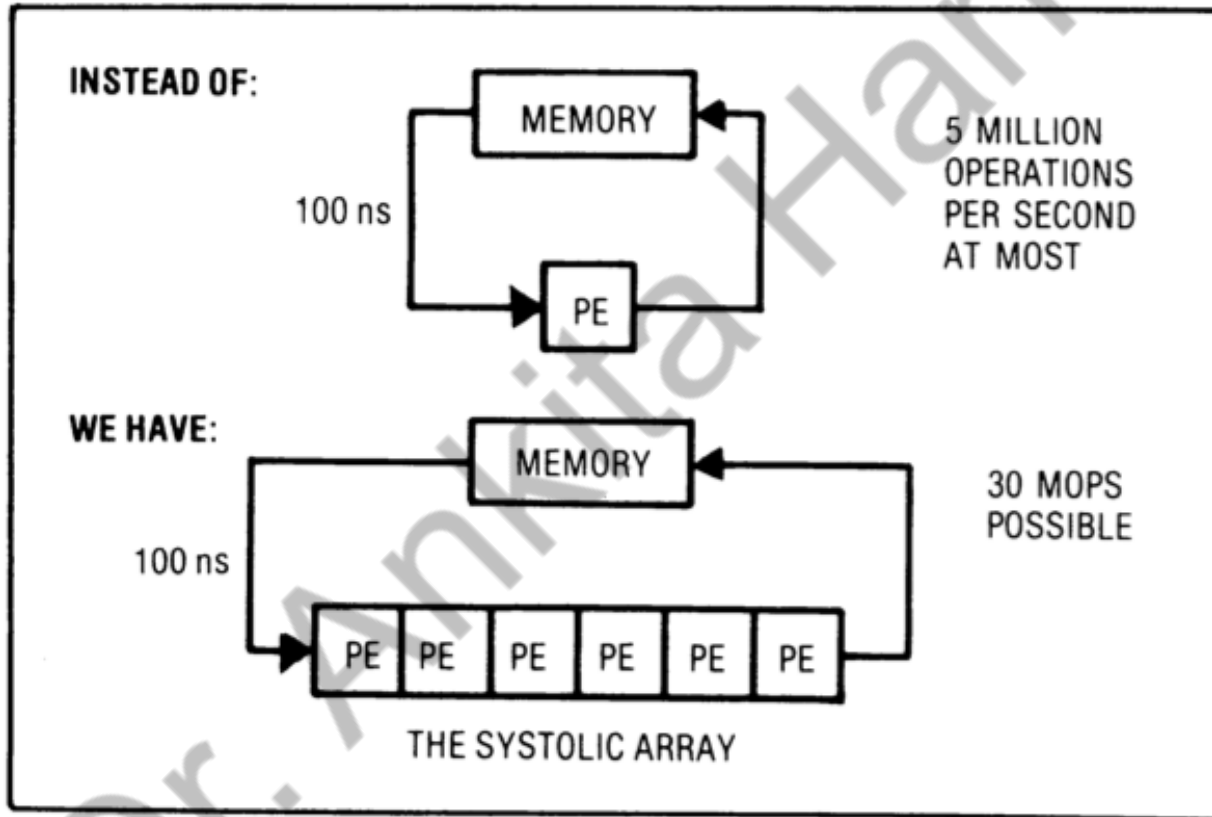
Dr. Ankita Harikare

Why Systolic Architectures?

- Idea: Data flows from the computer memory in a rhythmic fashion, passing through many processing elements before it returns to memory
- Similar to an assembly line
 - Different people work on the same car
 - Many cars are assembled simultaneously
 - Difference: Systolic arch. Can be non-linear and multi-dimensional
- Why? Special purpose accelerators/architectures need
 - Simple, regular designs (keep # unique parts small and regular)
 - High concurrency → high performance
 - Balanced computation and I/O (memory access)

Systolic Architectures

“Why Systolic Architectures?,” IEEE Computer
1982.



Memory: heart
PEs: cells

Memory pulses
data through
cells

Figure 1. Basic principle of a systolic system.

Systolic Architectures

- Basic principle: Replace a single PE with a regular array of PEs and carefully orchestrate flow of data between the PEs → achieve high throughput w/o increasing memory bandwidth requirements

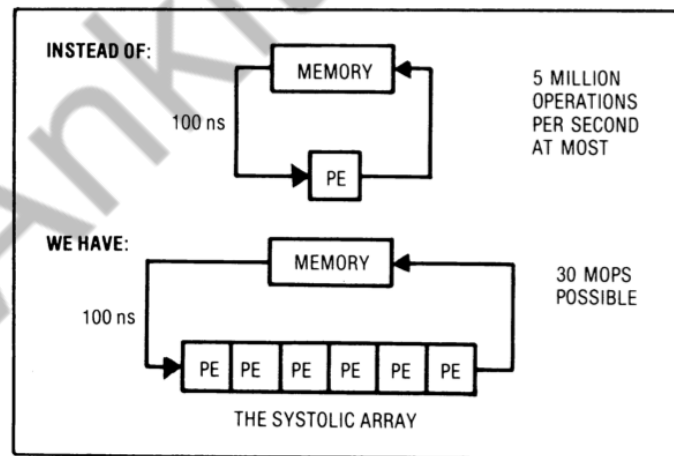


Figure 1. Basic principle of a systolic system.

Systolic Architectures

- Differences from pipelining:
 - Array structure can be non-linear and multi-dimensional
 - PE connections can be multidirectional (and different speed)
 - PEs can have local memory and execute kernels (rather than a piece of the instruction)



Vector Processors

Dr. Ankita Harkare

Overview

- History
- Description
- Advantages
- Disadvantages
- Applications
- Conclusions

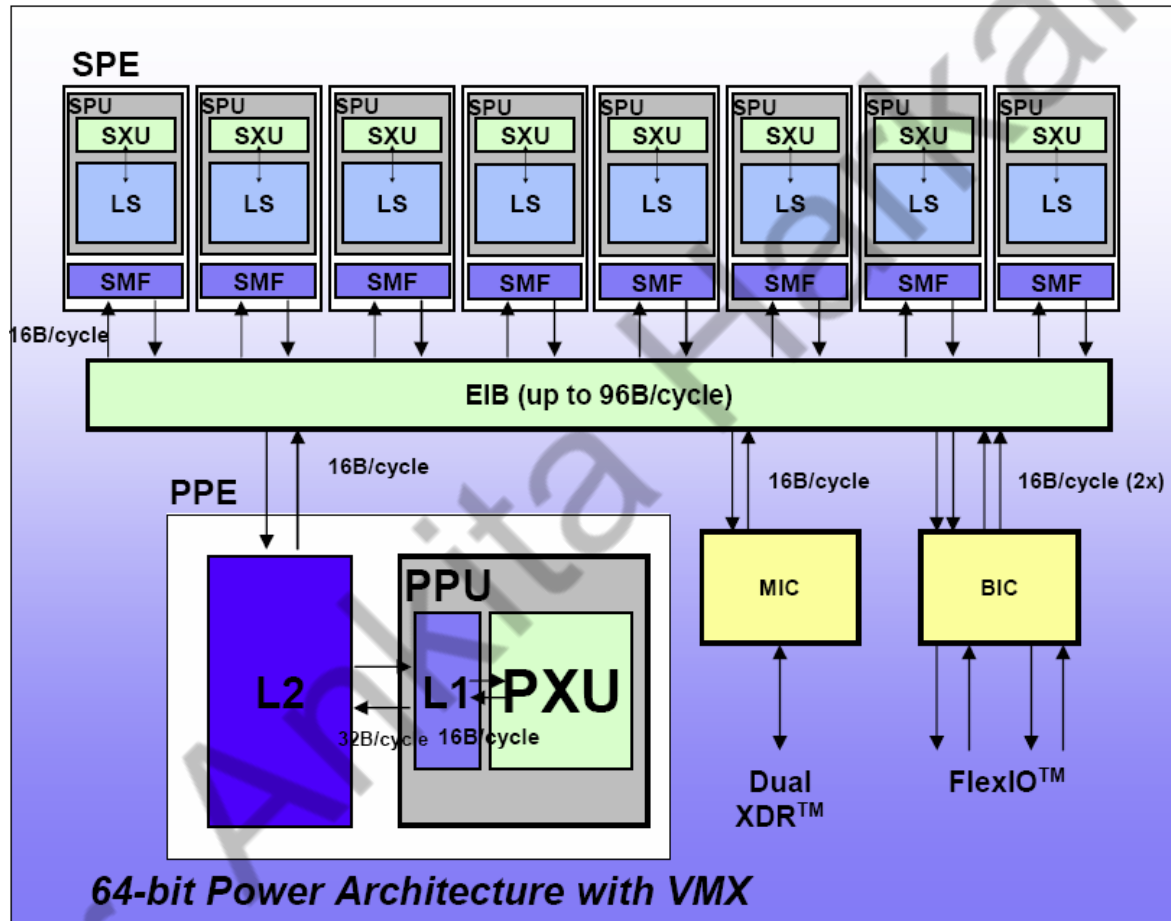
What is a Vector Processor?

- Also called an Array Processor.
- Runs multiple mathematical operations on multiple data elements simultaneously.
- Common in supercomputers of the 1970's 80's and 90's.
- Today most CPU designs contains at least some vector processing instructions, typically referred to as SIMD.
- Typically operate on a few vectors elements per clock cycle in a pipeline v. SIMD which will operate on all at once.

History

- 1962 University of Illinois Illiac IV - completed 1972 with 64 ALUs 100-150 MFlops (massively parallel computer)
- (1973) TI's Advance Scientific Computer (ASC) 20-80 MFlops
- (1975) Cray-1 first to have vector registers instead of keeping data in memory (8 registers with 64 64-bit words in each)
- Cray-1 had separate pipelines for different instruction types allowing vector chaining. 80-240 MFlops

How It Works



- Typical Vector Processor (Cell Processor)³

Advantages

- Each result is independent of previous results - allowing deep pipelines and high clock rates.
- A single vector instruction performs a great deal of work - meaning less fetches and fewer branches (and in turn fewer mispredictions).
- Vector instructions access memory a block at a time which allows memory latency to be amortized over many elements.
- Vector instructions access memory with known patterns, which allows multiple memory banks to simultaneously supply operands.
- Less memory access = faster processing time.

Disadvantages

- Not as fast with scalar instructions
- Complexity of the multi-ported VRF
- Difficulties implementing precise exceptions
- High price of on-chip vector memory systems
- Increased code complexity

Applications

- Servers
- Home Cinema
- Super Computing
- Cluster Computing
- Mainframes
- “Astrophysicist Replaces Supercomputer With 8 PS3’s” ²



Dr. Ankita Harkare

Thank You



Dr. Ankita Harkare



Dr. Ankita Harkare



Dr. Ankita Harkare



Dr. Ankita Harkare



Dr. Ankita Harkare