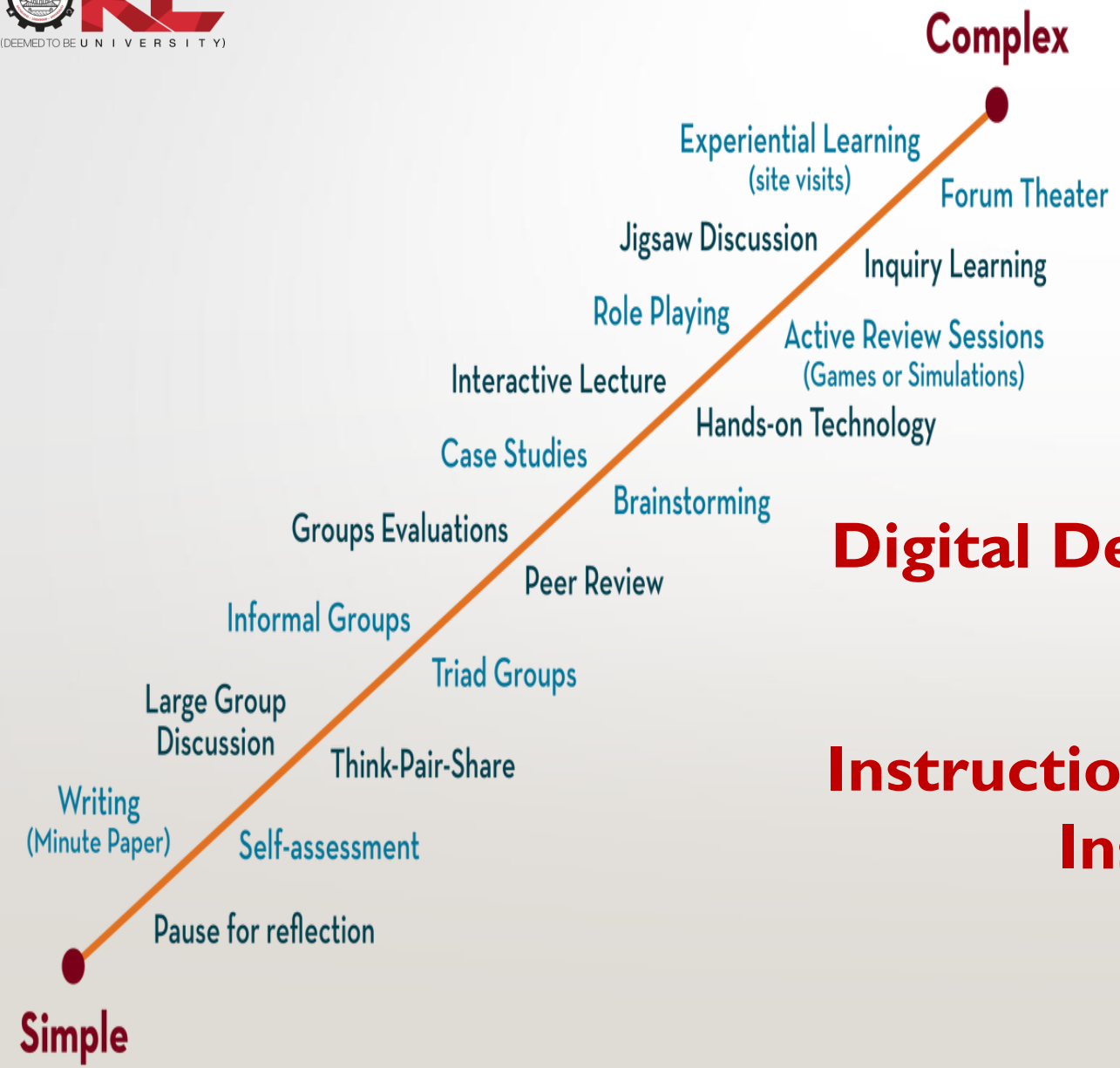


Department of BES-II

Digital Design and Computer Architecture

23EC1202

Topic:

Instruction pipelining and pipeline hazards, Instruction level parallelism

Session No: 25

AIM OF THE SESSION

To understand the concept of instruction pipelining and identify, analyze, and mitigate pipeline hazards

INSTRUCTIONAL OBJECTIVES

This Session is designed to:

1. Comprehend the Concept of Instruction Pipelining
2. Identify and Analyze Pipeline Hazards
3. Capable of analyzing a given pipeline scenario, implementing suitable hazard mitigation

LEARNING OUTCOMES

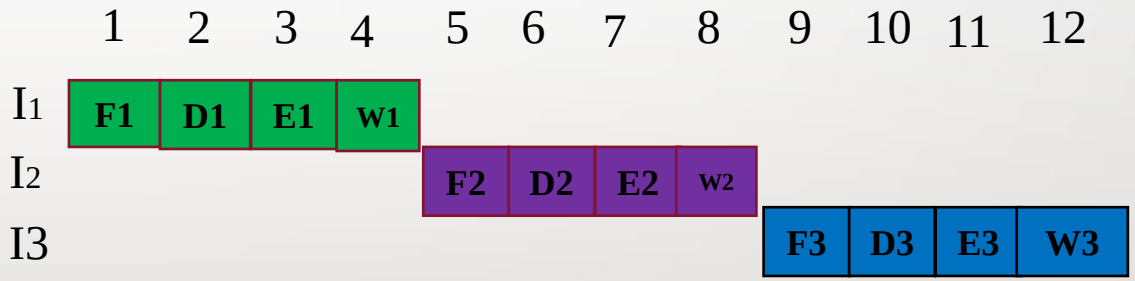
At the end of this session, you should be able to:

1. Proficiency in Understanding Pipeline Concepts
2. Ability to Identify and Mitigate Pipeline Hazards.
3. Application of Optimization Techniques in Pipelined Architectures.

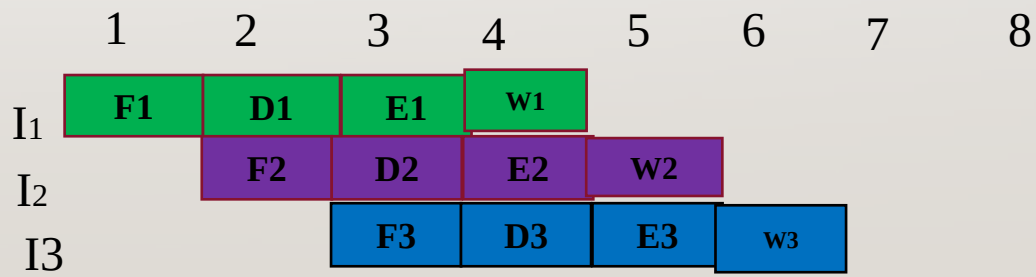
Introduction to pipelining

- Pipelining is a technique of decomposing a sequential process into sub-operations, with each sub-process being executed in a special dedicated segment that operates concurrently with all other segments.

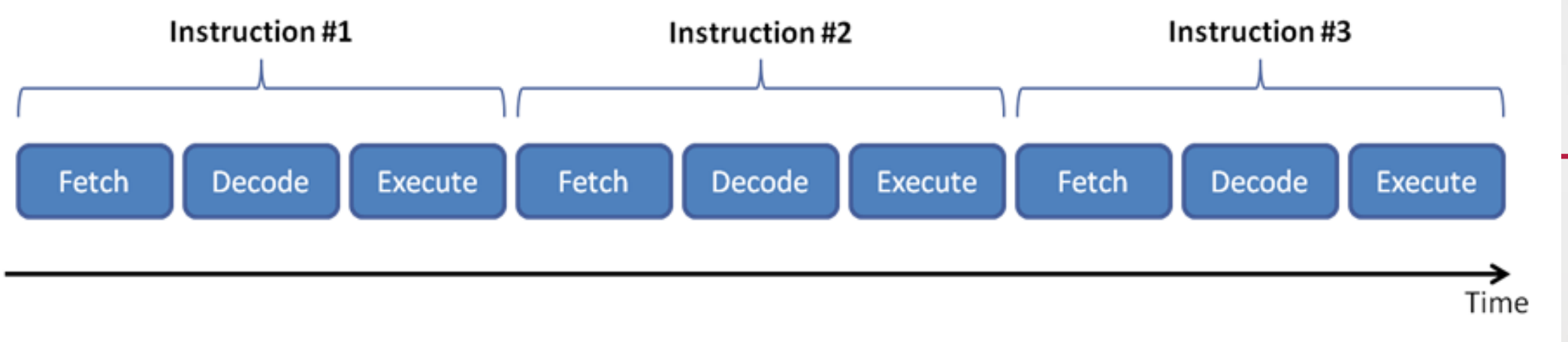
Conventional Sequential Execution



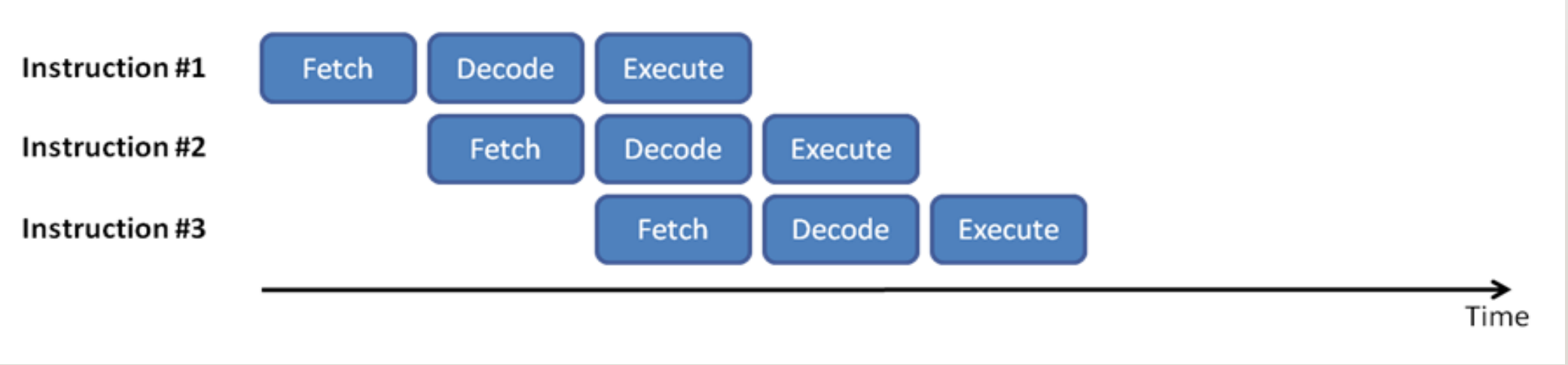
Pipelined Execution



Conventional Sequential Execution



Pipelined Execution



Four Stage Pipeline

F: Fetch, Read the instruction from the memory

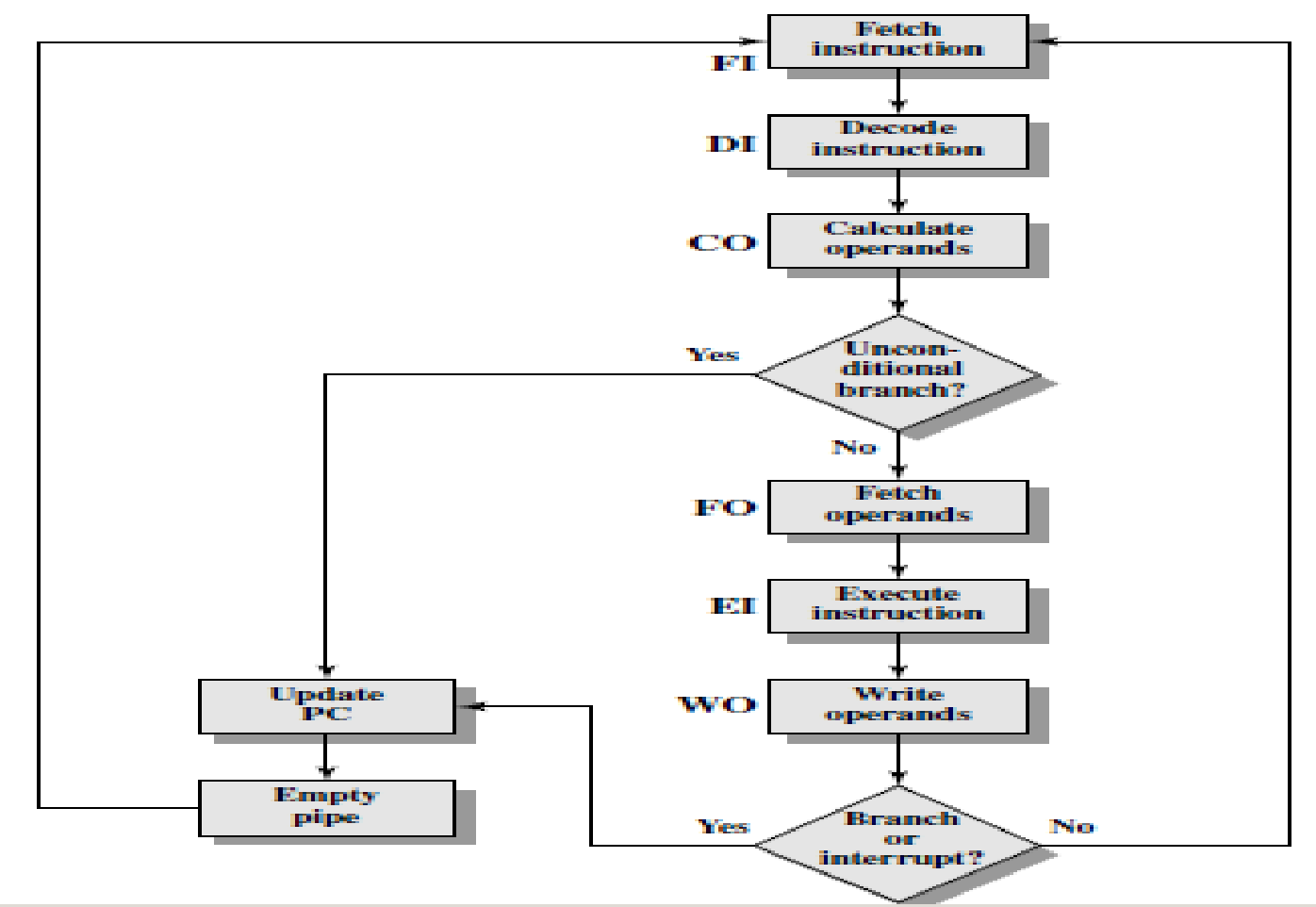
D: Decode, decode the instruction and fetch the source operand (S)

E: Operate, perform the operation

W: Write, store the result in the destination location.

Step	1	2	3	4	5	6	7	8	9
1	F1	D1	E1	W1					
2		F2	D2	E2	W2				
3			F3	D3	E3	W3			
4				F4	D4	E4	W4		
5					F5	D5	E5	W5	
6						F6	D6	E6	W6

Instruction Execution in a Six Stage Pipeline



Timing diagram of Six stage Pipelining

Time →

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Instruction 1	FI	DI	CO	FO	EI	WO								
Instruction 2		FI	DI	CO	FO	EI	WO							
Instruction 3			FI	DI	CO	FO	EI	WO						
Instruction 4				FI	DI	CO	FO	EI	WO					
Instruction 5					FI	DI	CO	FO	EI	WO				
Instruction 6						FI	DI	CO	FO	EI	WO			
Instruction 7							FI	DI	CO	FO	EI	WO		
Instruction 8								FI	DI	CO	FO	EI	WO	
Instruction 9									FI	DI	CO	FO	EI	WO

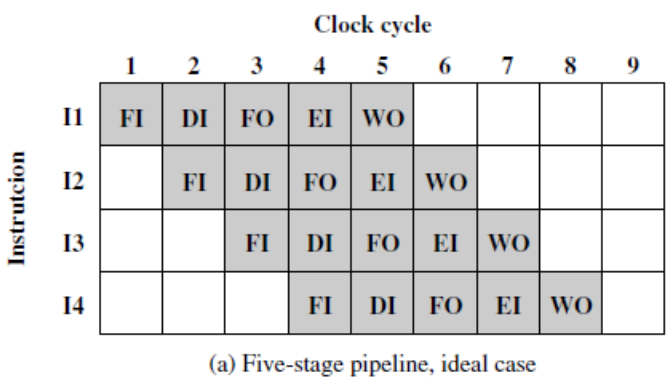
Pipeline Hazards

- While pipelining can significantly enhance performance, it also introduces several design issues and challenges.
- When the pipeline, or some portion of the pipeline, must stall because conditions do not permit continued execution then pipeline hazard occurs. Such a pipeline stall is also referred to as a pipeline bubble.
- **Structural hazards:** They arises from resource conflicts when the hardware not support all possible combinations of instruction simultaneously.
- **Data hazards:** They arises when an instruction depends on the result of previous instructions.
- **Control hazard:** Occur due to the pipelining of branches and other instructions that change the PC (Program Counter)

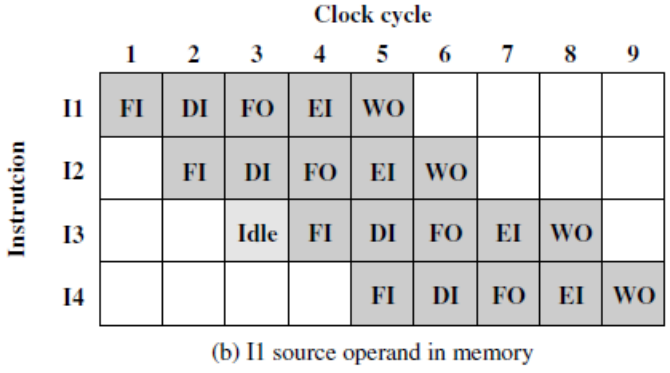
STRUCTURAL HAZARDS

DATA HAZARDS

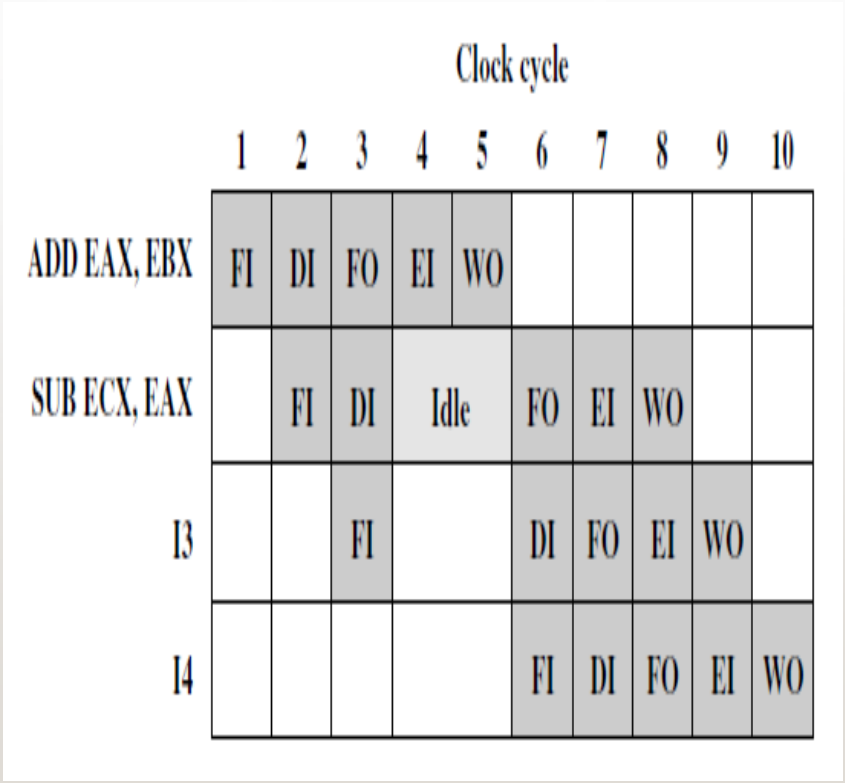
CONTROL HAZARDS



(a) Five-stage pipeline, ideal case

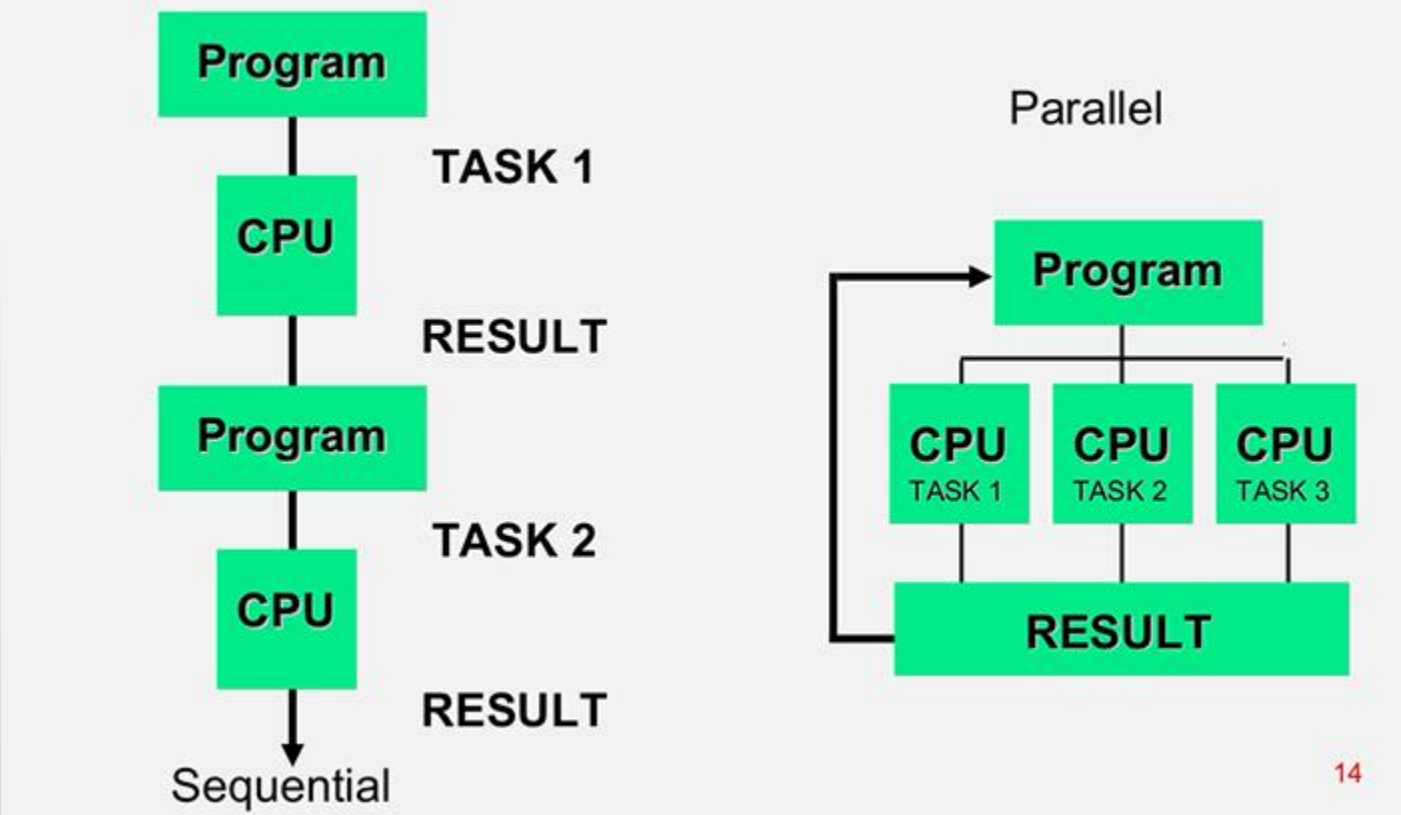


(b) I1 source operand in memory



- Multiple streams
- Pre-fetch branch target
- Loop buffer
- Branch prediction
- Delayed branch

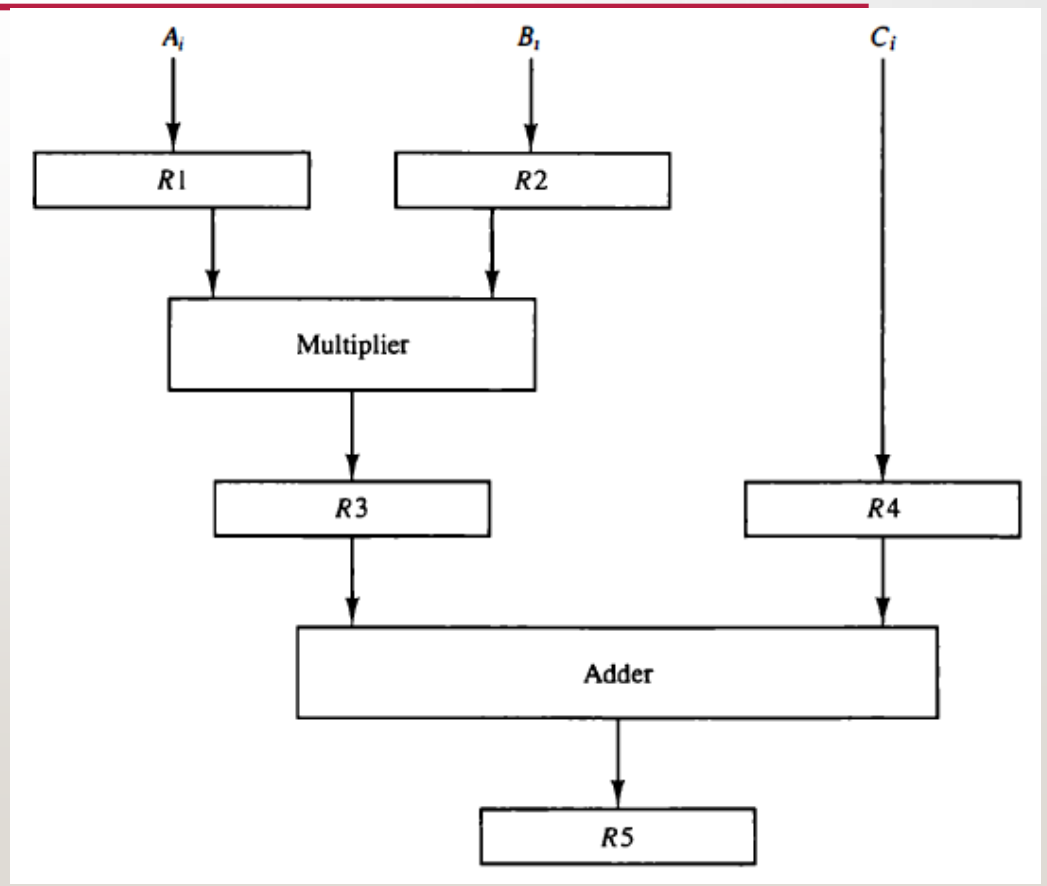
Serial Vs Parallel Computing



How Parallel processing works ?

$$A_i * B_i + C_i \qquad \text{for } i = 1, 2, 3, \ldots, 7$$

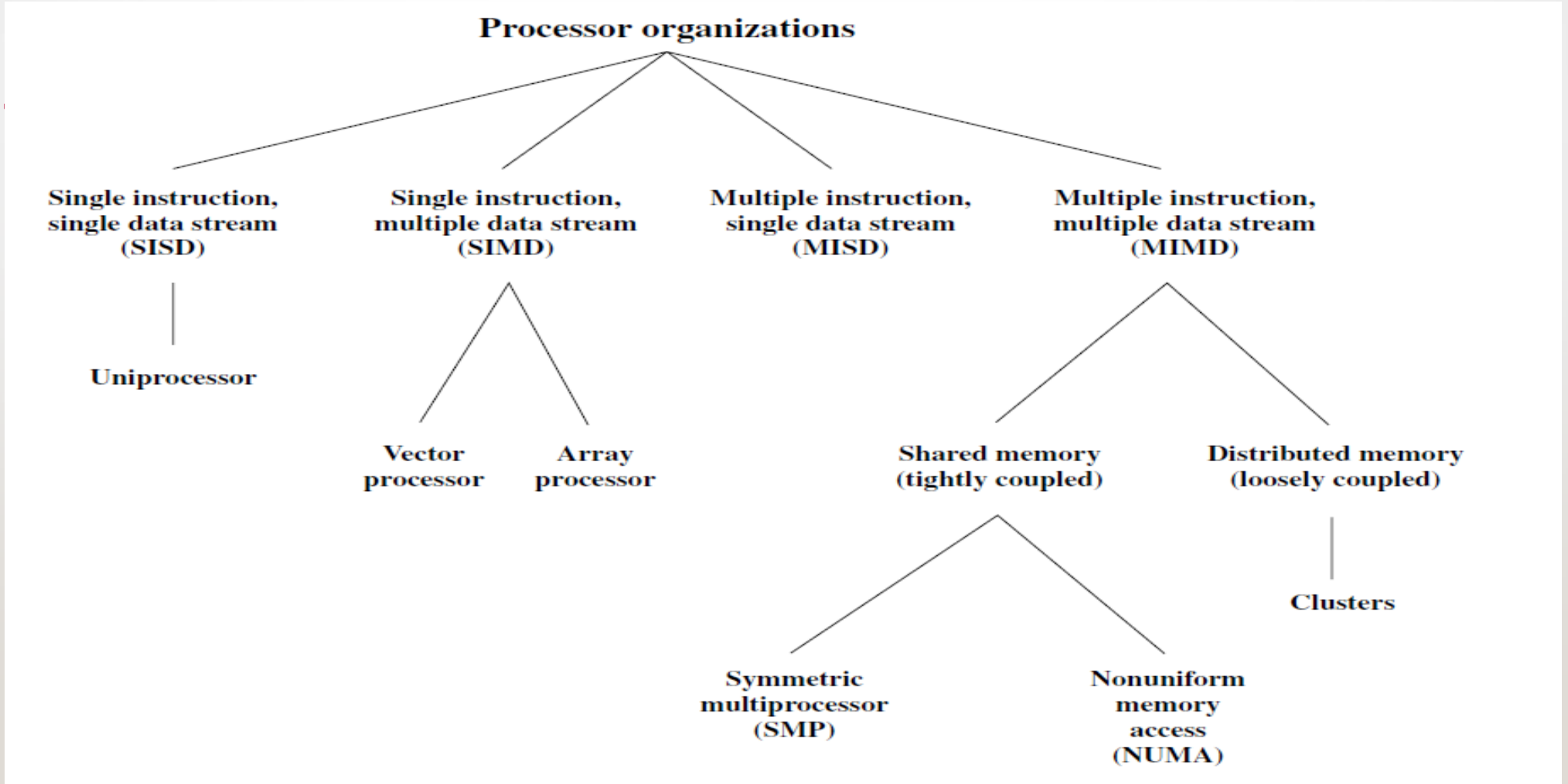
$R1 \leftarrow A_i, \quad R2 \leftarrow B_i$	Input A_i and B_i
$R3 \leftarrow R1 * R2, \quad R4 \leftarrow C_i$	Multiply and input C_i
$R5 \leftarrow R3 + R4$	Add C_i to product



Parallel processing

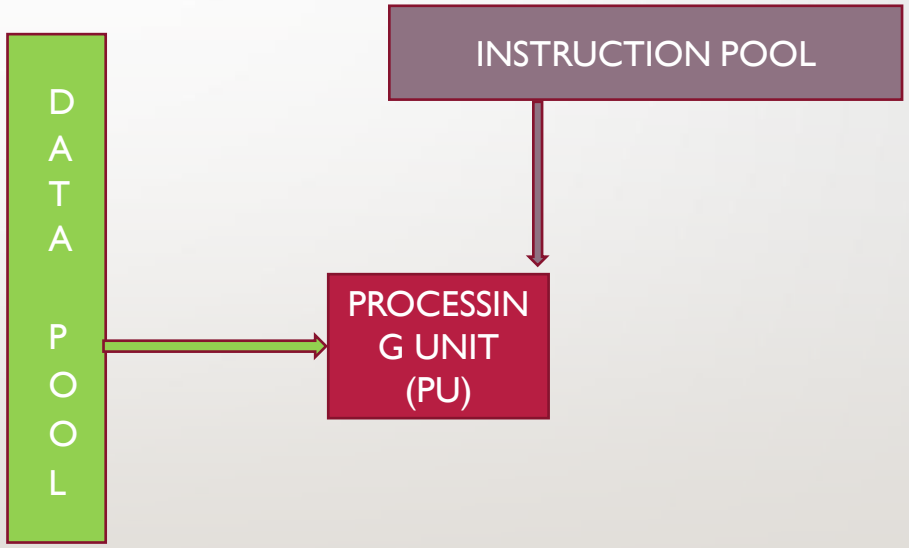
- Parallel processing may occur in the instruction stream, in the data stream, or in both.
- Flynn's classification divides computers into four major groups as follows:
    1. Single instruction stream, single data stream (SISD)
    2. Single instruction stream, multiple data stream (SIMD)
    3. Multiple instruction stream, single data stream (MISD)
    4. Multiple instruction stream, multiple data stream (MIMD)

Parallel processing Architectures



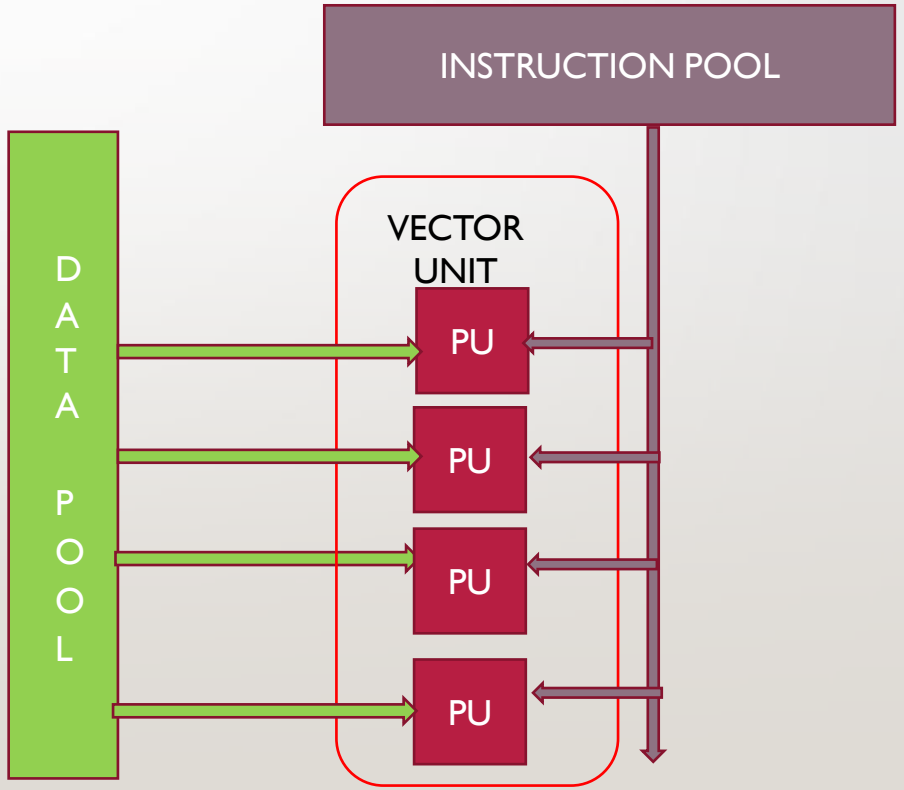
Single Instruction, Single data stream(SISD)

- A computer architecture in which a single uni-core processor executes a single instruction stream, to operate on data stored in a single memory.



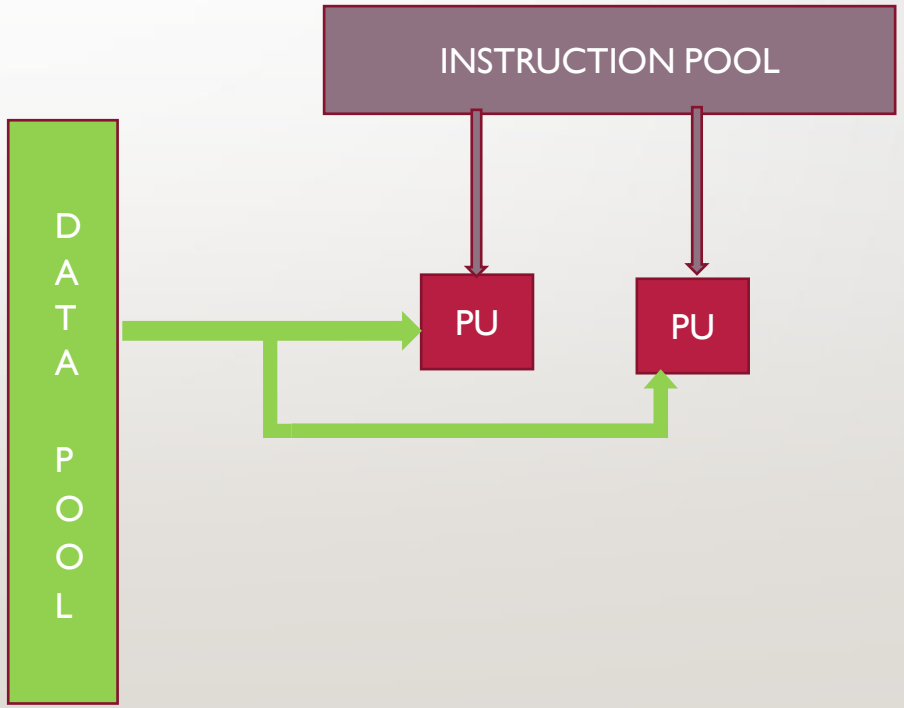
Single Instruction, Multiple data stream(SIMD)

- A single machine instruction controls the simultaneous execution of a number of processing elements on a lockstep basis.
- Each processing element has an associated data memory, so that each instruction is executed on a different set of data by the different processors. Vector and array processors fall into this category



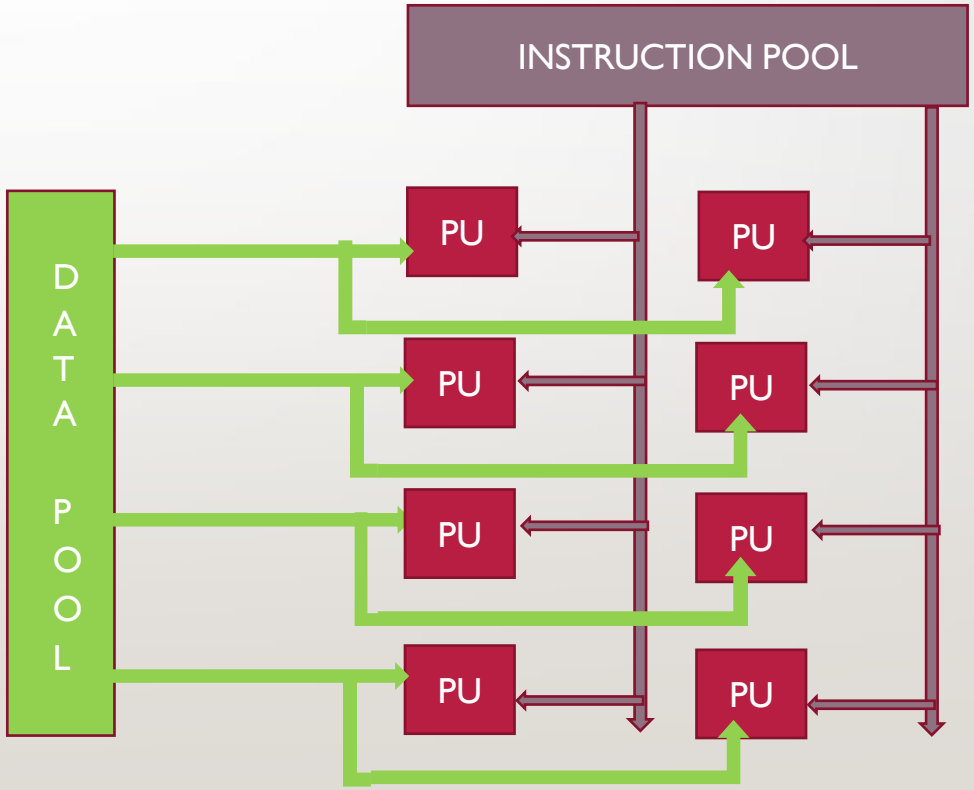
Multiple Instruction, Single Data Stream (MISD)

- A sequence of data is transmitted to a set of processors, each of which executes a different instruction sequence.
- This structure has never been implemented.



Multiple Instruction, Multiple Data Stream (MIMD)

- A set of processors simultaneously execute different instruction sequences on different data sets.
- Symmetrical multiprocessing (SMP) systems, clusters, and Non-uniform memory access (NUMA) systems fits into this category.



SELF-ASSESSMENT QUESTIONS

1. What is instruction pipelining?

A) A technique to reduce the execution time of instructions by breaking down the instruction execution process into several stages.
B) A process to execute multiple instructions in a single clock cycle.
C) A method to increase the clock speed of the processor.
D) A technique to reduce power consumption in processors.

2. What is a data hazard in instruction pipelining?

A) When the pipeline must be stopped due to unavailability of data.
B) When the next instruction cannot execute because it is waiting for the previous instruction to complete its data read/write.
C) When data is corrupted during transmission between stages.
D) When data is not available due to a cache miss.

SELF-ASSESSMENT QUESTIONS

3. Instruction-level parallelism (ILP) refers to:

A) Executing multiple programs at the same time.
B) Parallel execution of instructions from a single program.
C) Increasing the number of pipelines in a processor.
D) Reducing the instruction execution time by optimizing code at the compiler level.

4. Which of the following techniques is used to enhance instruction-level parallelism?

A) Out-of-order execution.
B) Increasing cache size.
C) Decreasing clock speed.
D) Reducing pipeline stages.

TERMINAL QUESTIONS

Short answer questions:

1. Illustrate the cause of structural hazards in pipelining.

Long answer questions:

1. Design the operational flow for a pipelined processor with four stages, outlining each stage's function and how instructions progress through the pipeline.
2. In order to determine the city with the most consistently warm temperatures from a large weather dataset, which **parallel processing architecture (SISD, SIMD, MISD, MIMD)** would be best suited for efficient analysis and why?
3. Investigate the concept of pipelining hazards in processor design, detailing the types of hazards, their causes.

REFERENCES FOR FURTHER LEARNING OF THE SESSION

Reference Books:

1. Computer Organization by Carl Hamacher, Zvonko Vranesic and Saftwat Zaky.
2. Computer System Architecture by M. Morris Mano
3. Computer Organization and Architecture by William Stallings

Sites and Web links:

1. https://www.geeksforgeeks.org/instruction-level-parallelism/
2. https://www.prepbytes.com/blog/computer-architecture/instruction-pipeline-and-hazards/

THANK YOU

(DEEMED TO BE UNIVERSITY)

Team – Digital Design & Computer Architecture