

1. What are the classification of data hazards?

ans Classification of Data Hazards:

- Read After Write (RAW)
- Write After Read (WAR)
- Write After Write (WAW)

2. How to prevent Data Hazards in Pipelining?

- operand forwarding (data forwarding)
- Instruction reordering
- Pipeline Stalling (Inserting NOPs)

3. State different types of hazards that can occur in Pipeline.

ans Types of Hazards in Pipelines:

- Data hazards
- Control hazards
- Structural hazards

4. What are Hazards?

ans • Hazards are problems that prevent the next instruction in the pipeline from executing during its designated clock cycle.

## Section - B

1. Differentiate between Vector and array Processor.

ans Vector vs Array Processor:

- Vector Processor: Operates on entire Vectors (Arrays) with a single instruction.
- Array Processor: Has multiple processing elements working in parallel on different data.
- key difference: Vector Processors use Pipelined ALUs; Array Processors use multiple ALUs simultaneously.

2. Operand forwarding Example:

- Used to resolve RAW hazards by sending output directly from one pipeline stage to another.

• Example: If instruction 1 computes  $R1 = R2 + R3$ , and instruction 2 uses  $R1$ , operand forwarding sends  $R1$ 's value to instruction 2 without waiting.



## Sec - C

1. Differentiate between RISC and CISC.

- RISC (Reduced Instruction Set Computer): Simple Instructions, fast execution, fixed Instruction length.

- CISC (Complex Instruction Set Computer): Complex instructions, fewer instructions per Program, variable length.

2. Explain Flynn's classification in details

ans Flynn's Classification:

- SISD: Single Instruction Single Data.
- SIMD: Single Instruction multiple Data.
- MISD: multiple Instruction single Data (Rare).
- MIMD: Multiple Instruction multiple

Data (most modern processors)

- SISD: One processor executes a single instruction on a single data stream.

Ex: Traditional uniprocessor systems.

- SIMD: One instruction is applied to multiple data elements in parallel.

Ex: GPUs, Vector processors.

- MISD: multiple processors execute different instructions on the same data stream (rarely used).

Ex: fault-tolerant systems.

- MIMD: multiple processors execute different instructions on different data independently.

Ex: multi-core processors, clusters.