

NP and Parallel computation.

Write difference between P and NP-Problem.

P (Polynomial Time)

definition problems solvable in Polynomial time

examples Sorting, searching, shortest Path

solution solved in polynomial time (efficient)

verification easy to verify solutions in polynomial time

complexity subset of NP

class

determinism solved using deterministic algorithm

false of solution solvable efficiently with current algorithms.

P vs NP
Relationship

$P \subseteq NP$

NP (Non-deterministic Polynomial time)

problems verifiable in Polynomial time.

sudoku, traveling salesman, integer factorization

verification takes polynomial time.

same as P

superset of P

verified using non-deterministic algorithms.

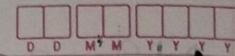
may not be solvable efficiently.

$NP \supseteq P$

Use in
cryptography

Not useful for most cryptographic problems

Basis of many cryptographic schemes.



Reduction P Problems cannot be reduced to NP-complete easily.

NP Problems can be reduced to NP-complete.

Real-world Impact Efficient solutions are practical.

Many real-world problems fall into this class.

Ques- Explain SISD and SIMD Computer Parallel computing system.

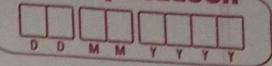


① SISD (Single Instruction Single Data) :-

- SISD refers to a computing system where a single processor executes one instruction on one data element at a time.
- This is the most basic and traditional computing model, often associated with serial or sequential execution.
- It represents the classical Von Neumann architecture, where instruction and data are processed in a step-by-step manner.

Characteristics :-

- ① Instruction Execution :- only one instruction is executed at any given moment.
- ② Data processing :- operates on one data element per instruction cycle.
- ③ Architecture :- Involves a single control unit and single processing unit.



- 4) Parallelism :- No inherent parallelism; tasks are executed sequentially.
- 5) Complexity :- simple design with straightforward hardware requirements.
- 6) Memory :- Lower memory requirements since there's no need to handle multiple data stream concurrently.

Applications :-

- 1) General-Purpose computing.
- 2) control systems that require sequential decision-making.
- 3) Embedded system.

Example :-

- 1) ENIAC and UNIVAC
 - 2) Intel 8086, Intel 8051
- 2) SIMD (Single Instruction Multiple Data) :-
- SIMD systems allow a single instruction to be executed on multiple data elements simultaneously.
 - This architecture is designed for tasks that involve applying the same operation across large datasets, making it highly effective for parallel processing.

Characteristics :-

- 1) Instruction Execution :- one instruction is broadcasted and applied across multiple data elements in parallel.
- 2) Data Processing :- simultaneously processes multiple data streams with a single instruction.

- ③ Architecture:- consists of multiple Processing units controlled by a single control unit. Each processing unit works on a different data element.
- ④ Parallelism:- High degree of data parallelism, significantly improving performance for large scale computations.
- ⑤ Complexity:- More complex to design due to the need for synchronization among multiple processing unit.
- ⑥ Memory:- Requires more memory to manage and process multiple data streams at once.

Applications:-

- ① Graphics Processing
- ② Scientific simulation
- ③ Machine Learning.
- ④ Signal Processing.

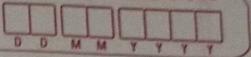
Examples:-

- ① Modern GPUs :- NVIDIA's CUDA cores
- ② Vector Processors :- Intel's AVX

Ques- Write note on Hypercube Parallel model.

Hypercube Parallel model:-

The Hypercube Parallel model is a high-performance computing architecture designed for parallel processing, characterized by its unique geometric structure that allows efficient communication and computation across multiple processors.



- It is based on a hypercube topology, which is a multi-dimensional extension of a cube.

Key Features:-

1) Topology:-

- The hypercube is an n -dimensional cube where each vertex represents a processor.
- Each processor is connected to n other processors, corresponding to a hypercube edge in each dimension.

For Example:-

2) Communication

- A 1-dimensional hypercube (1D) is a line with 2 nodes.
- A 2-dimensional hypercube (2D) is a square with 4 nodes.
- A 3-dimensional hypercube (3D) is a cube with 8 nodes.

3) Communication Efficiency:-

- The maximum number of hops between any two nodes is n , making communication highly efficient.
- This reduces communication overhead and improves parallel processing performance.

4) Scalability:-

- The model is highly scalable; doubling the number of processors only requires increasing the dimension by 1.
- For 2^n processors, the number of connections grows logarithmically with the number of nodes, ensuring manageable complexity.

Q) Fault Tolerance:-

- Due to multiple paths between nodes, the hypercube model exhibits fault tolerance.
- If a node or connection fails, alternative paths are available.

Advantages:-

- 1) Low Latency
- 2) High connectivity
- 3) Balanced Load Distribution.

Challenges:-

- 1) Complex wiring
- 2) Cost

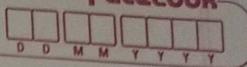
Applications:-

- 1) Scientific simulations
- 2) Data mining and ML
- 3) Distributed Databases.

Ques:- Explain PRAM parallel computational model.
 ⇒

~~PRAM (Parallel Random Access Machine) :-~~

- The PRAM is a theoretical model for designing and analyzing parallel algorithms.
- It assumes multiple processors working in parallel, accessing a shared memory, with the goal of solving computational problems efficiently.
- PRAM is often used as a foundational concept in parallel computing to evaluate



The Performance and scalability of Parallel algorithms.

Key Components of PRAM :-

1) Processors (P) :-

- A set of identical processors, each capable of executing instructions independently.
- All processors operate synchronously, meaning they execute instructions in lock step.

2) Shared memory (m) :-

- A global memory that all processors can access directly.
- Memory is assumed to have uniform access time (i.e. no delays based on memory location.)

3) Instruction Set :-

- Processors can perform basic operations (e.g. read, write, arithmetic, logic).
- The model ignores communication overhead between processors and memory.

PRAM variants Based on Memory Access :-

~~- PRAM models are classified based on how processors handle concurrent memory access:-~~

1) EREW (Exclusive Read, Exclusive Write) :-

- ~~- No two processors can read from or write to the same memory location simultaneously.~~
- ~~- safest and most restrictive model, preventing race conditions.~~

② QREW (concurrent Read, Exclusive write) :-

— Multiple processors can read from the same memory location simultaneously, but only one processor can write at a time.

③ ERCW (Exclusive Read, concurrent Write) :-

— only one processor can read from a memory location at a time, but multiple processors can write simultaneously.

④ CRCW (concurrent Read, concurrent Write) :-

— multiple processors can both read and write to the same memory location simultaneously

— Conflict Resolution strategies :-

① Common

② Arbitrary

③ Priority

Advantages :-

- ① Simplified Algorithm design
- ② Parallel speedup Analysis
- ③ Foundation for Real systems.

Limitations :-

- ① Unrealistic Assumptions
- ② Scalability issues
- ③ High complexity.

Applications :-

- ① Parallel Sorting Algorithms
- ② Graph Algorithms
- ③ Matrix multiplication.

②
30/11/24