23/8/22 (PDC)

single chip
single core system.

single chip
multi core system.

↳ It allows parallelism.

| Parallel | vs | concurrent |
|---|---|---|
| • run tasks in parallel | | • may give illusion of parallelism. |

→ PDS : Geographically distributed systems to perform task parallely.

19/9/2022 "PDC NOTES"

"Chap # 01"

- **Motivating Parallelism:**

→ Developing parallel hardware and software has traditionally been time and effort intensive.

- **Need for parallelism:**

→ Advances in microprocessor technology.
→ Clock rate 40 MHz → 2.0 GHz.
→ Cycle per instructions have increased.
→ Processors capable to run multiple instructions in same cycle.

↳ Concurrency is one solution but results in problems like:
- Multiplicity of data paths.
- Increased access to storage elements.
- Scalable performance.
- Lower costs.

- **Moore's Law:**

→ Proposed by Gorden E. Moore in 1965 and revised in 1975. It states:
"Processing speed, or overall processing power for computers will double every 18 months."
(OR more technically)
"The number of transistors on an affordable CPU would double every two years."

Q: Will Moore's law hold forever?
- Adding multiple cores on single chip causes heat issues.

• Moreover, transistors would eventually reach the limits of miniaturization at atomic levels.

→ "So we must look for efficient parallel software solutions to fulfill our future computational needs."

• Parallel Platforms:

→ Parallel platforms provide higher aggregate caches.
→ Parallel platforms also provide increased bandwidth to the memory system.
→ Parallel platforms also have ability to pump data to memory and disk faster.

"Distributed Systems"

→ A collection of autonomous computers, connected through a network and distribution middle ware. They also share the resources of the system.

"Distributed Computing"

→ A specific use of distributed systems, to split a large and complex processing into subparts and execute them in parallel, to increase the productivity.

"Parallel Computing"

→ The term is used for developing concurrent solutions for following types of the system:
① Multi-core architecture.
② GPU's.

"Distributed Computing"

→ This type is mainly concerned with developing algorithms for the distributed cluster systems. Here distributed means a geographical distance b/w computers with any shared-memory.

→ **Practical Applications of PDC:**

- Applications in astrophysics.
- Advances in computational physics & chemistry.
- Weather modeling.
- Flood prediction.
- Data-mining analysis.
- Large scale servers.
- Network intrusion detection.
- Cryptography.
- Graphics processing.

→ **Limitations of parallel computing:**

- Exploring the proper parallelism from a problem is a hectic process.
- The program must have low coupling and high cohesion. It's a difficult task.
- It needs relatively more technical skills to code a parallel program.

## "Chap # 02"

- **Pipelining & Superscalar executions:**

→ with respect to processors, breaking problems into smaller sub-problems enable faster clock rates, as the problem is now smaller.

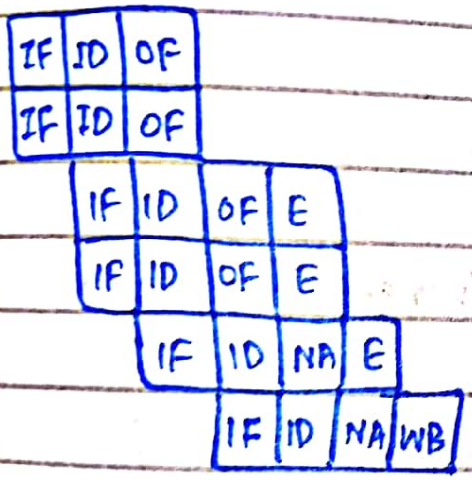
- **Super-pipelined processors:** A processor with two pipelines and the ability to issue two instructions simultaneously.
- **Superscalar execution:** The ability of a processor to issue multiple instructions in the same cycle.

→ Example:

(i)

```
load  R1, @1000
load  R2, @1008
add   R1, @1004
add   R2, @100C
add   R1, R2
store R1, @2000
```

→ Execution schedule:




(ii)

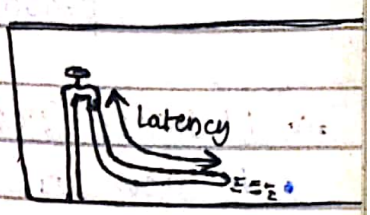
```
load  R1, @1000
add   R1, @1004
add   R1, @1008
add   R1, @100C
store R1, @2000
```

(iii)

```
load  R1, @1000
add   R1, @1004
load  R2, @1008
add   R2, @100C
add   R1, R2
store: R1, @2000
```

6/9/2022 "PDC"

→ Limitation of memory processors:
① Latency: Time between request and its status.
② Bandwidth:




③ Cache: fastest memory b/w Processor & main memory.
"low-latency, high-bandwidth."

→ Cache hit = search for spatial refrence of memory/data and if found. ✓
→ Cache miss = seach for spatial refrence of memory/data data if not found. ✗

• Impact of memory bandwidth:-
① contain more memory information/number.
② contagious memory / "pas pas"

• Alternate approaches for hiding memory latency:

① Pre-fetching. (request earlier/advance)
② Multi-threading. (open more tabs) "independent"

• Dichotomy of Parallel computing platforms.

| SIMD | MIMD |
|---|---|
| → Single Instruction multiple data. | → Multiple instruction multiple data. |
| → A lot of processor stalling. | |

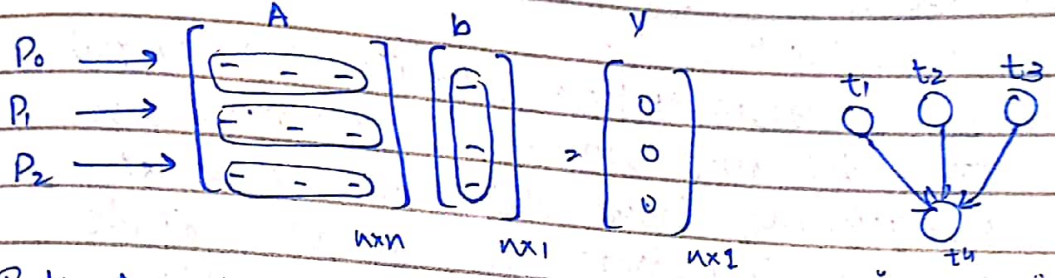
(Read "Sum")

15/09/2022 "PDC"

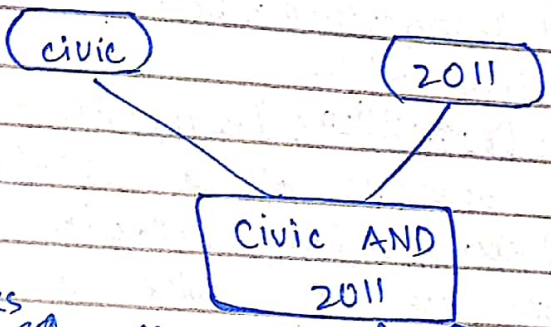
⇒ Designing Parallel algorithm:

3:1 ① Preliminaries:

↳ ① Dividing in smaller chunks / Decomposition
② Assigning them to different processors.




→ Task dependency graph:




"Granularity"

tasks
small = fine-grained
large tasks = coarse-grained

↳ Degree of concurrency: The maximum number of tasks that can be executed simultaneously.

↳ Critical Path: longest distance from starting node to ending node, and the sum of node cost is critical path length.

↳ average degree of concurrency:

$$\frac{\text{Total work}}{\text{C.P.L}} = A\left[\frac{63}{27}\right], B\left[\frac{64}{34}\right]$$

$= 2.4\ldots \quad = 1.9\ldots$

more , less

↳ task-Interaction graph: The pattern of interaction among tasks. share same equal.

Q.2

⇒ Decomposition Techniques:

① Recursive - Decomposition:- Divide & conquer
e.g: Quick sort -

② Data - Decomposition:- Particioning output data on different seperate procesors. also Particioning input data on different seperate processors. We can also divide input and output data simultaneously.

③ Exploratory - Decomposition:- Search space into smaller parts. e.g: Puzzle solving.

④ Speculative - Decomposition:- Decomposition on the basis of branches of a problem. (if-then-else)

⑤ Hybrid - Decomposition:- Collection of different decomposition used to solve a problem.
e.g: Quick sort.