Date: _____ Day: M T W T F S

Waleed Ishtiaq
21L-1856

PDC :-

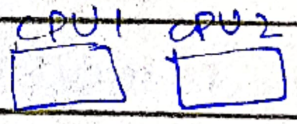
Uniprocessor:

- Concurrent processing → more than one jobs can be done, at a time, for some time part of job a is done after some time job b will be done and repeat
- Sequential processing → one ~~each~~ job done at a time, second job will start happening on completion of first job.

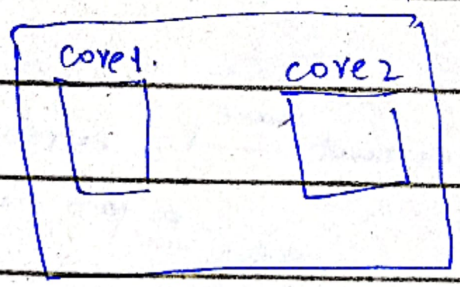
1) Parallel Computing:

Multiprocessor

- parallel processing → more than one job can be done simultaneously



Multicore → one cpu more cores.



→ Multicore more battery efficient.
better for inter process communication

Day: M T W T

Date:

→ In uni-processor, only one process is available and it single handedly deals with problem.

. In multiprocessor problem is divided in parts and seperate processors deals with separate part of problems

2) Distributed computing:

- Involves use of internet PC's available at diff. locations but performing one task.

→ We incur "latency" in distributed computing.

~~contain~~ have → integrated circuit contain have → transistors

(cores) perform computation

(performance)

Speed of processor → frequency based.

→ can do parallel computing in distributed computing not vice versa

Shared memory → diff cpu's access same memory

Each cpu can also have dedicated memory.

Moore's law → performance double after 18-24 months but failed now a days

Amdahl's law :

$$\text{Speed up } (p) = \frac{1}{F + \frac{1-F}{p}}$$

p → no. of processes

F → serial executions

$\frac{1-F}{p}$ → parallel executions

- $\text{Speedup } (p) = \frac{T(S)}{T(P)}$

  T(S) → For single processor

  T(P) → for multi processor.

- progress computation: Serial part + parallel part

$$\text{Speedup } (p) = \frac{1}{0.5} = 2 \rightarrow \text{speedup by } 2x.$$

Single processor :-

$$T(S) = F \cdot T(S) + (1-F)\, T(S)$$

F → sequential / serial executions

Date: _____ Day: M T W T

multiprocessor:

$$T(p) = F \cdot T(S) + \frac{(1-F)T(S)}{p}$$

$p \rightarrow$ no. of processors

* No advantage of multiprocessor in serial code.

→ $$\text{Speedup}(p) = \frac{T(S)}{T(p)}$$

$$= \frac{F \cdot T(S) + (1-F)T(S)}{F \cdot T(S) + \left(\frac{1-F}{p}\right)T(S)}$$

$$= \frac{\cancel{FT(S)} + 1T(S) - \cancel{FT(S)}}{T(S)\left[F + \frac{1-F}{p}\right]}$$

$$\text{Speedup}(p) = \frac{1}{\left[F + \frac{1-F}{p}\right]}$$

→ More sequential part of code less the speed.

Speedup depends on "F" entirely not on number of processors. Bottleneck in this is "F"

Date: _____ Day: M T W T F S

Example :

$1-F = .25$, $F = .75$

a) $P = 5$, $= \dfrac{1}{.75 + \dfrac{.25}{5}} = 1.25$

b) $P = \infty$, $\dfrac{1}{.75} = 1.33$ = speedup (∞).

To gain max speed, how many processors needed $P = ?$

$1.33 = \dfrac{1}{.75 + \dfrac{.25}{P}} =$

$1.33\,(.75P + .25) = 1$

$\boxed{P = 133}$

Karp-Flatt Metric

→ calculate serial fraction for parallel configuration.

speedup

$$e = \frac{1/S - 1/P}{1 - 1/P}$$ → no. of processors

Date: Day:

Example:

$P = 5$, $S = 1.25$

$\boxed{e = 0.75 \text{ or } 75\%}$

Parallel overhead → when a problem is divide to run on diff. processors, we will also need to sync results, which will cause overhead

→ Allocation of problem, getting results back add to overhead

→ e increases with time, since our code is usually mixed with sequential and parallel parts. So, parallel overhead adds to "e" with increase in "P" (depends on P also)

Types
- Data parallelism, • functional parallelism, • pipeling

~~Flynn's Taxonomy~~

~~multiprocessor~~

i) Centralized multi-processor (Shared memory)

- ↳ shared memory is same
- no. of CPU's are limited for it to work better (UMA)

Date: Day: M T W T F S

[Cache-coherence problem :-]

- inconsistency in diff. cache values for diff CPU's.

ii) Distributed-multiprocessor: (Shared memory)

- collection of memories forms one logical address space. (NUMA)
- Non-uniform memory access: → access time different because of physical location of memory.
- can accomodate greater no. of CPU's comparatively.

→ **Multi-Computer:**

- Distributed-memory (no cache coherence)

Assymetric multi computers

→ Front end computer others act like slaves

Symmetric multi computers

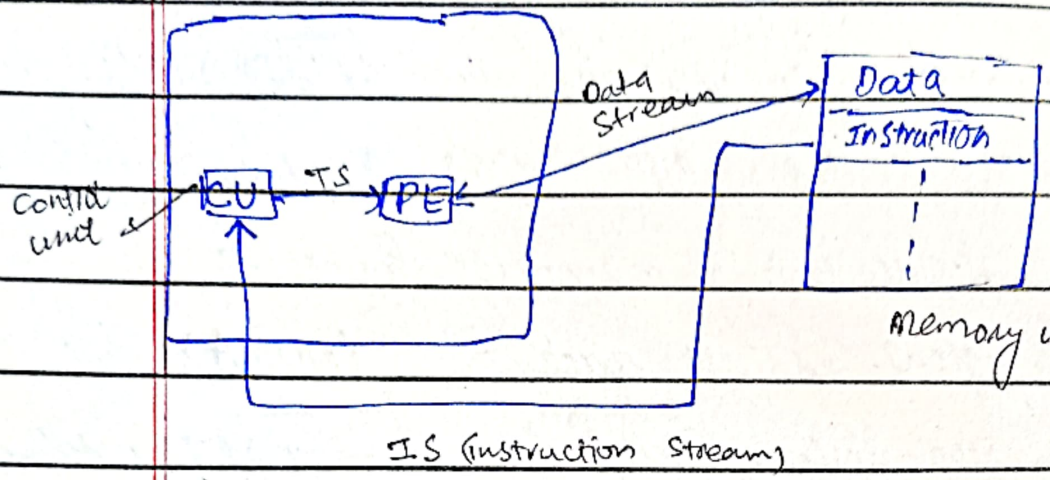
→ All are equal in terms of executions

Flyn's Taxonomy:

①	Single instruction	single data	uni-processor
②	" "	Multiple "	parallel
③	Multiple "	single "	parallel
④	" "	multiple data.	parallel

Date: _____ Day: M T W T

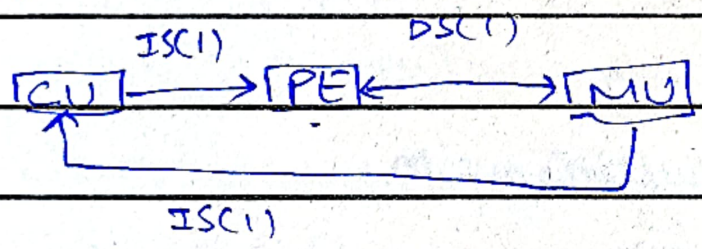
PE → processing element or Arthimetic logic unit



(SISD)

$z = a^2$ one instruc

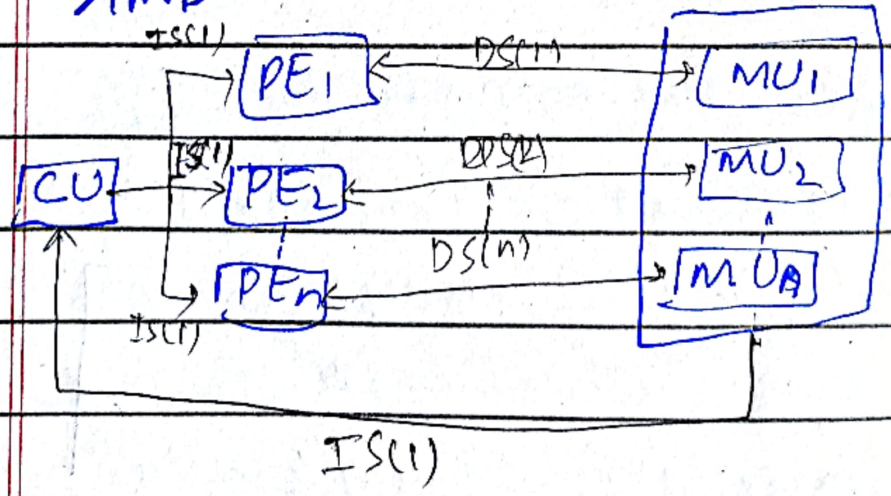
$2^2 = 4$ one data



1) CU → fetches instruction from MU
2) CU → gives it to PE.
3) PE fetches data (value) from MU. performs fun, and stores in MU.

SIMD

diff portion of RAM



Date: Day: M T W T F S

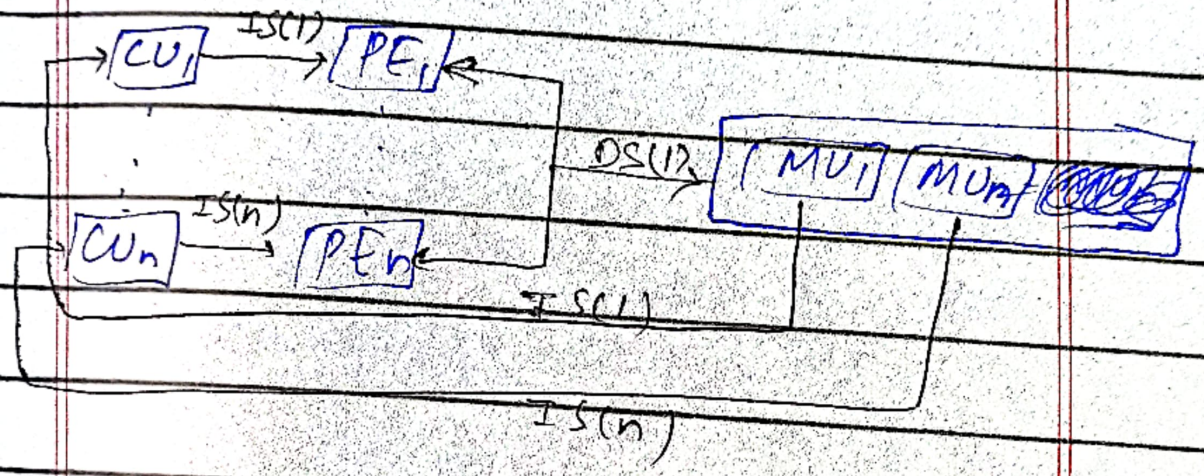
$z = a+b$, → Single instruction.

$2+3$
$3+1$) → mutiple data.

Great for matrix multiplication

MISD :

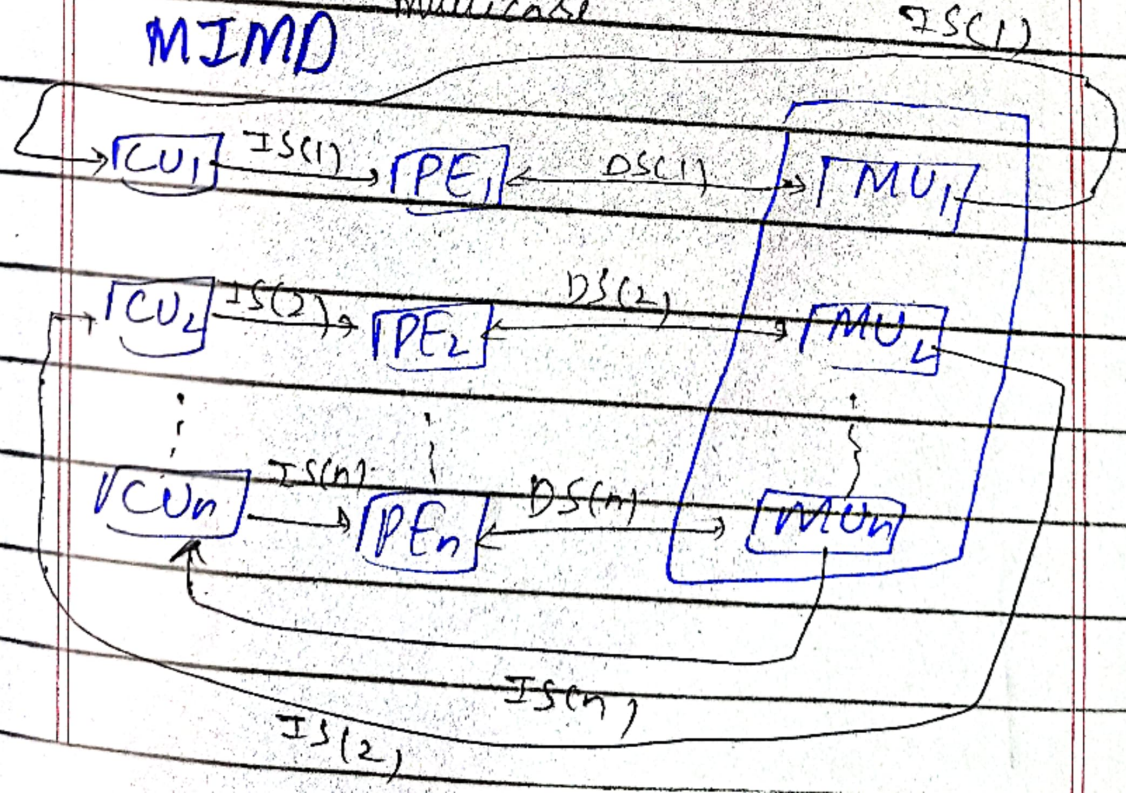
$\sin(x) + \cos(x)$



(Rarely used)

MIMD

multicore



Date: _____ Day: M T W

Example:-

↳ $\sin(x) + \cos(y) + \tan(z)$

~~ACC 05~~

Communication costs.

Date: 07/02/24. Day: M T ~~W~~ T F S

LEC 06

Network Topologies

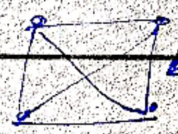
Dynamic Network Type :

1) Central data bus → cheap, less efficient
2) crossbar → expensive, most effic → dedication switches for each connection
3) Ω - network → mediocre, medium performance.

Static Network :

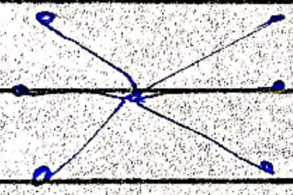
no switches in static network, only processors

1) Mesh Topology



costly

2) Star Topology

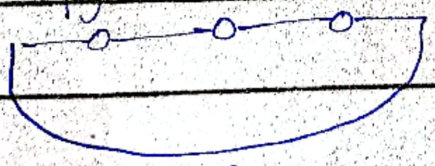


one main network, if main network goes down, everything shuts down.

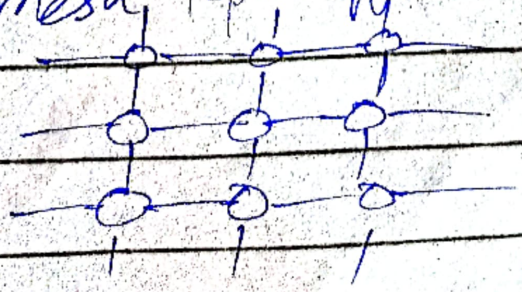
3) Linear Array

- each node 2 neighbours except first, last

4) Ring Topology $\boxed{\text{1-D Torus}}$



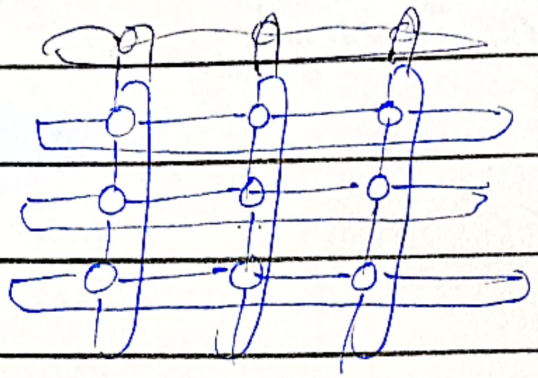
5) 2D-mesh topology without wrap around



$\boxed{\text{2D-Torus}}$

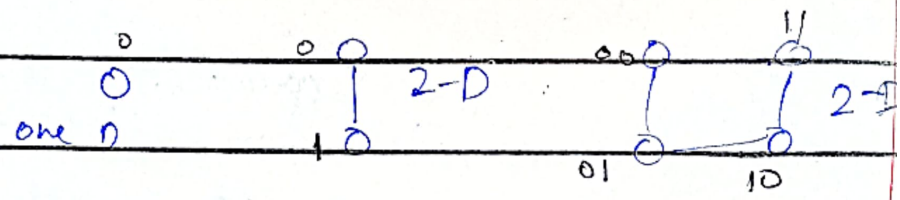
Date: _____ Day: M T W

each node 4 neighbours

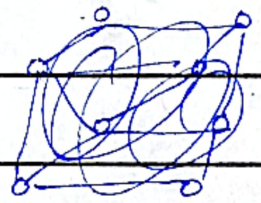


with wraparound

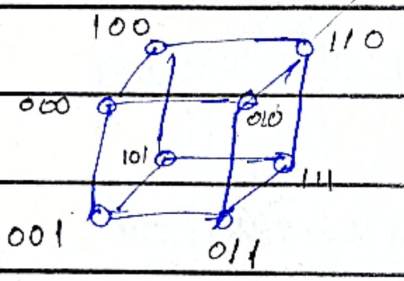
6 Dimensional Hyper cubes



join 2 one D to form 2D:
" 2 2D " " 3D.

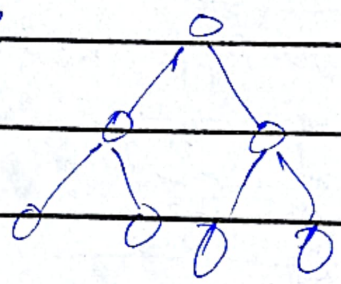


3-D

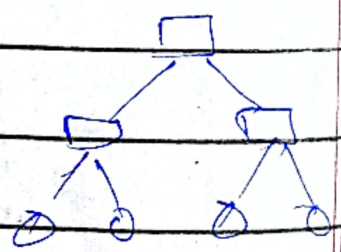


"each node binary difference with each its neighbour should be one"

7- Tree



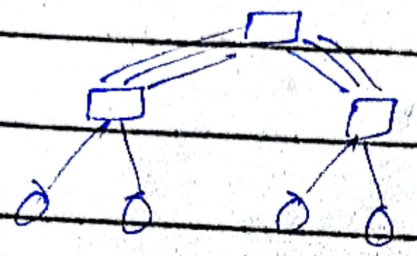
Static Tree



dynamic Tree with switches

Date: _____ Day: M T W T F S

To avoid congestion, increase route node links



To increase efficiency.

(FAT Tree)

(Evaluating static connections)

cost → no of links. → should be less

Diameter → distance b/w farthest nodes.

Bisection width → tells about bandwidth → should be higher

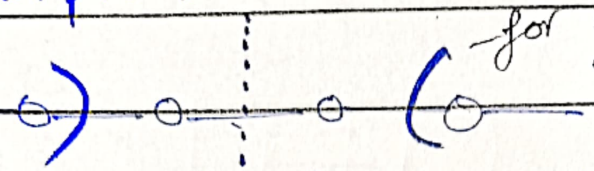
ARC-connectivity → tells how strong your network is.

- ARC-connectivity → to disconnect one node
- Bisection → divide in two equal networks

→ We find node with min. arc connectivity and that is arc-connectivity of network

Date: _____ Day: M T W T

Example :

i) 

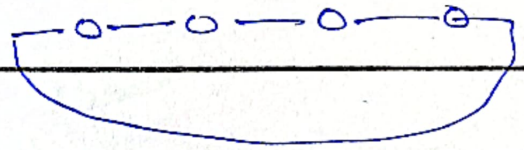
no. of processors = 4.

1) cost → P−1

2) Diameter → P−1

3) Bisection - width → 1

4) Arc-connectivity → 1

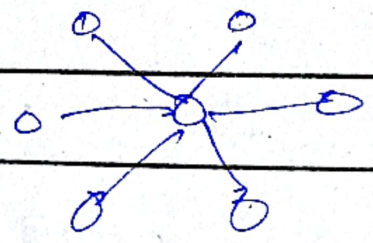
ii) 

cost → P.

Diameter → $\lceil P/2 \rceil$

Bisection → 2 → same for any no. of nodes

ARC-conn → 2

iii) 

Cost → P−1

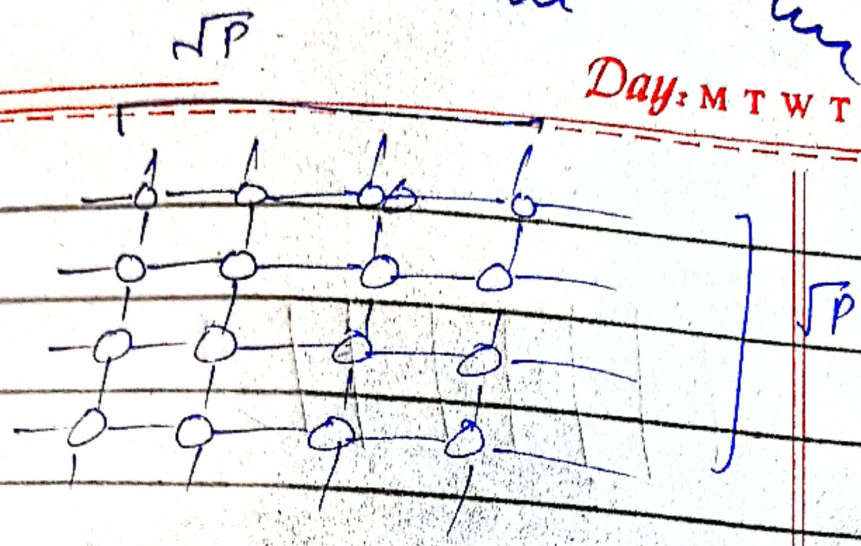
Diameter → 2

Bisection → 1 → we get almost equal

ARC → 1

Date: Day: M T W T F S

iv)



Total processor $= \sqrt{P}.\sqrt{P}$
$= P$

No. of links in each row $= \sqrt{P} - 1$

" " " col $= \sqrt{P} - 1$

Total cost $= \underbrace{\sqrt{P}}_{\text{Total no of processor}} \underbrace{(\sqrt{P}-1)}_{\text{Total links for each row}} + \sqrt{P}\underbrace{(\sqrt{P}-1)}_{\text{for each col.}}$

→ Cost $= 2(P - \sqrt{P})$

→ Diameter $= 2\sqrt{P} - 2$.

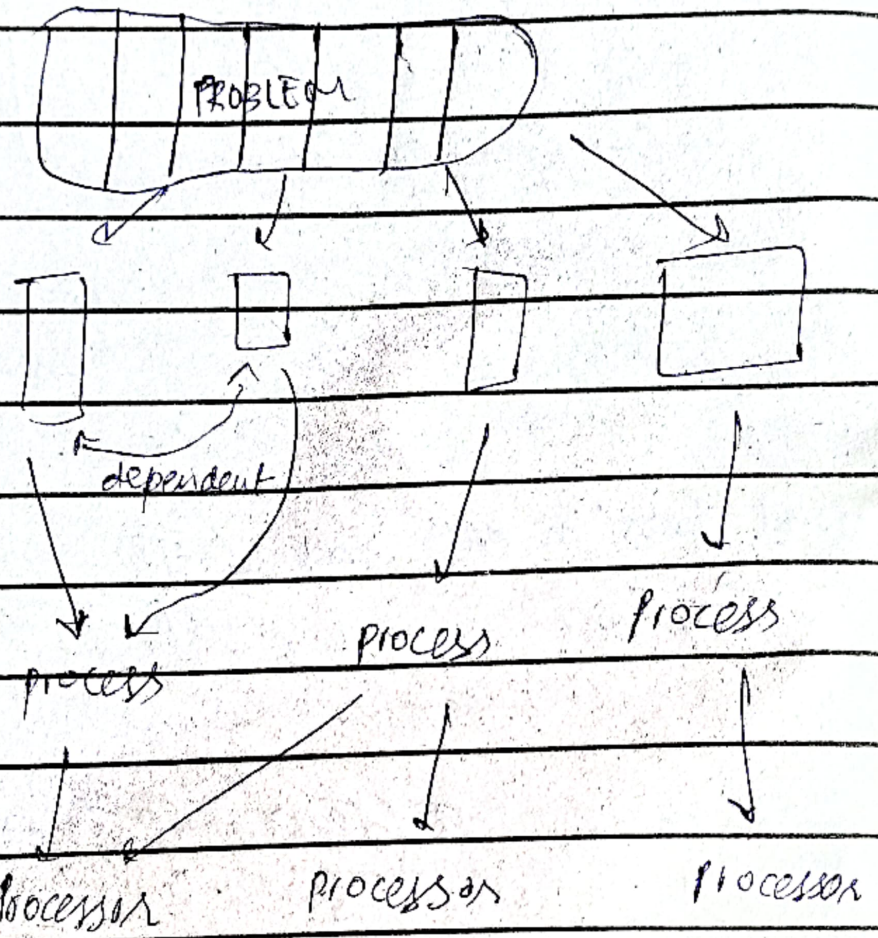
$(\sqrt{P}-1) + (\sqrt{P}-1)$

Bisection: $\sqrt{P}$

ARC : 2

Date: 14/02/24. Day: M T W T F

LEC 07



In hardware concurrency = parallelism