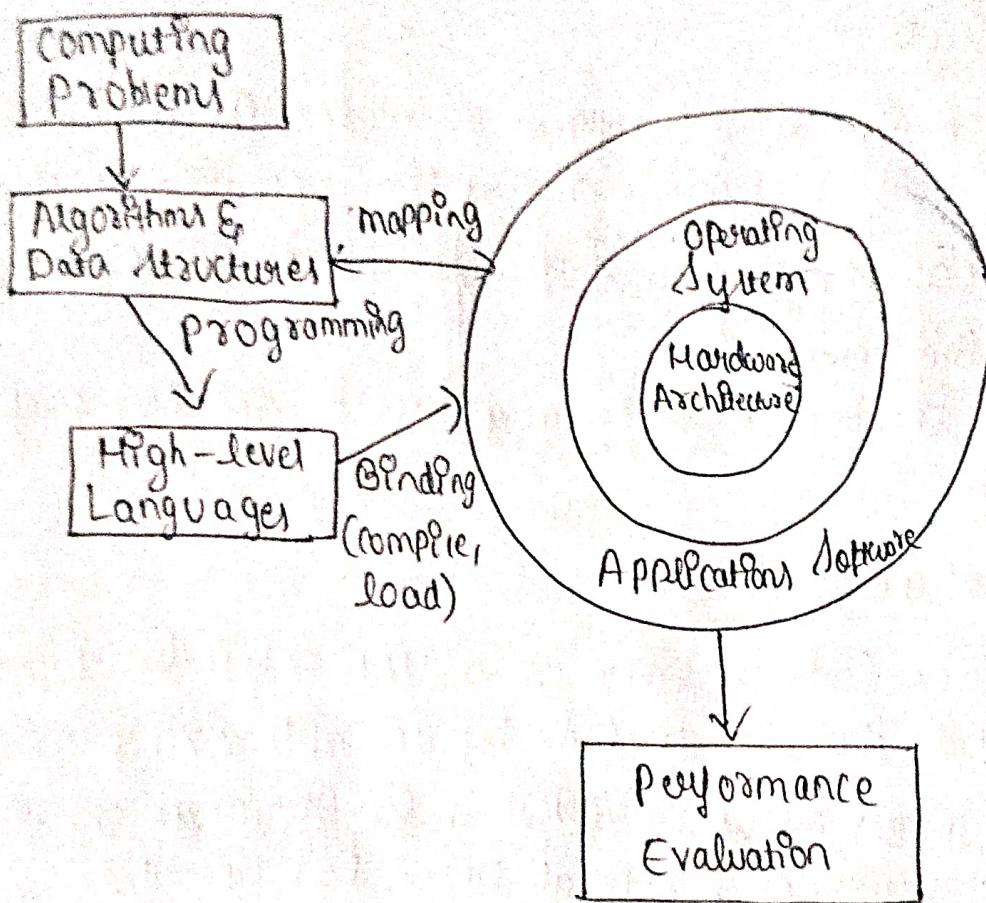


## Assignment - 1

i) Define Computer Architecture. With a neat diagram explain the elements of modern computer system.

Computer Architecture refers to the design and organization of Computer System and its hardware component, Instruction Set, memory system and Communication Interface.



### Computing problems

1. For numerical problems in Science & Technology, the solution demand Complex mathematical formulation.
- \* For alpha numerical problems in business & government, the solution demand accurate & transaction.
- \* For artificial intelligence problems, the solution demand logic inference & symbolic manipulation.

## Hardware Resources

- \* A modern Computer System demonstrates its power through coordinated effort by hardware resources, an OS & application software.
- \* Processor, memory & peripheral devices form the hardware core of a computer system.
- \* Special hardware interfaces are often built into I/O devices.

## Operating System

- \* An effective operating system manages the allocation & deallocation of resources during the execution of user programs.
- \* Beyond the OS, application software must be developed to benefit the user.
- \* Standard benchmark programs are needed for performance evaluation.

## System Software Support

- \* The source code written in a HLL must be first translated into object code by an optimizing compiler.
- \* The compiler assigns variables to registers or to memory words & reserves functional units for operators.

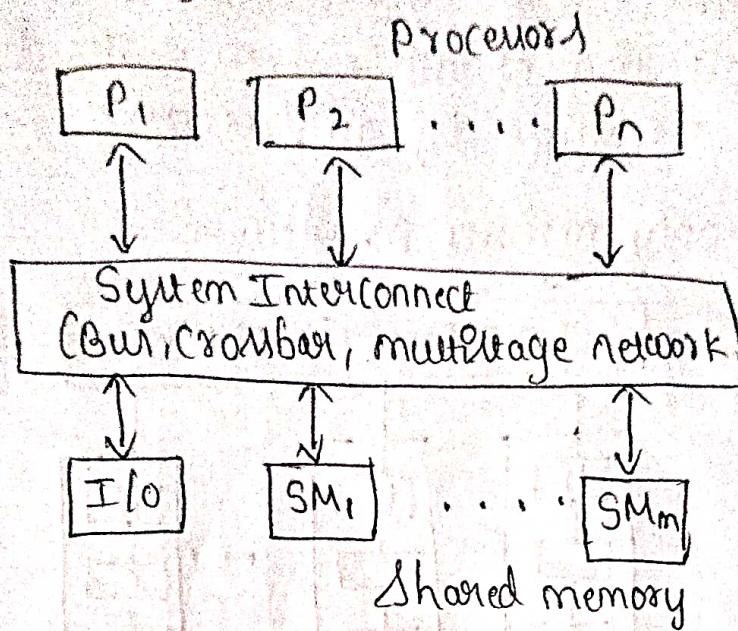
## Compiler Support

There are 3 Compiler Upgrade approaches:

- Preprocessor
- Precompiler
- Parallelizing Compiler.

2) Describe with a neat diagram different shared memory multiprocessor model.

⇒ Q) Uniform memory-Access (UMA) model

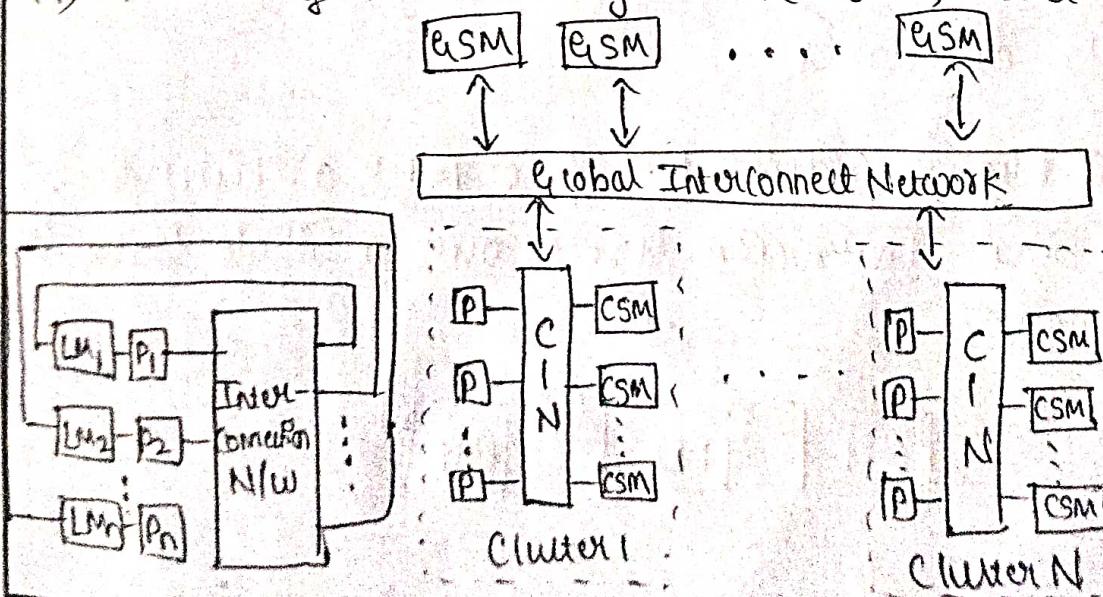


\* In a UMA multiprocessor model, the physical memory is uniformly shared by all the processors.

\* All processors have equal access time to all memory words, which is why it is called Uniform memory access.

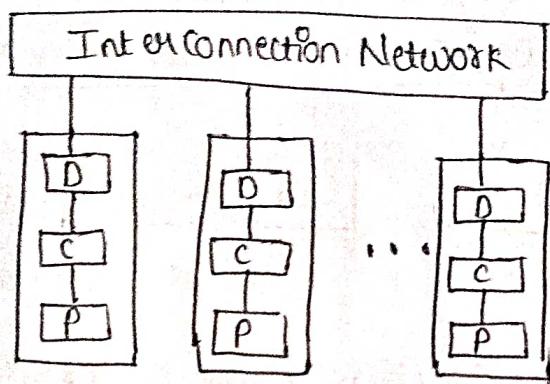
\* Each processor may use a private cache.

i) Non-Uniform-memory-Access (NUMA) model.



- \* A NUMA multiprocessor is a shared-memory system in which the access time varies with the location of the memory word.
- \* The shared memory is physically distributed to all processors, called local memory.
- \* The collection of all local memories forms a global address space accessible by all processor.

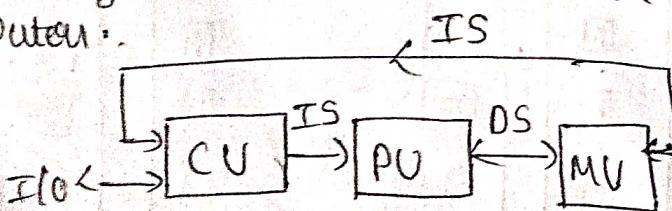
### (iii) Cache-only memory Architecture



- \* A multiprocessor using Cache-only memory follows the CONA model.
- \* There is no memory hierarchy at each processor node.
- \* All the caches form a global address space.
- \* Remote Cache access is handled by the distributed Cache directories.

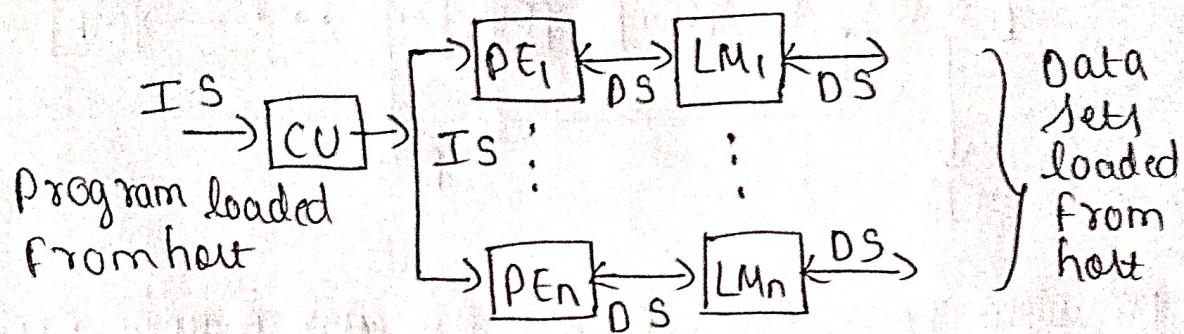
Q) Explain G Flynn's Classification of Computer architecture.

i) SISD (Single Instruction Stream over a Single Data Stream) Computer.



- \* Conventional Sequential machines are called SISD Computer.
- \* They are also called Scalar processor.
- \* Single Instruction: only one instruction stream is being acted on by the CPU during any one clock cycle.
- \* Single data: only one data stream is being used as input during any one clock cycle.
- \* Deterministic execution.

### ii) SIMD (Single Instruction Stream over Multiple Data Streams) machine

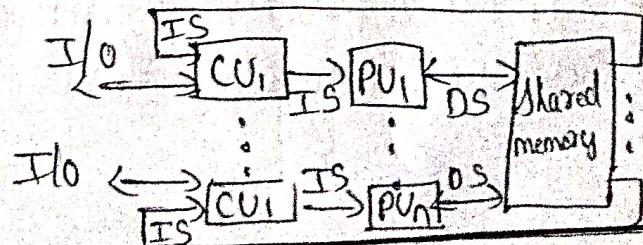


- \* Single Instruction: All processing units execute the same instruction issued by the control unit at any give clock cycle.

- \* Multiple data: Each processing unit can operate on a different data element.

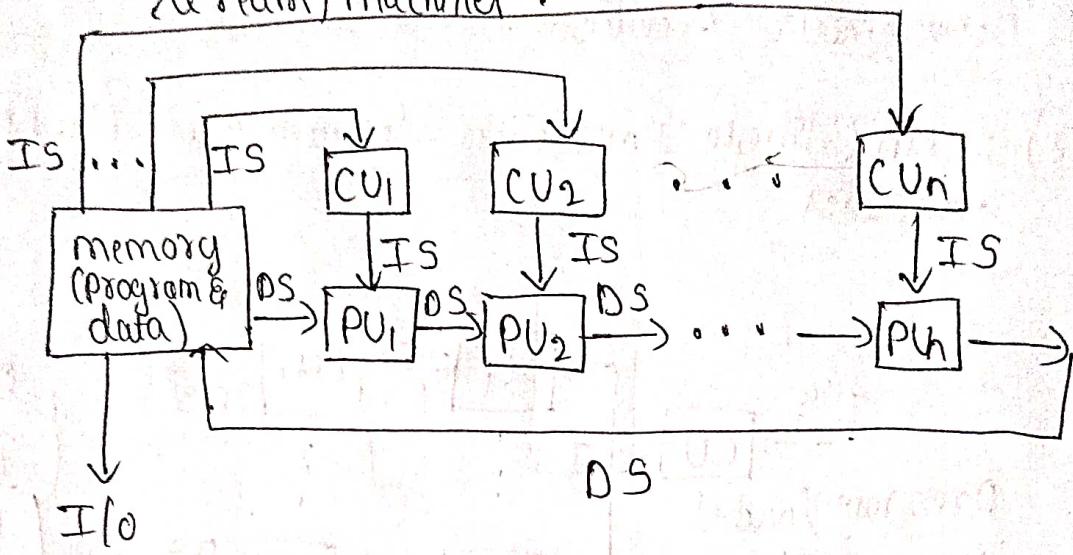
- \* Synchronous & deterministic execution.

### iii) MIMD (Multiple Instruction Stream over Multiple Data Streams) machine



- \* A single data stream is fed into multiple processing units.
- \* Each processing unit operates on the data independently via independent instruction streams.
- \* This architecture is also known as Syntactic Arrays.

iv) MISO (multiple Instruction Streams & a single Data Stream) machine.



\* multiple Instructions: Every processor may be executing a different instruction stream.

\* multiple Data: Every processor may be working with a different data stream, multiple data stream is provided by shared memory.

\* Execution can be Synchronous, asynchronous, deterministic or non-deterministic.

5) A 4 MHz processor was used to execute a benchmark program with the following instruction mix & clock cycle count.

Instruction Type	Instruction Count	Cycles/Instruction
Integer arithmetic	45000	1
Data transfer	32000	2
Floating Point	15000	2
Control transfer	8000	2

Determine the effective CPI, MIPS rate & execution time for this program.

$$\Rightarrow \text{number of Clock Cycles} = \sum I_i c_i \times \text{CPI}_i$$

$$= (45000 \times 1) + (32000 \times 2) + (15000 \times 2) + (8000 \times 2)$$

$$= 155000 \text{ CC}$$

$$\text{CPI} = \frac{\text{CC}}{\text{IC}} = \frac{155000}{100000} = 1.55$$

$$\text{MIPS} = \frac{f}{\text{CPI} \times 10^6} = \frac{4 \times 10^6}{1.55 \times 10^6} = 2.58$$

$$T = \frac{\text{IC} \times \text{CPI}}{f} = \frac{155000}{4 \times 10^6} = 0.03875 \text{ s}$$

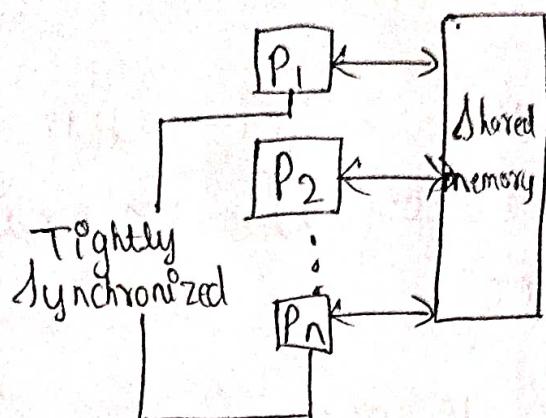
$$= 38.75 \text{ ms}$$

6) Briefly explain the PRAM & VLSI models with necessary diagram.

### PRAM model

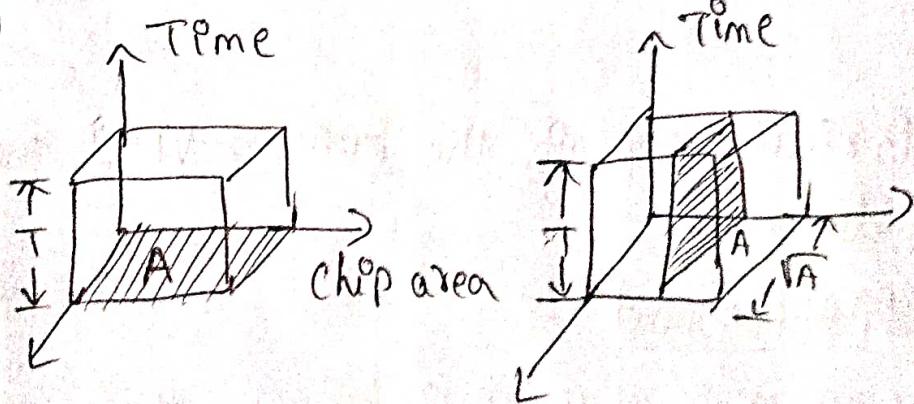
\* PRAM is a theoretical model of Parallel Computation.

- \* Processors may execute different instruction streams, but work synchronously.
- \* The machine size  $n$  can be arbitrarily large.
- \* The machine size  $n$  can be synchronous at the instruction level.
- \* Load Imbalance is the only form of overhead in the PRAM model.



- \* Four memory-update option are possible
  - Exclusive Read(ER)
  - Exclusive Write(EW)
  - Concurrent Read(CR)
  - Concurrent write(CW)

### VLSI models



- \* Parallel computers rely on the use of VLSI chips to fabricate the major components such as processor arrays, memory arrays & large scale switching networks.
- \* The rapid advent of very large scale integrated (VLSI) technology now computer architect.

With respect to dependencies between the instructions, discuss the following with example:

i) Data Dependency    ii) Control dependency    iii) Resource dependency

⇒ 5 type of data dependence are defined below:

i) Flow Dependence

\* A statement  $S_2$  is flow dependent on statement  $S_1$  if an execution path exists from  $S_1$  to  $S_2$  & if at least one output of  $S_1$  feeds in as input to  $S_2$ .

\* It is denoted as:

$$S_1 \rightarrow S_2$$

\* Example:

$S_1$ : Load R1, A

$S_2$ : ADD R2, R1

2) Anti Dependence

\* Statement  $S_2$  is anti dependent on the statement  $S_1$  if  $S_2$  follows  $S_1$  in the program order & if the output of  $S_2$  overwrites the input to  $S_1$ .

\* It is denoted as:

$$S_1 \xrightarrow{+} S_2$$

\* Example:

S1 : Add R<sub>2</sub>, R<sub>1</sub>

S2 : Move R<sub>1</sub>, R<sub>3</sub>

### 3) Output Dependence

\* Two statements are output dependent if they produce the same output variable.

\* It is denoted as:

S1 → S2

\* Example:

S1 : Load R<sub>1</sub>, A

S2 : Move R<sub>1</sub>, R<sub>3</sub>

### 4) I/O Dependence

\* Read & write are I/O statements. I/O dependence occurs not because the same variable is involved but because the same file is referenced by both I/O statements.

\* Denote as:

S1 →<sub>I/O</sub> S3

\* Example:

S1 : Read(A1), A(I)

S3 : Write(A1), A(I)

### 5) Unknown Dependence

\* When dependence relation between two statements cannot be determined is known as unknown dependency.

### ii) Control dependency

- \* This refers to the situation where the order of the execution of statements cannot be determined before run time.
- \* For e.g., all Condition Statement, where the flow of statement depends on the output.

### Control-dependent eg:

For ( $i=1; i < n; i++$ )

$\left\{ \begin{array}{l} \text{if } (a[i-1] < 0) \text{ } a[i] = 1; \\ \end{array} \right.$

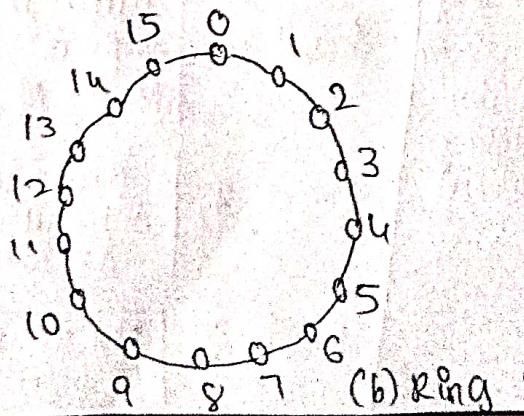
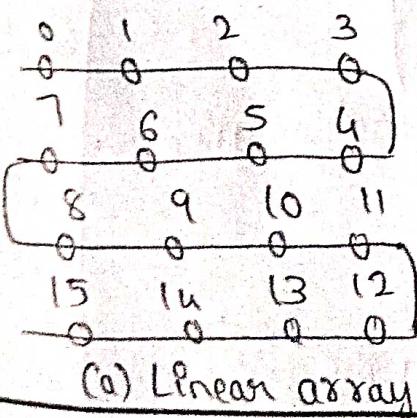
### iii) Resource dependence

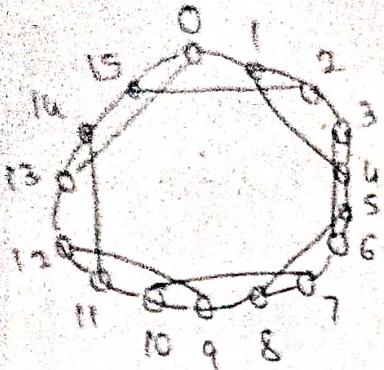
\* Data & Control dependencies are based on the independence of the work to be done.

\* ALU conflicts are called ALU dependence.

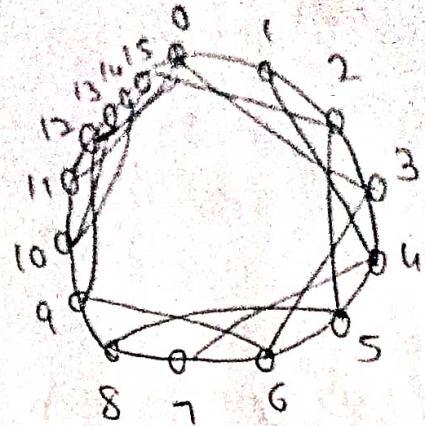
\* Memory conflicts are called storage dependence.

8) List the different types of static connection network & explain any three in detail.





(c) Chordal Ring of degree 3

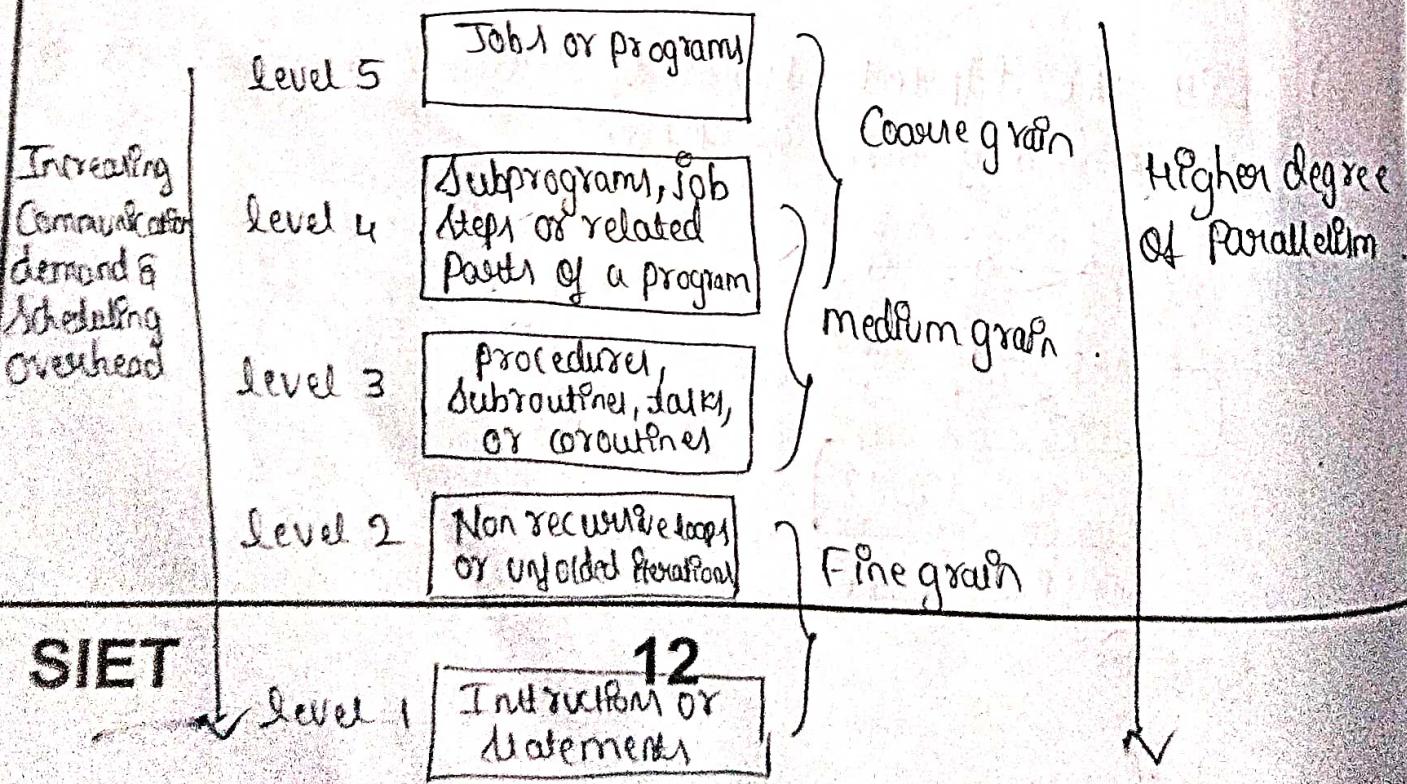


(d) Chordal Ring of degree 4.

- \* Static networks use direct links which are fixed once built.
- \* This type of network is more suitable for building computers where the communication pattern are predictable or implementable with static connection.

\* we describe their topology below in terms of network parameters & comment on their relative merits in relation to communication & Scalability.

a) Explain the levels of parallelism in program execution on modern computers.



## Instruction level parallelism

- \* This fine-grained, typically involves less than 20 instructions per grain.
- \* The number of candidates for parallel execution varies from 2 to thousands.

Advantages: There are many many candidates for parallel execution.

## Loop-level parallelism

- \* Typically loop has less than 500 instructions.
- \* most optimized program construct to execute on a parallel or vector machine.

## Procedure-level parallelism

- \* medium-sized grain: usually less than 2000 instructions.
- \* Communication requirement less than instruction level.

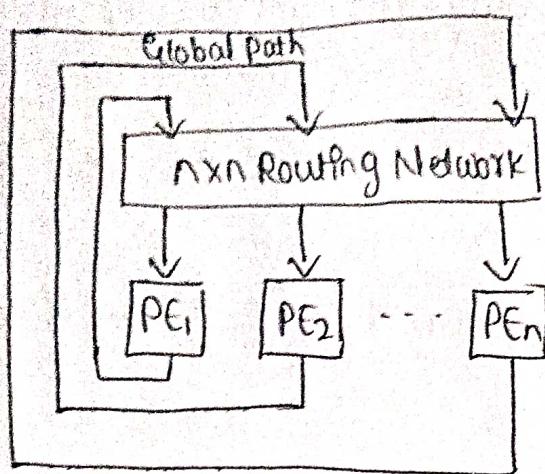
## Subprogram-level parallelism

- \* Job step level: grain typically has thousands of instructions; medium- or coarse-grain level.
- \* Job steps can overlap across different jobs.

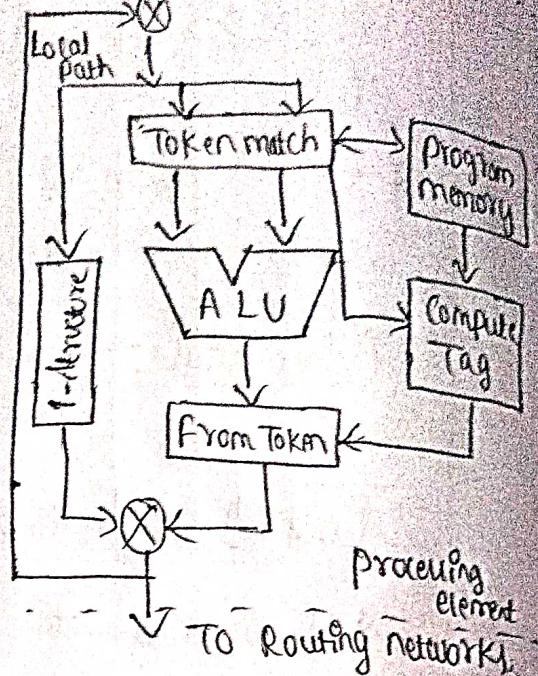
## Job or program-Level parallelism

- \* Corresponds to execution of essentially independent jobs.
- \* This is practical for a machine with a small no. of powerful processors.

(b) Explain briefly program flow mechanism.



(a) The global architecture



(b) Interior design of a processing element.

\* Control flow mechanism: Conventional machines used Control Flow mechanism in which order of program execution explicitly stated in user programs.

\* Data flow machines which instruction can be executed by determining operand availability.

\* Reduction machines trigger an instruction's execution based on the demand for its results.

Control flow machines used shared memory for instruction & data. Since variables are updated by many instructions, there may be side effects on other instruction. These side effects frequently prevent parallel processing. Single processor systems are inherently sequential.

## Data Flow Features

No need for

- shared memory
- program counter
- control sequences

Special mechanism are required to

- detect data availability
- match data token with instruction needing them
- enable chain reaction of asynchronous instruction execution

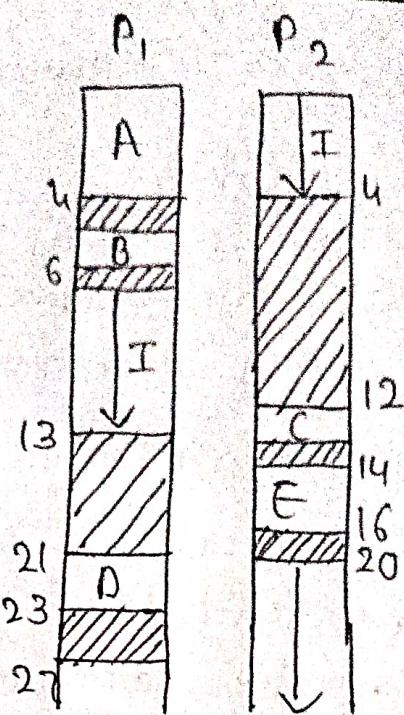
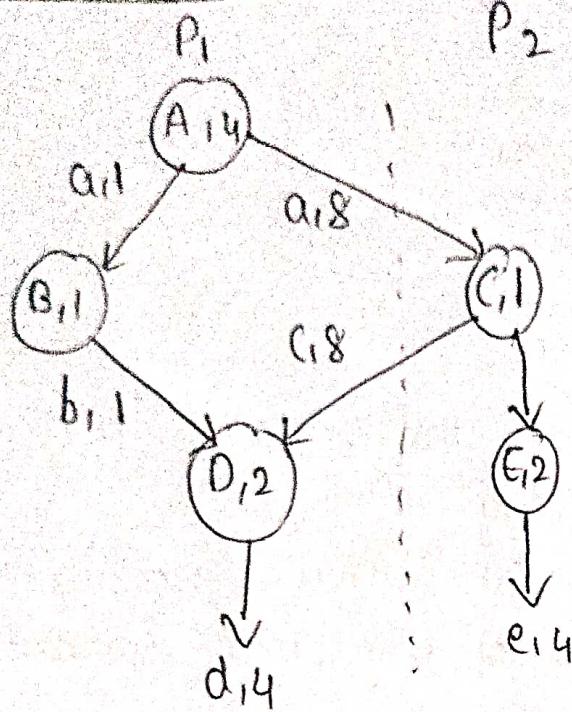
ii) what is node duplication? with an Example, Explain the node duplication scheduling to eliminate communication delays between processors.

\* ~~Grain Packing~~ may Pot

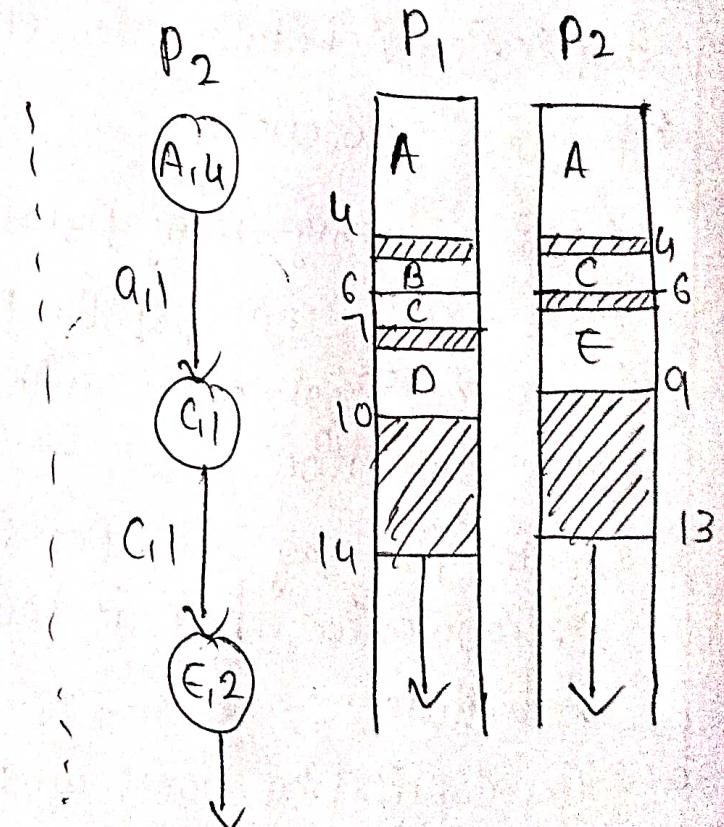
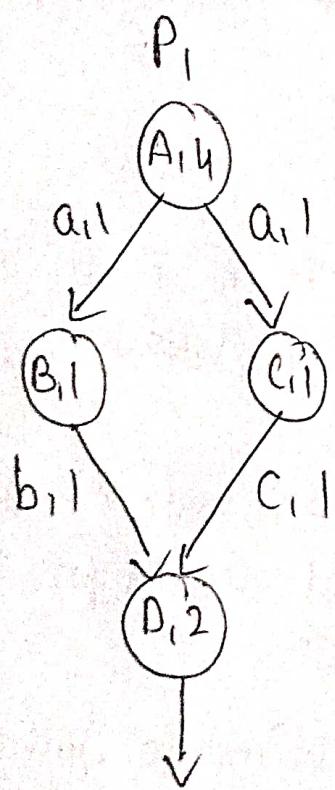
Copy & Paste many of the nodes in the Model Builder to create additional nodes with identical settings is called node duplication.

\* Grain Packing may potentially eliminate interprocessor communication, but it may not always produce a shorter schedule.

\* By duplicating nodes, we may eliminate some interprocessor communication, & thus produce a shorter schedule.



(a) Schedule without node duplication.



(b) Schedule with node duplication ( $A \rightarrow A$  &  $A'$ ),  
 $(C \rightarrow C$  &  $C')$

- \* Fig.a shows a schedule without duplicating any of the 5 nodes.
  - \* In Fig.b node A is duplicated. P1 to A's assigned to P2 besides retaining the original copy A in P1.
  - \* Similarly, a duplicated node C is copied onto P1 besides the original node C in P2.
  - \* The new schedule is shown in Fig.b is almost 50% shorter than that in Fig.a
- (2) write a note on Amdahl's law for a fixed workload.