

Assignment No. 1

PAGE NO.: / /
DATE: / /

Q1 What is Flynn's taxonomy? How Computer Architectures are classified.

→ 1 The most popular taxonomy of computer architecture was defined by Flynn in 1966. Flynn's classification scheme is based on the notion of a stream of information. Two types of information flow into a processor : instructions & data.

2 The instruction stream is defined as the sequence of instructions performed by the processing unit.

3 The data stream is defined as the data traffic exchanged between the memory & processing unit.

4 According to Flynn's classification, either of the instruction or data streams can be single or multiple.

5 Computer Architecture are classified into following distinct categories :

- single-instruction single-data stream (SISD);
- single-instruction multiple-data stream (SIMD);
- multiple-instruction single-data stream (MISD) &
- multiple-instruction multiple-data streams (MIMD).

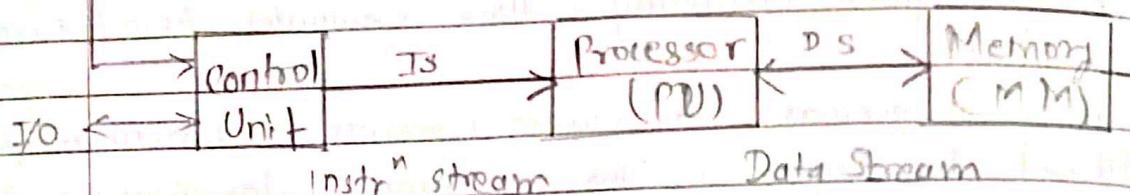
6 Conventional single-processor von Neumann computers are classified as SISD systems.

7 Parallel computers are either SIMD or MIMD.

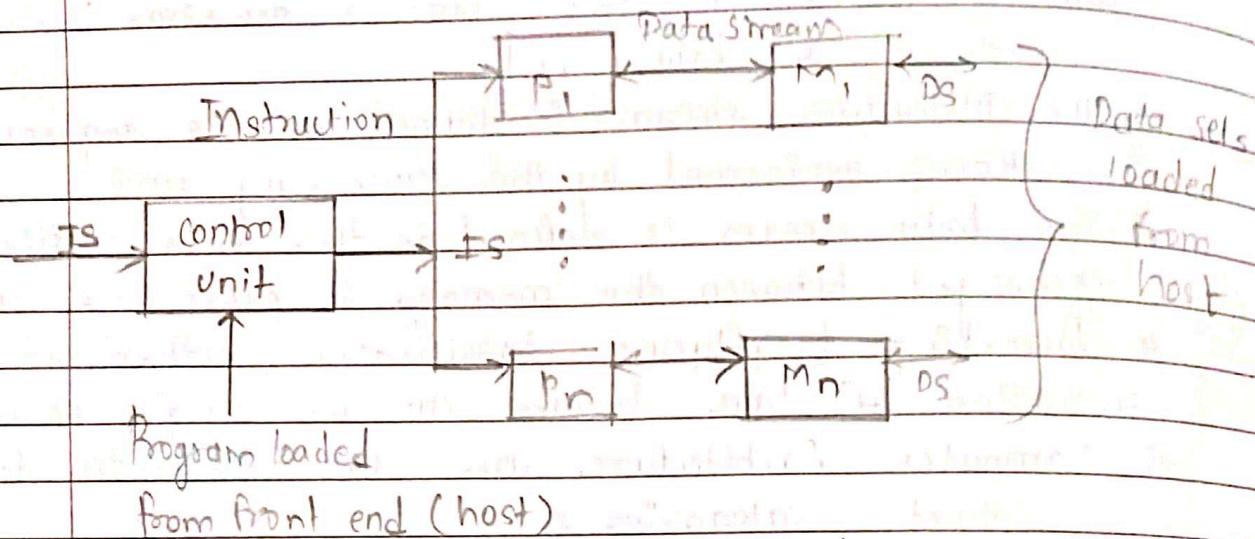
8 When there is only one control unit & all processors execute the same instruction in a synchronized fashion, the parallel machine is classified as SIMD.

9 In a MIMD machine, each processor has its own control unit and each can execute different instructions on different data.

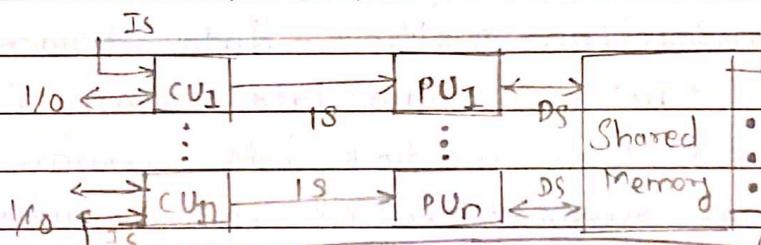
10 In the MISD category, the same stream of data flows through a linear array of processors executing different instruction streams.

Instruction Stream (IS)

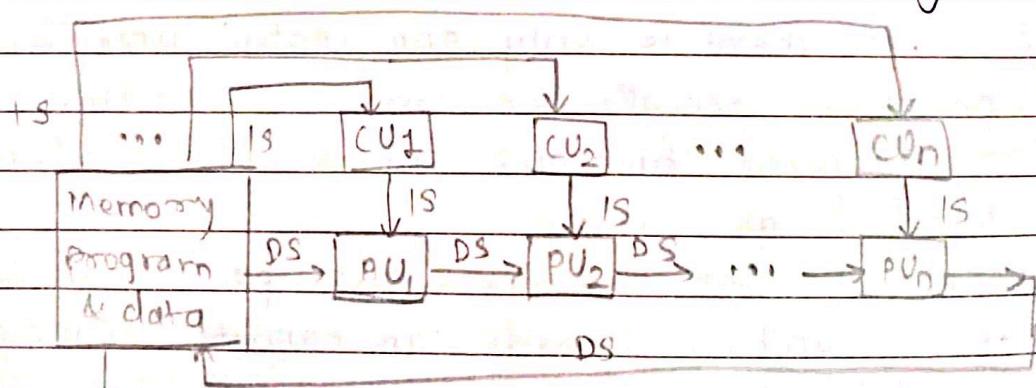
a) SISD Architecture or uniprocessor architecture



b) SIMD architecture (with distributed Memory)



c) MIMD architecture (shared memory)



d) MISD architecture (the systolic array)

Q2 Explain the concept of linear pipelining.

- 1 Pipelining offers an economical way to realize temporal parallelism in digital computers. The concept of pipeline processing in a computer is similar to assembly lines in an industrial plant.
- 2 Assembly lines have been widely used in automated industrial plants in order to increase productivity.
- 3 To achieve pipelining one must subdivide the input task (process) into a separate sequence of subtasks each of which can be executed by a specialized Hardware stage that operates concurrently with other stages in the pipeline.
- 4 Successive tasks are streamed into the pipe & get executed in an overlapped fashion at the subtask level.
- 5 The uniform-delay pipeline, all tasks have equal processing time in all station facilities. The stations in an ideal assembly line can operate synchronously with full resource utilization.
- 6 The precedence relation of a set of subtasks $\{T_1, T_2, \dots, T_k\}$ for a given task T implies that some task T_j cannot start until some earlier task T_i ($i < j$) finishes. These interdependencies form the precedence graph.
- 7 With a linear-precedence relation, task T_j cannot start until all earlier subtasks $\{T_i\}$, for all $i \leq j$ finish.
- 8 A Linear pipeline can process a succession of subtasks with a linear precedence graph.
- 9 A basic Linear-pipeline processor pipeline consists of a cascade of processing stages.
- 10 The stages are pure combinational circuits performing arithmetic or logic operations over the data stream flowing through the pipe.

- 11 The stages are separated by high-speed interface latches. The latches are fast registers for holding the intermediate results between the stages.
- 12 Information flows between adjacent stages are under the control of a common clock applied to all the latches simultaneously.

Q3 How pipeline architectures are classified?

- 1) According to the levels of processing, Handler (1977) has proposed the following classification scheme for pipeline architectures:
- Arithmetic pipelining
 - Instruction pipelining
 - Processor pipelining
- 2) Arithmetic pipelining : The arithmetic logic units of a computer can be segmentized for pipeline operations in various data formats. Well-known arithmetic pipeline examples are the 4-stage pipes used in star-100, 8-stage in T1-ASC, & upto 14-pipeline stages used in Cray-1, & 26 stages per pipe in Cyber-205.

3) Instruction pipelining : The execution of a stream of instructions can be pipelined by the overlapping the execution of current instruction with the fetch, decode, & operand fetch of subsequent instructions. This technique is also known as instruction lookahead. Almost all HPC are now equipped with "Instruction-execut" pipelines.

4) Processor pipeline : This refers to the pipeline processing of the same data stream by a cascade of processors, each of which processes a specific task. The data stream passes the first processor with results stored in a memory block which is also accessible by the second processor. The second processor then passes the refined results to the third & go on. The pipelining of multiple processors is not yet well accepted as a common practice.

Q4 What are the different performance measures in pipeline architectures?

→ 1) Clock period : The logic circuitry in each stage S_i has a time delay denoted by T_i . Let T_l be the time delay of each interface latch, the clock period of a linear pipeline is defined by

$$T = T_m + T_l \quad T_m = \max \{ T_i \}_{i=1}^k$$

The reciprocal of the clock period is called frequency $f = 1/T$ of a pipeline processor.

2) A linear pipeline with k -stages can process n tasks in $T_k = k + (n-1)$ clock periods, k cycles are used to fill up the pipeline or to complete execution of the first task & $n-1$ cycles are needed for remaining $n-1$ tasks.

3) The same task can be executed in a nonpipeline processor with an equivalent function in $T_1 - n \cdot k$ time delay.

4) Speedup : We define the speedup of a k -stage linear-pipeline processor over an equivalent nonpipeline processor as. $s_k = \frac{T_1}{T_k} = \frac{n \cdot k}{k + (n-1)}$

The maximum speedup is $S_k \rightarrow k$, for $n \geq k$.

5 The product of a time interval & a stage space in the space-time diagram is called a time-space span. A given time-space span can be in either a busy state or an idle state, but not both.

6 Efficiency : The efficiency of a linear pipeline is measured by the percentage of busy time-space spans over the total time-space span, which equals the sum of all busy & idle time-space spans.

7 Let n, k, T be the number of tasks (instructions), the number of pipeline stages, & the clock period of a linear pipeline, respectively.

The pipeline efficiency is defined by

$$\eta = \frac{n}{k + (n-1)T} \quad \text{Note that } \eta \rightarrow 1 \text{ as } n \rightarrow \infty$$

The larger the number of tasks flowing through the pipeline, the better is its efficiency.

8 Another formula $\eta = S_k/k$ provides another view of the efficiency of a linear pipeline as the ratio of its actual speedup to the ideal speedup k .

9 Throughput : The number of results (tasks) that can be completed by a pipeline per unit time is called its throughput. This rate reflects the computing power of a pipeline. In terms of efficiency η & clock period T of a linear pipeline, we define throughput as

$$W = \frac{n}{kT + (n-1)T} = \frac{n}{T}$$

Where, n = total no. of tasks being processed during time period = $Kt + (n-1)t$

Ideal case: $W = 1/t = f$ when $\eta \rightarrow 1$.

These means that maximum throughput of a linear pipeline is equal to its frequency, which corresponds to one output result per clock period.

Q5 Derive the equation for measurement of Speedup in pipeline architectures.

- 1 The system performance is measured in terms of processor utilization & the speedup over a serial computer.
- 2 The efficiency of a computer depends heavily on the inherent parallelism in the programs it executes.
- 3 The performance of a pipeline computer depends on the pipeline rate, the work load distributions, the vectorization ratio, & the utilization rate of system resources.
- 4 The system segments of a pipeline may operate on distinct data operands simultaneously.
- 5 Pipelining increases the bandwidth by a factor of K , since it may carry K independent operand sets in the K segments concurrently.
- 6 A pipeline computer requires more hardware & complex control circuitry than a corresponding serial computer.

K : no. of stages in functional pipe

T : total pipeline delay

n : the number of instructions contained in a test.

N_i : the length of vector operands used in i th instruction

W : the throughput of a pipeline computer

T_i : the time required to finish the i th instruction in a pipeline computer. ($1 \leq i \leq n$)

T_p : the total time req. to finish a task consisting of n instructions.

S_k : the speedup of a pipeline computer over corresponding serial computer

η : the efficiency of a pipeline computer

$$T_i = k \cdot t + (N_i - 1) \cdot t = (N_i + k - 1) \cdot \frac{t}{k}$$

$$T_p = \sum_{i=1}^n T_i = \frac{1}{k} \left[(k-1)n + \sum_{i=1}^n N_i \right]$$

$$S_k = \frac{T_s}{T_p}$$

$$= T \cdot \underbrace{\sum_{i=1}^n N_i}_{(k-1)n + \sum_{i=1}^n N_i}$$

$$T \left[(k-1)n + \sum_{i=1}^n N_i \right] / k$$

$$= k \cdot \sum_{i=1}^n N_i$$

$$(k-1)n + \sum_{i=1}^n N_i$$

$$\eta = \frac{T_s}{T_p} = \frac{S_k}{k}$$