Course Name: A Level (1st Sem) **Subject: CO**

Topic: Parallel Processing **Date: 01-06-20**

Parallel Processing: Parallel Processing is all about performing multiple processes at the same time either by using parallel hardware or by some kind of virtualization. Parallel hardware is always the first priority for processor developers because having multiple hardware bases retains the ease of processing instructions. Virtualization, on the other hand is a complex thing to achieve specially because, separate instructions are required for preparing a clone-like virtual processor. Even if we achieve this, we may use parallel hardware which in turn enables even more practicality to virtualization.

Here in this text, we are to discuss the following three ways of parallel processing:

1. Pipelining
2. Vector Processing
3. Array Processors

Pipelining: Conceptually pipelining refers to the fact that multiple tasks can be performed at the same time simply by letting those hardware do additional tasks that are free at the moment. In conventional pipelining tasks are distributed among parallel hardware in such a way that multiple sub tasks that are required for completion of a single parental task, are performed at the same time.

Pipelining is adopted in the following two ways:

A. Arithmetic Pipeline
B. Instruction Pipeline

Arithmetic Pipeline: Arithmetic pipeline is used in processors to deal with the complex arithmetic operations. The sub-operations that constitute a particular part of an expression can all be executed at the same time using different parallel hardware. The only thing to keep in mind is the precedence of operators. All operators of same precedence can be executed at the same time using different pipelines. Consider the following diagram:

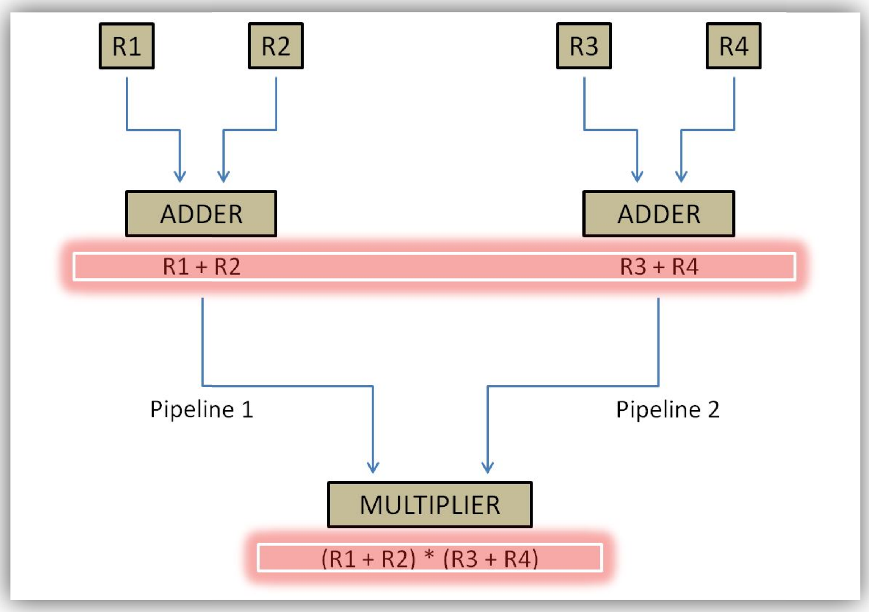
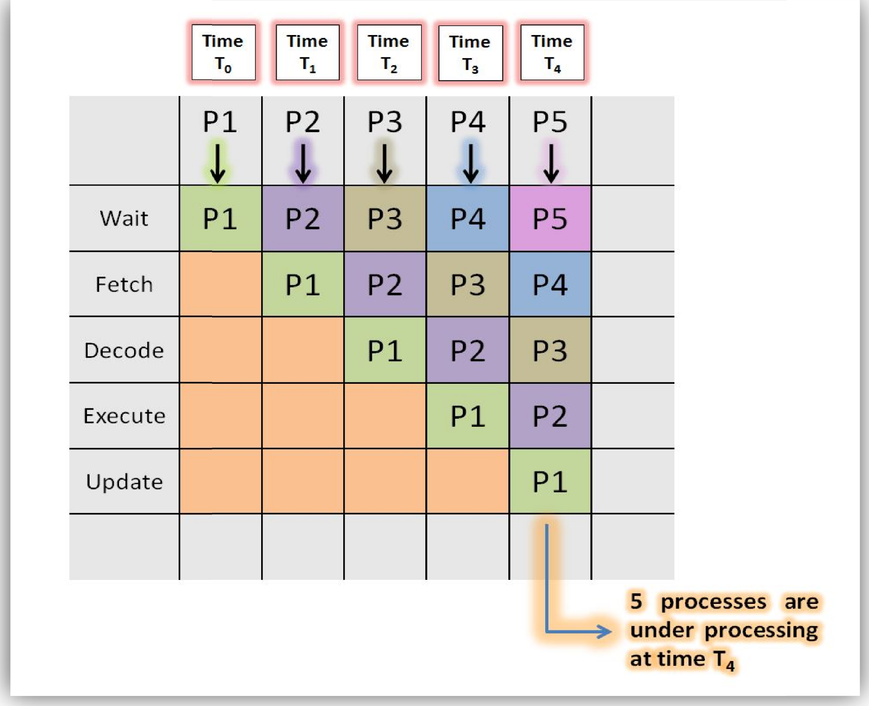


Fig.: Arithmatic Pipeline

Instruction Pipeline: Another pipelined approach is instruction pipeline. Every instruction goes through different hardware for different parts of its execution cycle. The goal here is to let the hardware deal with different instructions at the same time. For example, consider a conventional processor with three different hardware for FETCH, DECODE and EXECUTE tasks. When the first instruction is being fetched, the decode and execute circuits are free. But when the first instruction goes for decode, the fetch circuit can go busy with a second one. This way multiple instructions can be handled at the same time using instruction pipeline. Consider the following diagram:



	Time T_0	Time T_1	Time T_2	Time T_3	Time T_4
	P1 ↓	P2 ↓	P3 ↓	P4 ↓	P5 ↓
Wait	P1	P2	P3	P4	P5
Fetch		P1	P2	P3	P4
Decode			P1	P2	P3
Execute				P1	P2
Update					P1

5 processes are under processing at time T_4

Fig.: Instruction Pipeline

Vector Processing: Vectored processing is based upon the goal that, if a process requires two instructions to be executed and both on distant clocks, the processor must be aware about the projected next clock. This way processor can stay busy during the clock difference. In digital processing an instruction with the added information of its clock is considered as a vector and all these instructions (of this type) are called vector instructions. We denote these instructions as: V_1, V_2, V_3, ... V_n. A vectored processor keeps track of all such instructions from a separate memory block and all those vector instructions are executed at the same time for which the actual instruction part is same or similar.

Array Processing: Array processing is a slightly different approach that provokes asymmetric multiprocessing. The asymmetry here is in the terms of instruction type and complexity. Array processors are advanced to such an extent that they use different hardware for different types of instructions. Likewise, one hardware is doing simple instruction execution while a parallel one is performing complex statistical problems. Array processors are used specially in computers dealing with scientific operations.

Assignment:

1. What do you understand by parallel processing?
2. Differentiate between arithmetic pipeline and instruction pipeline.