Date 5.10.2020

Weekly Test 1

Name: Rutuja Deshmukh

Subject: HPC

Roll NO: COBE61

Q1

→ SIMD Architecture

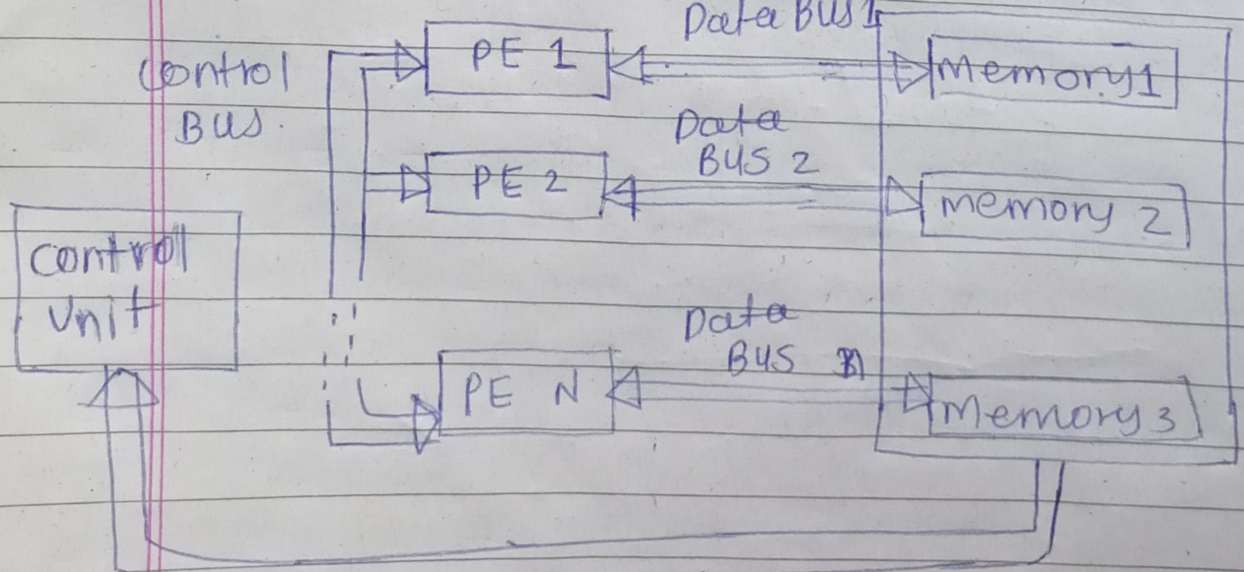
- Single Instruction Multiple Data.

→ single instruction is applied to a multiple data item to produce the same output.

→ Master instruction work on vector of operand.

→ No of processors running the same Instruction level parallelism.

→ communication network allow parallel synchronous communication between several processing element / memory modules



SIMD processor architecture.

→ All processors receive the same instruction from the control unit but operate on different items of data.

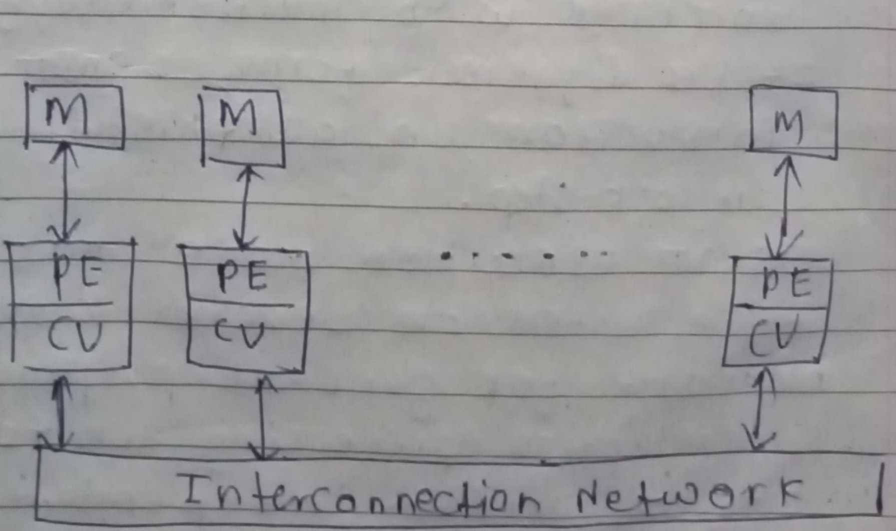
→ SIMD is mainly dedicated to array processing machines, however vector processors can also be seen as a part of this group.

→ MIMD architecture.

- MIMD stands for: Multiple Instruction & Multiple Data stream.

→ In this all processors in a parallel computer can execute different instructions & operate on various data at the same time.

→ In MIMD ~~each~~ each processor has a seperate program & an instuction stream is generated from each program.



where, M= memory module
PE = processing element
CU = control unit.

→ MIMD architecture may be used in a number of application areas such as computer aided design, simulation, modeling & as communication switenes.

→ MIMD architecture with shared memory have processors which share a common, central memory.

→ In the simplest form all processors are attached to a bus which connect them to memory.

- SIMT architecture.

→ It stands for single Instruction multiple threads.

→ SIMT is an execution model used in parallel computing where single instruction, multiple data (SIMD) is combined with multithreading.

→ It is different form SIMD in that all instruction in all threads are ran in lock-step.

→ SIMT execution model has been implemented on several GPUs & is relevant for general-purpose computing on graphics processing units.

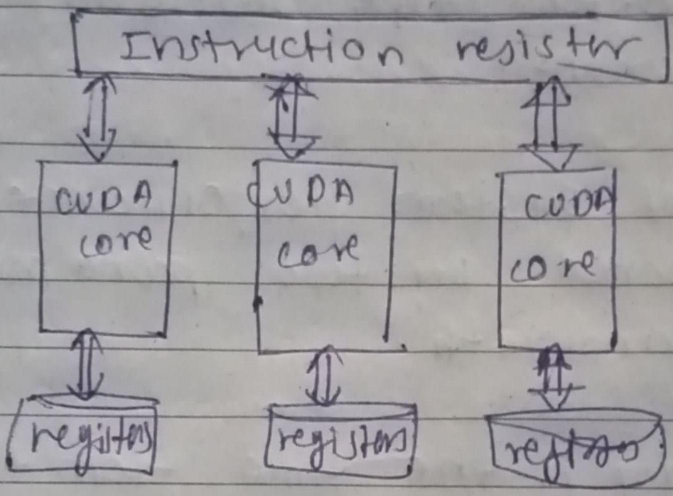
→ It is width-Independent programming model.

→ It has serial like code.

→ It has achieved by hardware with a little help from compiler.

→ All cores are executing the same Instruction.

→ each core has it's own set of registers



Example.

→ modern CPU desgignes include SIMT instructions to improve the performance of multimedia use

Q2. Limitetions of memory system performance.

→ Memory system & not processor speed is oftenn the bottleneck for many applications.

→ memory system performance is largely captured by two parameters, latency & bandwidth.

→ Latency is the time from the issue of memory request to the time data is available at the processor.

→ Bandwidth is rate at which data can be pumped to the processor by the memory system

→ memory latency : An example.

1. consider a processor operating at 1 GHz (1ns clock) connected to a DRAm with latency of 100 ns (no caches).

2. Assume that the processor has two multiply add units & is capable of executing four instruction in each cycle of 1 ns.

- The following observations follow.

1. The peak processor rating is 4 GFLOPs

2. Since the memory latency is equal to 100 cycles & block size is one word.

every time a memory request is made the processor must wait 100 cycles before it can process the data.

Q3.

→ Parallel computing.

- Is a type of computation where many calculations or the execution of processes are carried out simultaneously.
- Large problem can often be divided into smaller ones, which can then be solved at the same timle.
- It is one of type of computing architecture in which several processors simulatenosly execute multiple, smaller calculation broken down from an overall larger complex problem

Types of parallel computing.

1. Bit level parallelism.

→ It is the form of parallel computing which is based on the increasing processors size.

→ It reduces the number of instruction that system must execute in order to peform a task on large sized data.

2. Instruction level parallelism

→ A processor can only adress less than one instruction for each clock cycle phase. These instructions

→ can be reordered & grouped which are later on executed concurrently without affecting the result of program.

→

3. Task parallelism.

→ Task parallelism employs the decomposition of a task into subtasks & then allocating each of the subtasks for execution.