**Parallel Programming Foundation**

1. **Define the following: Tasks, Pipelining, Shared Memory, Communications, Synchronization.**

* Task: A program with a set of instructions that is executed by a processor. In parallel programming, multiple tasks run on multiple processors.
* Pipelining: The process of breaking down a task by different processor units. It is also a type of parallel computing.
* Shared Memory: A computer architecture where processors have direct access to a common physical memory. In parallel programing, it is a model where parallel tasks have access direct access and address the same logical memory location regardless of where the physical memory is located.
* Communications: The exchange of data by parallel tasks. They can be completed through a shared memory bus or over a network.
* Synchronization: The coordination of parallel task in real time. A synchronization point is established within an application where a task cannot proceed further until another task(s) reaches the same or logically equivalent point. Often associated with communications.

1. **Classify parallel computers based on Flynn’s taxonomy.**

Based on Flynn’s Taxonomy, there are four parallel computers

* Single Instruction, Single Data (SISD): A serial (non-parallel) computer. Only one instruction stream is performed by the CPU. Only one data stream is being used as an input during one clock cycle. Deterministic execution. Oldest type of computer.
* Single Instruction, Multiple Data (SIMD): A type of parallel computer. All processing units execute the same instruction at any given clock cycle. Each processing unit can operate on a different data element. Synchronous (lockstep) and deterministic execution. Two varieties: Processor Arrays and Vector Pipelines. Most modern computers employ SIMD instructions and execution units.
* Multiple Instruction, Single Data (MISD): A type of parallel computer. Each processing unit operates on data independently via separate instruction streams. A single data stream is fed into multiple processing units. Few actual parallel computers of this type have existed.
* Multiple Instruction, Multiple Data (MIMD): A type of parallel computer. Every processor is possible of executing a different instruction stream. Every processor is capable of working with a different data stream. Execution can be synchronous or asynchronous, deterministic or non-deterministic. It is currently the most common type of parallel computer.

1. **What are the Parallel Programming Models?**

The Parallel Programming Models are:

* Shared Memory without threads
* Threads
* Distribution Memory/ Message Passing
* Data Parallel
* Hybrid
* Single Program Multiple Data (SPMD)
* Multiple Program Multiple Data (MPMD)

1. **List and briefly describe the types of Parallel Computer Memory Architectures. What type is used by OpenMP and why?**

Some common characteristics include having the ability for all processors to access all memory as a global address space. Multiple processors can operate independently but share the same memory resources. Changes in a memory location effected by one processor is visible to all other processors.

* Uniform Memory Access (UMA): Commonly represented in the present as Symmetric Multiprocessor machines (SMP). Has identical processors. Equal access and access times to memory. Also known as Cache Coherent UMA; means if one processor updates a location in shared memory, all other processors know about the update and is accomplished in the hardware level.
* Non-Uniform Memory Access (NUMA): Created by physically linking two or more SMPs. One SMP directly accesses the memory of another SMP. Not all processors have equal access times to all memories. Memory access across link is slower. Can also be cache coherent.

1. **Compare Shared Memory Model with Threads Model?**
2. **What is Parallel Programming?**

Parallel Programming is when multiple programs are executed simultaneously. The work is divided into chunks or parts where they are further broken down into instructions. These instructions are executed concurrently on different processors/cores. Also, in Parallel Programming, multiple tasks are completed through the use of scheduling and prioritizing.

1. **What is system on chip (SoC)? Does Raspberry PI use system on SoC?**

System on chip, or SoC, is an entire computer on one chip. It contains a CPU, GPU, memory, USB controller, power management circuits and wireless radios (WiFi, 3G, 4G LTE, etc.). The Raspberry PI uses SoC.

1. **Explain what the advantages are of having a System on a Chip rather than separate CPU, GPU and RAM components.**

The main advantage of SoC over separate components is the size. SoC is only a bit larger than a CPU but contains way more functionality. With the smaller size, less wiring is required which uses less power to operate. It is also cheaper to build a computer using SoC.