Parallel Programming

Part A

1. **Define the following: Task, Pipelining, Shared Memory, Communications, Synchronizations.**

Task: set of instruction that the processor needs to run so that it can accomplish the program which was assigned by a programmer. Till now, we have covered the logical operations task, which are the basic arithmetic. Processors now are better than ever before, and CPU is now able to perform multiple tasks at once using a parallel programming.

Pipeline: pipeline breaks down the assigned tasks into the smaller task. So, the computer can group the task base on the same instruction set to save the performance time.

Shared Memory: memory is being shared regardless where memory is located. This is because the memory is steered only in one physical memory. An advantage of shared memory is that the all memory shares the same “address bus” and computer architecture, every logical operation would directly relate to another.

Communications: communication is inevitable when trying to perform a parallel task. Multi-task is currently trending and sharing the data within the memory is necessary. One way to communicate (exchanging data) is to use an address bus. One restriction for using an address bus is that the “shared memory” is necessary.

Synchronization: synchronization is like the control unit in CPU where it coordinates the all task at the same period. Synchronization often cooperate with the “communication” to accomplish the task. Synchronization checks whether the all tasks were accomplished before moving to another task.

1. **Classify parallel computers based on Flynn’s taxonomy. Briefly describe every one of them.**

Flynn’s taxonomy has a 4 classification in parallel computing. These are 1) “Single Instruction stream Single Data Stream (SISD) 2) Single Instruction stream Multiple Data Streak (SIMD) 3) Multiple Instruction Stream Single Data Stream (MISD) 4) Multiple Instruction Stream Multiple Data Stream (MIMD).

To explain the difference among all classification, the SISD was used in the oldest computer since it only can have single instruction; only one instruction can be performed by CPU at one; and Single Data; only one data can be used in one clock cycle. This is very inefficient to deal with data, and SIMD was invented.

Unlike SISD, SIMD handles the multiple data at one clock cycle. Each processer can operate the different data to solve an institution. This allow the computer to use an Arrays and vector. And this type of parallel programming is used in most of the modern computer. There is a compute that could handle the multiple instruction, but single data. It is MISD. MISD can only have one address bus between the Process unit but having a multiple process unit. This allow that the individual processing unit can operate their task as an independent, but the data is being fed into one data pool.

The most advanced one is called MIMD. This allow computer to perform a multiple instruction, and multiple data pool into one clock cycle. This parallel computing is used in a supercomputer, like server computer. Each Processing unit is independent from another, as well as the data

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

Parallel programming model is an abstraction of computer architecture, which tells the essential competent to perform a algorithms and arithmetic of set instruction

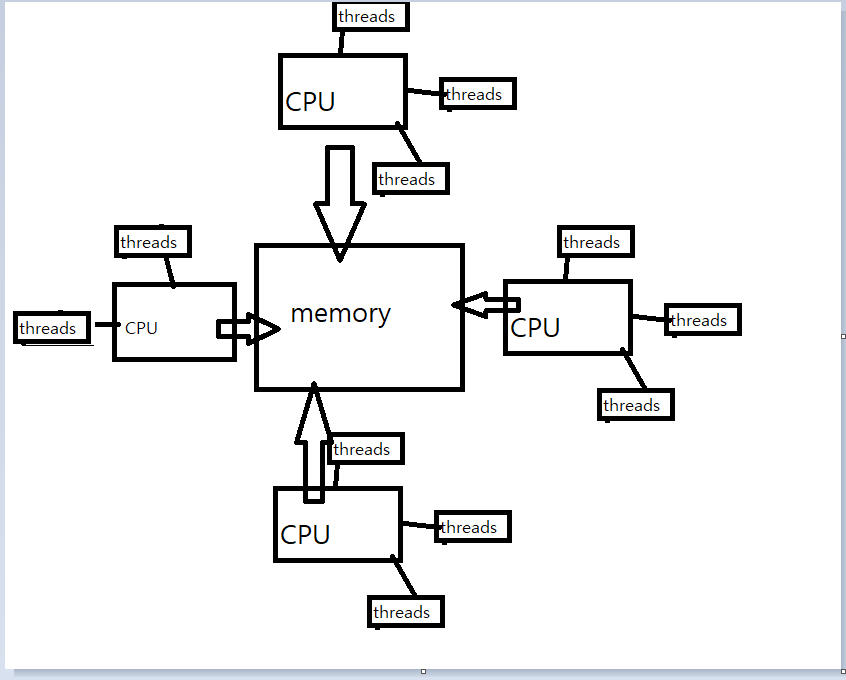
1. **List and briefly describe the types of Parallel Computer Memory Architectures. What type is used by OpenMP and why?**

Parallel Computer Memory has two form. One is Uniform Memory Access (UMA), and the other one is Non-Uniform Memory Access (NUMA). UMA is commonly called as the Symmetric multiprocessor machines because inside the CPU everything shares the common memory when performing the parallel computing. Using the identical memory will let the computer have equal access to both the time & memory. While, non-uniform memory access would have several memories, connected with Bus. This will allow CPU to be independent and is often made by physically linking several SMPs.

The OpenMP is uniform memory access. The OpenMP is a standard API to support the shared memory multiprocessing programming, and OpenMP shares the common memory access. Since NUMA has multiple memories connected using bus, it would be inefficient to use NUMA architecture. So, OpenMP uses UMA.

1. **Compare Shared Memory Model with Threads Model?**

Shared Memory model with threads - single threads could have a multiple weight. This allows computer to do a serial work and perform a number of tasks using threads. This eventually make you to run program to schedule and operating system concurrently.



1. **What is Parallel Programming?**

Parallel programming is a computer architecture that allows you to perform multiple tasks concurrently. Instead of one by one, parallel programming groups the set of instruction, and schedule appropriate to perform tasks faster.

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

SoC is an integrated chip that squeezes the CPU, Ram and GPU into one chip. This saves an enormous amount of space and allows you to perform as computer. The Raspberry Pi uses SoC that contains gARM1176JZF-S single core CPU, 512 SDRAM.

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

Since every component of computer is integrated in SoC, it requires a smaller footprint and space compare to other. Having one single chip increases the reliability since everything is connected to one, and little power is required.