CSC3210 Assignment A3 Task 3 – Tony Ngo

**Define the following: Task, Pipelining, Shared Memory, Communications, Synchronization. (in your own words)**

Task – A set of instructions that is executed by a processor.

Pipelining – Pipelining is taking a task and breaking it into separate steps performed by different processing units.

Shared Memory – Hardware-wise, shared memory shows where in the architecture where processors have direct access to physical memory (RAM). Programming-wise, it describes where parallel tasks can access the same logical memory regardless of where the memory actually exists.

Communications – Communication is considered exchanged data between parallel tasks.

Synchronization – Synchronization is the coordination of parallel tasks- usually implemented by creating a “synchronization point”.

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

SISD (Single Instruction, Single Data): This is the oldest type of computer that does not have parallel instruction. Single instruction means that there is only one instruction executing during a single CPU Clock Cycle. Single data means that there’s only one stream of data being used as input during a single CPU Clock Cycle.

SIMD (Single Instruction, Multiple Data): This is a parallel computer that has two types (Processor Arrays & Vector Pipelines); they are specialized for issue that need intense graphics/image processing. Computers that have GPUs have SIMD instruction & execution units. Single instruction means that there is only one instruction executing during each CPU cycle. Multiple Data means that each processing unit can operate on a different data unit.

MISD (Multiple Instruction, Single Data): This is a parallel computer that is very uncommon, they are used in very specific cases (i.e. decoding cryptography algorithms). Multiple Instructions means that each processing unit operates independently via separate instruction streams. Single Data means that single data streams are fed into multiple processing units.

MIMD (Multiple Instruction, Multiple Data): This is a parallel computer that integrate both multiple instructions and multiple data; the only computers that can do this are mainly supercomputers. Usually, MIMD computers utilize SIMD components. Multiple instruction means that each processing unit operates independently via separate instruction streams. Multiple Data means that each processing unit can operate on a different data unit.

**What are the Parallel Programming Models?**

Parallel Programming models:

* Shared Memory (w/out threads)
* Threads – A type of shared memory programming. Usually used in libraries/compiler directives (i.e. OpenMP).
* Distributed Memory/Message Passing – Distributed memory means that memory is physically distributed across a network of machines through specialized hardware and software.
* Data Parallel
* Hybrid
* Single Program Multiple Data (SPMD)
* Multiple Program Multiple Data (MPMD)

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

UMA (Uniform Memory Access) – Uniform Memory Access machines have identical processors which have equal access and access times to memory. It includes CC-UMAA which means that if one cache is updated, the rest are also updated. They are commonly represented by Symmetric Multiprocessor machines (SMP).

NUMA (Non-Uniform Memory Access) – Non-Uniform Memory Access machines consist of two or more SMPs and do not have equal access time to the memory, which means the processes are usually slower.

OpenMP uses UMA because it has equal access times to all memory locations, which is a characteristic of UMA.

**Compare Shared Memory Model with Threads Model? (in your own words and show pictures)**

The shared memory model is the most basic form of parallel programming which shows that processes/tasks are written into a singular space where they can read/write from. In the thread model, you can choose how much processing power is put into a single part of a program, so it lets you focus more threads on heavier tasks.

A close up of text on a whiteboard

Description automatically generated

**What is Parallel Programming? (in your own words)**

Parallel programming is the utilization of multiple computer threads or cores to execute a single instruction, thus making it faster and more efficient to solve complex issues.

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

A SoC (System on Chip) integrates a CPU, GPU, memory, and IO devices together onto a single chip. The Raspberry PI does use a System on Chip.

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

A System on Chip system allows you to decrease the size of a product significantly, which is extremely beneficial in smartphone development since you have more space to put a battery; also it utilizes less power since there is shorter wiring that is associated with it. However, there is no room for expandability, since the SoC is integrated directly onto the chip.