CS3006 - Parallel and Distributed Computing

Lecture 03: Flynn's Taxonomy and Classification based on Parallelism

Syed Faisal Ali

email: faisal.ali@nu.edu.pk

National University of Computer and Emerging Sciences - FAST, Karachi Campus

Tuesday, Aug 22, FALL - 2023

An Overview of Parallel Processing

Parallel Processing

Parallel processing is a method to imporve computer system performance by executing two or more instructions simultaneously.

Goals of Parallel Processing

- One goal is to reduce the "wall-clock" time or the amount of real time that you need to wait for a problem to be solved.
- Aother goal is to solve bigger problems that might not fit in limited memory of a single CPU.

Parallelism in Uniprocessor Systems

- It is possible to achieve parallism with a uniprocessor system.
- Some examples are the instructions pipeline, arithmetic pipeline, I/O processor.
- Note that a system that performs different operations on the same instruction is not considered a parallel.
- Only if the system processes two different instructions simultaneously can it be considered parallel.

Classification of Parallel Computers

Two Kinds of Classification

1. Flynn's Classifciation
    - SISD (Single Instruction Stream, Single Data Stream)
    - SIMD (Single Instruction Stream, Multiple Data Stream)
    - MISD (Multiple Instruction Stream, Single Data Stream)
    - MIMD (Multiple Instruction Stream, Multiple Data Stream)
2. Classifciation by Memory Status.
    - Share Memory
    - Distributed Memory

Organizaton of Multiprocessor Systems

Flynn's Classification

- Was proposed by researacher Micheal J. Flynn in 1996.
- It was the most commonly accpected taxonomy of computer organization.
- In this classification, computers are classified by whether it processes a single instruction at a time or multiple instructions simultaneously, and whether it operates on one or multiple data sets

Flynn's Classification of Taxonomy

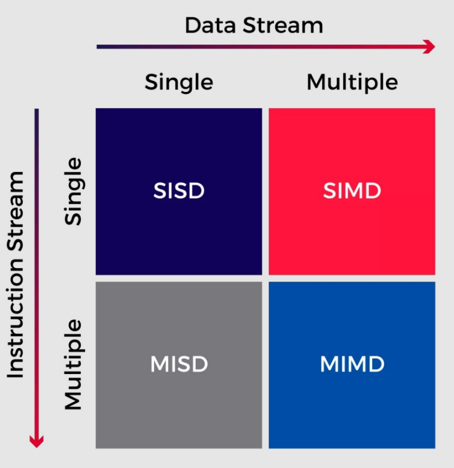

Figure: Flynn's classification of taxonomy in details according to Data Stream and Instruction Stream

Single Instruction Single Data

- Von Neuman's one processor computer

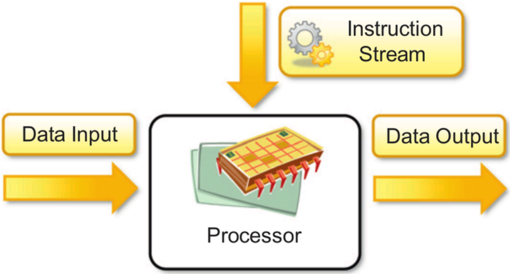

Figure: Single Instruction and Single Data Stream Architecture

Single Instruction and Single Data Stream Architecture

- A serial (non-parallel) computer.
- Single Instruction: only one instruction stream is being acted on by the CPU during any one clock cycle.
- Single Data: only one data stream is being used as input during any one clock cycle.
- Deterministic execution.
- This is the oldest and until recently, the most prevalent form of computer.
- Examples: most PCs, single CPU workstations and mainframes.

Single Instruction Multiple Data

- Processors execute the same instructions on different data.
- Operations of processors are synchronized by global clock.

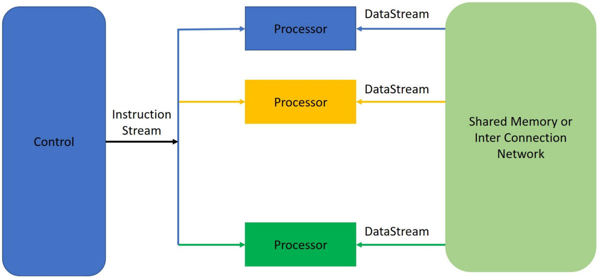

Figure: Single Instruction and Multiple Data Stream Architecture

Single Instruction and Multiple Data Stream Architecture

- A type of parallel computer.
- Single Instruction: All processing units execute the same instruction at any given clock cycle.
- Multiple Data: Each processing unit can operate on a different data element.
- This type of machine typically has an instruction dispatcher, a very high-bandwidth internal network, and a very large array of very small-capacity instruction units.
- Best suited for specialized problems characterized by a high degree of regularity,such as image processing.

Single Instruction and Multiple Data Stream Architecture

- Synchronous (lockstep) and deterministic execution.
- Two Varieties: Processor Arrays and Vector Pipelines.
  Examples:
- Processor Arrays: Connection Machine CM-2, Maspar MP-1, MP-2.
- Vector Pipelines: IBM 9000, Cray C90, Fujitsu VP, NEC SX-2, Hitachi S820.

Multiple Instructions Single Data

- All processors share a common memory, have their own control devices and execute their own instructions on same data.

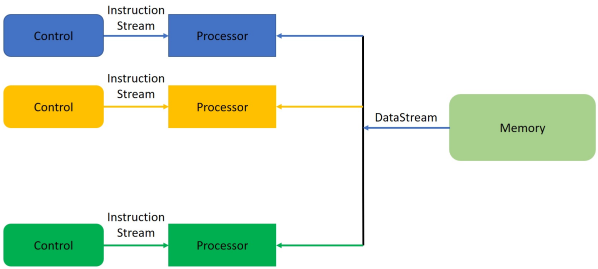

Figure: Multiple Instructions and Single Data Stream Architecture

Multiple Instructions and Single Data Stream Architecture

- A single data stream is fed into multiple processing units.
- Each processing unit operates on the data independently via independent instruction streams.
- Few actual examples of this class of parallel computer have ever existed. One is the experimental Carnegie-Mellon C.mmp computer (1971).
- Some conceivable uses might be:
- Multiple frequency filters operating on a single signal stream.
- Multiple cryptography algorithms attempting to crack a single coded message.

Multiple Instructions Multiple Data

- Processors have their own control devices, and execute different instructions on different data.
- Operations of processors are executed asynchronously in most time.
- It is also called as distributed computing system

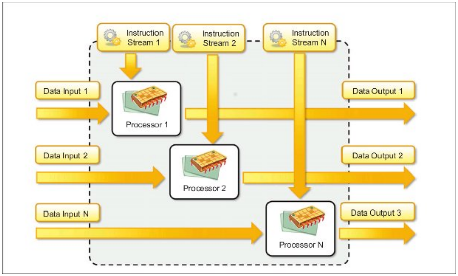

Figure: Multiple Instruction and Multiple Data Stream Architecture

Multiple Instructions and Multiple Data Stream Architecture

- Currently, the most common type of parallel computer. Most modern computers fall into this category.
- Multiple Instruction: every processor may be executing a different instruction stream
- Multiple Data: every processor may be working with a different data stream
- Execution can be synchronous or asynchronous, deterministic or non-deterministic
- Examples: most current supercomputers, networked parallel computer "grids" and multi-processor SMP computers - including some types of PCs.

Flynn's Classification of Taxonomy in One Diagram

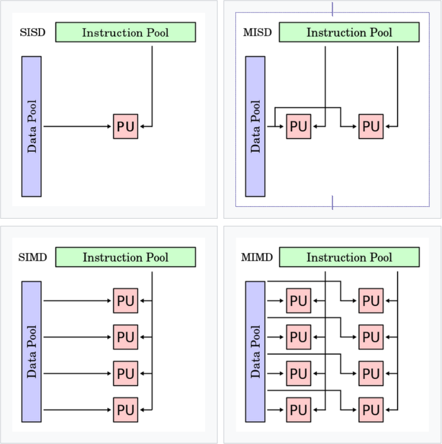

Figure: Four Architectures

Some General Parallel Terminologies

1. Task
   - A logically discrete section of computational work. A task is typically a program or program-like set of instructions that is executed by a processor.
2. Parallel Task
   - A task that can be executed by multiple processors safely (yields correct results).
3. Serial Execution
   - Execution of a program sequentially, one statement at a time. In the simplest sense, this is what happens on a one processor machine. However, virtually all parallel tasks will have sections of a parallel program that must be executed serially.

Symmetric vs. Asymmetric Multiprocessing Architecture

1. Asymmetric Multiprocessing
   - In asymmetric multiprocessing systems, all the CPUs are not equal. In other words, one processor might be executing the operating system code while another processor is performing input and output tasks. Asymmetric multiprocessing applications are used when each processor is dedicated to performing a specific task.

Symmetric vs. Asymmetric Multiprocessing Architecture

1. Asymmetric Multiprocessing
   - Asymmetric multiprocessing uses a master-slave approach. One processor works as the master while the other processors work as slave processors. Master processor monitors the other processors and assigns tasks to each slave processor. Assume that there are four CPUs as C1, C2, C3 and C4. C4 is the master processor, and it assigns tasks to the other processors. Assume that C1 is assigned with process P1, C2 is assigned with process P2, and C3 is assigned with process P3. Each processor will be working only on the assigned processes.

Symmetric vs. Asymmetric Multiprocessing Architecture

1. Symmetric Multiprocessing
   - In a symmetric multiprocessing system, two or more processors are connected to a single, shared main memory. All the processors have full access to the input and output devices. The operating system treats all these processors equally. It is a tightly coupled multiprocessor system with a set of similar processors running independently. Each processor executes different programs and works on different sets of data. They share the common resources such as memory, IO device, etc.

Reading

1. Reading Assignment: Read Chapter 1: Introduction to Parallel Computing 12 to page 20, from Introduction to Parallel Computing 2nd Edition by Ananth Grama.
2. List three major supercomputers with their specifications. What type of architecture do you think supercomputers have?
3. Find the hardware configuration of your PC. Write them in detail.
4. Open task window and list down the softwares which consumes maximum memory.

End of Lecture