Parallel Computation

P.Y. Wang

Department of Computer Science 4A5

George Mason University

Fairfax VA 22030-4444 U.S.A.

Introduction

- **Parallel computation:**
  Solving a computational problem by using a multi-processor or multiple computers
- **Motivation:**
  a galaxy simulation with $n = 10^{11}$ stars using a computer that requires $1\mu S$ per calculation

Number of calculations	Time for One Iteration
n^2	10^9 years
$n \lg n$	1 year

- Multiprocessor systems and parallel programming have been around since the 1960's
- Thus, parallel computation can be studied from two perspectives: a *hardware* view and a *software* view

A Hardware View

- What is a parallel computer?
- Architectural design issues
  - Amount of parallelism
  - Classification of designs

A Software View

- Programming models and algorithm design issues
- Parallel programming languages
- Implementation of data structures and algorithms
- Complexity and performance measures

A Hardware View

- Models
  - Flynn's Taxonomy
  - The PRAM
- The Evolution of Parallel Architectures

Theoretical Models (Flynn's Taxonomy)

- Flynn's Taxonomy is based on the notion of instruction streams and data streams
- An instruction stream is a sequence of instructions
- A data stream is a sequence of data

Let **S** = Single, **M** = Multiple, **I** = Instruction, and **D** = Data. Then

- **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

SISD

Single Instruction stream, Single Data stream

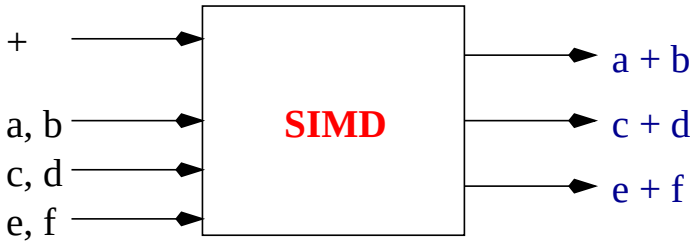
One instruction is performed at a time on one set of data:

This class contains the traditional sequential computers.

SIMD

Single Instruction stream, Multiple Data stream

One instruction is performed at a time on possibly different data:

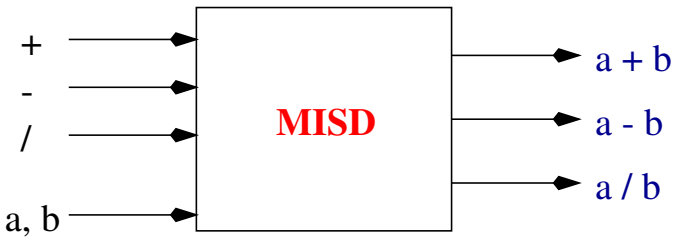


This class contains machines like the ILLIAC IV, STARAN, MPP, DAP, GAPP, CM-200

MISD

Multiple Instruction stream, Single Data stream

Different instructions can be performed on the same data:



MIMD

Multiple Instruction stream, Multiple Data stream

Multiple instructions are executed in parallel on multiple data sets:

This class contains machines like the Intel iPSC Hypercube, Sequent Balance, BBN Butterfly, INMOS Transputer System, NCUBE, Intel Paragon, CM-5

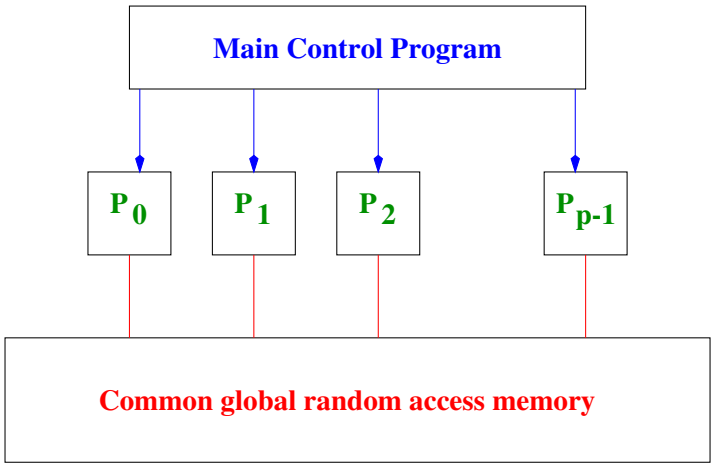
Flynn's Taxonomy does not address many architectural issues, including:

- Memory: shared vs. distributed
  - How much?
  - Both available?
  - How accessed?
- System granularity:
  - How many processors?
  - How powerful?
- Communication issues:
  - Routing hardware?
  - Control mechanisms?

The PRAM

- PRAM : Parallel Random Access Machine
- It has p processors (RAMs) that are connected to a large shared memory M
- The processors work in synchronized mode
- They can have private (local) memory for their own computations
- Communication takes place only through the shared memory

PRAM Model



CREW - PRAM

- Any number of processor can simultaneously read from the same memory location
- No two processors can write simultaneously to the same memory location
- An initial threshold time of $\log p$ is implicitly presumed– it activates p processors at the beginning of any computation

Algorithm Pseudo-Code

- PRAM construct for data set X in common global memory:

  forall P_i do in parallel
  `instruction(x)`

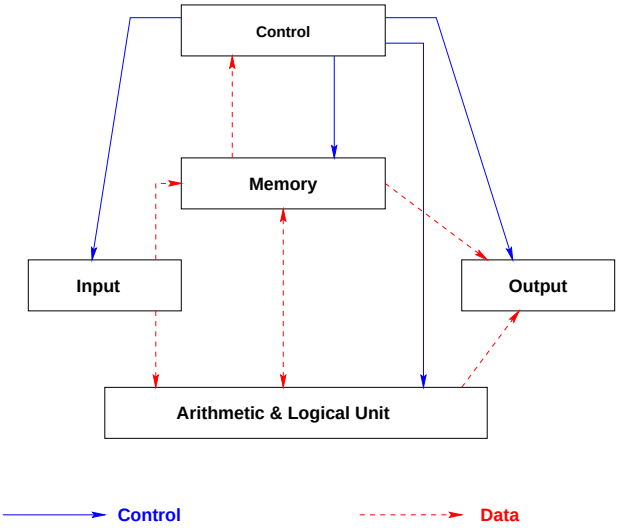
- Each processor P_i is assigned one of the x $\in$ X
- In parallel, each processor executes `instruction` on its own x

Other PRAM Models

- EREW: Exclusive Read, Exclusive Write (weakest model)
- CRCW: Concurrent Read, Concurrent Write (strongest model)
- W-RAM:
  - simultaneous writes to selected memory locations
  - can only write 0's and 1's to these locations
  - simultaneous write must be all 1's
- Common-Write PRAM: all processors can only write simultaneously if they write the same value

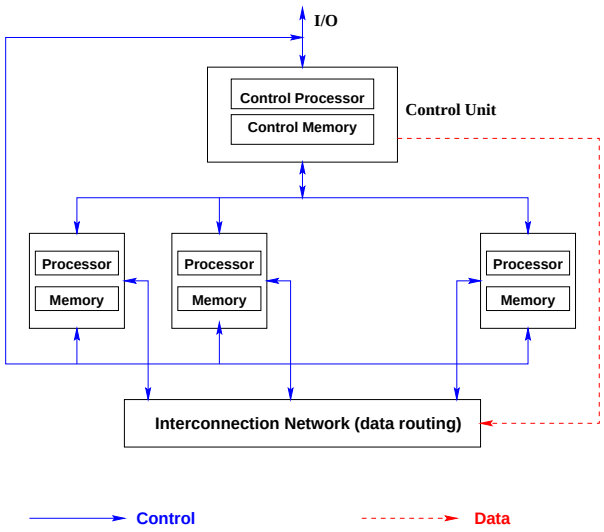
- Priority-Write PRAM:
  - processors can write simultaneously
  - the one with the highest priority always wins
  - highest subscript (self-address)
- Arbitrary-Write PRAM:
  - all processors can write simultaneously
  - arbitrarily, one will succeed
- Summation-Write PRAM:
  - all processors can write simultaneously
  - the summation (or any associative operation) of the values will be written

The Evolution of Parallel Architectures



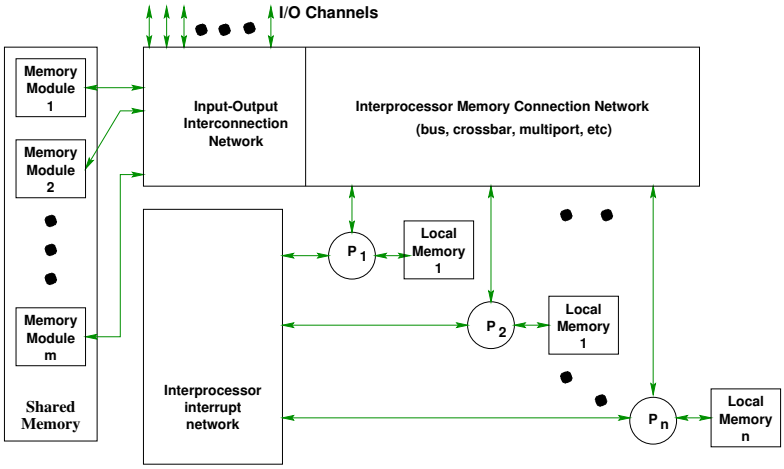
Serial Computer Organization

Data-Parallel/SIMD Computers



MIMD Architectures

Multiprocessors and Multicomputers

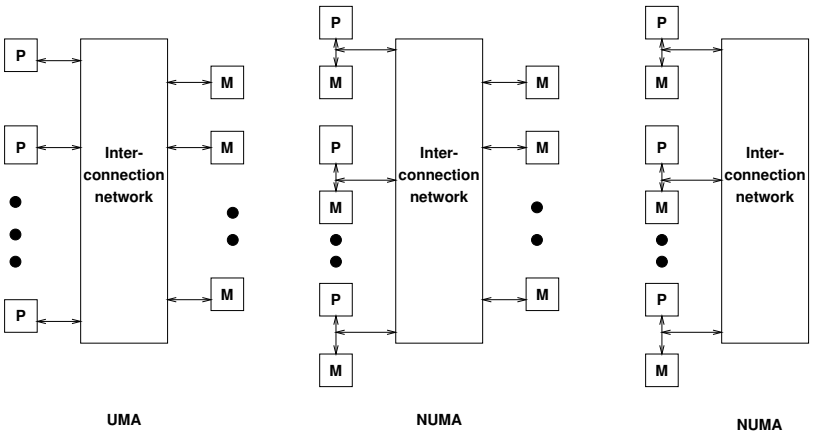


Message Passing vs. Shared-Address-Space

- Message Passing:
  - each processor has local memory
  - processors can only interact by passing messages
  - also called distributed memory machines and/or multicomputers
- Shared-Address-Space:
  - processors interact by modifying objects stored in shared address space
  - also called multiprocessors

UMA vs. NUMA Shared-Address-Space Architectures

- UMA (Uniform Memory Access) Computers: the time taken by a processors to access any memory word in the system is identical
- NUMA (Non-Uniform Memory Access) Computers: the time taken by a processor to access a remote memory bank is longer than the time to access a local memory bank
- most shared-address-space architectures have local cache



Cluster Computing

- Network of workstations or PC's can be a cost-effective platform for parallel computing
- 1980's
  - Berkeley NOW (network of workstations)
  - U-Wisconsin COW (Cluster of workstations)
  - NASA Beowulf

Beowulf Systems (NASA GSFC)

- COTS
- Pentium-class processors
- Fast/gigabit Ethernet; Myrinet; Infiniband
- Linux OS; GNU

Typical Cluster Configuration