Unit 1
INTRODUCTION

Introduction

- The traditional logical view of a sequential computer consists of a memory connected to a processor via a Datapath
- In this unit, we will discuss an overview of architectural concepts to parallel processing

Pipelining and Superscalar Execution

- Processors have long relied on pipelines for improving execution rates
- The assembly-line analogy works well for understanding pipeline
  - Example : Water bottle packing
- To improve instruction execution rate is to use multiple pipelines

Very long Instruction Word Processors

- The parallelism extracted by superscalar processors is often limited by the instruction look-ahead
- An alternate concept for exploiting instruction-level parallelism used in very long instruction word (VLIW) processors relies on the compiler to **resolve dependencies and resource availability at compile time**

Example : Superscalar execution

```
1. load R1, @1000
2. load R2, @1008
3. add R1, @1004
4. add R2, @100C
5. add R1, R2
6. store R1, @2000
```

(i)

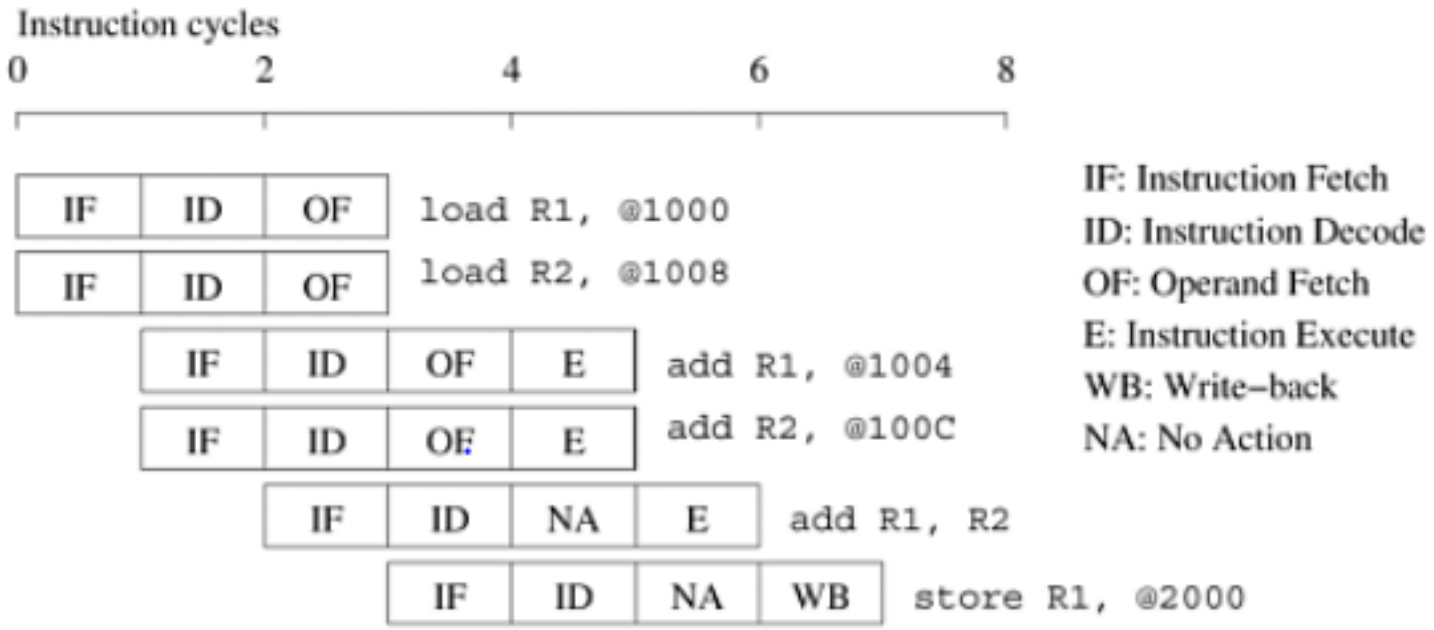
```
1. load R1, @1000
2. add R1, @1004
3. add R1, @1008
4. add R1, @100C
5. store R1, @2000
```

(ii)

```
1. load R1, @1000
2. add R1, @1004
3. load R2, @1008
4. add R2, @100C
5. add R1, R2
6. store R1, @2000
```

(iii)

Example : Superscalar execution



Limitations of Memory System Performance

- The effective performance of a program on a computer relies not just on the speed of the processor but also on the **ability of the memory system to feed data to the processor**
- A memory system, possibly consisting of multiple levels of caches, takes in a request for a memory word and returns a block of data of size b containing the requested word after l nanoseconds.

Improving Effective Memory Latency Using Caches

- One innovation addresses the speed mismatch by placing a smaller and faster memory between the processor and the DRAM
- The fraction of data references satisfied by the cache is called the cache hit ratio of the computation on the system
- The effective computation rate of many applications is bounded not by the processing rate of the CPU, but by the rate at which data can be pumped into the CPU. Such computations are referred to as being memory bound

Limitations of Memory System Performance

Impact of Memory Bandwidth

- Memory bandwidth refers to the rate at which data can be moved between the processor and memory.
- One commonly used technique to improve memory bandwidth is to increase the size of the memory blocks

Alternate Approaches for Hiding Memory Latency

- Prefetching
- Multithreading

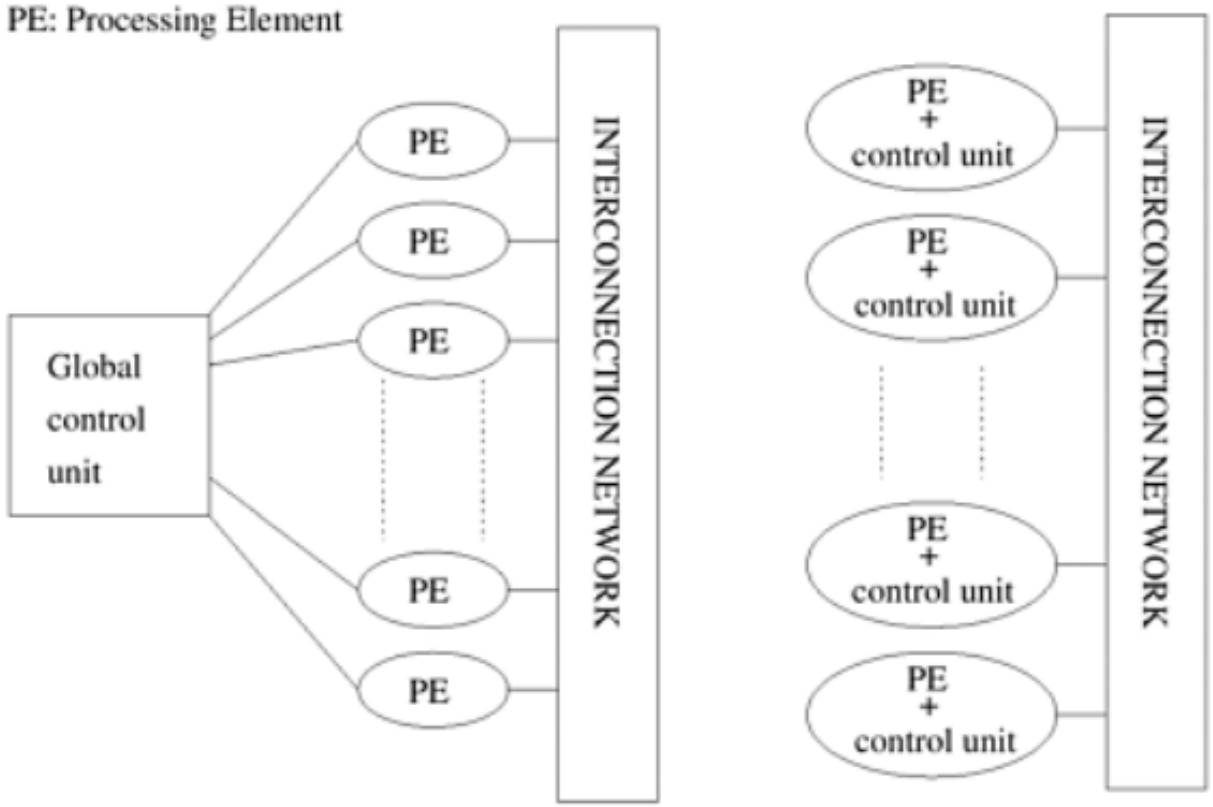
Dichotomy of Parallel Computing Platforms

- An explicitly parallel program must specify concurrency and interaction between concurrent sub tasks
- The two critical components of parallel computing from a programmer's perspective are ways of expressing parallel tasks and mechanisms for specifying interaction between these tasks
- The former is sometimes also referred to as the control structure and the latter as the communication model

Control Structure of Parallel Program

- Parallel tasks can be specified at various levels of granularity
- At one extreme, each program in a set of programs can be viewed as one parallel task.
- At the other extreme, individual instructions within a program can be viewed as parallel tasks
- Processing units in parallel computers either operate under the centralized control of a single control unit or work independently
- If there is a single control unit that dispatched the same instruction to various processors (that work on different data), the model is refers to as single instruction stream, multiple data stream(SIMD)
- Computers in which each processing element is capable of executing a different program independent of the other processing elements are called multiple instruction stream, multiple data stream (MIMD) computers.

Control Structure of Parallel Program



Communication Model of Parallel Platforms

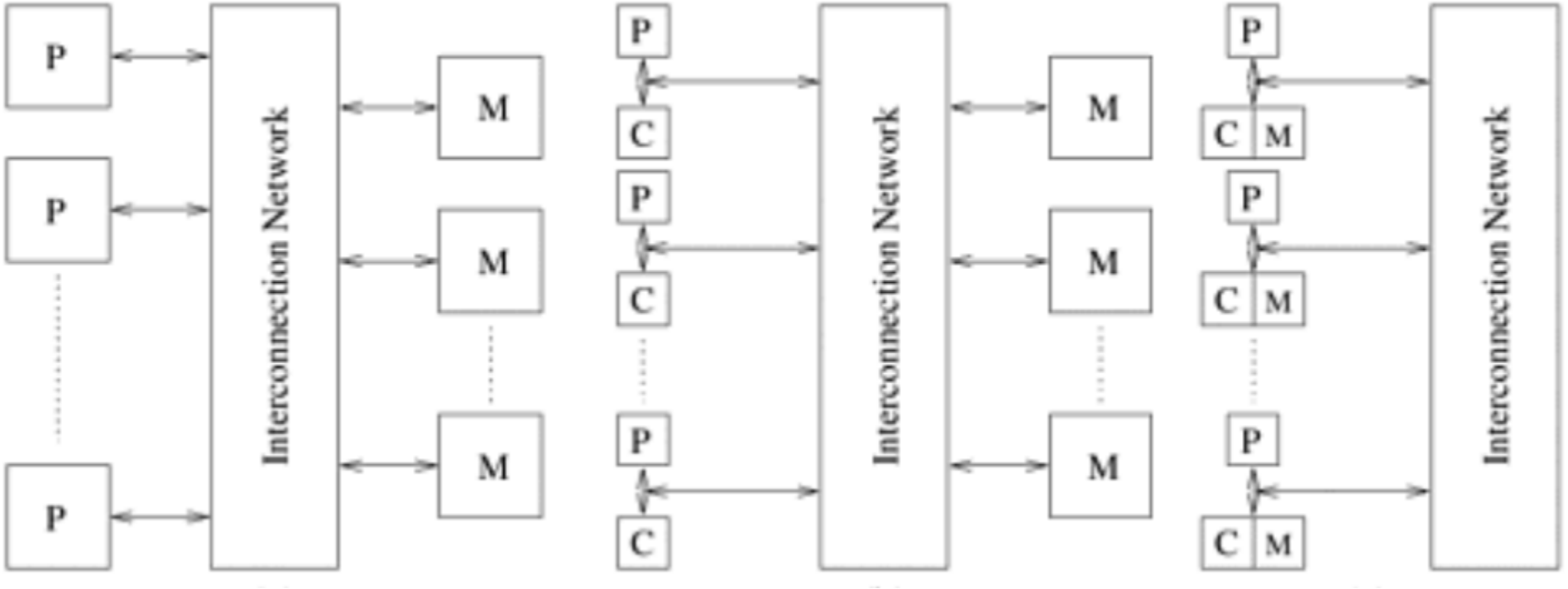
- There are two primary forms of data exchange between parallel tasks.
  - Accessing a shared data space
  - Exchanging messages
- Platforms that provide a shared data space are called **shared-address-space machines** or multiprocessors.
- Platforms that support messaging are also called **message passing platforms** or multicomputer

Shared-Address-Space Platforms

- Part (or all) of the memory is accessible to all processors
- Processors interact by modifying data objects stored in this shared-address-space
- If the time taken by a processor to access any memory word in the system global or local is identical, the platform is classifies as a uniform memory access(UMA), else, a non-uniform memory access(NUMA) machine

Communication Model of Parallel Platforms

NUMA and UMA Shared-address-space platforms



Communication Model of Parallel Platforms

Message-Passing Platforms

- These platforms comprise of a set of processors and their own memory
- Instances of such a view come naturally from clustered workstations and non-shared-address-space multicomputer
- These platforms are programmed using (variants of) send and receive primitives
- Libraries such as MPI and PVM provide such primitives

Physical Organization of Parallel Platforms

Architecture of an ideal Parallel Computer

- A natural extension of the Random access machine serial architecture is the parallel Random access Machine, or PRAM.
- PRAM's consists of p processors and a global memory of unbounded size that is uniformly accessible to all processors
- Processors share a common clock but may execute different instructions in each cycle.
- Depending on how simultaneous memory accesses are handled, PRAM's can be divided into four subclasses.
  - Exclusive-read, exclusive-write (EREW) PRAM
  - Concurrent-read, exclusive-write (CREW) PRAM
  - Exclusive-read, concurrent-write (ERCW) PRAM
  - Concurrent-read, concurrent-write (CRCW) PRAM

Interconnection Networks for Parallel Computers

- Interconnection networks provide **mechanisms for data transfer between processing nodes or between processors and memory modules**
- Interconnection networks can be classified as **static or dynamic**
- **Static networks** consist of point-to-point communication links among processing nodes and are also referred to as direct networks.
- **Dynamic networks**, on the other hand, are built using switches and communication links and are also referred to as indirect networks
- Switched map a fixed number of inputs to outputs

Network Topologies

- A variety of network topologies have been proposed and implemented
- These topologies trade off performance for cost
- Commercial machines often implements hybrid of multiple topologies for reasons of packaging, cost, and available components

Bus-Based Networks

- Some of the simplest and earliest parallel machines used buses
- All processors access a common bus for exchanging data
- The distance between any two nodes is O(1) in a bus. The bus also provides a convenient broadcast media
- However, the bandwidth of the shared bus is a major bottleneck
- Typical bus based machines are limited to dozens of nodes. Sun enterprise servers and Intel Pentium based shared- bus multiprocessors are examples of such architecture

Network Topologies

Crossbar Networks

- The cost of a crossbar of p processors grows as $O(p^2)$
- This is generally difficult to scale for large values of p
- Examples of machines that employ crossbar include the Sun Ultra HPC 10000 and the Fujitsu VPP500

Multistage Networks

- Crossbars have excellent performance scalability but poor cost scalability
- Buses have excellent cost scalability, but poor performance scalability
- Multistage interconnects strike a compromise between these extremes

Network Topologies

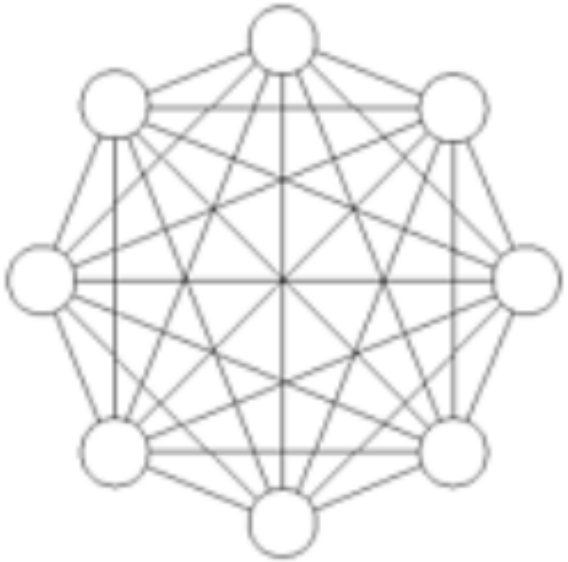
Completely connected network

- Each processor is connected to every other processor
- The number of links in the network scales as $O(p^2)$
- While the performance scales very well the hardware complexity is not realizable for large values of p
- In this sense, these networks are static counterparts of crossbars

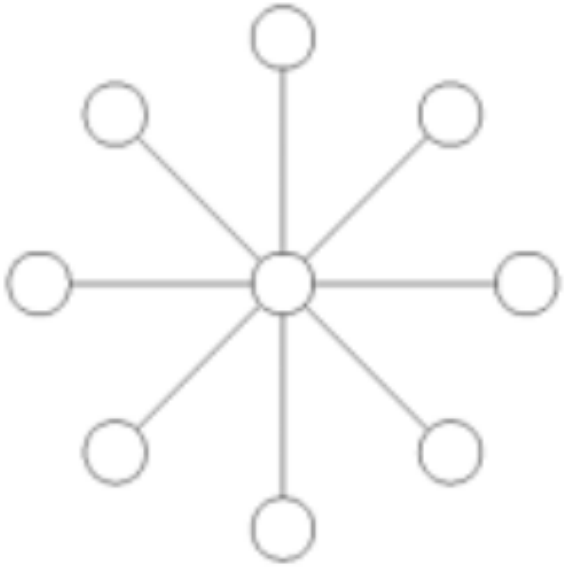
Star Connected Networks

- Every node is connected only to a common node at the centre
- Distance between any pair of nodes is O(1). However, the central node becomes a bottleneck
- In this sense, star connected networks are static counterparts of buses

Completely connected network and star connected networks



(a)

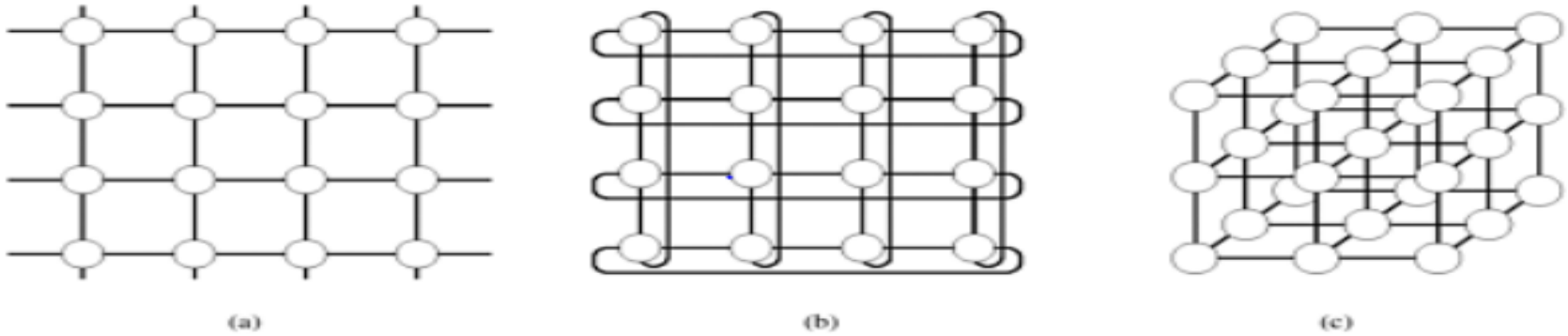


(b)

Network Topologies

Linear Arrays, Meshes, and k-d Meshes

- In a linear array, each node has two neighbours, one to its left and one to its right. If the nodes at either end are connected, we refer to it as a 1-D torus or a ring
- A generalization to 2D has nodes with 4 neighbours, to the north, south, east, and west.
- A further generalization to d dimensions has nodes with 2d neighbours
- A special case of a d-dimensional mesh is a hypercube. Here d=logp, where p is the total number nodes



Network Topologies

Tree Based network

- The distance between any two nodes is no more than 2logp
- Links higher up the tree potentially carry more traffic than those at the lower levels
- For this reason, a variant called a fat-tree, fattens the links as we go up the tree
- Trees can be laid out in 2D with no wire crossings . This is an attractive property of trees

Communication Costs in Parallel Machines

- Along with idling and contention, communication is a major overhead in parallel programs
- The cost of communication is dependent on a variety of features including the programming model semantics, the network topology, data handling and routing, and associated software protocols

Message Passing Costs in Parallel Machines

The total time to transfer a message over a network comprises of the following

- Startup time (ts): The startup time is the time required to handle a message at the sending and receiving nodes
- Per-hop time (th): After a message leaves a node, it takes a finite amount of time to reach the next node in its path
- Per-word transfer time (tw): If the channel bandwidth is r words per second, then each word takes time tw = 1/r to traverse the link

Communication Costs in Parallel Machines

Store and forward routing :- In store-and-forward routing, when a message is traversing a path with multiple links, each intermediate node on the path forwards the message to the next node after it has received and stored the entire message

Packet Routing :- Packet routing breaks messages into packets and pipelines them through the network. Since packets may take different paths, each packet must carry routing information, error checking , sequencing, and other related header information

Cut- Through routing :- Takes the concept of packet routing to an extreme by further dividing messages into basic units called flits. Since flits are typically small, the header information must be minimized. This is done by forcing all flits to take the same path, in sequence. A tracer message first programs all intermediate routers. All flits then take the same route. Error checks are performed on the entire message, as opposed to flits. No sequence numbers are needed

Impact of Process-Processor Mapping and Mapping Techniques

When mapping a graph G(V,E) into G'(V',E'), the following metrics are important:

- The maximum number of edges mapped onto any edge in E' is called the congestion of the mapping.
- The maximum number of links in E' that any edge in E mapped onto is called the dilation of the mapping.
- The ratio of the number of nodes in the set V' to that in set V is called the expansion of the mapping.

Impact of Process-Processor Mapping and Mapping Techniques