CS4379: Parallel and Concurrent Programming
CS5379: Parallel Processing

Lecture 3

Dr. Yong Chen

Associate Professor

Computer Science Department

Texas Tech University

Course Info

- **Lecture Time**: TR, 12:30-1:50
- **Lecture Location**: ECE 217
- **Sessions**: CS4379-001, CS4379-002, CS5379-001, CS5379-D01
- **Instructor:** Yong Chen, Ph.D., Associate Professor
- **Email:** yong.chen@ttu.edu
- **Phone:** 806-834-0284
- **Office**: Engineering Center 315
- **Office Hours**: 2-4 p.m. on Wed., or by appointment
- ***TA: Mr. Ghazanfar Ali, Ghazanfar.Ali@ttu.edu***
- ***TA Office hours: Tue. and Fri., 2-3 p.m., or by appointment***
- ***TA Office: EC 201 A***
- **More info:**
  - http://www.myweb.ttu.edu/yonchen
  - http://discl.cs.ttu.edu; http://cac.ttu.edu/; http://nsfcac.org

Announcements

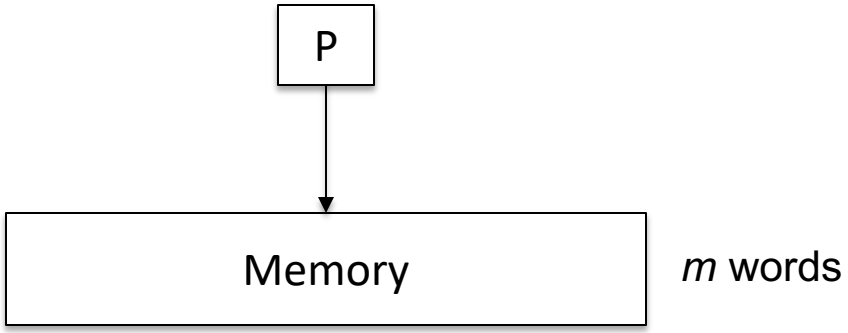
- Quiz#1 planned to be conducted on Jan. 28th, covering Lecture#3 (this lecture) and Lecture#4 (next lecture)

Outline

- Questions?
- Architecture of an ideal parallel computer
- Interconnection networks for parallel computer

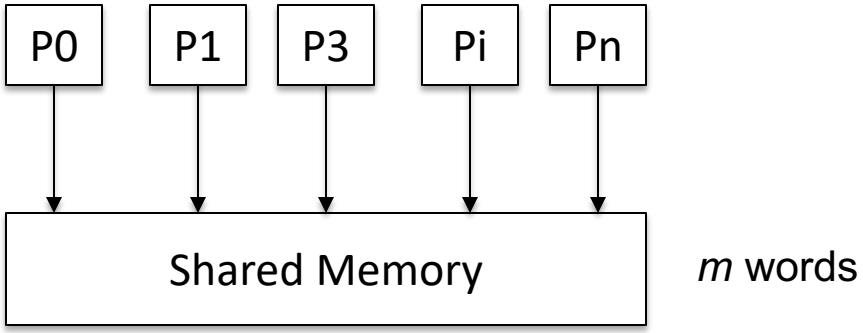
Random Access Machine (RAM) model

- Abstract machine
- Primarily used for computational complexity analysis
- Random access each word in memory
- Closest to the common notion of computer (v.s. Turing machine)



Architecture of an Ideal Parallel Computer

- A natural extension of the Random Access Machine (RAM) serial architecture is the Parallel Random Access Machine, or PRAM
- PRAMs consist of
  - *n* processors
  - a global memory of unbounded size
  - uniformly accessible to all processors
- Processors share a common clock but may execute different instructions in each cycle.

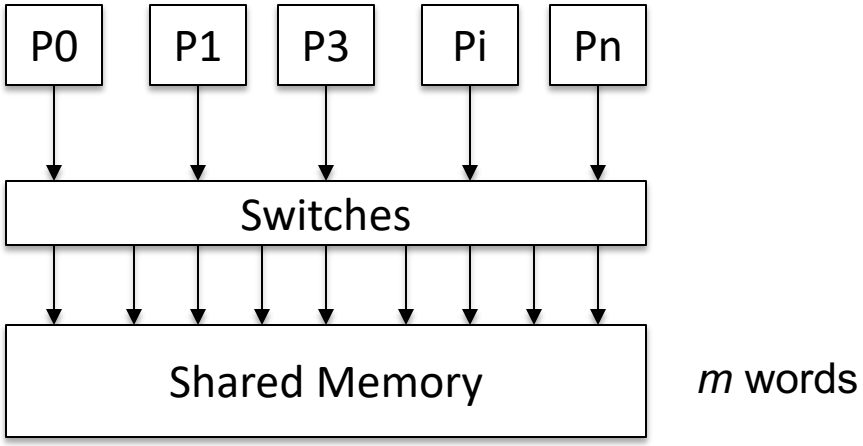


Architecture of an Ideal Parallel Computer

- Depending on how simultaneous memory accesses are handled, PRAMs can be divided into four subclasses
  - Exclusive-read, exclusive-write (EREW) PRAM
    - Minimum concurrency, weakest model
  - Concurrent-read, exclusive-write (CREW) PRAM
    - Writes are serialized
  - Exclusive-read, concurrent-write (ERCW) PRAM
    - Reads are serialized
  - Concurrent-read, concurrent-write (CRCW) PRAM
    - Most powerful model
  - Note: concurrent writes need the semantics to be defined (to let programs meet expectations)

Physical Complexity of an Ideal Parallel Computer

- Processors and memories are connected via switches
- Since these switches must operate in *O(1)* time at the level of words, for a system of *n* processors and *m* words, the switch complexity is *O(mn).*
- Clearly, for meaningful values of *n* and *m*, a true PRAM is not realizable.



Outline

- Questions?
- Architecture of an ideal parallel computer
- Interconnection networks for parallel computers
  - Parallel computer architectures: processors (processing units), memory, interconnect
  - Buses
  - Crossbars
  - Multistage networks

Interconnection Networks for Parallel Computers

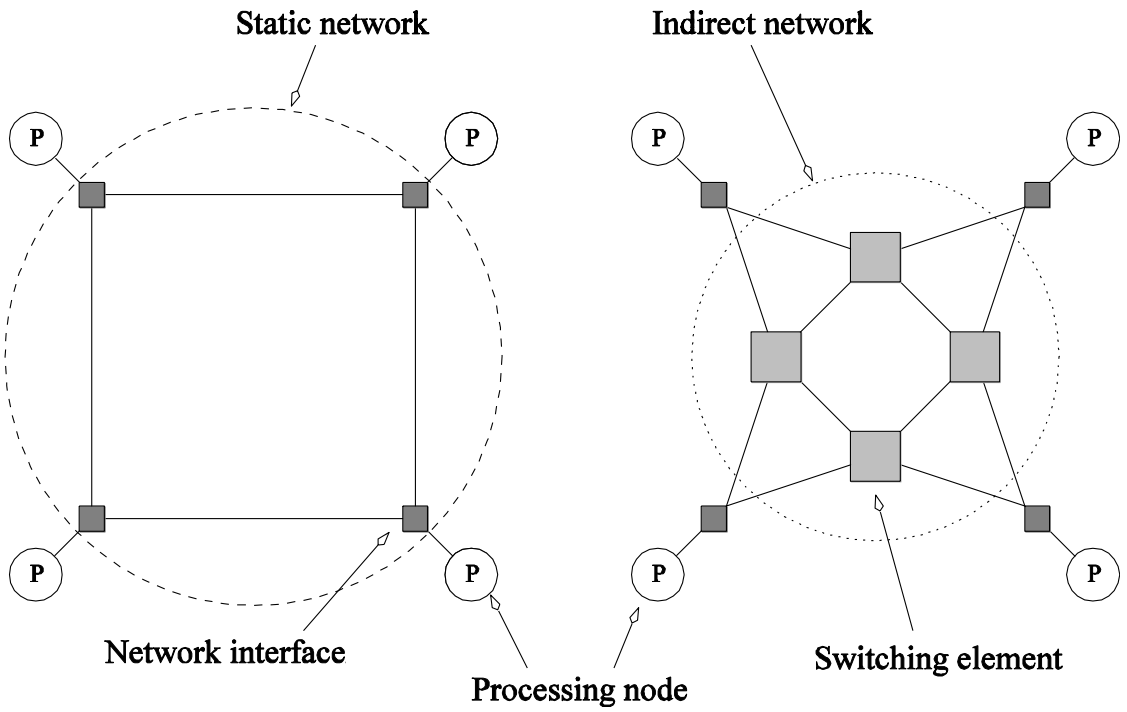
- Interconnects are made of links (wires, fiber) and switches.
- Switches map a fixed number of inputs to outputs (minimal functionality)
  - Internal buffering (when output port is busy)
  - Routing (handling congestion on network)
  - Multicast (same output on multiple ports)

- The total number of ports on a switch is the *degree* of the switch.
  - How does the cost grow in terms of the degree?
  - The cost of a switch grows as the square of the degree of the switch, the peripheral hardware linearly as the degree

Interconnection Networks for Parallel Computers (cont.)

- Interconnects can be classified as dynamic or static, depending on whether switches being used or not.
- Static networks consist of point-to-point links among processing nodes and are also referred to as *direct* networks.
- Dynamic networks are built using both switches and links. Dynamic networks are also referred to as *indirect* networks.

Static and Dynamic Interconnection Networks



Classification of interconnection networks: (a) a static network; and (b) a dynamic network.

Network Topologies

- A variety of network topologies have been proposed and implemented.
- These topologies tradeoff performance for cost.
- In practice, commercial machines often implement hybrids of multiple topologies for reasons of packaging, cost, and available components.

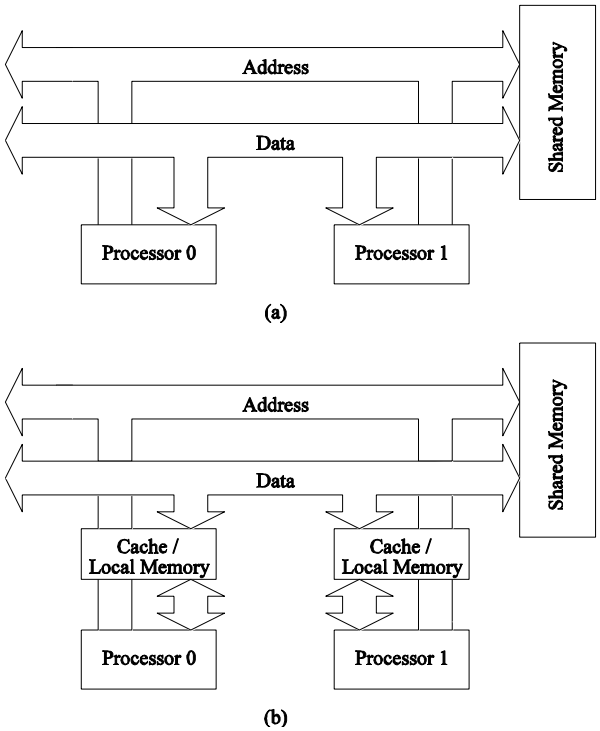
Network Topologies: Buses

- 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 architectures.

Network Topologies: Buses

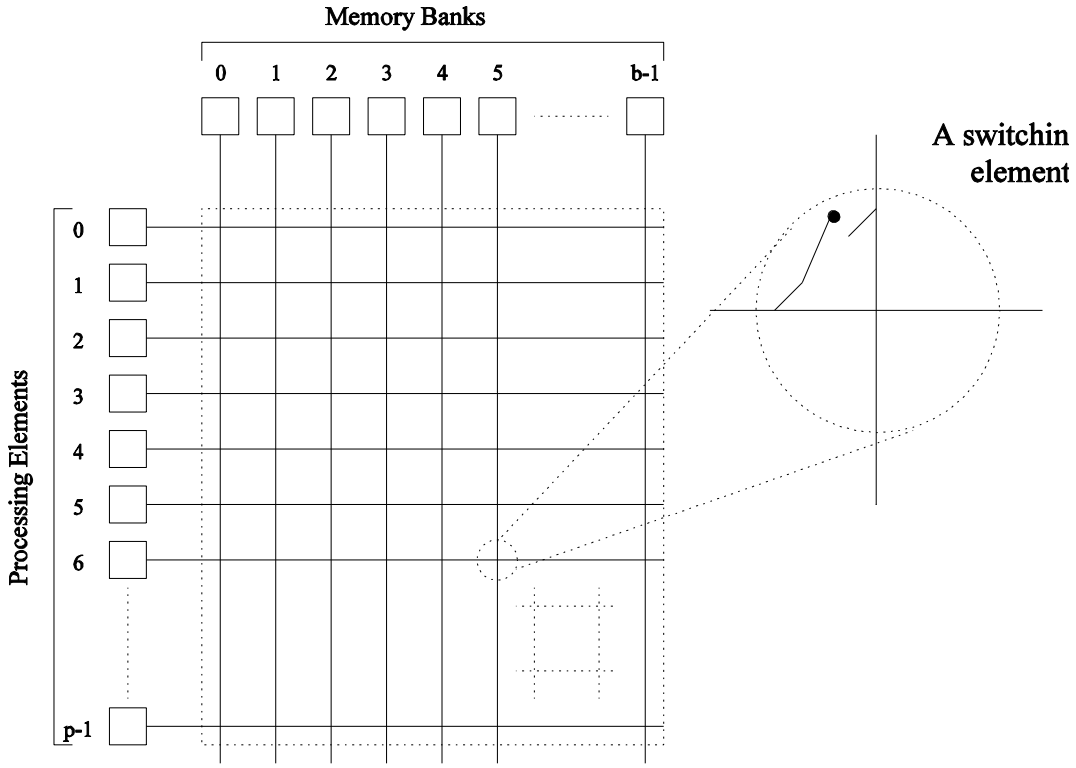


Bus-based interconnects (a) with no local caches;
(b) with local memory/caches.

Since much of the data accessed by processors is local to the processor, a local memory can improve the performance of bus-based machines.

Network Topologies: Crossbars

A crossbar network uses an $p \times b$ grid of switches to connect p inputs to b outputs in a non-blocking manner.



A completely non-blocking crossbar network connecting p processors to b memory banks.

Network Topologies: Crossbars

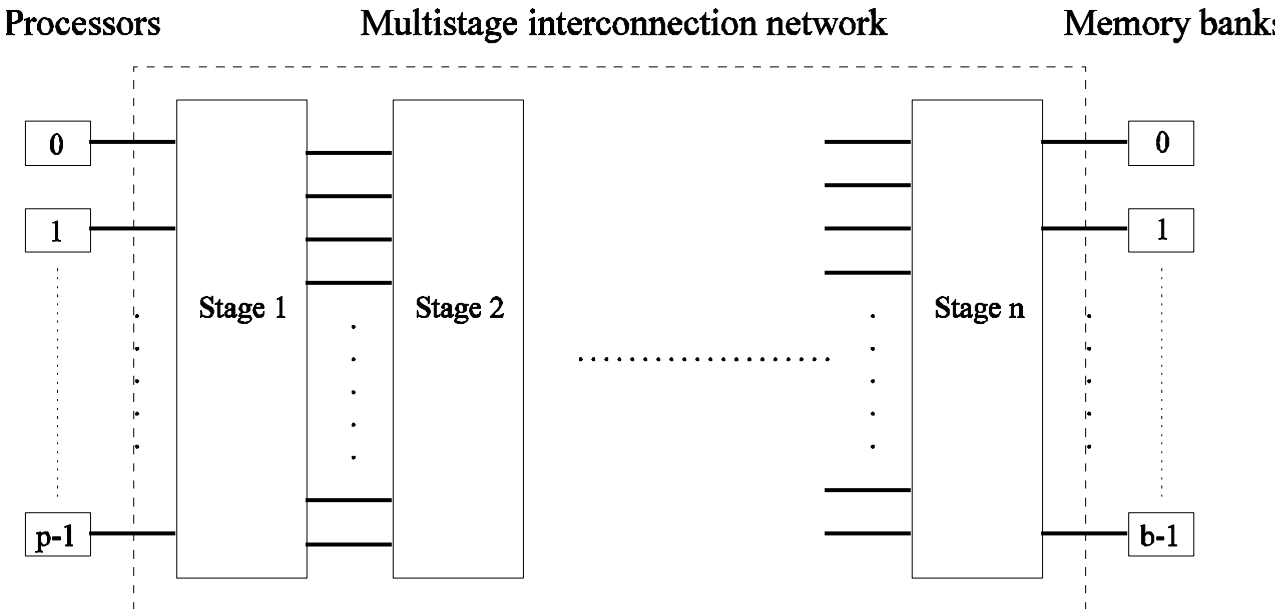
- The cost of a crossbar grows as $\Theta(pb)$, or $\Omega(p^2)$ as $b>p$ usually.
- This is generally difficult to scale for large values of p.
  - Not scalable in terms of cost

- Examples of machines that employ crossbars include the Sun Ultra HPC 10000 and the Fujitsu VPP500.

Multistage Networks

- Buses have great cost scalability, but poor performance scalability
  - Bandwidth of the shared bus is a major bottleneck
- Crossbars have great performance scalability but poor cost scalability
  - The cost of a crossbar of p processors grows as $\Omega(p^2)$
  - This is generally difficult to scale for large values of p
- Multistage interconnects strike a compromise between these extremes
  - More scalable than bus in terms of performance, and more scalable than the crossbar in terms of cost

Multistage Networks



The schematic of a typical multistage interconnection network.

Multistage Omega Network

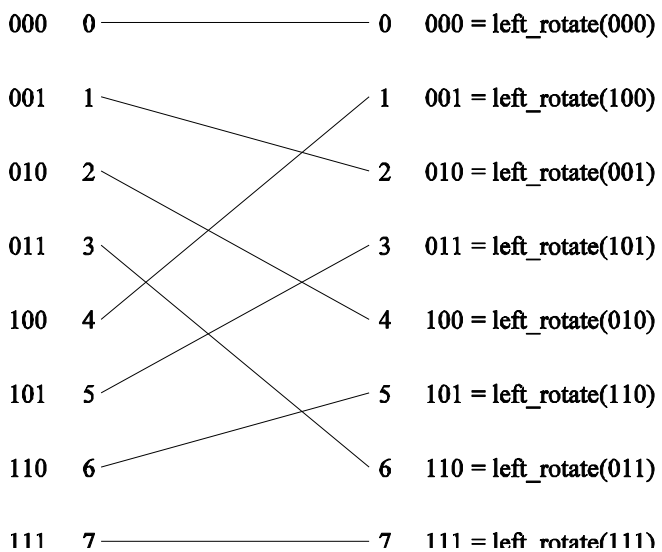
- One of the most commonly used multistage interconnects is the Omega network.
- This network consists of *log p* stages, where *p* is the number of inputs/outputs.
- At each stage,
  - *p* inputs and outputs
  - input *i* is connected to output *j* if:

$$j = \begin{cases} 2i, & 0 \leq i \leq p/2 - 1 \\ 2i + 1 - p, & p/2 \leq i \leq p - 1 \end{cases}$$

Represents a left-rotation operation on the binary representation of *i* to obtain *j*

Multistage Omega Network

Each stage of the Omega network implements a shuffle interconnection pattern as follows:



A shuffle interconnection for eight inputs and outputs.

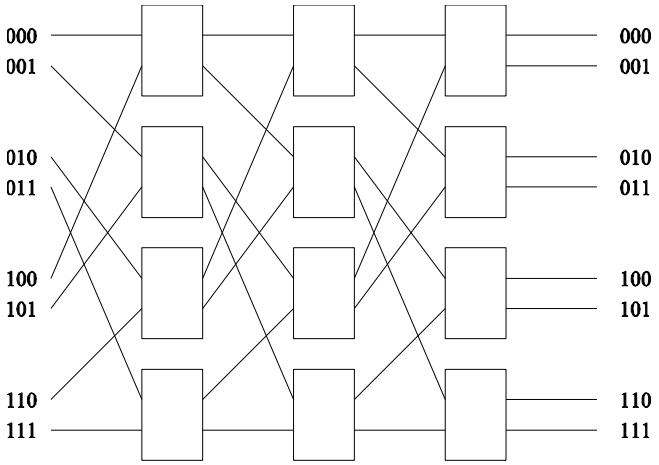
Multistage Omega Network

- The shuffle patterns are connected using 2×2 switches.
- The switches operate in two modes – passthrough or crossover.

Two switching configurations of the 2 × 2 switch:
(a) Pass-through; (b) Cross-over.

Multistage Omega Network

A complete Omega network with the perfect shuffle interconnects and switches can now be illustrated:



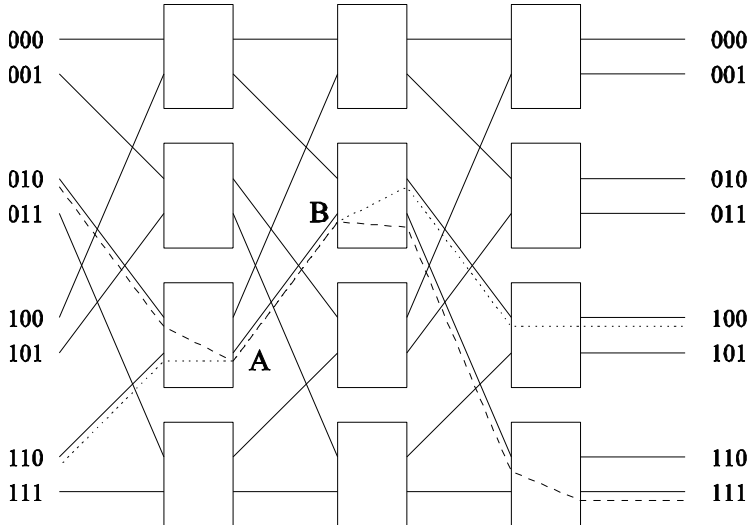
A complete omega network connecting eight inputs and eight outputs.

An omega network has $p/2 \times \log p$ switching nodes, and the cost of such a network grows as $(p \log p)$.

Multistage Omega Network – Routing

- Let *s* be the binary representation of the source and *d* be that of the destination
- The data traverses the link to the first switching node. If the most significant bits of *s* and *d* are same, then the data is routed in pass-through mode by the switch; otherwise it switches to crossover.
- This process is repeated for each of the *log p* switching stages.
- Note that this is a blocking network

Multistage Omega Network – Routing



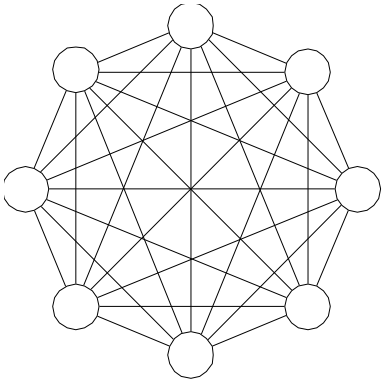
An example of blocking in omega network: one of the messages (010 to 111 or 110 to 100) is blocked at link AB.

Outline

- Questions?
- Architecture of an ideal parallel computer
- Interconnection networks for parallel computers (cont.)
  - Completely-connected and star-connected network
  - Linear arrays, meshes, and generalized meshes
  - Tree-based networks

Completely Connected Network

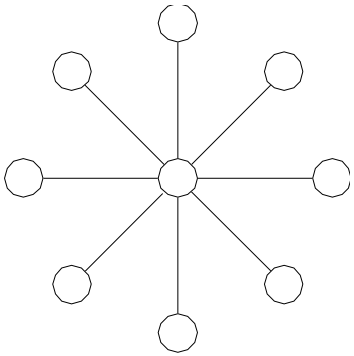
- Each node is connected to every other node.
- The number of links in the network scales as? $O(p^2)$
- While the performance scales very well, the hardware complexity is not realizable for large values of p.



A completely-connected network of eight nodes

Star Connected Network

- Every node is connected only to a common node at the center.
- Distance between any pair of nodes is *O(1)*
- However, the central node becomes a bottleneck.



A star connected network of nine nodes

Readings

- Reference book ITPC, Chapter 2
  - 2.4
  - 2.8
- Foster, DBPP, 3.7
  - https://www.mcs.anl.gov/~itf/dbpp/text/node33.html#SECTION02472000000000000000

Questions?

Questions/Suggestions/Comments are always welcome!

Write me: yong.chen@ttu.edu
Call me: 806-834-0284
See me: ENGCTR 315

If you write me an email for this class, please start the email subject with [CS4379] or [CS5379].