Parallel and Distributed Computing

Chapter 3: Models of Parallel Computers and Interconnections

Jun Zhang

Laboratory for High Performance Computing & Computer Simulation
Department of Computer Science
University of Kentucky
Lexington, KY 40506

3.1a: Architecture of Theoretical Parallel Computer

- Parallel Random Access Machine (PRAM) is a theoretical model of parallel computer, with

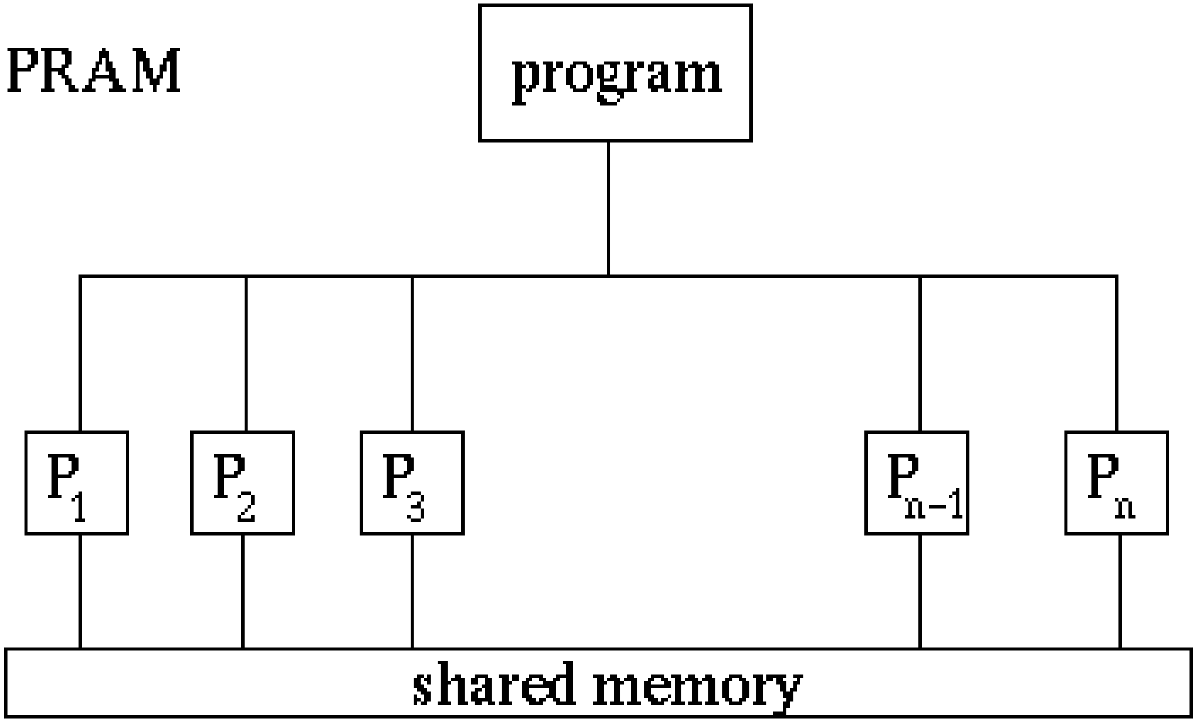
  1.) *p* identical processors

  2.) a global memory of unbounded size

  3.) memory is uniformly accessible to all processors

- Processors share a common clock
- They may execute different instructions in each cycle
- There are four subclasses of PRAM, based on the memory access protocols

3.1b: Illustration of the PRAM Model



3.1c: PRAM Subclasses

- Exclusive-read, exclusive-write (**EREW**) PRAM: Access to a memory location is exclusive. No concurrent read or write operations are allowed

  The weakest PRAM model, affording minimum concurrency in memory access
- Concurrent-read, exclusive-write (**CREW**) PRAM: Multiple read accesses to a memory location is allowed. Multiple write accesses to a memory location is serialized
- Exclusive-read, concurrent-write (**ERCW**) PRAM: Multiple write accesses are allowed to a memory location. Multiple read accesses are serialized
- Concurrent-read, concurrent-write (**CRCW**) PRAM: Both multiple read and multiple write accesses to a memory location are allowed

  This is the most powerful PRAM model

3.1d: PRAM Semantics

- Concurrent read access to a memory location by all processors is OK.
- Concurrent write access to a memory location presents semantic discrepancy and requires arbitration
- The most frequently used arbitration protocols are:
- **Common**: Concurrent write is allowed if all the writing processors have the same value
- **Arbitrary**: An arbitrary processor is allowed to proceed with the write operation, and the rest fail
- **Priority**: Processors are prioritized *a priori,* the processor with the highest priority writes and others fail
- **Sum**: The sum of all the quantities is written

3.2: Processor Granularity

- **Coarse-grained**: Few powerful processors
- **Fine-grained**: Many less powerful processors
- **Medium-grained**: between the above two
- The granularity definition is relative
- Another definition of granularity is with respect to the relative rates of communication to computation

Fine-grained: shorter duration between communication

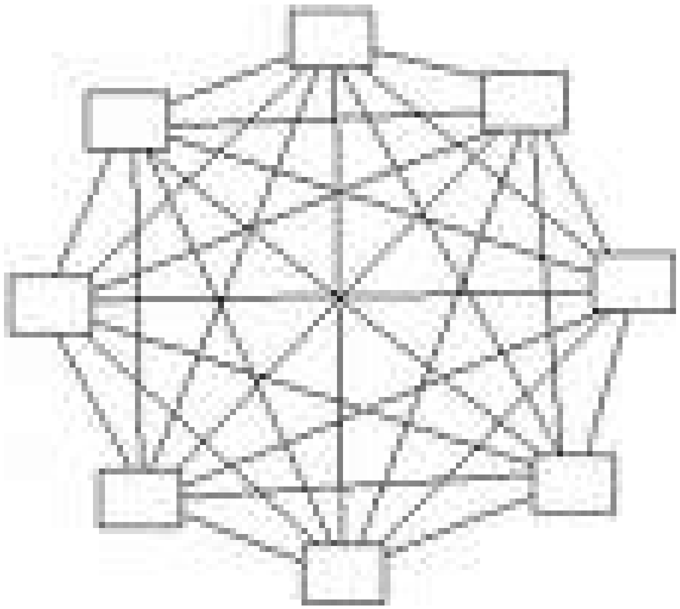
Coarse-grained: longer duration between communication

3.3a: Interconnection Networks

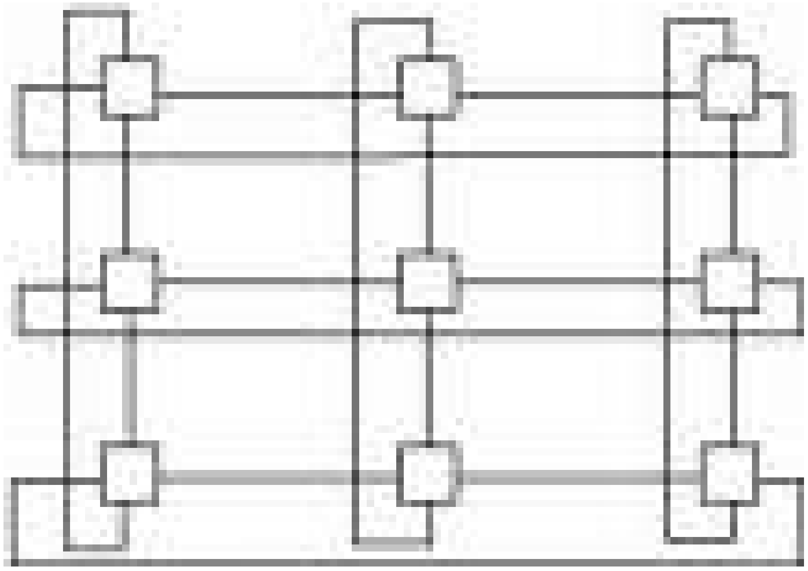
- **Static networks**: Processing nodes are connected by point-to-point communication links (direct networks)

  Mostly used for message-passing computers
- **Dynamic networks**: Communication links are connected to one another dynamically by the switches to establish paths among processing nodes and memory banks (indirect networks)
- Mostly used for shared memory computers

3.3c: Examples of Static Interconnections



Fully connected



2D mesh with wraparound

3.3d: Switch Board Connections

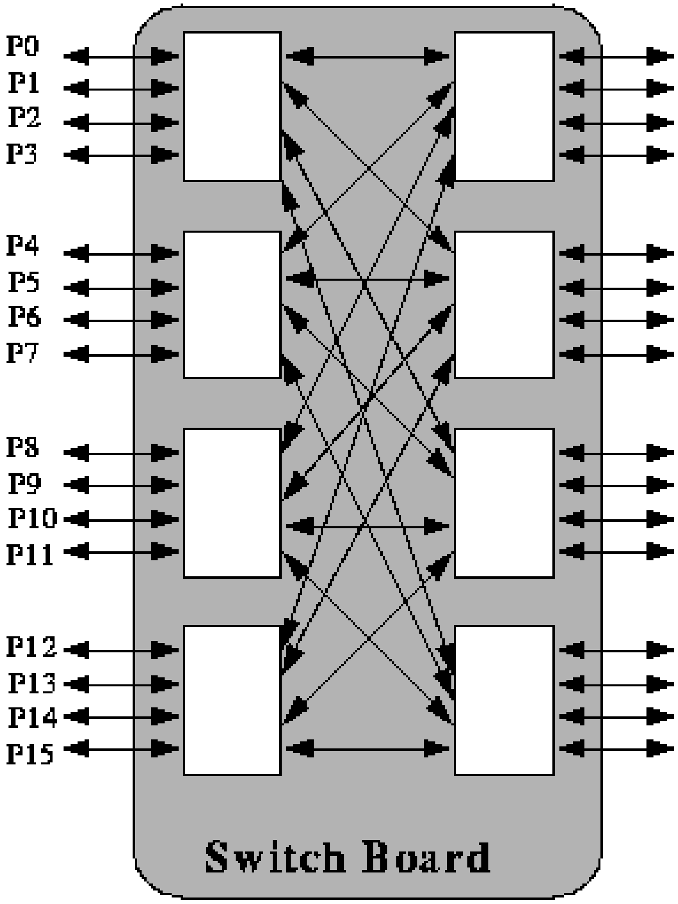
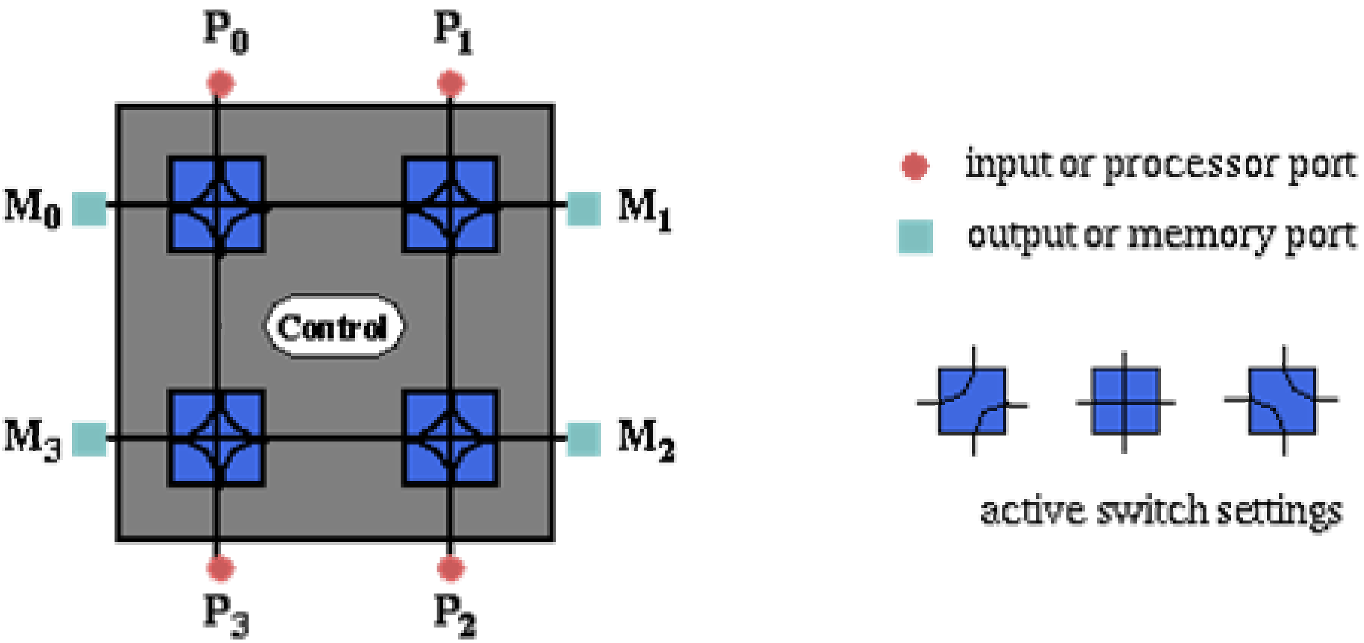
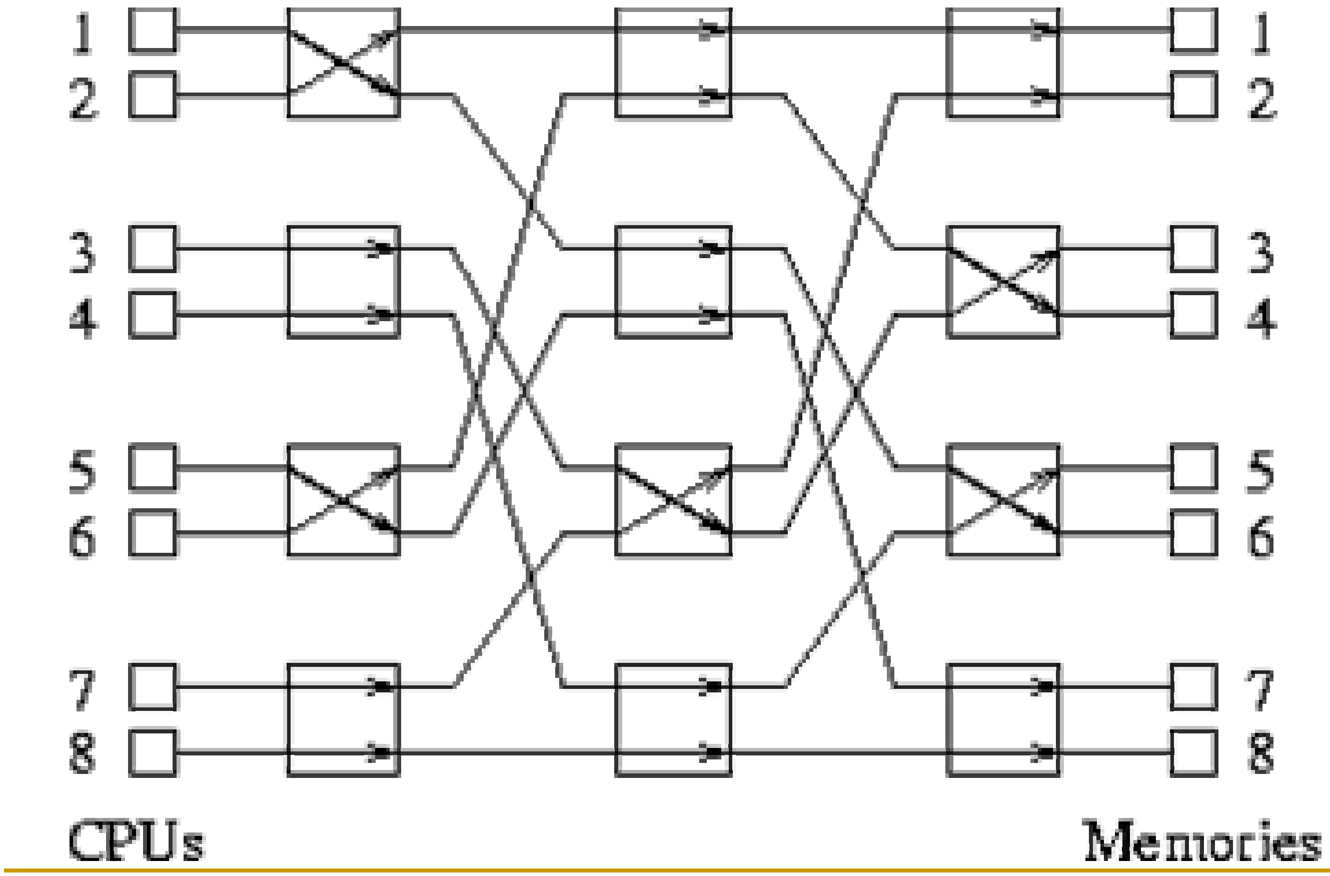


Figure 1. Switch Board Connections (16-way)

3.3e: Dynamic Interconnection



3.3f: Multistage Dynamic Interconnection



Example Configuration:

1 to 6
2 to 1
3 to 3
4 to 5
5 to 7
6 to 4
7 to 2
8 to 8

3.3g: Switch Functionalities

- A single switch has a set of input ports and a set of output ports
- The switch functionalities include:

1) a mapping from input to output ports (basic)

2) additional support for

internal buffering (when the requested output port is busy)

routing (to alleviate network congestion), and

multicasting (same output on multiple ports)

- The degree of a switch is the total number of ports

3.3h: Cost of a Switch

- The cost of a switch is influenced by the cost of mapping hardware, the peripheral hardware, and packaging costs
- The mapping hardware typically grows as the square of the degree of the switch
- The peripheral hardware grows linearly as the degree
- The packaging costs grow linearly as the number of pins

3.3h: Network Interface (Static)

- Network interface is to handle the connectivity between the node and the network
- Network interface has input and output ports that pipe data into and out of the network
- Its functionalities include:

1) packetizing data

2) computing routing information

3) buffering incoming and outgoing data

4) error checking

3.3i: Approximation of Network Costs

- For dynamic interconnection networks: Its cost is proportional to the number of switches used in the network
- For static Interconnection networks: Its cost is proportional to the number of links

3.4a: Network Topologies

- Multiple processors must be working together to solve a common task
- They must communicate during the course of solving the task
- The communication is provided by the interconnection networks
- How to connect multiple processors in a parallel system --

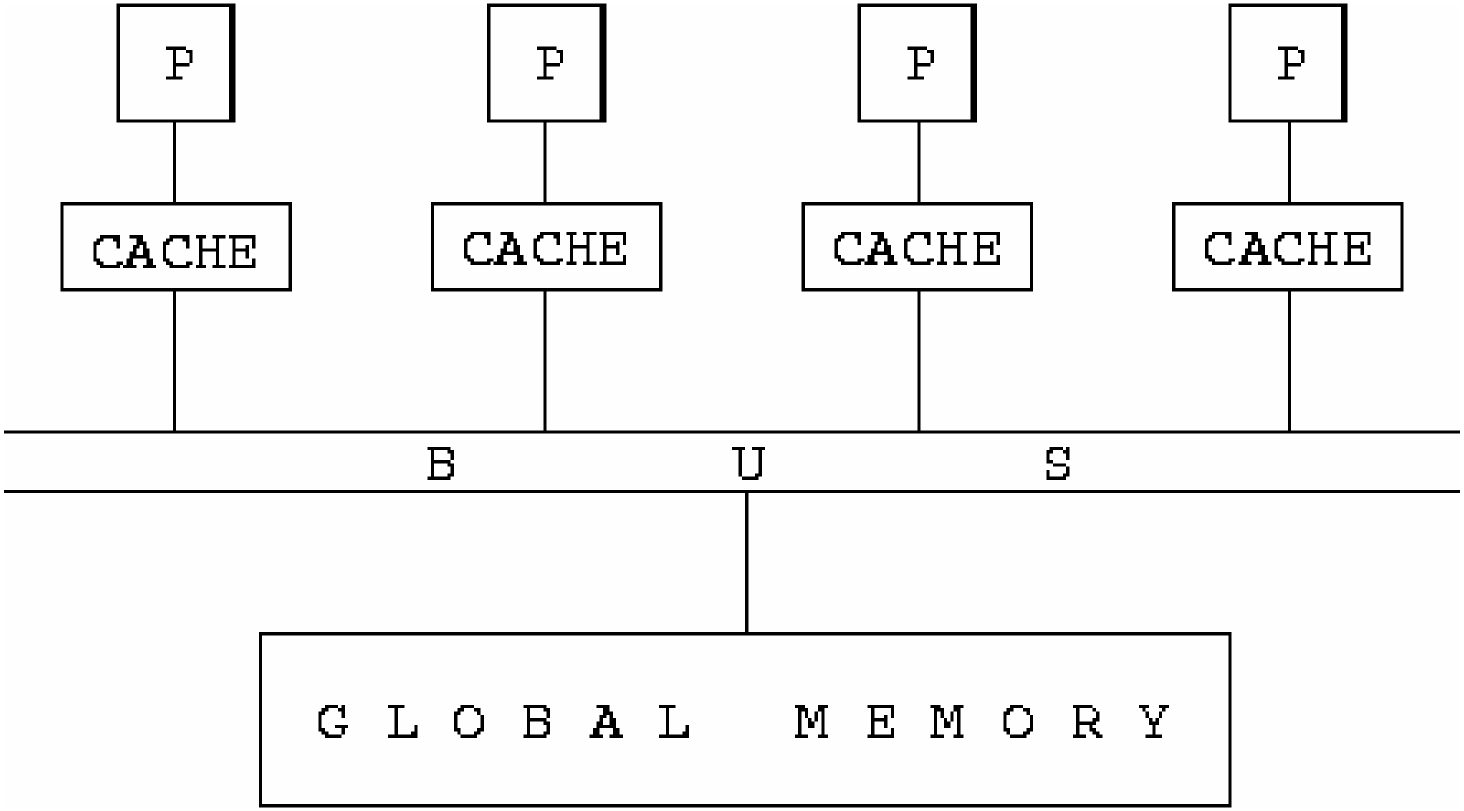
This is a trade-off between **cost** and scalability with **performance**

3.4b: Bus-Based Networks

- A bus-based network consists of a shared medium that connects all nodes
- The cost of a bus-based network scales linearly with respect to the number of processors *p*
- The distance between any two nodes in the network is constant *O*(1)
- Ideal for broadcasting information
- Disadvantage: bounded bandwidth & blocking

  Performance is not scalable with respect to the number of processors *p*

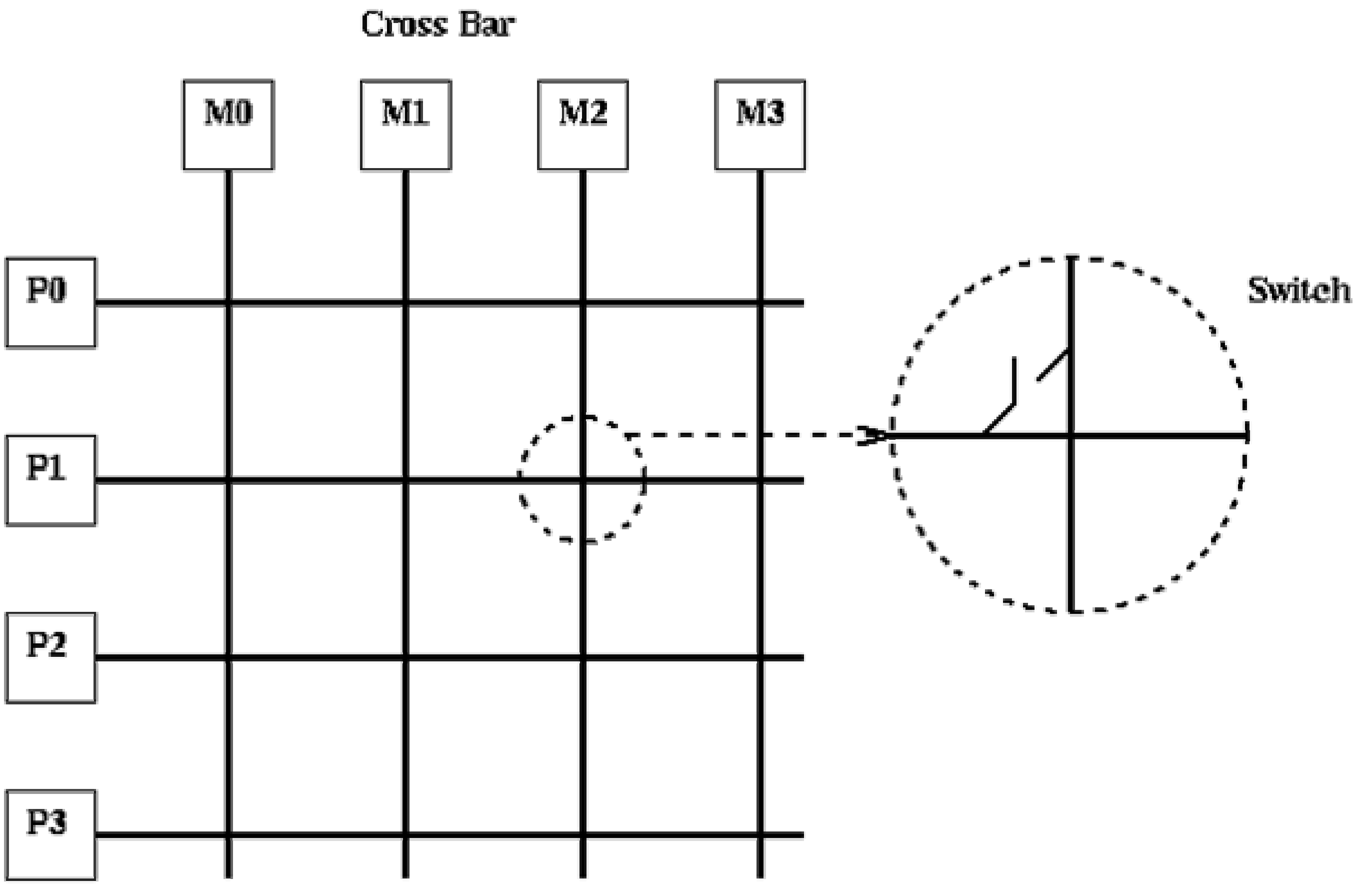
3.6b: Bus-Based Interconnect with Cache



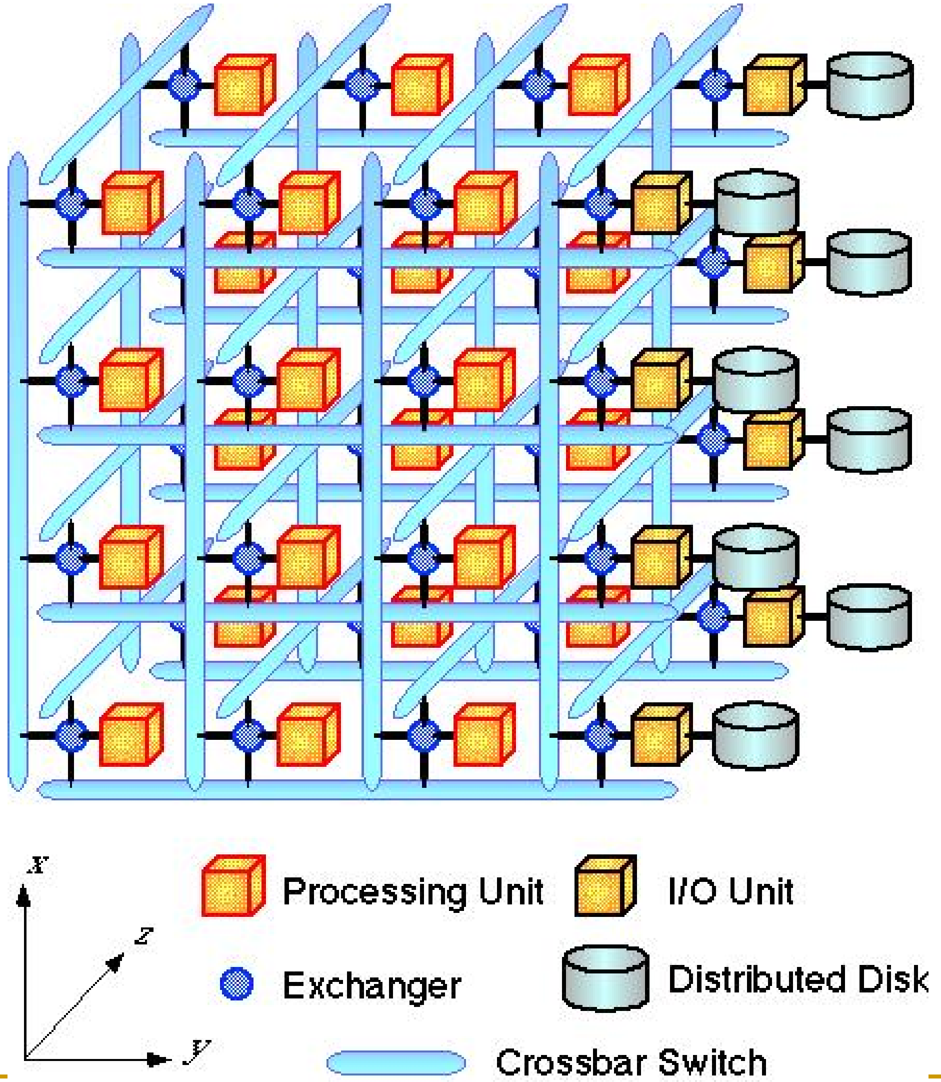
3.6c: Crossbar Network

- A crossbar network uses a grid of switches or switching nodes to connect *p* processors to *b* memory banks
- It is a non-blocking network
- The total number of switching nodes is $\Theta(pb)$
- In many cases, *b* is at least on the order of *p,* the complexity of the crossbar network is $\Omega(p*p)$
- Disadvantage: Switch complexity is difficult to realize at high data rates
- Scalable in terms of **performance**, but not scalable in terms of **cost**

3.6d: Crossbar Network (I)



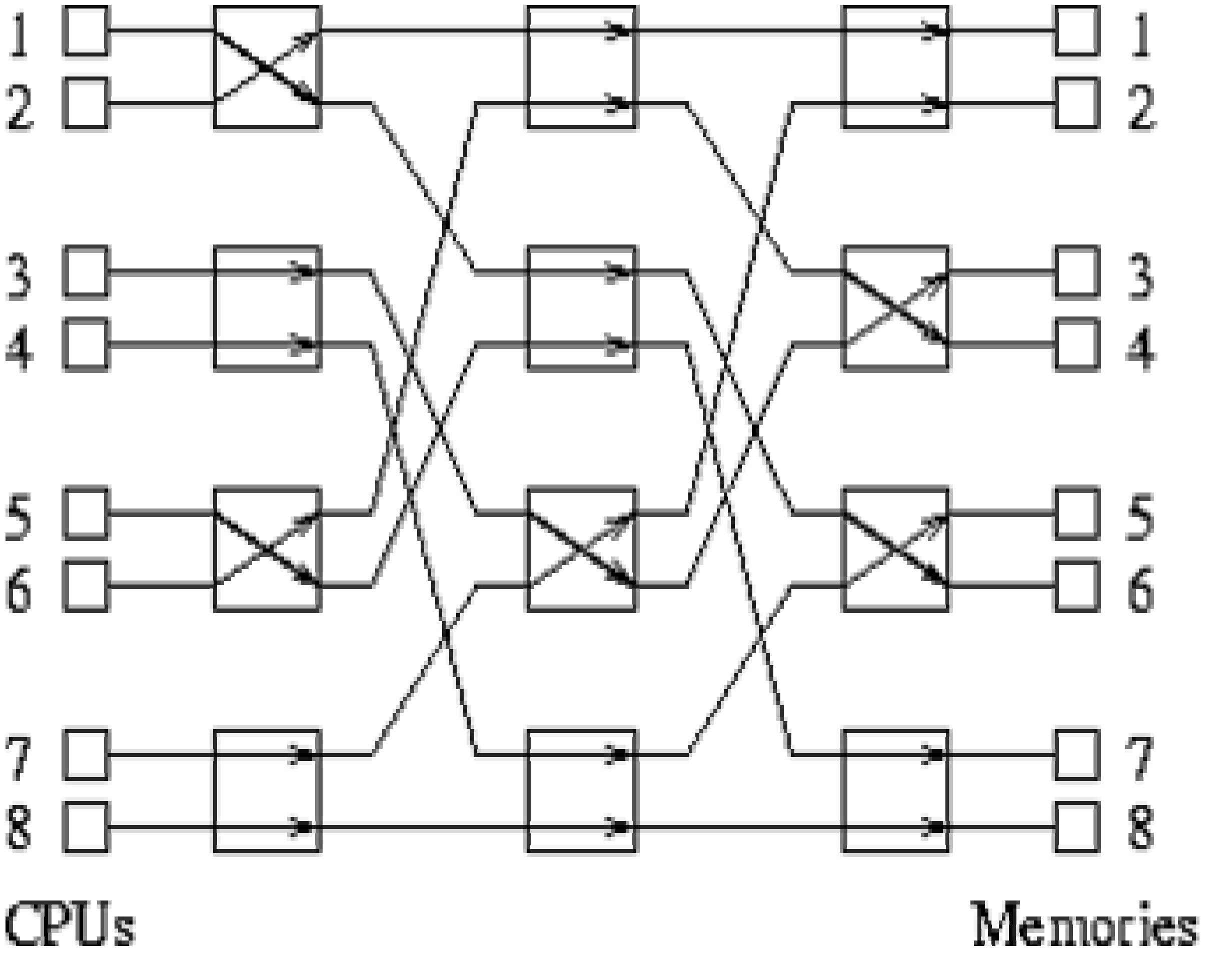
3.6e: Crossbar Network (II)



3.6f: Multistage Networks

- To balance the scalability between performance and costs
- Allowing multiple stages between processors and memory banks
- Switches are installed at each stage
- It is more scalable than bus-based networks in terms of performance
- It is more scalable than the crossbar networks in terms of costs
- A special multistage interconnection network is the **omega** network

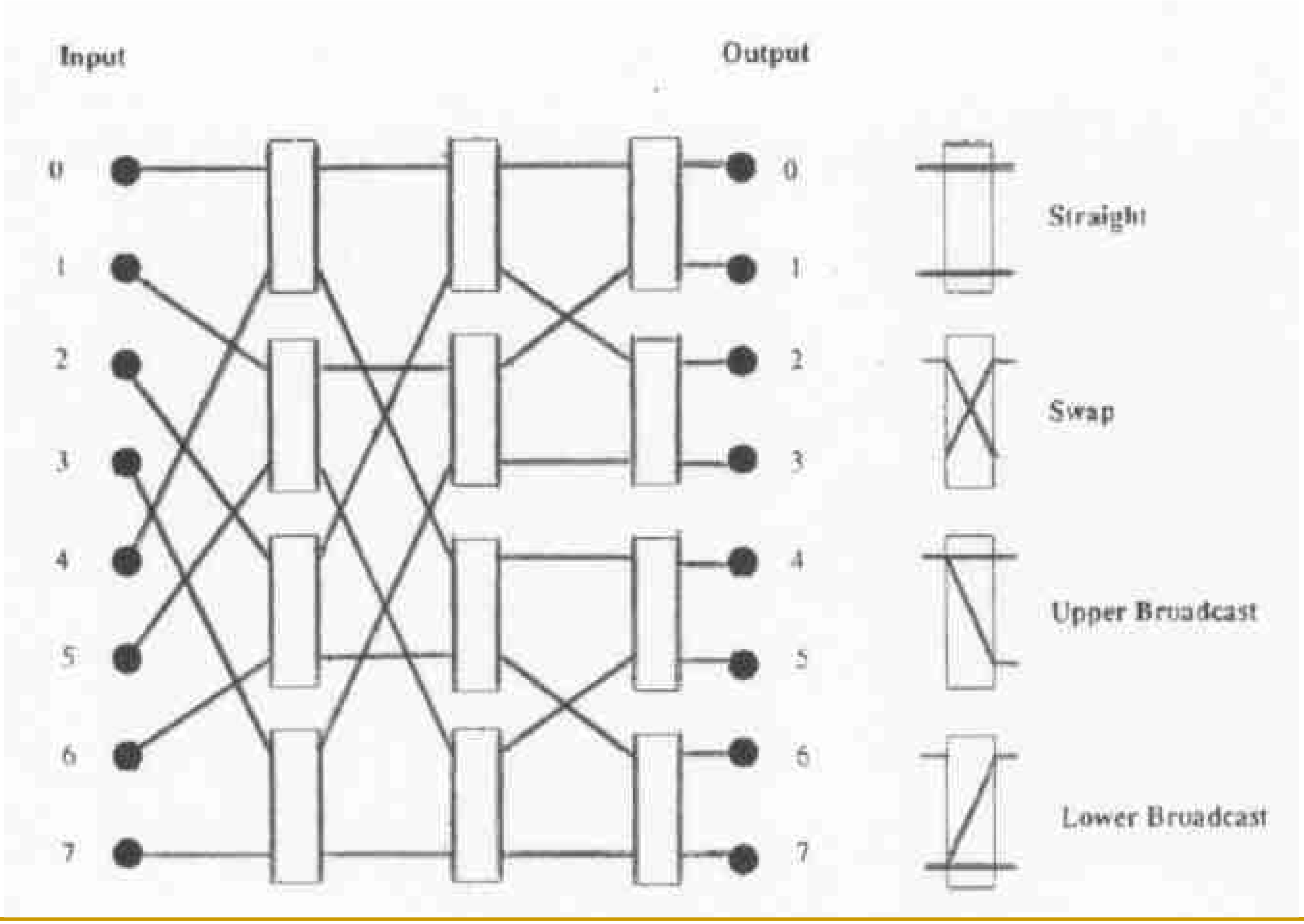
3.6g: Multistage Interconnection (I)



Example Configuration:

1 to 6
2 to 1
3 to 3
4 to 5
5 to 7
6 to 4
7 to 2
8 to 8

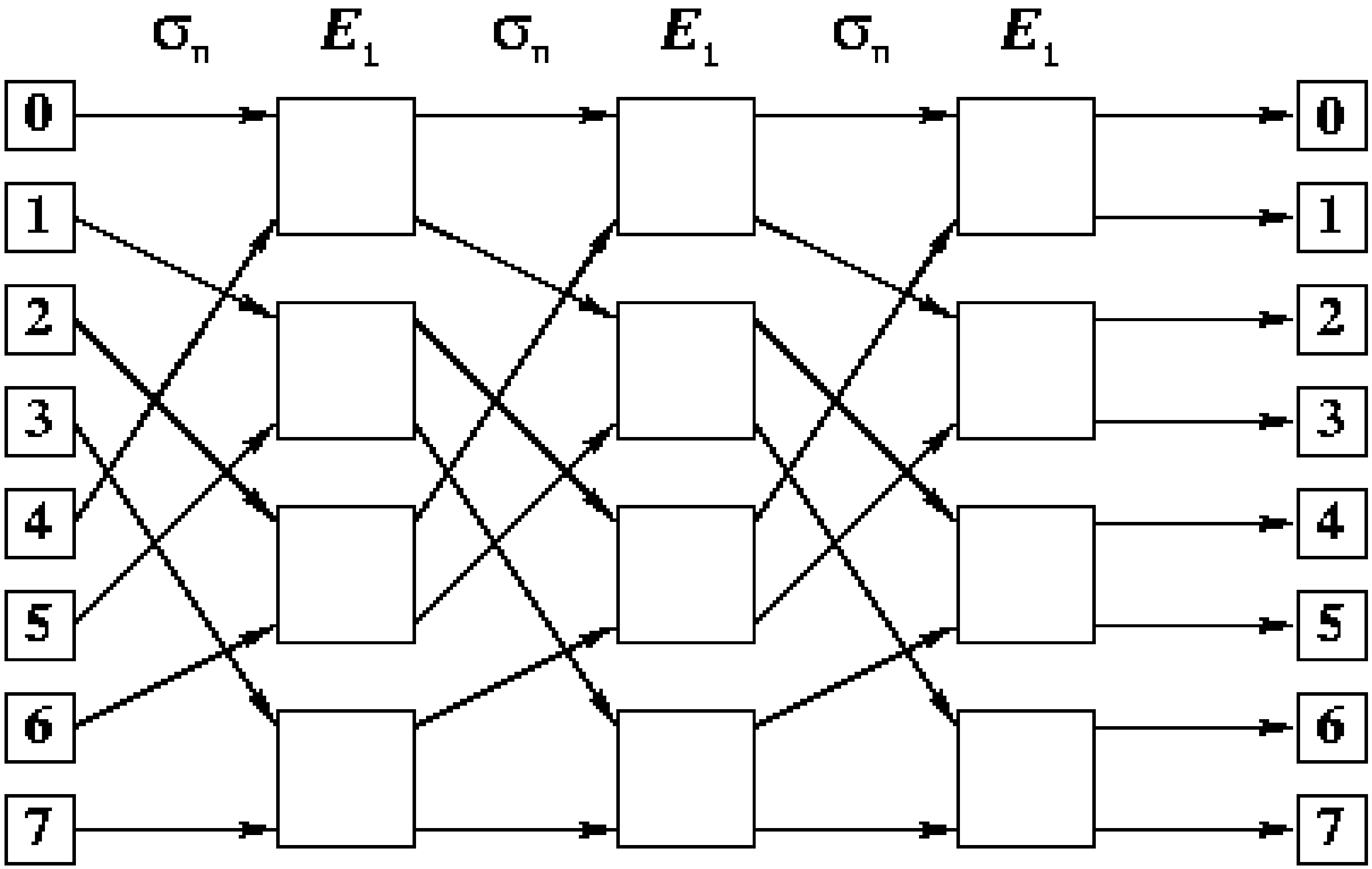
3.6h: Multistage Interconnection (II)



3.6i: Omega Network

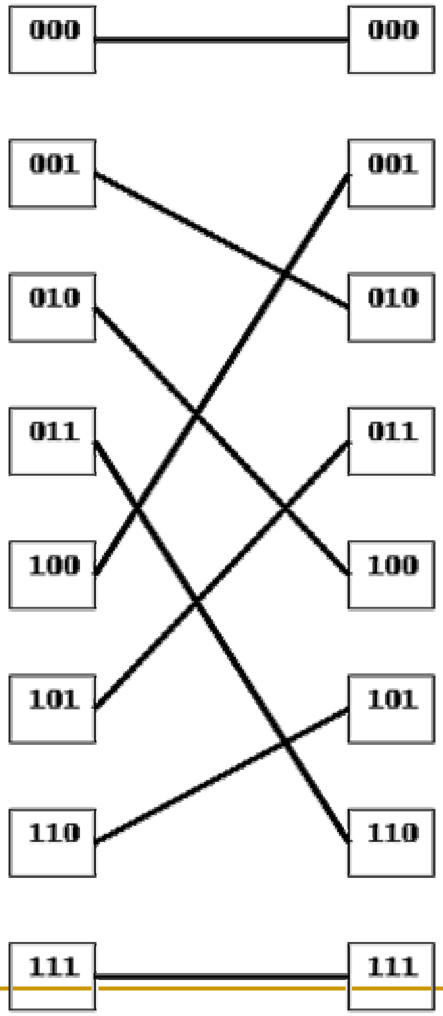
- Exactly log *p* stages
- There are *p* processors in the network
- There are *p* memory banks
- Each stage of the omega network consists of an interconnection pattern that connects *p* inputs and *p* outputs
- A link between input *i* and output *j* based on the perfect shuffle, a left-rotation on the binary representation of *i* to obtain *j*

3.6j: Illustration of Omega Network

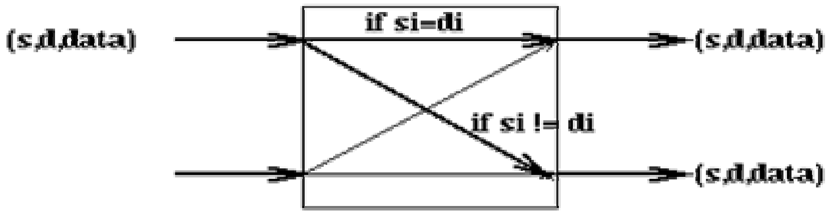


3.6k: Perfect Shuffle

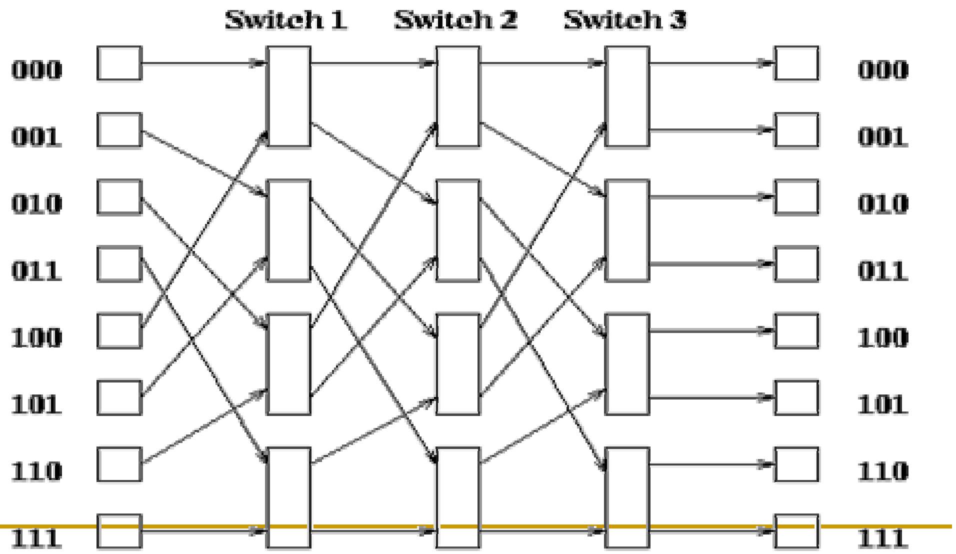
Perfect Shuffle on 8 processors



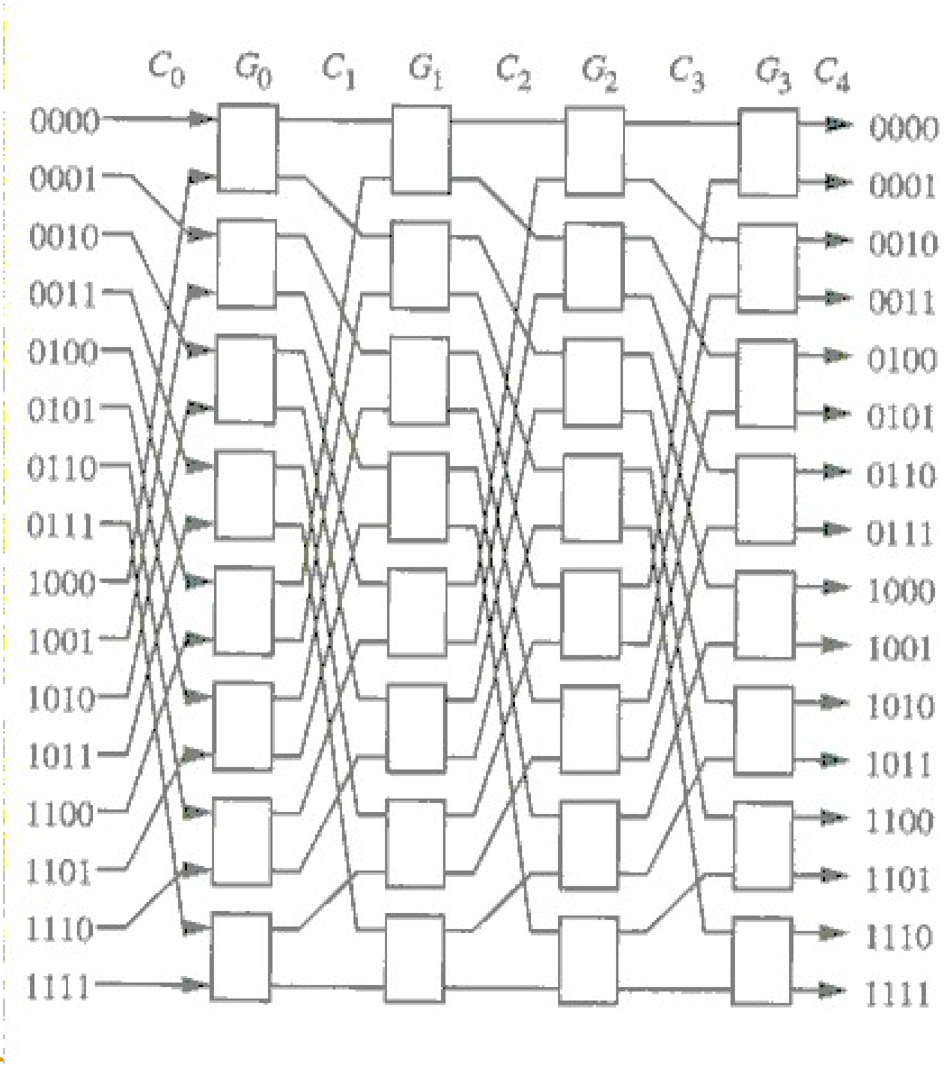
Switch i in an Omega Network



Omega network on 8 processors



3.6l: Four-Stage Omega Network



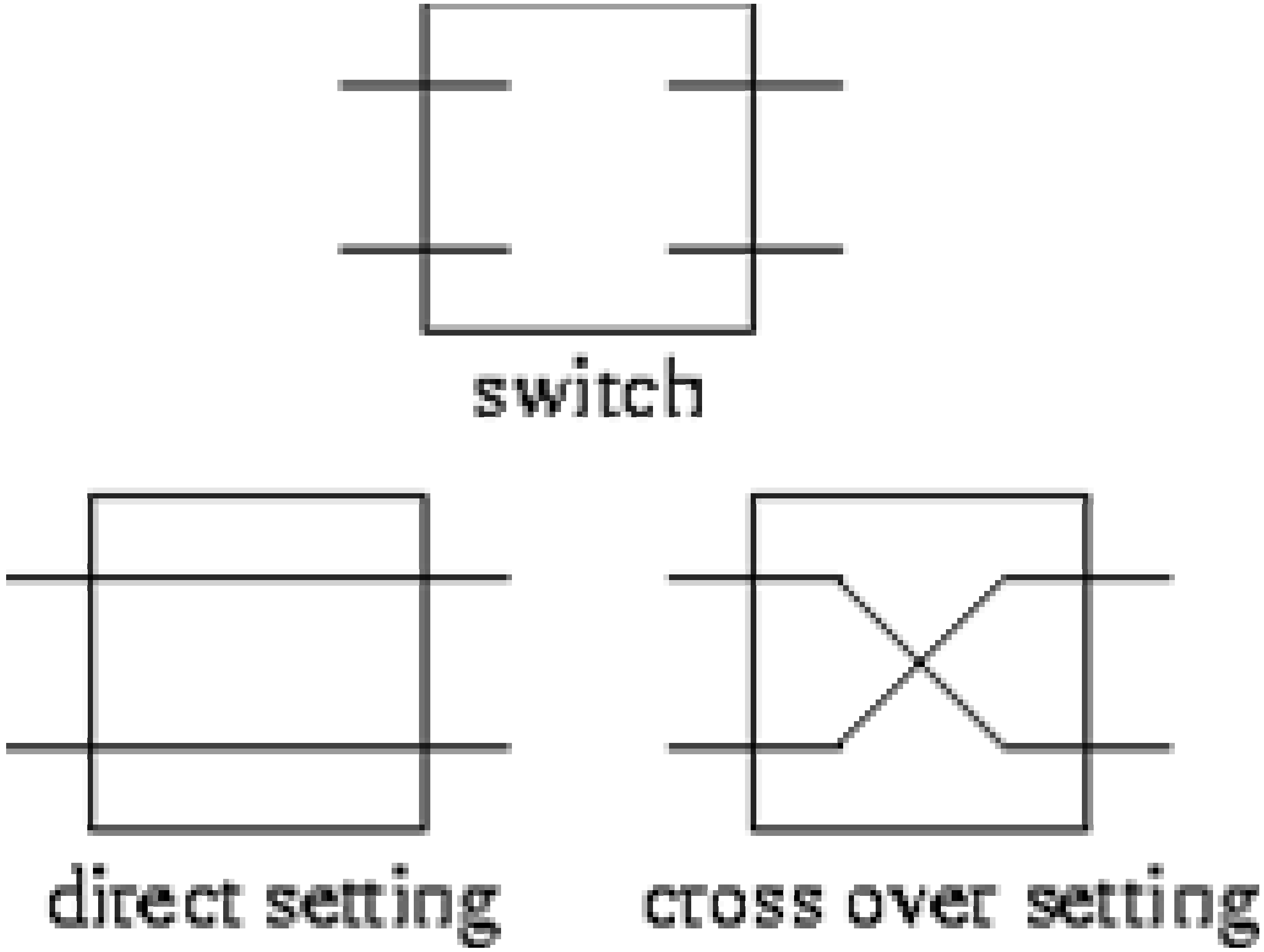
3.6m: Two Connection Modes

- At each stage, a perfect shuffle interconnection pattern feeds into a set of *p*/2 switches
- Each switch is in one of the two connection modes:

Pass-through: the inputs are sent straight through to the outputs

Cross-over: the inputs to the switching nodes are crossed over and sent out

3.6n: Switch (Pass-through and Cross-over)



3.6o: Cost and Communication of Omega Network

- The cost of an omega network is Θ(p log p)
- Data routing scheme in an omega network:

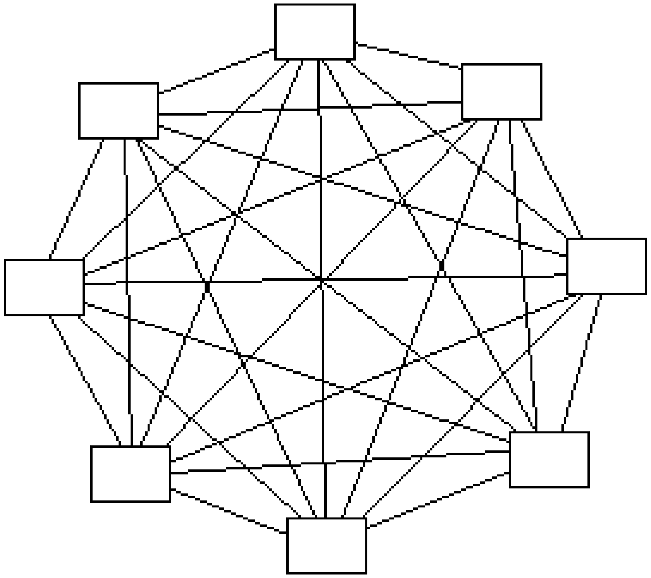
  1) Let *s* be the binary representation of the processor to communicate to a memory bank *t*

  2) If the most significant bits of *s* and *t* are the same, the data is routed in pass-through mode by the switch; otherwise in cross-over mode

  3) Traversing log *p* stages uses all log *p* bits in the binary representations of s and *t*
- This is a **blocking** network

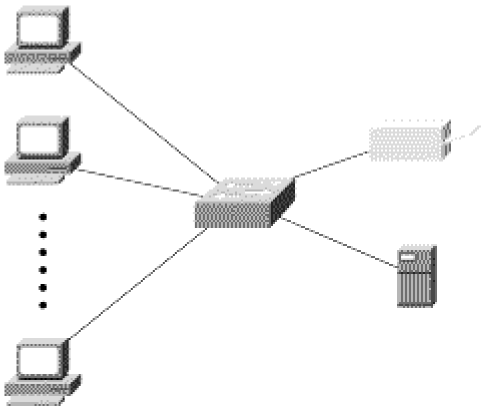
3.7: Completely-Connected Network

- Each node has a **direct** communication link to every other node in the network (non-blocking)
- How many communication links are needed?
- Scalable in terms of performance, not scalable in terms of cost

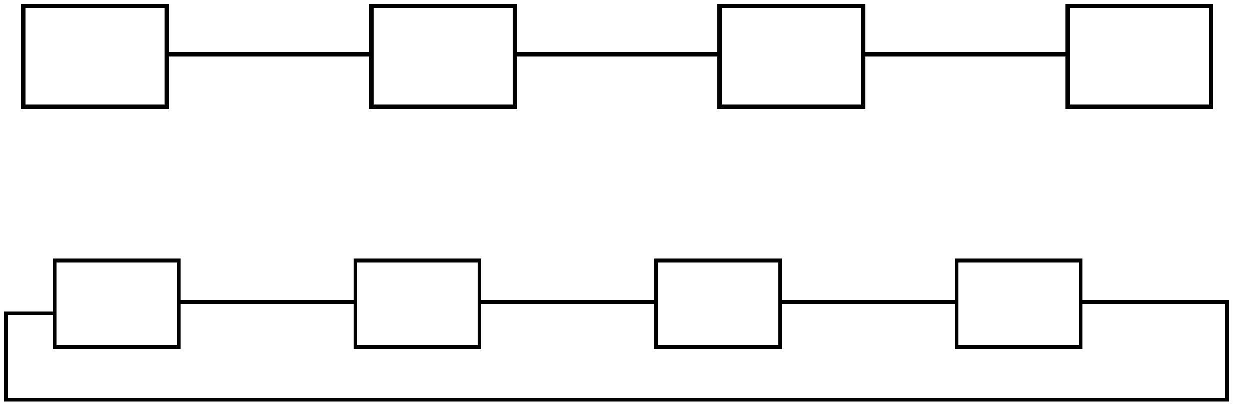


3.8: Star-Connected Network

- One processor acts as the central processor
- Every other processor has a communication link with this processor
- Congestion may happen at the central processor
- This is a blocking network
- Scalable in terms of cost, not scalable in terms of performance

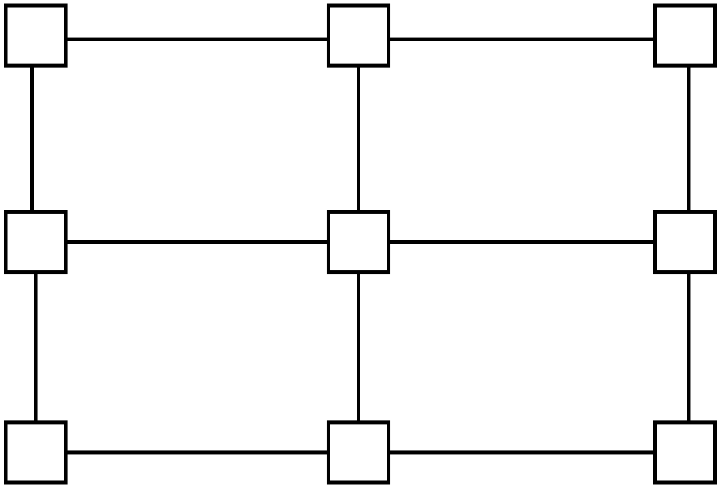


3.9: Linear Array and Ring Networks



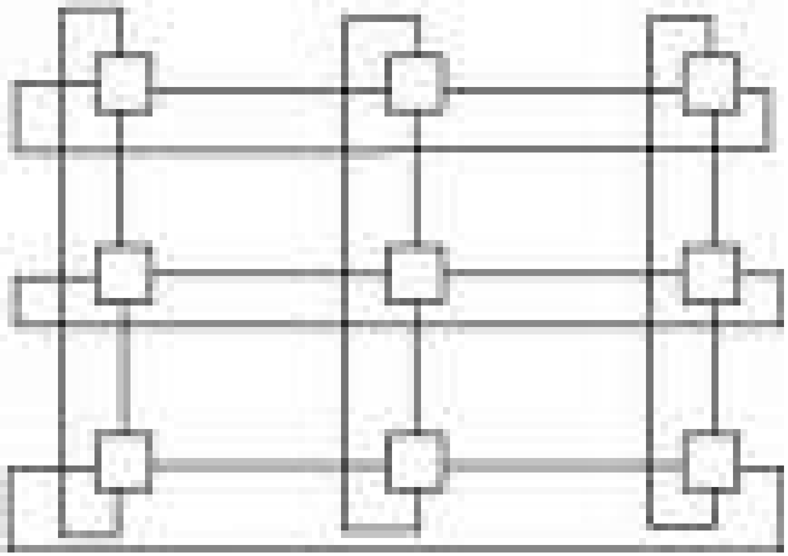
Scalable in terms of costs, not scalable in terms of performance

3.10: 2D Mesh Networks

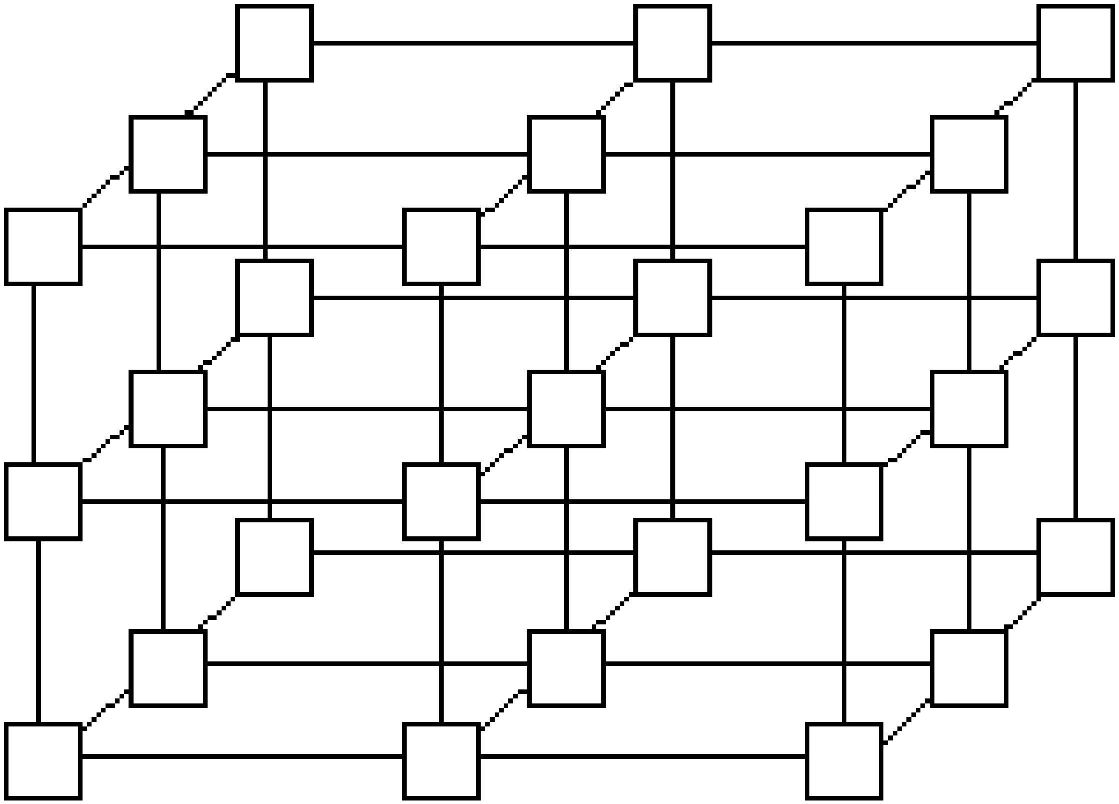


2D mesh network

2D mesh network with wraparound

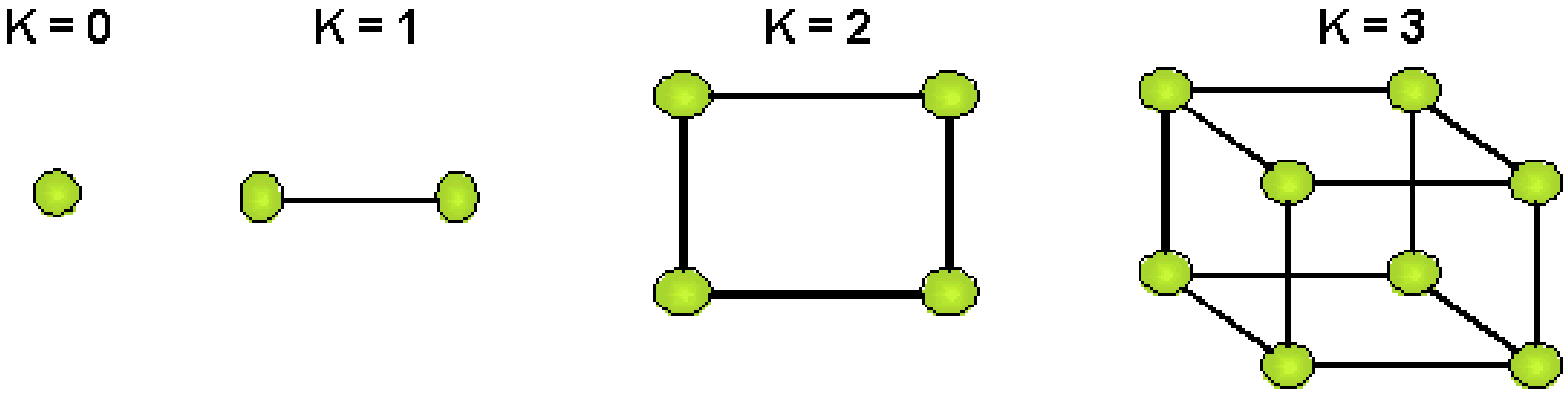


3.11: 3D Mesh Network



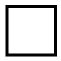
Many physical simulations can be mapped naturally to a 3D network topology. 3D mesh interconnection is common

3.12a: Hypercube Networks

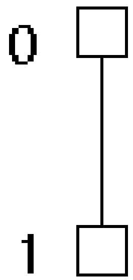


$O(\log_2 n)$ longest path
$O(\log_2 n)$ average path
$O(\log_2 n)$ connections
No bottle-neck

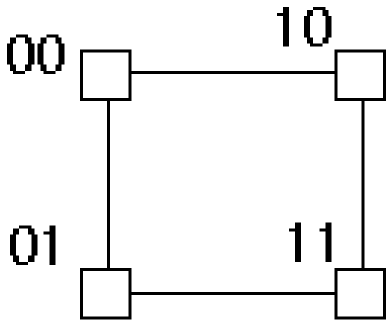
3.12b: Labeling Hypercube Networks



0-D

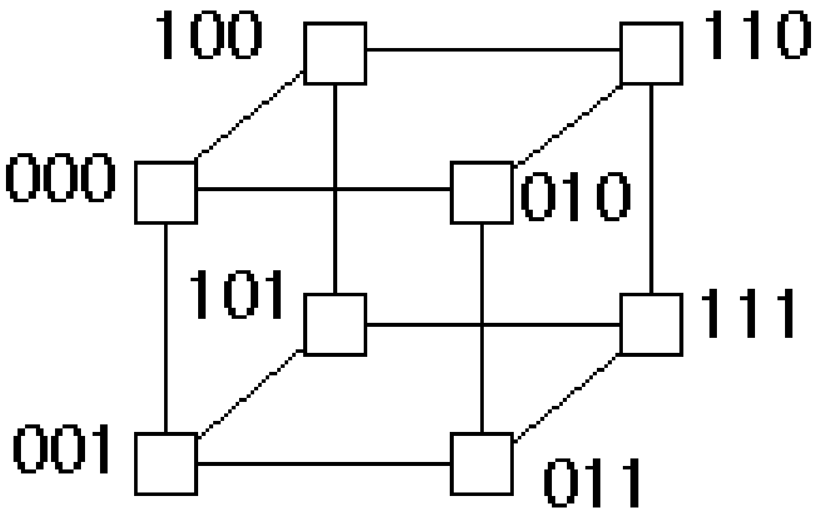


1-D



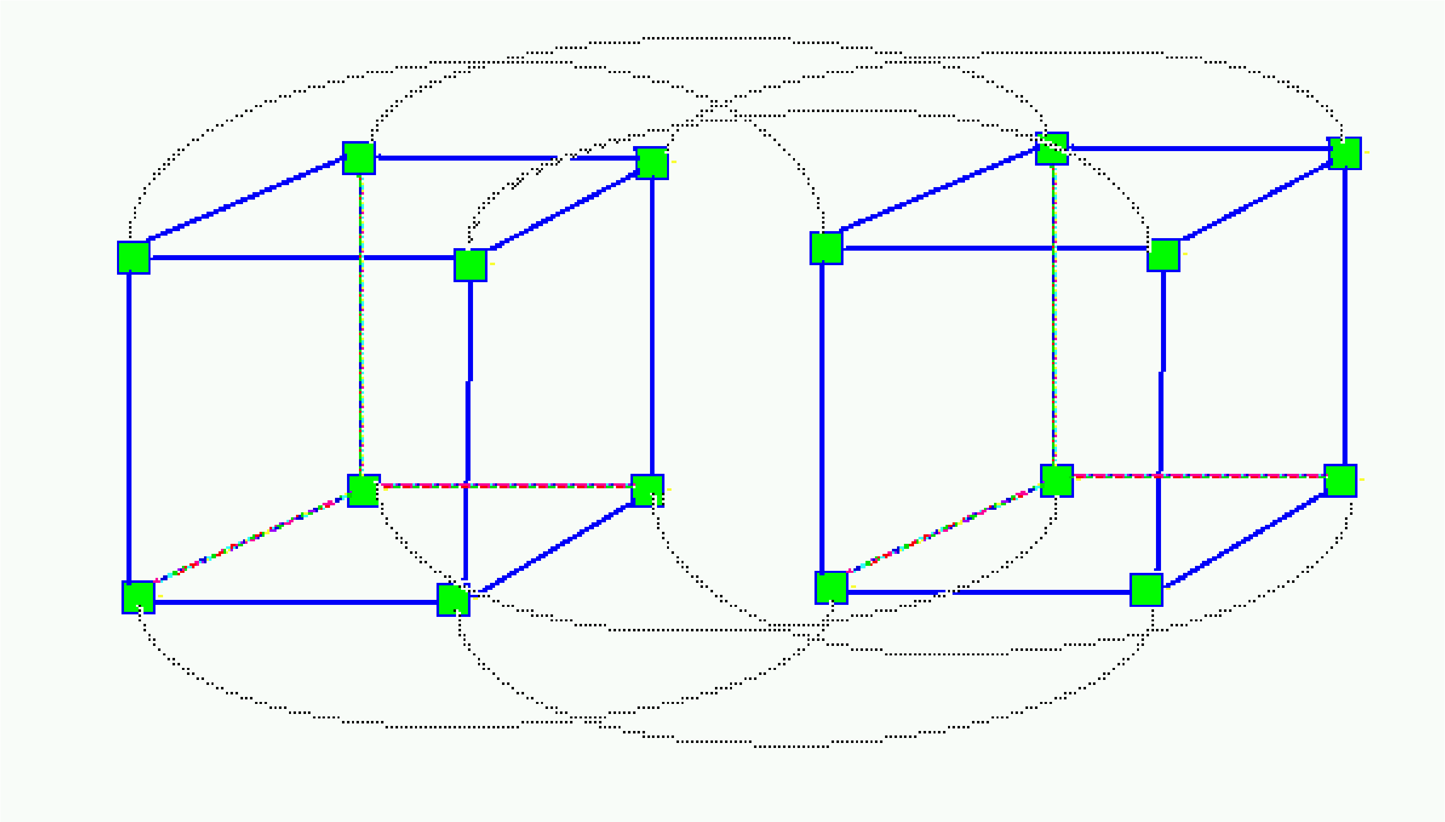
2-D

Minimum distance between 2 processors is the number of binary digits that are different in their labels: Hamming distance



3-D

3.12c: 4D Hypercube



3.12d: Partition Hypercube

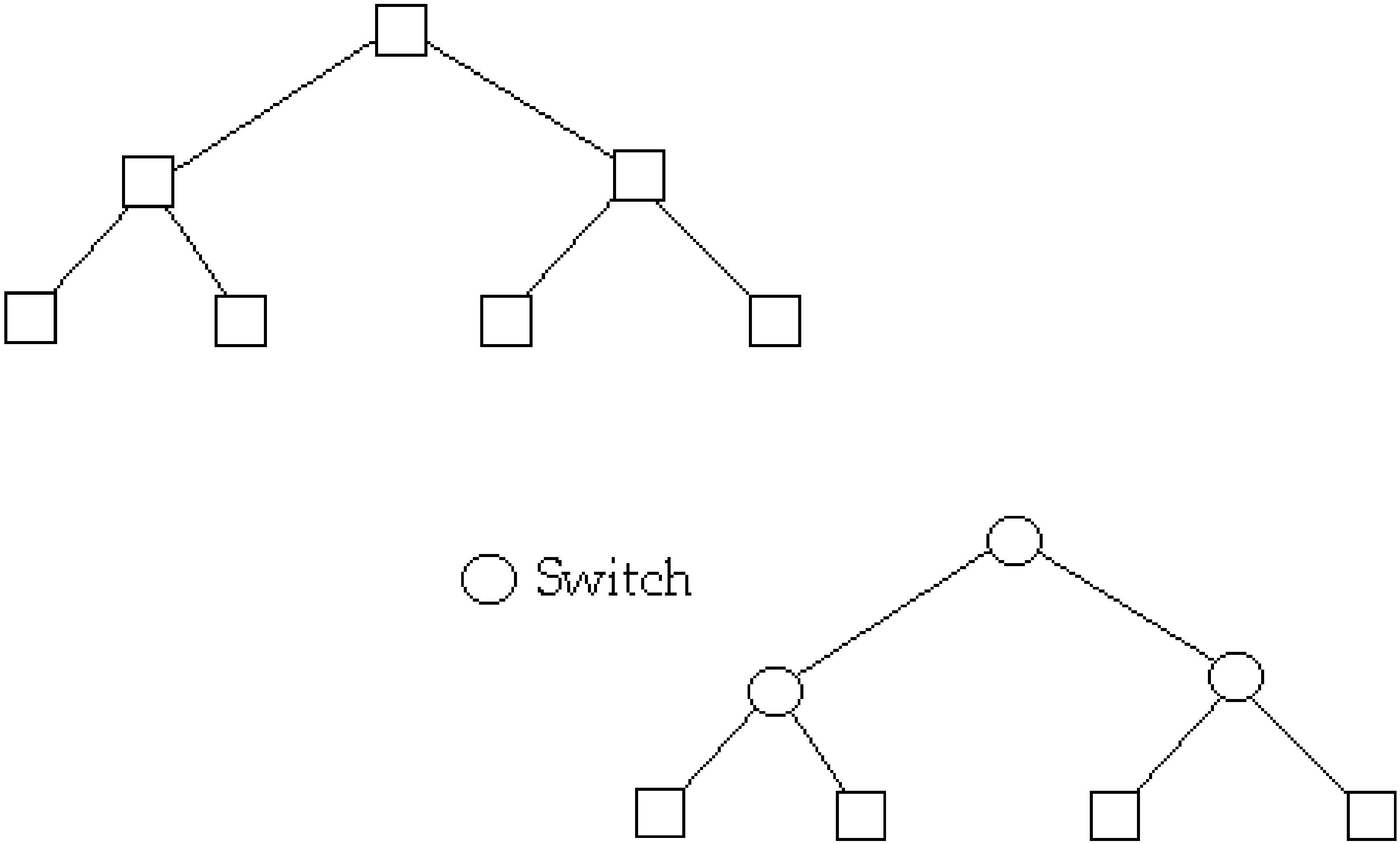
- A *d* dimensional hypercube can be partitioned into two *d-1* dimensional hypercubes
- There are *d* different partitions
- Each partition takes nodes with fixed one bit value
- Hypercubes have many theoretical properties that can be utilized to develop communication algorithms

3.13a: Tree-Based Networks

- There is only one path between any pair of nodes
- Linear array and star-connected networks are special cases of tree networks
- Tree networks can be static or dynamic
- In case of dynamic interconnection, the intermediate level processors are switching nodes and the leaf nodes are processing elements

3.13b: What is This Tree Network?

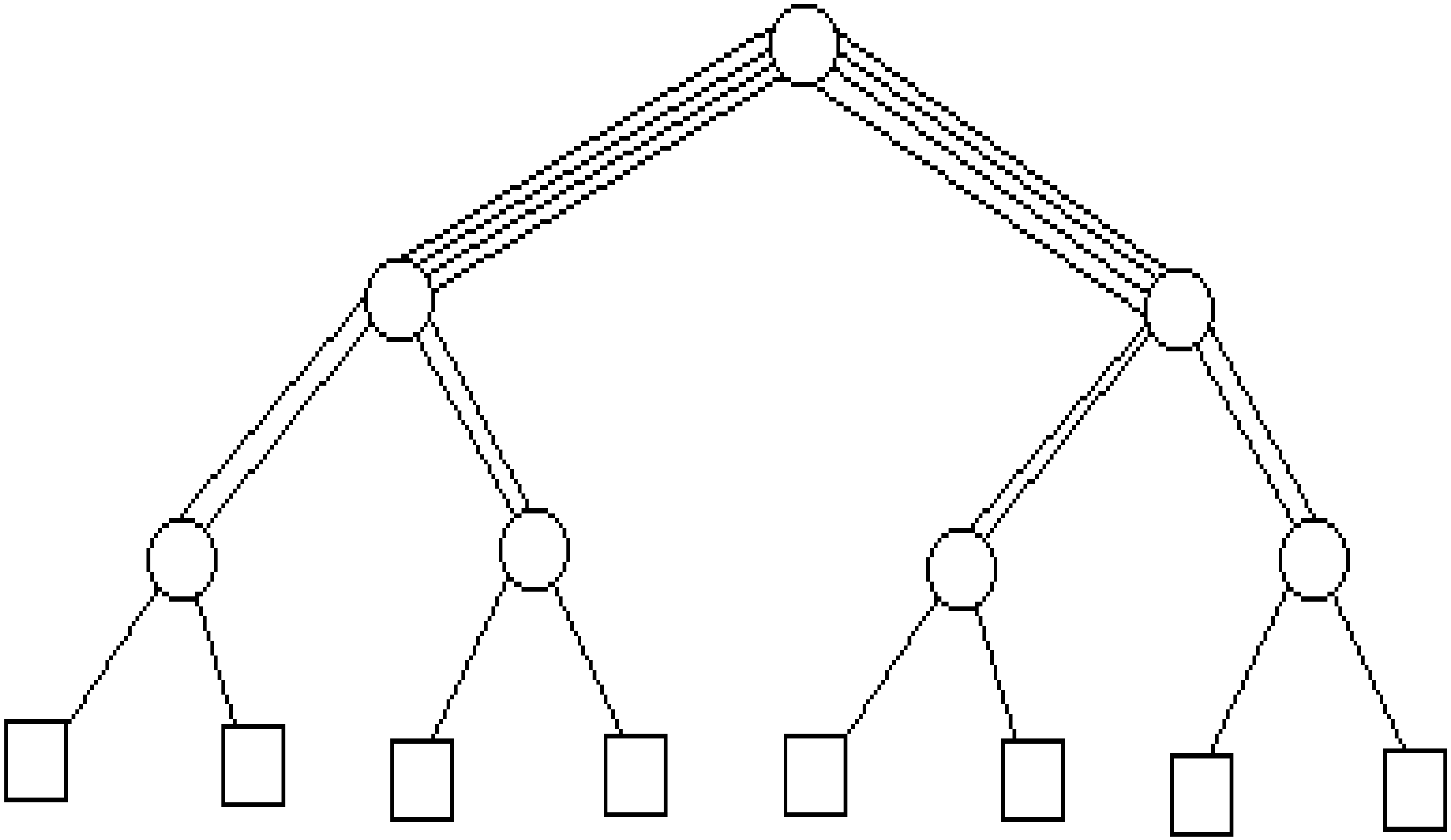
3.13c: Static and Dynamic Tree Networks



3.13d: Communication in Tree Networks

- Messages from one half tree to another half tree are routed through the top level nodes
- Communication bottleneck forms at higher levels of the trees
- The solution is to increase the number of communication links and switching nodes at the higher levels
- The fat tree is suitable for dynamic networks

3.13e: Fat Tree Network

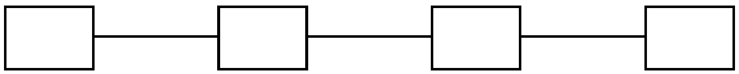


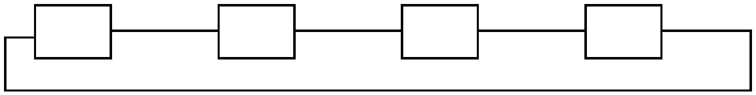
3.14a: Evaluating Static Interconnection Networks

- There are several criteria to characterize the cost and performance of static interconnection networks
- Diameter
- Connectivity
- Bisection Width
- Bisection Bandwidth
- Cost

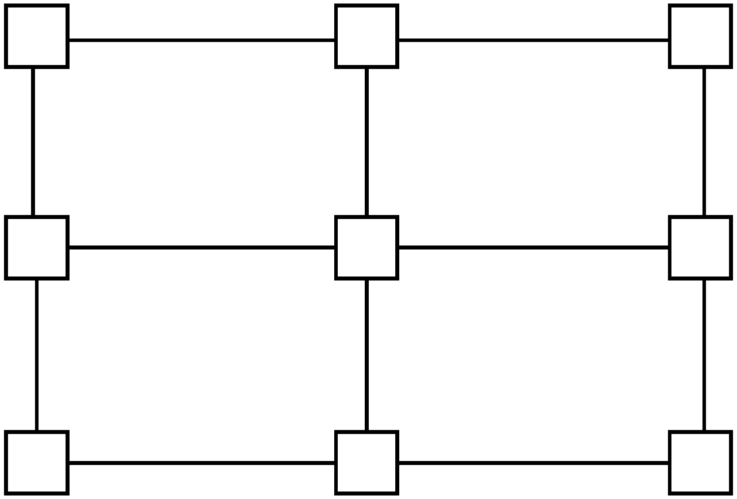
3.14b: Diameter of a Network

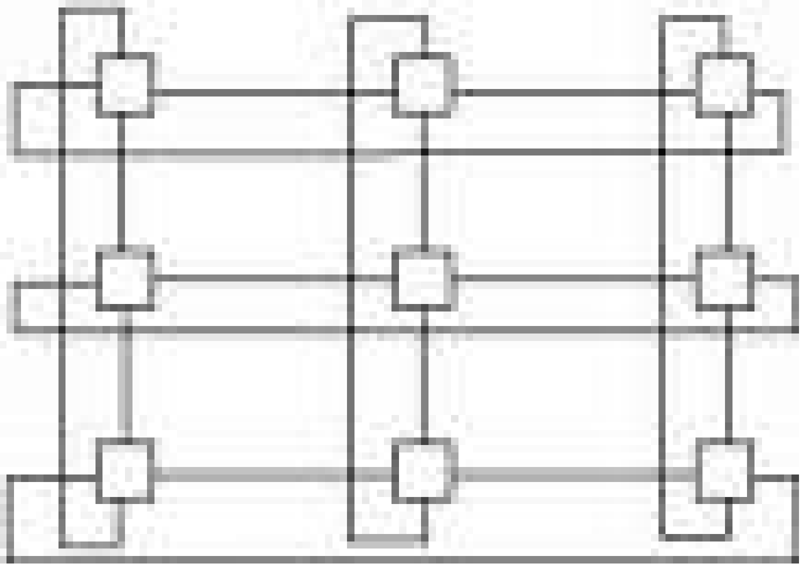
- The **diameter** of a network is the the maximum distance between any two processing nodes in the network
- The distance between two processing nodes is defined as the shortest path between them



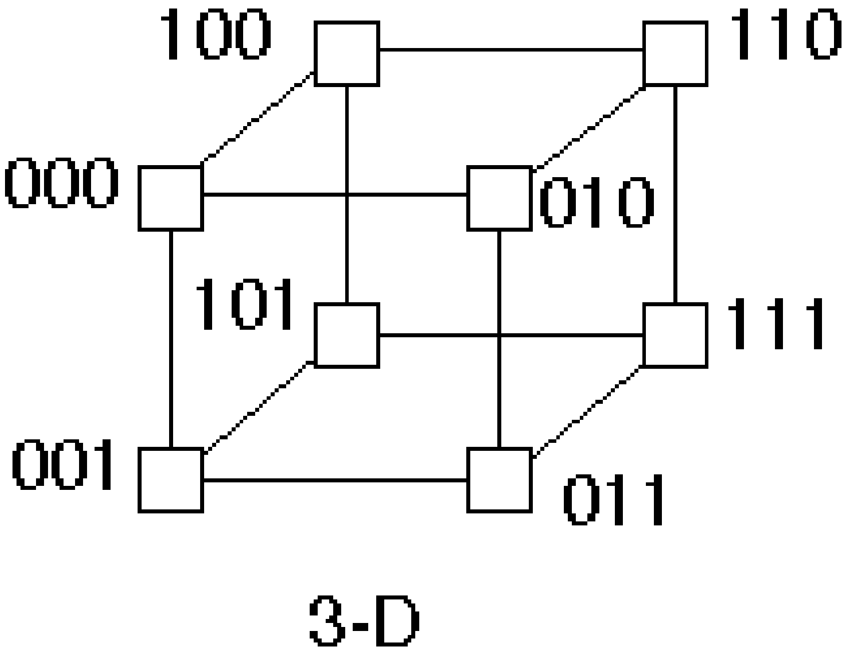


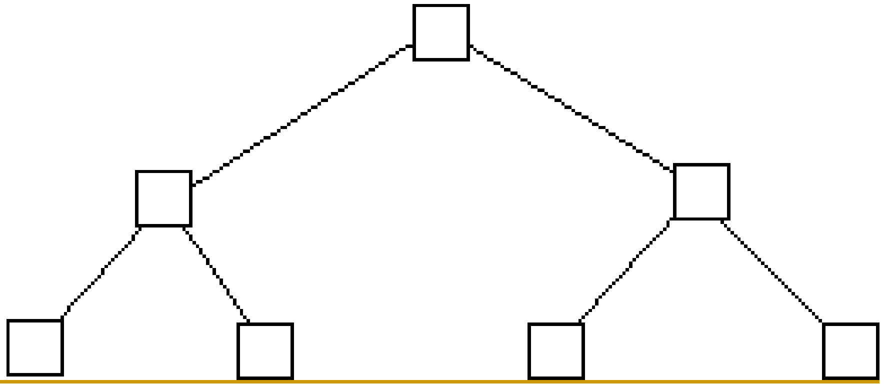
3.14c: Diameters of Mesh Networks





3.14d: Diameters of Hypercube and Tree

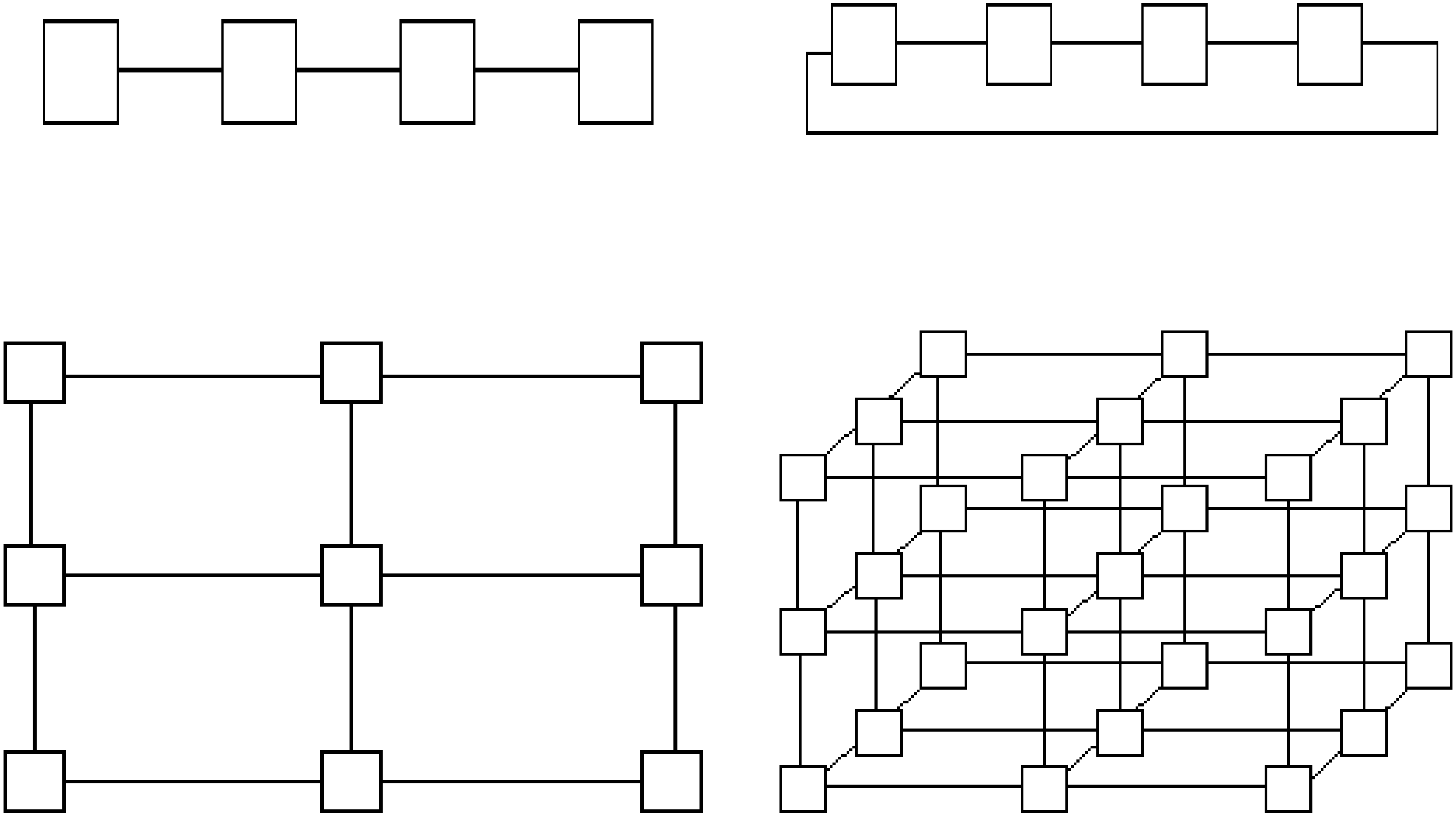




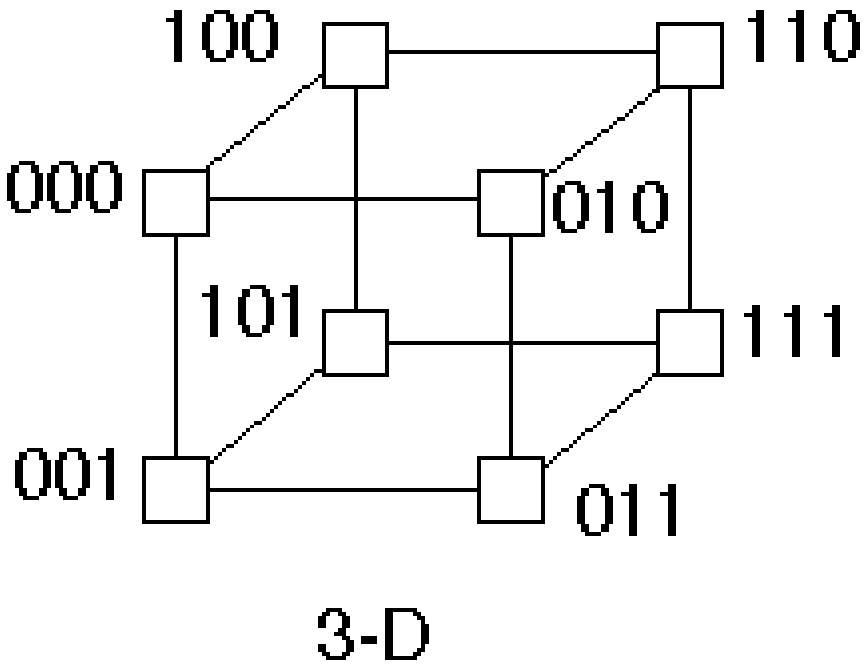
3.15a: Connectivity of Networks

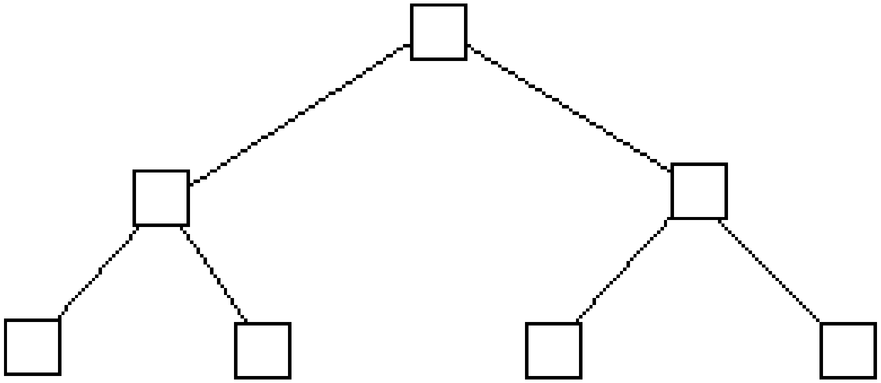
- The **connectivity** of a network is a measure of the multiplicity of paths between any two processing nodes
- The arc connectivity is the minimum number of arcs that must be removed from the network to break it into two disconnected networks
- A network with high connectivity has lower contentions for communication resources

3.15b: Connectivity of Mesh Array



3.15c: Connectivity of Hypercube & Tree





3.16a: Bisection Width & Channel Width

- The **bisection width** is the minimum number of communication links that must be removed to partition the network into two equal halves
- The **channel width** is the number of bits that can be communicated simultaneously over a link connecting two nodes
- Channel width is equal to the number of physical wires in each communication link

3.16b: Channel Rate & Channel Bandwidth

- The peak rate a single physical wire can deliver bits is called **channel rate**
- The **channel bandwidth** is the peak rate at which data can be communicated between the ends of a communication link
- Channel bandwidth is the product of channel rate and channel width
- The **bisection bandwidth** is the minimum volume of communication allowed between any two halves of the network
- It is the product of bisection width and channel bandwidth

3.16c: Characteristics of Static Networks

Network	Diameter	Bisection Width	Arc Connect.	Number of Links
Fully conn-ted	1	$p^2/4$	*p-1*	*p(p-1)/2*
Star	2	1	1	*p-1*
Binary tree	$2\log((p+1)/2)$	1	1	*p*-1
Linear array	*p-1*	1	1	*p-1*
Ring	|*p*-2|	2	2	*p*
2D mesh	$2(\sqrt{p}-1)$	$\sqrt{p}$	2	$2(p-\sqrt{p})$
2D meshwrap	$2\lfloor\sqrt{p}/2\rfloor$	$2\sqrt{p}$	4	*2p*
Hypercube	log *p*	*p/2*	log *p*	$(p\log p)/2$

3.17: Cost of Static Interconnection Networks

- The cost of a static network can be defined in proportion to the number of communication links or the number of wires required by the network
- Another measure is the bisection bandwidth of a network

  a lower bound on the area in a 2D packaging or the volume in a 3D packaging

  Definition is in terms of the order of magnitudes
- Completely connected and hypercube networks are more expensive than others

3.18: Evaluating Dynamic Interconnection Networks

- Need consider both processing nodes and switching units
- Criteria similar to those used with the static interconnection networks can be defined
- The **diameter** is defined as the maximum distance between any two nodes in the network
- The **connectivity** is the maximum number of nodes (or edges) that must fail to break the network
- The cost of a dynamic network is determined by the number of switching nodes in the network