Multiprocessors

Based on M. Morris Mano "Computer System Architecture"--Assist. Lecturer Ahmed Salah Hameed

Characteristics of Multiprocessors

- ***A multiprocessor system*** is an interconnection of two or more CPUs with memory and input-output equipment.

- The term "processor" in multiprocessor can mean either a central processing unit (CPU) or an input-output processor (IOP).

- A single CPU + one or more IOPs is not a (multiprocessor system)?Why?

- Multiprocessors are classified as ***multiple instruction stream***, ***multiple data stream*** **(MIMD)** systems.

Multiprocessor and Multicomputer

- Both support concurrent operations.
- Computers are interconnected with each other by means of communication lines to form a computer network.
- The network consists of several autonomous computers that may or may not communicate with each other.
- A multiprocessor system is controlled by one operating system that provides interaction between processors and all the components of the system cooperate in the solution of a problem.
- Multiprocessing (reliable system): a failure or error in one part has a limited effect on the rest of the system.

Based on M. Morris Mano "Computer System Architecture"--Assist. Lecturer Ahmed Salah Hameed

Multiprocessor system as parallel system

- **The system derives its high performance from the fact that computations can proceed in parallel in one of two ways:**

1. **Multiple independent jobs can be made to operate in parallel.**
2. **A single job can be partitioned into multiple parallel tasks.**

- **Improving performance by decomposing a program into parallel executable tasks. (two ways)**

1. **The user can explicitly declare that certain tasks of the program be executed in parallel. (done prior to loading the program) and (with help of OS).**
2. **(more efficient) way is to provide a compiler with multiprocessor software that can automatically detect parallelism in a user's program.**

- **What is the parallelizing compiler?**

Based on M. Morris Mano "Computer System Architecture"--Assist. Lecturer Ahmed Salah Hameed

Multiprocessors classification

Tightly coupled or Shared- Memory System

- Each processor can have its own local memory (cache memory)
- A global common memory (Shared memory) that all CPUs can access.
- Information shared among the CPUs by placing it in the common global memory.

Loosely Coupled: or Distributed-Memory System

- Each processor element has its own private local memory.
- The processors are tied together by a switching scheme designed to route information from one processor to another through a message-passing scheme.
- The processors relay program and data to other processors in packets.

NOTES

- Loosely coupled systems are most efficient when the interaction between tasks is minimal.
- Tightly coupled systems can tolerate a higher degree of interaction between tasks.

Based on M. Morris Mano "Computer System Architecture"--Assist. Lecturer Ahmed Salah Hameed

Interconnection Structures

- **The components that form a multiprocessor system are:**

1. **CPUs,**
2. **IOPs connected to input-output devices**
3. **Memory unit that may be partitioned into a number of separate modules.**

- **The interconnection between the components depending on:**
- The number of transfer paths that are available between the processors and memory in a **shared memory system**
- The number of transfer paths that are available among the processing elements in a **loosely coupled system.**

- **Some of interconnection forms:**

1. **Time-shared common bus**
2. **Multiport memory**
3. **Crossbar switch**
4. **Multistage switching network**
5. **Hypercube system**

Based on M. Morris Mano "Computer System Architecture"--Assist. Lecturer Ahmed Salah Hameed

Time Shared Common Bus-a single bus structure

- A number of processors connected through a common path to a memory unit.

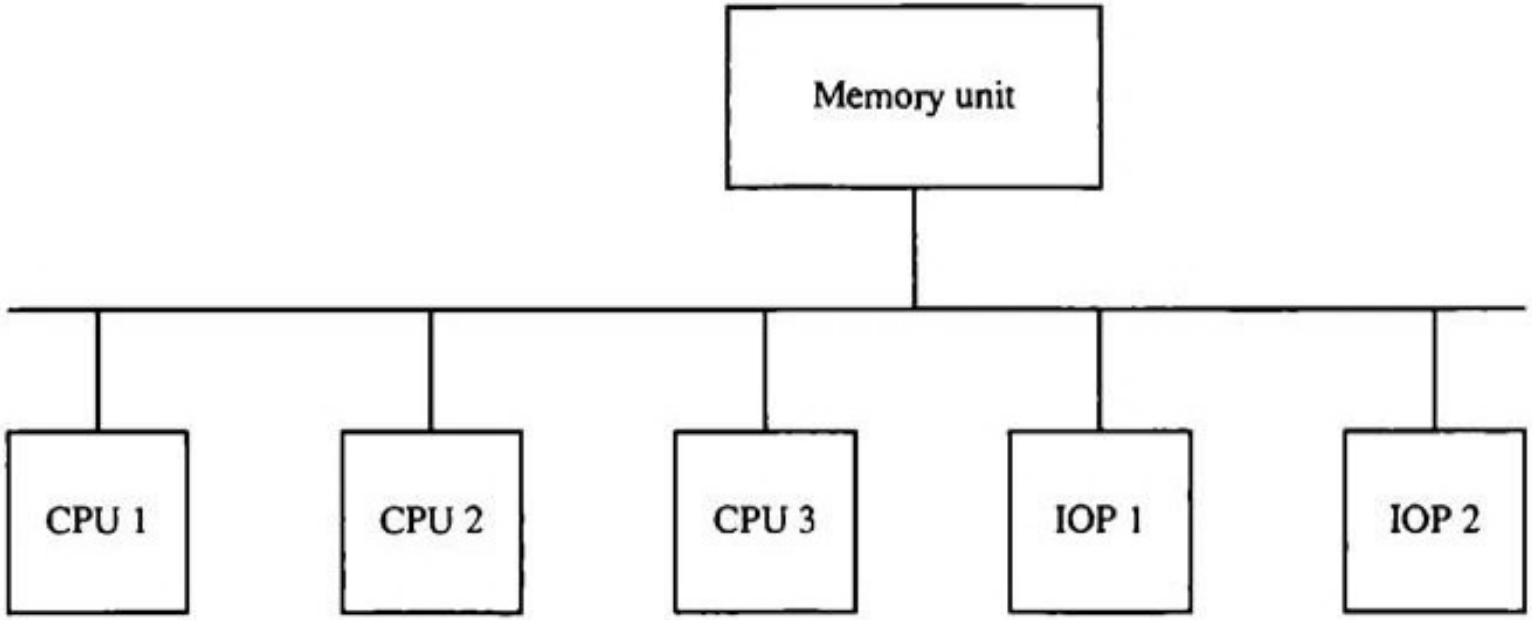


Figure 13-1 Time-shared common bus organization.

- A single common-bus system is restricted to one transfer at a time.
- All other processors are either busy with internal operations or must be idle waiting for the bus.
- Conflicts must be resolved by incorporating a bus controller that establishes priorities among the requesting units.

Time Shared Common Bus-a dual bus structure

- **Each local bus connected to its own local memory and to one or more processors.**
- **A system bus controller links each local bus to a common system bus.**
- **Only one processor can communicate with the shared memory and other common resources through the system bus at any given**

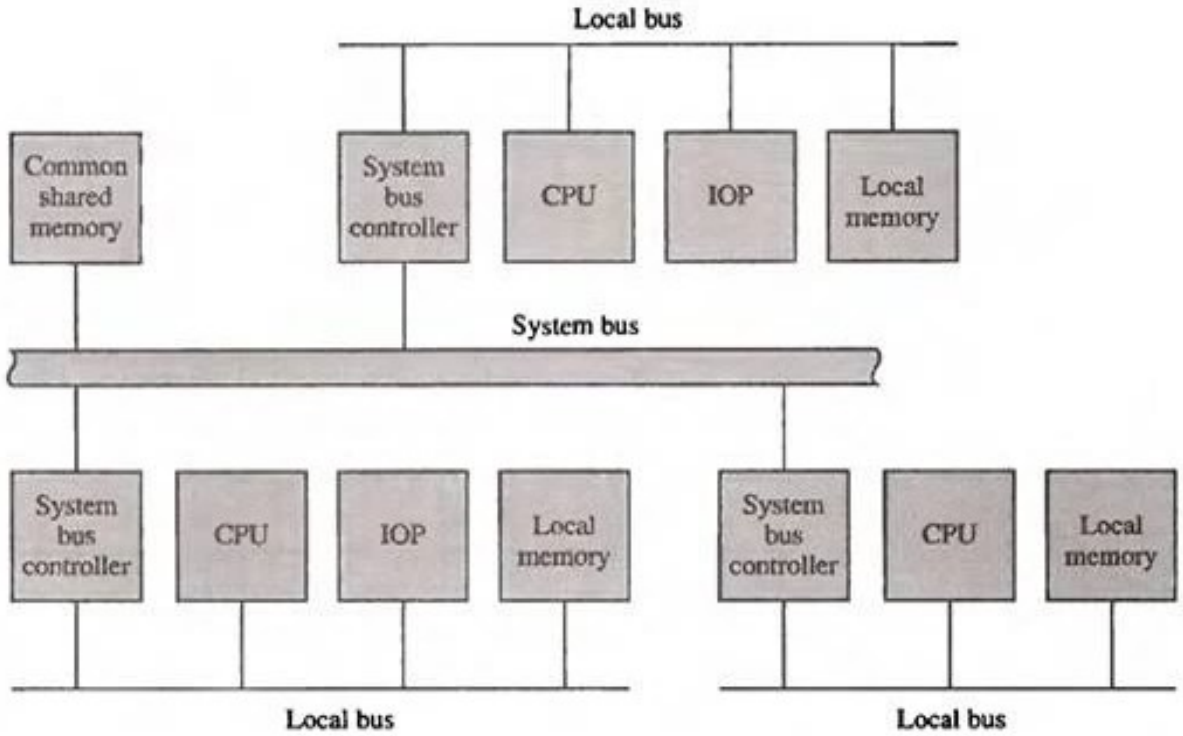


Figure 13-2 System bus structure for multiprocessors.

Based on M. Morris Mano "Computer System Architecture"--Assist. Lecturer Ahmed Salah Hameed

Multiport Memory

- **A multiport memory system employs separate buses between each memory module and each CPU.**
- **A processor bus consists of the address, data, and control lines required to communicate with memory.**

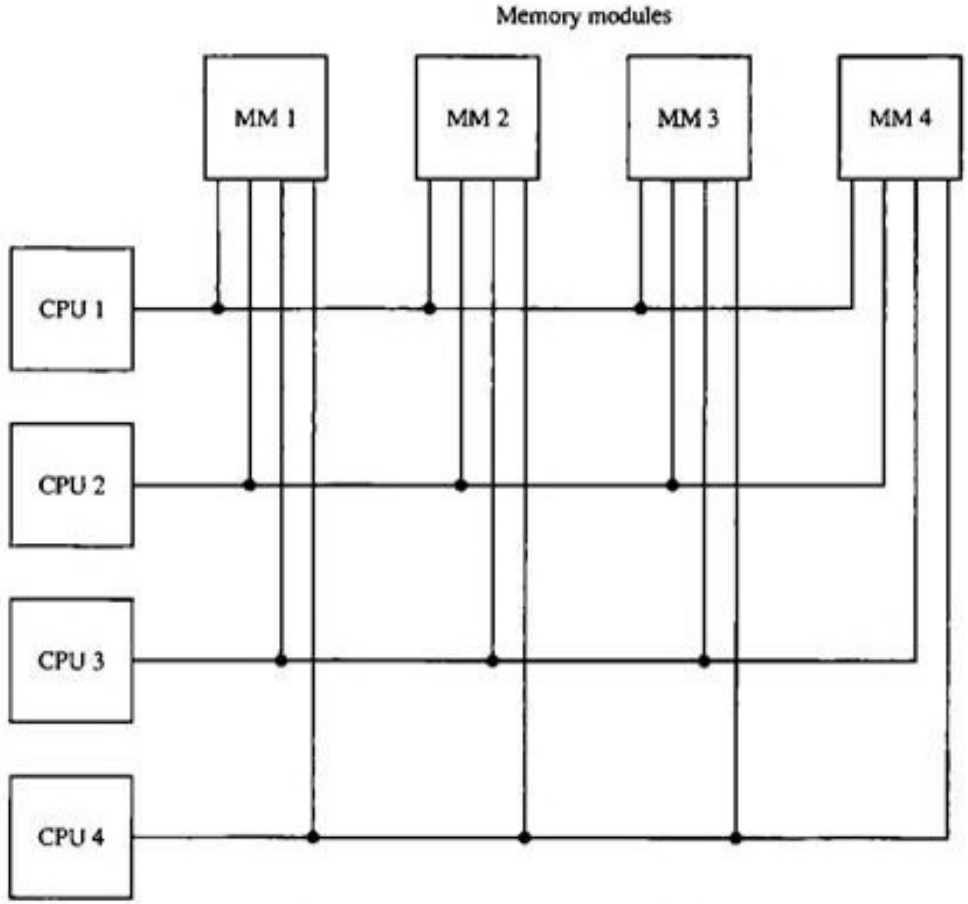


Figure 13-3 Multiport memory organization.

Based on M. Morris Mano "Computer System Architecture"--Assist. Lecturer Ahmed Salah Hameed

Multiport Memory

- **Memory access conflicts are resolved by assigning fixed priorities to each memory port.**

Example: The physical port position as priority address:

CPU 1 will have priority over CPU 2, CPU 2 will have priority over CPU 3, and CPU 4 will have the lowest priority.

The advantage of the multiport memory organization

- **The high transfer rate that can be achieved because of the multiple paths between processors and memory.**

The disadvantage

- **It requires expensive memory control logic and a large number of cables and connectors. As a consequence, this interconnection structure is usually appropriate for systems with a small number of processors.**

Crossbar Switch

- **The crossbar switch organization consists of a number of cross points that are placed at intersections between processor buses and memory module paths.**
- **A switch that determines the path from a processor to a memory module.**
- **Each switch point has control logic to set up the transfer path between a processor and memory.**

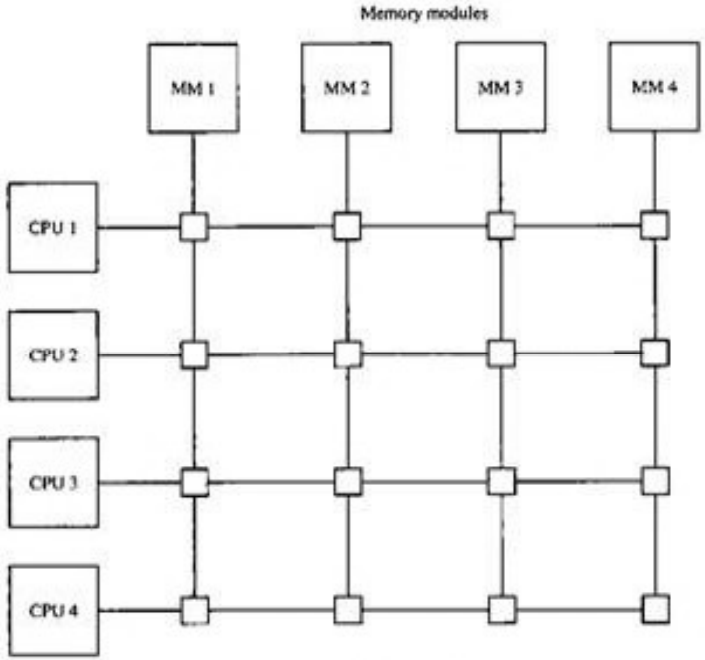


Figure 13-4 Crossbar switch.

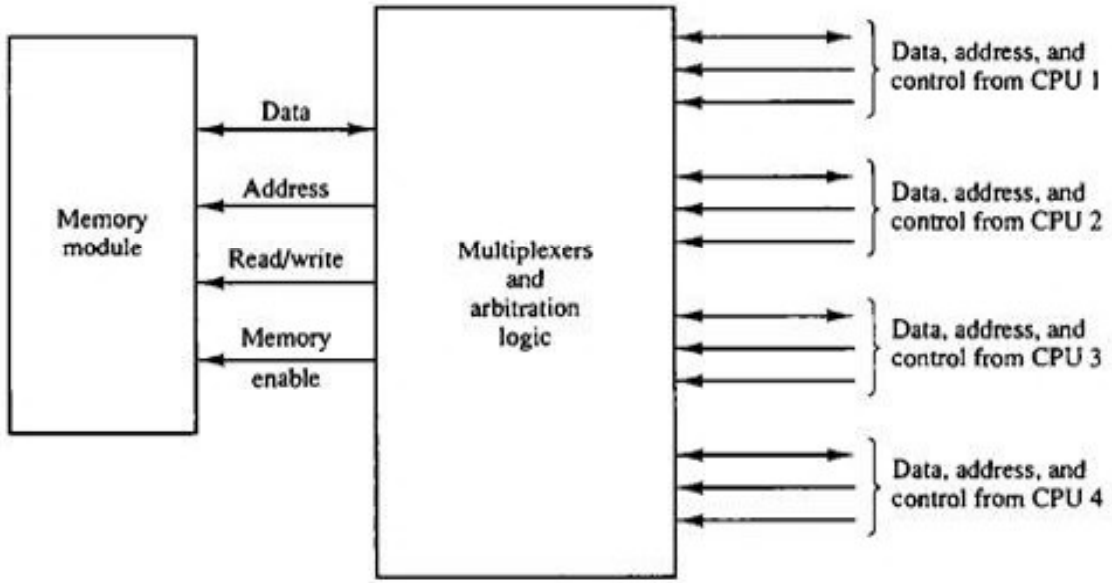
Based on M. Morris Mano "Computer System Architecture"--Assist. Lecturer Ahmed Salah Hameed

Crossbar Switch

Crossbar switch-control logic

- **It examines the address that is placed in the bus to determine whether its particular module is being addressed.**
- **It resolves multiple requests for access to the same memory module on a predetermined priority basis.**
- **Priority levels are established by the arbitration logic to select one CI when two or more CPUs attempt to access the same memory.**

Figure 13-5 Block diagram of crossbar switch.



Based on M. Morris Mano "Computer System Architecture"--Assist. Lecturer Ahmed Salah Hameed

Crossbar Switch

Advantage

- **A crossbar switch organization supports simultaneous transfers from memory modules because there is a separate path associated with each module.**

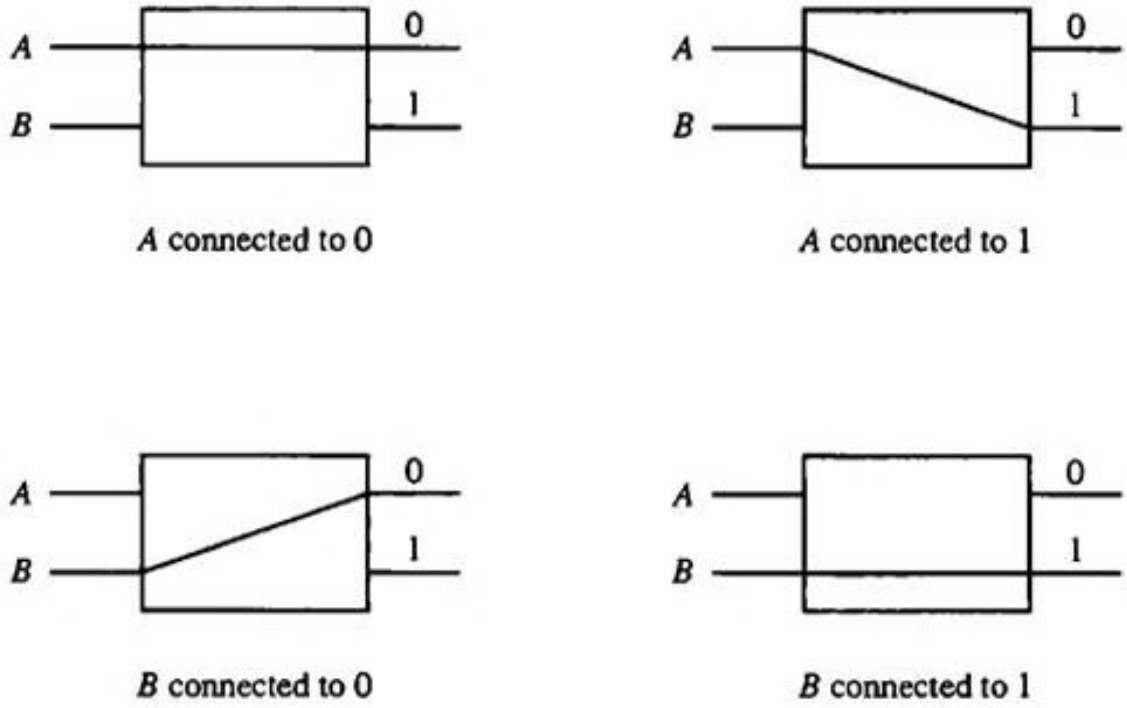
Disadvantage

- **The hardware required to implement the switch can become quite large and complex.**

Multistage Switching Network

The basic component of a multistage network is a two-input, two-output interchange switch.

Figure 13-6 Operation of a 2 × 2 interchange switch.



Multistage Switching Network-The binary tree

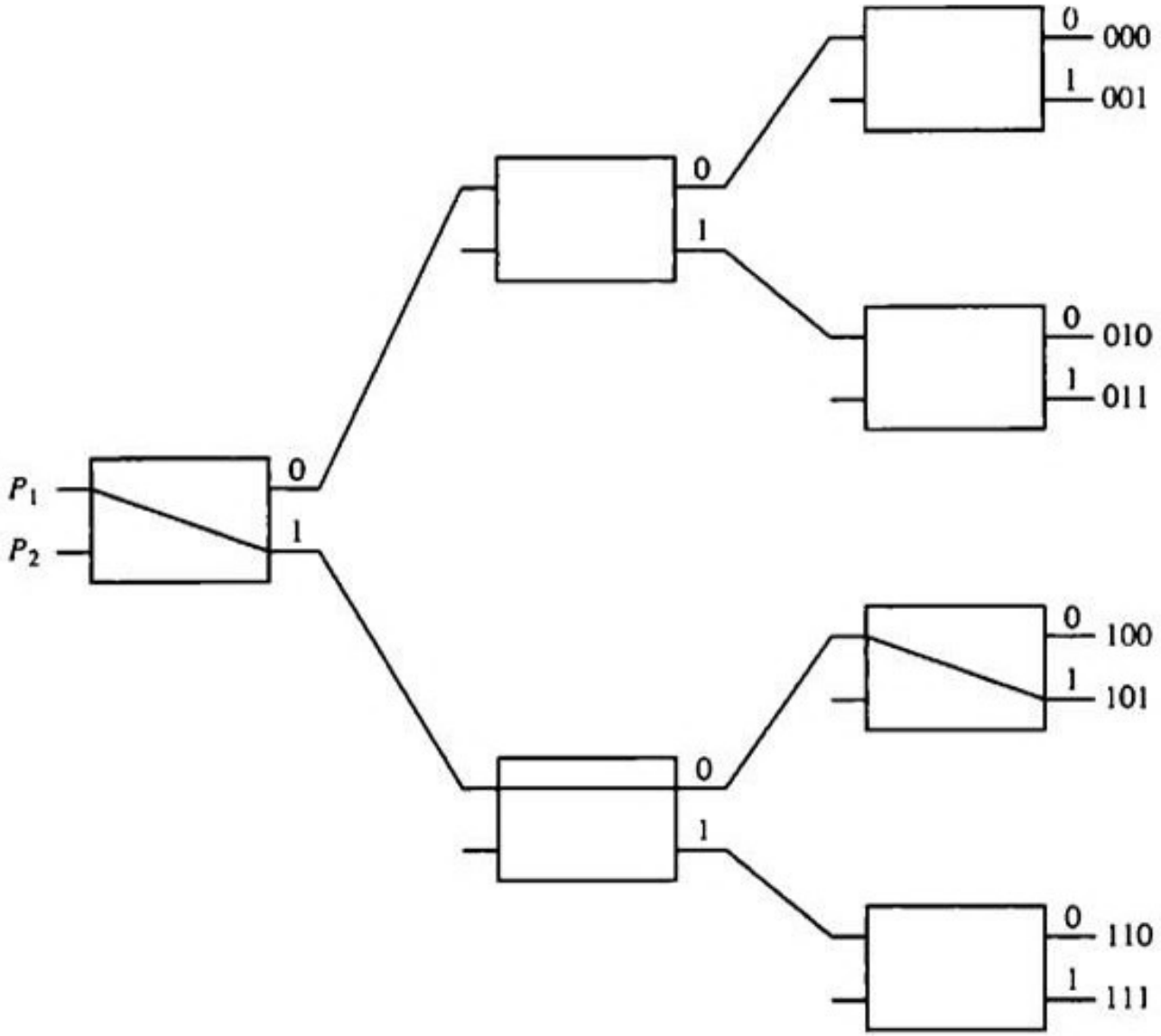


Figure 13-7 Binary tree with 2 × 2 switches.

Multistage Switching Network-Omega network

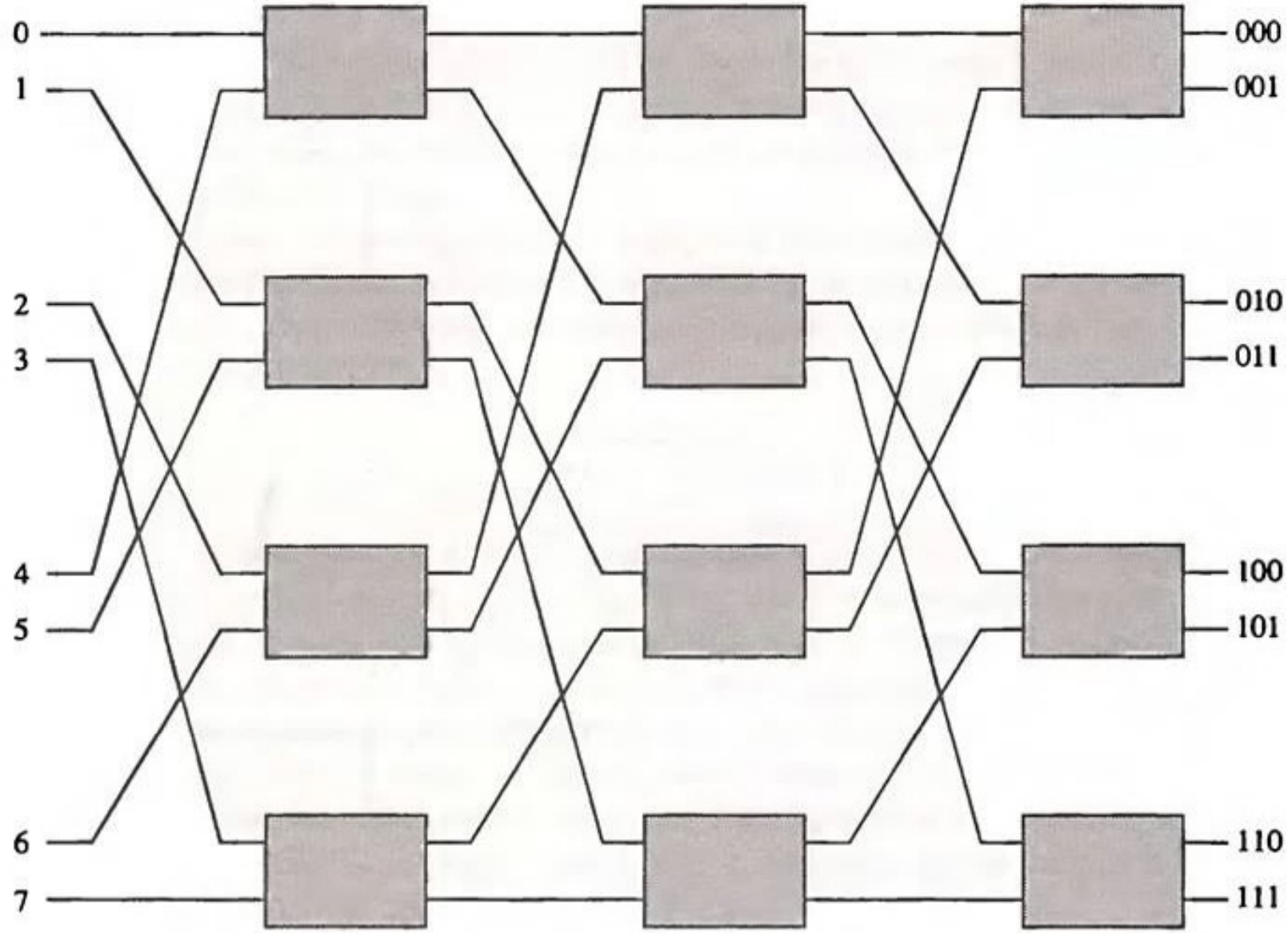


Figure 13-8 8 × 8 omega switching network.

Based on M. Morris Mano "Computer System Architecture"--Assist. Lecturer Ahmed Salah Hameed

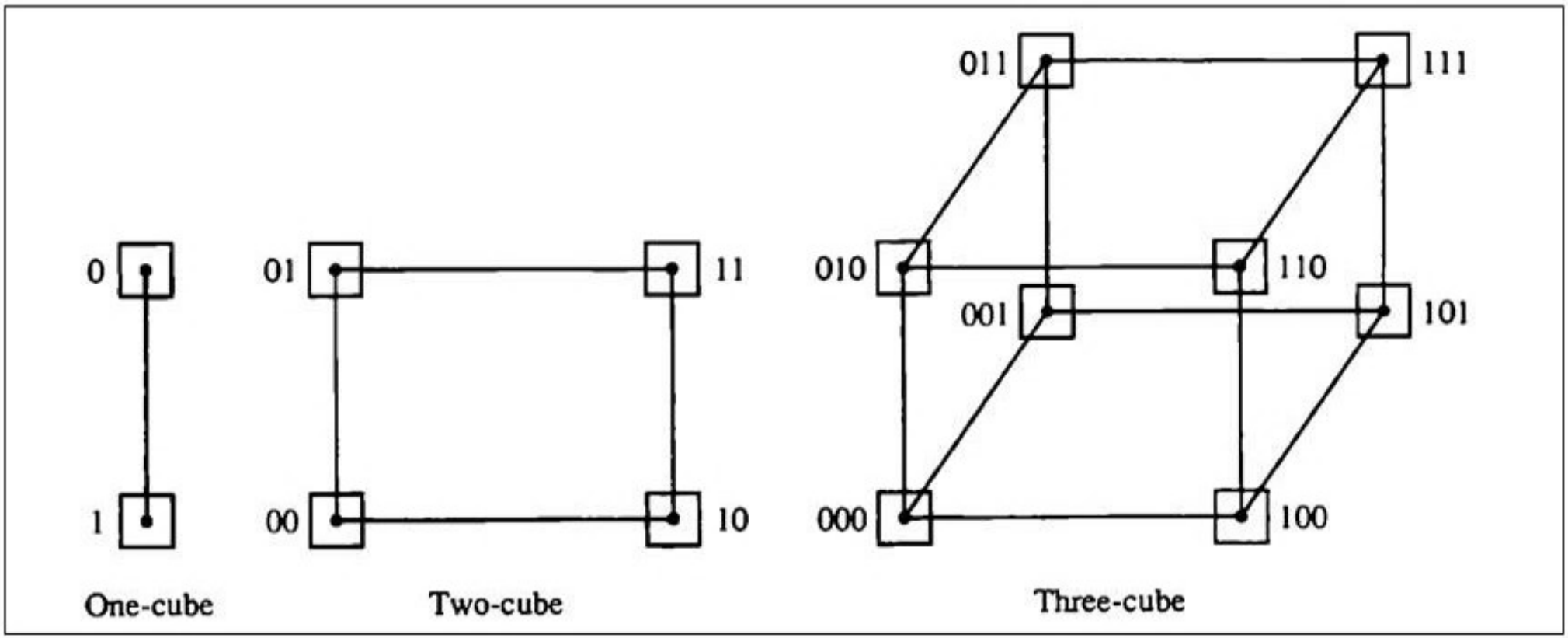
Hypercube Interconnection

- **Also called binary n-cubes.**
- **It is a loosely coupled system composed of $N = 2^n$ processors interconnected in an n -dimensional binary cube.**
- **Each processor forms a node of the cube.**
- **Each processor has direct communication paths to n other neighbor processors.**
- **These paths correspond to the edges of the cube. There are 2^n distinct n-bit binary addresses that can be assigned to the processors.**
- **Each processor address differs from that of each of its n neighbors by exactly one bit position.**

Based on M. Morris Mano "Computer System Architecture"--Assist. Lecturer Ahmed Salah Hameed

Hypercube Interconnection Labeling

- **Each node is assigned a binary address in such a way that the addresses of two neighbors differ in exactly one bit position.**

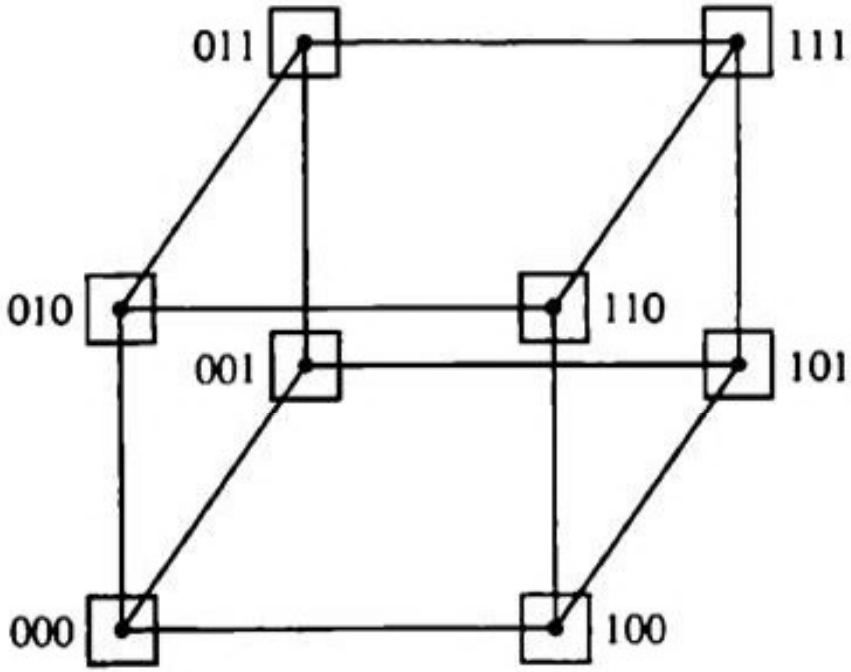


- **Four-cube and five-cube?**

Based on M. Morris Mano "Computer System Architecture"--Assist. Lecturer Ahmed Salah Hameed

Hypercube Routing

- **With a three-cube structure, node 000 can communicate directly with node 001.**
- **Node 000 must cross at least two links to communicate with 011 (from 000 to 001 to 011 or from 000 to 010 to 011).**
- **It is necessary to go through at least three links to communicate from node 000 to node 111.**



Three-cube

Based on M. Morris Mano "Computer System Architecture"--Assist. Lecturer Ahmed Salah Hameed

Hypercube Routing

A routing procedure

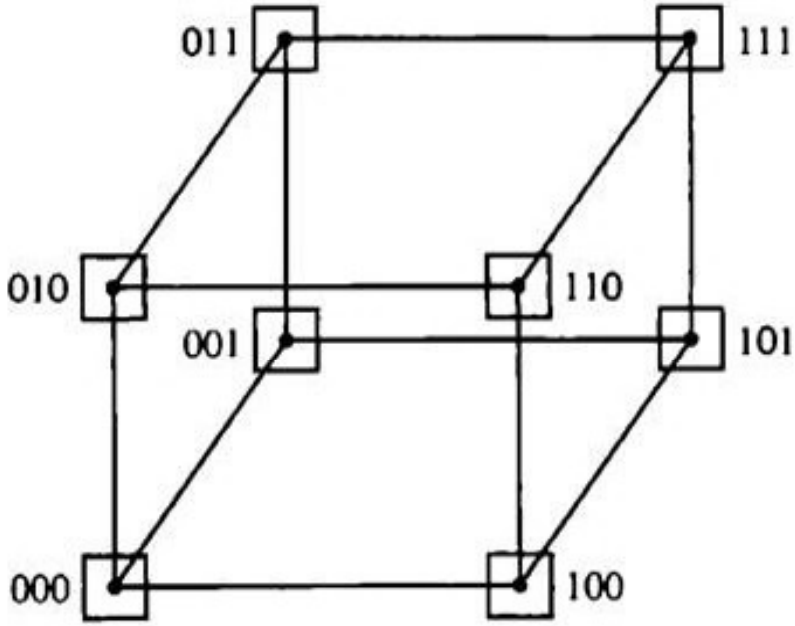
- **Computing the exclusive-OR of the source node address with the destination node address.**
- **The resulting binary value will have 1 bits corresponding to the axes on which the two nodes differ.**
- **The message is then sent along any one of the axes.**

Example 1: Message at 010 going to 001

(010 x-or 001) = 011.

Example 1: Message at 000 going to 111

(000 x-or 111) = 111.



Three-cube

PROBLEMS

- Discuss the difference between tightly coupled multiprocessors and loosely coupled multiprocessors from the viewpoint of hardware organization and programming techniques.

- What is the purpose of the system bus controller shown in Fig. 13-2? Explain how the system can be designed to distinguish between references to local memory and references to common shared memory.

- How many switch points are there in a crossbar switch network that connects p processors to m memory modules?

Switches = number of Processors × number of Memory modules

PROBLEMS

- The 8x8 omega switching network of Fig. 13-8 has three stages with four switches in each stage, for a total of 12 switches. How many stages and switches per stage are needed in an n x n omega switching network?

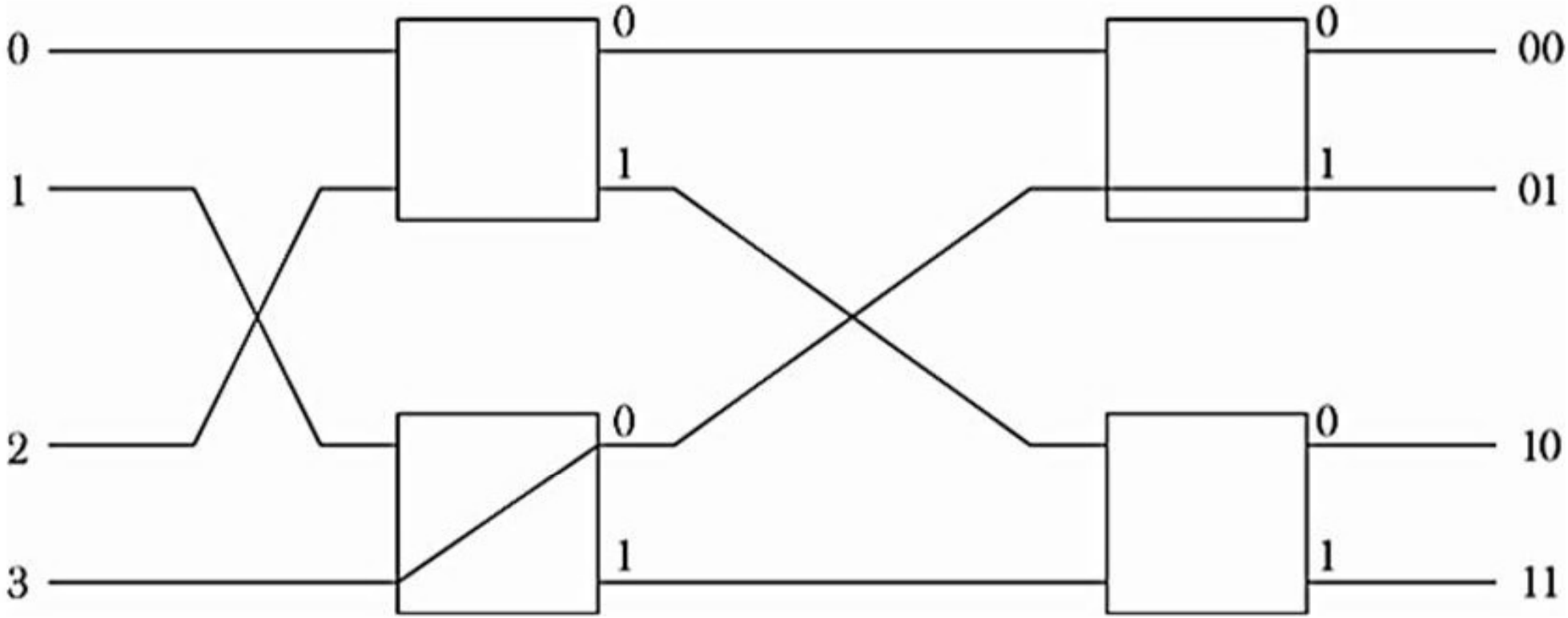
Number of stages = Log of (n) for base 2

Number of switches per stage = n/2

- Suppose that the wire breaks between the switch in the first row, second column and the switch in the second row, third column in the omega switching network of Fig. 13-8. What paths will be disconnected?

PROBLEMS

- Construct a diagram for a 4 x 4 omega switching network. Show the switch setting required to connect input 3 to output 1.



Based on M. Morris Mano "Computer System Architecture"--Assist. Lecturer Ahmed Salah Hameed

PROBLEMS

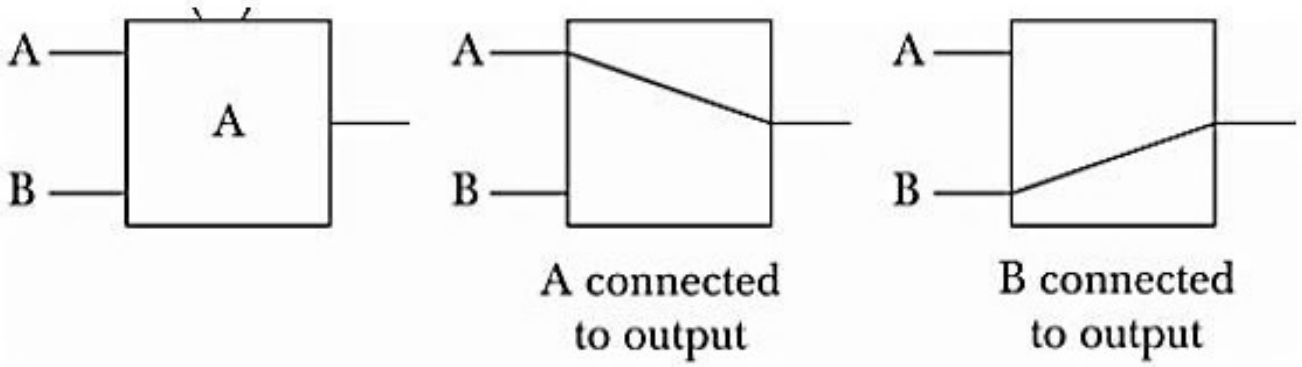
- Three types of switches are used to design a multistage interconnection network: an interchange switch with two inputs and two outputs as in Fig. 13-6, an arbitration switch with two inputs and one output, and a distribution switch with one input and two outputs.

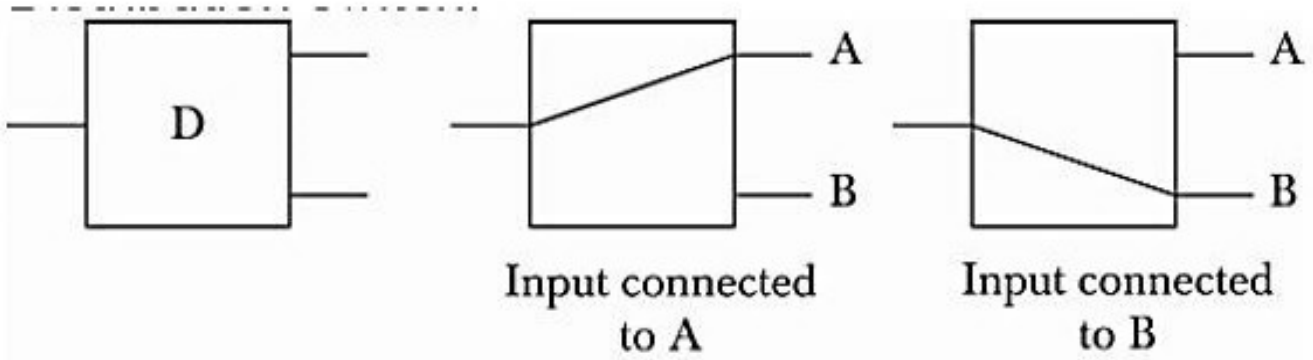
a. Show how the arbitration and distribution switches operate.

b. Using arbitration and interchange switches, construct an 8 x 4 network with a unique path between any source and any destination.

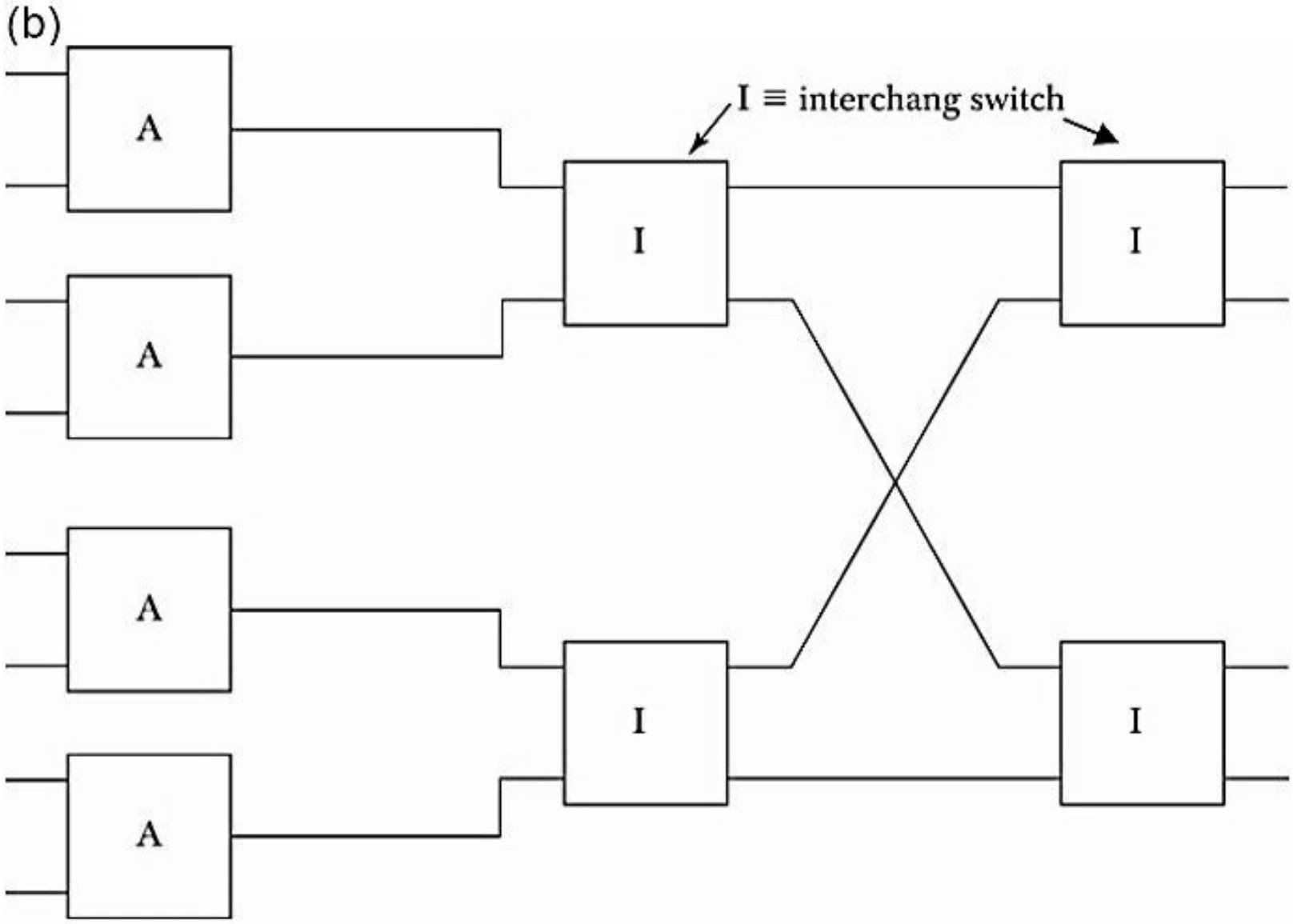
c. Using distribution and interchange switches, construct a 4 x 8 network with a unique path between any source and any destination.

PROBLEMS


A
B
A
A
B
A connected
to output
A
B
B connected
to output


D
A
B
Input connected
to A
A
B
Input connected
to B

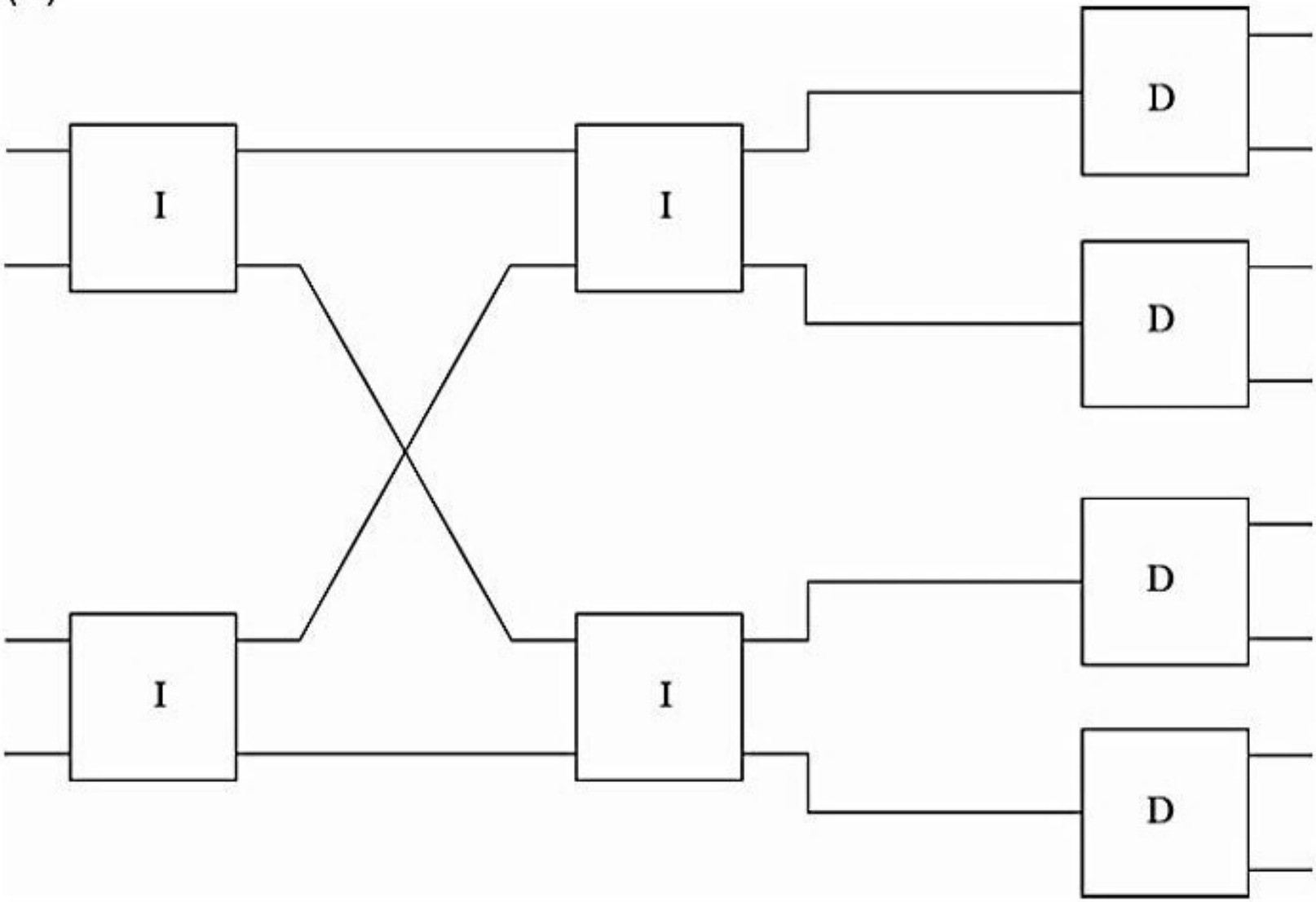
PROBLEMS



Based on M. Morris Mano "Computer System Architecture"--Assist. Lecturer Ahmed Salah Hameed

PROBLEMS

(c)



Based on M. Morris Mano "Computer System Architecture"--Assist. Lecturer Ahmed Salah Hameed

PROBLEMS

- Draw a diagram showing the structure of a four-dimensional hypercube network. List all the paths available from node 7 to node 9 that use the minimum number of intermediate nodes.

Interprocessor Arbitration

Buses

- **What is Bus?**
- ***A memory bus*** consists of lines for transferring data, address, and read/write information.
- ***An I/O bus*** is used to transfer information to and from input and output devices.
- ***A system bus*** connects major components in a multisystem bus processor system, such as CPUs, IOPs, and memory. A typical system bus consists of approximately 100 signal lines. These lines are divided into three functional groups: data, address, and control.
- **Bus conflict?**
- Arbitration logic resolves bus conflict
- It would be the part of system bus controller

System Bus & Data transfers

1) synchronous bus (common clock source)
2) asynchronous bus (handshaking control signals)

Based on M. Morris Mano "Computer System Architecture"--Assist. Lecturer Ahmed Salah Hameed

Interprocessor Arbitration

- Arbitration logic resolves bus conflict
- It would be the part of system bus controller

Arbitration logic (Two types)

1) Serial arbitration procedure (daisy-chain connection): The processors connected to the system bus are assigned priority according to their position along the priority control line. The device closest to the priority line is assigned the highest priority.
2) Parallel arbitration procedure

- Arbitration procedures services all processor requests on the basis of established priorities

Serial arbitration procedure

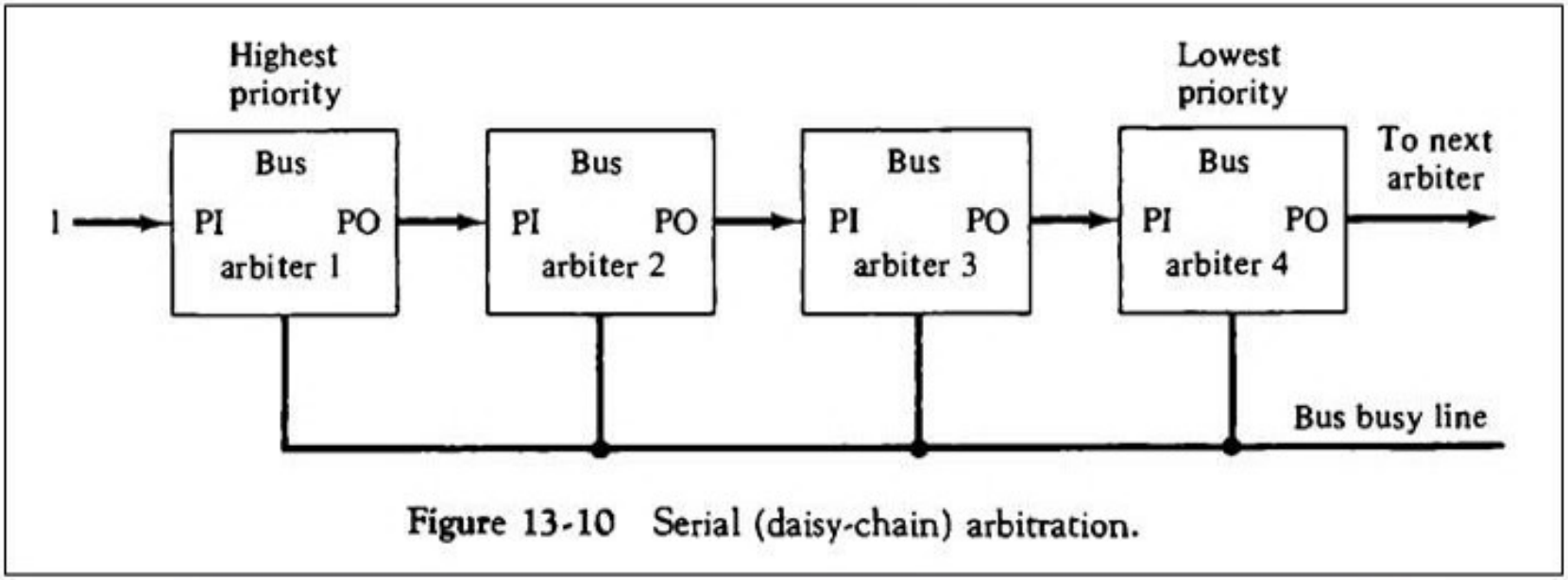
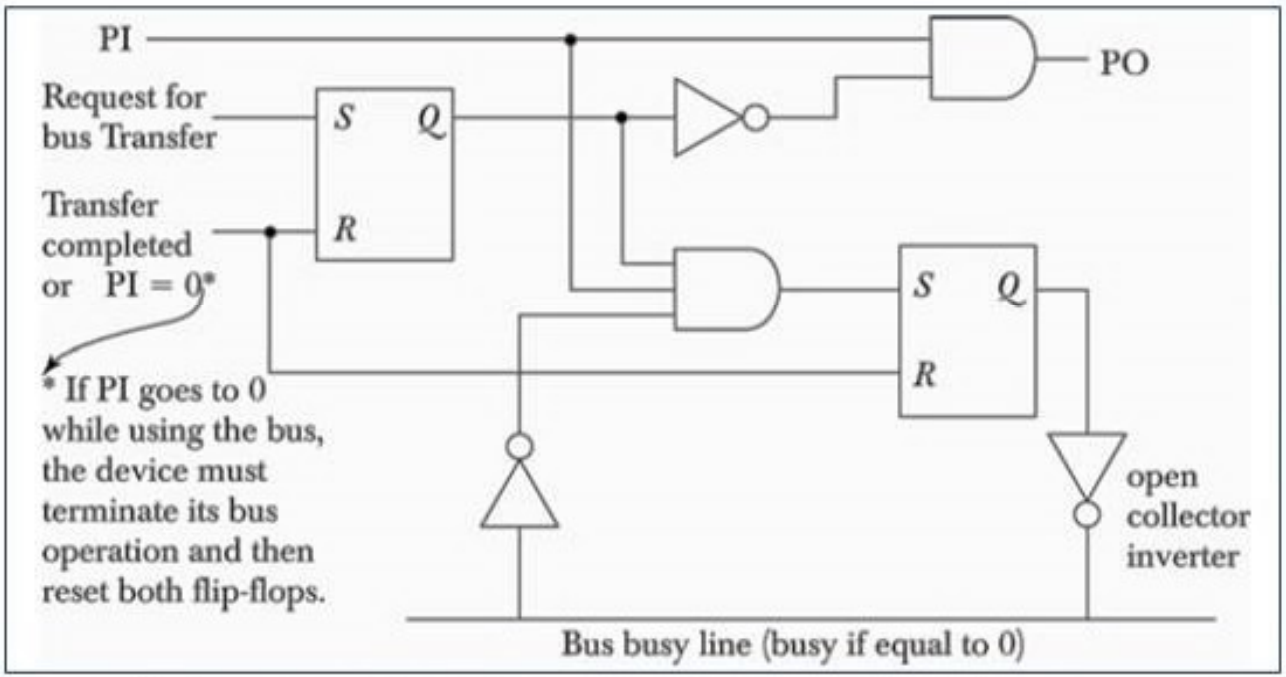


Figure 13-10 Serial (daisy-chain) arbitration.



Based on M. Morris Mano "Computer System Architecture"--Assist. Lecturer Ahmed Salah Hameed

Parallel arbitration procedure

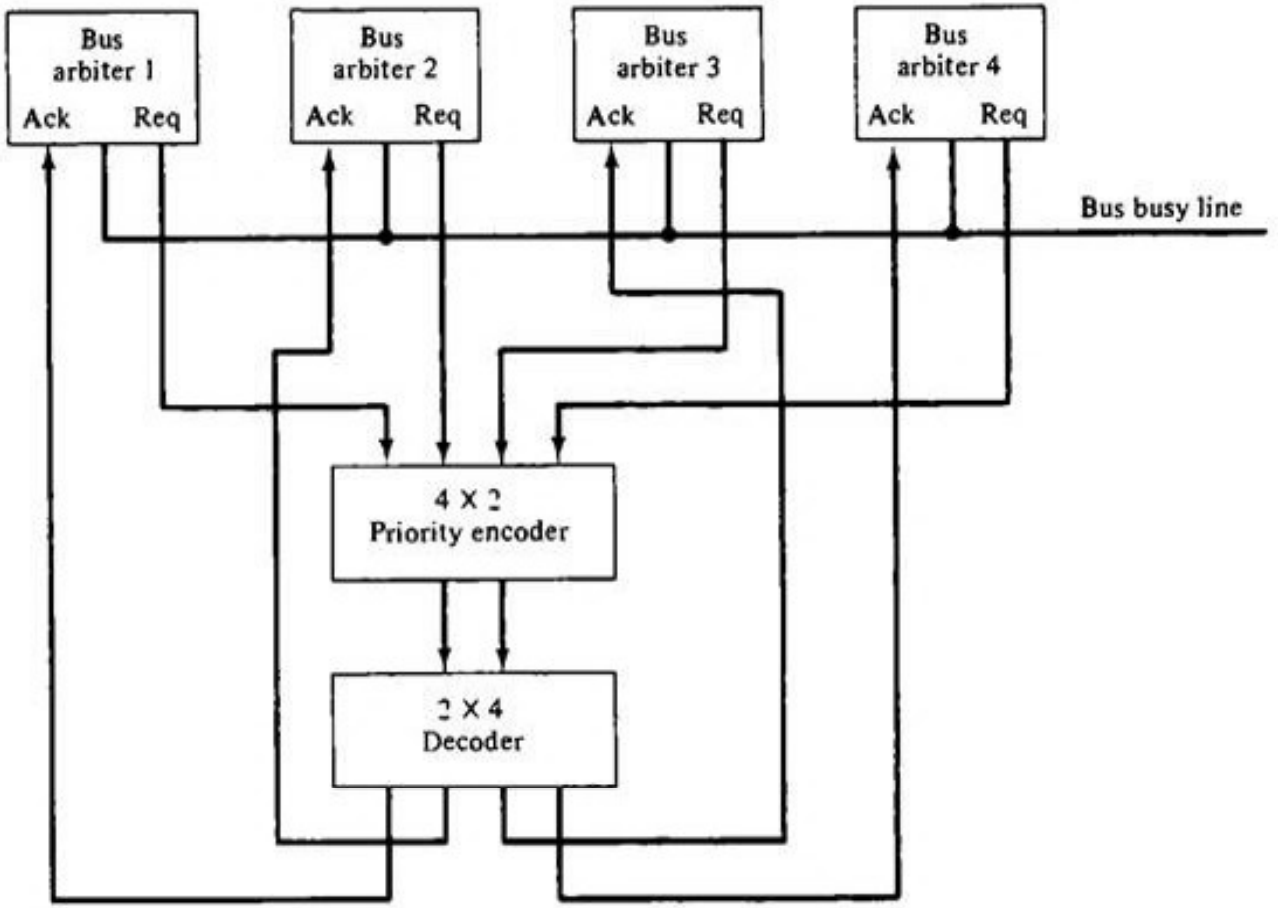


Figure 13-11 Parallel arbitration.

TABLE 11-2 Priority Encoder Truth Table

Inputs				Outputs			Boolean functions
I_0	I_1	I_2	I_3	x	y	IST	
1	×	×	×	0	0	1	
0	1	×	×	0	1	1	$x = I_0' I_1'$
0	0	1	×	1	0	1	$y = I_0' I_1 + I_0' I_2'$
0	0	0	1	1	1	1	$(IST) = I_0 + I_1 + I_2 + I_3$
0	0	0	0	×	×	0	

Based on M. Morris Mano "Computer System Architecture"--Assist. Lecturer Ahmed Salah Hameed

Dynamic Arbitration Algorithms

- Static priority (Serial and Parallel)

Dynamic Arbitration Algorithms:

1) The time slice algorithm
2) Polling algorithm
3) The least recently used (LRU) algorithm
4) The first-come, first-serve (FIFO) algorithm
5) The rotating daisy-chain procedure

Interprocessor Communication & Synchronization

- ***Communication*** refers to the exchange of data between different processes. For example, parameters passed to a procedure in a different processor constitute interprocessor communication.

- ***Synchronization*** refers to the special case where the data used to communicate between processors is control information. Synchronization is needed to enforce the correct sequence of processes and to ensure mutually exclusive access to shared writable data.

- One of the most popular methods is through the use of a binary semaphore.

Interprocessor Communication & Synchronization

Shared memory

- Communication is by common area of shared memory
- Sender alert the receiver by interrupt and data transfer through I/O path.

Loosely coupled

- Communication is through message passing I/O channels
- One calls a procedure which resides in the memory of the destination
- O.S. in each node controls the communication

Organizations used in design of OS for multiprocessors

1) master-slave configuration
2) separate operating system
3) distributed operating system.

Mutual Exclusion with a Semaphore

- Mutual exclusion.
- Critical section.
- A semaphore
- Testing and setting
- Hardware lock

Cache Coherence

- The primary advantage of cache is its ability to reduce the average access time in uniprocessors.

Read procedures

- When the processor finds a word in cache during a read operation, the main memory is not involved in the transfer.

Write procedures

- ***write-through policy*** :both cache and main memory are updated with every write operation.
- ***write-back policy***, only the cache is updated and the location is marked so that it can be copied later into main memory.

- What is **cache coherence** problem?
- Cache coherence problems exist in multiprocessors with private caches because of the need to share writable data. Read-only data can safely be replicated without cache coherence enforcement mechanisms.

Based on M. Morris Mano "Computer System Architecture"--Assist. Lecturer Ahmed Salah Hameed

Example

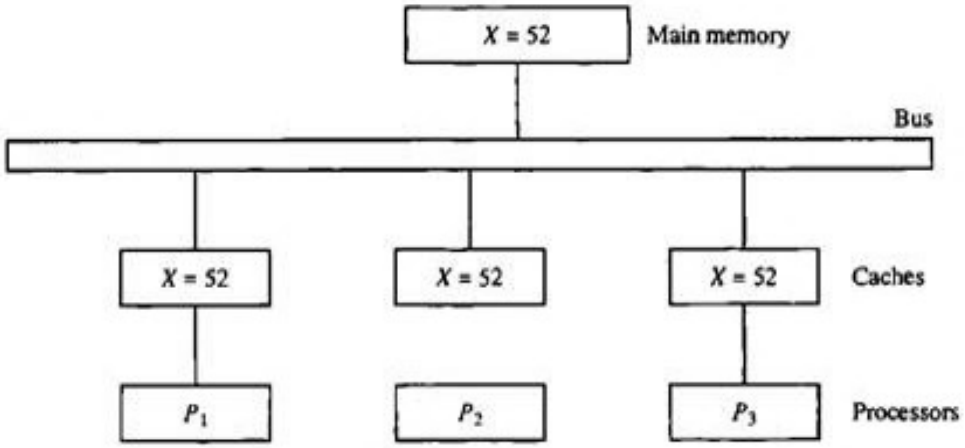
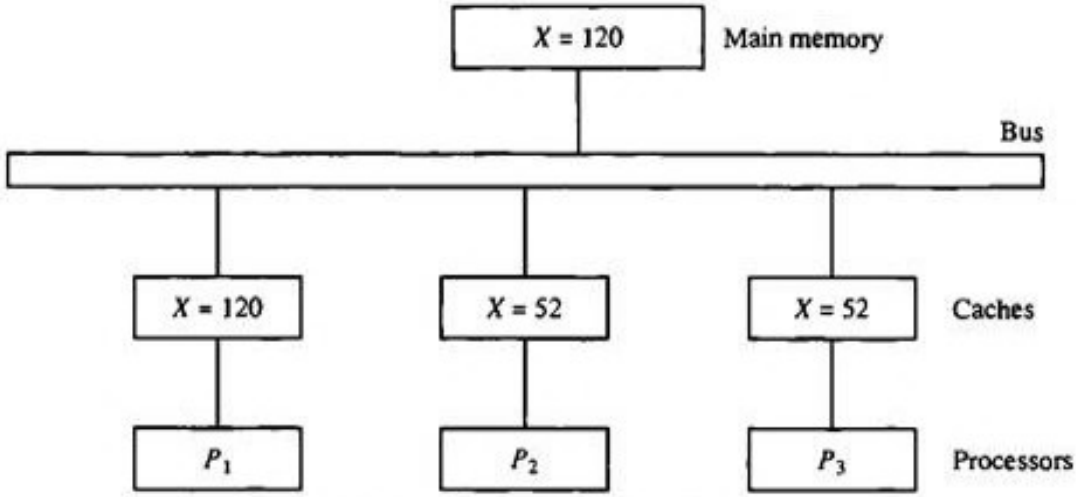
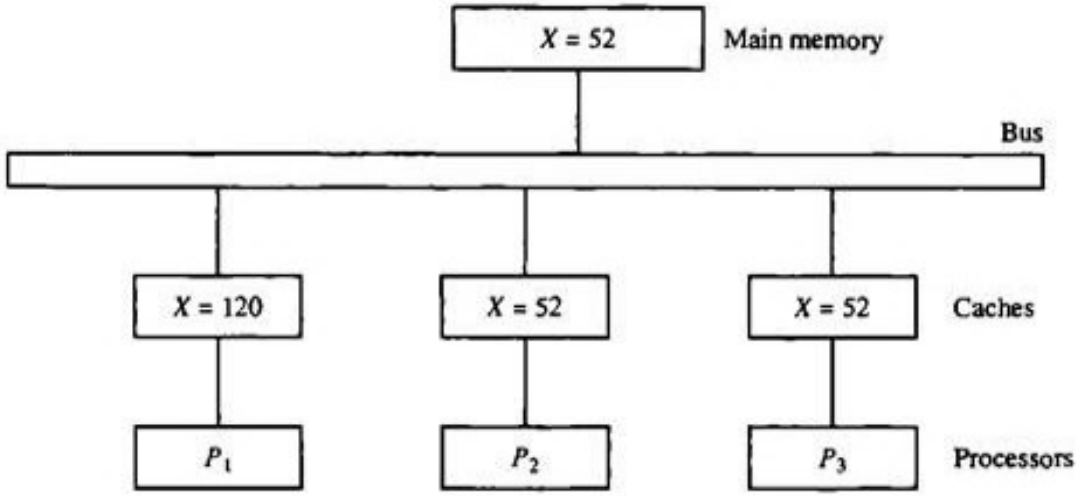


Figure 13-12 Cache configuration after a load on X.

Figure 13-13 Cache configuration after a store to X by processor P_1.



(a) With write-through cache policy



(b) With write-back cache policy

Based on M. Morris Mano "Computer System Architecture"--Assist. Lecturer Ahmed Salah Hameed

Solutions to the Cache Coherence Problem

- For performance considerations it is desirable to attach a private cache to each processor.

Solutions to the cache coherence problems

- Disallow private caches for each processor and have a shared cache memory associated with main memory.
- Only nonshared and read only data to be stored in caches : cachable (Shared writable data are noncachable)
- Centralized global table in its compiler: A scheme that allows writable data to exist in at least one cache.

Hardware solutions:

- Snoopy cache controller–monitor all bus requests
- All caches constantly monitor the bus for possible write operations
- Write through policy.

PROBLEMS

- **13-9.** Draw a logic diagram using gates and flip-flops showing the circuit of one bus arbiter stage in the daisy-chain arbitration scheme of Fig. 13-10.
- **13-10.** The bus controlled by the parallel arbitration logic shown in Fig. 13-11 is initially idle. Devices 2 and 3 then request the bus at the same time. Specify the input and output binary values in the encoder and decoder and determine which bus arbiter is acknowledged.
- **13-11.** Show how the arbitration logic of Fig. 13-10 can be modified to provide a rotating daisy-chain arbitration procedure. Explain how the priority is determined once the bus line is disabled.
- **13-12.** Consider a bus topology in which two processors communicate through a buffer in shared memory. When one processor wishes to communicate with the other processor it puts the information in the memory buffer and sets a flag. Periodically, the other processor checks the flags to determine if it has information to receive. What can be done to ensure proper synchronization and to minimize the time between sending and receiving the information?
- One way to speed the transfer would be to send an interrupt request to the receiving processor.

Based on M. Morris Mano "Computer System Architecture"--Assist. Lecturer Ahmed Salah Hameed

PROBLEMS

- **13-13.** Describe the following terminology associated with multiprocessors, (a) mutual exclusion; (b) critical section; (c) hardware lock; (d) semaphore; (e) test-and-set instruction.
- **13-14.** What is cache coherence, and why is it important in shared-memory multiprocessor systems? How can the problem be resolved with a snoopy cache controller?