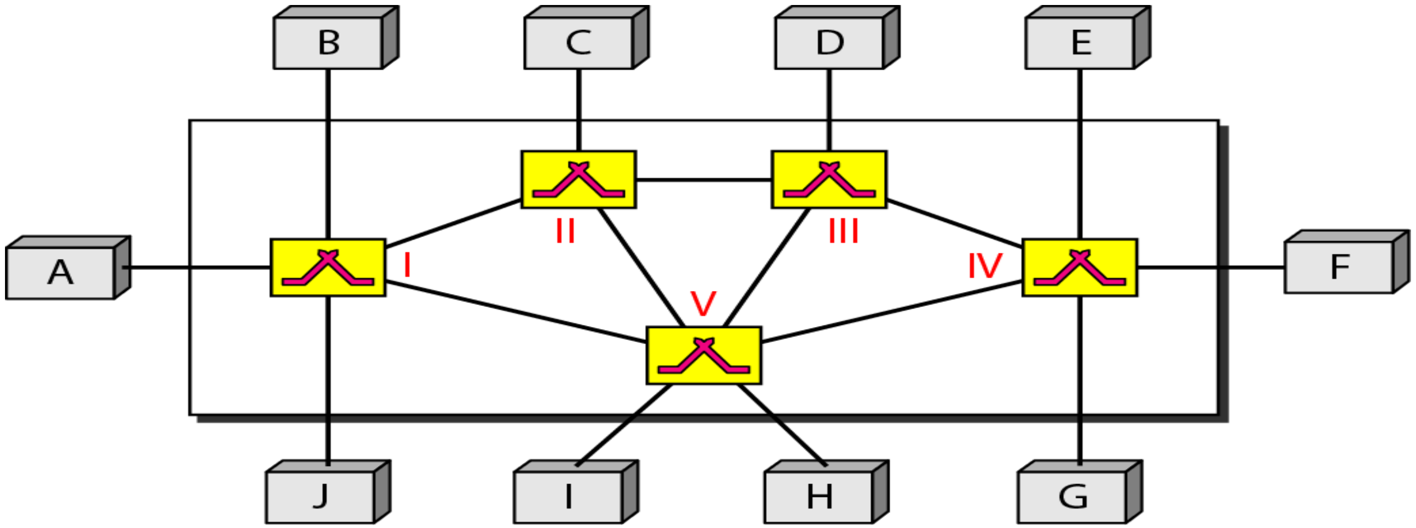
# CSE 4255: Telecommunication
# Lecture 3: Switching

# Why Switching?

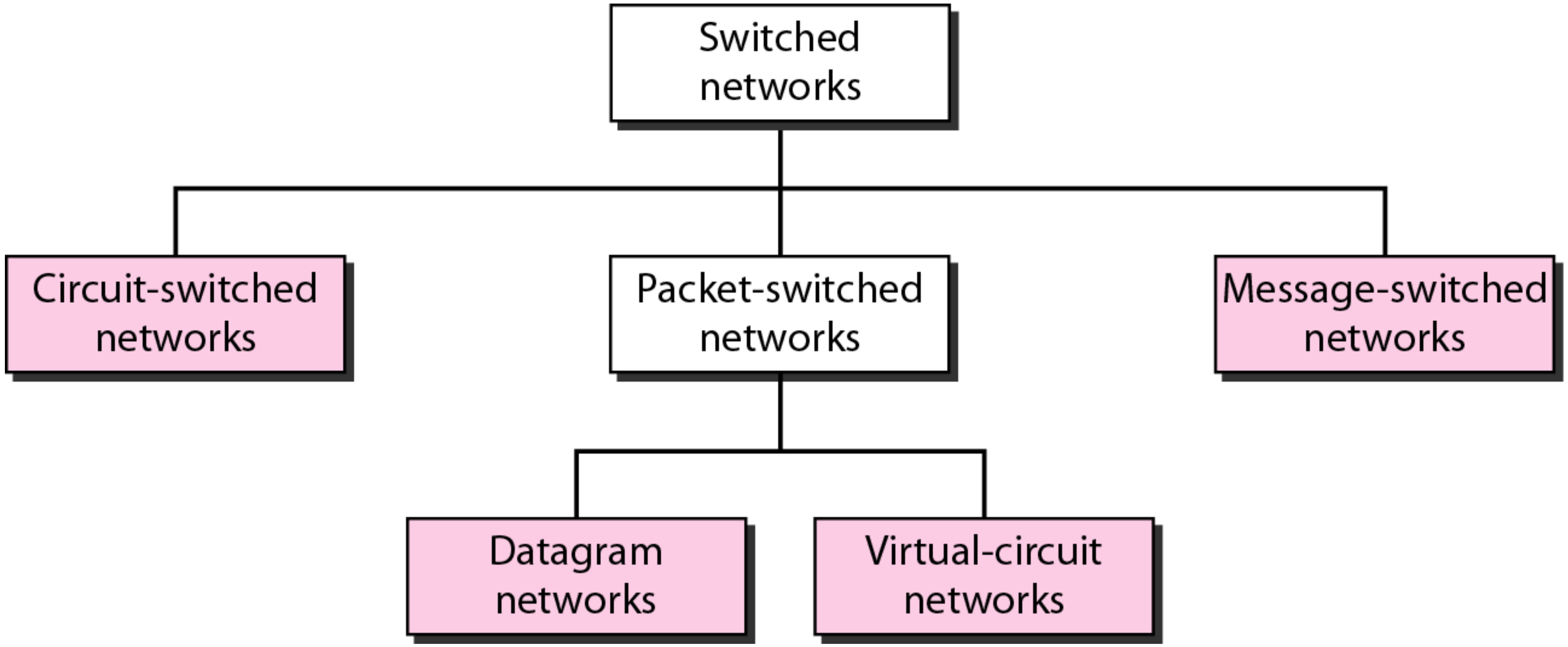
- A network is a set of connected devices. Some of the conventional ways of interconnecting devices are:
  a) Point-to point connection between devices as in a mesh topology.
  b) Connection between a central device and every other device as in star topology.
  c) Bus topology – not practical if the devices are at great distances.
- All these techniques require extensive cabling, dependence on a central server or a central bus.
- The solution to this interconnectivity problem is switching.

# Switching Network:

- A switched network consists of a series of interlinked nodes, called switches.
- Switches are devices capable of creating temporary connections between two or more devices linked to the switch.
- In a switched network, some of these nodes are connected to the end systems (computers or telephones, for example). Others are used only for routing.



# Types of Switching Network:



# Circuit Switching

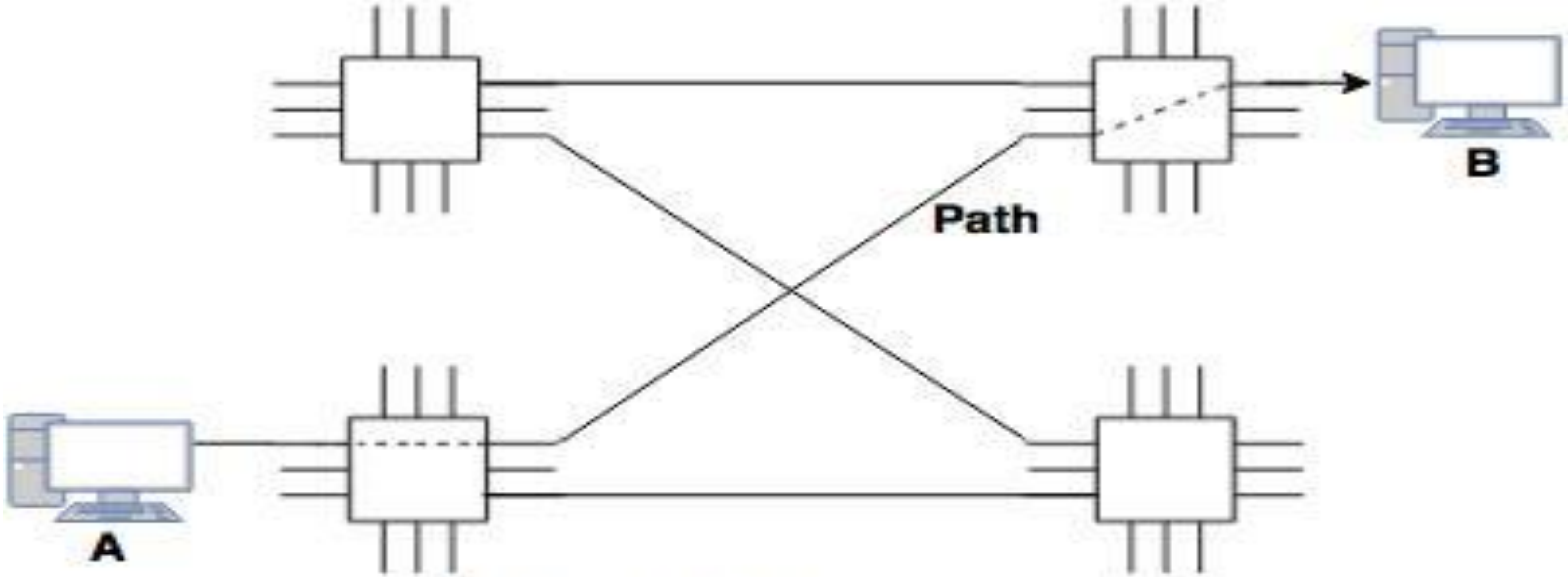



Fig: Circuit Switching

# Circuit Switching

- Circuit switching takes place at the physical layer.
- The actual communication in a circuit-switched network requires three phases: connection setup, data transfer, and connection teardown.
- Before starting communication, the stations must make a reservation for the resources to be used during the communication.
- Data transferred between the two stations are a continuous flow.
- There is no addressing involved during data transfer but there is end-to-end addressing used during the setup phase.

# Message Switching

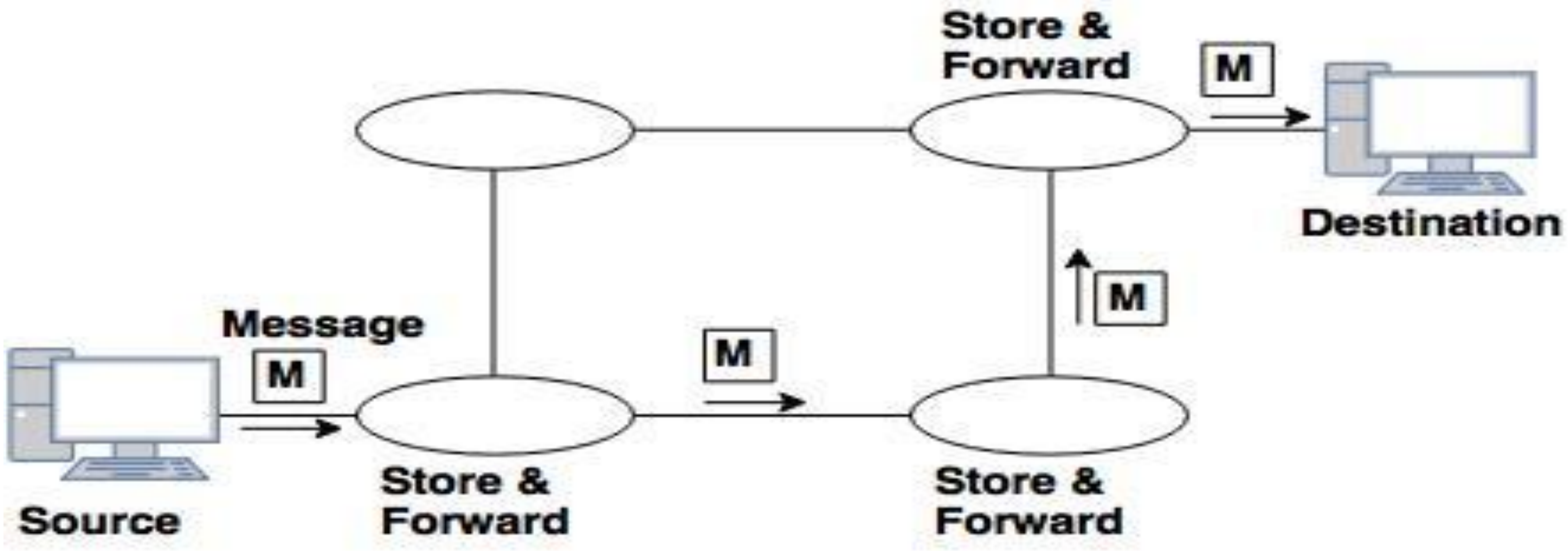



Fig: Message Switching

# Message Switching

- With message switching there is no need to establish a dedicated path between two stations.
- When a station sends a message, the destination address is appended to the message.
- Each node receives the entire message, stores it in its buffer, and then transmits the message to the next node.
- A message will only be delivered if the next hop and the link connecting it are both available, otherwise it’ll be stored indefinitely.
- This type of network is called a store-and-forward network.

# Packet Switching

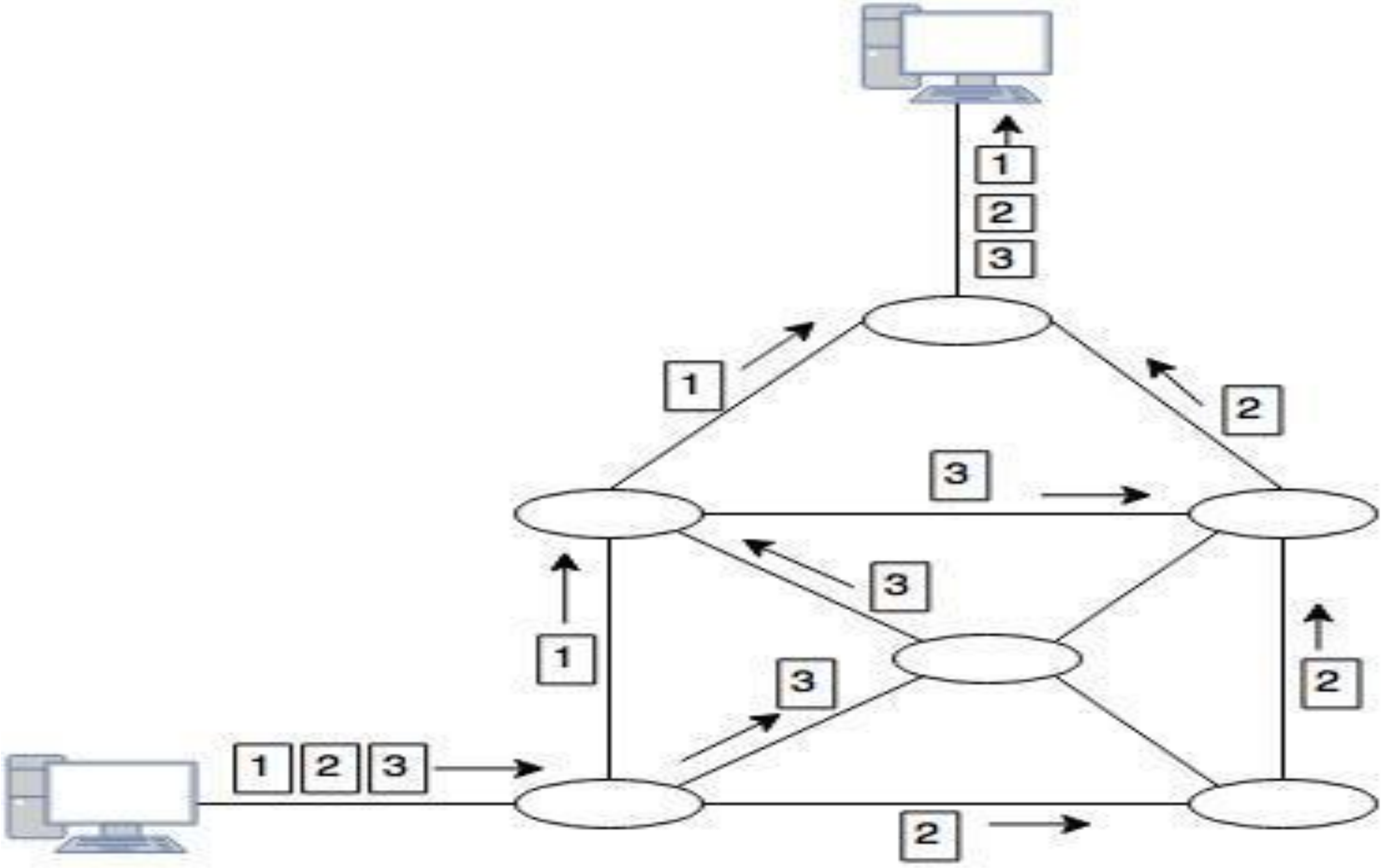



Fig: Datagram Packet Switching

# Packet Switching

- There are two methods of packet switching: Datagram and virtual circuit.
- In both packet switching methods, a message is broken into small parts, called packets.
- Each packet is tagged with appropriate source and destination addresses.
- Since packets have a strictly defined maximum length, they can be stored in main memory instead of disk, therefore access delay and cost are minimized.
- Also the transmission speeds, between nodes, are optimized.
- With current technology, packets are generally accepted onto the network on a first-come, first-served basis. If the network becomes overloaded, packets are delayed or discarded (``dropped").

# Types of Switches

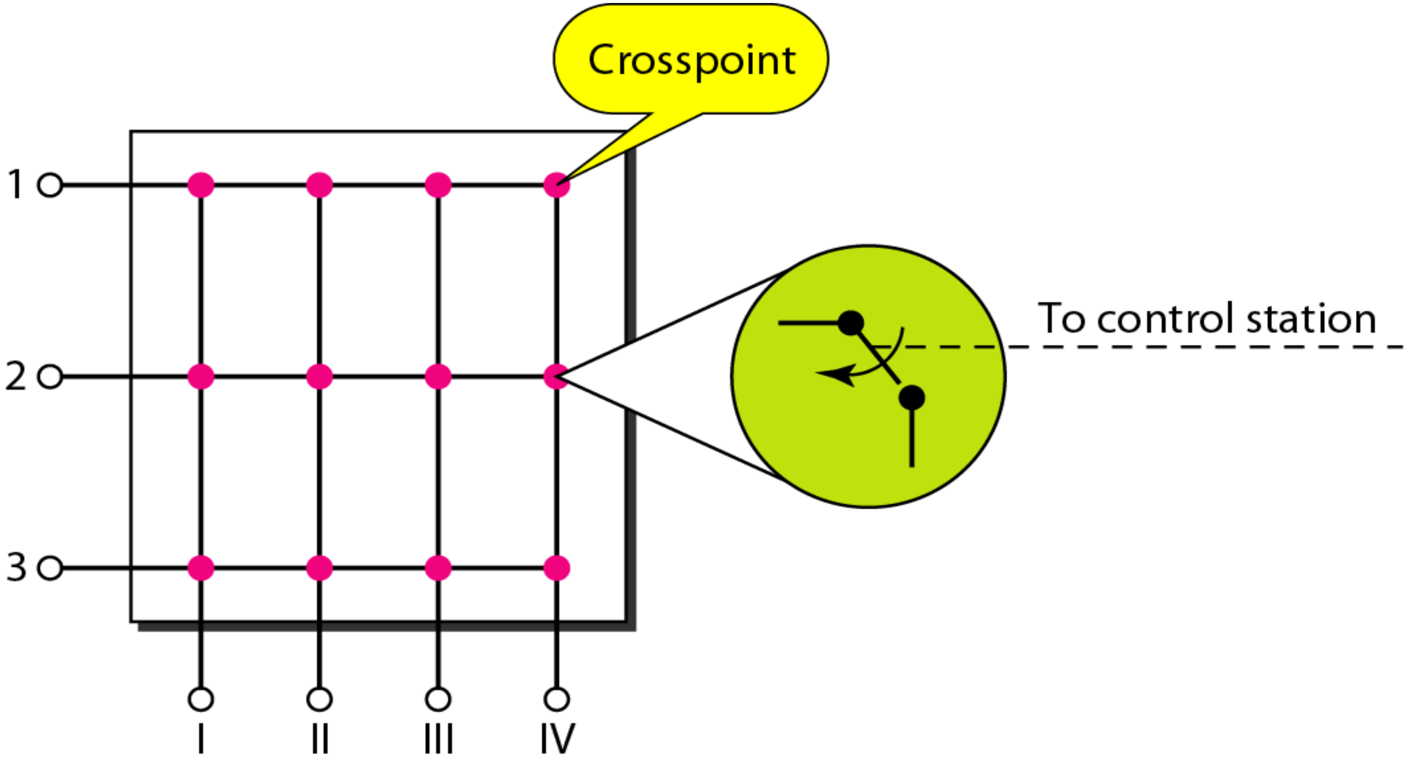
- We use switches in circuit-switched and packet-switched networks.
- Circuit Switching:
  - Spaced Division Switching
    - *Crossbar Switch*
    - *Multi-stage Switch*
  - Time Division Switching
    - *TSI: Time-slot Interchange*
    - *TDM bus*
- Packet Switching:
  - Banyan Switch

# Space-division switch

- In space division switching, the paths in the circuit are separated with each other spatially.
- Different ongoing connections, at a same instant of time, uses different switching paths, which are separated spatially.
- Some of the space switches are
  1. Crossbar switch
  2. Multi-stage switch

# Crossbar Switch

- A crossbar switch connects *n* inputs to *m* outputs in a grid, using electronic microswitches (transistors) at each crosspoint.
- As all the stations are allowed to be connected with all possible connections as long as the called party is free, this Crossbar Switching is called the Non-Blocking.



# Limitations

- The major limitation of this design is the number of crosspoints required.
- To connect *n* inputs to *m* outputs using a crossbar switch requires n x m crosspoints.
- For example, to connect 1000 inputs to 1000 outputs requires a switch with 1,000,000 crosspoints.
- A crossbar with this number of crosspoints is impractical.
- Such a switch is also inefficient because statistics show that, in practice, fewer than 25 percent of the crosspoints are in use at any given time. The rest are idle.

# Multistage Switch-Why?

- The solution to the limitations of the crossbar switch is the multistage switch, which combines crossbar switches in several (normally three) stages.
- In a single crossbar switch, only one row or column (one path) is active for any connection. So we need N x N crosspoints.
- If we can allow multiple paths inside the switch, we can decrease the number of crosspoints.
- Each crosspoint in the middle stage can be accessed by multiple crosspoints in the first or third stage.

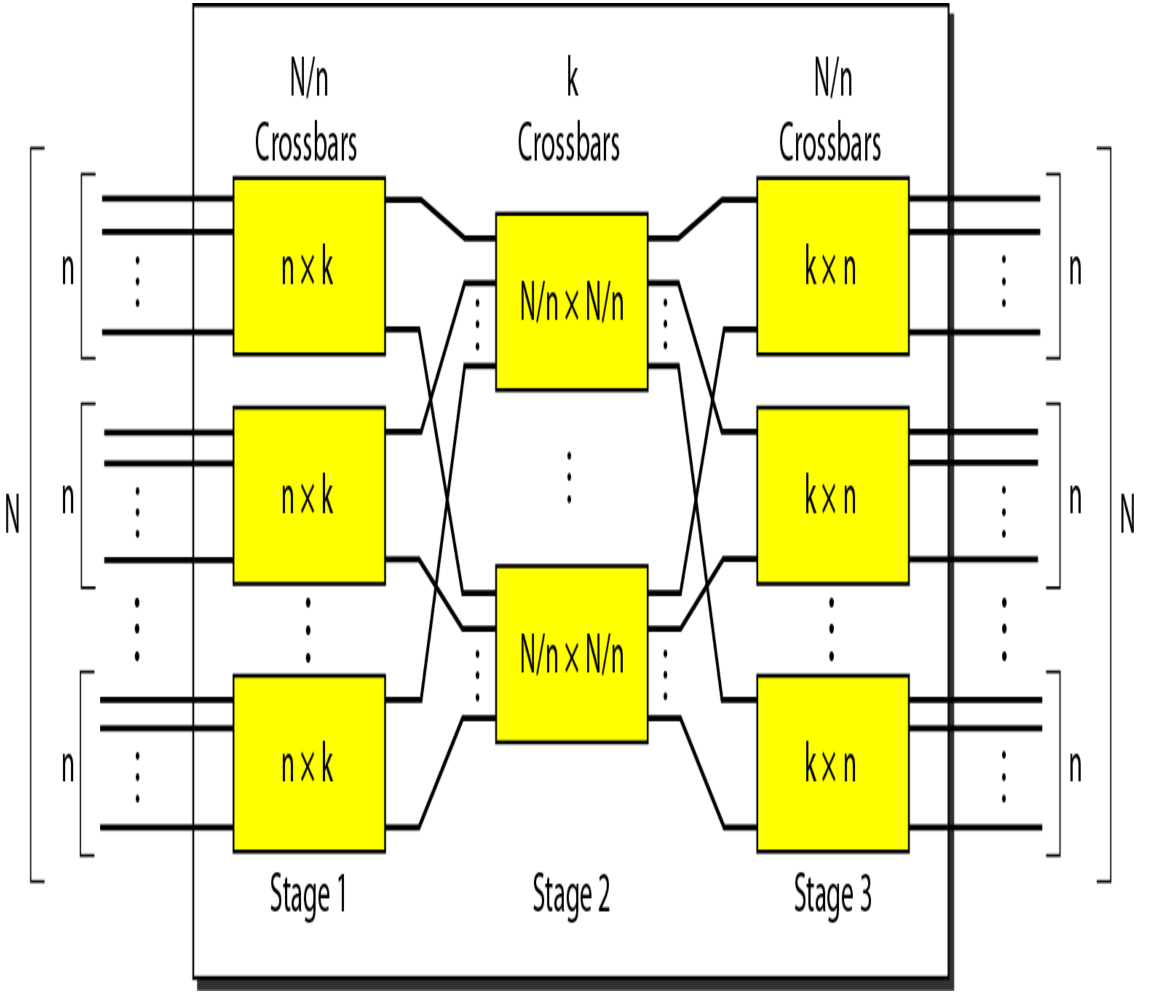
# Multistage Switch Structure

- To design a three-stage switch, we follow these steps:

1. We divide the N input lines into groups, each of n lines. For each group, we use one crossbar of size n x k, where k is the number of crossbars in the middle stage. In other words, the first stage has N/n crossbars of n x k crosspoints.
2. We use k crossbars, each of size (N/n) x (N/n) in the middle stage.
3. We use N/n crossbars, each of size k x n at the third stage.

- We can calculate the total number of crosspoints as follows:

$N/n\ (n \times k) + k\ (N/n \times N/n) + N/n\ (n \times k) = 2kN + k(N/n)^2$



# Example: Multistage Switch

*Design a three-stage, 200 × 200 switch (N = 200) with k = 4 and n = 20.*

*Solution*

*In the first stage we have N/n or 10 crossbars,*
*each of size 20 × 4.*
*In the second stage, we have 4 crossbars,*
*each of size 10 × 10.*
*In the third stage, we have 10 crossbars,*
*each of size 4 × 20.*

*The total number of crosspoints is* $2kN + k(N/n)^2$*, or 2000 crosspoints.*

*This is 5 percent of the number of crosspoints in a single-stage switch (200 × 200 = 40,000).*

# Blocking or Non-blocking

- Blocking
  - Blocking refers to times when one input cannot be connected to an output because there is no path available between them-all the possible intermediate switches are occupied.
  - In a single-stage switch, blocking does not occur because every combination of input and output has its own crosspoint; there is always a path.
  - In a multistage switch, The small number of crossbars at the middle stage creates blocking; if $k < n$ , the switch is blocked.
- Non-blocking
  - Non-blocking permits all stations to connect (in pairs) at once.
  - Clos investigated the condition of nonblocking in multistage switches and came up with the following formula.
  - According to the Clos criterion: $n = (N/2)^{1/2}$



$$k > = 2n - 1$$

# Example- Non-blocking

*Redesign the previous three-stage, 200 × 200 switch, using the Clos criteria with a minimum number of crosspoints.*

Solution

*We let* $n = (200/2)^{1/2}$, *or* $n = 10$.

*We calculate* $k = 2n - 1 = 19$.

*In the first stage, we have 200/10, or 20, crossbars, each with 10 × 19 crosspoints.*

*In the second stage, we have 19 crossbars, each with 20 × 20 crosspoints.*

*In the third stage, we have 20 crossbars each with 19 × 10 crosspoints.*

*The total number of crosspoints is*

$20(10 \times 19) + 19(10 \times 10) + 20(19 \times 10) = 15200.$

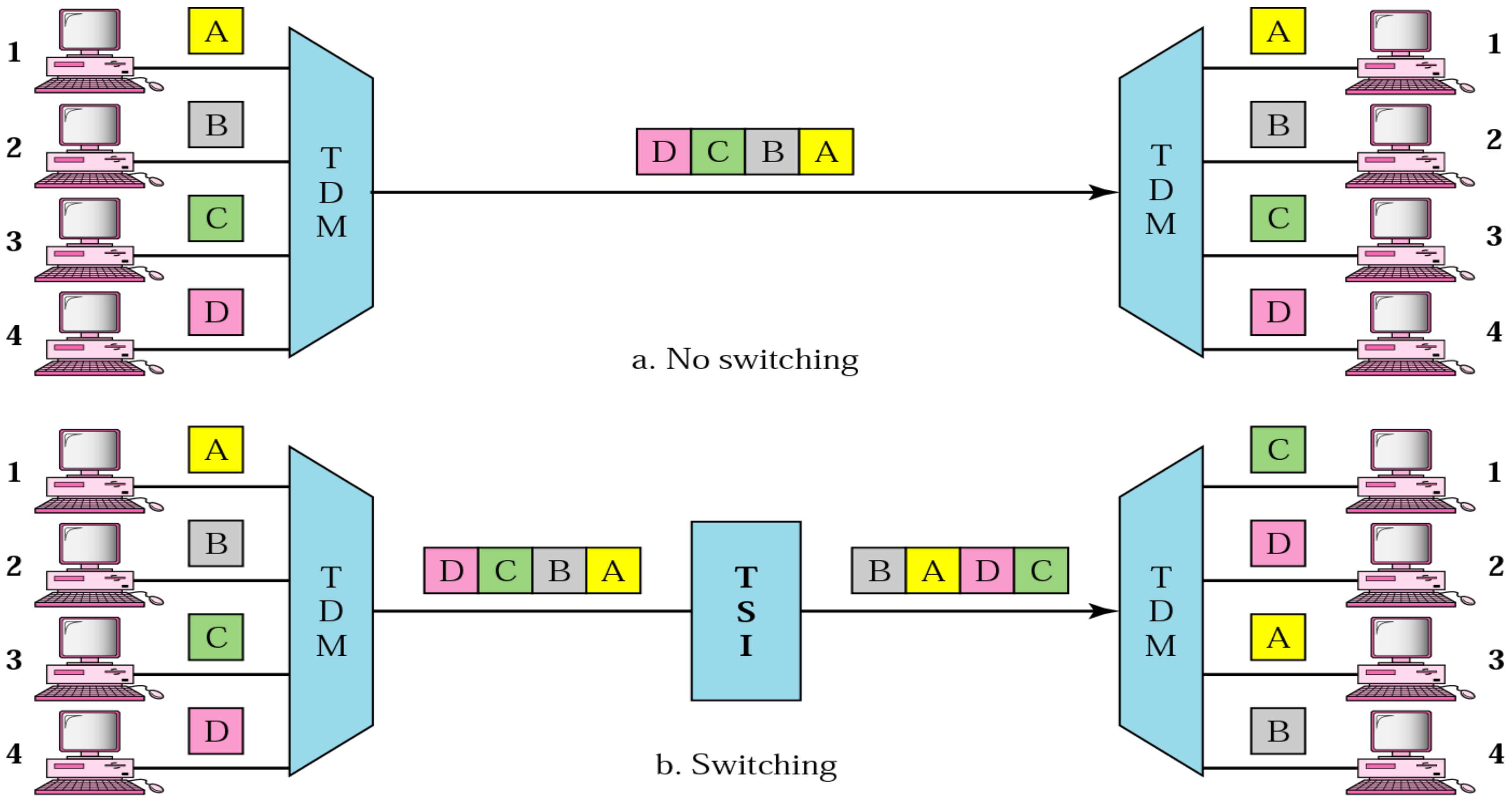
# Time-Division Switch

- Time-division switching uses time-division multiplexing (TDM) inside a switch.
- There are two most popular technology used in TDM:
  1. The time-slot interchange (TSI)
  2. The TDM bus

# Time-slot Interchange

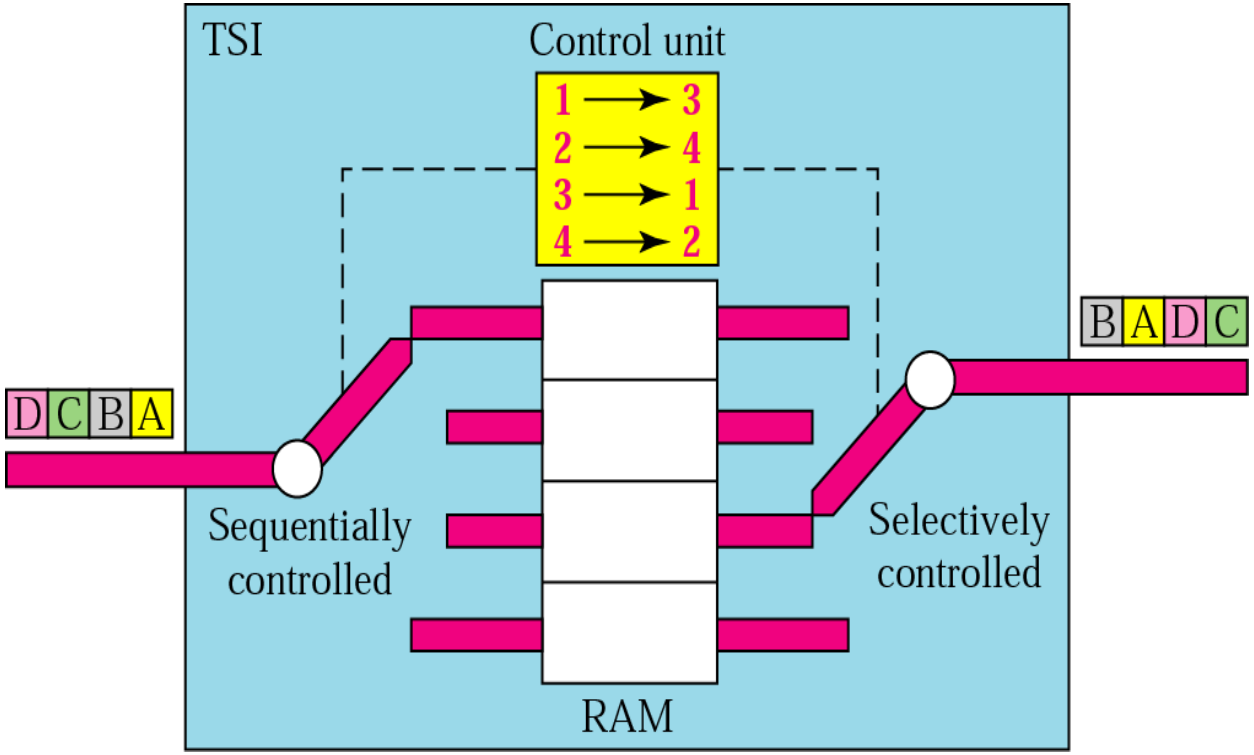
- Consider a system connecting four input lines to four output lines. Imagine that each input line wants to send data to an output line according to the following pattern:

$$1 \rightarrow 3,\ 2 \rightarrow 4,\ 3 \rightarrow 1,\ 4 \rightarrow 2.$$



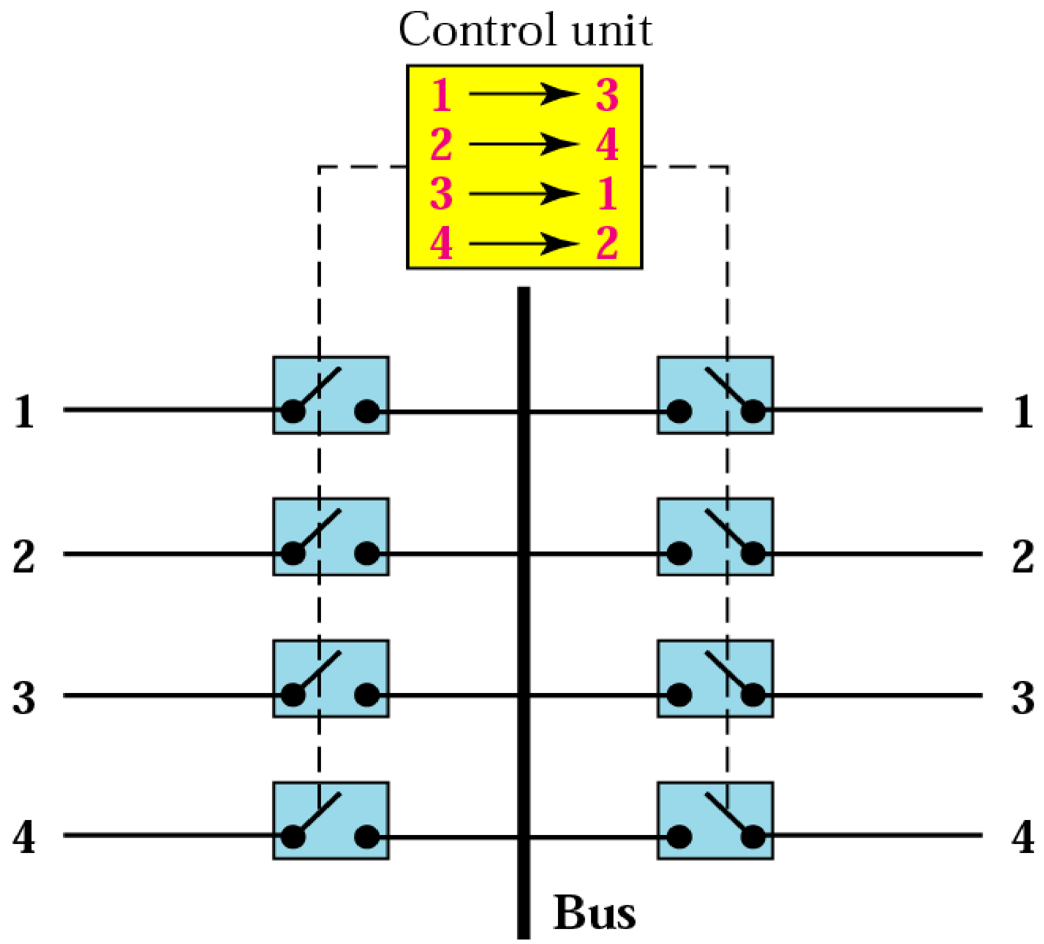
# Structure of TSI

- TSI consisting of random access memory (RAM) with several memory locations.
- The size of each location is the same as the size of a single time slot.
- The number of locations is the same as the number of inputs (in most cases, the numbers of inputs and outputs are equal).
- The RAM fills up with incoming data from time slots in the order received.
- Slots are then sent out in an order based on the decisions of a control unit.



# TDM BUS

- The input and output lines are connected to a high speed bus through input and output gates (microswitches).
- Only one input gate is closed during one time slot and the same time slot only one output gate is closed.
- This pair of gates allows a burst of data to be transferred from one specific input line to one specific output line using the bus.
- The control unit opens and closes the gates according to switching need.



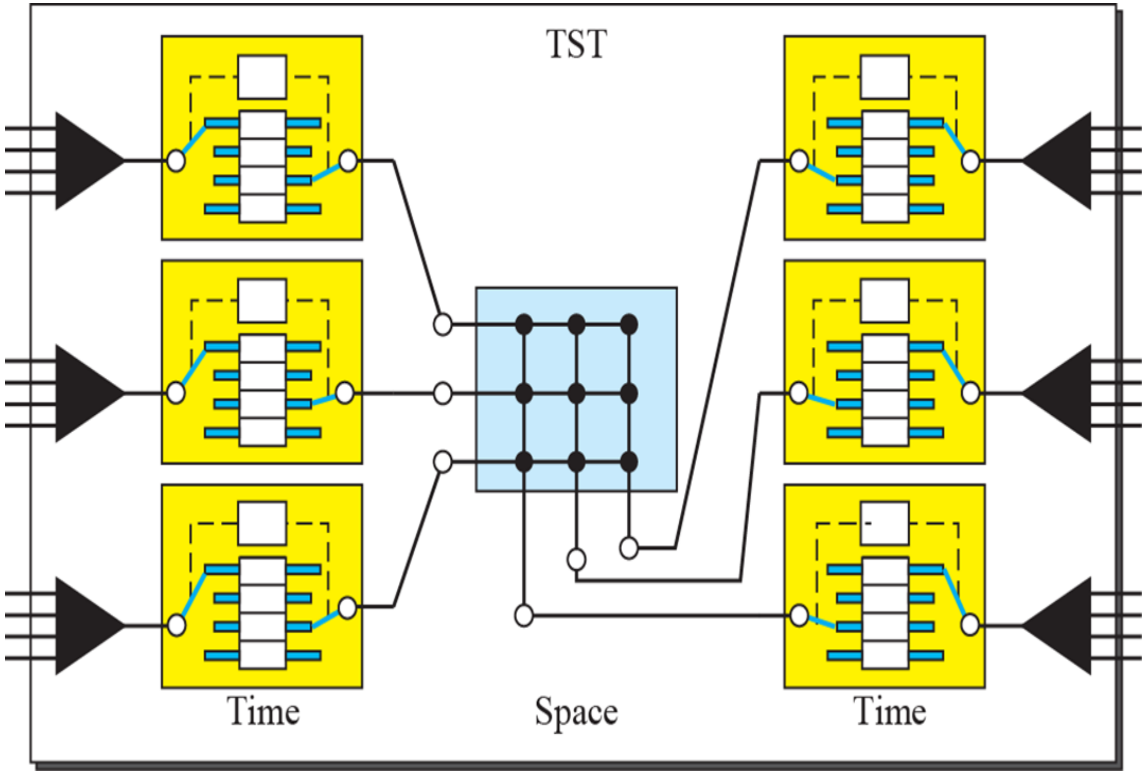
# Comparison Between Time and Space Switches

- Space-Division Switching
  - The advantage of space-division switching is that it is instantaneous.
  - Its disadvantage is the number of crosspoints required to make space-division switching acceptable in terms of blocking.

- Time-Division Switching
  - The advantage of time-division switching is that it needs no crosspoints.
  - Its disadvantage, in the case of TSI, is that processing each connection creates delays. Each time slot must be stored by the RAM, then retrieved and passed on.

# Time-Space-Time (TST) Switch

- A simple TST switch that consists of two time stages and one space stage and has 12 inputs and 12 outputs.
- Instead of one time-division switch, it divides the inputs into three groups (of four inputs each) and directs them to three time-slot interchanges.
- The result is that the average delay is one-third of what would result from using one time-slot interchange to handle all 12 inputs.
- The last stage is a mirror image of the first stage.
- The middle stage is a space- division switch (crossbar) that connects the TSI groups to allow connectivity between all possible input and output pairs (e.g., to connect input 3 of the first group to output 7 of the second group)



# TST Switch: Example-1

- We need a three-stage time-space-time switch with N = 100. We use 10 TSIs at the first and third stages and 4 crossbars at the middle stage.

  a) Draw the configuration diagram.

  b) Calculate the total number of crosspoints.

  c) Calculate the total number of memory locations we need for the TSIs.

b) The total number of cross points = 4(10 × 10) =400

c) Each time slot must be stored by the RAM

In first stage there each TSI has 10 input to multiplexer so the number of slots is 10

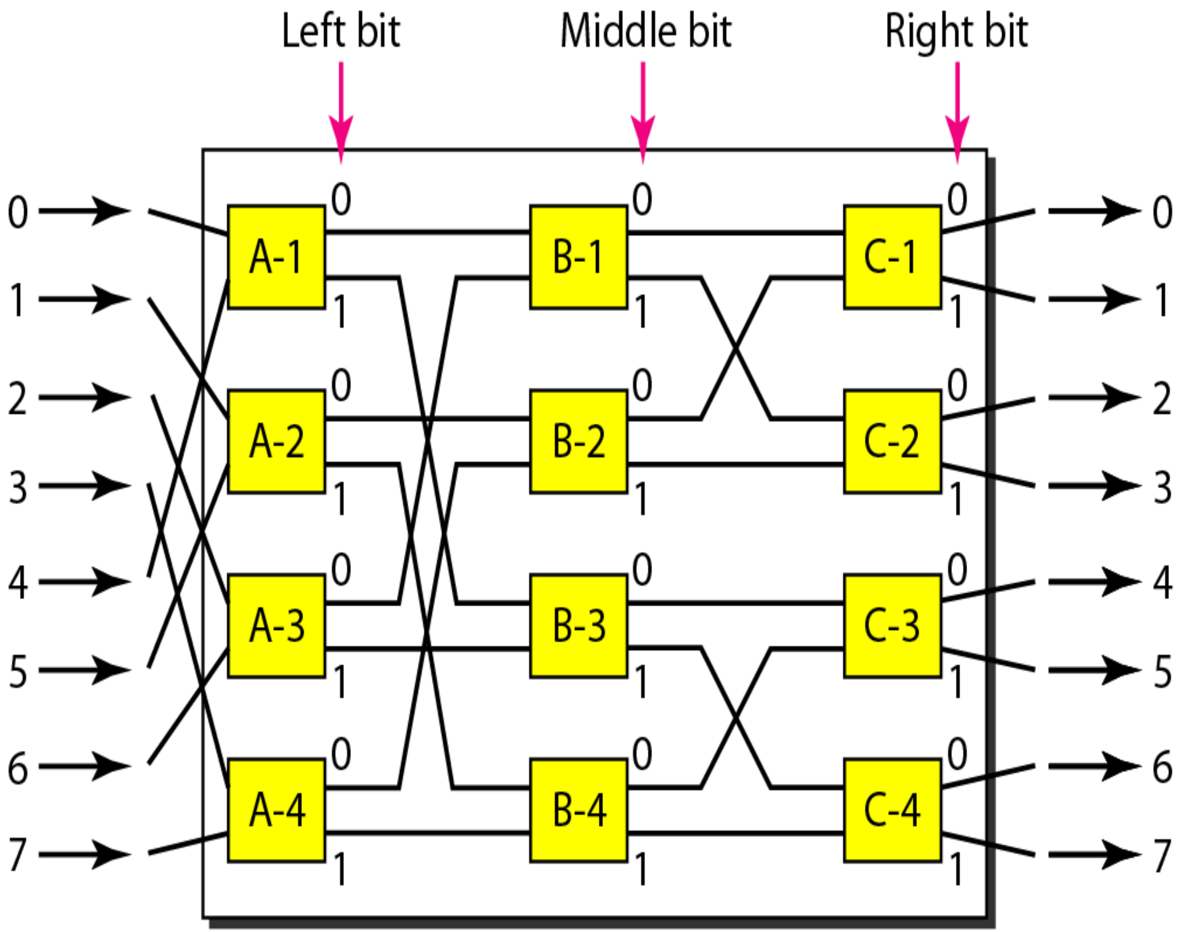
Total of slot number in first stage = 10*10=100

Also total of slot number in third stage = 10*10=100

Total number of memory locations we need for the TSIs= 100+100 =200

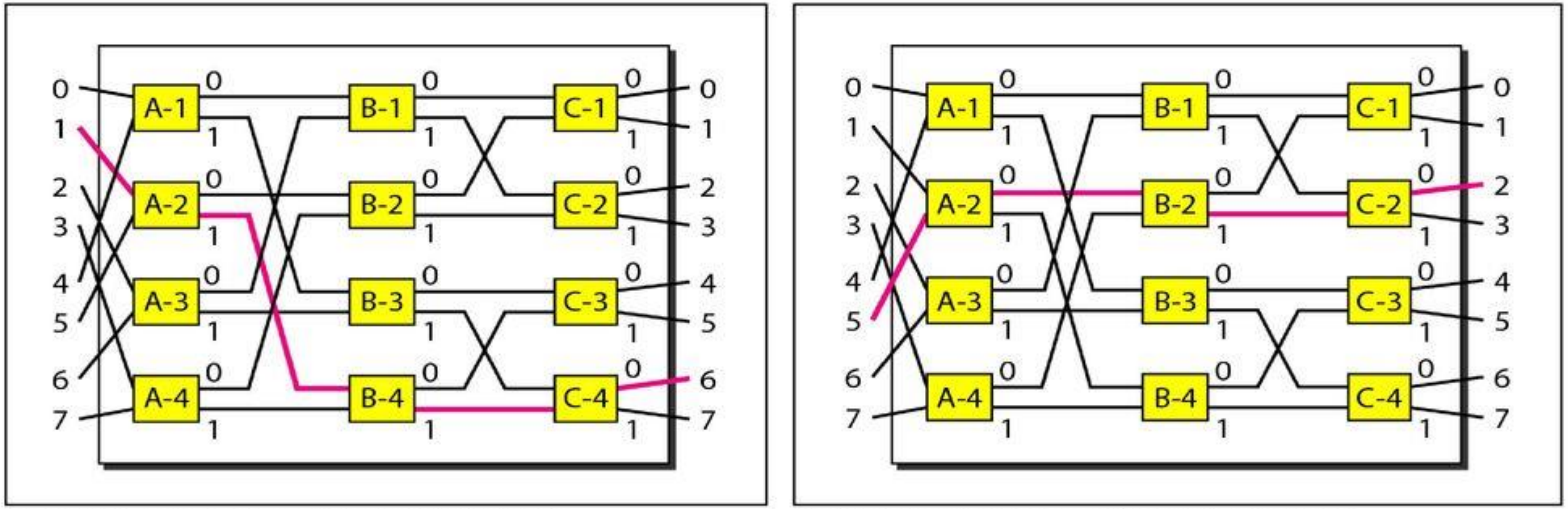
# Banyan Switch

- A banyan switch is a multistage switch with microswitches at each stage that route the packets based on the output port represented as a binary string.
- For n inputs and n outputs, we have $\log_2 n$ stages with n/2 microswitches at each stage.
- The first stage routes the packet based on the high-order bit of the binary string.
- The second stage routes the packet based on the second high-order bit, and so on.
- Here, eight inputs and eight outputs. The number of stages is $\log_2(8) = 3$.



# Banyan Switch

**Figure 8.25** ***Examples of routing in a banyan switch***




a. Input 1 sending a cell to output 6 (110)

b. Input 5 sending a cell to output 2 (010)