Static Vs Dynamic Networks

- Based on connectivity and control networks can be divided into two classes
  - Static networks – that don't change dynamically (e.g., trees, rings, meshes (not crossbars))
  - Dynamic networks – that change interconnectivity dynamically

Dynamic Networks

- Implemented with switched channels
  - dynamically configured to meet the communication needs of user programs
  - E.g: system buses, crossbar switches, multistage networks

Dynamic Interconnection Networks

- Switches are used to provide an **indirect connection between nodes**.

- From the processors' point of view, such a network forms an interconnection unit into which data can be sent and from which data can be received.

- A dynamic network consists of **switches that are connected by physical links**.

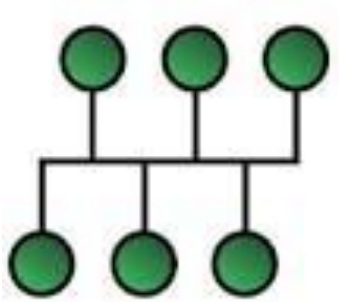
Dynamic Interconnection Networks

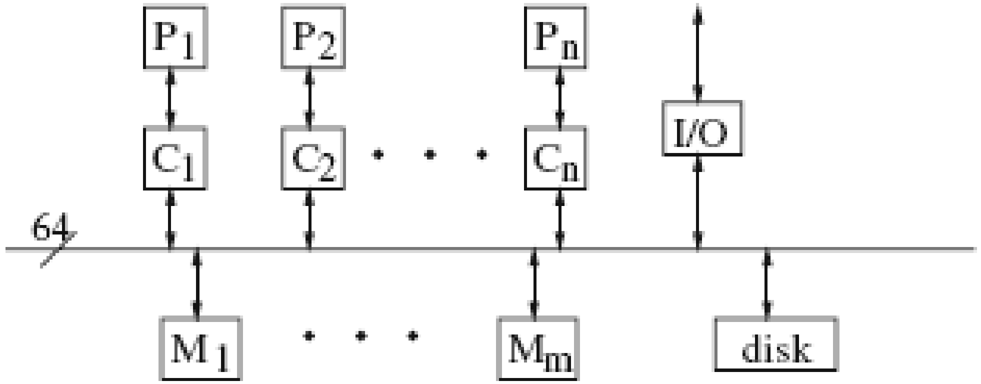
- For a message transmission from one node to another node, the **switches can be configured dynamically** such that a connection is established.

- Popular forms are **bus networks**, **multistage networks**, and **crossbar networks**.

Bus Networks

- A bus essentially consists of a set of wires which can be used to transport data from a sender to a receiver.
- **At each point in time**, only **one data transport** can be performed via the bus.





Bus Networks

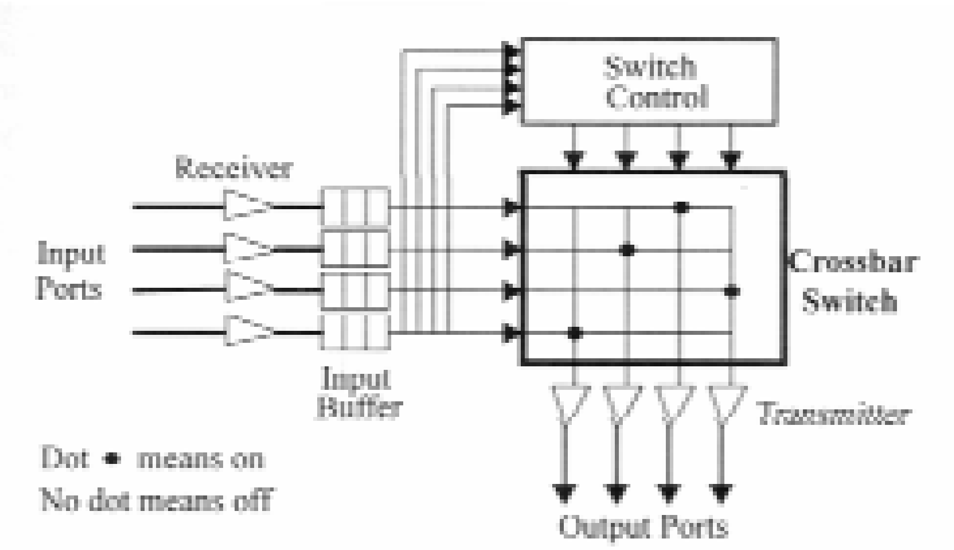
- When several processors attempt to **use the bus simultaneously**, a **bus arbiter** is used for the **coordination**.

- Bus networks are typically used for a **small number of processors** only. **Why????**

Crossbar Networks

- An $n \times m$ crossbar network has n inputs and m outputs.
- For a system with a **shared address space**, the **input nodes** may be **processors** and the **outputs** may be **memory modules**.
- For a system with a **distributed address space**, both the **input** nodes and the **output** nodes may be **processors**.
- For each request from a specific input to a specific output, a **connection in the switching network is established**.

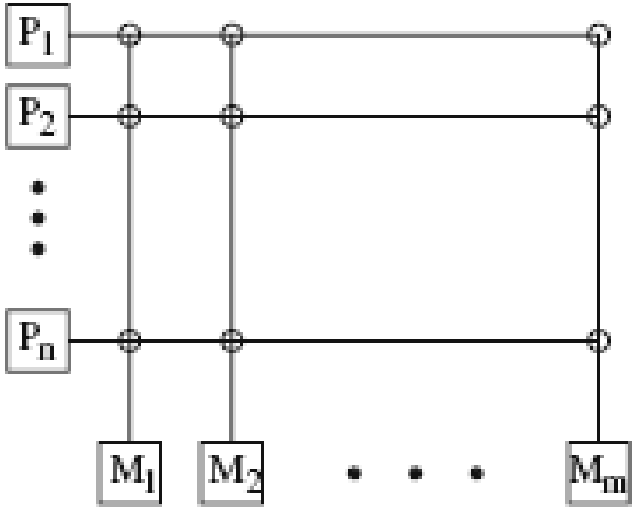
Network Components

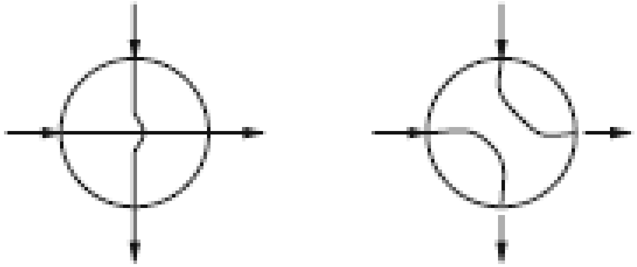


Crossbar Networks

- Depending on the specific input and output nodes, the switches on the connection path can have different states:
    - **Straight change**.
    - **Direction change**.

- Crossbar networks are used only for a **small number** of processors because of the **large hardware overhead required**.

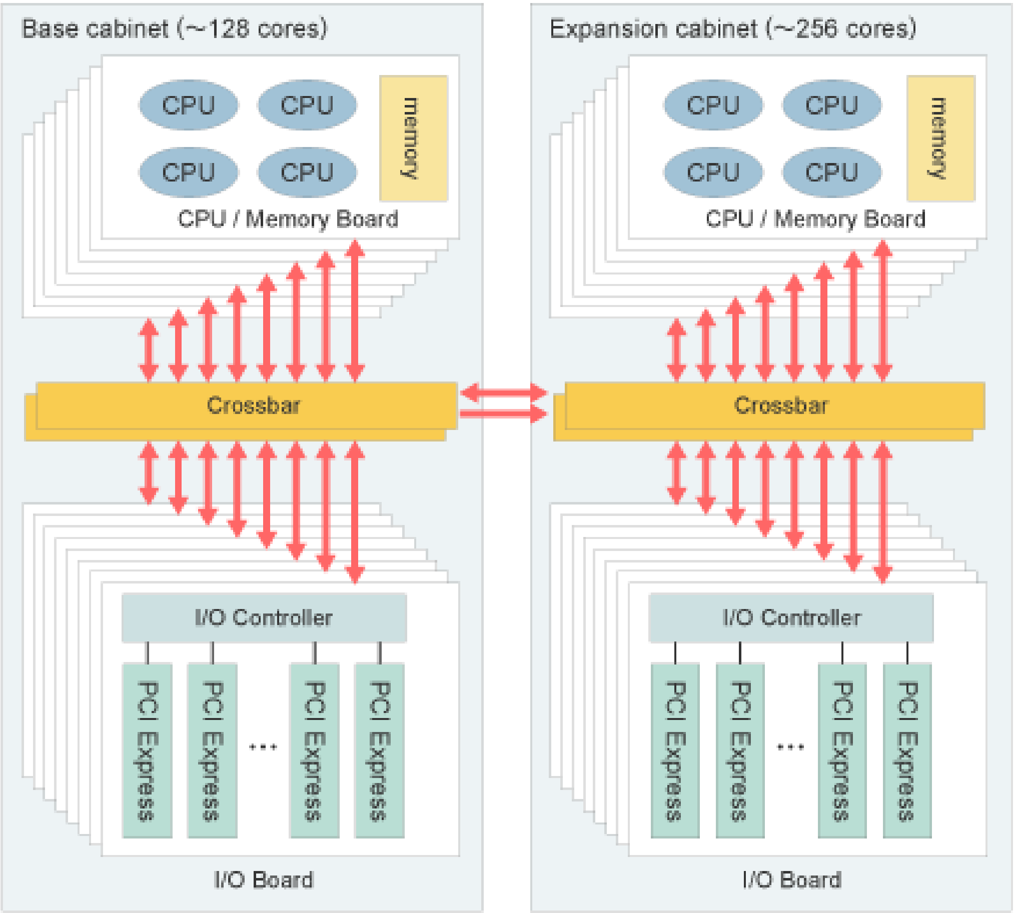




Crossbar Networks

- SPARC Enterprise M9000 from Fujitsu use crossbar interconnect.

- SPARC is a mainframe with up to 256 processors and 4TB RAM.



SPARC Enterprise M9000

Multistage Switching Networks

- **Multistage switching networks** consist of several stages of switches with **connecting wires between neighboring stages**.
- The network is used to connect input devices to output devices.
- Input devices are typically the processors of a parallel system.
- Output devices can be processors (for distributed memory machines) or memory modules (for shared memory machines).
- The goal is to **obtain a small distance for arbitrary pairs** of input and output devices to ensure **fast communication**.

Multistage Switching Networks

- The **internal connections** between the stages can be **represented as a graph**.

- **Switches** are represented by **nodes** and **wires** between switches are represented by **edges**.

- Input and output devices can be represented as **specialized nodes** with edges **going into the actual switching** network graph.

- The **construction** of the switching **graph** and the **degree** of the **switches** used are important characteristics of multistage switching networks.

Multistage Switching Networks

- Multistage networks have the **advantages of a constant node degree**.

- Usually the network is **built** from **smaller k x k switching elements**. These switching elements are arranged in $\log_k N$;

- Where N is the number of inputs (and outputs) in the switch, and **k is a small integer usually 2** (if 2 the switching element is a **2 x 2 switch**).

Regular multistage interconnection networks

- They characterized by a **regular construction method** using the **same degree** of **incoming** and **outgoing** wires for all switches.

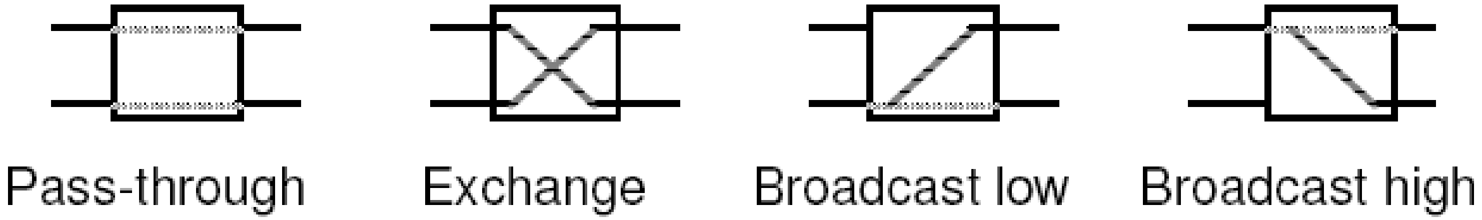
- The switches are arranged in stages such that neighboring stages are connected by fixed interconnections.

- **Connections** from input devices to output devices are performed by **selecting a path from a specific input device to the selected output device**.

- And setting the switches on the path such that the connection is established.

Regular multistage interconnection networks

- Popular regular multistage networks are **omega**, **baseline (Clos)**, **butterfly**, **Beneš**, and **fat-tree** networks.
- These networks **use 2× 2 crossbar** switches which are arranged in **log n** stages.
- Each switch can be in **one of four states**.

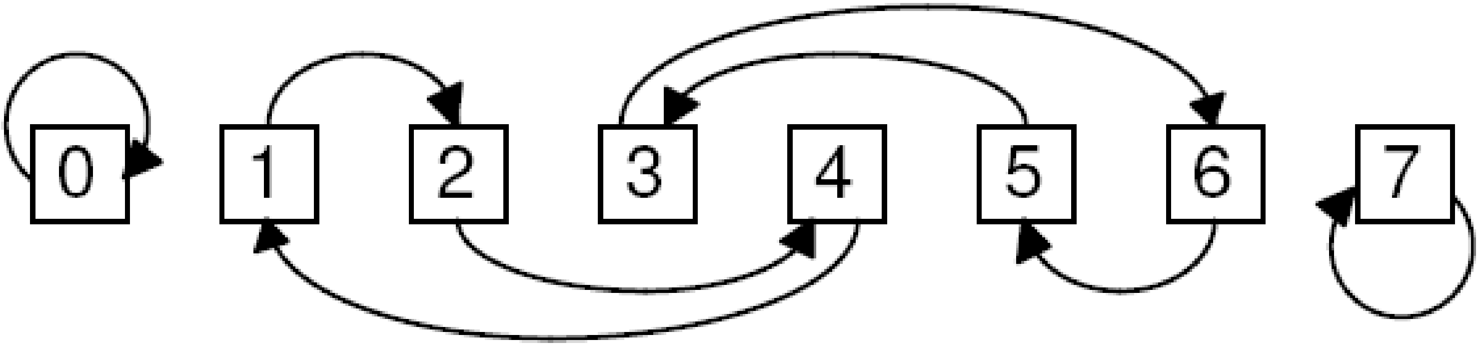


Omega Network

- An **$n \times n$ omega** network is based on **2×2 crossbar switches**.
- These switches are arranged in **$\log n$** stages.
- Such that each stage contains **$n/2$ switches** where **each switch** has **two input links and two output links**.
- Thus, there are **$(n/2) \cdot \log n$** switches in total, with $\log n \equiv \log_2 n$.
- Each switch can be in one of four states,
- **Permutation function** describing the connection between neighboring stages is the same for all stages.

Omega Network (Permutation function)

$$j = \begin{cases} 2i, & \text{for } 0 \le i \le n/2 - 1, \\ 2i + 1 - n, & \text{for } n/2 \le i \le n - 1. \end{cases}$$

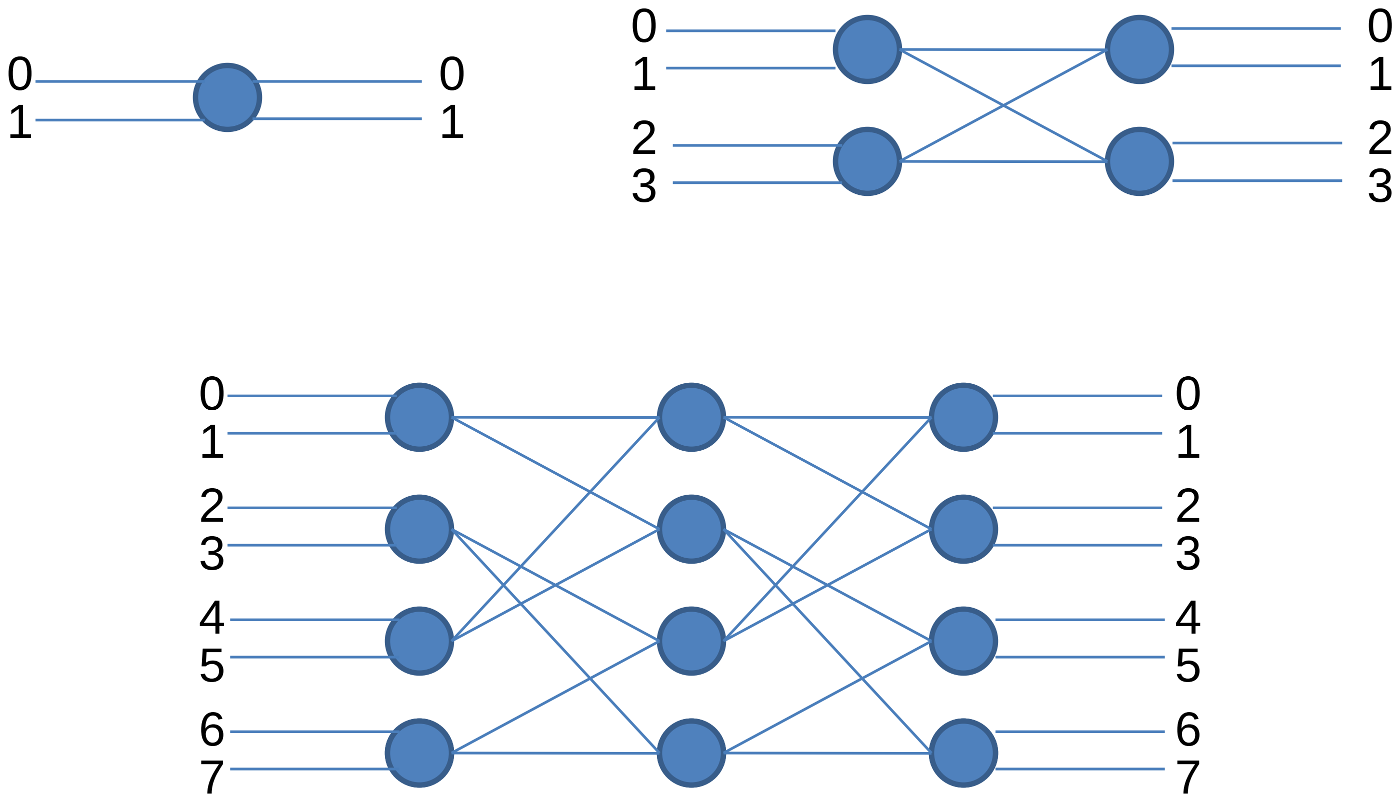


Omega Network (Permutation function)

$$j = \begin{cases} 2i, & \text{for } 0 \le i \le n/2 - 1, \\ 2i + 1 - n, & \text{for } n/2 \le i \le n - 1. \end{cases}$$

- **For n =8 number of switches = n/2 = 4 stages = 3**
- **I = 0 ➔ j = 2 x 0 = 0** **I = 4 ➔ j = 2 x 4 + 1 – 8 = 1**
- **I = 1 ➔ j = 2 x 1 = 2** **I = 5 ➔ j = 2 x 5 + 1 – 8 = 3**
- **I = 2 ➔ j = 2 x 2 = 4** **I = 6 ➔ j = 2 x 6 + 1 – 8 = 5**
- **I = 3 ➔ j = 2 x 3 = 6** **I = 7 ➔ j = 2 x 7 + 1 – 8 = 7**
- **For n = 4 number of switches = n/2 = 2 stages = 2**
- **I = 0 ➔ j = 2 x 0 = 0** **I = 2 ➔ j = 2 x 2 + 1 – 4 = 1**
- **I = 1 ➔ j = 2 x 1 = 2** **I = 3 ➔ j = 2 x 3 + 1 – 4 = 3**
- **For n = 2 number of switches = n/2 = 1 stages = 1**
- **I = 0 ➔ j = 2 x 0 = 0** **i= 1 ➔ j = 2 x 1 + 1 – 2 = 1**

Omega Network (Permutation function)



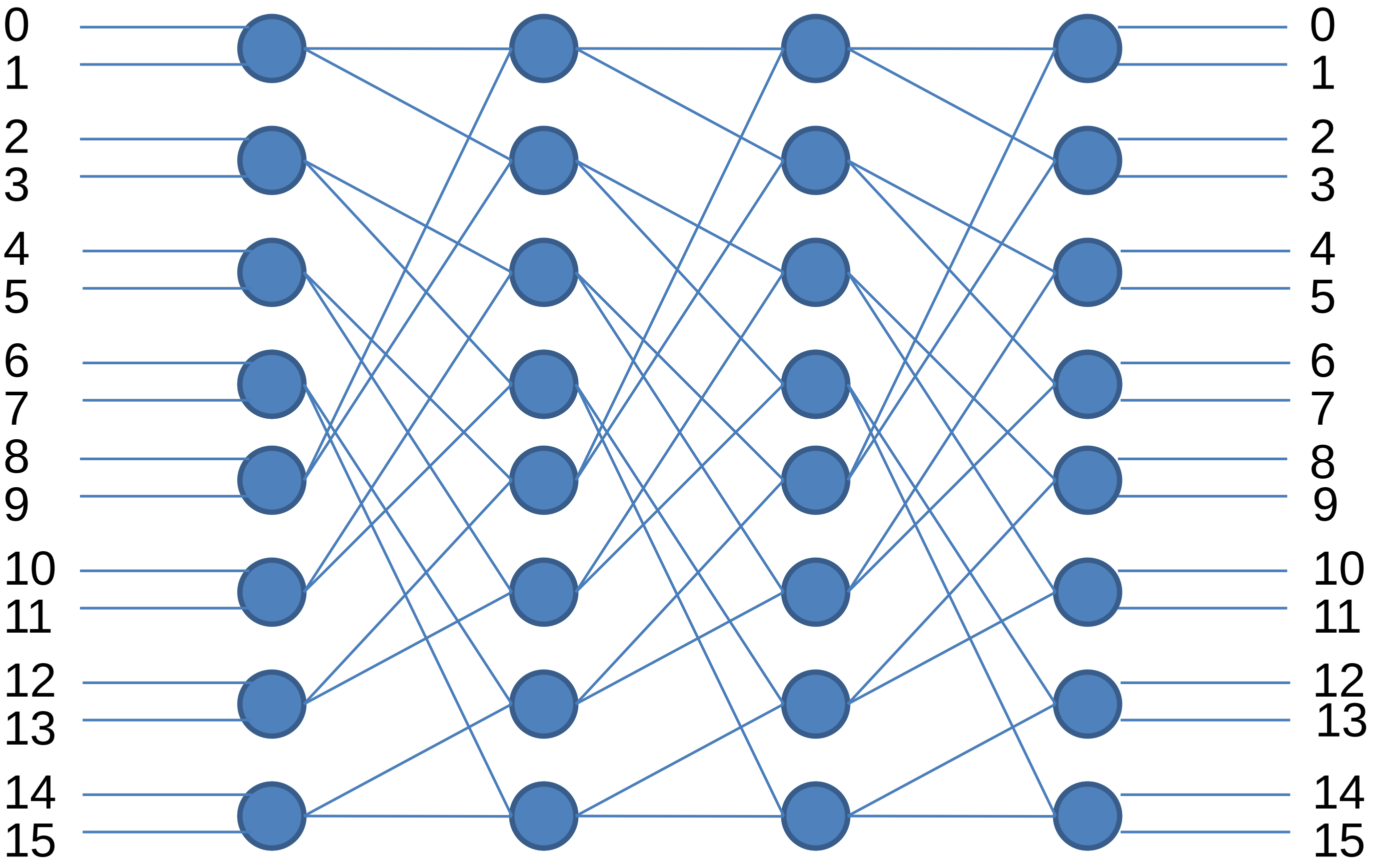
Omega Network (Permutation function)

$$j = \begin{cases} 2i, & \text{for } 0 \le i \le n/2 - 1, \\ 2i + 1 - n, & \text{for } n/2 \le i \le n - 1. \end{cases}$$

- For n =16 number of switches = n/2 = 8 stages = 4
- I = 0 ➔ j = 2 x 0 = 0
- I = 1 ➔ j = 2 x 1 = 2
- I = 2 ➔ j = 2 x 2 = 4
- I = 3 ➔ j = 2 x 3 = 6
- I = 4 ➔ j = 2 x 4 = 8
- I = 5 ➔ j = 2 x 5 = 10
- I = 6 ➔ j = 2 x 6 = 12
- I = 7 ➔ j = 2 x 7 = 14

I = 8 ➔ j = 2 x 8 + 1 – 16 = 1
I = 9 ➔ j = 2 x 9 + 1 – 16 = 3
I = 10 ➔ j = 2 x 10 + 1 – 16 = 5
I = 11 ➔ j = 2 x 11 + 1 – 16 = 7
I = 12 ➔ j = 2 x 12 + 1 – 16 = 9
I = 13 ➔ j = 2 x 13 + 1 – 16 = 11
I = 14 ➔ j = 2 x 14 + 1 – 16 = 13
I = 15 ➔ j = 2 x 15 + 1 – 16 = 15

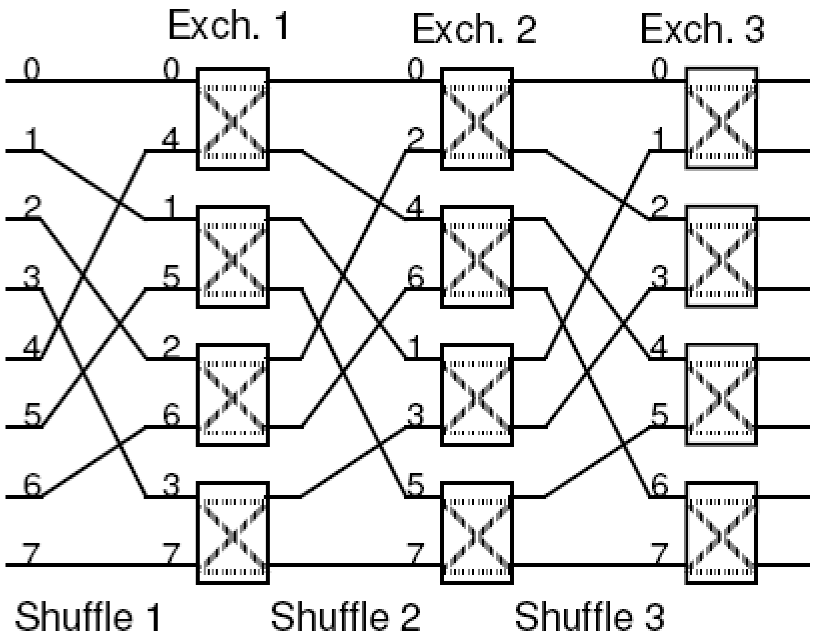
Omega Network (Permutation function)



Omega routing function

- The routing function from input line i to output line **j considers only j** and the **stage number s**, where $s \in [0, \log_2 n-1]$.

- In a stage s switch, if the s+1th MSB (most significant bit) of j is 0, the data is routed to the upper output wire, otherwise it is routed to the lower output wire.



Omega routing function

- One of the **main advantages** of the Omega network (banyan networks in general) **is the ease of routing**. Routing is completely distributed.

- There is **no need for a central controller**, when the message reaches the i^{th} stage, the switch examine the i^{th} bit in the destination addresses, according to this bit, the switch is set as either straight or cross.

- One of the **main disadvantages** of the Omega network is that there is a **fixed distance between any two nodes** For binary switches the distance between any two nodes is $\log_2 N$.

Omega routing function

- That means there is **no concept of locality** or two nodes close to each other.

- That is suitable for systems that require **uniform communication**. However, in **systems** where there is **clustering**, it is much more **advantageous to use a network that has a concept of locality**.

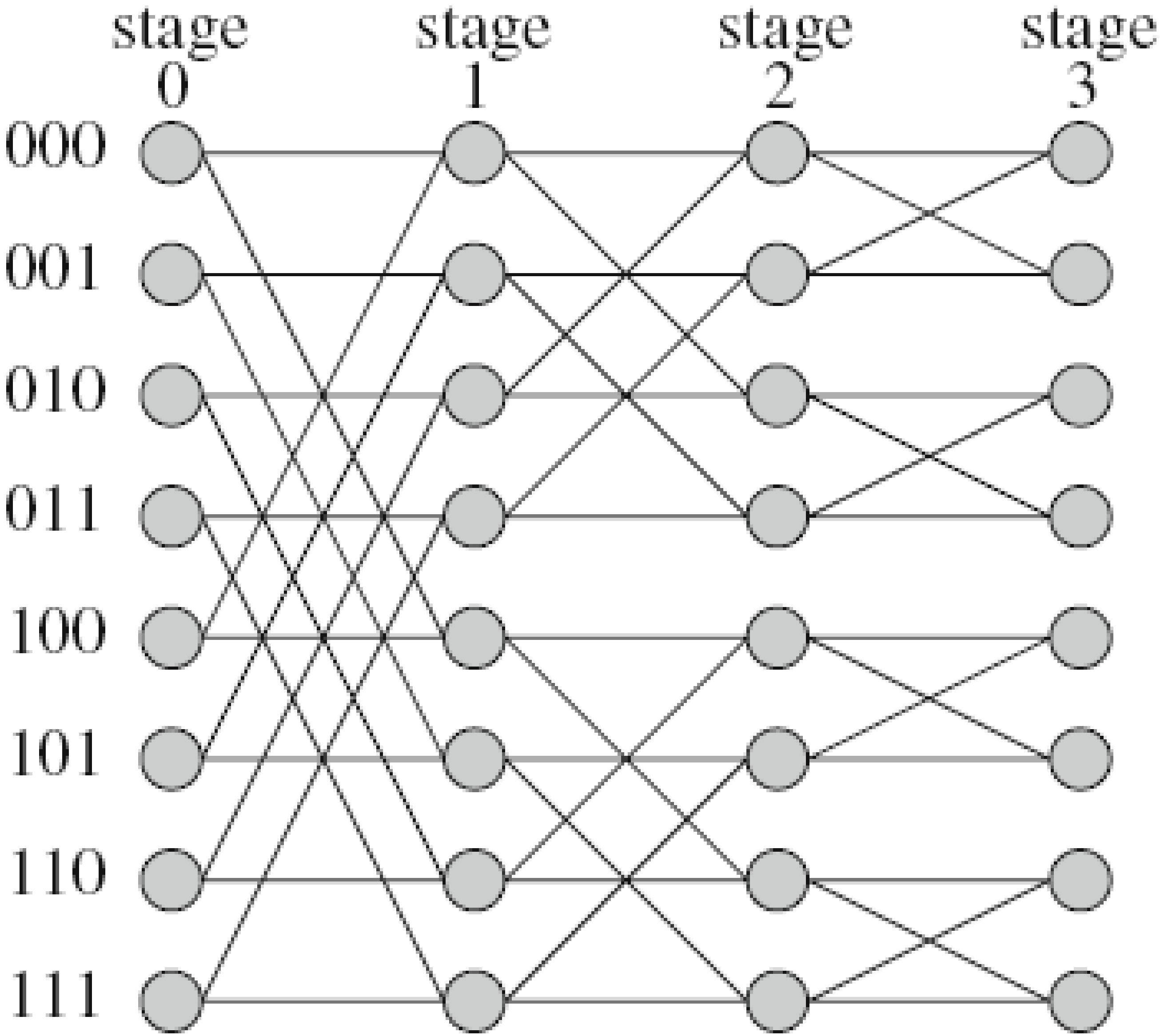
Butterfly Network

- The two outgoing edges from any switch <x, s> are as follows:

- There is an edge from switch <x, s> to switch <y, s+1>
  - if (i) x = y
  - or (ii) x XOR y has exactly one 1 bit,
  - Which is in the $(s+1)^{th}$ MSB. For stage s, apply the rule above for M/2s switches.

- Whether the two incoming connections go to the upper or the lower input port is not important because of the routing function.

Butterfly Network



Butterfly Network

- **Butterfly routing function**:

- In a stage s switch, if the $s+1^{th}$ MSB of j is 0, the data is routed to the upper output wire, otherwise it is routed to the lower output wire.

- Observe that for the **Butterfly** and the **Omega** networks, the paths from the different inputs to any one output form a spanning tree.

- This implies that **collisions will occur** when data is **destined to the same output line**.

Butterfly Network

- Butterfly network has a **simple recursive structure**.
- It can host **multiple source nodes**.
- It can give the full advantages of a **high number of routes**.
- It does not have path diversity, in **high load** on the **network congestion situation** which is **more serious**.

Butterfly Network

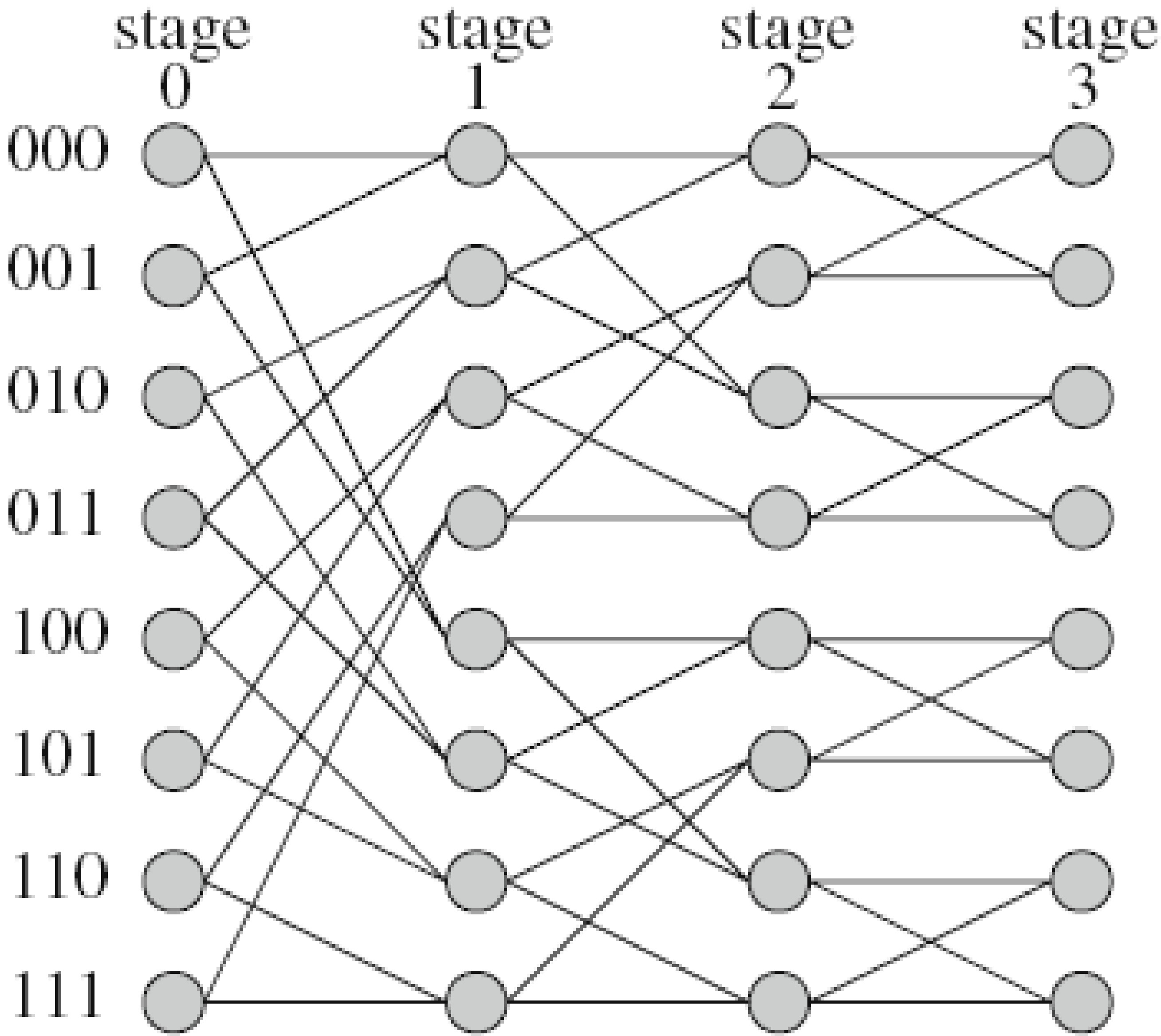
- The disadvantages:
- Firstly, it **lacks of path diversity**. There is only one path from a source node to a destination node.
- Secondly, long wires are **inevitable**. **Long wires must transverse half of the diameter of the network.**

Baseline Network

- The k-dimensional baseline network has the same number of nodes, edges, and stages as the butterfly network.

- Neighboring stages are connected as follows: Node (α, i) is connected to node $(\beta, i + 1)$ for $0 \leq i < k$ if and only if
  - β results from α by a cyclic right shift on the last $k - i$ bits of α or.
  - β results from α by first inverting the last (rightmost) bit of α and then performing a cyclic right shift on the last $k - i$ bits.

Baseline Network



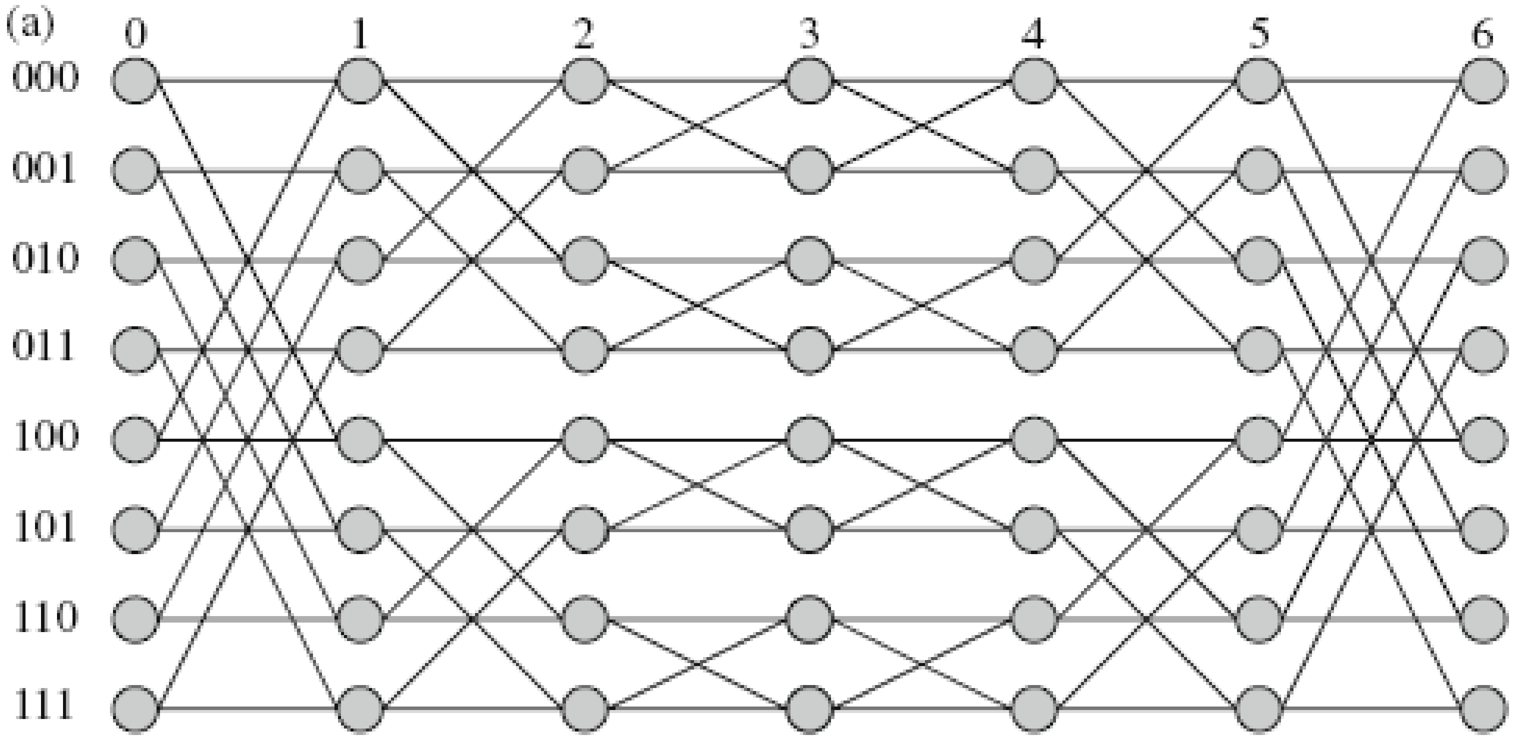
Beneš Network

- The k-dimensional **Beneš** network is constructed from **two k-dimensional butterfly networks** such that the first **k + 1 stages are a butterfly network** and the **last k + 1 stages are a reverted butterfly network**.

- The last stage (k + 1) of the first butterfly network and the first stage of the second (reverted) butterfly network are **merged**.

- In total, the k-dimensional **Beneš** network has 2k + 1 stages with 2^k switches in each stage.

Beneš Network



Beneš Network Network

- BeneŠ network has a **good path diversity**, which can provide **multiple data link between each pair of nodes**, so it has a good solution to the problem of network congestion.

- However, the routing process must use a lot of middle-class switching module, **which needs more access lines resulting in much higher routing delay than the butterfly network**, and the network overhead is larger than the butterfly network.

- **Large hardware overhead required.**

Routing and Switching

- A routing algorithm determines a path in a given network from a source node A to a destination node B.
- The path consists of a sequence of nodes such that neighboring nodes in the sequence are connected by a physical network link.
- **A set of messages is in a deadlock situation** if each of the messages is supposed to be transmitted over a **link that is currently used by another message of the set**.

Routing and Switching

- The following issues are important for the path selection:

- **Network topology**: The topology of the network determines which paths are available in the network to establish a connection between nodes A and B.

- **Network contention**: Contention occurs when two or more messages should be transmitted at the same time over the same network link, thus leading to a delay in message transmission.

Routing and Switching

- **Network congestion**: Congestion occurs **when too many** messages are assigned to a restricted resource (like a network link or buffer) such that **arriving messages have to be discarded since they cannot be stored anywhere**. Thus, in contrast to contention, congestion leads to an overflow situation with message loss.

- A **large variety of routing algorithms** have been proposed in the literature.

- Using the path length, **minimal and non-minimal** routing algorithms can be distinguished.

Routing and Switching

- Minimal routing algorithms **always select the shortest message transmission**.

- But this may **lead to congestion situations.**

- Non-minimal routing algorithms **do not always use paths** with minimum length if this is necessary to **avoid congestion at intermediate nodes**.

Routing and Switching

- A further classification can be made by distinguishing **deterministic routing algorithms** and **adaptive routing algorithms**.

- A **routing algorithm is deterministic** if the path selected for message transmission **only depends on the source and destination nodes** regardless of other transmissions in the network.

- Therefore, **deterministic routing** can lead to **unbalanced network load**. Path selection can be done source oriented at the sending node or distributed during message transmission at intermediate nodes.

Routing and Switching

- Adaptive routing tries to avoid such contentions by dynamically **selecting the routing path based on load information**.
- Between any pair of nodes, **multiple paths are available**.
- The **path to be used is dynamically** selected such that network traffic is spread evenly over the available links, thus **leading to an improvement of network utilization**.
- Moreover, **fault tolerance is provided**, since an **alternative path can be used in case of a link failure**.