Interconnection Networks II

Dynamic Networks

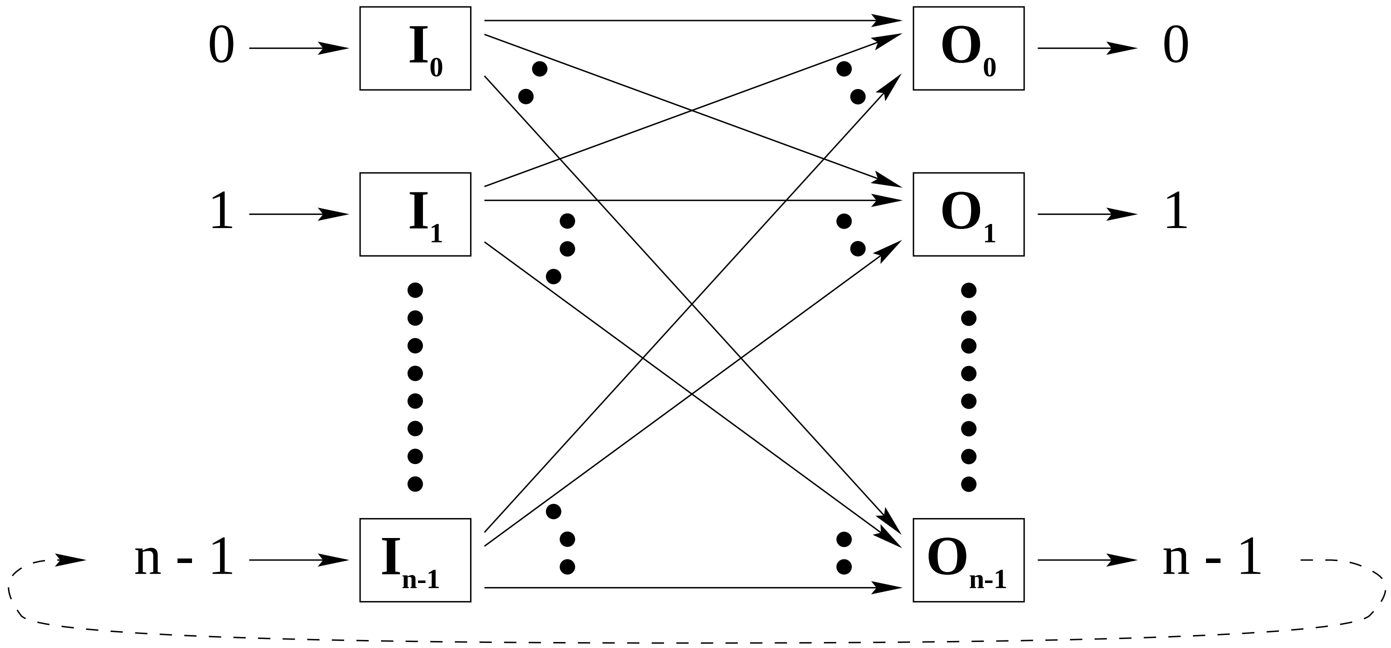
Tópico 7

Dynamic Networks

Dynamic networks can be classified according to the routing functions realised and whether they are *single-stage* (or *single stage with re-circulation*), or *multi-stage* networks.

Remember: “dynamic” means a virtual network, connections between inputs and outputs are set up (“programmed”) over a physical network.

Single-stage (recirculating) Networks

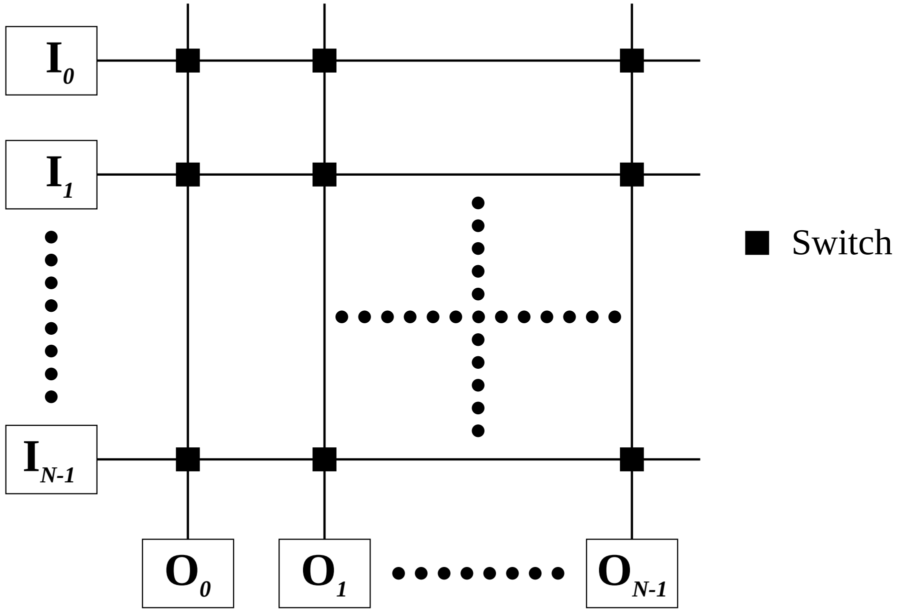


Single-stage (recirculating) Network$_{(cont)}$

- Each input may connect to one of a set of possible outputs (depending on the physical/hardware network connectivity). The network, on the previous slide, is an example of *full* connectivity (a crossbar switch).

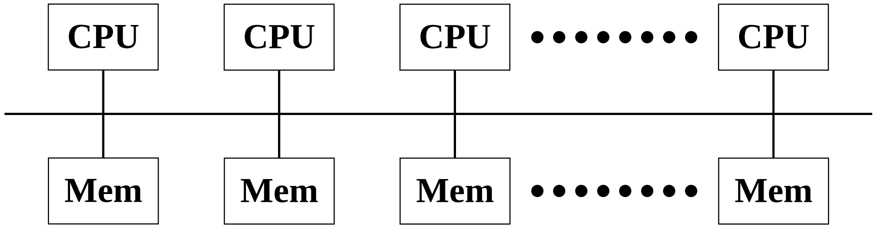
- If the network does not have full connectivity, several passes through the network may be needed to do an $I_x \rightarrow O_y$ routing, hence "re-circulating".
  - The greater the connectivity (wires) $\rightarrow$ the less re-circulating required.

X-bar (Crossbar Switch)

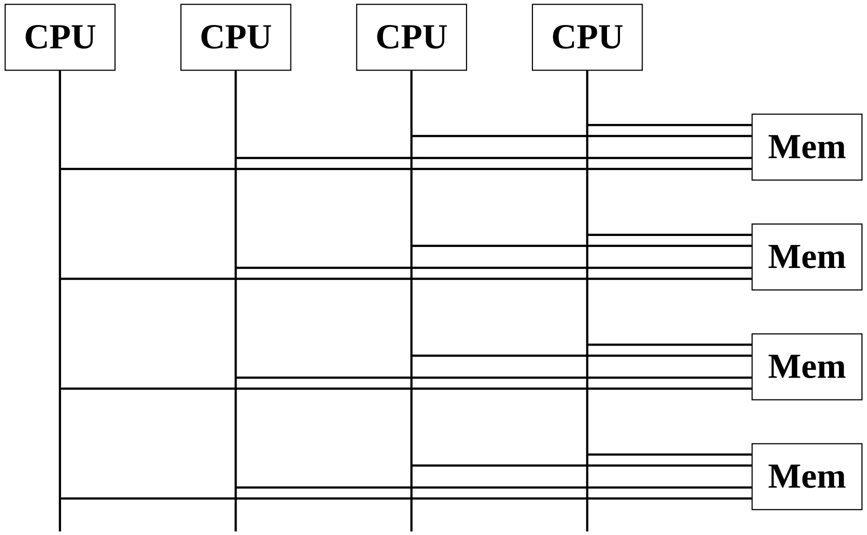


- Cost $\mathcal{O}(N^2)$ – A rough estimate on the amount of hardware required for N inputs and N outputs.
- Expensive – in terms of switches and wires.
- "One recirculation" – high bandwidth.

Implementations of the Basic Idea

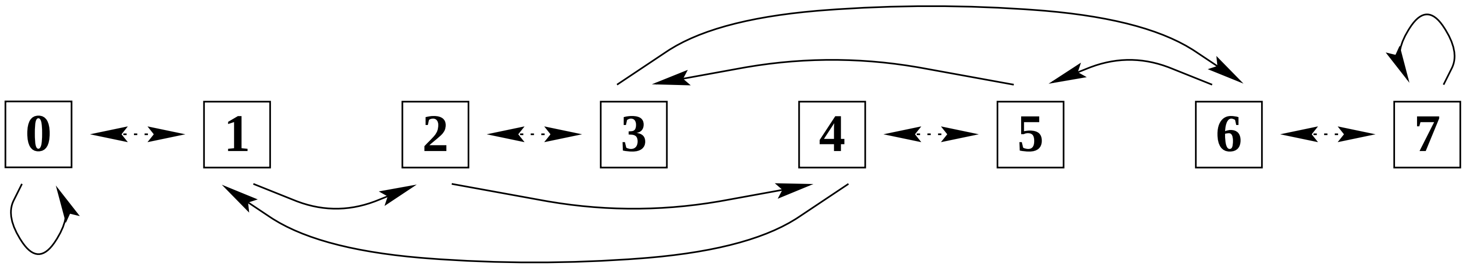


Sequential Bus



Multiported Memory

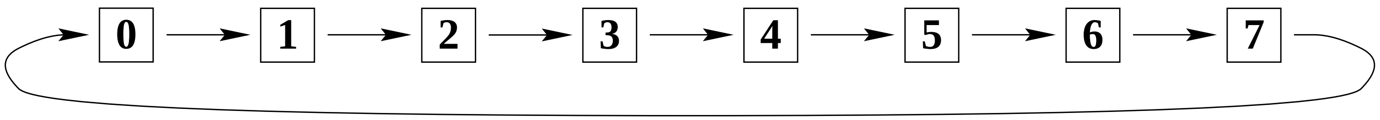
The Shuffle-Exchange Network



Note that this network can be specified in terms of the permutation routing functions it implements (which were presented earlier).

e.g. $\sigma(\epsilon_1(x))$ [Exercise !]

A Simple Shift Network



Routing function $\alpha(x) = (x + 1)$ MOD n;

The main point being it is extendible to bi-directional and multi-dimensional recirculating networks.

Multi-Stage Networks

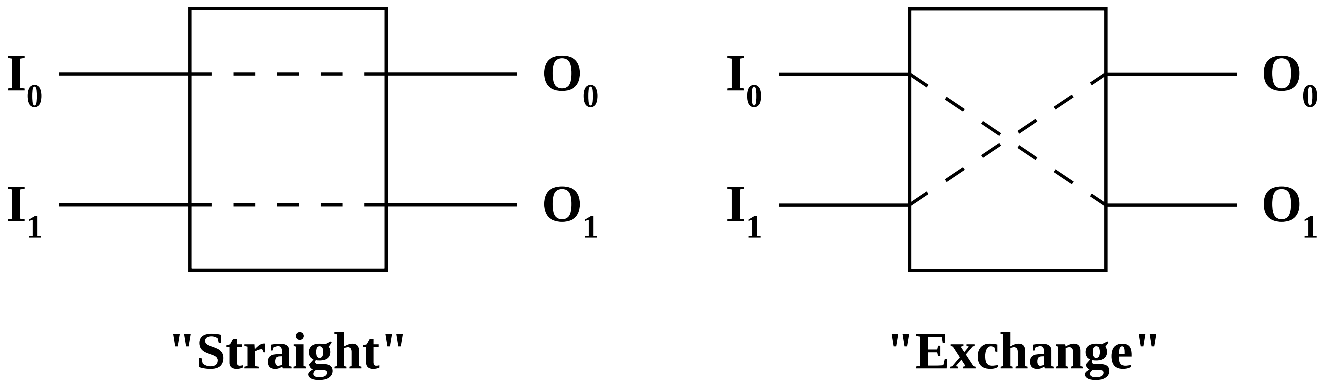
In general, a *multi-stage* network consists of n stages and N inputs and N outputs, where $N = 2^n$.

- $\frac{N}{2}$ "switch boxes" per stage, (in general). Two wires into and two out of each box.

- Each stage is connected to N inputs and N outputs.

Multi-Stage Networks$_{(cont)}$

- Each switch box/module is a crossbar switch that only allows one-to-one permutations:



- Network logic/hardware cost is thus $\mathcal{O}(N \cdot \log_2 N)$ compared with $\mathcal{O}(N^2)$ for a crossbar.

Control Strategies for Multi-Stage Networks

How are the switch boxes set for routing?

- *Centralised* (or *Statically*): Pre-compute the necessary switch settings for the routing and then send signals to the individual stages of the network.
  - SIMD
  - Circuit switched

Control Strategies for Multi-Stage Networks$_{(cont)}$

- *Distributed* (or *Dynamically*): Send the information through the network along with the message; when the message gets to a box, the associated information is used to determine the box setting.
  - MIMD
  - Packet Switched
- Note: Routing conflicts happen! Therefore need resolution strategies.

Various Multi-stage Networks

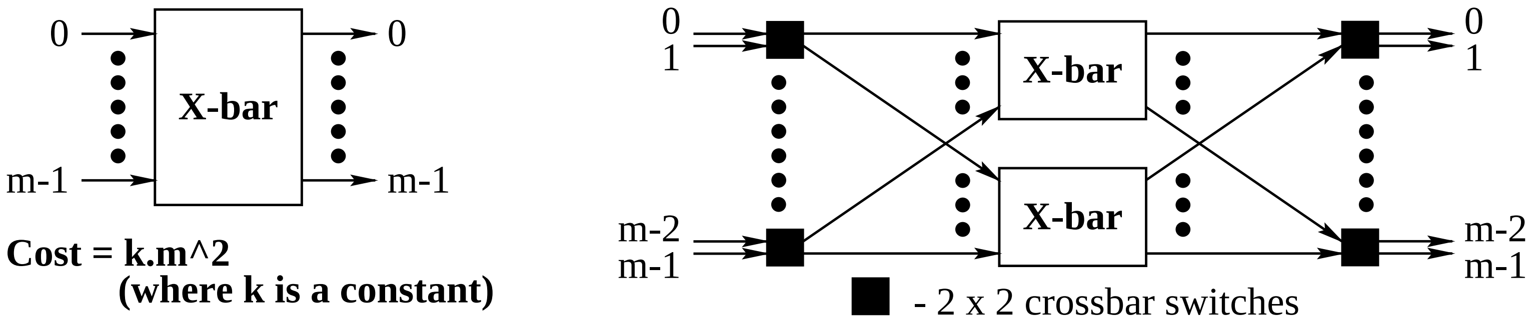
Different networks arise from:

- Different details of the switch boxes and their operational modes.
  - Can generally have $2^x \times 2^x$ switch boxes (**small** x of course, but x does not have to equal one).
  - Can generalise to other functions besides "straight" and "exchange".
- Different interconnections patterns between stages.

Switch box functionality is where "dynamic", "reconfigurable" networks come from, (since the interconnection structure between stages is fixed physically by wires).

The Benes Network

Consider a crossbar switch $(m \times m)$



Divide into two connected m/2 x m/2 crossbars

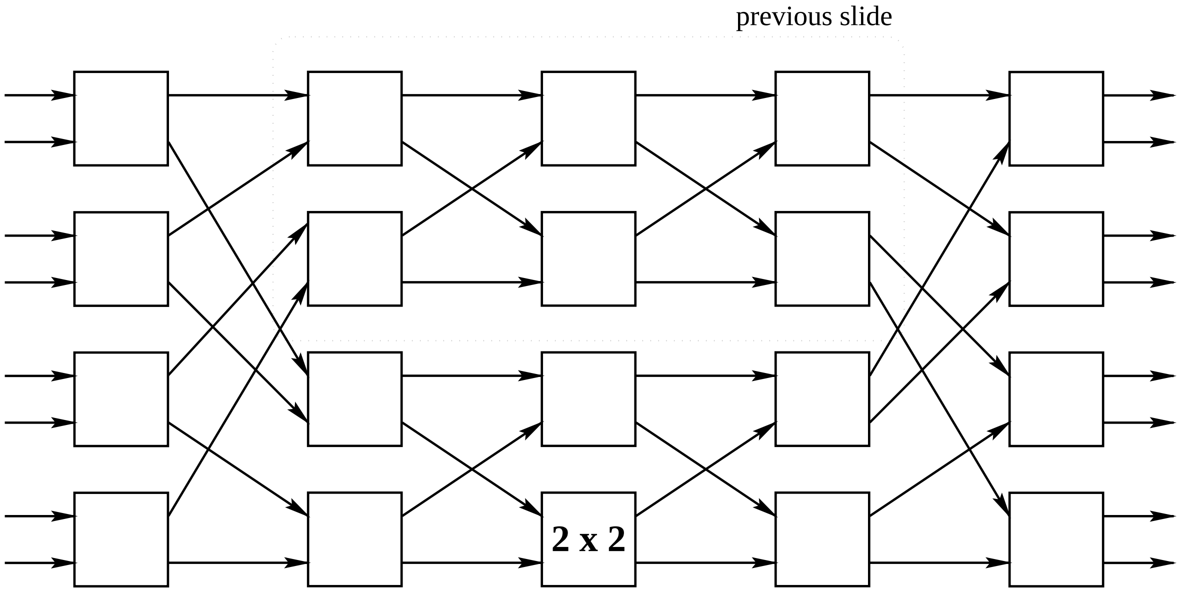
Cost:

$$\begin{aligned} C_2 &= 1 \\ C_m &= 2 \cdot C_{\frac{m}{2}} + O(m) \\ &= k_1 \cdot (\frac{m}{2})^2 + k_2 \cdot m \end{aligned}$$

Where k_1 and k_2 are constants for just **one** division.

The Benes Network$_{(cont)}$

Apply this transformation recursively until $m = 2$, i.e. until **all** of the boxes are 2×2 crossbar switches.



- Note that the inter-stage connections are (*inverse* and *perfect*) *shuffle* permutations!

The Benes Network$_{(cont)}$

- A Benes network has 2(log m - 1) + 1 stages
- Each stage has $\frac{m}{2}$ switch boxes
- Rough cost = $m \cdot$(log m - 1) + $\frac{m}{2}$
  = $\mathcal{O}(m \cdot \log m)$
- Note also the tradeoffs between **latency** and **cost**.

	Latency	Cost
X-bar	$\mathcal{O}(1)$	$\mathcal{O}(m^2)$
Benes	$\mathcal{O}(\log m)$	$\mathcal{O}(m \cdot \log m)$

Can we reduce the cost further?

- Note that Benes is a *non-blocking network*. Via rearrangement, all possible connects between pairs of inputs and outputs are realisable.

- A *blocking network* is one where the simultaneous connections of multiple input-output pairs results in conflicts in switches or wires. These networks generally consist of log m stages and thus may be considered cheaper (log $m < 2 \cdot \log m - 1$), although weaker (due to blocking).

Blocking Networks

Examples of blocking networks include Omega, Baseline, Multi-stage Cube networks etc; generally networks that have "shuffle type" connections between stages (as we will see).

- Only some subset of all possible input $\rightarrow$ output connections can be active simultaneously (due to the lack of connectivity, routing conflicts are possible).

- Blocking = Contention situations – two messages want to use the same line thus one must block the other.

Non-blocking Networks

Examples of this type of network include Benes and Clos. The network can perform all possible connections between input and output by rearranging its existing connections so that some new path from an input to an output may be established.

Non-blocking is rather specialised – it is still too costly; you have seen the Benes network. A whole general theory of multistage networks has grown around the blocking class.

Recall Permutation Notation

$$
\begin{aligned}
\sigma_k(a_{n-1}....a_0) &= a_{n-1}...a_k a_{k-2}...a_0 a_{k-1} \\
\sigma_k^{-1}(a_{n-1}....a_0) &= a_{n-1}...a_k a_0 a_{k-1}...a_1 \\
\epsilon_i(a_{n-1}....a_0) &= a_{n-1}...\overline{a_{i-1}}...a_0 \\
\beta_k(a_{n-1}....a_0) &= a_{n-1}...a_k a_0 a_{k-2}...a_1 a_{k-1}
\end{aligned}
$$

The Identity permutation $I(x) = x$ or
$I(a_{n-1}....a_0) = a_{n-1}....a_0$

- The ϵ (Exchange) and I (Identity) permutations are implemented by switch boxes.
- The σ (Shuffle) and β (Butterfly) permutations are implemented by wiring between stages.

The General Exchange Permutation

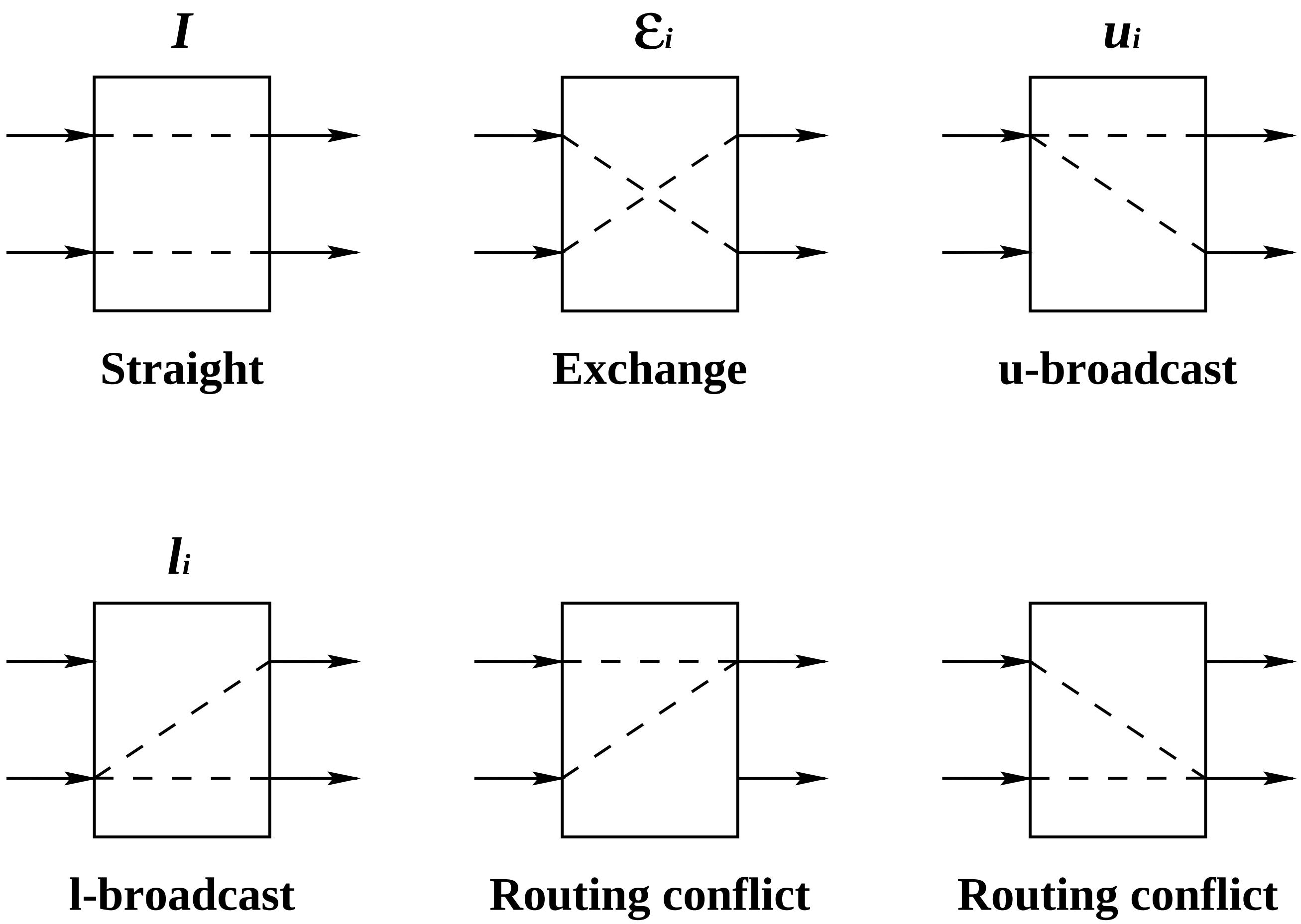
$$E_i(x) = x$$

is defined in terms of four simple permutations:

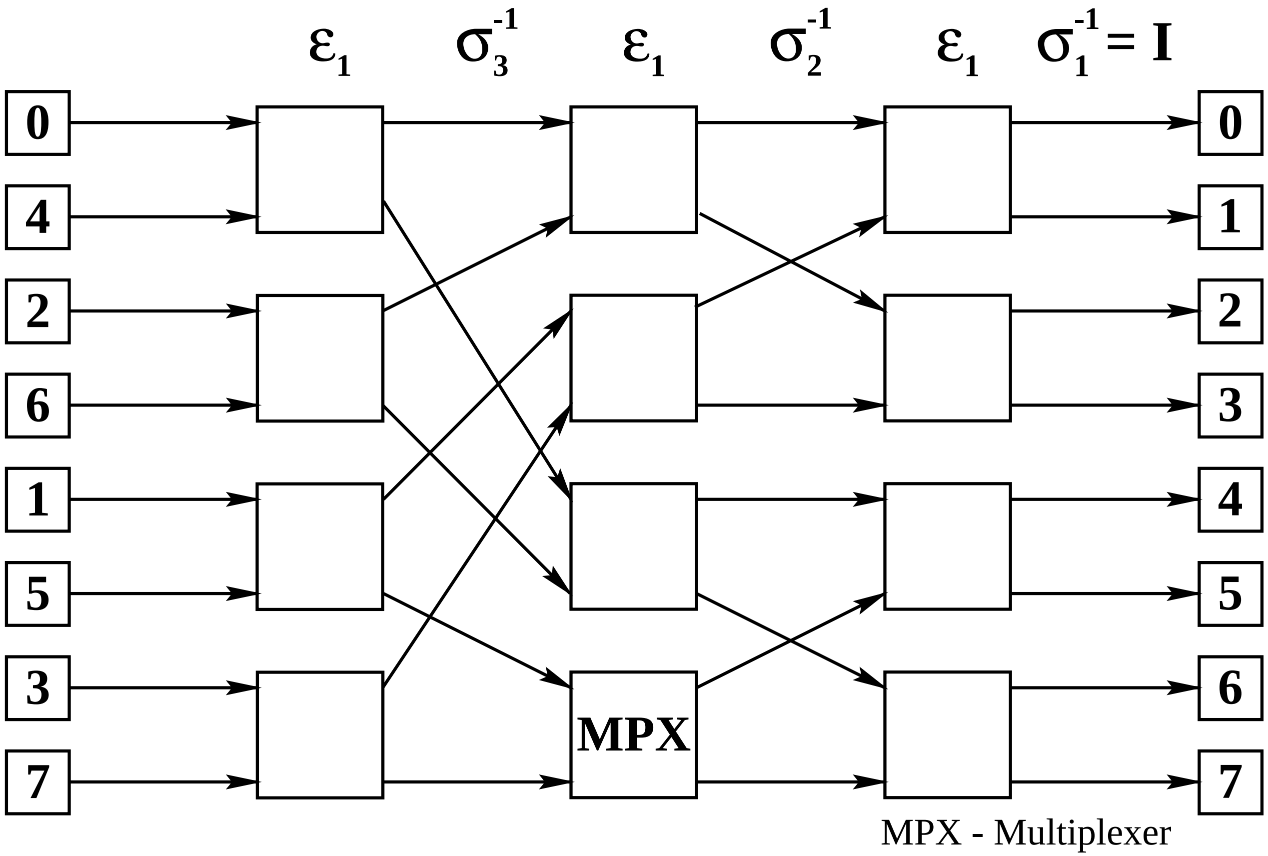
1. ϵ_i = "regular exchange";
2. I = Identity;
3. u_i = Upper broadcast;
4. l_i = Lower broadcast;

$$\begin{aligned} I(x) &= a_{n-1}....a_0 \\ u_i(x) &= a_{n-1}...a_i 1 a_{i-2}...a_0 \\ l_i(x) &= a_{n-1}...a_i 0 a_{i-2}...a_0 \\ E_i(x) &= \{\epsilon_i|I|u_i|u_i\}(x) \end{aligned}$$

General Exchange Switch Boxes



The Theory of Multi-stage Interconnection Networks (MSINs)



A Baseline Network

"Shuffle-Exchange"-type Networks

The *Baseline Network* is one of the "shuffle-exchange" type networks:

- All are blocking networks
- All have $\log_2 N$ stages, $(N = 2^n)$

The Baseline Network can be shown to be topologically equivalent to other networks e.g. Omega (see next slide) and Flip Networks.

"Shuffle-Exchange"-type Networks$_{(cont)}$

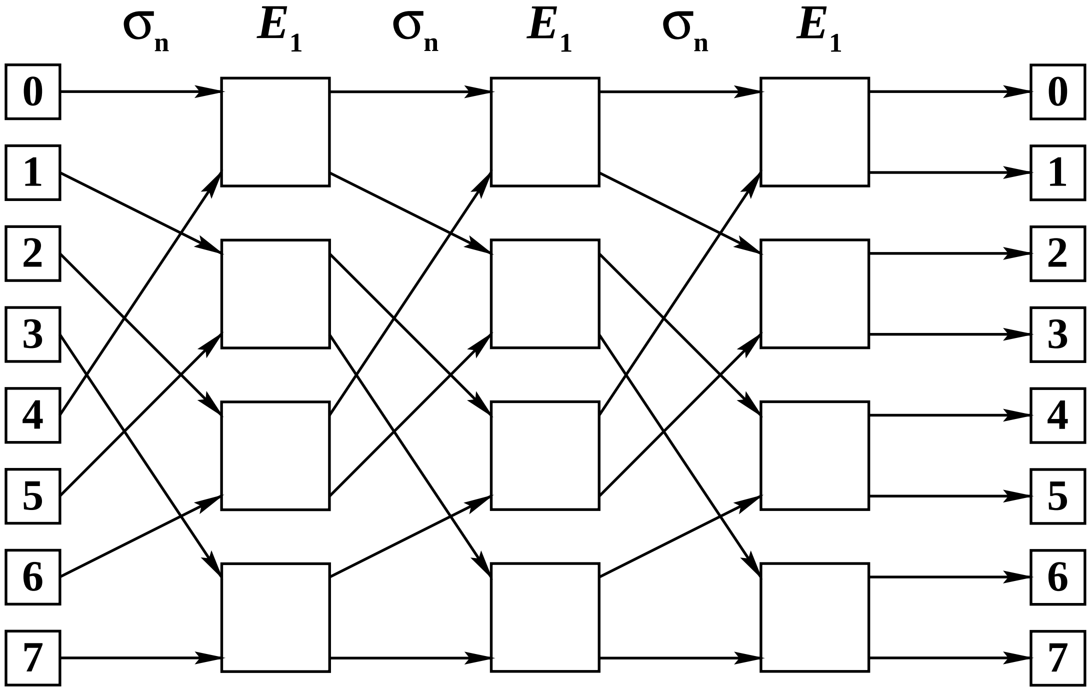
Other networks:

- The *Omega Network* - is "the" shuffle-exchange network since its wires connect stages in an explicit perfect shuffle pattern. Others are called shuffle-exchange types because they are very similar $\rightarrow$ "topologically equivalent".
- The *Banyan Network*; probably better known as the *Butterfly Network*.
- The *Indirect Binary n-cube*; otherwise known as the *Multi-stage Cube Network*.

Omega Network

- n ($= \log N$) identical stages, with $\frac{N}{2}$ switch boxes per stage.
- Connections from stage i to stage $i + 1$ ($0 \leq i \leq n$-1) are arranged in a perfect shuffle pattern.
- **General** exchange switch boxes.

Omega Network



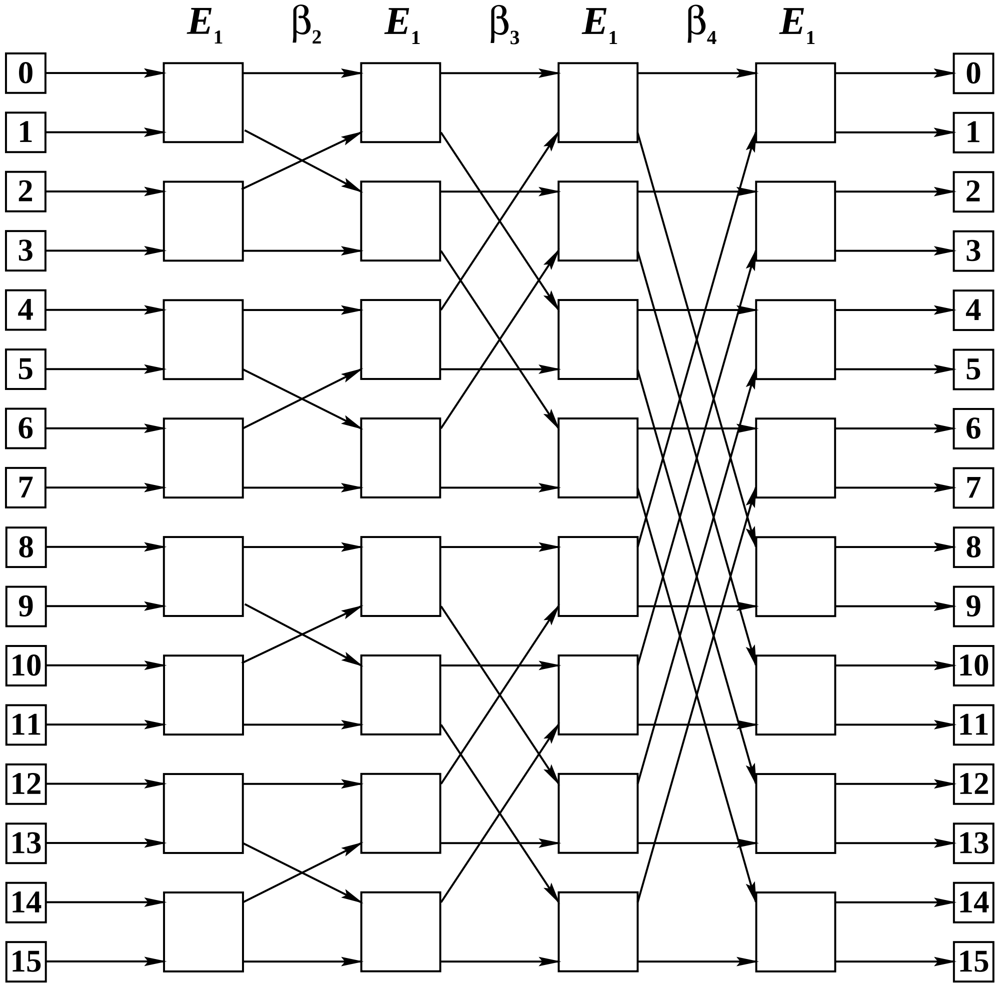
- Routing description in terms of permutations : $(\sigma_n E_1)^n$

Butterfly (Banyan) Network

Sometimes also known as a "N-way butterfly switch".

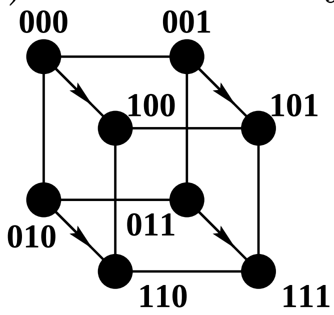
- Again, $n = \log N$ stages, with $\frac{N}{2}$ switch boxes per stage.
- Connections from stage i to stage $i+1$ $(0 \leq i \leq n\text{-}1)$ are $(i+2)^{th}$ sub-butterfly formations.
- **General** exchange switch boxes (in principle).

Butterfly Network$_{(cont)}$



Multi-stage Cube Network

Recall a static (direct) n-cube or hypercube, e.g. $n = 3$

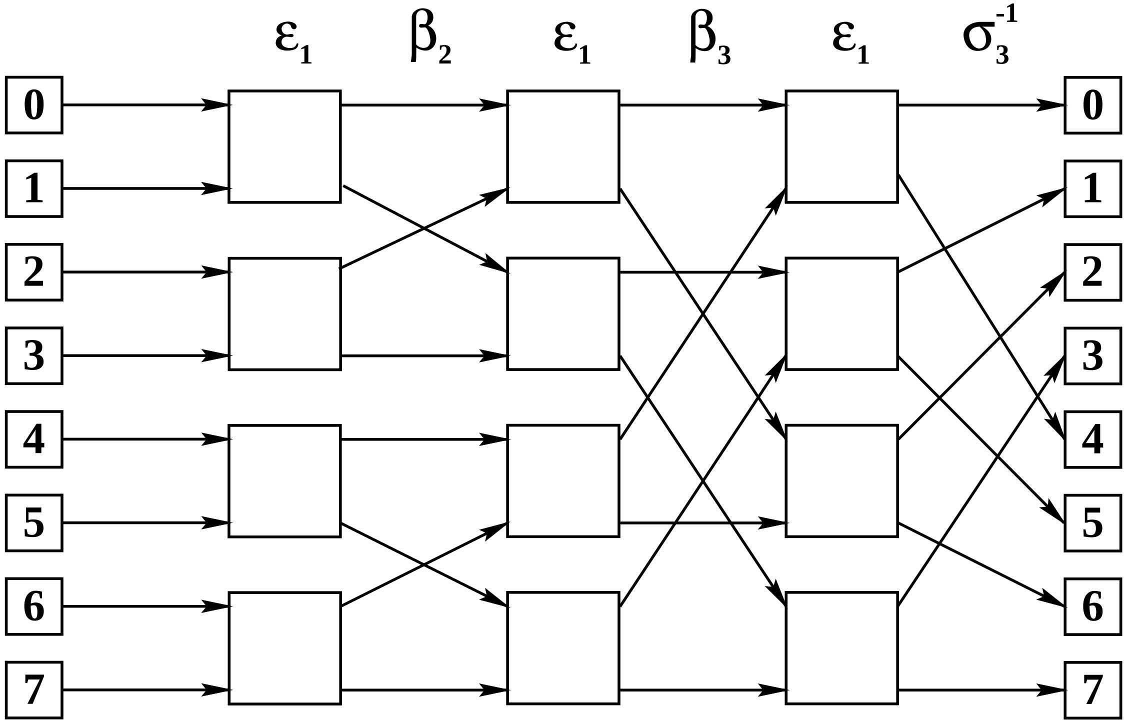


and routing functions :

$$f_i(x) = a_{n-1}...\overline{a_{i-1}}...a_0$$

The same set of routing functions, $0 \leq i \leq n$-1, can be realised by an n-stage network.

Multi-stage Cube Network$_{(cont)}$



The routing description can be described in terms of β, ϵ, and one additional permutation!

Multi-stage Cube Network$_{(cont)}$

Switch boxes are NOT general exchange switches, but have just the two functions:

- Straight or I, and
- Exchange ϵ_i.

The Cube routing functions complement (or not) the i^{th} bit.

Thus, stage i of the network realises the function f_i.

Simple dynamic routing

Let S = $(s_{n-1}...s_i...s_0)$ be the source address
D = $(d_{n-1}...d_i...d_0)$ be the destination address

For any stage i, $(0 \leq i \leq n\text{-}1)$, we can define the routing operation for an incoming message as the boolean function EXCH_i:

$$\mathrm{EXCH}_i = (s_i = d_i)$$

- Note that switch boxes may still have routing conflicts.

Cost/Performance of Networks

We just want a rough comparative understanding of networks.

- *Cross-bar*
  - Total link capability for any bijection.
  - Constant time point-to-point delay (latency).
  - Cost $\mathcal{O}(N^2)$

Cost/Performance of Networks$_{(cont)}$

- *Square Mesh of N nodes*
  - Limited connectivity.
  - 4 wires per node, maximum (edges may have 2 or 3).
  - On mesh with no wrap-around links, latency is $2(\sqrt{N-1})$. Slow ?
  - Low cost $\mathcal{O}(N)$: $\leq$ 4 wires per node.

Cost/Performance of Networks$_{(cont)}$

- *Hypercubes*
  - "Moderate" connectivity.
  - $\log_2 N$ wires per node.
  - Latency $\log_2 N$.
  - Cost $\mathcal{O}(N \cdot \mathrm{Log}_2 N)$
  - Limitation: when building large hypercubes, ones ends up needing some long wires for layout in 3D space (resulting in non-uniform node-to-node transmission times).

Cost/Performance of Networks$_{(cont)}$

- *Multi-stage Networks* (in general) - Benes, Omega, Butterfly, multi-stage cube (and others too).
  - Constant number of wires per node.
  - Latency $\log_2 N$ (except for Benes)
  - Cost $O(N \cdot \log_2 N)$

Cost/Performance of Networks$_{(cont)}$

- *k-ary n-cubes* (encompassing the hypercube and torus)
  - n is the dimension of the network;
  - k is the number of nodes along each dimension, connected as a ring.
  - k^n nodes with node degree $= 2{\cdot}n$ [Cost]
  - Network diameter $= n \cdot \lfloor \frac{k}{2} \rfloor$ [Latency]
  - Bisection bandwidth $= 2{\cdot}k^{n-1}$

Summary

- *Permutations* are useful in two respects since they can be used to:

  1. describe communication patterns within programs (efficiently?); and

  2. implement interconnection patterns within networks (cheaply?).

Summary $_{(cont)}$

- *Multi-stage Networks* are a compromise between expensive, high bandwidth, fully-connected Crossbar networks and cheap, low bandwidth buses.
- *Static Networks* are simpler and comparatively cheaper than MSINs but lack routing flexibility.