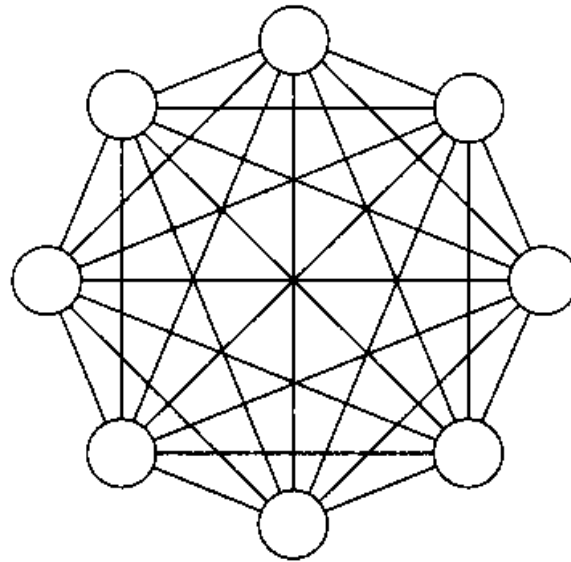
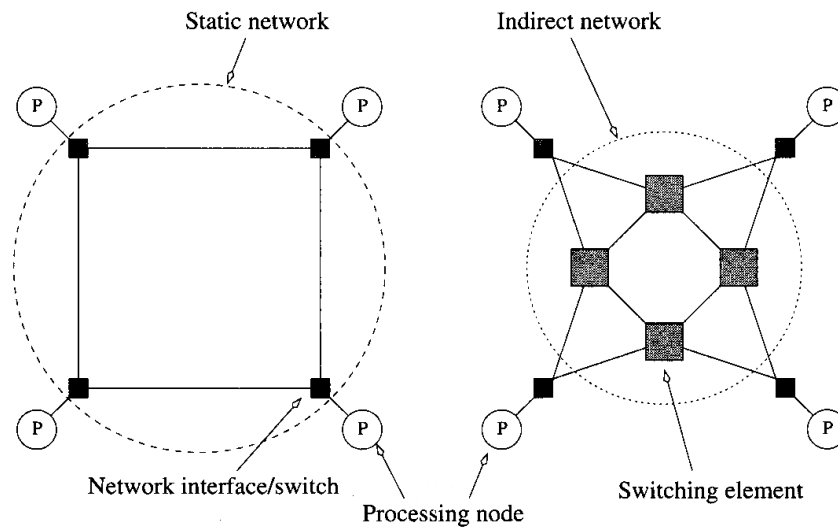


PE to PE interconnect: The most expensive supercomputer component  
Possible implementations:

**FULL INTERCONNECTION:**  
The ideal – Usually not attainable



Each PE has a direct link to every other PE. Nice in principle but costly:  
Number of links is proportional to the square of the number of PEs. For large number of PEs this becomes impractical. Therefore we will try two compromises:

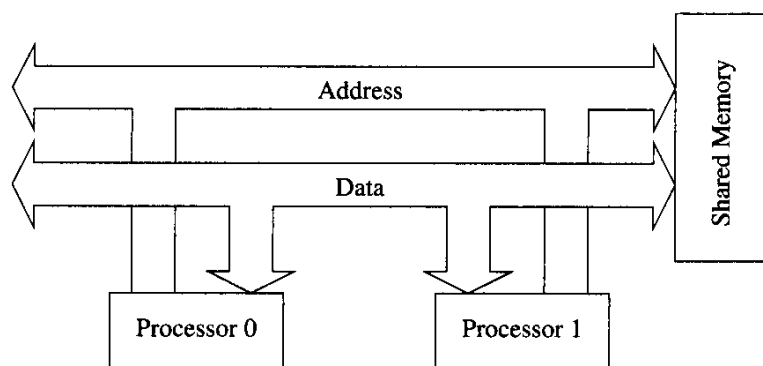


Static interconnect networks and dynamic interconnect networks.

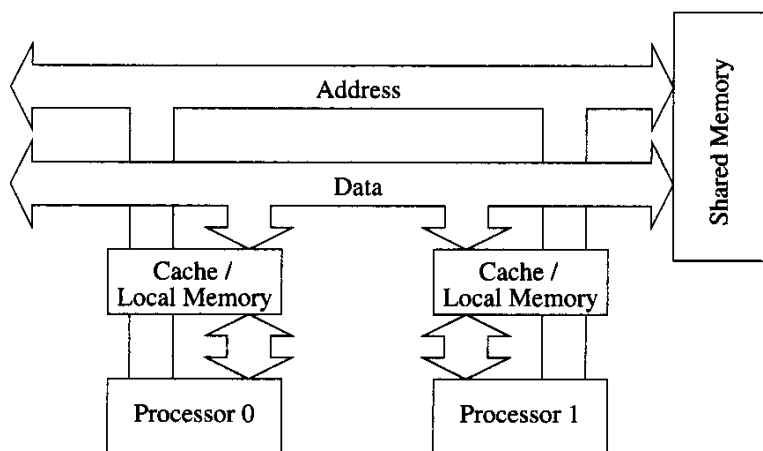
## BUS

Bus-based networks are perhaps the simplest: they consist of a shared medium common to all nodes. Cost of the network is proportional to the number of nodes, distance between any two nodes is constant  $O(1)$ . Ideal for broadcasting info among nodes.

However: Bounded bus bandwidth limits total number of nodes. Partial remedy: use caches – new problem: cache contamination.



(a)

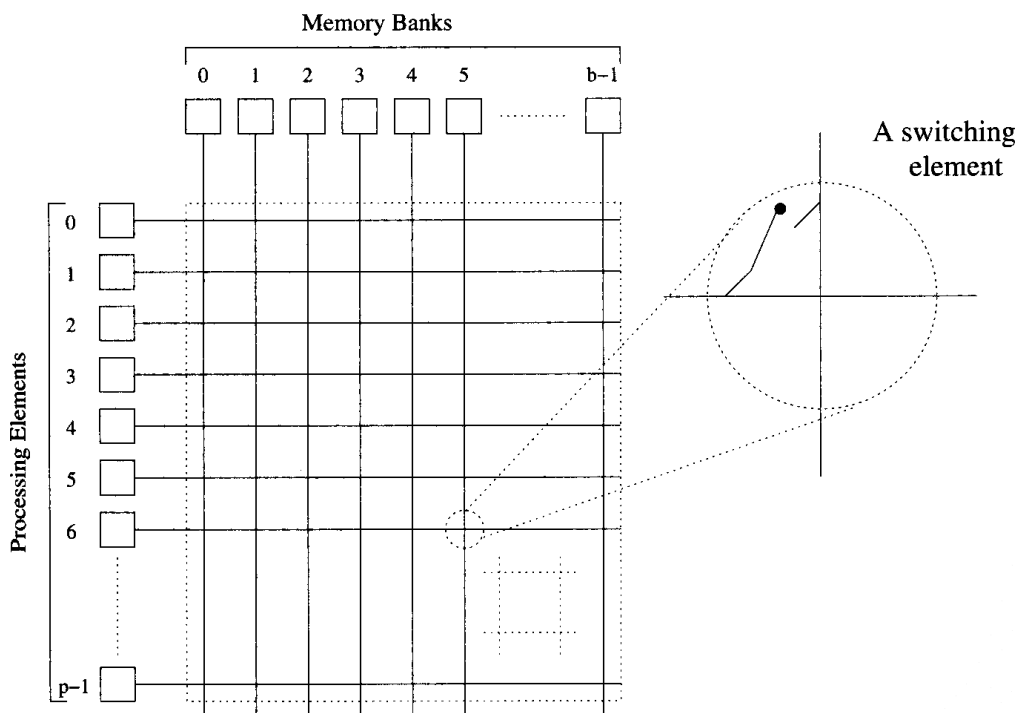


(b)

Bus networks: (a) Without local memory / caches, (b) with local memory / caches

## CROSSBAR

A crossbar network connecting  $p$  processors to  $b$  memory banks is shown below:

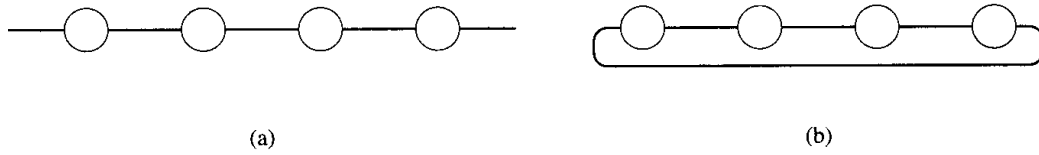


This is a non-blocking network: a connection of one processor to a given memory bank does not block a connection of another processor to a different memory bank.

There must be  $p \times b$  switches. It is reasonable to assume that  $b > p$ . From this follows that the cost of crossbar is high, at least  $O(p^2)$ , so it is not very scalable – like the fully connected network.

**LINEAR + RING**

In an attempt to reduce interconnect cost, we try out sparser networks:

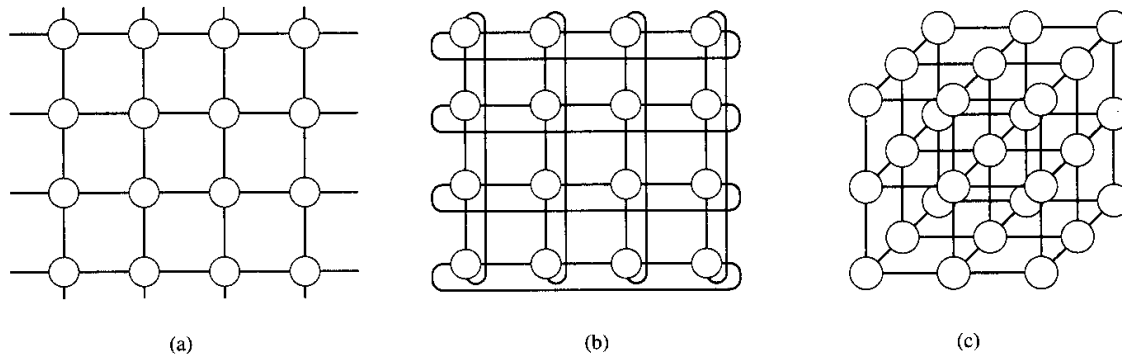


- (a) Linear network: every node has two neighbours (except terminal nodes)
- (b) Ring or 1D torus: every node has exactly two neighbours.

Note that by providing the wraparound link we halve the maximum distance between the nodes and double the bandwidth.

We may attempt a multidimensional generalization:

**MESH + TORUS: 2D, 3D, etc.**



- (a) 2D mesh, (b) 2D torus, (c) 3D mesh.

Designers like 2D meshes due to easy wiring layout.

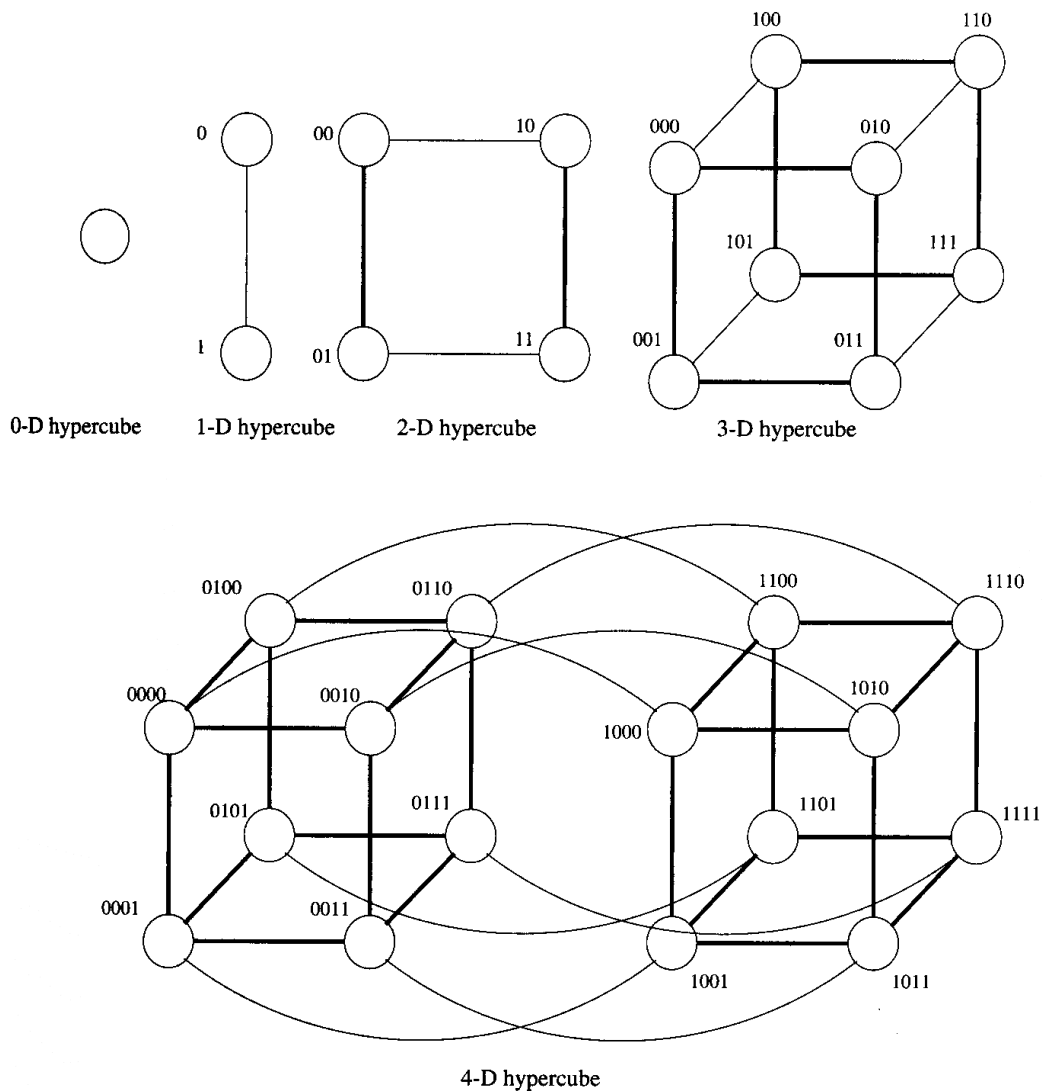
Users like 3D meshes and 3D tori because many problems map naturally to 3D topologies (like weather modeling, structural modeling, etc.). This is so because we seem to inhabit a 3D universe.

Note that nD meshes and nD tori need not have the same number of nodes in each dimension. This facilitates upgrades at cost of increased node-to-node distance.

Another multidimensional generalization:

So far, when increasing the number of processors we kept the network dimensionality constant. How about another approach: Let's keep the number of processors in any given dimension constant (say, 2) and keep increasing dimensionality. We get hypercube.

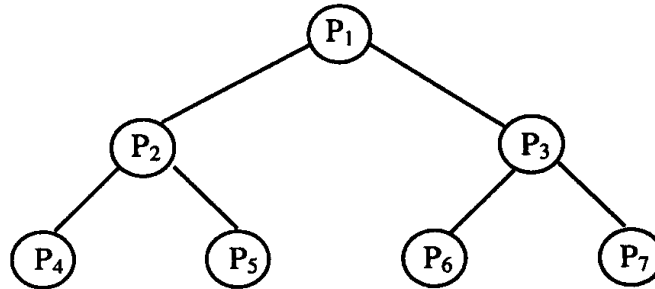
**HYPERCUBE (a.k.a. n-CUBE)**



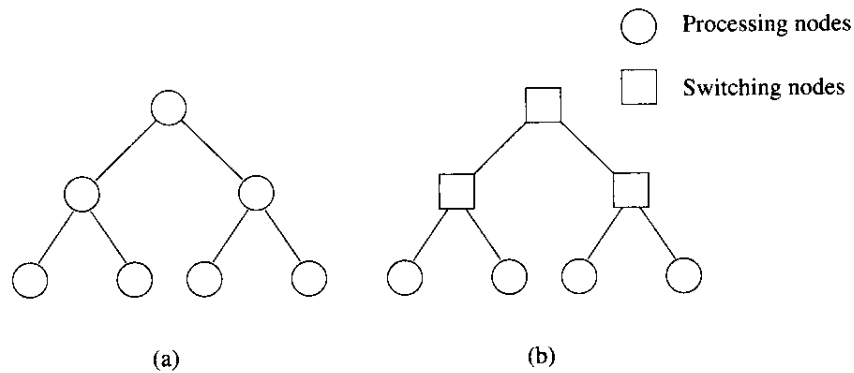
Observe a clever numbering scheme of nodes in a hypercube, facilitating message forwarding.

### TREE

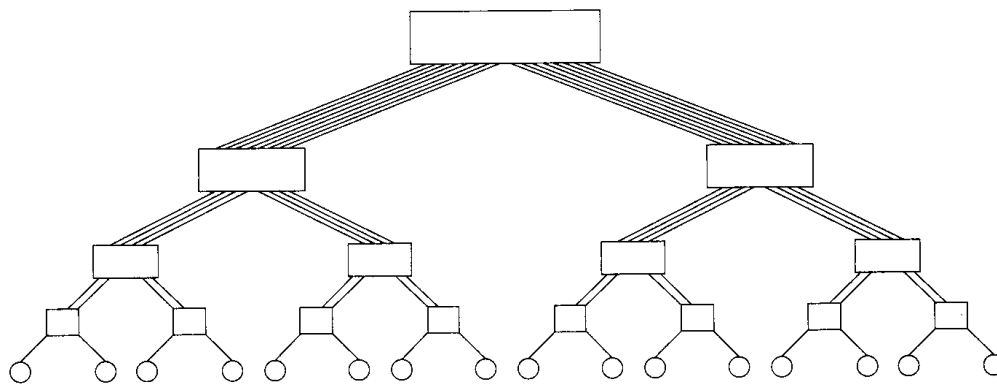
Basic concept:



In a tree network there is only one path between any two nodes. The taller the tree, the higher is communication bottleneck at high levels of the tree. Two remedies are possible:



We may have (a) static tree networks, or (b) dynamic tree networks. Alternatively, we may introduce fat tree networks (see below).

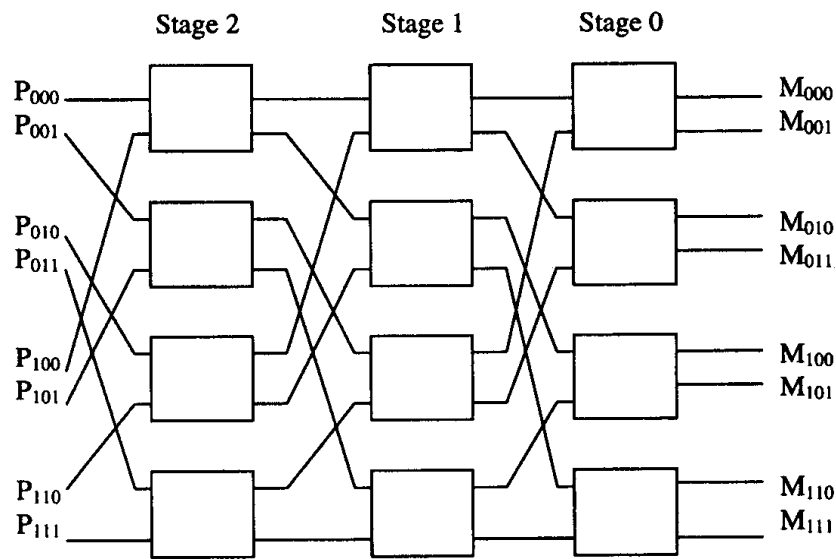


Fat tree network.

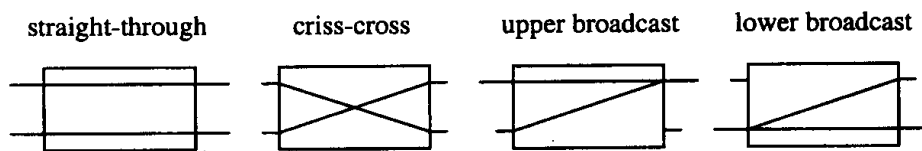
The crossbar network is scalable in terms of performance, but not scalable in terms of cost. Conversely, the bus network is scalable in terms of cost but not in terms of performance, hence some designers feel the need to compromise:

### MULTISTAGE NETWORKS

A multistage network connects a number of processors to a number of memory banks, via a number of switches organized in layers, viz:



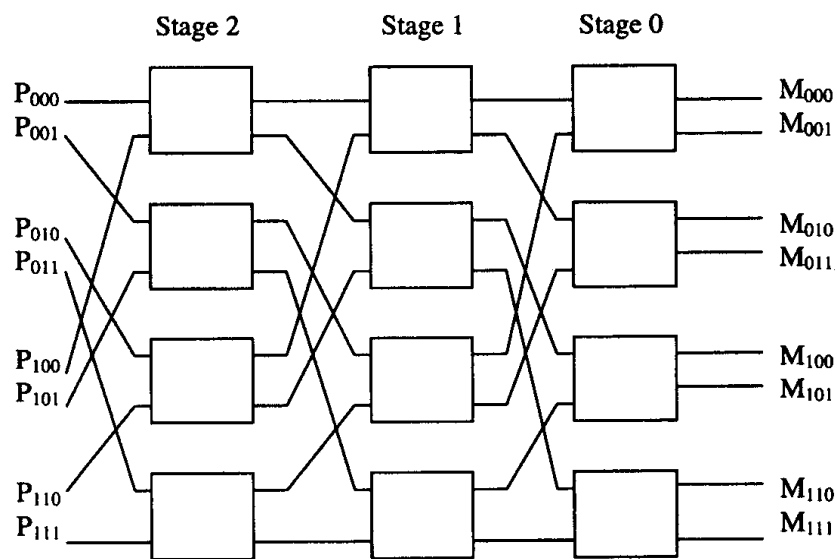
Each switch can be in one of the following positions:



The example above is that of the Omega Network.

**OMEGA**

Omega network connecting  $P$  processors to  $P$  memory banks (see below for  $P=8$ )

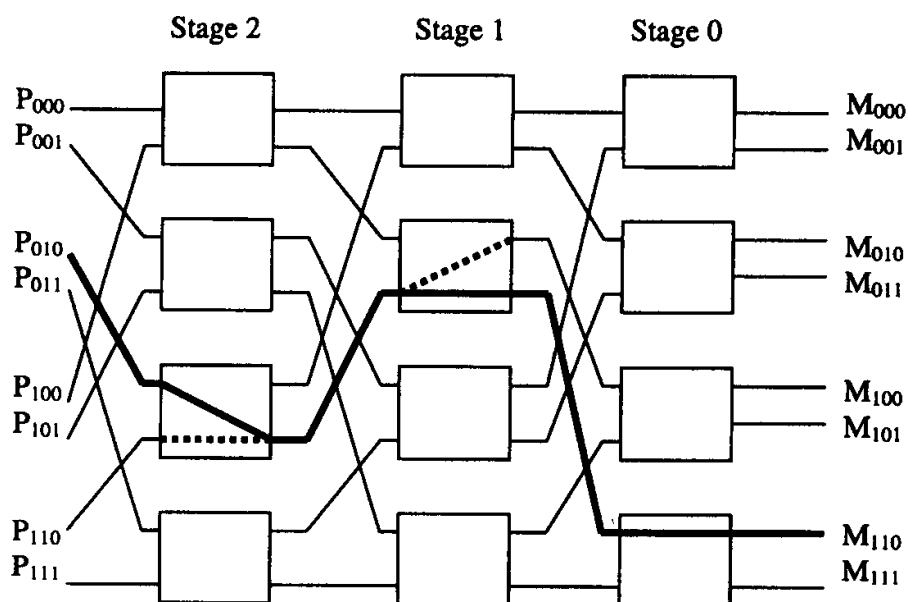


Omega network has  $P/2 * \log(P)$  switches, so the cost of this network is lower than the crossbar network.



**OMEGA (continued)**

Omega network belongs to the class of blocking networks:



Observe that, in the diagram above, when P<sub>2</sub> is connected to M<sub>6</sub>, P<sub>6</sub> cannot talk to M<sub>4</sub>.

An Omega network can be static: switches may remain in fixed position (either straight-thru or criss-cross). An Omega network can also be used to connect processors to processors. Example of such a network:

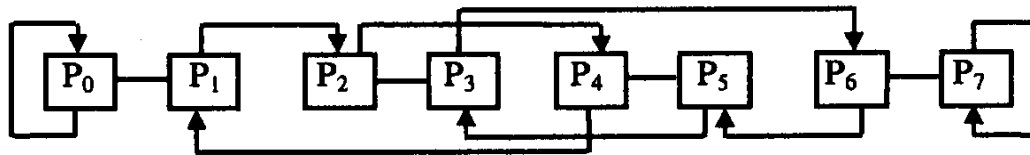
### SHUFFLE EXCHANGE

Consider a set of  $N$  processors, numbered  $P_0, P_1, \dots, P_{N-1}$

Perfect shuffle connects processors  $P_i$  and  $P_j$  by a one-way communications link, if

$$j = 2*i \text{ for } 0 \leq i \leq N/2 - 1 \quad \text{or} \quad j = 2*i + 1 - N \text{ otherwise.}$$

See below an example for  $N=8$  where arrows represent shuffle links and solid lines represent exchange links.

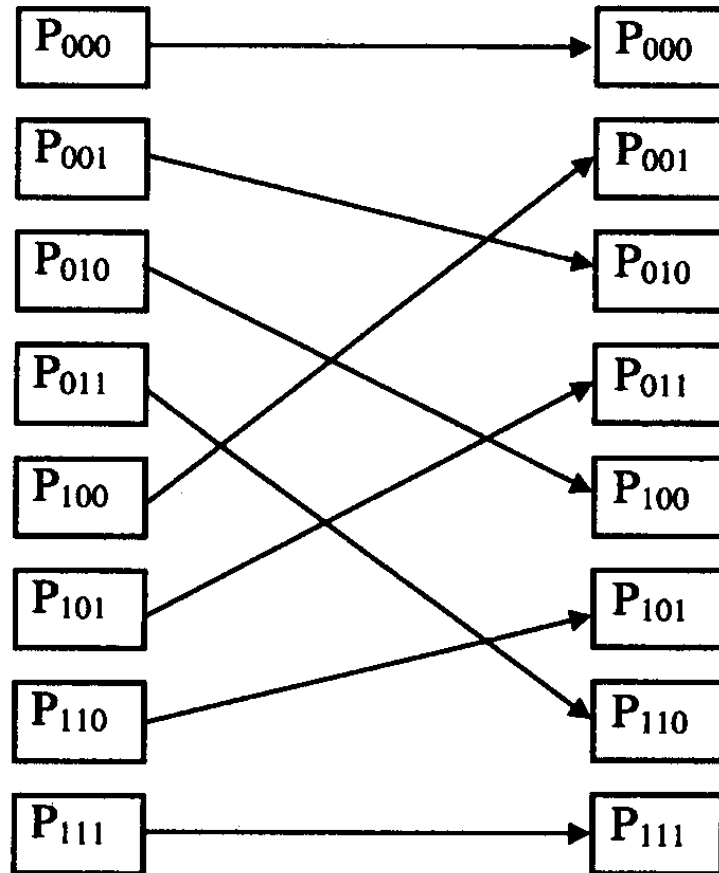


In other words, perfect shuffle connects processor  $i$  with  $(2*i \text{ modulo } (N-1))$ , with the exception of the processor  $N - 1$  which is connected to itself.

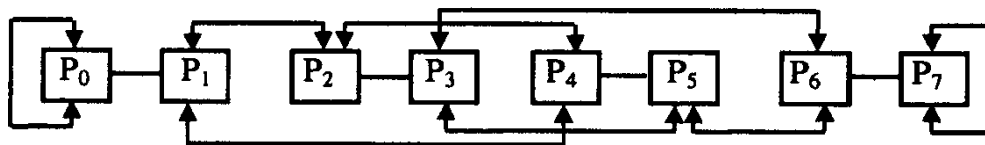
Having trouble with this logic? Consider the following:

**SHUFFLE EXCHANGE (continued)**

Let's represent numbers  $i$  and  $j$  in binary. If  $j$  can be obtained from  $i$  by a circular shift to the left, then  $P_i$  and  $P_j$  are connected by one-way communications link, viz.:



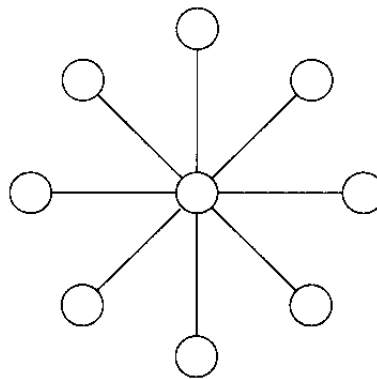
A perfect unshuffle can be obtained by reversing the direction of arrows or making all links bi-directional.



Other interconnect solutions:

### STAR

A naïve solution:



In this solution the central node plays the same role as the bus in bus networks. It also suffers from the some shortcomings.

However, this idea can be generalized:

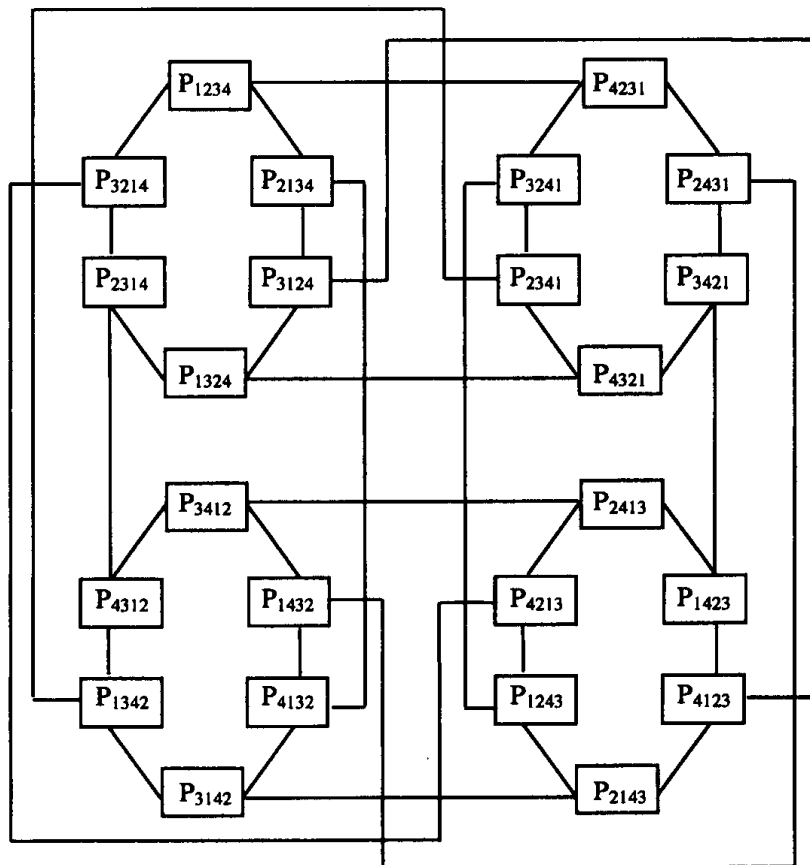
**STAR** (continued)

A generalized star interconnection network has the property that for a given integer  $N$ , we have exactly  $N!$  processors.

Each processor is labeled with the permutation to which it corresponds.

Two processors  $P_i$  and  $P_j$  are connected if the label  $i$  can be transformed into label  $j$  by switching the first label symbol of  $i$  with a symbol of  $j$  (excluding 1<sup>st</sup> symbol of  $j$ )

Below we have a star network for  $N=4$ , i.e. a network of  $4! = 24$  processors.  
 Example: Processors labeled 2134 and 3124 are connected with two links.



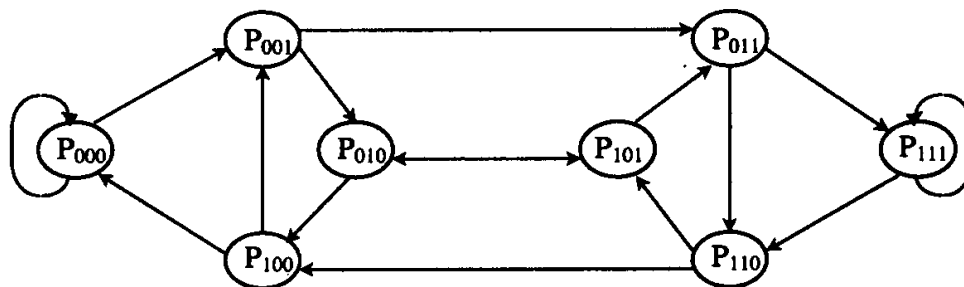
NOTE: The whole idea is to make each node a center node of a small star!

## DE BRUIJN

A network consisting of  $N = d^k$  processors, each labeled with a  $k$ -digit word  $(a_{k-1} a_{k-2} \dots a_1 a_0)$  where  $a_j$  is a digit (radix  $d$ ), i.e.  $a_j$  is one of  $(0, 1, \dots, d-1)$

The processors directly reachable from  $(a_{k-1} a_{k-2} \dots a_1 a_0)$  are  $(a_{k-2} \dots a_1 a_0 q)$  and  $(q a_{k-1} a_{k-2} \dots a_1)$  where  $q$  is another digit (radix  $d$ ).

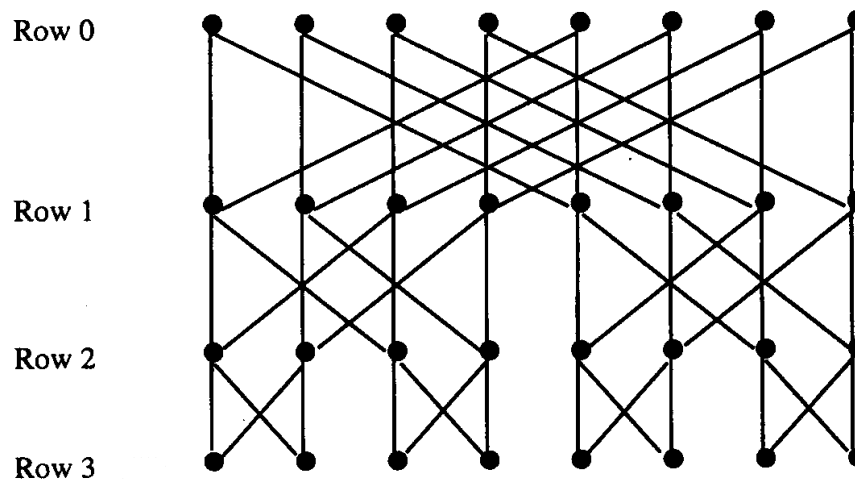
Shown below is a de Bruijn network for  $d=2$  and  $k=3$



De Bruijn network can be seen as a generalization of a shuffle exchange network. It contains shuffle connections, but has smaller diameter than the shuffle exchange (roughly half the diameter).

## BUTTERFLY

A Butterfly network is made of  $(n + 1) \cdot 2^n$  processors organized into  $n+1$  rows, each containing  $2^n$  processors.



Rows are labeled 0 ... n. Each processor has four connections to other processors (except processors in top and bottom row).

Processor  $P(r, j)$ , i.e. processor number  $j$  in row  $r$  is connected to

$$P(r-1, j) \text{ and } P(r-1, m)$$

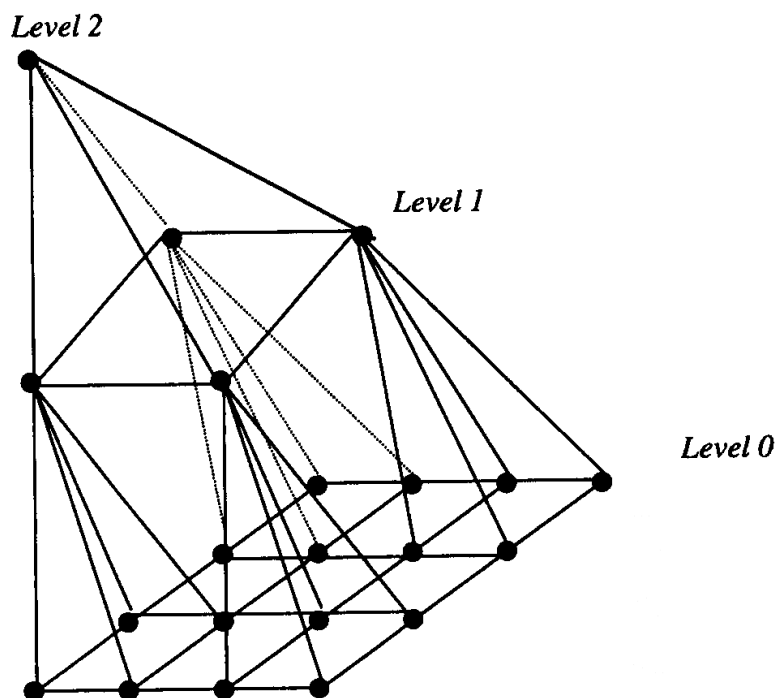
where  $m$  is obtained by inverting the  $r^{\text{th}}$  significant bit in the binary representation of  $j$ .

## PYRAMID

A pyramid consists of  $(4^{d+1} - 1)/3$  processors organized in  $d+1$  levels so as:

- Levels are numbered from  $d$  down to 0
- There is 1 processor at level  $d$
- Every level below  $d$  has four times the number of processors than the level immediately above it.

Note the connections between processors.



Pyramid interconnection can be seen as generalization of the ring – binary tree network, or as a way of combining meshes and trees.



## COMPARISON OF INTERCONNECTION NETWORKS

Intuitively, one network topology is more desirable than another if it is

- More efficient
- More convenient
- More regular (i.e. easy to implement)
- More expandable (i.e. highly modular)
- Unlikely to experience bottlenecks

Clearly no one interconnection network maximizes all these criteria. Some tradeoffs are needed.

Standard criteria used by industry:

- **Network diameter** = Max. number of hops necessary to link up two most distant processors
- **Network bisection width** = Minimum number of links to be severed for a network to be into two halves (give or take one processor)
- **Network bisection bandwidth** = Minimum sum of bandwidths of chosen links to be severed for a network to be into two halves (give or take one processor)
- **Maximum-Degree of PEs** = maximum number of links to/from one PE
- **Minimum-Degree of PEs** = minimum number of links to/from one PE

**COMPARISON OF INTERCONNECTION NETWORKS** (continued)

Interconnect comparison at-a-glance:

<b>Network Topology</b>	<b>Number of Nodes</b>	<b>Node Degree</b>
Linear and Ring	$d$	2
Shuffle-Exchange	$2^d$	3
2D Mesh	$d^2$	4
Hypercube	$2^d$	$d$
Star	$m!$	$m-1$
De Bruijn	$2^d$	4
Binary Tree	$2^d - 1$	3
Butterfly	$(d+1) * 2^d$	$d+1$
Omega	$2^d$	2
Pyramid	$(4^{d+1} - 1)/3$	9