Recursive Decomposition of The Clos Network

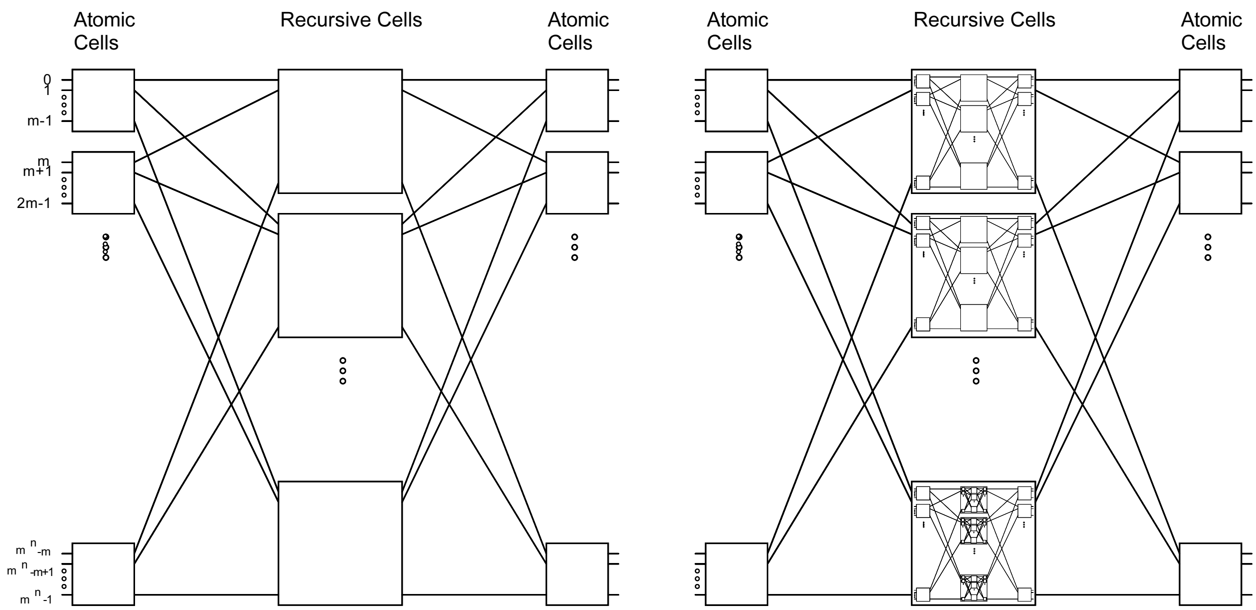
Result is a much-lower-cost rearrangeable network.

In contrast, the network obtained when recursive decomposition was applied to the Omega network had the same properties.

For scalable Clos networks with $n > 1$:

Center-stage cells are recursive.

Other cells are atomic.



Two Methods To Compute Cost

The powerful way: write recurrence equations:

Let $C(n,m)$ be cost of network of size n using $m \times m$ cells.

$$\begin{aligned} C(1,m) &= m^2 \text{ xp} \\ C(n,m) &= m^2 m^{n-1} \text{ xp} + mC(n-1,m) + m^2 m^{n-1} \text{ xp} \\ &= 2m^{n+1} \text{ xp} + mC(n-1,m) \end{aligned}$$

Equations like this can easily (more or less) be solved in closed form.

The easy? clever? way:

Observation: *every stage consists of* m^{n-1} *cells.*

Number of stages: $2n-1$.

$$\begin{aligned} C(n,m) &= m^2 m^{n-1}(2n-1) \text{ xp} \\ &= m^{n+1}(2n-1) \text{ xp} \end{aligned}$$

Substituting $N = m^n$ and $n = \log_m N$:

$$C(N,m) = mN(2\log_m N - 1) \text{ xp}$$

Cost is almost twice the cost of an omega network.

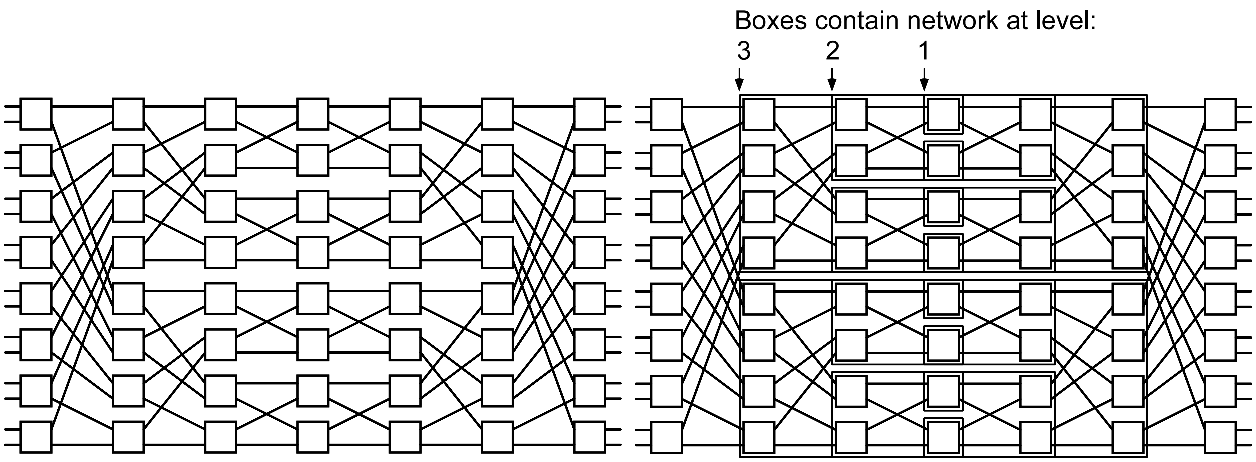
But, it is much less than non-recursive Clos network.

And it's still a permutation network.

The Beneš Network

Named after V. Beneš, described in a 1962 BSTJ paper.

It is a recursively decomposed Clos network.



Routing:

Use looping algorithm several times:

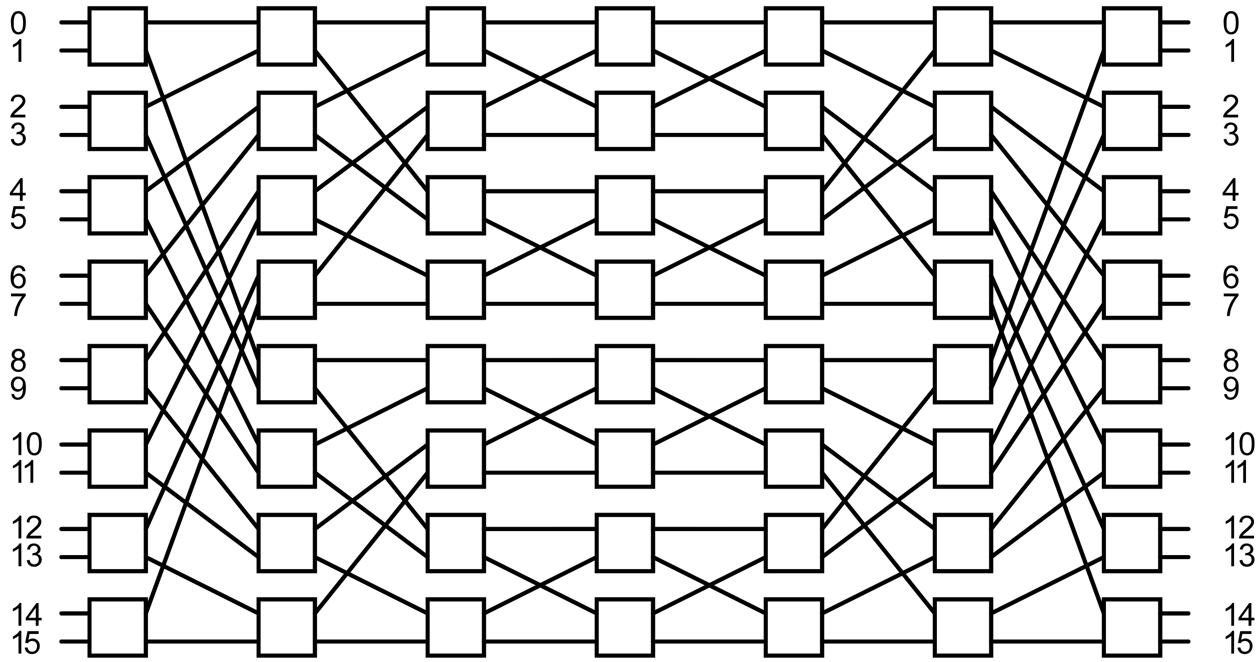
- We're finished if network consists of a single 2×2 crossbar.
- Otherwise, use looping algorithm, remembering that this is a recursive network.

  Result is settings for first and last stages, and

  permutations for two center-stage recursive cells.

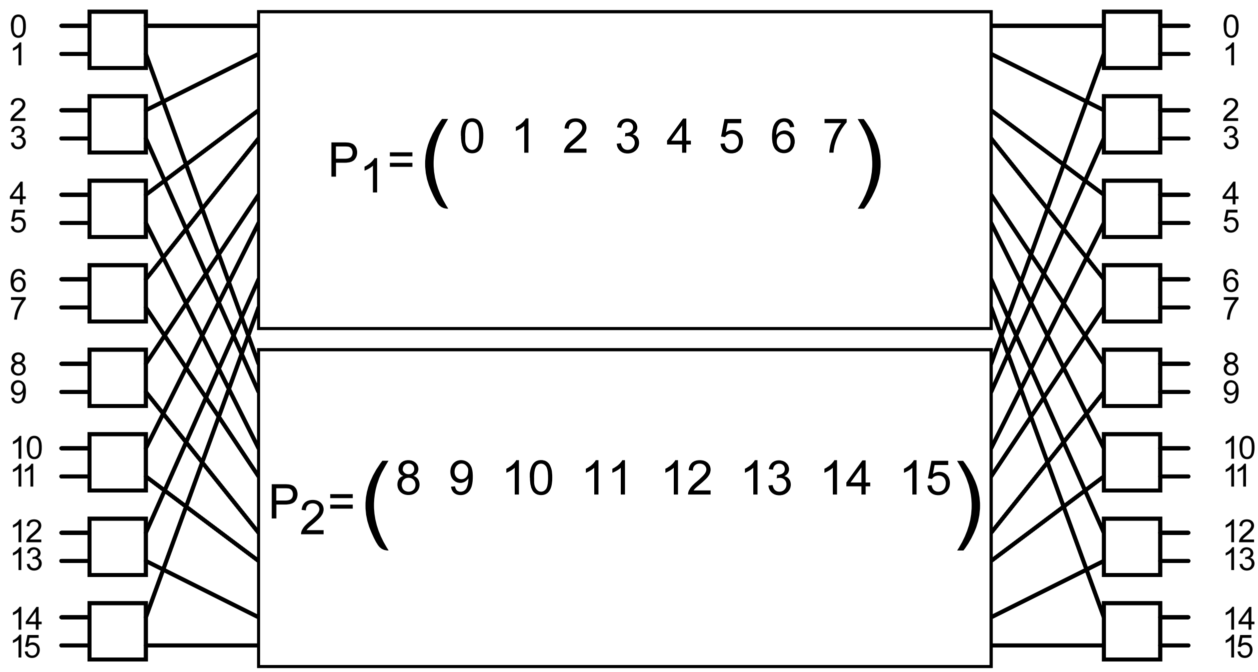
- Route each center-stage recursive cell using this procedure.

Routing Example



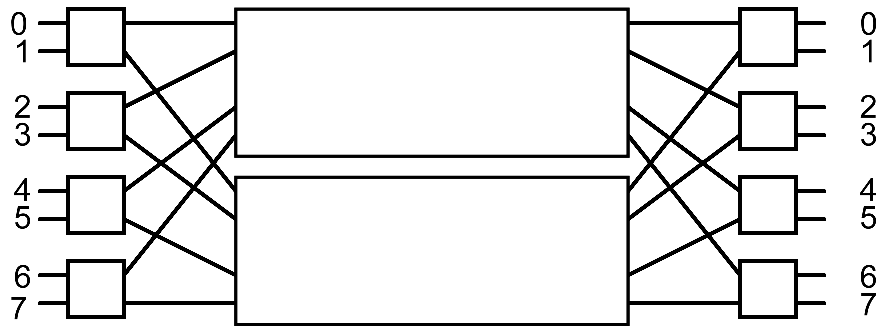
$$P = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 5 & 15 & 13 & 7 & 3 & 0 & 8 & 11 & 9 & 6 & 12 & 1 & 10 & 2 & 14 & 4 \end{pmatrix}$$

Routing Example, Continued

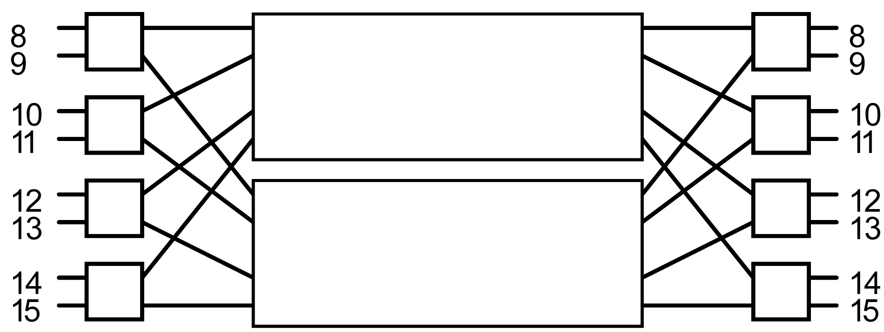


$$P = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 5 & 15 & 13 & 7 & 3 & 0 & 8 & 11 & 9 & 6 & 12 & 1 & 10 & 2 & 14 & 4 \end{pmatrix}$$

Routing Example, Continued.

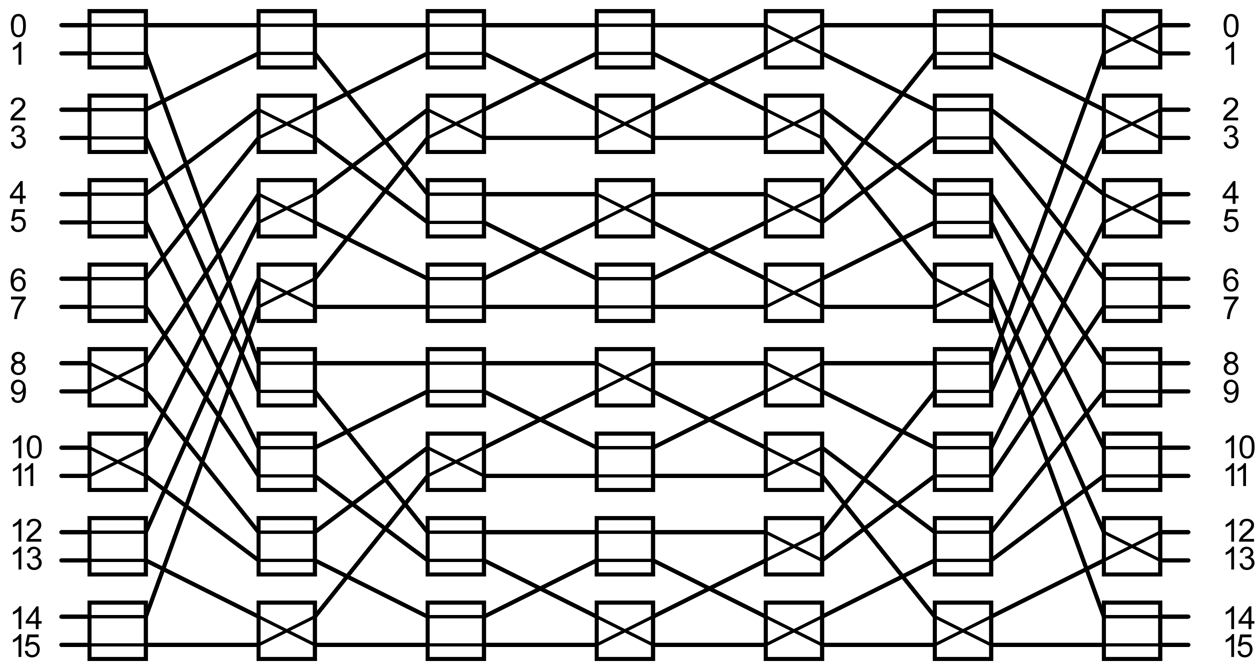


$$P_1 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 2 & 6 & 1 & 4 & 3 & 0 & 5 & 7 \end{pmatrix}$$



$$P_2 = \begin{pmatrix} 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 15 & 11 & 8 & 13 & 12 & 14 & 9 & 10 \end{pmatrix}$$

Routing Example, Finished.



$$P = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 5 & 15 & 13 & 7 & 3 & 0 & 8 & 11 & 9 & 6 & 12 & 1 & 10 & 2 & 14 & 4 \end{pmatrix}$$

Minimum Cost Permutation Networks

Lower Bound on Permutation Network Cost

Result due to Shannon, BSTJ 1950.

Idea: Compare amount of information to code a permutation to amount of information in Beneš network state.

Amount of information can be measured in number of bits.

Amount of information to code a permutation $I(\Sigma_N) = \log_2 N!$.

Using Sterling's approximation:

$$\begin{aligned}
I(\Sigma_N) &= \log_2 N! \\
&\approx \log_2 \left(\sqrt{2\pi N} \frac{N^N}{e^N} \right) \\
&= \frac{1}{2} \log_2 2\pi N + \log_2 N^N - \log_2 e^N \\
&= \frac{1}{2} \log_2 2\pi 2^n + \log_2 (2^n)^{(2^n)} - \log_2 e^{(2^n)} \\
&= \frac{1}{2} \log_2 \pi + \frac{n+1}{2} \log_2 2 + n2^n \log_2 2 - 2^n \log_2 e \\
&= \frac{1}{2} \log_2 \pi + \frac{n+1}{2} + n2^n - 2^n \log_2 e \\
&= n2^n - 2^n \log_2 e + \frac{n+1}{2} + \frac{1}{2} \log_2 \pi
\end{aligned}$$

The number of bits to code a Beneš network state.

(When routing permutations) each cell can be in two states:

Identity and transpose.

So state of each cell can be coded with one bit.

The number of bits then is equal to the number of cells:

$$I(\text{Beneš}) = 2^{n-1}(2n-1) = n2^n - 2^{n-1}$$

$$I(\Sigma_N) = n2^n - 2^n \log_2 e + \frac{n+1}{2} + \frac{1}{2}\log_2 \pi$$

The highest order terms of the two expressions are equal.

Therefore, the Beneš network is asymptotically optimal.

(Which is not quite as good as being optimal.)

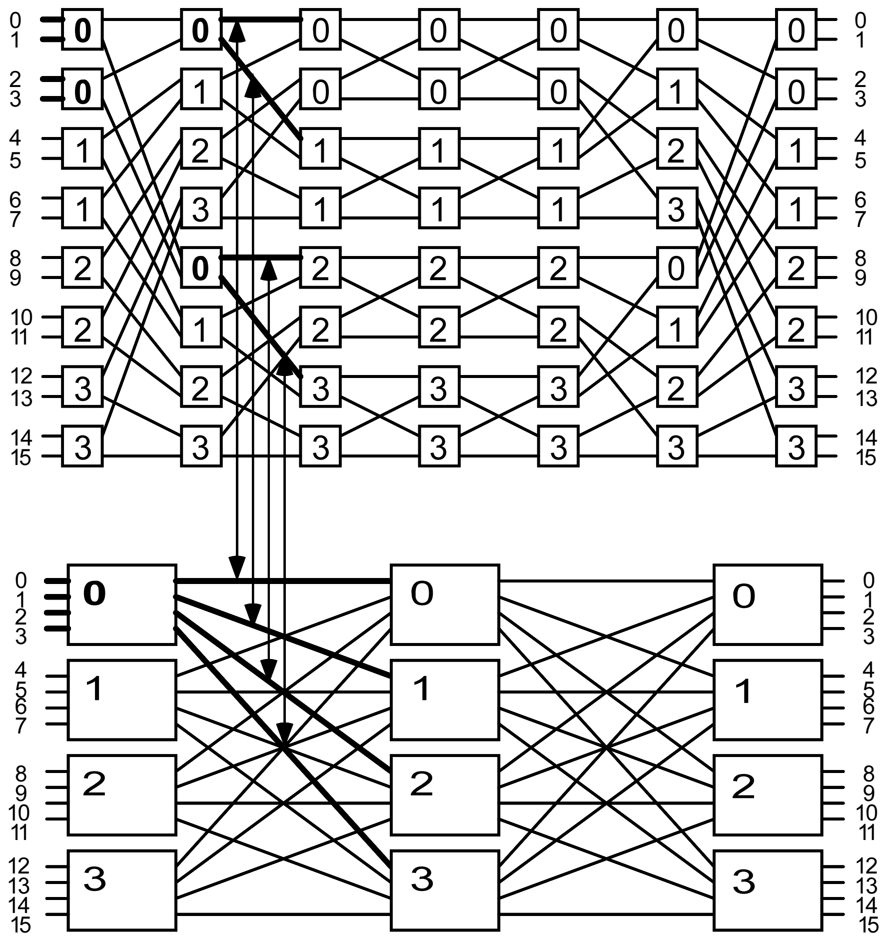
Correspondence Between Clos and Beneš Networks

Cells in Beneš network can be mapped to Clos in a many-to-one fashion.

This mapping reveals important properties.

Used as basis for routing algorithm.

Used to determine which connection assignments the input- and output-stage cells need to realize.

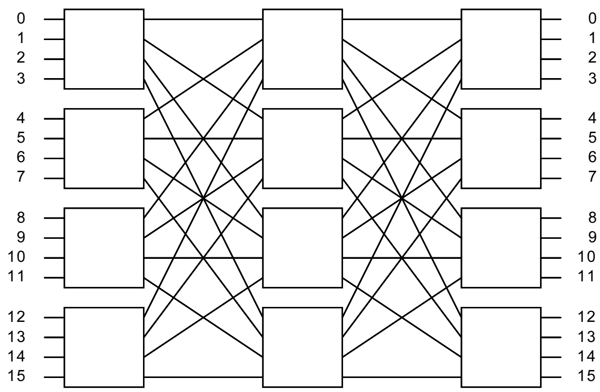


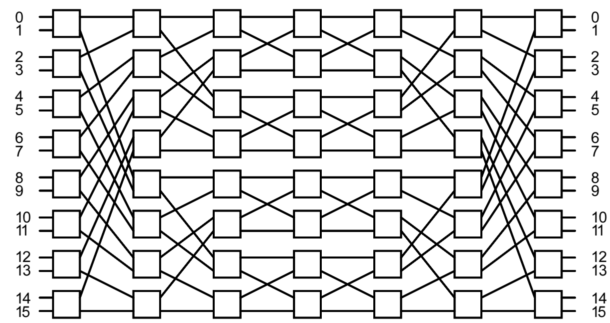
Example: Using Correspondence For Routing

Problem: How to route an $m = 2^\mu$, $k = 2^\kappa$ Clos network, $\mu > 1$ and $\mu + \kappa = n$?

Solution: Route a $N = 2^{\mu+\kappa} = 2^n$ Beneš network ...

... find Clos-network cell settings based on Beneš-network cell settings.





$$P = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 5 & 15 & 13 & 7 & 3 & 0 & 8 & 11 & 9 & 6 & 12 & 1 & 10 & 2 & 14 & 4 \end{pmatrix}$$

$$\pi_{\langle 0,0\rangle} = \begin{pmatrix} 0 & 1 & 2 & 3 \\ & & & \end{pmatrix} \quad \pi_{\langle 0,1\rangle} = \begin{pmatrix} 0 & 1 & 2 & 3 \\ & & & \end{pmatrix} \quad \pi_{\langle 0,2\rangle} = \begin{pmatrix} 0 & 1 & 2 & 3 \\ & & & \end{pmatrix}$$

$$\pi_{\langle 0,3\rangle} = \begin{pmatrix} 0 & 1 & 2 & 3 \\ & & & \end{pmatrix} \quad \pi_{\langle 1,0\rangle} = \begin{pmatrix} 0 & 1 & 2 & 3 \\ & & & \end{pmatrix} \quad \pi_{\langle 1,1\rangle} = \begin{pmatrix} 0 & 1 & 2 & 3 \\ & & & \end{pmatrix}$$

$$\pi_{\langle 1,2\rangle} = \begin{pmatrix} 0 & 1 & 2 & 3 \\ & & & \end{pmatrix} \quad \pi_{\langle 1,3\rangle} = \begin{pmatrix} 0 & 1 & 2 & 3 \\ & & & \end{pmatrix} \quad \pi_{\langle 2,0\rangle} = \begin{pmatrix} 0 & 1 & 2 & 3 \\ & & & \end{pmatrix}$$

$$\pi_{\langle 2,1\rangle} = \begin{pmatrix} 0 & 1 & 2 & 3 \\ & & & \end{pmatrix} \quad \pi_{\langle 2,2\rangle} = \begin{pmatrix} 0 & 1 & 2 & 3 \\ & & & \end{pmatrix} \quad \pi_{\langle 2,3\rangle} = \begin{pmatrix} 0 & 1 & 2 & 3 \\ & & & \end{pmatrix}$$

Correspondence Between Beneš and Clos, Formally

First, Models of Beneš and Clos Networks

Beneš Network LGM

$$I = \{\, \langle \mathrm{I},\, j \rangle \mid j \in \langle 2^n \rangle \,\} \quad O = \{\, \langle \mathrm{O},\, i \rangle \mid i \in \langle 2^n \rangle \,\}$$
$$V = I \cup O \cup \{\, \langle x,\, i \rangle \mid x \in \langle 2n-1 \rangle ,\ i \in \left\langle 2^{n-1} \right\rangle \,\}$$

$$E = \left\{ \left(\langle \mathrm{I},\, i \rangle , \langle 0,\, i' \rangle \right) \mid i \in \langle 2^n \rangle ,\ i' = \left\lfloor \frac{i}{2} \right\rfloor \right\} \cup$$

$$\{\, \left(\langle x,\, i \rangle , \langle x+1,\, i' \rangle \right) \mid x \in \langle n-1 \rangle ,\ i \in \left\langle 2^{n-1} \right\rangle ,\ d \in \{0,1\},$$
$$i' = i_{(n-2:n-1-x)}\, d\, i_{(n-2-x:1)} \,\} \cup$$

$$\{ \left(\langle x,\, i \rangle , \langle x+1,\, i' \rangle \right) \mid x' \in \langle n-1 \rangle ,\ x = x' + n - 1,\ i \in \left\langle 2^{n-1} \right\rangle ,$$
$$d \in \{0,1\},\ i' = i_{(n-2:x-n+2)}\, i_{(x-n:0)}\, d \,\} \cup$$

$$\{\, \left(\langle 2n-2,\, i \rangle , \langle \mathrm{O},\, i' \rangle \right) \mid i \in \left\langle 2^{n-1} \right\rangle ,\ \ d \in \{0,1\},\ i' = 2i + d \,\}$$

Clos Network LGM

$$I = \{\, \langle \mathrm{I},\, j \rangle \mid j \in \langle mk \rangle \,\} \quad O = \{\, \langle \mathrm{O},\, i \rangle \mid i \in \langle mk \rangle \,\}$$
$$V = I \cup O \cup \{\, \langle x,\, i \rangle \mid x \in \{0,2\},\ i \in \langle k \rangle \,\} \cup \{\, \langle 1,\, i \rangle \mid i \in \langle m \rangle \,\}$$

$$E = \left\{ \left(\langle \mathrm{I},\, i \rangle , \langle 0,\, i' \rangle \right) \mid i \in \langle mk \rangle ,\ i' = \left\lfloor \frac{i}{m} \right\rfloor \right\} \cup$$
$$\{\, \left(\langle 0,\, i \rangle , \langle 1,\, i' \rangle \right) \mid i \in \langle k \rangle ,\ i' \in \langle m \rangle \,\} \cup$$
$$\{\, \left(\langle 1,\, i \rangle , \langle 2,\, i' \rangle \right) \mid i \in \langle m \rangle ,\ i' \in \langle k \rangle \,\} \cup$$
$$\{\, \left(\langle 2,\, i \rangle , \langle \mathrm{O},\, i' \rangle \right) \mid i \in \langle k \rangle ,\ \ d \in \langle m \rangle ,\ i' = im + d \,\}$$

Mapping Function

This will *not* be a bijective map (as was used for equivalence).

Instead, several Beneš-network nodes (cells) will be mapped to a single Clos-network cell.

Input and output labels do not change:

$$f(\langle \mathrm{I},\, i\rangle) = \langle \mathrm{I},\, i\rangle\,, \qquad f(\langle \mathrm{O},\, i\rangle) = \langle \mathrm{O},\, i\rangle$$

For cells, consider a path in the Beneš and Clos networks.

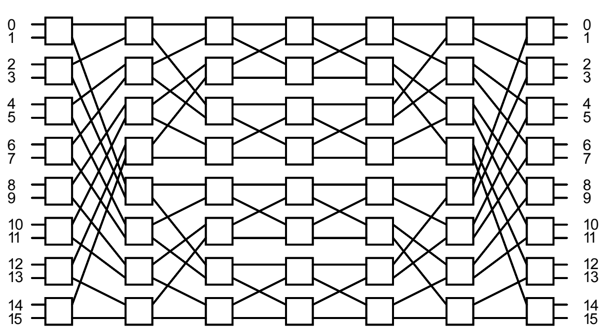
Let all the cells in both networks be set to the identity state.

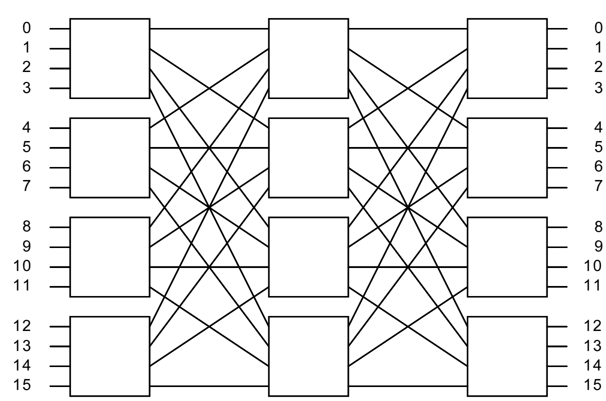
Consider stage 0 in the Clos network.

Consider stages 0 to $\mu - 1$ in the Beneš network.

Consider a path from the same input through these stages in both networks.

The cells on this path in the Beneš network are mapped to the cell on this path in the Clos network.





The equivalent mapping, mathematically:

Consider the first stage of both networks:

$$f(\langle 0,\, i\rangle) = \left\langle 0,\, \lfloor i2^{1-\mu} \rfloor \right\rangle$$

Consider the second stage of the Beneš network:

$$f(\langle 1,\, i\rangle) = \left\langle 0,\, \lfloor i2^{2-\mu} \rfloor \bmod 2^{n-\mu} \right\rangle$$

For stages 0 to $\mu - 1$:

$$f(\langle x,\, i\rangle) = \left\langle 0,\, \lfloor i2^{1-\mu+x} \rfloor \bmod 2^{n-\mu} \right\rangle$$

for $0 \le x < \mu$.

A similar procedure is followed for the last μ stages:

$$f(\langle 2n-2,\, i\rangle) = \left\langle 2,\, \lfloor i2^{1-\mu} \rfloor \right\rangle$$

$$f(\langle 2n-3,\, i\rangle) = \left\langle 2,\, \lfloor i2^{2-\mu} \bmod 2^{\mu} \rfloor \right\rangle$$

$$f(\langle x,\, i\rangle) = \left\langle 2,\, \lfloor i2^{2n-1-x-\mu} \bmod 2^{\mu} \rfloor \right\rangle$$

for $2n - 1 - \mu \le x < 2n - 1$.

The center-stages case is straightforward:

$$f(\langle x,\, i\rangle) = \left\langle 1,\, \lfloor i2^{1-n+\mu} \rfloor \right\rangle$$

for $\mu \le x < 2n - 1 - \mu$.

Proof that the Mapped Beneš Network is a Clos Network

Use the mapping function on the set of edges (from the Beneš) network LGM definition.

Eliminate self edges. (*E.g.*, $(\langle x,\, i\rangle\,, \langle x,\, i\rangle)$).

Compare the mapped-Beneš edges to the Clos edges.

If they are equal, then the mapped Beneš is a Clos network.

Let $E_Y(x)$ denote the stage-x to stage-$(x+1)$ links in the $Y \in \{\mathrm{B,C}\}$ network, where $\mathrm{I}+1=0$ and the B and C networks are just what you'd think they would be.

The input-to-first-stage edges obviously correspond:

$$E_{\mathrm{B}}(\mathrm{I}) = \left\{ \left(\langle \mathrm{I},\, i\rangle, \langle 0,\, i'\rangle\right) \mid i \in \langle 2^n\rangle,\ i' = \left\lfloor \frac{i}{2} \right\rfloor \right\}$$

$$\left\{ \left(\left\langle \mathrm{I},\, i_{(n-1:0)}\right\rangle, \left\langle 0,\, i_{(n-1:1)} \mid i \in \langle 2^n\rangle\right\rangle\right) \right\} \in E_{\mathrm{B}}(\mathrm{I})$$

$$\left\{ \left(f\left(\left\langle \mathrm{I},\, i_{(n-1:0)}\right\rangle\right), f\left(\left\langle 0,\, i_{(n-1:1)}\right\rangle\right)\right) \mid i \in \langle 2^n\rangle \right\} \in f\left(E_{\mathrm{B}}(\mathrm{I})\right)$$

$$= \left\{ \left(\left\langle \mathrm{I},\, i_{(n-1:0)}\right\rangle, \left\langle 0,\, i_{(n-1:\mu)}\right\rangle\right) \mid i \in \langle 2^n\rangle \right\} \in E_{\mathrm{C}}(\mathrm{I})$$

Beneš-network links "inside" Clos network cells form self loops.

$$\left\{ \left(\left\langle x,\, i_{(n-2:0)}\right\rangle, \left\langle x+1,\, i_{(n-2:n-1-x)}\, d\, i_{(n-2-x:1)}\right\rangle\right) \right\} \in E_{\mathrm{B}}(x)$$

$$\left\{ \left(f\left(\left\langle x,\, i_{(n-2:0)}\right\rangle\right), f\left(\left\langle x+1,\, i_{(n-2:n-1-x)}\, d\, i_{(n-2-x:1)}\right\rangle\right)\right) \right\} \in f\left(E_{\mathrm{B}}(x)\right)$$

$$= \left\{ \left(\left\langle 0,\, i_{(n-x-2:\mu-x-1)}\right\rangle, \left\langle 0,\, i_{(n-x-2:\mu-x-1)}\right\rangle\right) \right\}$$

for $0 \le x < \mu - 1$, where $i \in \left\langle 2^{n-1}\right\rangle$ and $d \in \langle 2\rangle$.

The stage-$\mu-1$ to stage-μ edges in the two networks should correspond:

$$\left\{ \left(\left\langle \mu-1,\, i_{(n-2:0)} \right\rangle, \left\langle \mu,\, i_{(n-2:n-\mu)}\, d\, i_{(n-\mu-1:1)} \right\rangle \right) \right\} \in E_{\mathrm{B}}(\mu-1)$$

$$\left\{ \left(f\left(\left\langle \mu-1,\, i_{(n-2:0)} \right\rangle \right), f\left(\left\langle \mu,\, i_{(n-2:n-\mu)}\, d\, i_{(n-\mu-1:1)} \right\rangle \right) \right) \right\} \in f\left(E_{\mathrm{B}}(\mu-1) \right)$$

$$= \left\{ \left(\left\langle 0,\, i_{(n-\mu-1:0)} \right\rangle, \left\langle 1,\, i_{(n-2:n-\mu)}\, d \right\rangle \right) \right\} \qquad \in E_{\mathrm{C}}(0)$$

where $i \in \left\langle 2^{n-1} \right\rangle$ and $d \in \langle 2 \rangle$.

Note that there is a one-to-one correspondence despite the fact that edges for Clos network are specified differently.

Proof for the other network half is similar.

Use of Map for Routing

Start with connection assignment meant for Clos network.

Route this connection assignment on the Beneš network.

Let $(a, \alpha')_x$ be the stage-x link to which input a is routed.

Set input-stage crossbars using requests (a, α'') where $f(\langle \mu,\, \alpha' \rangle) =< 1, \alpha'' >$.

Clos-Network Input and Output Stage Cells.

We have seen the correspondence between Clos and Beneš network cells.

Only μ stages of Beneš-network cells are mapped to a Clos network input- or output-stage cell.

This means the Clos network input- and output-stage cells do not have to be crossbars.

In fact, they are terminal equivalent to *inverse baseline* and *baseline* networks.

These are omega networks with the input- or output-stage shuffle removed.