# Omega-Network Connection Assignments

A connection assignment that a network can satisfy is an *admissible permutation.*

The set of all permutations admissible by an $m^n$-input omega network is denoted $\Omega_{m,n}$.

The set of all permutations admissible by an $m^n$-input inverse omega network is denoted $\Omega_{m,n}^{-1}$.

Simple lemma: If $P \in \Omega_{m,n}$ then $P^{-1} \in \Omega_{m,n}^{-1}$.

The contents of $\Omega_{m,n}$ is of interest to those:

- writing parallel algorithms and
- designing networks.

Two families of admissible permutations will be studied:

- Shift and
- Bitonic

## Shift Permutations

Used to connect input $i$ to output $pi + c$, where

$i, c \in \langle 2^n \rangle$ and $p \bmod 2 = 1$ ($p$ is odd).

A permutation is a $p$, $c$ *shift permutation* of size $2^n$, denoted $S_{p,c}$, if for all $x \in \langle 2^n \rangle$

$$S_{p,c}(x) \equiv xp + c \pmod{2^n},$$

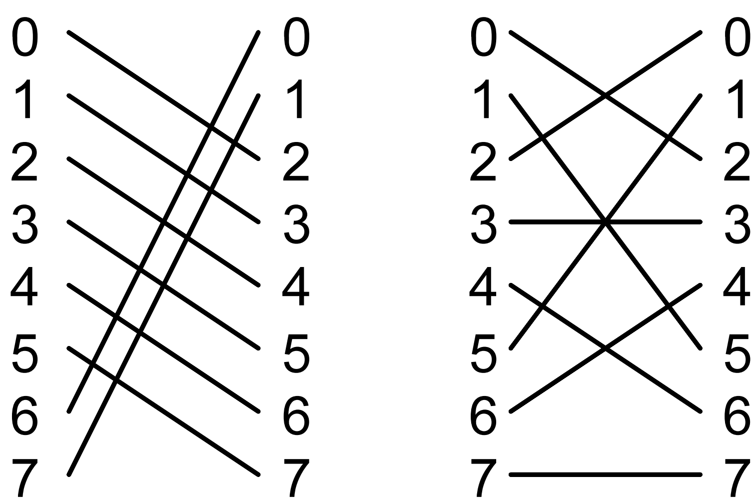
where $c$ is a nonnegative integer and $p$ is a nonnegative odd integer.

Examples:

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

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

Examples, illustrated:



The set of all shift permutations of size $2^n$ is given by

$$\mathcal{S}(2^n) = \bigcup_{x,c\in\langle\infty\rangle} \{S_{2x+1,c}\}.$$

Assertion: Any shift permutation can be satisfied by an omega network, that is, $\mathcal{S}(2^n) \subseteq \Omega_{2,n}$.

Proof Outline

Consider $A = (a,\alpha) \in S_{p,c}$ for an $N = 2^n$-input omega network.

By definition, $\alpha = pa + c$.

Consider a second request $B = (b,\beta) \in S_{p,c}$, $b \neq a$.

First prove $\alpha \neq \beta$ for all $a, b \in \langle N\rangle$.

Find the stage terminal needed by requests $A$ and $B$ at cell outputs in stage $i \in \langle n\rangle$.

Request $(a,\alpha)$ uses

$$\left\langle i,\, a_{(n-2-i)}a_{(n-3-i)}\ldots a_{(0)}\alpha_{(n-1)}\alpha_{(n-2)}\ldots\alpha_{(n-1-i)}\right\rangle.$$

Request $(b,\beta)$ uses

$$\left\langle i,\, b_{(n-2-i)}b_{(n-3-i)}\ldots b_{(0)}\beta_{(n-1)}\beta_{(n-2)}\ldots\beta_{(n-1-i)}\right\rangle.$$

Show that: $\left\langle i,\, a_{(n-2-i)}a_{(n-3-i)}\ldots a_{(0)}\alpha_{(n-1)}\alpha_{(n-2)}\ldots\alpha_{(n-1-i)}\right\rangle \neq \left\langle i,\, b_{(n-2-i)}b_{(n-3-i)}\ldots b_{(0)}\beta_{(n-1)}\beta_{(n-2)}\ldots\beta_{(n-1-i)}\right\rangle$.

for all $i \in \langle n\rangle$, $A = (a,\alpha) \in S_{p,c}$, $B = (b,\beta) \in S_{p,c}$, $a \neq b$, $p, c \in \langle N\rangle$.

This proof is simple when $p$ is limited to 1.

## Admissibility Proof for Shift Permutations, $p = 1$

Since $a \neq b$ (by definition) they differ in $\geq 1$ digit.

Let $x + 1$ be the lowest-numbered differing digit.

That is, $a_{(0:x)} = b_{(0:x)}$ and $a_{(x+1)} \neq b_{(x+1)}$.

Lemma: $\alpha_{(x+1)} \neq \beta_{(x+1)}$.

Proof:

Consider addition of $\alpha = a + c$ and $\beta = b + c$ by bits.

Let $r_{(x+1)}$ be carry to be added to digit $x + 1$.

Since $c$ and digits up to $x$ in $A$ and $B$ are identical...

...the carry $r_{(x+1)}$ must also be identical.

Bitwise addition:

$\alpha_{(x+1)} = a_{(x+1)} \oplus c_{(x+1)} \oplus r_{(x+1)}$

and $\beta_{(x+1)} = b_{(x+1)} \oplus c_{(x+1)} \oplus r_{(x+1)}$.

Since only difference is $a_{(x+1)} \neq b_{(x+1)}$, then $\alpha_{(x+1)} \neq \beta(x + 1)$.

Back to admissibility proof:

In stage $i$,...

... either $a_{(0:n-2-i)} \neq b_{(0:n-2-i)}$ or $\alpha_{(n-1-i:n-1)} \neq \beta_{(n-1-i:n-1)}$...

...either way there is no contention.

# Admissibility Proof for Shift Permutations, $p$ Odd

Similar to proof above:

Let $x+1$ be the lowest-numbered differing digit:

$a_{(0:x)} = b_{(0:x)}$ and $a_{(x+1)} \neq b_{(x+1)}$.

Lemma: $\alpha_{(x+1)} \neq \beta_{(x+1)}$.

Proof:

Need to compare $(ap)_{(x+1)}$ and $(bp)_{(x+1)}$.

$$(ap)_{(x+1)} = \left( \sum_{z=0}^{x+1} p_{(z)} a_{(x-z+1)} \right) + R_{x+1} \bmod 2,$$

where $R_{x+1}$ is the carry.

Splitting the sum yields:

$$(ap)_{(x+1)} = p_{(0)} a_{(x+1)} + \left( \sum_{z=1}^{x+1} p_{(z)} a_{(x-z+1)} \right) + R_{x+1} \bmod 2,$$

Since $a_{(0:x)} = b_{(0:x)}$ expressions for $(ap)_{(x+1)}$ and $(bp)_{(x+1)}$ differ ...

... only in $p_{(0)} a_{(x+1)}$ and $p_{(0)} b_{(x+1)}$ ...

... (noting that since $p$ is odd, $p_{(0)} = 1$)...

... and so $(ap)_{(x+1)} \neq (bp)_{(x+1)}$...

... and therefore $\alpha_{(x+1)} \neq \beta_{(x+1)}$.

Remainder of proof is the same:

In stage $i$,…

… either $a_{(0:n-2-i)} \neq b_{(0:n-2-i)}$ or $\alpha_{(n-1-i:n-1)} \neq \beta_{(n-1-i:n-1)} \cdots$

…either way there is no contention.

## Bitonic Permutations

A sequence of numbers is *bitonic* if the magnitude of the numbers first increases then decreases, or if the magnitude of the numbers first decreases then increases.

Examples:

1,1,2,5,5,4,0

1,2,1

5,3,0,8

Not bitonic: 1,2,0,3 and 1,2,3,4.

The definition of a bitonic permutation will have a slight difference:

The sequence is a sequence of integers,

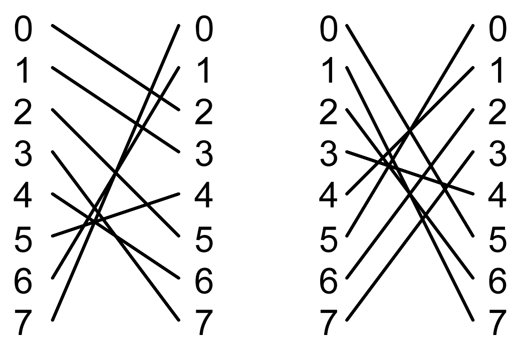
the integers are a permutation of $\langle m^n \rangle$, and

the sequence *when shifted* is bitonic.

A permutation for which the sequence $P(c)$, $P(c+1)$, $P(c+2), \ldots,$ $P(c+m^n-1)$ is bitonic is called a *bitonic permutation*, where $P$ is a permutation of $\langle m \rangle^n$, $c$ is an integer, and arithmetic is modulo $m^n$.

Examples:

$$P_1 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 2 & 3 & 5 & 7 & 6 & 4 & 1 & 0 \end{pmatrix} P_2 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 5 & 7 & 6 & 4 & 1 & 0 & 2 & 3 \end{pmatrix}$$



Theorem: Let symbol $\mathcal{B}(m^n)$ denote the set of all bitonic permutations of $\langle m^n \rangle$. Then $\mathcal{B}(m^n) \subseteq \Omega_{m,n}$.

Proof Introduction:

Let $P \in \mathcal{B}$ and $(a, \alpha) \in P$ and $(b, \beta) \in P$, $(a \neq b)$.

Link used by $(a, \alpha)$ at cell output in stage $x$ is…

… $a_{(n-x-2)} a_{(n-x-3)} \cdots a_{(0)} \alpha_{(n-1)} \alpha_{(n-2)} \cdots \alpha_{(n-x-1)}$.

Similarly $(b, \beta)$ uses $b_{(n-x-2)} b_{(n-x-3)} \cdots b_{(0)} \beta_{(n-1)} \beta_{(n-2)} \cdots \beta_{(n-x-1)}$.

To prove $(a, \alpha)$ and $(b, \beta)$ don't share a link…

…sufficient to show that…

…if $\alpha_{(n-1)} \cdots \alpha_{(n-x-1)} = \beta_{(n-1)} \cdots \beta_{(n-x-1)}$

…then $a_{(n-x-2)} a_{(n-x-3)} \cdots a_{(0)} \neq b_{(n-x-2)} b_{(n-x-3)} \cdots b_{(0)}$.

Proof Introduction In words:

Call $a_{(n-x-2)} a_{(n-x-3)} \cdots a_{(0)}$ and $b_{(n-x-2)} b_{(n-x-3)} \cdots b_{(0)}$ the (input) *LSDs*.

Call $\alpha_{(n-1)} \cdots \alpha_{(n-x-1)}$ and $\beta_{(n-1)} \cdots \beta_{(n-x-1)}$ the (output) *MSDs*.

Need to prove: if the MSDs of two requests match,…

…the LSDs must be different.

For rest of proof consider only requests with

$$\alpha_{(n-1)} \cdots \alpha_{(n-x-1)} = \beta_{(n-1)} \cdots \beta_{(n-x-1)}$$

## Proof Observation:

Because $P$ is a permutation…

…for each choice of $\alpha_{(n-1)}\cdots\alpha_{(n-x-1)}\cdots$

…there are $m^{n-x-1}$ requests in $P$.

**If** the LSDs of the inputs of those $m^{n-x-1}$ requests are distinct…

…(*e.g.*, $a_{(n-x-2)}a_{(n-x-3)}\cdots a_{(0)} \neq a_{(n-x-2)}a_{(n-x-3)}\cdots a_{(0)}$)

…then $(a,\alpha)$ and $(b,\beta)$ won't share a link.

So, that's what will be proven.

## Proof Outline:

Show that for a given $\alpha_{(n-1)}\cdots\alpha_{(n-x-1)}$ …

…the possible input numbers form up to 3 runs of consecutive numbers.

(By property of bitonic sequences.)

Show that gaps between runs contain a multiple of $m^{n-x-1}$ inputs.

(Show directly for $\alpha_{(n-1)}\cdots\alpha_{(n-x-1)} = m^{x+1}-2$, proceed with induction on $\alpha_{(n-1)}\cdots\alpha_{(n-x-1)}$.)

Show a bijection between $m^{n-x-1}$ consecutive integers…

…and the inputs.

For example consider $P = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 3 & 7 & 6 & 5 & 4 & 2 & 0 \end{pmatrix}$ for $\Omega_{2,3}$.

In stage 0 let $\alpha_{(2)} = 1$.

Then inputs form one run: 2,3,4,5 and there is no gap.

In stage 0 let $\alpha_{(2)} = 0$.

Then inputs form two runs: 0,1 and 6,7 with a gap of 4.

Note that LSDs of inputs are 0,1,2,3. (LSDs might need to be sorted.)

In stage 1 let $\alpha_{(2:1)} = 11_2 = 3$.

Then inputs form one run: 2,3.

LSDs form sequence 0,1.

In stage 1 let $\alpha_{(2:1)} = 10_2 = 2$.

Then inputs form one run: 4,5.

In stage 1 let $\alpha_{(2:1)} = 01_2 = 1$.

Then inputs form two (single-digit) runs: 1 and 6 with a gap of 4.

## Application: Spreading, Copying, and Packing

The bitonic permutations are related to three useful families of connection assignments:

*Spreading Connection Assignment:* A 1-limited GCA (generalized connection assignment) in which consecutive inputs are routed to outputs, preserving order.

*Copy Connection Assignment:* An $N$-limited GCA in which consecutive inputs are routed (multicast) to outputs, preserving order.

*Packing Connection Assignment:* A 1-limited GCA in which a subset of inputs is connected to consecutive outputs, preserving order.

## Spreading Connection Assignments

Examples:

$\{(0,2),(1,5),(2,7)\}$ is a spreading CA.

$\{(0,2),(2,5),(3,7)\}$ is not a spreading CA. (Input 1 is skipped.)

$\{(0,2),(1,7),(2,5)\}$ is not. (The requests do not appear in the same order when sorted by outputs.)

Assertion: An omega network can satisfy all spreading connection assignments.

Proof outline:

It is known that an omega network can realize all bitonic permutations.

It will be shown that a bitonic permutation can be constructed from any spreading CA.

Consider $\{(0,2),(1,5),(2,7)\}$:

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

Construct a bitonic permutation by adding $(3,6)$, $(4,4)$, etc.

It can easily be shown that this procedure will work in all cases.

## Copy Connection Assignments

Examples:

$$\{(0,2),(0,3),(1,5),(2,6),(2,7)\}$$

In the CA above, two "copies" made of data at inputs 0 and 2. One copy made of data at 1.

These can be realized in omega networks with broadcast capability.

In such networks a single cell input must be able to connect to both outputs.

Assertion: All copy CAs can be satisfied by an omega network.

Proof outline:

Proof is by contradiction.

Suppose there is a copy CA that cannot be realized.

Let $X$ be such a CA.

For at least one cell, two requests in $X$ from different inputs must need the same cell output.

Call the requests $A = (a, \alpha)$ and $B = (b, \beta)$.

By definition of $A$ and $B$, $a \neq b$.

Construct a spreading CA, $X'$ in the following way:

Put $A$ and $B$ in $X'$.

Add one request for each of the other inputs in $X$ to $X'$.

The result is a spreading CA, which can be satisfied by an omega network.

Since paths in an omega network are unique, if $A$ and $B$ do not conflict in $X'$ they cannot conflict in $X$.

## Packing Connection Assignments

These are the mirror image (inverse) of spreading CAs.

Examples:

$\{(3,0),(7,1),(9,2)\}$ is a packing CA.

$\{(4,2),(7,3),(11,4)\}$ is a packing CA.

Assertion: An inverse omega network can satisfy all packing connection assignments.

Proof outline:

Show that packing CA is mirror image of spreading CA.

If $P \in \Omega$ then $P^{-1} \in \Omega^{-1}$.