network with other crossoffers or buses
very often used as a component in a larger more complex

0 is on the order 400

very fast, but typically small - u in the biggest crossoffer

high cost - (m\(\times\)n) \(O\) switches or gates

development a crossoffer can be used to have full connectivity between m sources and u

a bus is essentially a sequential medium

Crossbar
still $O(n^2)$ complexity in gates
can also be built out of MUX-DEMUX.
many can be set in each row and column
various designs place restrictions on how
each crosspoint is an independent switch

Memory

Crossbar Interconnect

Processors
Related to the study of permutations and sorting •

Require fast local routing and low latency and area

Computer networks for processor memory interconnects •

Substantial work done on networking for commercial •

An attempt to get full connectivity (or as near as

MULTITASK INTERCONNECTION NETWORKS
Clos-Benes networks are best example

- Connectivity (all or some permutations)
- Control cost (setup of switches)
- Latency (time or depth)
- Area (gates)

Parameters

- Many different types

Optimal Rearrangeable Alignment Networks (ORANs)
Clos-Benes 4 by 4

to (0, 1, 2, 3) to (0, 2, 1, 3)
P denotes parallel connect in switch
C denotes crossover connect in switch

for global broadcast connections
Lower or upper broadcast also available

P P P
(0, 1, 2, 3) to (0, 2, 1, 3) C C C

P P P
(0, 1, 2, 3) to (0, 1, 2, 3) P P P

Clos-Benes 4 by 4
Clos-Benes 8 by 8

shuffle

unshuffle
not acceptable in high-performance memory

\[ \text{time} (2 \log n - 1) \text{ levels} \]

\[ \text{complexity} \left( \frac{2}{u} \right) (\log n - 1) \times 2 \times 2\]

on whole

high control costs \( O(n \log n) \), i.e., path for each depends

complete permutations

compare the identity settings to a nontrivial permutation

note multiple paths from 1 to 2
only one maximum and one minimum.

Essentially if the sequence is viewed in a ring fashion and the interchanged.

descending. If remains pristine if split and the parts of two monotonic sequences (one ascending and one juxtaposition.

Definition: A pristine sequence is the juxtaposition due to Batchelor.

we first consider a special sequence and sorting circuit sort in hardware a list of numbers.

to insure full permutation connectivity consider how to

Sorting Networks
\[
\begin{align*}
(17, 13, 11, 5, 4, 3) &= \varphi \\
(1, 3, 5, 6) &= \varphi
\end{align*}
\]

Required properties:

\[
\text{Define } p.\quad \text{min}(a, b) = \begin{cases} a & \text{if } a \leq b \\ b & \text{if } a > b \end{cases}
\]

Both are bijective sequences since they are the same when split.

\[
\begin{align*}
(17, 13, 11, 5, 4, 3, 1, 3, 5, 6, 12, 18) &= \varphi \\
12 &\geq 1
\end{align*}
\]
In sorted order, the parallel or cross selection to output the two input values switches which looks for the leading one position and chooses.

Note that the $d$ and $e$ generation is easily done via a $2$

\[
\begin{align*}
(17, 13, 11, 6, 12, 18) &= e^2 \\
(1, 3, 5, 6, 12, 18) &= ?^{+u} a
\end{align*}
\]
to get a completely sorted sequence.

property so we can apply the theorem recursively 

appropriately to sort separately and that have the biotonic

So we can split 4 into two sequences that are bounded

\[
\begin{align*}
\max & (p_1, \ldots, p_r) \\
& \geq (\min(e_1, \ldots, e_r))
\end{align*}
\]

and et al. Furthermore,

**Definition:** If \( r \geq 2n \), is biotonic then so are \( d \).

So what? The following theorem allows the construction of a

biotonic sorter.
can be used to get a general sorter

\[ m \leq \log m \]

\[ \text{gates} \text{ (area)} \leq \log m \]

\[ \text{time} \text{ (depth)} \leq \log m \]

A bitonic sorter on a sequence of size \( m = 2^p \)
to do better we must sacrifice some connectivity

\( O( \log^3 m) \) detection on two inputs - it is essential leading ones

control is local since it is essentially leading ones

\( O( \log^3 m \log m) \) time (depth)

\( O( \log^3 m \log m) \) gates (area)

SIMPLE ALTERNATIONS IN CONNECTIVITY

DESCENDING SEQUENCES (THIS ENTAILS

NOTE THE ALTERNATION IN ASCENDING AND

•
simple patterns or repeated shuffles
very popular with higher-degree crossovers as basic switch
easy local routing decision
acceptable area $O(m \log m)$
less delay $O(\log m)$
as a result BLOCKING POSSIBLE
but not all
supports all main permutations — shifts, reversals, etc.
Omega Networks
Routing algorithm uses binary pattern of destination:

Use bit i on stage i, (leftmost bit first)

8 by 8 Omega network using 2 by 2 crosstersaps
8 by 8 Omega network using 2 by 2 crossbars

Blocking is possible as seen on this permutation which is blocked at marked gates
Reverse network for a write.

- on forward network and address (acknowledge) on
- and data on reverse network for a read, address and data
- are normal – address on forward network and address
- shared memory traffic tends to assume small packet sizes
- arbitration strategies to handle blocks and traffic flow
- each stage must have queues, throttle mechanisms, and
- permutations
- In reality, the system does not operate in lock step with
- q-way switches where q is the order of the switch.
- Higher order switches may be used – requires the use of
• deadlock is not possible since there is always someone taking traffic off the network, i.e., simple source to drain model
• latency mitigation is the main design issue (prefetch and multiple outstanding requests to memory)
• key model issue: number of hops versus wire length
• key basis for algorithm design and data layout
• modeling of contention and latency and their effects on performance are key basis for algorithm design and data exploitation spatial locality
• sometimes multiple words are taken from each bank to
16 by 16 Omega network using 4 by 4 crossbars

two 4-way shuffles