not in cache) from a processor is the same

the cost to access all locations in physical memory

such a system has uniform memory access time, i.e.,

other

on one side of the network and memory banks on the

physically centralized memory places processors

of the memory system

banks in the system determines is the basic attribute

the relative placement of the processors and memory

Architectural Paradigm
nonuniform memory access time
the memory bank in its pair this system has a
since the processor has a distinct and faster path to
other pairs via the network
and memory bank in a pair that communicates with
physically distributed memory places a processor
(read or written)

request needs to be involved when data is accessed
therefore, no processor other than the one issuing the
physical location in the entire memory system

Any and can supply the associated data
banks can supply the associated data
and memory can issue an address and the network and memory
physically shared memory means that a processor

the memory system
the communication support is the key aspect of
needs it.

by explicitly sending the data to the processor that
owns the data must be involved

bank that contains the data.

sent by the processor in the pair with the memory
another pair it must receive the data in a message
processor wants data located in the memory bank in
memory bank in the same pair as the processor. If a
issued from a processor is only interpreted in the
physically private memory means that an address
memory

version and is called *incorporated physically shared*

updated versions of the data it may not get the latest

memory and not check caches for more recently

in the system simply goes to addresses A in physical

retrieving address A will get the latest contents

may or may not guarantee that a processor

on a physically shared memory system the system

system with d processors

can be 1 copies or an addresses data in the a

if each processor has a cache then potentially there

Caches and Parallel Processes
system is coherent physically shared memory

- latest copy of the contents of address A then the
  - if the system supports automatically finding the
    - code
  - shared memory and can often produce very efficient
  - expert users are required to program in coherent
  - nonexpert users, e.g., data parallel is often used
  - restricted more than pure shared memory for
  - for such systems the programming paradigm is often
  -
There is no reason for this other than engineering history. In fact, both types of networks are now used for both types of architectures.

Traditionally direct networks are used for distributed memory

Indirect (dynamic)

Direct (static)

There are two basic classes of networks
performance are dominant theory (where isomorphism, embedding and theoretical between practice (where details make the decision) and not architectural dynamic purity so there is still a gap. Most networks are designed to address engineering tradeoffs (incoherent physically shared memory). For example, the Cray T3E is a 3-D toroidal mesh with direct...
building block in combination with other interconnects

used at many different levels of the system, often as a

broadcast with selective read

control single communication pathway seen by all

multiple resources (often very heterogeneous) compete to

simplest and most often used.

Bus interconnection
network with other crossbars or buses

- Very often used as a component in a larger more complex

  is on the order 400

  very fast, but typically slower - in the biggest crossbar

  high cost - \( O(mn) \) switches or gates

  destinations a crossbar can be used
to have full connectivity between \( m \) sources and \( n \)

  a bus is essentially a sequential medium

Crossbar
various designs place restrictions on how many can be set in each row and column. Each crosspoint is an independent switch and can also be built out of MUX-DEMUX's, still \( O(n^2) \) complexity in gates.
related to the study of permutations and sorting

require fast local routing and low latency and area

computer networks for processor memory interconnects

interesting stuff

reasons e.g., the phone company did most of the
substantial work done on networking for commercial

\( O(n \log n) \) in gates.

An attempt to get full connectivity (or as near as

MULTISTAGE INTERCONNECTION NETWORKS
Clos-Benes networks are best example

- Connectivity (all or some permutations)
- Control cost (set up of switches)
- Latency (time or depth)
- Gates (area)

Parameters:

- Many different types

Optimal Rearrangible Alignment Networks (ORAN)
Clos-Benes 4 by 4

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

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

For global broadcast connections, lower or upper broadcast also available.
P denotes parallel connect in switch.
C denotes crossover connect in switch.
Clos-Benes 8 by 8

unshuffle

shuffle
not acceptable in high-performance memory

\( \frac{n \log u}{u} \quad \text{levels} \)

\( 2 \times 2 \log \frac{u}{2} \)

complexity

on whole

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

complete permutations

compare the identity settings to a nontrivial permutation

note multiple paths from \( i \) to \( j \)
only one maximum and one minimum.

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

descending). It remains bijective if split and the parts of two monotonic sequences (one ascending and one descending) the juxtaposition is the insertion position.

Definition: A bijective sequence is the juxtaposition due to Batcher.

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
Required properties.

matter which splitting you use, $d$ and $e$ will have the

Define $d = \min(a^n, a^{n+1})$ and $e = \max(a^n, a^{n+1})$. It does not

in two different places.

Both are bitonic sequences since they are the same split

\[
(17, 13, 11, 5, 4, 3, 1, 3, 5, 6, 12, 18) = \begin{cases} a^u \\ \forall n \geq 1 \\ \forall n \geq 1 \end{cases}
\]

\[
(1, 3, 5, 6, 12, 18, 17, 13, 11, 5, 4, 3) = \begin{cases} a^u \\ \forall n \geq 1 \\ \forall n \geq 1 \end{cases}
\]
In sorted order, the parallel or cross setting to output the two input values which looks for the leading one position and chooses one that the \( \mu \) and \( \nu \) generation is easily done via a \( \Omega \),

\[
\begin{align*}
(17, 13, 11, 6, 12, 18) &= \ \varepsilon \\
(1, 3, 5, 6, 12, 18) &= ?^a_u \nu \\
(17, 13, 11, 5, 4, 3) &= ?^a_v \\
(1, 3, 5, 4, 3) &= ?^a_p
\end{align*}
\]
to get a completely sorted sequence. Property so we can apply the theorem recursively \( \log n \) times and separately that have the binary.

So we can split \( a_i \) into two sequences that are bounded by:

\[
\max (d_1, \ldots, d_m) \leq \min (e_1, \ldots, e_n)
\]

and \( e_i \) is bounded.

Furthermore, if \( a_i \) is binary then so are \( a_i \).

**Definition:** If \( a_i \) is \( \geq 2^n \), it is binary.

So what? The following theorem allows the construction of a binary sort.
2^n Bitonic Sequence Sorter

n-item bitonic sorter

sorted list

C_1
C_n
C_{n+1}

a_1
da_2
da_3

a_{2n-2}
a_{2n-1}
a_{2n}
a_{2n+1}
a_{2n+2}
a_{2n+3}

C_{2n}

a_{n-2}
a_{n-1}
a_n

a_{n+1}
a_{n+2}
a_{n+3}

C_{n+1}
C_{2n}
can be used to get a general sort

\[ \frac{\text{area}}{m} \] \( \log m \) •

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

\[ n = 2n = 2d \] A bitonic sorter on a sequence of size \( m \) = \( 2n \) •
to do better we must sacrifice some connectivity

\( O(m \log^2 m) \) detection on two inputs

control is local since it is essentially leading ones

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

\( O(m \log^2 m) \) gates (area)

SIMPLE ALTERNATIONS IN CONNECTIVITY

DESCENDING SEQUENCES (THIS ENTAILS
NOTE THE ALTERNATION IN ASCENDING AND
simple pattern of repeated shuffles •

very popular with higher degree crossings 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 – shits, 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 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 shuffle where q is the order of the switch.

Higher order switches may be used – requires the use of

•
model taking traffic off the network, i.e., simple source to drain

deadlock is not possible since there is always someone
multiple outstanding requests to memory

Latency mitigation in the main design issue (prefetch and

dominant model

Key model issue: number of hops versus wire length

Layout

Performance are key basins for algorithm design and data

modelling of contention and latency and their effects on

exploit spatial locality

sometimes multiple words are taken from each bank to
16 by 16 Omega network using two 4-way whuffles

4 by 4 crossbars

16 by 16 Omega network using
Scalability - ease of expansion

Hardware complexity - cost of implementation

Bandwidth - maximum data transfer rate

Network latency - worst case time for unit message

Functionality - support for various operations

Performance criteria:

- Control strategy: centralized or distributed
- Switching method: circuit or packet
- Timing protocol: synchronous or asynchronous
- Topology: assumed static here

Operational Characteristics:

Direct or Static Networks
avoiding contention

core connectivity (and thus arc connectivity) is desirable for

distinct parts; it is a measure of core connectivity. High

to be removed in order to break the network into two

core connectivity is the minimum number of links that need

multiplicity of paths between processors. Arc

Arc core connectivity: core connectivity is a measure of the

(hops) that need to be traversed between processors.
The distance is the least number of links

Diameter: maximum distance between any two

Interconnects: Evaluation
volume of traffic that can be handled by the network. When a network is bisected, the width is a measure of the volume of traffic that can be handled by the network. When a network is bisected, the width is a measure of the volume of traffic that can be handled by the network. The minimum number of links that need to be removed in a network to separate the processors into two halves. Bisection width: the minimum number of links that need to be removed in a network to separate the processors into two halves. Bisection width: the minimum number of links that need to be removed in a network to separate the processors into two halves. Bisection width: the minimum number of links that need to be removed in a network to separate the processors into two halves.
Bisection width of a ring is $2$.

Arc connectivity of a ring is $2$.

Diameter of a ring is $\left\lceil \frac{\pi}{d} \right\rceil$.

Each processor connected to $2$ other processors.

Interconnects: Rings.
Processes can be increased without increasing dimension

Mesh with wrap around - torus

In practice, only 2 or 3 dimensional meshes are constructed

Each processor in an $d$ dimensional mesh is connected to $2d$ other processors except for the corner processors

Interconnected Multidimensional Meshes

of mesh
Multi-dimensional Meshes
tours it is $2^d$.

- Bisection width of a 2-dimensional mesh is $2^d$; for a torus it is 4.
- Arc connectivity of a 2-dimensional mesh is 2 and for a torus it is 4.
- Diameter of a 2-dimensional mesh is $2(1 - d/2)$; diameter of a 2-dimensional mesh is 2.

**Interconnects:** Multidimensional Meshes
is main problem

Cost and performance of increasing complexity switches

Higher levels in the tree

A fat tree solves the problem by using wider links at

shared among all processors below creating bottlenecks

In a simple tree, the bandwidth on higher level links is
between a pair of processors

A tree network is one where there is exactly one path

Interconnects: Fat Tree
• Bisection width is $\frac{p}{2}$ for a fat tree.

• Arc connectivity is 1

• Diameter of a fat tree is $2\log(1 + d)(1/2)$

Interconnects: Fat Tree
Interconnects: Fat Tree
new hypercube

original hypercubes are prefixed with a 0 or a 1 for the
dimensionional hypercube, the labels on the nodes of the
dimensionional hypercubes is used to construct a $p + 1$
dimensional hypercube can be constructed recursively when two $p$
hypercubes are connected with $p$ other processors.

In a $p$ dimensional hypercube, each processor is

two processors in each dimension

A hypercube is a multidimensional mesh with exactly

Interconnects: Hypercube
algorithms

• Used in logic design, error correcting codes, graph

• It is the minimal path length between two processors

For example if \( s = 001 \) and \( t = 101 \) distance is 1

which their labels differ

distance defined to be the number of bit positions in

Two processors with labels \( s \) and \( t \) have a Hamming

connected to exactly \( p \) other processors

• In a \( d \) dimensional hypercube, each processor is

exactly one bit position

Two processors are connected if their labels differ in

Interconnects: Hypercube
\[ \frac{\gamma}{d} \]

- Dilation width of a hypercube
- (\(\infty \leftarrow d\) problem as scaling) (scaling) (d) (log) (d) (d) (log)

- Arc connectivity of a hypercube is (smaller than others)

- Diameter of a hypercube

Interconnects: Hypercube
connected cycle

replace each node in the hypercube to create a k-cube

of the dimension k of the hypercube. a ring of k nodes can

In order to avoid the growing degree of the node as a function
k-cube connected cycle $k=3$ K * 2 nodes

3 dimensional hypercube $k=3$ K

2 nodes $k=3$ K
Note that those are all similar to shared memory issues.

- Try to fetch large amounts of data at the same time
- Latency amortization
- Progress

- Do some other work while a data transfer is in progress
- Latency hiding

By a combination of hardware and software

- Deadlock can occur. In general, needs to be addressed
- Traffic patterns on the network can create hot spots

Effects of Interconnects on Programming
avoid congestion

- Non-minimal may sometimes take a longer path to
  (several min. paths)
- Minimal always takes the shortest path (maybe)

Minimal or non-minimal routine:

destination processor

through the network when going from a source to a

Routine mechanism determines the path a message takes

Message Routing Mechanisms
Deterministic or adaptive:

- Deterministic routine always takes the same path between processors (typical of shared memory, and
  older distributed memory)

- Adaptive routing can take different paths depending on congestion (becoming more common)
be used to get a deterministic minimal path
s gives the bits that have to be switched, any order can

\( p \) is reached

Continue this process at each successive processor until

from \( p \) where \( k \) is the least significant non-zero bit in \( s \)
compute \( s = p \oplus p \).
Send message along dimension \( k \)
The message is at processor \( p \) and needs to go to \( p \).

Example: E-Cube Routing in Hypercubes
Scheme is minimal and deterministic

destination processor

Now message is sent along Y dimension till it reaches coordinate is reached

Message sent along X dimension till destination X

Example: XY Routing in 2d Mesh