Scalable Network Architectures Using The Optical Transpose Interconnection System (OTIS)

Francis Zane†, Philippe Marchand‡, Ramamohan Paturi†, and Sadik Esener‡
† Department of CSE, ‡ Department of ECE
University of California, San Diego
9500 Gilman Drive, San Diego, CA 92122
{fzane, pmarchand, rpaturi, sesener}@ucsd.edu

Abstract

The Optical Transpose Interconnection System (OTIS) proposed in [13] makes use of free-space optical interconnects to augment an electronic system with non-local interconnections. In this paper, we show how these connections can be used to implement a large-scale system with a given network topology using small copies of a similar topology. In particular, we show that, using OTIS, an N^2 node 4-D mesh can be constructed from N copies of the N-node 2-D mesh, an N^2 node hypercube can be constructed from N copies of the N-node hypercube, and an $(N^2, \alpha^2, c/2)$ expander can be constructed from N copies of an (N, α, c) expanders, all with small slowdown. We also show how this expander construction can be used to build multibutterfly networks in a scalable fashion. Finally, we demonstrate how the OTIS connections can be used to produce a bit-parallel crossbar using many copies of bit-serial crossbars with minimal overhead.

1. Introduction

In principle, optical interconnect technologies offer several advantages over electrical systems. Connections can be made at higher speeds with less crosstalk and less power consumption than electrical channels. The power required is nearly independent of the length of the connection, at least over the lengths of connections involved within a parallel machine. While some routing of optical links is possible using lenses and computer-generated holograms (CGH), arbitrary connections are harder to implement as space-variant optics than as wires on a VLSI circuit, multi-chip module or printed circuit board. In this paper, we attempt to demonstrate how a combination of optical and electrical technologies can achieve many of the advantages of optics as well as the routability of electronics.

The results of this paper focus on fine-grained, massively parallel systems, consisting of many chips with many processing elements (PEs) per chip. Using the optical transpose global interconnections, we show how large systems can be built from smaller subsystems. These smaller pieces could then be implemented on individual chips, with connections made with on-chip electrical lines. Beginning with several copies of the very simple 2-D mesh topology, we show that the addition of the OTIS connections actually creates the more powerful 4-D mesh topology. In a similar fashion, the optical connections allow many small hypercube networks to simulate a large hypercube. The same techniques can also be used to construct large expanders and splitters, randomly wired graphs useful in many routing applications. Normally, the random connections of these graphs prohibit large-scale implementations. Finally, the same techniques can be applied to construct crossbars which switch entire words in parallel by connecting bit-serial crossbar chips using the optical transpose.

2. Optical Transpose Interconnection System

The Optical Transpose Interconnection System (OTIS) is an optoelectronic Multistage Interconnection Network (MIN) developed for parallel processing systems [13]. In an OTIS based free-space optoelectronic MIN, electronic bypass-and-exchange switches are required to do the local

routing. It has been shown that for an optoelectronic MIN with N^2 inputs and N^2 outputs, the bandwidth and the power consumption of the network are optimized if the electronic switch planes are partitioned into N switches [10]. Thus, in an OTIS based parallel system, N^2 processor nodes are logically divided into groups of N nodes each. In practice, these groups can be thought of as being implemented by a single chip, or perhaps a small number of densely-connected chips. Connections between groups are achieved via free-space optics: each processor node has an optical transmitter/receiver pair with which it sends and receives optical signals. Transmitters and receivers are connected via two planes of lens arrays each consisting of N lenses. These optical links connect the pth processor of the gth group to the gth processor of the pth group: a transpose of group and position coordinates. These optical connections also can be thought of as higher-order generalizations of the shuffle edges in the perfect shuffle (Fig. 1).

The OTIS implementation offers several advantages for large scale interconnection networks. In general, optical technologies offer the advantage that the bandwidth and power requirement of a link are independent of the length of the link. For example, it has been shown in the literature [3, 8, 7] that for line lengths greater than a few millimeters free–space optical interconnections require less power for a given bandwidth. In addition, free space optoelectronic systems also facilitate electronic layout due to the fact that the I/O can be performed over the 2–dimensional area of the chip and is not limited to the 1–dimensional boundary of the chip.

The OTIS topology offers some additional advantages: the system can be folded to establish connections from a chip onto itself, it can be made bidirectional to provide interconnections between two processing planes, or it can be cascaded to accommodate successive processing planes. It can also accommodate any arbitrary optoelectronic layout as long as the layout of all the groups are identical. Finally, the optics in OTIS can be designed to allow bit-serial and/or bit-parallel communications between node (g, p) and node (p, g) in the system.

OTIS relies on optoelectronic flip-chip bonding technology and free–space optics. The integration of 8,000 optical transmitters and detectors on a single 0.8 micron silicon chip using flip–chip bonding has been demonstrated [6]. A simple free-space optical system for which 4096 bidirectional channel connections has also been demonstrated [13]. To validate the concept of OTIS an experimental chip is now being built which combines electronic switches [9] with the AT&T technology for optoelectronic device integration on silicon.

Extensive modeling of an OTIS based switching system using the OTIS-Hypercube topology has also been performed [5]. Performance of the system in terms of throughput and cost in terms of system power consumption, area, volume, and maximum power dissipation per unit area have been computed. The modeling includes the VLSI switches, the optoelectronic receivers and transmitters, the optical interconnection system, and the main laser required to power-up the modulators and its associated optics. Note that in this modeling, it is assumed that the VLSI switches contain circuits to detect hot spots and allow the OTIS network to resend data packets that have been dropped due to contention [9]. This modeling also assumes no contention in the network, a single switch plane, and the required optics to fold the interconnections back onto the chips. It has been shown that for such a 4096 channel system running at a clock speed of 125 MHz the following numbers can be achieved:

Total Throughput	1 Tbits/sec
Total Power Consumption	55 W
Optical Power at the plug	10 W
Electrical Switch Power	25 W
Receivers Electrical Power	10 W
Transmitters Electrical Power	10 W
Total Silicon Area	8.8 cm^2
Longest electrical wire	2.2 mm
Power/channel	13 mW
Area/channel	460x460 $microns^2$
Power Density	5.6 W/cm^2

The modeling results of the OTIS based switching system are very encouraging in terms of the feasibility of a large scale implementation (4096 channels) of the system. It can be seen that although the required silicon area is quite large (8.8 cm^2) it can be tiled into smaller chips since the longest wire is only 2.2 mm long which makes Multi Chip Module implementation relatively easy. As mentioned previously, OTIS will be able to accommodate such a layout. In addition, a total power requirement of 55 W is low for such a large system and the power density projections (below 10 W/cm^2) remain within air cooling limits. If packaging issues related to integrating free-space optics with optoelectronic chips can be resolved at a reasonable cost, this system will prove competitive with electronic alternatives.

3. Terminology

In the rest of the paper, we present emulations of various topologies by networks augmented by OTIS. In order to keep the notation simple, N refers to the size of a group in OTIS (i.e., OTIS networks have size N^2.) Also, OTIS processors will be referred to by pairs (g, p), where g represents the group the processor belongs to and p represents the position of the processor within the group. The basic topologies used (mesh, hypercube, expander) refer to the connections within each group. The optical transpose links,

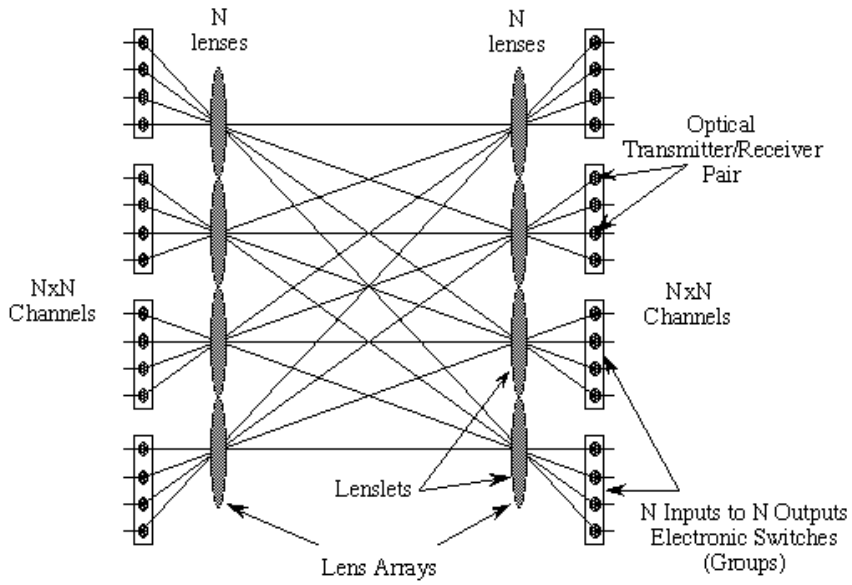

Figure 1. Optical Interconnects

which connect nodes (g, p) and (p, g), provide the only connections between nodes in different groups. We also make a few additional assumptions for carrying out the emulations. All links are assumed to be bi-directional. The mesh and hypercube networks are assumed to run in SIMD fashion; that is, each node is allowed to send along only one of its edges at any time, and this choice must be uniform for all processors. Since such a definition would not make sense for the expander, we assume the stronger property that each node can process a message per edge. If this assumption is made in general, the other two networks can be used in MIMD fashion. If the optical links run at the same speed as the electronic links, this MIMD simulation will be slowed by contention for the optical links, because several paths being simulated will need to use the same optical link. If the optical links can run d times faster for a degree d network, no additional slowdown will be incurred. Faster optical than electrical communication speeds are certainly supported by the current and projected technologies; the difference between the two speeds in an integrated system is more difficult to ascertain.

4. OTIS-Mesh

Consider each group of OTIS has having the very simple 2-D mesh topology: each PE is connected to the PEs which lie to the north, south, east, and west of it within the same group. Since each PE has degree at most four, and every connection is short, meshes are simple and can be implemented with a single level of electronic wiring. However, for the same reasons, the mesh topology is quite weak, having large diameter and (relatively) small bisection width.

The *OTIS-Mesh* architecture consists of N groups, each of which is an N-node (i.e., $\sqrt{N} \times \sqrt{N}$) 2D-mesh. The OTIS interconnections provide communication between groups, connecting processors (g, p) and (p, g) for all $1 \leq p, g \leq N$

Theorem 1 *OTIS-Mesh can simulate a 4-dimensional ($\sqrt{N} \times \sqrt{N} \times \sqrt{N} \times \sqrt{N}$) mesh with a slowdown of at most a factor of 3.*

Proof:

For each PE of the OTIS-Mesh, interpret its address (g, p) as (g_x, g_y, p_x, p_y) by dividing the bits representing the group and position into two equal pieces. With this interpretation, the mesh within the group g connects (g_x, g_y, p_x, p_y) to the four PEs $(g_x, g_y, p_x \pm 1, p_y)$ and $(g_x, g_y, p_x, p_y \pm 1)$, except at the boundaries.

Now, each of $g_x, g_y, p_x,$ and p_y will be a coordinate of the 4-D mesh address. Using the mesh connections within each group as described above, we can simulate the two last dimensions of the 4-D mesh. To simulate communication across the remaining two dimensions, we will need three steps. First, send the data across the optical transpose links of OTIS. The information initially stored in PE (g_x, g_y, p_x, p_y) is now stored in PE (p_x, p_y, g_x, g_y). Then, using the mesh connections, this data can be moved to PE $(p_x, p_y, g_x \pm 1, g_y)$ or $(p_x, p_y, g_x, g_y \pm 1)$, depending on the intended destination. Finally, use the optical transpose again, bringing the data to $(g_x \pm 1, g_y, p_x, p_y)$ or $(g_x, g_y \pm 1, p_x, p_y)$, which is precisely the desired connectivity.

■

This simulation allows more efficient solutions to problems which have faster algorithms on higher-dimensional meshes, like routing and sorting. However, it also can be viewed as a way to minimize wire lengths in large systems. By using a folded OTIS system, any two points on different chips can be connected with only wires across two chips. Using the same ideas as in the proof above, a signal is routed across the first chip to the location of the optical link to the destination chip. Then the signal is sent across the optical link, and routed across the destination chip to the desired location. With this technique, a long off-chip line across a PCB or MCM is replaced by an optical link plus on-chip wires to route the signal from source to transmitter and from receiver to destination. Similar on-chip connections would also be required in the all-electrical case to reach the I/O pads, however, so the additional overhead is minimal.

5. OTIS-Hypercube

The hypercube or n–cube is a versatile network for multiprocessor architectures. The hypercube architecture can simulate many other important topologies, such as meshes, meshes-of-trees, butterflies, and even PRAMs. Also many problems such as sorting and routing have efficient hypercube algorithms [11]. Several practical implementations of the hypercube are available as commercial products. However, it is difficult to construct larger–dimensional hypercubes using electronic technology. Since the degree of a node is $\log n$, and not constant, the number of wires leaving each chip or group of chips must grow as the size of the network increases.

The OTIS-Hypercube consists of small hypercubes linked by an interconnection pattern which is significantly sparser than that of a hypercube. Each node in an N^2-node OTIS-Hypercube has degree $(\log N) + 1$; there are $\log N$ connections to other nodes in its group as well as 1 optical link. An N^2-node hypercube would have degree $2 \log N$. In a manner similar to the shuffle-exchange graph, OTIS can be used to simulate the connectivity of a full hypercube. The OTIS-Hypercube network is also closely related to the hierarchical cubic network (HCN) proposed by Ghose and Desai [4]. The OTIS-Hypercube lacks the 'diameter' links of the HCN, but this does not affect its ability to simulate the hypercube.

Let $n = \log N$. Label the OTIS node (g, p) by $g_1 \ldots g_n p_1 \ldots p_n$, where $g_1 \ldots g_n$ is the binary representation of g and $p_1 \ldots p_n$ is the binary representation of p. We will now show how to connect (g, p) to every node which differs from it in exactly one bit position.

If the bit position in question lies in the second half (ie, the bits corresponding to the position within the group), then there is direct connection since all the processors $(g, *)$ form a hypercube group.

If the bit position lies in the first half (i.e., the bits of the group address), then we first perform an optical transpose. This will interchange the bits of the group and the bits of the position. Now, the bit to be changed lies in the second half, so by the same argument as before, the nodes to be connected are now linked by an electrical wire. Finally, another optical transpose restores the nodes to their original position.

More formally:

Theorem 2 *An N^2-node OTIS-Hypercube network can simulate an N^2-node hypercube with a slowdown factor of at most 3.*

Proof:

Label the OTIS nodes by $2n$-bit strings $x_1 \ldots x_{2n}$ as above. The hypercube edges we wish to simulate are of the form:

$$x_1 \ldots x_i \ldots x_{2n} \rightarrow x_1 \ldots \bar{x_i} \ldots x_{2n}$$

For $i > n$, these are the connections provided by the electrical hypercubes.

For $i \leq n$, we use the routing path:

1. From $x_1 \ldots x_i \ldots x_{2n}$
2. to $x_{n+1} \ldots x_{2n} x_1 \ldots x_i \ldots x_n$ via optical transpose
3. to $x_{n+1} \ldots x_{2n} x_1 \ldots \bar{x_i} \ldots x_n$ via hypercube edges
4. to $x_1 \ldots \bar{x_i} \ldots x_{2n}$ via optical transpose.

■

This factor of three slowdown in the proof is needed to ensure that the right 'parity' is preserved; that is, that the group and position coordinates have returned to their original roles. In some problems, this is not necessary, and the simulation can run correspondingly faster. In particular, the optical transpose is necessary only when the communication switches between the low $(1, \cdots, n)$ and the high $(n+1, \cdots, 2n)$ dimensional edges of the hypercube. Thus, problems in which this happens seldom, like routing, can be done even more efficiently that this simulation describes.

6. OTIS-Expander

In a similar fashion to the OTIS-Mesh and OTIS-Hypercube constructions, in this section we will show how to construction large randomly-wired graphs known as expanders in a hierarchical fashion. This will make the construction of large expanders more feasible; normally the random wiring of expanders does not scale as the size of the graph becomes large.

Unlike the mesh and the hypercube, expander graphs do not refer to an explicit set of connections. Rather, they refer to any graph which has the property that any for any set of nodes S with $|S| \leq \alpha N$, the neighborhood of S has size at least $c|S|$, for some constant $c > 1$. An expander graph G with N nodes is one with constant degree, d, which has the property just described. We will use the notation (N, α, c)-expander to denote such a graph. The constant-degree restriction is intended to capture some notion of efficiency; without it, the complete graph on graph on N nodes would be an $(N, \alpha, 1/\alpha)$-expander. While there are explicit constructions of graphs of this kind, the graphs with the strongest form of this property (i.e., large values of c) are constructed by choosing graphs at random. (For a simple proof which shows that a randomly chosen d-regular graph is likely to have this property, see Section 5.3 of [14].) For large enough values of N, one can show that a random choice of G is likely to have c close to d, for values of α satisfying $c\alpha < 1$.

The OTIS-Expander is constructed from N identical copies of a fixed (N, α, c) expander, which are connected to one another using the transpose connections of OTIS. With this definition, we can show:

Theorem 3 *Let G be an N^2 OTIS-Expander constructed from N copies of an (N, α, c) expander. Then G can simulate an $(N^2, \alpha^2, c/2)$ expander with a slowdown of a factor of two.*

Proof:

To show that a graph has expansion, we need to show that any sufficiently small set of nodes expands to a much larger set by following the edges of the graph. The intuition behind this theorem is that expansion will happen as long as the sets involved are not too large, and that the sets before and after the optical transpose cannot both be large.

Let S be a set of nodes of size at most $\alpha^2 N^2$. We will divide the nodes into two classes: those which begin in 'big' groups, (groups with more than αN elements of S) and those which begin in 'small' groups (groups with at most αN such elements). Since every node is in exactly one of these categories, one category contains at least half the nodes in S. The proof considers these two cases (depicted in Fig. 2) individually.

If the 'small' groups have more nodes, ignore all nodes in large groups. By doing so, we only decrease the size of the set of neighbors of S and underestimate the expansion of the graph, because the size of the neighborhood of S is a monotonic function of the nodes of S. Consider the nodes of S group by group, and let S_i denote the number of nodes of S in group i, not counting the nodes in large groups that we ignored. Because we only consider nodes in small groups, for all i, $|S_i|$ is at most αN. Therefore, the neighborhood of each S_i has size at least $c|S_i|$, and the neighborhood of S has size at least $c \sum |S_i|$. Since at least half of the nodes in S were in small groups, $\sum |S_i| \geq |S|/2$. This implies that the neighborhood of S has size at least $c|S|/2$, and thus the graph has expansion at least $c/2$.

If the 'large' groups have more nodes, ignore all nodes in small groups. There are at most αN large groups, because each group we did not ignore has at least αN nodes of S, and S has size at most $\alpha^2 N^2$. From each such node, follow the optical link. The OTIS connections provide only one link between each pair of groups, so each group can only receive at most one node from each other group. However, since before the transpose there were at most αN groups containing nodes, during the transpose each group receives at most one node from each non-empty group, or at most αN nodes. Then, we can make a similar argument to the one used in the small groups case above: each groups has at most αN nodes, so within each group, the neighborhood is larger by a factor of at least c. Since the number of nodes we did not ignore is at least $|S|/2$, the size of the neighborhood across all the groups has size at least $c|S|/2$, and the graph has expansion at least $c/2$.

■

7. Scalable Multibutterfly Construction

It may seem counterintuitive that random or near-random wiring could be helpful, there are several results which demonstrate that randomness or expander graphs are useful in many types of routing or sorting problems. For example, Valiant and Brebner [16] demonstrated that choosing random intermediate destinations prevent worst-case behavior in butterfly routing. Instead of using randomness on-line, results involving expanders utilize the random-like connections of the network to obtain good worst-case performance from deterministic routing algorithms. The AKS sorting network [1] used expander graphs to demonstrate that sorting could be done in parallel in $c \log N$ steps, for a sufficiently large value of c.

More practical work on using expanders for routing has centered on the *multibutterfly* network studied in [15] and [12]. In a butterfly network, the each bit of the destination address is used to divide the packets into two classes, and so each node has an up wire and a down wire to nodes in the next stage corresponding to these two possibilities. These connections are made in a regular fashion: at the ith stage, node $x_1 \ldots x_n$ is connected to the same node of the next stage as well as node to $x_1 \ldots \overline{x}_i \ldots x_n$. In a multibutterfly, the address bits are used to partition packets into two classes, and each node has wires to each class in the next stage. However, each node will have several such connections, and they will not form a regular pattern as in the butterfly. At level i, the nodes can be naturally divided into 2^i partitions based on the address bits followed to that point.

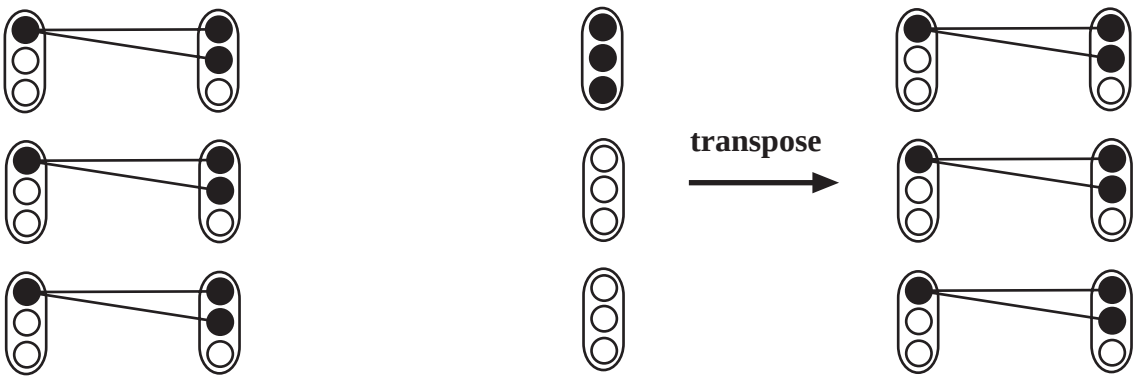

Figure 2. OTIS-Expander cases

In a multibutterfly, the bipartite graphs formed between each partition and either of the partitions connected to it in the next level are expander graphs. Bipartite graphs with this property are called *splitters* because of their applications to routing. This two-way expansion property guarantees that in order to cause a few nodes at level i to be blocked, many nodes at level $i+1$ would need to be blocked. By using this reasoning level-by-level, one can show that it is difficult to cause the inputs to become blocked, even in the presence of faults.

However, circuits which provide the connectivity of expander graphs are quite difficult to produce. While explicit constructions of expander graphs are known, they produce small values of c relative to d, which limits their usefulness. As mentioned above, random graphs are likely to have good expansion properties. However, if we attempt to partition a large random graph across many chips, we expect that nearly every edge of the graph will connect nodes on different chips. As the size of the required expander becomes larger, each chip requires more pins. The OTIS-Expander addresses this problem by producing an expander which can be partitioned into smaller pieces with manageable communications between the pieces.

Graphs of this type, referred to as hierarchical expanders, were first considered in [2]. There, the authors show how one can build large expander graphs without increasing the number of different wires required. Essentially, each chip or module has a small number of cables, and each cable is made 'thicker' as the number of nodes is increased. Here, however, the optical transpose provides wires from every group of nodes to every other group of nodes without the explosion in wiring complexity that the construction in [2] sought to avoid. At the same time, that construction relies on many different, independent random choices of wiring. For the systems of boards connected by cables that the authors envisioned, this can be accomplished simply by connecting cables to boards in a random fashion. However, at the finer scale of parallelism we envision, this would require the fabrication of many different chips, each with its own random wiring, greatly increasing the cost of such a system. However, the OTIS-Expander allows hierarchical expanders to be constructed using many copies of the same randomly-wired chip.

To construct a multi-butterfly using the OTIS-Expander, we use a similar construction. Here we describe the construction of the first stage, later stages use similar designs on smaller scales. As in the OTIS-Expander, each group is a copy of a single expander graph. However, each group will have N inputs, $N/2$ up outputs, and $N/2$ down outputs. The connections between the inputs and the outputs form a splitter; that is, any set S of at most αN inputs is connected to at least $c|S|$ up outputs and $c|S|$ down outputs. We refer to such a graph as an (N, α, c)-splitter. Obviously, this implies that $c\alpha$ must be smaller than $\frac{1}{2}$. The OTIS connections will be used to connect the input nodes of different groups, rather than to connect input nodes to output nodes. That is, the pth input of the gth group is connected to the gth input of the pth group via OTIS.

In order to show that this network can simulate a multibutterfly, it is necessary to show that any small set S of input nodes is connected to at least $c'|S|$ output nodes, for some constant $c' > 1$. The simulation of a stage of the multibutterfly will require three steps. In the first step, each input node communicates with every output node to which it is connected, and also to the node to which it is connected by the OTIS transpose link. In the second step, every input which received a message from the optical link in step one communicates that message to the outputs to which it is connected. Finally, in the third step, the packets at the output nodes are sent via OTIS connections to the next stage.

Theorem 4 *Let G be an OTIS-Expander constructed from N copies of an (N, α, c)-*splitter*. Then G can simulate an $(N^2, \alpha, c/2)$-splitter with a slowdown of a factor of three.*

Proof:

The same argument which showed that the OTIS-Expander has expansion will show that both the up and down edges have expansion, or equivalently, that the graph is a splitter. Since the up and down cases are symmetric and packets traveling in different directions never contend for the same edge, we consider only the packets traveling upwards. We then show that for any set S of inputs nodes which begin with packets traveling upwards. As in the OTIS-Expander proof, we examine two cases, depending on the initial distribution of packets among input nodes.

In the first case, the majority of these packets are in groups with less than αN other packets traveling upwards. Ignoring the groups with more than αN nodes, the number of nodes in each group expands by a factor of c during the first stage, as the inputs communicate with the outputs via the expander connections. Since at least half of the numbers were in small groups, this demonstrates expansion $c/2$.

In the second case, the majority of these packets are in groups with at least αN such packets. We ignore the nodes in groups with less than αN nodes. Each group will receive at most one packet from each other group while communicating across the OTIS transpose edges. Thus, at the beginning of stage two, each group will have at most αN packets that have neither been ignored or been transmitted to the outputs. This ensures that during stage two, each group will expand by a factor of c as it communicates with the outputs. Since we ignored at most half of the packets initially, this implies that graph has expansion at least $c/2$.

At this point, packets traveling upwards have gone through some expander edge and (possibly) OTIS and have now been sent to some of the first $N/2$ outputs of some groups. Likewise, packets traveling downwards have been sent to the last $N/2$ outputs of each group. Now, we use the OTIS connections once more, this time to connect the outputs of the different groups together rather than the inputs. Passing the data though these OTIS connections on the output nodes, packets traveling upwards reach the nodes in groups 1 through $N/2$, while packets traveling downwards finish in groups $N/2 + 1$ through N.

■

At the end of the first stage, the nodes traveling up are sent to one set of groups, and the nodes traveling down are sent to a different set of groups. To construct a complete multibutterfly, the same construction would then be applied separately to each of these sets of groups, continuing in a recursive fashion until each set contains only one group. At this point, no communication between different chips is necessary, and a single-chip multibutterfly can be used to perform the routing in the final stages.

8. Bit-Parallel Crossbar

In the applications of OTIS considered thus far, we have only discussed implementing networks which correspond to connected graphs; that is, networks in which it is possible for any node to communicate with any other node, possibly by routing the messages through intermediate destinations. However, there are applications in which this is not necessary. If each processor wishes to communicate an N–bit word, it is not necessary for the first pin on any processor to be able to reach the second pin of any other processor. For this application, we will have N processors, each of which begins with an N bit word and the address of another processor. The goal will be to route the data so that the first bit of the word destined for processor q arrives at the first pin of q, the second bit at the second pin, and so on. We will assume that the desired routing is a permutation to avoid discussing the effects of blocking; the same techniques will apply to the blocking case.

In this section, we will show how this routing can be facilitated by the use of transpose connections. The *Bit-Parallel Crossbar* will consist of two planes, each with N groups of N optical I/O pins. The first plane corresponds to the N I/O pins of each of the N chips (refered to as processors to distinguish them from the single-bit PEs considered earlier) to be connected. There are no connections between any of the pins in this plane. Each of the N groups in the second plane is an identical copy of a graph (possibly with more than N nodes) capable of realizing all permutations of its N inputs. In practice, a chip with crossbar connectivity would likely be used. We will assume that these chips are self-routing; that is, each node must start with the destination address to which it should route its information.

Theorem 5 *Let G be the topology of the routing chips, and let T_G be the number of bit–steps G requires to route an arbitrary permutation, including the time necessary to load the message bit and destination address for each input. Then the Bit Parallel Crossbar constructed from G can route any permutation of the input words in $T_G + 1$ bit–steps.*

Proof:

Initially, the OTIS node (i, j) of the first plane holds the value of the jth bit of the ith processor's word. The destination of the word held by the ith processor is $\pi(i)$. The goal is then, for all j, to route the value at node (i, j) to node $(\pi(i), j)$.

In the first step, each OTIS node (i, j) of the first plane sends its bit across the optical link, followed by the bits needed to specify the destination $\pi(i)$ (note that these steps are included in T_G). In the process, the jth group of the second plane receives the jth bit of each of the words, as well as the information needed to specify the routing π. By assumption, the routing chips are capable of performing this

routing, along with the loading of values and addresses, in T_G steps. At the end of this routing, the bit which began in processor i, position j is now located in OTIS node $(j, \pi(i))$ of the second plane. Another optical transpose moves this to node $(\pi(i), j)$ of the first plane, as desired, causing one additional step of overhead.

■

Here we have described a packet-switched mode of operation; each processor sends address information along with one word, the word is routed, and arrives at its destination. However, the same technique could be used to establish a circuit-switched path. After the first bit is sent as described above, no new address bits are sent, and all subsequent message bits follow the same path as the first one.

9. Conclusion

The Optical Transpose Interconnection System provides an efficient means of providing a particular set global interconnections using optical communications. The use of free-space optics allows efficient, high-density, high-speed long distance communication. Using these interconnections, small groups with only local connections can simulate networks with powerful global connectivity. The OTIS-Mesh connects groups with 2-D mesh topology into a 4-D mesh. This technique can also be used to reduce wire lengths in large systems. By connecting groups with hypercube topologies with OTIS, one produces the OTIS-Hypercube, which is capable of simulating a hypercube connecting all nodes in all groups with fewer wires. Using this result, many popular networks like meshes, meshes of trees, and butterflies can be simulated. Finally, this technique allows large expander graphs to be constructed in a scalable fashion by connecting many copies of a small expander into the OTIS-Expander network. The same techniques used to construct these scalable expanders can be used to construct large splitter graphs, for use in multibutterfly routing networks. Finally, this approach can be used to construct crossbar switches for routing entire words using bit-serial crossbar chips.

This research is supported by ARPA under grant F30602-93-C-0173 administered by Rome Laboratory.

References

[1] M. Ajtai, J. Komlos, and E. Szemeredi. Sorting in $c \log n$ parallel steps. *Combinatorica*, 3, 1983.

[2] E. Brewer, F. T. Chong, and F. T. Leighton. Scalable expanders: exploiting hierarchical random wiring. In *Proceedings of the 26th Annual ACM Symposium on the Theory of Computing*, Montreal, Quebec, May 1994.

[3] M. Feldman, S. Esener, C. Guest, and S. Lee. Comparison between electrical and free-space optical interconnects based on power and speed considerations. *Applied Optics*, 27(9), May 1988.

[4] K. Ghose and K. R. Desai. Hierarchical cubic networks. *IEEE Transactions on Parallel and Distributed Systems*, 6(4), April 1995.

[5] W. Hendrick, O. Kibar, P. Marchand, C. Fan, D. V. Blerkom, F. McCormick, I. Cokgor, M. Hansen, and S. Esener. Modeling and optimization of the optical transpose interconnection system. In *Optoelectronic Technology Center, Program Review*, Cornell University, September 1995.

[6] S. Hinterlong. High performance SEED-based optical computing systems. In *1995 ARPA MTO Program Review*, Big Sky, Montana, July 1995.

[7] H. Hinton. *An Introduction to photonic switching fabrics*. Plenum Press Ed., 1993.

[8] F. Kiamilev, P. Marchand, A. Krishnamoorthy, S. Esener, and S. Lee. Performance comparison between optoelectronic and VLSI multistage interconnection networks. *Journal of Lightwave Technology*, 9(12), December 1991.

[9] O. Kibar, P. Marchand, and S. Esener. High-speed CMOS switch designs for free-space optoelectronic MINs. submitted to *IEEE transactions on VLSI*.

[10] A. Krishnamoorthy, P. Marchand, F. Kiamilev, and S. Esener. Grain-size considerations for optoelectronic multistage interconnection networks. *Applied Optics*, 31(26), September 1992.

[11] F. T. Leighton. *Introduction to Parallel Algorithms and Architectures*. Morgan Kaufmann Publishers, 1992.

[12] F. T. Leighton and B. Maggs. Expanders might be practical: Fast algorithms for routing around faults on multibutterflies and randomly-wired splitter networks. *IEEE Transactions on Computers*, 41(5):1–10, May 1992.

[13] G. Marsden, P. Marchand, P. Harvey, and S. Esener. Optical transpose interconnection system architectures. *Optics Letters*, 18(13):1083–1085, 1 May 1993.

[14] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[15] E. Upfal. An $o(\log n)$ deterministic packet routing scheme. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing*, pages 241–250, Seattle, WA, May 1989.

[16] L. Valiant and G. Brebner. Universal schemes for parallel communication. In *Proceedings of the 13th Annual ACM Symposium on Theory of Computation*, pages 263–277, Milwaukee, WI, May 1981.