Small Depth Beam-Steered Optical Interconnect

Miles Murdocca, Hans Raj Nahata, and Yu Zhou

Department of Computer Science, Rutgers University, Hill Center, New Brunswick, NJ, 08903
murdocca@cs.rutgers.edu (Murdocca); hans@george.rutgers.edu (Nahata);
yuzhou@remus.rutgers.edu (Zhou)

Abstract

An optical beam-steering mechanism augments an optical perfect shuffle in a proposed small depth interconnect. In an abstract view, a beam is quickly steered directly from any source to any destination. In practice, beam-steering is limited to only a small number of positions, and is limited to slow steering speeds. The limited number of steering positions is compensated by changing the topologies of the communication graphs to match the topology of the interconnect. Applications generally exhibit a degree of communication locality that offsets the time spent in reconfiguration, and so the slow steering speed is not a serious drawback of the approach as long as the beam-steering speed is matched with a period of stasis after reconfiguration.

1. Introduction

A variety of architectures span ordinary computers, parallel processors, and special purpose processors (such as high speed network switches). Regardless of the architecture, however, it is almost universally the case that communication patterns change slowly with respect to the bit rate. This property of *communication locality* remains true across widely varying applications, and motivates the development of a means for reconfiguring an architecture that supports high bit rates, but does not need to reconfigure as quickly as the bit rate.

In the work reported here, an optical beam-steering mechanism augments a two-dimensional (2D) optical perfect shuffle in a proposed small depth interconnect. In an ideal view, a beam is quickly steered directly from any source to any destination. In practice, beam-steering is limited to only a small number of arbitrarily located ports (*e.g.*, 4, 16, 27), and is limited to slow steering speeds (1 ms – 1 μs, depending on the technology). Although the resulting interconnect is permutation incomplete, the effect of permutation completeness is achieved by manipulating the communication graphs to match the form of the physical interconnect.

The focus of the work reported here is on an exploration of the effectiveness of the interconnection scheme. In Section 2, we give the background on the interconnection model. In Section 3, we present simulation results of mapping arbitrary communication graphs onto the interconnection model and discuss the results. In Section 4, we discuss variations and extensions of the work. Finally, in Section 5, we conclude that a small amount of beam steering has a significant impact on the capabilities of the interconnection scheme, but that nearest-neighbor beam-steering is not sufficient for the general case unless complexity is added to the interconnect.

2. The Interconnection Model

In prior work, a novel beam-steering mechanism makes use of liquid crystal imbibed diffraction gratings [1]. A grating is exposed onto a porous photopolymer, which is then filled with liquid crystals. During operation, the orientation of the crystals is electrically controlled so that a window is created (providing straight through connections), or so that the grating is enabled, splitting incoming beams into equally spaced positions at the outputs. In this approach the gratings are switched at slow speeds (~ 1 ms) with respect to the bit rate but are high in efficiency. Despite the slow switching rate of the gratings, several observations provide evidence that bit-rate reconfiguration is not necessary for an effective switching mechanism. For instance, for the emerging Asynchronous Transfer Mode (ATM) [2] method of network communication, a switch may only need to reconfigure between the arrival times of two cells (fixed length packets of 53 bytes), which is 424 bit periods. Thus, a switching mechanism that is 1/424 as fast as the bit rate may be sufficient for this type of communication [3].

Within a parallel processor, reconfiguration speed is more relaxed, although latency is much more important. Pinkston [4] shows that the frequency of repeating communication traffic patterns between processors and memory in a parallel processor can be as great as 10,000:1 as a result of *cluster locality*, which is a common property in many parallel programs, but low latency is still a crucial requirement of a high performance system.

Given these observations that communication patterns exhibit a significant period of stasis after reconfiguration, we are motivated to develop a low latency, high bit rate interconnect that does not need to reconfigure as quickly as the bit rate.

2.1 Static Interconnects. Traditional computer architectures are static in the sense that the connectivity among the internal components does not change, except when the hardware is physically rewired or circuit boards are changed. Even for permutation networks, the wiring topology is fixed. Despite the static nature of traditional computer architectures, widely varying behavior is achieved as a result of modifying the software that the architectures execute. The physical underlying connectivity among the components, however, remains unchanged.

There are a number of opportunities in computing and in communication for a mechanism that physically redirects signals, rather than simply logically redirecting them. Processing elements (PEs) of a parallel processor are typically interconnected through several stages of a multistage interconnection network (MIN) or are interconnected with a single stage of a MIN (like a perfect shuffle) that is traversed several times. In either case, a significant delay is introduced between the time that a signal enters the interconnect and the time that the signal reaches its destination. By physically redirecting communication channels, network depth can be reduced, which in turn reduces latency.

MINs are commonly used for switching fabrics in packet and circuit based networks. The depths of these networks are not very critical to performance when the number of lines serviced is small. The depth of a MIN, however, adds a significant delay as the number of serviced lines increases (the depth of a perfect shuffle based permutation MIN typically grows as $3\log_2 N$-1 for N lines [5]). As a practical constraint, CCITT recommends an average delay of 450 μs through an ATM switch, which includes buffer delays, and so N cannot be very large and still maintain a low switch latency.

Although reconfiguration for these applications can be implemented with a variety of static electronic approaches, a significant latency is introduced when components are interconnected with an electronic MIN. Further, a significant amount of time is required to reconfigure communication channels.

2.2 The Shifted Shuffle. One way that the latency problem can be approached is through the use of beam-steering, which is analogous to physically rewiring an electronic processor. This is a desirable goal, but it is difficult to arbitrarily steer a beam in two dimensions over a large number of positions. An alternative approach is to provide a dense array of free-space connections, and then selectively enable the connections that are needed. This is a practical approach for the laboratory, but is not generally practical when there are great variations in connectivity patterns, because the photonic signals are subdivided as many times as there are potential paths that a particular signal may take.

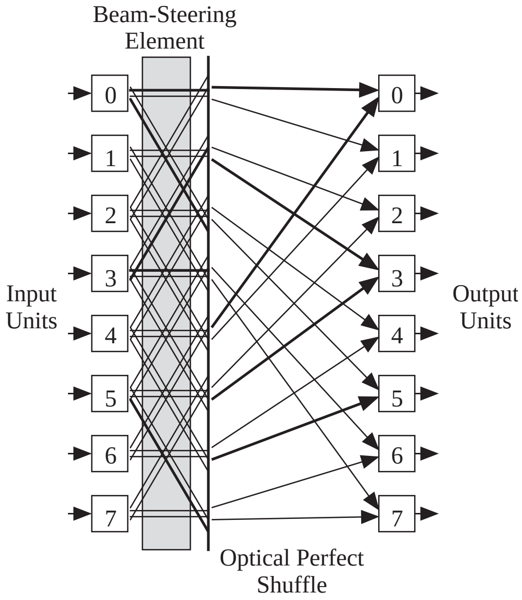
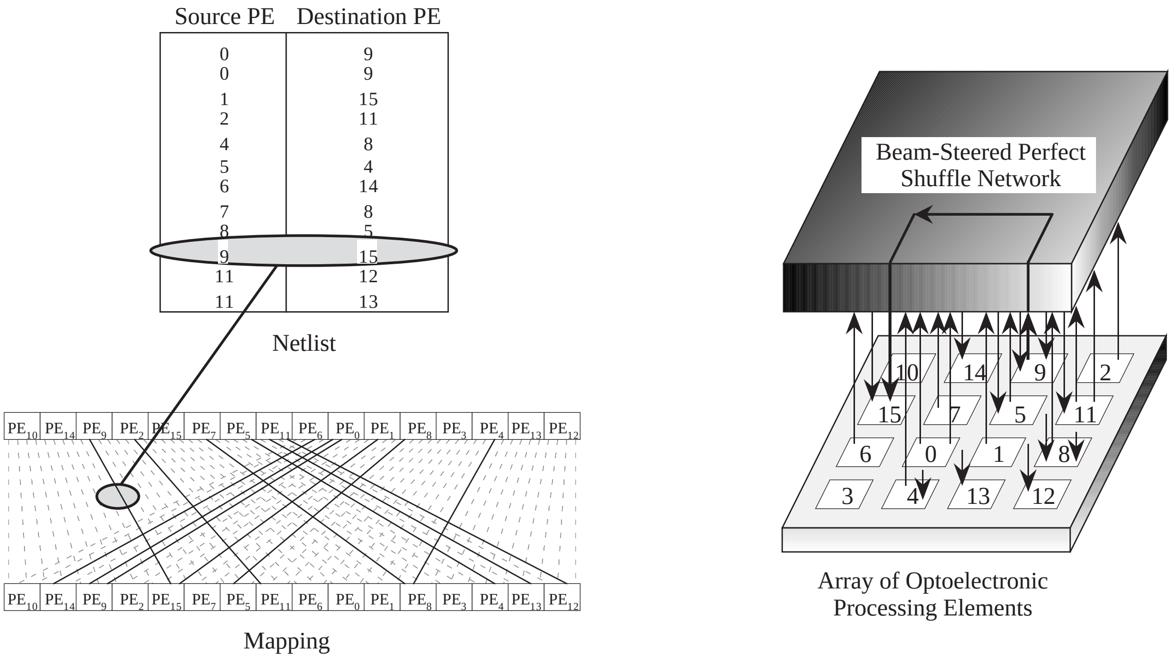


Figure 1: The shifted shuffle.

Network depth can be reduced for general permutation in a number of ways, such as by retrieving a fixed set of patterns stored in a volume hologram (as demonstrated at the University of California San Diego), in which the low depth is achieved through the application of the AKS switching algorithm. The network depth, however, is still much greater than a single stage.

A compromise approach that may work well in practice is to combine a good static free-space interconnect with a small amount of beam-steering [6]. Figure 1 illustrates the *shifted shuffle* interconnection pattern. An optical beam-steering stage [1] precedes a single stage of a one-dimensional (1D) optical perfect shuffle [7-9]. The perfect shuffle permutes the inputs in a fixed pattern, which provides a level of coarse permutation. The beam-steering stage handles fine adjustments separately for each channel. (The ordering of the beam-steering and perfect shuffle elements is justified in the next section.) A shift of only two positions to the left or right is indicated in Figure 1, but in practice, much larger shifts can be made, covering the same extent as the perfect shuffle.

This approach has a very low latency and scales well in terms of network depth, but does not support arbitrary permutations if only a single pass is made through the interconnect. While full permutation completeness may not be needed for the majority of computing and communication applications, some applications may need full permutation.



(a) (b)

Figure 2: (a) A netlist is mapped onto a 1D array of perfect shuffle connected PEs (without beam-steering.) (b) A target 2D application with a 2D beam-steering element.

For these applications multiple passes through the shifted shuffle may be needed, but an overall smaller switching fabric depth may still be realized through the use of beam-steering.

In the next section, we explore the effectiveness of this interconnection model by mapping a range of synthetic communication graphs onto the interconnect.

3. Simulation Results

In order to validate this model, there are several parameters that need to be considered, such as the number of input/output ports in the interconnect, the number of PEs, and the physical extent of the interconnect. In this section, we choose a range of parameters and investigate how much beam-steering is sufficient.

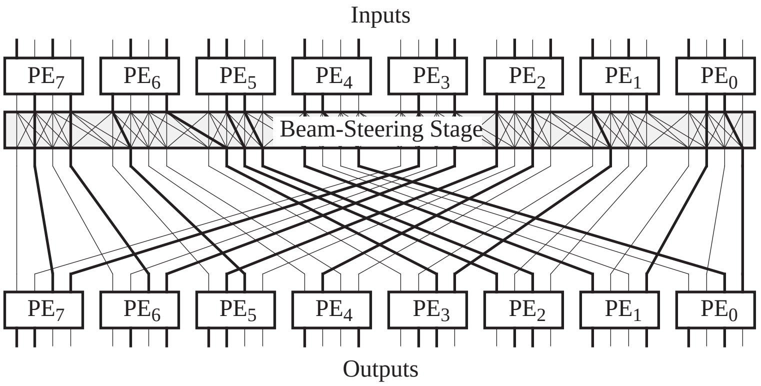
In Figure 2a, a netlist (a Source → Destination list of connections among PEs) is mapped onto a single stage of a 1D perfect shuffle. There is no beam-steering for this example. The PE indices in Figure 2a correspond to the PE indices shown in Figure 2b, which is for a topologically similar 2D optical perfect shuffle of the same size [10] that includes a 2D beam-steering element.

The overall problem is to find an arrangement of PEs that satisfies the communication needs of the netlist. For instance, an output port of PE 9 connects to an input port of PE 15 (highlighted) which is satisfied by the arrangement shown. Physically, the PEs have fixed positions, and it is the functions that are assigned to the PEs that are rearranged to satisfy the netlist.

A single stage of a perfect shuffle is not permutation complete, and a large number of netlists cannot be satisfied with this approach without adding complexity to the interconnect. Figure 3 shows a representative sample of data collected for randomly generated communication graphs. In this sample, eight PEs that each have four input ports and four output ports are interconnected with a 32-channel perfect shuffle. At most, four of the eight input and output ports are used (the *degree* of a PE is four). The number of connections that are used (the *link load*) varies from 2 to a maximum of 32 (the total width of the network). As shown in the table, the number of randomly generated samples that can be mapped onto the single-stage perfect shuffle markedly drops as the link load increases. That is, a single pass through a perfect shuffle does not provide sufficient connectivity for the majority of the randomly generated cases.

If we allow beam-steering to the right by just one port, then the number of successful mappings increases significantly. In the table, the 11.3% success rate for a link load of eight increases to 38.1% when the shuffle precedes the shift. The success rate improves to 69.9% when the shift precedes the shuffle. The success rate improves to 96.5% for a shift to the left or right by 1, followed by a shuffle. In general, the success rate improvement with beam-steering becomes more

No. PEs	Link load	Shuffle Only	Shuffle first then right by 1	Right by 1 then shuffle	Shuffle then right by 1 or 2	Right by 1 or 2 then shuffle	Shuffle then left or right by 1	Left or right by 1 then shuffle
8	2	100.0	100.0	100.0	100.0	100.0	100.0	100.0
8	3	97.6	100.0	100.0	100.0	100.0	100.0	100.0
8	4	87.9	99.1	99.3	99.3	99.3	99.2	100.0
8	5	69.8	95.5	97.9	97.8	97.9	96.9	99.7
8	6	50.0	86.5	95.5	95.0	95.6	91.4	99.9
8	7	26.9	65.1	89.1	85.9	89.1	77.3	98.6
8	**8**	**11.3**	**38.1**	**69.9**	**64.7**	**70.2**	**51.3**	**96.5**
8	9	3.5	18.6	49.0	41.0	50.2	29.4	89.8
8	10	0.8	7.1	27.8	21.8	30.2	12.6	79.4
8	11	0.4	2.6	11.8	9.6	13.6	5.1	57.8
8	12	0.0	1.1	5.3	4.5	7.8	2.5	36.0
8	13	0.0	0.1	1.7	1.5	2.7	0.6	17.6
8	14	0.0	0.0	1.0	0.3	1.9	0.0	9.0
8	15	0.0	0.2	0.4	0.4	0.6	0.2	4.1
8	16	0.0	0.0	0.1	0.1	0.3	0.0	1.8
8	17	0.0	0.0	0.0	0.0	0.0	0.0	0.8
8	18	0.0	0.0	0.0	0.0	0.0	0.0	0.4
8	19	0.0	0.0	0.0	0.0	0.0	0.0	0.2
8	20	0.0	0.0	0.0	0.0	0.0	0.0	0.1



Sample Graph:
PE degree = 4 (at most 4 of the 8 input/output ports are used for each PE)
PE width = 4 ports
No. PEs = 8
No. connections = 15
Beam-steering connectivity:

- Straight (no steering)
- Right one port
- Right two ports

Figure 3: Synthetically generated random graph mappings. Each entry represents 1000 samples, except when full enumeration occurs in less than 1000 samples. The highlighted entry is used as an example. Mapping success rates are all zeros for link loads greater than 20.

pronounced as the unshifted shuffle mappings grow progressively worse.

The success rate when the shift precedes the shuffle is always as good as or better than the corresponding case in which the shuffle precedes the shift. To see why this is the case, consider Figure 4, which compares the difference in effective connectivity when a shift by one position to the right is placed before and after a 1D shuffle. The number of different source → destination links is 9 when the shift succeeds the shuffle but is 11 when the shift precedes the shuffle. This is true in general, for small shifts.

4. Discussion

The data shown in Figure 3 support the notion that a small amount of beam-steering is effective, but might also be interpreted as supporting the use of a static interconnect. In this competing scenario, no beam-steering is used, and all of the potential connections are simultaneously made avail-

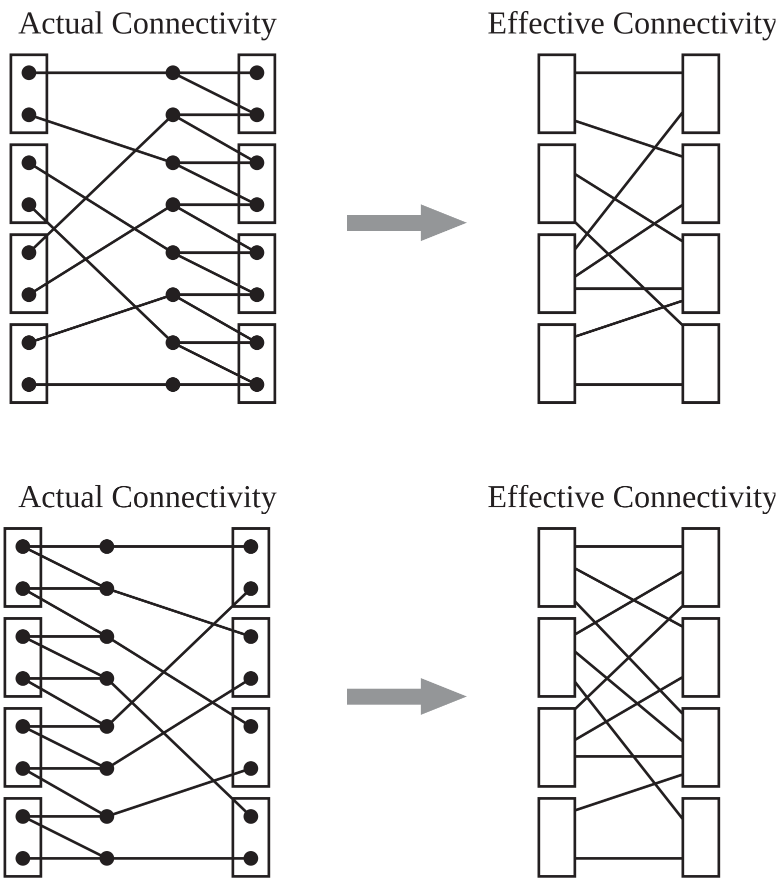


Figure 4: Comparison of preceding and succeeding positions of the beam-steering stage with respect to the shuffle.

able with only a subset used at any time. For a low speed system this may be appropriate, but for a high speed system, the static approach suffers from scalability constraints. These constraints fall into three categories: fan-out, physical distance, and logical distance.

If we allow beam-steering to connect any source to any destination within a 5×5 neighborhood, then the fan-out is 1 because the source only needs to drive one of the 25 destinations in the neighborhood. A competing static electronic or static optical approach would either need a greater fan-out (optical approach), or would suffer from the capacitive load associated with the unused 24 destinations (electronic approach).

The physical distance between the PEs may be large (100 μm or more), which can pose a significant performance limitation if an electronic interconnect is used. For a static optical approach, the capacitive load of a 100 μm connection is of no concern, but the fan-out issue arises again because the source signal must be subdivided as many times as there are potential destinations (25 for the above example).

Finally, although the data shown in Figure 3 are for small neighborhoods in a 1×32 array of input/output channels, much larger neighborhoods may need to be considered, in which case a static approach that implements all possible neighborhood connections may be prohibitively expensive. The logical extent of the beam-steering neighborhood can be increased to cover the array (depending on the physical dimensions of the array), without requiring a source to be steered to every site within the extent of the neighborhood. The potential for increasing the size of the neighborhood may be a crucial aspect in scaling the steered shuffle to large sizes.

Although the observations of the previous section indicate that a small amount of beam-steering is effective for a shallow interconnect, greater variability and wider angles are needed for the general case. The beam-steering approaches that look most promising [1] include elements that cleanly switch the beams among two or three locations. Cascades of two or three stages of these beam-steering elements should be practical and will allow for 1-of-4 to 1-of-27 selectivity in a 2D neighborhood that can potentially be very large, covering the extent of the array. Thus, the effective interconnection pattern will be more complex, which may improve the success rates. As a last resort, general per-

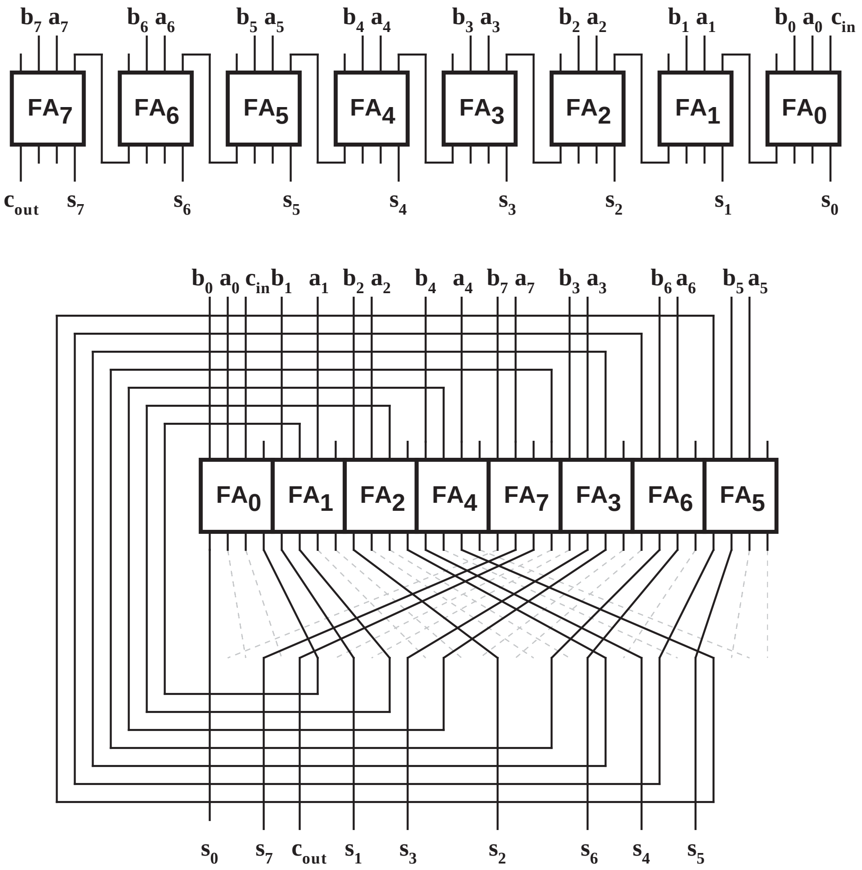


Figure 5: A single perfect shuffle stage interconnects an eight-bit ripple carry adder, without the use of beam-steering.

mutation techniques for shuffle-exchange networks can be applied, which guarantee a successful mapping in a multistage interconnection network [5].

When general permutation is not needed, however, small depth multistage interconnects may be appropriate which have smaller depth than for general permutation but have greater depth than the shifted shuffle. A number of networks exist for mapping arbitrary Boolean equations of N variables onto $2N$ stages of perfect shuffle connected logic gates [11]. Closed form solutions exist for certain other types of problems [12]. For example, a ripple-carry adder of any size can be mapped onto a single perfect shuffle stage. Figure 5 shows a solution for an eight-bit ripple carry adder. The fast Fourier transform (FFT) [13] also has a natural mapping onto a single perfect shuffle stage.

5. Conclusion

The following results are obtained from the simulated random graph mappings:

(1) The shifted shuffle interconnect provides sufficient connectivity for a number of randomly generated communication graphs in a small depth network, using one or two local shifts in conjunction with a 32-channel perfect shuffle;

(2) Local shifts should precede the perfect shuffle, rather than succeed it;

(3) Successful mappings drop off sharply when the link load is greater than 25%;

(4) A number of shifts in different directions augment the perfect shuffle more successfully than the same number of shifts applied in the same direction. For example, shifts to the left or right by 1 are better than shifts to the right by 1 or 2.

(5) Unsuccessful mappings can be made with additional passes through the interconnect, using ordinary permutation capabilities of shuffle exchange networks [5].

6. References

[1] Malcuit, M. S. and T. W. Stone, “Optically Switched Volume Holographic Elements,” *Optics Letters*, vol. 20, no. 11, pp. 1328-1330, (June 1, 1995).

[2] DePrycker, M., *Asynchronous Transfer Mode: Solution for Broadband ISDN*, 2/e, Ellis Horwood, New York, (1993).

[3] Eng, K. Y., M. A. Pashan, R. A. Spanke, M. J. Karol, and G. D. Martin, “A High-Performance Prototype 2.5 Gb/s ATM Switch

for Broadband Applications," Proceedings of *IEEE Globecom 1992*, **1**, pp. 111-117, (1992).

[4] Pinkston, T. M., "Design Considerations for Optical Interconnects in Scalable Parallel Computers," *Proceedings of the International Parallel Processing Symposium '94 – Massively Parallel Processing Using Optical Interconnects*, Cancun, IEEE Computer Society Press, (May 1994).

[5] Wu, C.-L. and T.-Y. Feng, "The Universality of the Shuffle-Exchange Network," *IEEE Trans. Comp.*, **C-30**, (5), 324, (1981).

[6] Murdocca, M., H. R. Nahata, and Y. Zhou, "Control Strategy for an Optically Reconfigurable Architecture," in *Optical Computing*, Institute of Physics Conference Series, no. 139, pp. 79-82.

[7] Brenner, K. H., and A. Huang, "Optical Implementations of the Perfect Shuffle Interconnection," *Appl. Opt.*, **27**, p. 135, (1988).

[8] E. J. Restall, B. Robertson, M. R. Taghizadeh, and A. C. Walker, "Two Dimensional Spatially Variant Optical Interconnects," *OSA Optical Computing Technical Digest Series*, **6**, p. 49, (1991).

[9] Stone, T., "Image Plane Perfect Shuffle Interconnects," presentation TuH3, *1993 Optical Society of America Annual Meeting*, Dallas, (October 2-7, 1994).

[10] Cheng, L., and A. A. Sawchuk, "Three-Dimensional Omega Networks for Optical Implementations," *Applied Optics*, **31**, pp. 5468-5479, (1992).

[11] Murdocca, M. J., *A Digital Design Methodology for Optical Computing*, The MIT Press, (1990).

[12] Murdocca, M., J. Battiato, D. Berger, R. Bussjager, P. Cook, and T. Stone, "Reconfigurable Architecture Based on Selective Enabling of Microlasers," accepted for publication in the *International Journal of Optical Computing and Processing*.

[13] Cooley, J. W. and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Math. Comp.*, **19**, 297, (1965).

7. Acknowledgment

This work was supported by the National Science Foundation on grant MIP 92-24707.