AFOSR-TR- 88-0735

# OPTICAL SYMBOLIC PROCESSOR FOR EXPERT SYSTEM EXECUTION
# ANNUAL TECHNICAL REPORT

June 1, 1987 to May 31, 1988

Sponsored by
Air Force Office of Scientific Research
and
Advanced Research Projects Agency (DOD)
ARPA Order No. 5794
Contract #F49620-86-C-0082





Prepared by

Aloke Guha
Julian Bristow
Subra Natarajan

Honeywell Corporate Systems Development Division
Honeywell Sensors and Signal Processing Laboratory

Submitted by: _____
Aloke Guha, Principal Investigator

Approved by: _____
Anis Husain, Section Head

Approved by: _____
Ben Hocker, Department Manager

SECURITY CLASSIFICATION OF THIS PAGE

ADA197668

# REPORT DOCUMENTATION PAGE

UNCLASSIFIED

| | |
|---|---|
| 1a. REPORT SECURITY CLASSIFICATION | 1b. RESTRICTIVE MARKINGS |
| 2a. SECURITY CLASSIFICATION AUTHORITY | 3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution unlimited. |
| 2b. DECLASSIFICATION/DOWNGRADING SCHEDULE | |
| 4. PERFORMING ORGANIZATION REPORT NUMBER(S) | 5. MONITORING ORGANIZATION REPORT NUMBER(S): AFOSR-TR- 88-0735 |
| 6a. NAME OF PERFORMING ORGANIZATION: Honeywell Inc. Physical Science Center; 6b. OFFICE SYMBOL (If applicable) | 7a. NAME OF MONITORING ORGANIZATION: AFOSR/NE |
| 6c. ADDRESS (City, State and ZIP Code): 10701 Lyndale Ave So, Bloomington, MN 55420 | 7b. ADDRESS (City, State and ZIP Code): Bolling AFB, DC 20332-6448 |
| 8a. NAME OF FUNDING/SPONSORING ORGANIZATION: AFOSR/NE; 8b. OFFICE SYMBOL (If applicable): NE | 9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: F49620-86-C-0082 |
| 8c. ADDRESS (City, State and ZIP Code): Bolling AFB, DC 20332-6448 | 10. SOURCE OF FUNDING NOS. |

| PROGRAM ELEMENT NO. | PROJECT NO. | TASK NO. | WORK UNIT NO. |
|---|---|---|---|
| 61102F | DARPA | | |

11. TITLE (Include Security Classification): Optical Symbolic Processor for Expert System Execution

12. PERSONAL AUTHOR(S): HUSAIN

| 13a. TYPE OF REPORT | 13b. TIME COVERED | 14. DATE OF REPORT (Yr., Mo., Day) | 15. PAGE COUNT |
|---|---|---|---|
| Annual | FROM 01/06/87 TO 31/05/88 | | |

16. SUPPLEMENTARY NOTATION

| 17. COSATI CODES | | | 18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) |
|---|---|---|---|
| FIELD | GROUP | SUB. GR. | |
| | | | |
| | | | |

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

In this year we began with a detailed performance evaluation of our optical architecture, SPARO, for combinator graph reduction. Since we had determined that the interconnection network was the bottleneck in the performance of the architecture, our focus was on the message throughput of the simple register-based network. We derived an accurate performance model, for the equivalent bidirectional ring network and found, both by analysis and simulation, that the net parallelism in the architecture was restricted by the low message traffic in the network. When messages exhibited no locality, the throughput for a 1024 processor network was limited to 8. With local messages, the maximum throughput for the same network was 27.

| 20. DISTRIBUTION/AVAILABILITY OF ABSTRACT | 21. ABSTRACT SECURITY CLASSIFICATION |
|---|---|
| UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS ☐ | UNCLASSIFIED |

| 22a. NAME OF RESPONSIBLE INDIVIDUAL | 22b. TELEPHONE NUMBER (Include Area Code) | 22c. OFFICE SYMBOL |
|---|---|---|
| GILES | (202) 767-4931 | NE |

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE.

SECURITY CLASSIFICATION OF THIS PAGE

## SUMMARY

The goal of this program is to develop a concept for an optical computer architecture for symbolic computing by defining a computational model of a high level language, examining the possible devices for the ultimate construction of a processor, and by defining required optical operations.

In this year we began with a detailed performance evaluation of our optical architecture, SPARO, for combinator graph reduction. Since we had determined that the interconnection network was the bottleneck in the performance of the architecture, our focus was on the message throughput of the simple register-based network. We derived an accurate performance model for the equivalent bidirectional ring network and found, both by analysis and simulation, that the net parallelism in the architecture was restricted by the low message traffic in the network. When messages exhibited no locality, the throughput for a 1024 processor network was limited to 8. With local messages, the maximum throughput for the same network was 27.

The poor performance of the simple ring network motivated us to examine other more elaborate but efficient interconnection network topologies. The alternatives considered were hypercubes, multistage interconnection networks (MINs), and single-stage shuffle-exchange networks (SENs) and replicated SENs. On the basis of analysis and extensive simulations, we found SENs, especially replicated SENs, to be the most feasible and promising. Recent investigations have indicated that SENs could be implemented efficiently in optics. Furthermore, we established that replicated SENs can provide a high throughput competitive with any other interconnection network.

While the shuffle connection of the SEN is feasible in optics using passive devices, a full-scale exchange switch which handles conflict resolution among competing messages is much more difficult. The functionalities required for the exchange switch and its controls were therefore analyzed. These functionalities were then assessed for optical implementation. A reasonable approach appeared to be to

construct the basic exchange switch, and then incrementally add the necessary functionalities. We found that while the basic switch and the representation of the message can be done with relative ease in optics using different information encoding techniques, the conflict resolution function is far too complex to be implemented optically. Even using brute-force techniques such as holographic look-up tables to implement combinational logic that underlies the exchange switch, a large network (1024 or more) would require exchange switches of prohibitive sizes. We conclude that optically controlled network exchange switches will be a reality only when optics technology promises basic switching logic to be competitive in size and speed with electronics.

In conjunction with our work on the optical exchange switch, we also evaluated the advantages and the relative feasibility of hybrid optical designs for the complete SEN. We evaluated electronic and hybrid SEN implementations in terms of complexity and performance. The hybrid design refers to the use of an electronic exchange switch in conjunction with an optical shuffle connection. The analysis of electronic SENs required designing the interface between the processors and the SEN, the smart exchange switch, and means of laying out the perfect shuffle within the board. We considered both GaAs and ECL technologies to determine the highest performance of an electronic SEN. Our results showed that when a large number (1024 or more) of specialized graph reduction SPARO processors, whose complexity and sizes were estimated on paper, are packed on a board for high speed parallel computing, there is a severe performance degradation due to the limited parallelism in transferring messages. Our focus was therefore directed to using optics for implementing a high-bandwidth and high-density SEN multiboard architectures.

The requirement of high density I/O for boards is not unique to SENs. This was based on our analysis of the general I/O requirements of parallel architectures that are implemented as multiboard systems using other interconnection networks such as crossbars and hypercubes. A formal analysis of board I/O requirements in transferring messages in parallel between PEs was conducted to compare SENs, hypercubes, and crossbars. A particular example, the Connection Machine, was also examined to obtain a real-world reference. It was clear that as larger levels of parallelism are employed, existing electrical connection technology would be hard pressed to provide the high degree of connectivity and parallelism necessary for

high performance. Our results revealed that if a large number of boards are used in implementing the architecture, then a single-stage SEN is the best choice if the network load is not very high.

The exchange switch analysis as well as the earlier performance analysis of SPARO motivated us to focus our energies in determining the optical techniques and devices that would be the most promising in providing the high bandwidth and high density interconnections. We therefore compared five optical approaches, fibers, polymer waveguides, planar holograms, volume holograms, and microoptics. Our assessment reveals that polymer waveguides and, possibly, planar holograms offer the most promise.

Our task for the next year will be to finalize on the scheme for using waveguides on the backplane to implement a high-density backplane shuffle connection. Concurrently, we plan to design methods for providing multiple optical sources and receivers that will be compatible with electronics. We have currently an outline of a project that can be used to demonstrate an optical perfect shuffle network for a small-scale parallel processor consisting of two boards.

## 1 INTRODUCTION

The thrust of the OSPESE program underwent a major change in the past year. This was driven by the conclusions we reached early in the beginning of the second year of the program. After the SPARO (Symbolic Processing Architecture in Optics) design had been completed, we embarked on a rigorous performance evaluation of both the serial and the parallel throughput possible. Our analyses revealed that while SPARO was novel in mapping the combinator graph reduction model onto a two-dimensional optical plane (where the evaluation could be done optically), the performance of the architecture was poor due to the inordinate sizes of the present-day devices required for solving non-trivial problems A more significant discovery was that the relatively simple shift register based ring network employed in SPARO was a severe bottleneck in achieving high throughput. Since fine-grained processing, as employed in the architecture of SPARO, depends critically on the performance of the interconnection network, this discovery motivated us to examine

optical interconnection more closely. The emphasis of the program thus shifted from optical processing to optical interconnection networks.

The search for the ideal optical interconnection networks has led to the examination of different known networks such as hypercubes, shuffle-exchanges and the crossbar. Rigorous studies of the requirements of large parallel processing systems reveal that parallelism of connectivity is the key problem in implementing interconnection networks both in electronics and optics. Based on our analysis of the computing requirements and feasibility in optics, the single stage shuffle-exchange network (SEN) appears to be the interconnection network of choice.

The problem we have therefore examined is to determine what optical techniques are most appropriate to implement a high-density and high-bandwidth SEN.

This report is organized as follows. The next section presents the detailed analysis of interconnection networks for SPARO. Section 3 examines the key issues in implementing purely optical SENs. Section 4 outlines the design of hybrid SENs, those implemented in a mix of optics and electronics. For comparison, an electronic SEN design is discussed. The interconnection requirements of highly parallel systems are analyzed in Section 5, while Section 6 examines optical techniques that could be used to meet those requirements. Section 7 presents a short description of how a small multiboard SEN could be demonstrated. Finally, Section 8 entails the conclusions.

## 2 ANALYSIS OF INTERCONNECTION NETWORKS FOR SPARO

Our initial analysis focuses on the performance of the ring network that was proposed earlier. The performance metrics, both throughput and waiting time of messages are derived analytically, and compared with simulated results. The other candidate interconnection network, the shuffle exchange network, which has been found suitable for optical implementation, was also analyzed. The results presented for the candidate networks are mostly from simulations. We show how shuffle-exchange networks, especially replicated shuffle exchange networks, can provide significant improvement in the message throughput and thus guarantee a greatly improved performance for SPARO.

The performance of the proposed optical architecture, SPARO (Symbolic Processing Architecture in Optics), was shown to be dominated by the performance of the interconnection network. In order to determine the expected throughput of the messages in the SPARO network, and thus the rate of reductions, it is necessary to analyze the network used. A shift register based ring network was proposed in the original architecture. It was expected that the systolic nature of the ring network would accomplish a high throughput of messages, and therefore provide fast execution of combinator graphs by using a high degree of parallelism. Our analytical model reveals, however, that even a bidirectional ring network of large sizes cannot provide significant parallelism. Thus, while the simple ring network is amenable for optical implementation, its performance is not acceptable. This motivated us to examine alternative candidates for the network. Among the networks found suitable for packet switching, the single-stage SEN and the binary hypercube are promising candidates. We have therefore examined and analyzed the SEN, and compared its performance against that of the hypercube and also the popular multistage interconnection network (MIN), the delta network. The reason for using the delta network is that it is considered a standard high-performance interconnection topology. Unfortunately, the implementation of a MIN is much more complex than the SEN. Our intent in the comparisons was to show that despite the simpler implementation of the SEN in optics, (or in optoelectronics) the performance of the SEN is quite competitive with a MIN.

We have provided the detailed analytical modeling for the bidirectional ring network. We derive the expressions for both the throughput and the waiting time. Two cases are considered in our analysis of the ring network performance: i) the case where messages for any processor node are equally probable, i.e., no locality is assumed, and ii) locality of messages are assumed whereby the probability of the message to a destination node varies inversely as the distance from the source.

Our analysis is based on the work presented earlier by Lawrie and Padua [5]. It is assumed that the conflicts in the SEN are resolved by giving priority to the message closest to its destination. We show, from the results of our simulations, that for a packet switching network, even a modest message generation can throttle the network. This underscores the inappropriateness of the loading factor used by Lawrie and Padua to characterize the networks. To alleviate the problem of

increased waiting times and low throughput, we studied buffered SENs. Contrary to naive expectations, the introduction of buffers does not improve performance. We present arguments as to why such behavior is observed, since analytical modeling of buffered SENs with priority strategy for resolving conflicts is extremely difficult.

We also present our simulation results on replicated SENs. We show how replication of SENs can dramatically improve throughput. Based on our results, we show what order replication would be recommended, given performance and cost constraints. The other alternative to replication is to use an enlarged network or a network that is about four times as large as the number of processors to be connected. The choice of replication or enlarging the network would be determined by the relative difficulty of merging multiple networks (in the case of replication) and the maximum size of the network that can be implemented (in the case of enlarging the network).

Finally, we present a summary on the performance analyses of hypercubes and delta networks for comparison with the SENs. The comparison is based on our simulated network results as well as the theoretical work done by other researchers.

## 2.1 Performance of ring networks

Before presenting the model used to represent the ring network, we present below the assumptions made in our analysis. We also preface our assumptions with a brief description of the network.

### 2.1 1 Principle of operation

The register-based network originally designed for SPARO, purely by serendipity, looks quite similar to that proposed earlier for the ZMOB parallel processor [3] intended for image processing applications. The SPARO network is composed of at least 1024 registers (the size being determined by the size of problems that the architecture is targeted for) connected in a conveyor belt fashion. Each stage or register is associated at any time with a single processor node as in Figure 2.1. There are thus 1024 processor nodes. Each processor node can receive or send a message by accepting a message from or loading into its associated register in the network. Messages are delivered by the network by shifting the registers in a

conveyor belt fashion. Since each message has a destination address, the message reaches its destination when the processor node address matches that of the message. Unlike the ZMOB network, the SPARO network is bidirectional. The network is assumed to recognize the direction which results in a shorter delivery path for a message. The analysis of unidirectional and bidirectional network is only trivially different.

In terms of operation, the following three-phase sequence is followed in SPARO:

i) Each processor in the network examines if its message output buffer is empty, that is, if the previous message has been delivered. If the output buffer is full, the processor cannot load its new message into the buffer and therefore enters a wait state. Otherwise, the processor will load the output buffer with its message and continue processing.

ii) The ring network shifts and the ring register associated with each processor examines if it has a message to deliver to the processor. This is done by comparing the destination address of the message to that of the processor. If the addresses match, the message is delivered to the input buffer of the processor.

iii) The processor checks if the associated ring register is full, that is, the message in the register is meant for another processor. If the register is empty, then the message in the output buffer is loaded into the ring register. Otherwise, the processor waits to offload its message in the next network cycle.

Since in phase (iii) the processor buffer cannot be emptied, the processor cannot generate more messages. This allows the ring to proceed uninterrupted at its full speed, and also ensures that no messages are lost. This assumption also implies that the message generation rate is influenced by the loading of the network. (The assumption reflects, especially in the case of fine-grained processing, that a processor operates on simple sequential tasks and cannot proceed until the previously sent message has been delivered and a response message has been received.) We now examine the analytical model of the ring network in brief.

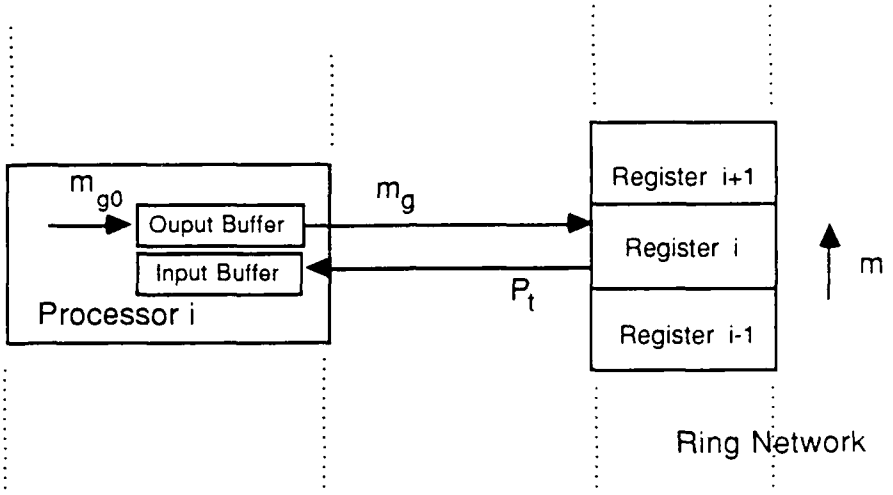



**Figure 2.1 Processor and Ring Network Interface**

2.1.2 Analytical model of the ring network

Figure 2.1 shows the representation of an individual stage of the network and its communication with the processors. We define below the following parameters that are used to define and compute the throughput of the network.

$N = 2n$: total number of nodes in the network
*p(i)*: probability that the destination of a message is i shifts away from source
*mg0*: rate of message generation at each node under no loading restrictions
*mg*: effective rate of message generation at each node
*m*: total rate at which messages arrive at a node via the network
*Pt*: probability of termination of a message arriving at a node

In the first part of our analysis, we consider the case that messages in the network are equally likely to be delivered to any node in the network. This contrasts with the case that the messages exhibit locality, or that the probability of accessing remote destinations is lower than those nearer to the source node. In the equiprobable case, the probability of generating a message with a destination that is i shifts away is independent of i. This probability is $2/N$ for a bidirectional ring ($1/N$ for a unidirectional ring) where $N = 2n$ is the ring size.

We can calculate the rate *m* at which messages are traversing the network. If *mg* be the effective probability of message generation in each processor, and *Pt* the

probability that a message has terminated or reached its destination, then in the steady state, the rate of generation of messages, $mg$, will be equal to the rate of consumption $m\ Pt$. Then, in a bidirectional ring,

$$m = mg\ /\ 2\ Pt$$

In the case of the unidirectional ring, $m$ is twice that of the bidirectional ring.

Note that the above expression involves the effective message generation rate $mg$ and not the actual message generation rate which we denote by $mg0$. This modification reflects the fact that the effective message generation rate depends on the load on the network. As traffic on the network, indicated by $m$, increases, the message generation rate will decrease. Thus, $mg \leq mg0$. The effective message generation rate can be computed by knowing when the buffer in the processor is full. If $q1$ and $q0$ be the probability that the buffer is full and empty, respectively, then we can find q1 using the relation

$$q1 = a01\ q0 + a11\ q1$$

where $a01$ and $a11$ are the probabilities of transition from $q0$ to $q1$ and vice versa. $a01$ is thus the message generation rate when the network register is occupied, or

$$a01 = mg0\ m(1 - Pt).$$

$a11$ is the probability that a non-terminating message arrives at the processor node or

$$a11 = m\ (1 - Pt).$$

Since q0 = 1 - q1, we can show $q0$ and $q1$ to be

$$q0 = [1 - m(1 - Pt)]/\ [1 - m(1 - Pt)(1 - mg0)]$$

$$q1 = m\ mg0\ (1 - Pt)/[1 - m(1 - Pt)(1 - mg0)]$$

Using the expression for $m$ previously derived, we find that $m$ is the solution to the following equation.

$$m^2 [2\ Pt\ (1 - Pt\ )(1 - mg0\ )] - m[\ 2\ Pt - mg0\ (1 - Pt\ )] + \ mg0 = 0$$

In the case of the unidirectional ring, the corresponding equation for $m$ can be obtained by removing all occurrences of 2 from the above equation.

It can be shown that one of the roots of the above quadratic equation for m is not viable. The legitimate value of m is found to be

$$m = a - \sqrt{b}/c$$

where $a$, $b$ and $c$ are defined to be:

$$a = 2Pt + mg0(1 - Pt)$$
$$b = [2Pt - mg0(1 - Pt)]^2 + 8Pt(1 - Pt)mg0^2$$
$$c = 4Pt(1 - Pt)(1 - mg0)$$

To calculate the throughput, we need to determine $Pt$, the probability of termination of any message. $Pt$ in turn can be computed if the distribution of messages in steady state is known. By steady state distribution we mean the distribution probability of messages with different destination distances, from 1 to $n$ ($N$ for the unidirectional case).

To find th termination probability $Pt$, we first derive the distribution probability for messages with random destinations. The method used for deriving the message probabilities is similar to the one used by Abraham and Padmanabhan [2]. Note from Figure 2.2 that there are two possible message sources (processor nodes) in a bidirectional ring that are $i$ shifts away when $i < n$, but only one when $i = n$. Suppose two shifts have taken place in the network. The distribution of messages can be derived as follows. The number of messages that require $(i - 1)$, $i \leq n$, shifts to reach its destination is 2 + 2. (The first term corresponds to the number of messages generated in the second shift that require $i - 1$ shifts, while the second term corresponds to those that need the same number of shifts but were generated in the first shift.) Note that the 2 appearing in each term, except for the destination n away from the source, is due to the fact that we are considering bidirectional networks.

Since the maximum number of shifts required for the ring network is n, the distribution of messages with different required shifts reaches steady state after n shifts. The number of messages requiring $i - 1$ shifts is $2(n - i) + 1$.

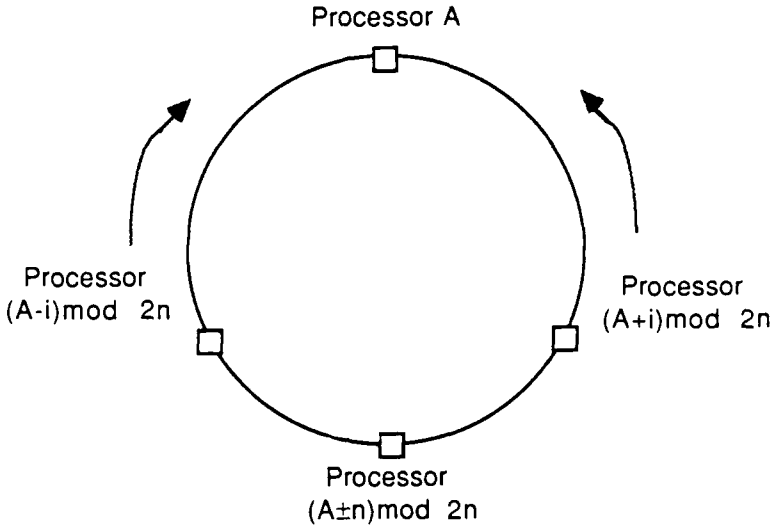



**Figure 2.2 Message Sources for Bidirectional Ring**

To derive the normalized probability, we need to find A where

$A = \Sigma [2(n - i) + 1] = n^2$ where $i$ varies from 1 to n

Thus, the probability that a message requires $i - 1$ shifts to reach its destination:

$p(i - 1) = [2(n - i) + 1] / n^2$

The termination probability $Pt$ is the probability that the destination of the message is 0 shifts away. This can be found by simply setting $i = 1$ in the expression for $p(i - 1)$, which gives $Pt = (2n - 1) / n^2$ . The termination probability for the unidirectional case can be shown to be $1/n$.

When messages exhibit locality, the same analysis can be carried out, except that the probability of messages requiring $i$ shifts decreases as $i$ increases. In case of a harmonic distribution of messages, the probability of a message requiring $i$ shifts is inversely proportional to $i$ .

The procedure to derive the expression for $p(i - 1)$ is exactly similar to that used in the earlier case, except that messages requiring different number of shifts are assigned different probabilities. Thus, $p(i - 1)$ can be written as:

$p(i - 1) = \Sigma (2/j - 1/n) / A$ where $j$ varies from $i$ to $n$.

where A can be shown to be:

$A = \Sigma \Sigma (1/j - 1)$ where $j$ varies from $i$ to $n$ while $i$ varies from $1$ to $n$

Setting $i = 1$ in $p(i - 1)$, we obtain $Pt$

$Pt = \Sigma (2/j - 1/n) / A$ where j varies from 1 to n

Neither the numerator and denominator can be expressed in closed form.

2.1.3 Throughput of ring networks

The throughput is defined as the average number of messages delivered at the end of each cycle or shift of the ring. Since $m$ is the rate at which messages arrive at a node via the network, the number of messages delivered at a node is $m\ Pt$. Since there are $N$ nodes in the network, the total throughput, denoted by $T$, can be expressed as follows.

For bidirectional ring $T = 2\ m\ Pt\ N = 8\ m\ (N - 1)/N$.
For unidirectional ring $T = m\ N\ Pt = 2\ m$

In the case of locality, the throughput expression for the bidirectional ring cannot be expressed in closed form. It can be seen that the throughput for no locality asymptotically levels off to $2$ and $8$ for the unidirectional and bidirectional rings, respectively. Thus, the throughput of unidirectional rings can be quadrupled at only twice the cost.

Figures 2.3 and 2.4 graphically depict the throughput for bidirectional rings when messages have random and local destinations. Figure 2.3 shows the near-exponential increase in the total delay time (theoretical derivations are not included

here) which is composed of the waiting time in the buffer and the transit time over the network. As can be seen, the analytical results agree closely with the simulated results. It is of interest to note that when messages exhibit locality, the throughput reaches as much as 27. This is more than three times the throughput of rings with no local messages.

Figure 2.3 Ring Network Throughput for Random Message Distribution

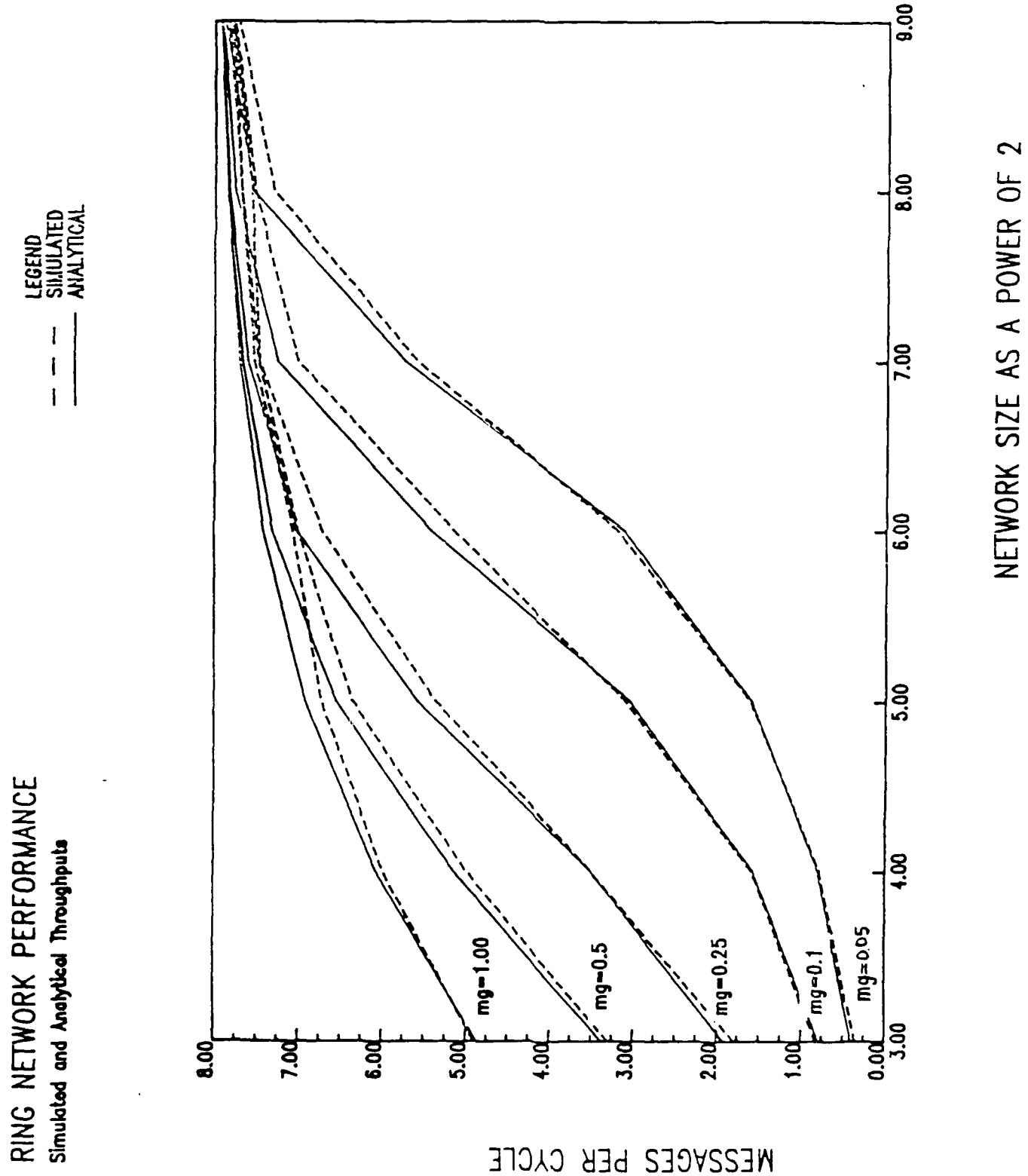
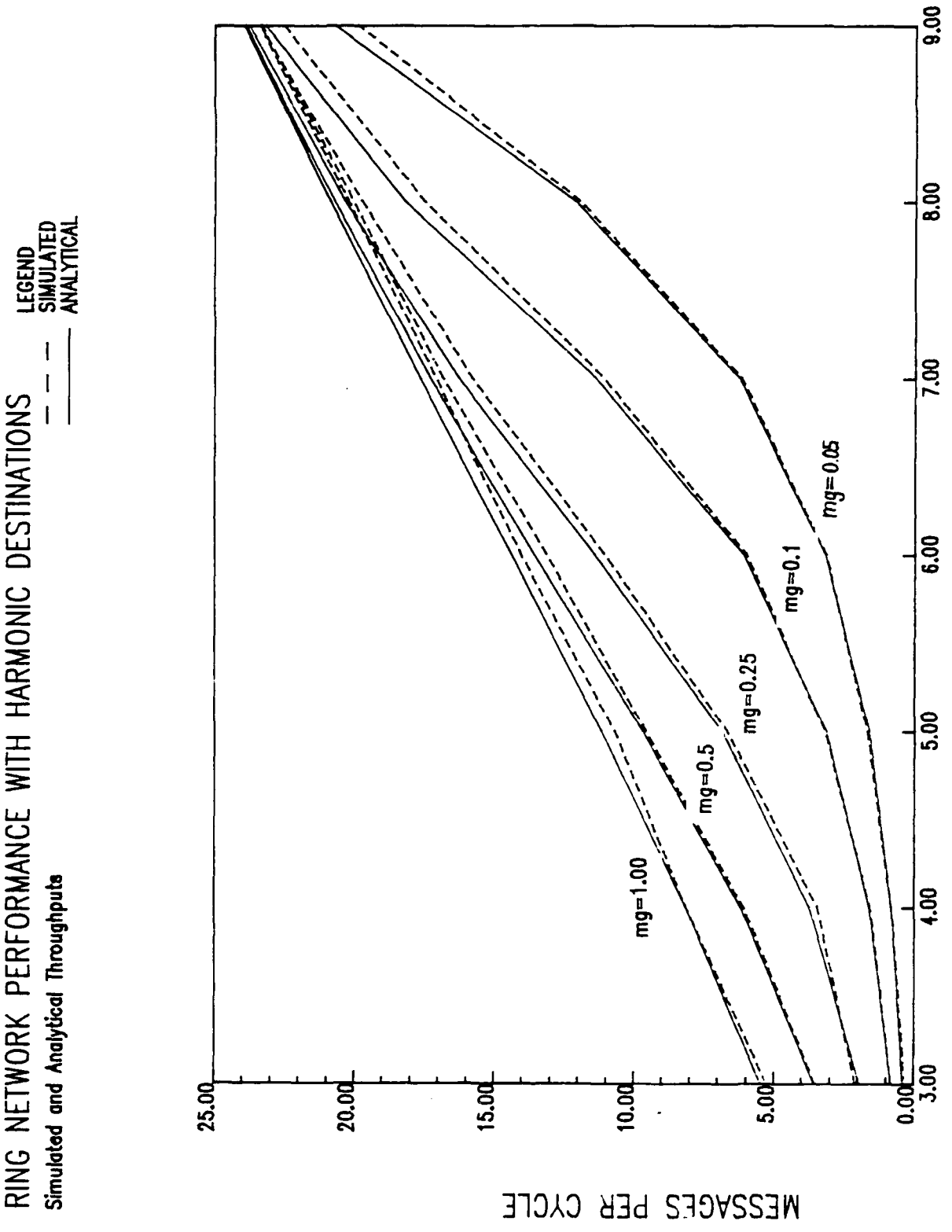


Figure 2.4 Ring Network Throughput for Harmonic Message Distribution



### 2.1.4 Conclusions

We have presented an analytical model to evaluate the throughput of ring interconnection networks in message-passing environments. Although the model is relatively simple, it effectively shows the serious limitations of the register-based ring networks. Clearly, from Figures 2.3. and 2.4, for processors using message passing for communications, the ring network cannot provide an acceptable throughput for more than about 16 processors.

We next evaluate other network topologies as possible candidates interconnecting the processors in SPARO.

## 2.2 Performance of single-stage and replicated shuffle-exchange networks

The first alternative topology that we examined in detail is the shuffle-exchange network. We also examined the potential of employing replicated shuffle-exchange networks which have been used in electronic network designs to improve the performance of the single-stage network. Although analytical models for predicting the performance of these networks have been studied, the question of what the desired level of replication should be, has been left unanswered. We therefore examine, from an architectural perspective, which of the following possible networks is desirable:

i) a single shuffle-exchange network (SEN),
ii) a full multistage interconnection network consisting of $\log_2 N$ stages when N processor nodes are to be interconnected, or
iii) replicated SENs where the degree of replication is between 1 and $\log_2 N$ .

### 2.2.1 The shuffle-exchange network

We will initially consider the single stage SEN. The operation of the SEN that we are considering is described in detail by Lang in [6]. A one-stage SEN contains $N = 2^n$ registers or buffers, indexed from 0 to N - 1, when N processors are connected to the network. Figure 2.5 shows the SEN for connecting 8 processors or network modules. To deliver messages from the processors, the network operates by cycles.

In each cycle, a message and its destination tag (binary address of the destination processor) passes through an exchange element and then undergoes a shuffle permutation. If the destination processor is reached, the message is delivered to the processor, or else it is injected back into the network. It takes at most n periods for a message to reach its destination.

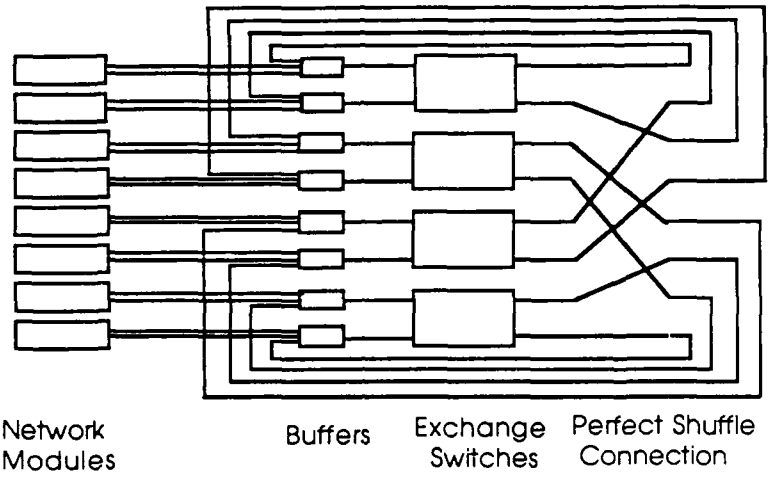



**Figure 2.5 Single-stage SEN for 8 PEs**

The shuffle stage of the SEN is used to realize a perfect shuffle among the N elements. The perfect shuffle can be described by the following relation:

$$S(i) = (2i + \lceil 2i/N \rceil) \bmod N$$

where S(i) denotes the destination of the message from the ith processor due the shuffle permutation and $\lceil\ \rceil$ represent the lower integer ceiling of the enclosed expression. Figure 2.5 can be used to verify the shuffle stage for N = 8. The second part of the SEN consists of N/2 exchange elements. The purpose of the exchange element is to exchange the destinations of messages arriving from two adjacent processors. The exchange is done based on the value of a control bit. Depending on the control bit, the exchange element will either exchange the position of the input messages or leave them unchanged. Instead of using separate control bits for routing a message through the SEN, one can conveniently use the destination tag method [10] for setting the control of the exchange elements. In this method, if $b_n$ $b_{n-1}$ ...$b_1$ be the binary representation of the destination processor, then during the (i - 1)th period, bit $b_i$ is used to set the exchange element. Depending on the controlling

bit, the switch will either exchange, i.e., cross-connect the input and the output message packets or let them pass through undisturbed. Clearly, since each input message can provide the switch setting independently, there is a fifty-fifty chance of conflict when two messages arrive at an exchange element.

### 2.2.2 Operational model of the SEN and its analysis

A good operational model and its analysis has been presented in [5]. Here we will give a brief description of the model to motivate the study of replicated SENs.

In the normal operational mode, several messages will be circulating in the network. Both from an analysis as well as from an implementation viewpoint, it is easier to consider the synchronous operation of the network. Synchronous operation of the network implies that the exchange elements and the registers (that hold the messages to be injected to the network) are latched simultaneously. On the average, messages in a SEN can be delivered within 2n cycles. There is no upper bound on the number of cycles required since in each pass through the network a message may get blocked, due to possible conflicts arising at the exchange elements of the network. The exchange element resolves this conflict by allowing one message to go through to the proper destination. If messages cannot be dropped, then the message that loses in the conflict resolution is routed via a longer path, thus increasing the 'delay time' or the number of cycles required to deliver it. Two schemes for resolving conflicts are often used:

i) random selection, and
ii) closest first selection.

In the random selection scheme, as the name implies, the message chosen for routing to the proper destination is chosen randomly. The message that loses in the random selection starts afresh in the routing cycle. To represent the status of a message in the routing, a counter is associated with the message. The counter is initially set to one. If the message is successfully routed in one pass or period of the SEN, its counter is incremented. However, if the message loses during a conflict resolution, its counter is reset. Thus, a message reaches its destination when the counter value is n+1. In the prioritized case of the closest first selection, the conflict is resolved by selecting the message with the larger counter value (randomly, if

there is tie). Results from [5] show that, as expected, the message delay is smaller and the throughput is higher (for small loads) in the second case. Therefore, we have focussed our attention on SENs utilizing the prioritized conflict resolution scheme.

Previous analysis of the SEN considered the state transition of the counter to determine the probability that a message has been delivered (i.e., the probability that the message has a counter value of n+1). This would then yield both the throughput and the expected number of periods (delay) that a message stays in the network. Unfortunately, the authors could not find a closed form expressions for these metrics for the prioritized selection case. Instead, numerical solutions have been provided for different 'loads', where the load is defined as the fraction of active messages to the total number (N) of processors. Note that the load is distinct from the rate of message generation by the processors. The results showed that for loads below 0.25, the expected delay is $1.5\log_2 N$ periods. For larger loads, the expected delay rises while the throughput falls off, especially for large N.

Previous researchers [5] have mentioned briefly the use of replicated SENs for improved throughput when the load is high. A detailed study of the effectiveness of replicated networks has not been undertaken. We have chosen to examine the advantages of replicated SENs, in the operational mode described above, in greater detail.

### 2.2.3 Replicated Shuffle-Exchange Networks

Figure 2.6 illustrates what a replicated SEN (RSEN) looks like. If a k-replicated network is used, then k networks are connected in parallel. A message to be delivered can be routed to any available network. Similarly, at most k routed messages can arrive at the input of a processor. For each of the k networks, the effective number of processors generating messages is at most N/k. Thus, the expected load in each SEN in the k-replicated network is at most 1/k. In our study of RSENs, we have limited k to be less than or equal to n since any message takes at most n passes to be routed if no conflict occurs.

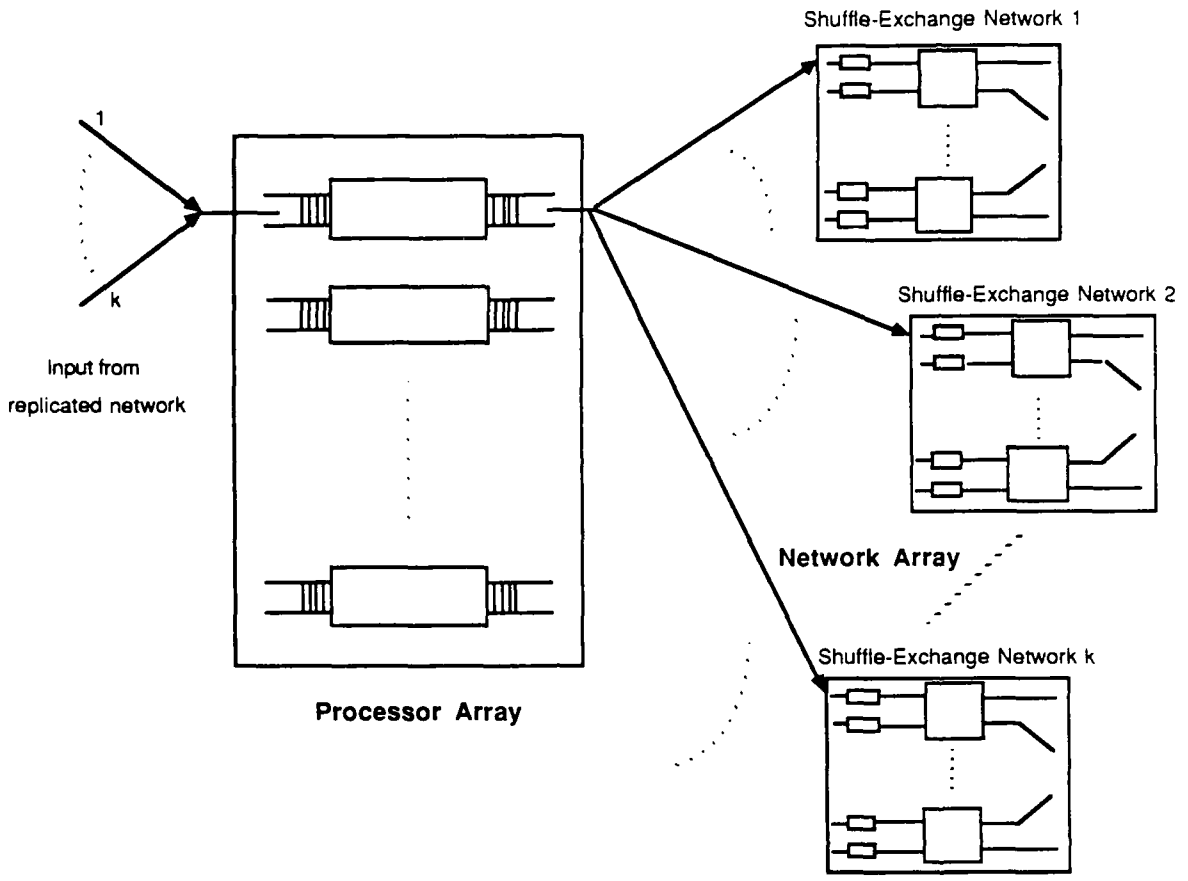



**Figure 2.6 Replicated SEN**

Our focus in the study of replicated networks is on the performance of k-replicated SENs for different values of k and for different message generation rates. The advantage in using RSENs, besides the increased throughputs, is the flexibility of varying the amount of replication. As Figure 2.8 shows the level of replication can be increased from 1 to 10 for a 1024 network for increased throughput. The cost and the desired performance will decide what level of replication is most suitable.

We next present the results on the performance of single SEN with different network sizes, and the performance of k-replicated networks for different k.

### 2.2.4 Performance of single and replicated SENs

Two different sets of results have been obtained by simulation. First, we have examined the effect of the size of the network on the throughput for a fixed message

generation rate. This has been done for a single SEN for the purposes of studying the effect of size on performance. Second, we have considered the effect of the degree of replication on the both the throughput and the delay time.

The performance of the single-stage SEN and the RSEN is given graphically in Figures 2.7 and 2.8. Figure 2.7 depicts the throughput and delay times for delivery of messages in a single-stage SEN of various sizes up to 1024. The message generation rate considered is only 0.25 since rates higher than this lead to a fully loaded network. A fully loaded network exhibits a very large number of conflicts resulting in very few deliveries per cycle. According to our simulations, a 1024 network has a throughput of only 40 when 256 processors can generate a message on average (message generation rate of 0.25). The RSEN performance (Figure 2.8) shows the throughput can be increased using replication. Note that the throughput shown in Figure 32.8 is the throughput per individual network. Thus, the total throughput for a 10-replicated network is greater than 380 when the message generation rate is 1.0. The dramatic increase in throughput in RSENs is due to the decreased loading in each network which results in fewer conflicts than in the single-stage SEN.

Another result obtained from simulations which influences the implementation is the effect of the delivery schedule of messages. In the normal delivery scheme, a message is delivered to its destination when the counter associated with the message reads $\log_2 N$. We had expected that delivering messages on the basis of the comparison of the destination address with the address of the node at the end of each cycle would be more efficient. However, simulations show there is no perceptible difference in the total delay time of messages or the throughput when the second delivery scheme is adopted. The lack of difference can be attributed to the effect of conflicts that erode any advantages expected in the scheme using address comparisons.

Figure 2.7 Performance of Single-stage SEN

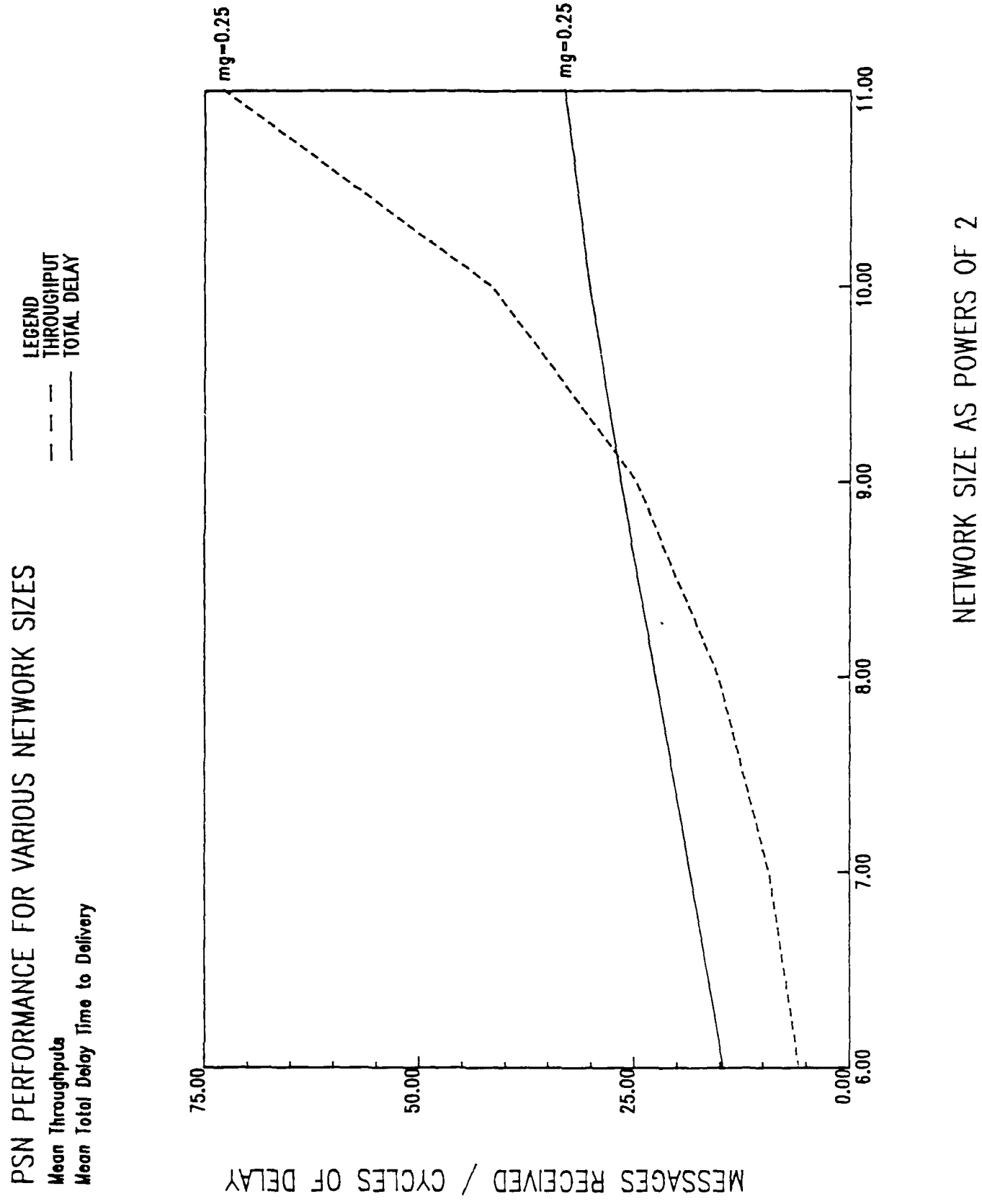
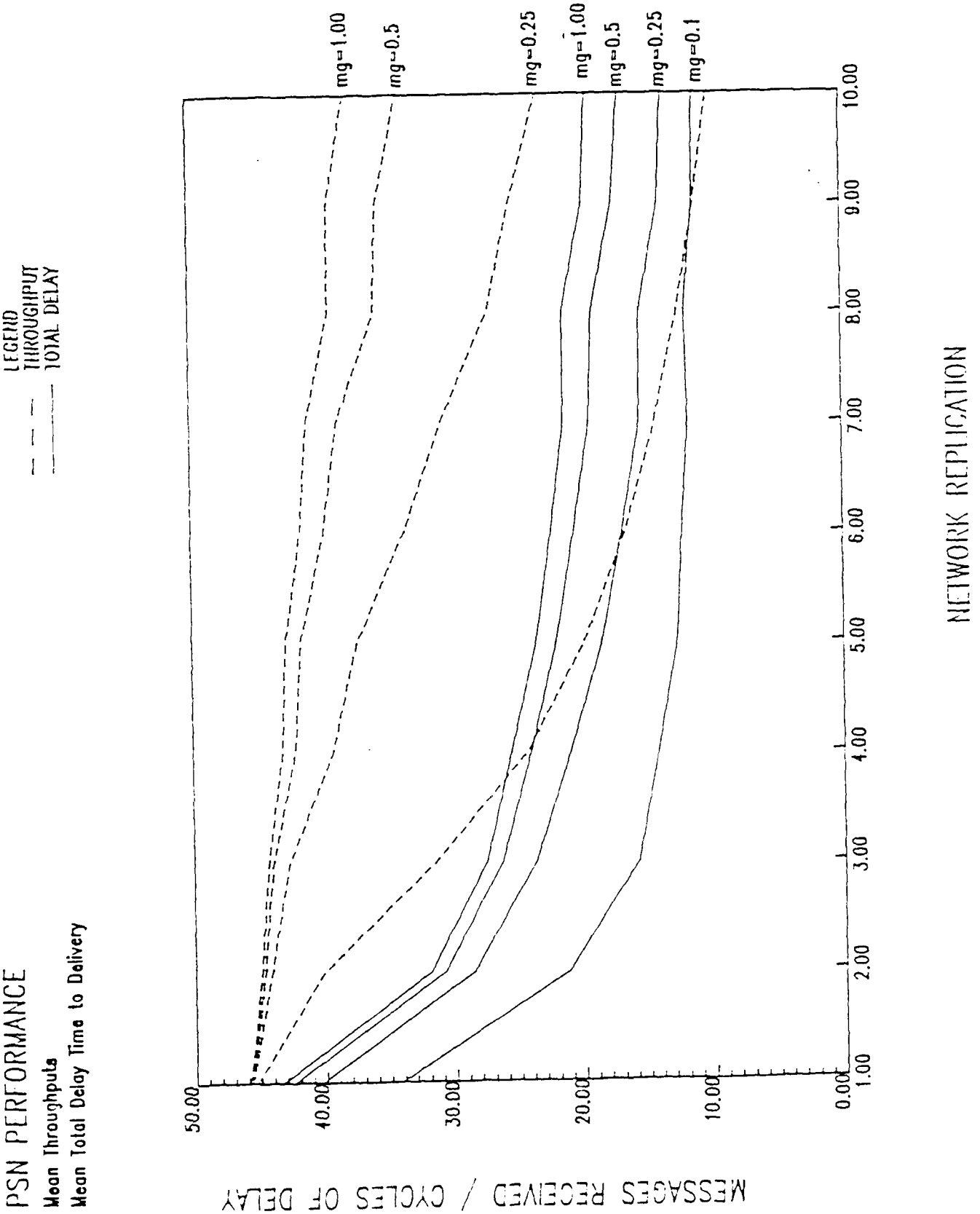


**Figure 2.8 Performance of RSEN**



We have also examined the use of buffers in single stage SENs to improve performance. We examined the effect of first-in-first-out buffers to queue arriving messages being injected in the network. Interestingly, the performance of buffered SENs, even for modest loads, degrades drastically as the number of buffers increases. This is because message delays increase almost exponentially while the throughput remains almost constant once the network is fully loaded. While the results may not be obvious at first, the results can be explained as follows. When buffers are used, processors can generate messages while buffers are not full even when the network routes messages through them. At some point, when all buffers are non-empty, the network essentially has a load of 1. From earlier results [5], we know that the performance degrades as the load on the network is increased above 0.25.

Another problem in buffered single stage SENs is the problem of routing messages into a processor whose buffer is full. One approach [9] is to provide handshaking capability between source and destination processors. However, in a single-stage SEN, a chain of up to n handshakes maybe required (as in the multistage SEN [9]). When a buffer is full and a message has to be accommodated in the input queue, some message in the buffer has to be removed to some other processor. Such a scheme will require more complex control and therefore a buffered SEN does appear attractive.

To examine the performance of SENs and RSENs more objectively, we examined other candidate networks under similar conditions of size and load.

## 2.3 Comparison of RSENs with other networks

The strength of RSENs can be judged best on the basis of its performance relative to other well-known networks. We have therefore examined a number of interconnection networks that are commonly used in electronic parallel processing architectures.

### 2.3.1 Comparisons with multistage interconnection networks

Multistage interconnection networks (MINs) are very popular in implementing large parallel processors. An example of such a network in a commercial machine is the BBN Butterfly. Figure 2.9 shows the topology for an 8 X 8 MIN. To compare the RSEN with the MIN, we have relied on the results on unbuffered delta networks (one class of MINs) given by Patel [8] and on the results on buffered delta networks by Dias and Jump [9]. Although both sets of figures, presented in Figure 2.9, are obtained from analytical models, Dias and Jump have verified their results by simulation.

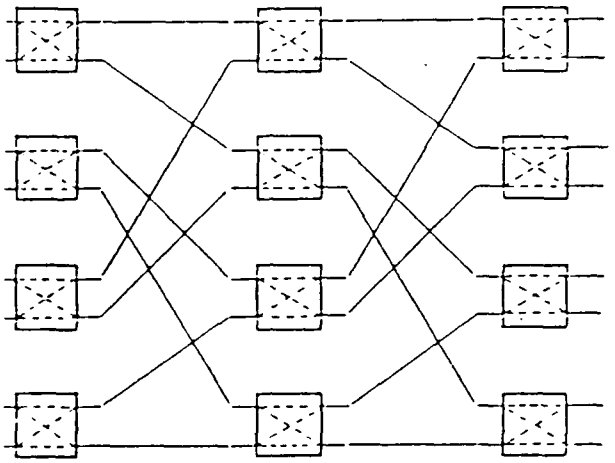


**Figure 2.9 8X8 Multistage Interconnection Network**

The results for a MIN show that for a 1024 10-stage MIN, the normalized throughput (that is, the ratio of the absolute throughput to the total number of processors) is about 0.2 for an unbuffered network and 0.39 for a network with a single buffer. These figures translate to a throughput of 205 in case of the unbuffered network and a throughput of approximately 400 for a network with a single buffer. This is comparable to the throughput of a RSEN (380) composed of 10 networks. The normalized delay time to deliver a message in the MIN is about 1.5 or 15 cycles for a 10-stage network. This also compares well with the 15 - 20 cycle range observed in the RSEN.

In comparing the RSEN with the MIN, note that while the MIN has multiple ($\log_2 N$) switching stages, the processors in a k-replicated RSEN must be able to accept up to k messages on its input port. However, in the case of a RSEN there is an added flexibility of using less than $\log_2 N$ shuffle-exchange networks if less than maximum throughput is acceptable.

2.3.2 Comparisons with the hypercube

To compare the performance of RSENs with another popular interconnection network of comparable size, we have examined the hypercube topology. The hypercube has recently become popular by making its appearance in two commercial machines the Connection Machine (CM) [11] and the Intel Hypercube. Figure 2.10 shows the topology of a hypercube of 4 dimensions. In our analyses, we have considered a 10-dimensional hypercube.

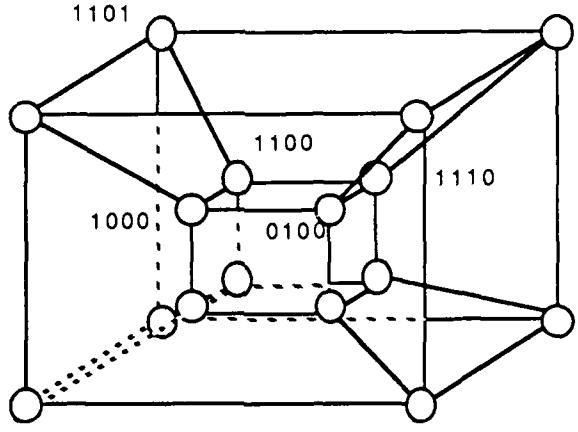



**Figure 2.10 Hypercube for 16 PEs**

As in the CM, each node in the hypercube is assumed to possess a routing buffer of length k ($0 \leq k \leq \log_2 N = n$). The routing in the hypercube is determined by forming the bit-by-bit EXOR of the source and destination addresses. The bit positions of the result indicate the dimensions along which routing takes place. The network operates synchronously with one dimension being active at a time. There is no set transfer sequence amongst the dimensions for a message to be routed.

When two processors are connected during some dimension cycle, each processor examines its own buffer to check for messages that need to be transferred. If none are found, a processor checks to determine if its buffer is full. If both processors have messages to transfer, a two-way transfer takes place. If only a single message packet needs to be transferred, the transfer is possible only if the buffer of the destination processor is not full. Two modes of operation are possible when considering the delivery of messages. In the first, both message generation and delivery are allowed in every dimension cycle. In the second, there is an upper

bound on the number of messages that can be generated and delivered in every n dimension cycles or one network cycle. The second mode of operation is followed in the CM.

In the first mode of operation, since messages can be delivered in each dimension cycle, one expects a smaller waiting and therefore a smaller delay time. While this should result in better performance, the control is expected to be more complex and the dimension cycle would be longer than in the CM mode. The network cycles in the two modes of operation of the hypercube therefore have different meanings. In the first mode where deliveries are allowed in each dimension cycle, an individual dimension cycle is longer to allow for message deliveries. In the CM mode of operation, the network cycle is composed of simpler dimension cycles during which messages can only be transferred. All deliveries take place at the end of the network cycle. While we focussed our attention on the first method, we simulated both modes of operation for comparison.

Figure 2.11 shows how the throughput varies with the size of the hypercube when a single buffer is used at a message generation rate of 0.25 per cycle. We have assumed the CM mode of operation. With a message generation rate of 0.25, the throughput is only about 25 for a hypercube of 10 dimensions. The poor throughput results from the overflow of the input buffer in each dimension cycle when more than one message have to be transferred across pairs of processors. Figure 2.12 examines the throughput for a 1024 hypercube as a function of the buffer size. The maximum throughput for such a network is about 190 when a buffer of size 10 is used.

Figure 2.11 Performance of Hypercube as function of network size

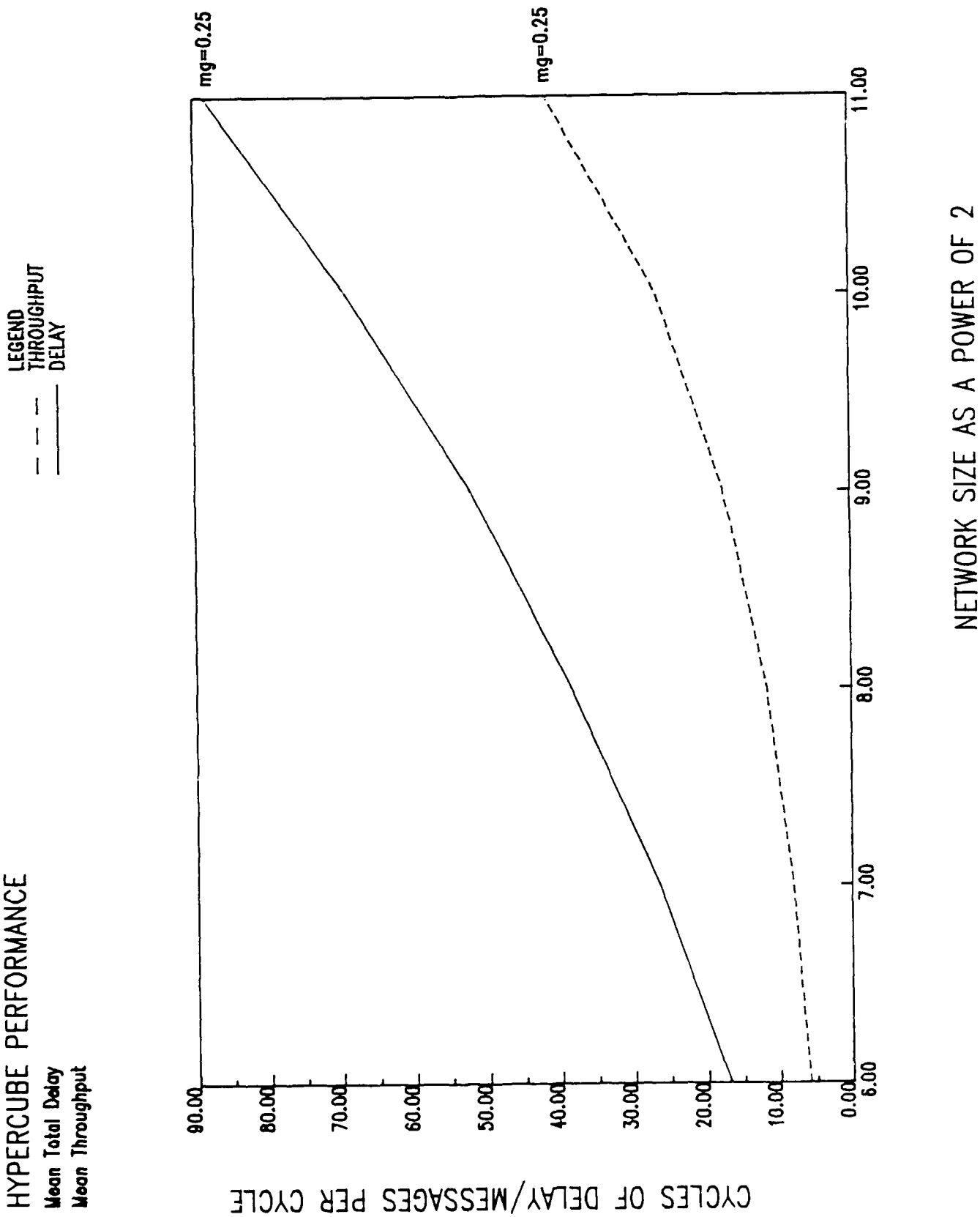
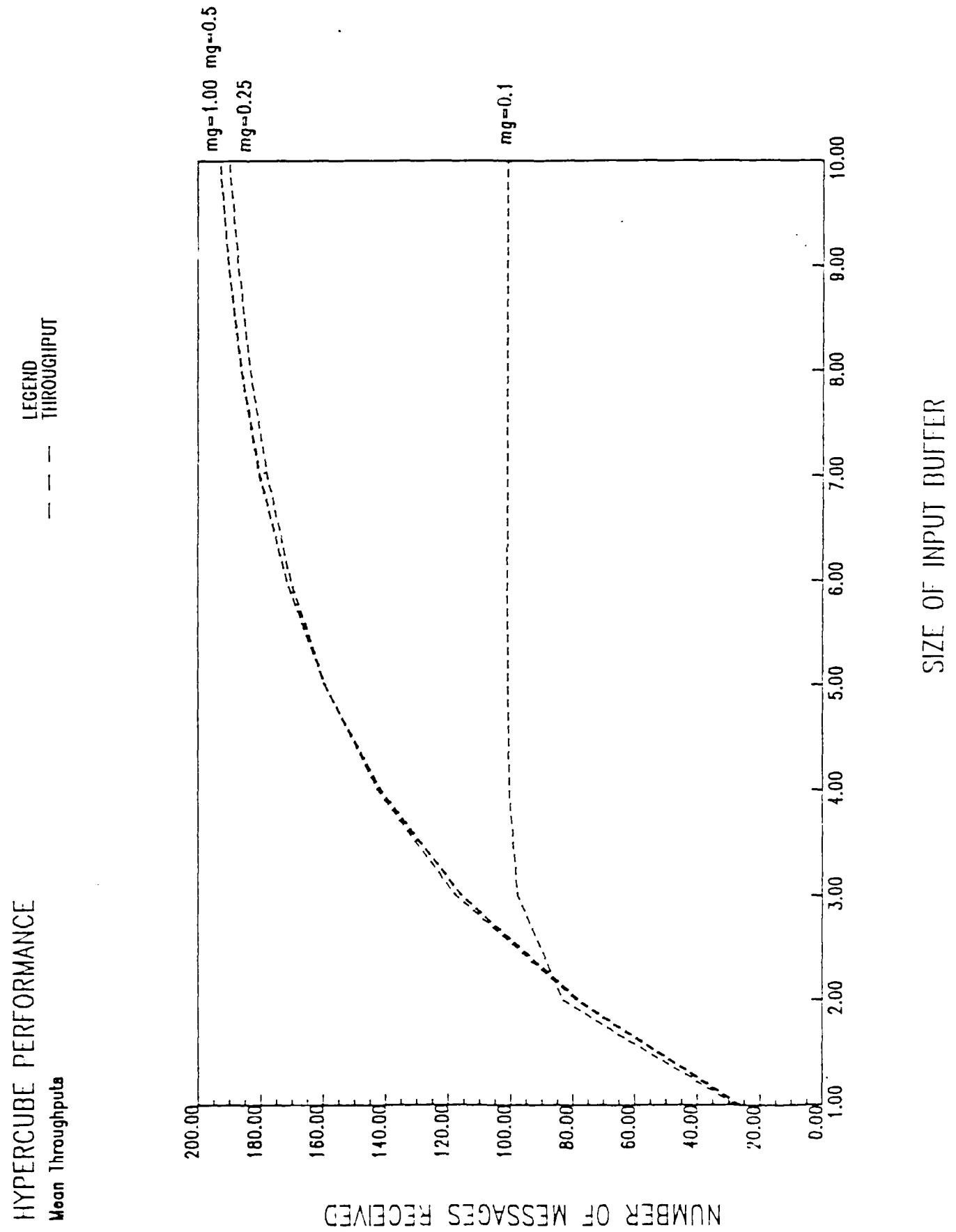


**Figure 2.12 Performance of Hypercube as function of Buffer Size**



In the CM mode of operation, where 1 message generation and 1 message delivery was allowed (that is, we allow 1 message to be generated, if the buffer is not full, and 1 delivery, if any message was generated, at the end of each network cycle), the throughput reaches a maximum value of about 1000 per network cycle. In either mode of operation of the hypercube (a maximum of 200 per dimension cycle), we believe that the RSEN is very competitive (385 per single cycle). It must be noted that it is difficult to make an exact quantitative comparison between the two networks since the RSEN cycle of operation is different from the dimension cycle in the hypercube. As pointed out earlier, the dimension cycle in the CM mode of operation is shorter since no message deliveries occur until a network cycle is completed. Actual implementation issues must be considered before accurate comparisons can be made.

In terms of optical implementation, the SEN appears more attractive than the hypercube which must use large buffers for each node in the processor. On such simple first order analysis, the SEN cycle would be expected to be shorter than that of the hypercube. Thus, given that the two networks are competitive on the basis of throughputs, the RSEN would appear to be a better candidate.

## 3 IMPLEMENTATION ISSUES IN OPTICAL SHUFFLE-EXCHANGE NETWORKS

Our analysis of interconnection networks, based mostly on performance, reveals the SEN to be competitive with other commonly used networks with low loads (<0.25). For larger loads, replicated SENs or RSENs are very competitive with other commonly used topologies. The SEN also appears more attractive than other networks because of recent work on the optical implementation of the perfect shuffle [12 - 15]. A number of different optical shuffle implementations have been proposed. Lohmann [12] and Midwinter [15] initially showed how the perfect shuffle can be implemented very effectively using passive optical elements such as lens and prism combinations or holograms. Eichmann and Li [14] have later shown an even more compact implementation in optics which reduces the total optical path length from Lohmann's approach by a factor of 6. Their results indicate that the channel spacing d and the spot size $\alpha$ are the limiting factors.

$d = D / \{N - 1\}, \quad \alpha \leq D / \{2(N - 1)\}$

where D is the aperture of the lens used (see Figure 3.3.). With a 50 X 50 sq. mm aperture lens, the optical perfect shuffle can handle as many as 40,000 light channels. The channel spacing and spot size is assumed to be 0.25 and 0.1 mm, respectively. While using bulk optics, as proposed in [14] may lead to larger-sized SENs, alternative holographic approaches and guided approaches may be used to accomplish the same shuffle permutation. We will examine these alternative approaches in the next section.

To make the design process of the optical SEN more tractable, we examined the proposed optical designs and also extracted the key requirements for the critical portion of the SEN, the exchange switch.

## 3.1 Existing optical shuffle-exchange designs

There has been increasing interest recently in the implementation of the optical perfect shuffle for sorting networks but very few efforts on the exchange switch implementation [13]. Here we outline the work that has been reported thus far.

### 3. 1. 1 Perfect shuffle

Lohmann [12] appears to be the first to present an optical perfect shuffle design. He proposes a setup using prisms and lenses (Figure 3.1). The input elements are divided into two halves, upper and lower, which are stretched in one direction to match the size of the original inputs. The stretched halves are then recombined by interlacing to achieve the perfect shuffle of the inputs. The outputs on recombination appear in reverse shuffle order but can easily be 'unreversed' by standard optical means. The total optical path length is two times the sum of the focal lengths, f1 and f2, of the lenses required to separate and recombine the two halves of the input set. To maintain the same output channel spacing as that of the input, f2 must be twice f1. The total length is therefore 6f1.

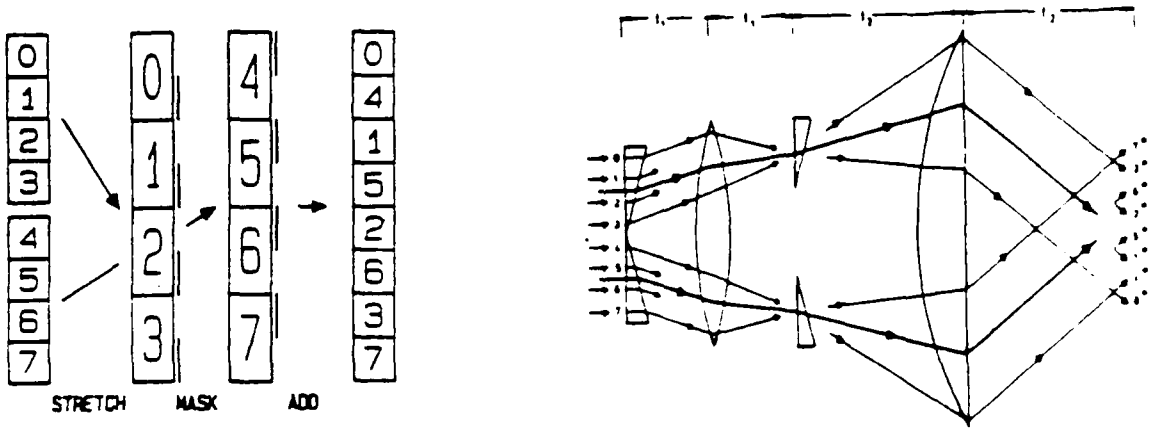



**Figure 3.1 Perfect Shuffle (Lohmann's scheme)**

Another implementation that has been proposed is one by Midwinter [15] which is suitable for one-sided operation, that is, the input and output elements are on the same side of the system. The advantage in this design is that the exchange switch logic array, which if not purely optical, can be on one side separated from the purely optical shuffle. The approach here is also, as in the previous scheme, to stretch-mask (shear)-add the inputs to obtain the perfect shuffle. However, the same optical system is folded in such a way as to incorporate a return path to the input side. The bottom half of the system only does a one-to-one imaging of the exchanged elements to the output port. Having the I/O on the same side is an advantage from the point of view of implementation. Figure 3.2 shows the folded perfect-shuffle scheme.

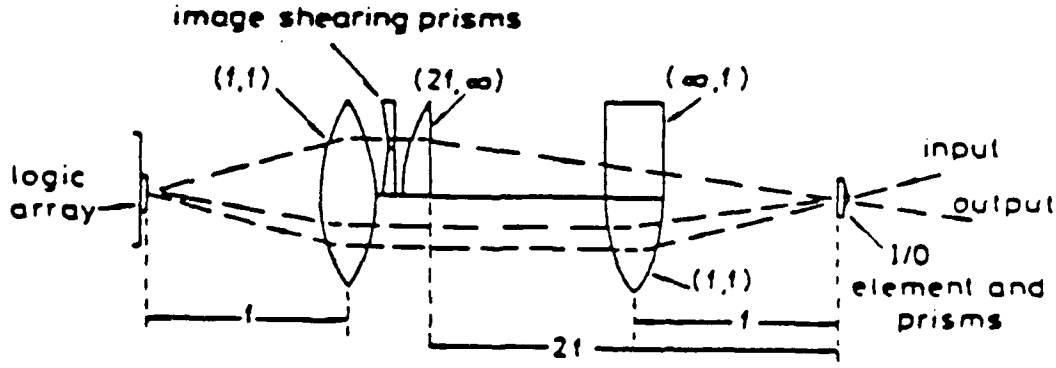



**Figure 3.2 Reflected Folded Perefect Shuffle**

The third scheme by Eichmann [14], mentioned previously, is more compact than the previous two methods. Two versions have been proposed: one in which two identical negative cylindrical lens are used side by side, and another where only one negative cyclindrical lens can be used with two prism wedges. In either case

collimated input beams are required. The total path length for the scheme is 2f (for the first implementation) as shown in Figure 3.3.

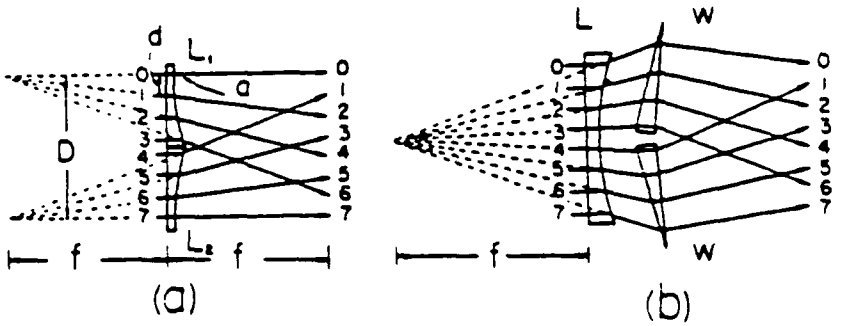



**Figure 3.3 Compact Perfect Shuffle**

### 3.1.2 Exchange switch

The focus by most researchers has been on the realization of the perfect shuffle connection. Since the shuffle connection can be done using passive elements, the system can operate essentially at optical bandwidths. The exchange switch for the SEN cannot be realized with simple passive devices since some form of control is required to either pass uninterrupted or deflect the input beams to the output of the switch. Unfortunately, scant work is evident on designing the exchange switch. We examine some alternatives in implementing a special case of the exchange switch as described by one reported work.

Stirk, et al, [13] examine means of constructing a compare and exchange module which is assumed to always receive two inputs. The exchange is based on comparison and not on any prioritized scheme as in the case of the message passing network of SPARO. However, some bulk optical techniques are discussed for realizing the basic exchange function, that is, the cross or bar (pass) configuration. Among the passive routing techniques suggested is polarization encoded switching using Wollaston prisms and controllable half-wave plates [12]. The retardance of the half-wave plate is controlled electrooptically by some photoconductor. When the photoconductor is activated by the comparison signal, the dynamic half-wave plate rotates the polarization of the orthogonally polarized input signals through $90^{c}$ (Figure 3.4). A polarizable beamsplitter or Wollaston prism can then spatially separate the two input signals. The performance of this scheme is bound by the bandwidth of the half-wave plate. These devices, due to their limited switching

power dissipation, can respond at millisecond speeds. With newer ferroelectric liquid crystals, one can expect to push this response time to the microsecond range.

An electrooptic approach has also been suggested using a system of detectors and modulators [13]. It uses nonlinear modulators which are normally transmitting unless a signal from the corresponding detectors converts them to be opaque. The scheme requires an electrical reset for the system to be operable on subsequent messages. Use of the electrical control thus requires using electrooptic devices such as PLZTs or ferroelectric liquid crystals, the performance of which would limit the message bandwidth.

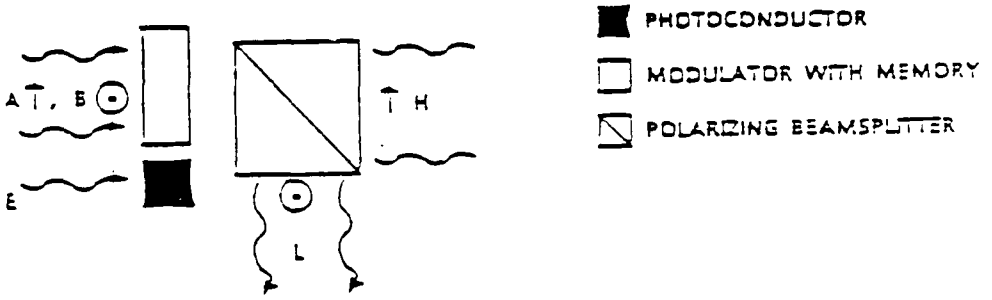



**Figure 3.4 Compare and Exchange Implementation (Stirk et al)**

In the above proposals, the response time of the control of the switch is always the limiting factor and offsets much of the speed advantages of the purely optical shuffle. (However, it is important to note that there are other advantages an optical shuffle can conceivably provide over an electronic one besides speed such as increased simplicity and physical compactness. We will visit these advantages later.) It is important to point out, that the above designs ignore many of the key functions that are essential to the exchange switch. A simple compare and exchange module will not be useful in the case of parallel processing that uses message passing. The next subsection summarily outlines the important functionalities required in the exchange switch as well as in the complete network implementation.

## 3.2 Required functionalities of the exchange switch

The survey of current attempts in designing optical SENs reveals that while the basic exchange switch has been examined [3, 6], some important requirements of the exchange switch have been totally ignored. These requirements are necessary when the network is employed in a parallel processing environment as in SPARO.

### 3.2.1 Conflict resolution by the exchange switch

The most important problem that has to be considered in a message passing environment is that of the resolution of possible conflicts among messages that arrive simultaneously at an exchange switch. Conflicts between messages occur since the correct routing of both messages may require different settings of the exchange switch.

There are a number of approaches to address this conflict depending on the desired complexity of the switch. Since one message is going to lose the conflict, these approaches differ in how to treat the losing message. One approach is to drop the losing message and let the sender processor (the source of the message) wait for a response. If a response is not received within a specified time, the sender processor will retransmit the message. The processor thus follows a specific message protocol sequence. In the case of fine-grained processing, implementing a protocol or handshake with each message is too high an overhead and therefore not acceptable. In the second approach, the losing message is rerouted, that is, the message is deliberately passed through the wrong output but certain modifications are made so that it reaches its destination. In case of the single-stage SEN, the rerouting, consists of simply resetting the counter field that indicates the number of passes the message had made. Because of the way the counter is typically implemented in electronics, it will be referred to as the mask. The mask denotes the age of the message -- in a network connecting N processors, a message is delivered in $\log_2 N$ passes when no conflicts occur. Thus to route a message, this mask is reset to 1 when the delivery is on mask value $\log_2 N$, usually represented modulo 1. No other modifications are necessary since the destination address does not change. The resolution of the conflict or the switch setting, as mentioned earlier, is done by

selecting the message which has a higher mask (counter) value (prioritized selection).

### 3.2.2 Delivery of messages

A message is delivered when it has successfully cycled $\log_2 N$ times around the network. The delivery of the messages thus requires examining the mask or counter value. When the mask value has reaches $\log_2 N$, then it must be extracted from the network and delivered to the processor. This is envisioned to be simpler than comparing the destination address with the address of the processor associated with the output of each shuffle. Such a mask checking scheme would be especially attractive if the processors are operating electronically while the messages are optical signals. In this scheme, the mask of a message (or some optical equivalent) could be checked optically or electrooptically at every cycle without removing the message from the network system. When the message has indeed reached its destination, it can be sent out of the network, converted into an electronic signal and queued at the input message buffer of the processor.

### 3.2.3 Detection of a valid message

Because of the possibility of noise in an optical system, it is important that the network can distinguish a noisy null message from a valid message. Accepting noise as a real message can ruin the network performance by causing unnecessary conflicts at the exchange switches as well as send spurious data to the processors. The traditional electronic approach recommends providing a message header with each message. In the case of an optical implementation, one may either provide a header bit or stream or a separate signal which indicates if a valid message is present.

We assume a synchronous operation of the network so that it can work efficiently with electronic processors. A synchronous design will also be easier to design and implement. The network cycle will be synchronized with the processor array clock whose rate is determined by the speed of operation of the complete shuffle-exchange. At the beginning of each network cycle, the processors will be polled for messages generated for routing during the previous network cycle. The message present signal for every processor will therefore be examined during the beginning of every

network cycle. Figure 3.5 shows the simplified schematic of the network and processor array interface. The message register in the network is required to hold a new message from the processor or a recirculating message from the network. An overview of the processor array and network connection is shown in Figure 4.1.

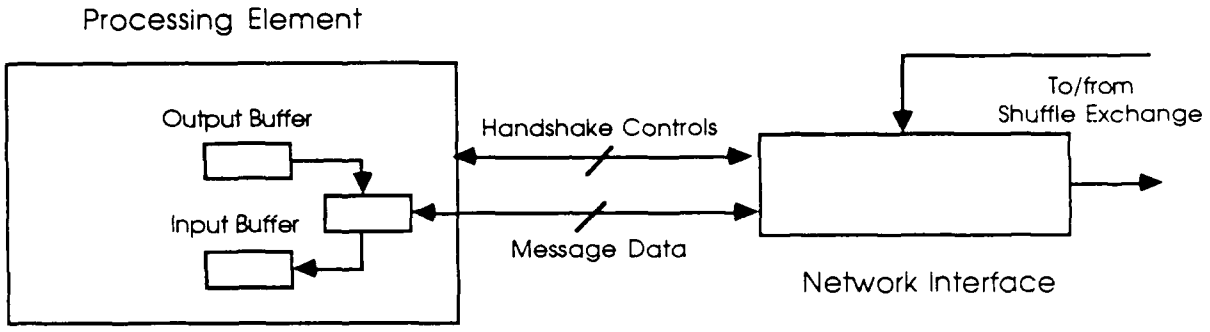



**Figure 3.5 Schematic of Network and Processor Array interface**

### 3.3 Possible approaches to an optical exchange switch design

The most difficult part of an optical SEN is the exchange switch design. Besides implementing the conflict resolution, the exchange switch design also governs the delivery of messages (since this depends on the way masks are represented and updated). Similarly, the optical technique used for implementing the basic exchange switch, that is, the simple cross or pass switch, also determines how the controls for conflict resolution will be realized. Our initial investigation indicates that the method of representing the mask information is critical to the nature of the switch design.

Here we discuss briefly the different candidate optical techniques that could be used for implementing the basic exchange switch. These are: acoustooptic gates, polarization encoding gates, waveguide or coupler, and photorefractive gates based on four-wave mixing. The goal is to pick the technique that results in the most speed-efficient basic exchange switch and then add the required functionalities of the network. We also provide one exchange switch design that we have investigated using Fredkin gates which have been proposed as an optical computing devices [18, 19]. While no optical implementation of Fredkin gates are known, they can be viewed as a useful computing primitive from which complex computing structures

can be built. The Fredkin gate design will be viable if the basic Fredkin gate can be implemented optically in a compact fashion.

### 3.3.1 Polarization encoding gate

The polarization encoding gate concept requires input message signals to encode the switching information as polarization. Thus, each message can have one of two polarization levels indicating whether the switch has to be in a pass or a cross configuration. The data in each message is assumed to be intensity-encoded. The optical switch is essentially a birefringent plate with a stored grating. The grating is visible to a message beam only if the message has the 'cross' polarization, in which case the input beam is diffracted across the plate. The situations when no conflicts occur, that is both messages require a cross or pass configuration of the switch, is relatively easy to implement. The cross configuration can be realized by allowing a negative diffraction for the upper beam and a positive diffraction for the lower beam. More details on this method are described in the next subsection on the final switch design.

Note that this approach differs significantly from the polarization switching gate discussed by Shamir et al [18]. In that gate, signals passing through a electrooptic modulator are rotated by 90° when the gate is activated electrically. The approach presented here is purely optical and appears to hold the most promise. However, the critical issue of conflict resolution can be incorporated in this encoding technique is not clear.

### 3.3.2 Acousto-optic gate

The acoustooptic gate is not very different from the polarization switching gate of [18], except that the switching information of the exchange gate is encoded in an acoustic signal. The gate is essentially an acoustooptic deflector that can be implemented in bulk or as an integrated SAW device. If the there is no acoustic signal on the gate control line, the input messages pass undeflected, otherwise they are deflected across. The problem with this approach is that the bandwidth of messages is in acoustic range which is lower than electronic bandwidths. While this is acceptable for transfer of large messages which arrive infrequently, it is too

slow for messaging in a fine-grained computing environment where the rate of computing depends on the rate at which messages can be delivered.

### 3.3.3 Photorefractive gate

A photorefractive gate based on four-wave mixing is an all-optical approach mentioned by the authors in [18]. Besides the two incident input message beams, the control consists of two pump beams. The inputs are transmitted if the control is absent, otherwise they are phase-conjugated resulting in switching between the outputs. Given the state of art in four-wave mixing, this approach appears the least feasible for implementation.

### 3.3.4 Waveguide or coupler

A modulated waveguide or fiber coupler is used for the switch. By electrooptically changing the guided mode effective index, one can change the coupling between the two coupled waveguides.

### 3.3.5 Fredkin gate implementation

Fredkin gates have been proposed recently as building blocks for optical computing. As Figure 3.6 shows, the Fredkin gate is a controlled crossover device that can be used for constructing circuit primitives (such as crossover, fanout, and delay) and computing primitives (such as AND, OR, and NOT). A Fredkin gate is also a conservative logic gate [20], that is, it is reversible (information lossless) and and bit-conservative (conserves the number of 1s and 0s that are present at the input). A control-specific Fredkin gate [19] is one in which the control and data lines are fundamentally different and cannot be interchanged.

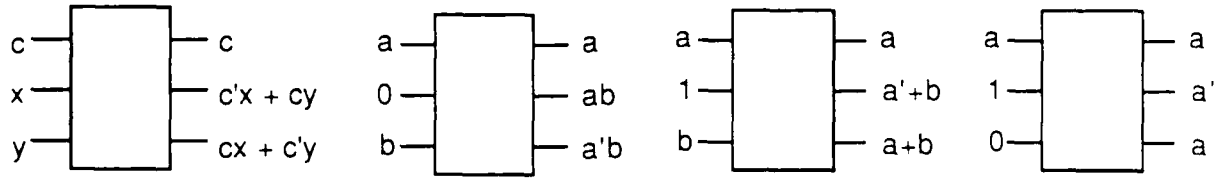



**Figure 3.6 Fredkin Gate and realizations of AND, OR, and NOT**
**a' is NOT(a)**

We have investigated the use of Fredkin gate for realizing the exchange switch and its controls. Our approach has been to map a Boolean functional description into a circuit using Fredkin gates. We completed, as an example, a minimal Fredkin gate design [19] for the switch control. (At this stage we have ignored the mask update control.) We describe below the functional specification and the corresponding realization. The minimal circuit realization derived is control-specific with respect to the mask comparison information only.

We define five inputs to the exchange switch. These are:

P1: Presence signal for the upper input of the switch, indicating whether a message has arrived. A 1 indicates the presence of a message while a 0 indicates no message.

P2: Presence signal for the lower input of the switch.

M: Mask comparison signal, or the outcome of the comparison M1 ≥ M2 where M1 and M2 are mask values of the upper and lower input messages, respectively.

A1: The destination address bit under the mask M. This bit decides the switch configuration required for the message in the current network cycle.

A2: The destination address bit under the mask M2.

C: The output of the Fredkin gate circuit which represents the control input to the last Fredkin gate that acts as the basic exchange switch. A C value of 0 implies a straight or pass configuration while a value of 1 represents an exchange or cross configuration.

The truth table for setting the control C is shown in Table 3.1. Note that R1 and R2 in the truth table represent the mask reset controls that are necessary to handle conflicts. A logic minimization of the combinational function for C yields the following sum of products form expression. A ¬ before a variable name denotes the complement of that variable.

$$C = P1\ \neg P2\ A1 + \neg P1\ P2\ \neg A2 + P1\ A1\ \neg A2 + P2\ \neg A2\ \neg M + P1\ A1\ M$$

The expressions for the reset signals are:

$$R1 = P1\ P2\ \neg M\ (\neg A1\ \neg A2 + A1\ A2)$$

$$R2 = P1\ P2\ M\ (\neg A1\ \neg A2 + A1\ A2)$$

Figure 3.7 shows a six gate implementation for the exchange switch in terms of the five input variables. The total delay in this switch is three Fredkin gate delays.

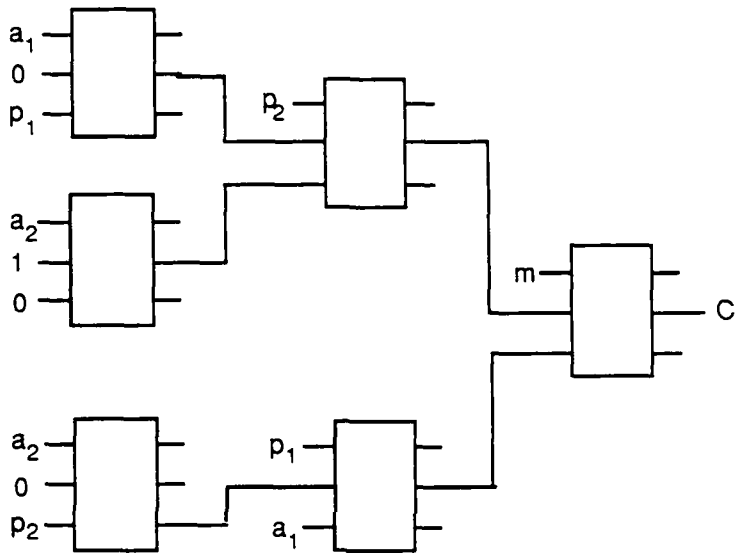



**Figure 3.7 Exchange Switch implementation from Fredkin gates**

As indicated earlier, the feasibility of the above implementation depends on the feasibility of the Fredkin gate implementation. As yet a feasible optical Fredkin gate has not been demonstrated. There have been some discussions [19] of cascading Priese gates or interaction gates to construct a single Fredkin gate but they have not been explored in detail. The speed and complexity issues of the our design will determine whether such an implementation is viable.

We now present a possible optical switch design based on the study of the methods described here. The results of this effort indicates that while parts of the exchange function for the switch can be implemented very elegantly using optical computing techniques, the optical implementation of the entire exchange function and its control is beyond the feasibility for current demonstration.

### 3.4 Design and analysis of an optical exchange switch

The basic exchange switch is based on the polarization coding method described previously.

3.4.1. A passive optical exchange using polarization as control.

As described earlier, passive exchange of signals in a SEN is possible by forming a grating structure in a birefringent material so that light propagating with one polarization is affected by the grating and deflected to the appropriate path, while light of the crossed polarization is unaffected by the grating and passes through the material essentially undeflected.

To operate such a polarization-based switch, the polarization of the light beams (messages) must be switched or set based on the switch setting control logic that is a function of the address bits and the mask values or the message ages. (Note that in case of an optical implementation, we will use message ages to denote mask values since the latter has an electronic implementation.) While many devices are available which can switch polarization quickly based on electrical control [24, 25], all-optical devices tend to perform this operation either slowly or over a relatively long interaction distance. While the actual routing is passive and optical, the control for this polarization setting operation must be electronic if it is to be implemented with components and materials existing at this time.

One advantage of this approach is that the information carried along with the message regarding whether an exchange should occur or not does not have to be actively decoded at the routing device. Instead, when this information is calculated, it can be represented in such a way that the decoding is a passive beam propagation phenomenon. Since the calculation of whether a particular message should be exchanged or not must be performed at each step through the network anyway, the overall switching time can be reduced.

3.4.2. A passive optical message age update.

In the SEN, in addition to address information which conveys to a switching node whether it should exchange inputs or not, a message also carries its age information. Again, the large number of information representation formats for

optical computing can be used to reduce the overall speed of updating this information. Since the propagation of information through the network can be thought of as routing in a plane (yz in Figure 3.8), in real spatial dimensions, an optical routing implementation might be performed with planar geometry, using mirrors, prisms or gratings to deflect the paths of the beams carrying the messages through the network. This leaves another dimension (x) orthogonal to this plane to use as information encoding. One simple method of representing and updating the number of passes a particular message has taken through the network is to represent the number of passes as a spatial position in this orthogonal direction. With each pass through the network, the position of the message beam is shifted one unit in the perfect shuffle plane by means of a prism or a grating. Because this information is represented spatially in a dimension (x) orthogonal to the interconnection plane, shifts in this dimension can be independent of the routing pattern in the interconnection plane. Since there is no decision on whether to shift or not because propagation of any message for another pass represents a shift, this can be performed with a permanently configured passive optical system of gratings or prisms.

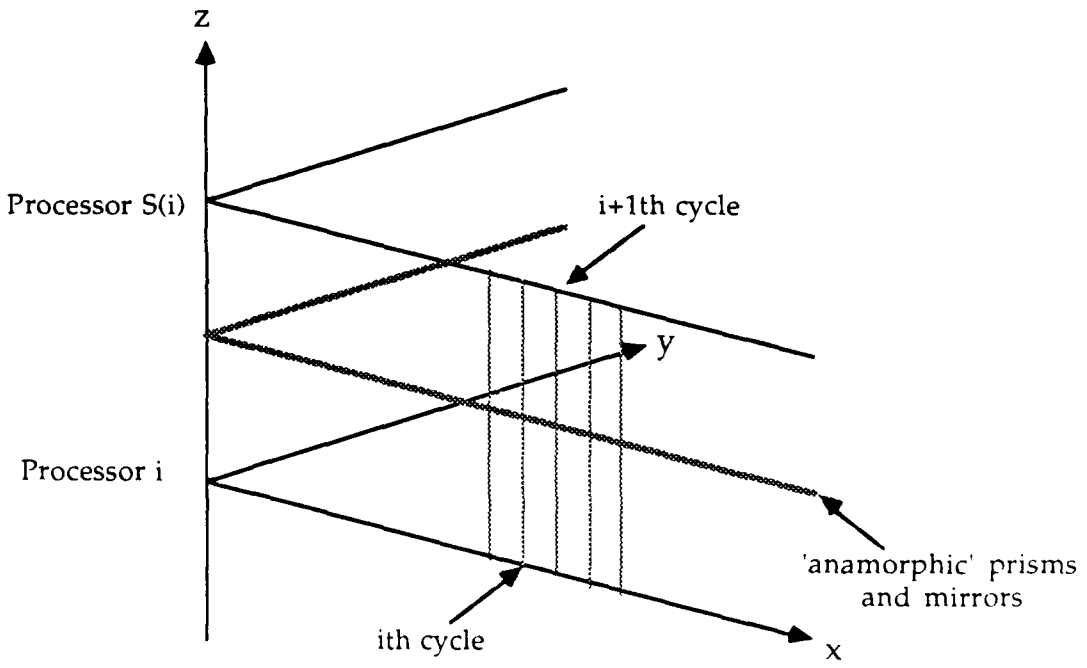



**Figure 3.8 Representation of message age by spatial encoding**
**S(i): perfect shuffle destination of i.**

3.4.3. Detection of age and collision decisions

Perhaps the biggest shortcoming of the all-optical exchange-switch is the lack of a method or devices to perform the decision on which of the the two messages gets priority of the exchange control when a collision occurs at a node in the network. This decision is based on the ages of the two messages, usually with the older message getting priority and the younger message starting routing afresh in the network.

Optical computing strategies have traditionally had serious problems implementing integer comparisons. One method that could accomplish this with light-speed throughput is a look-up table hologram in which all combinations of age comparisons are stored, and the appropriate combination is recalled through Bragg reconstruction of the correct output. However, the number of combinations which the hologram would need to store for a large system would be prohibitive. Referring to the last four rows of truth table shown in Table 3.1, one can verify that the number of entries in the truth table is equal to the order of the number of message comparisons or $O((\log_2 N)^2)$ for a SEN connecting N processors (PEs). The actual number of entries required is $4(\log_2 N)^2+5$. Thus for a SEN connecting 1K or 1024 PEs the look-up table required for every exchange switch control must have stored in it 405 entries. The number of inputs is 24, while the number of outputs (control of switch configuration, reset controls for the age of each message) is 3. A single hologram of this size is not a problem. But since N/2 exchange switches are required, the 1K network must be able to physically accommodate 512 such holograms.

| | $P_1$ | $P_2$ | M | $A_1$ | $A_2$ | C | $R_1$ | $R_2$ |
|---|---|---|---|---|---|---|---|---|
| No Packets | 0 | 0 | - | - | - | - | 0 | 0 |
| One packet | 1 | 0 | - | 0 | - | 0 | 0 | 0 |
| | 1 | 0 | - | 1 | - | 1 | 0 | 0 |
| | 0 | 1 | - | - | 0 | 1 | 0 | 0 |
| | 0 | 1 | - | - | 1 | 0 | 0 | 0 |
| Two packets, No conflict | 1 | 1 | - | 1 | 0 | 1 | 0 | 0 |
| | 1 | 1 | - | 0 | 1 | 0 | 0 | 0 |
| Two packets, Conflict (Message Comparisons required) | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |

**Table 3.1 Lookup table for smart exchange switch**

Other possible approaches would require either the representation of the ages or the number of passes as analog levels and a threshold comparison of the two intensities or a Boolean algebra calculation of the comparison. While fast analog threshold detection might be possible with multiple quantum-well devices, the fabrication technology of these devices is not yet advanced enough for them to be used reliably in large-scale network implementations. Boolean operations for optical computing are also not feasible for large systems at this time. Devices which perform fast Boolean logic operations in optics tend to be multiple quantum-well devices, which are difficult to use for large systems. Spatial light modulators, which can be easily expanded to perform Boolean operations for large systems, unfortunately operate at very low speeds.

### 3.4.4 Conclusions in designing an optical smart exchange switch

In summary, there are two aspects of the exchange operation in which optical implementation can be applied to decrease routing delay time for messages passing through a SEN. These are the passive exchange of polarization coded messages, and the passive update of the number of passes that a message has taken through the network on the way to its destination. Unfortunately, this is not enough to justify

an all-optical implementation of exchange operation, because the control of polarization changes and the decisions for handling collisions cannot be implemented easily in high-speed devices with optical control using components which are available at this time.

# 4. HYBRID SEN IMPLEMENTATIONS

To quantitatively evaluate the relative advantages of an optical SEN, specifically the optical shuffle, we examined two different SEN implementations. The first implementation is a purely electronic design of both the exchange switch and the shuffle connection. A logic design and analysis was conducted to determine the complexity of the design in terms of silicon area and wire lengths required for connections. The same analyses also yields the first-order estimate of the speed of operation of the SEN when the fastest semiconductor technology is used. The second implementation is a hybrid one that uses an optical shuffle in conjunction with an electronic exchange switch. The optical shuffle section is expected to be faster and more compact than a hardwired electronic implementation [17]. A similar analysis as in the case of the first was undertaken to assess the second design. An optical SEN implementation can then be compared to both these implementations on the basis of their speed-complexity product.

## 4.1 SEN and processor array interface

The single-stage SEN is redrawn in Figure 4.1 showing the Processor Array, consisting of 1024 processing elements (PEs) for fine-grained computing, communicating with the network through the Network Interface (NI), or the control portion of the network that handles the transfer of messages between the PEs and the network. Here we will focus on the complexity and the cycle time, the delay experienced by a message to pass once through the shuffle-exchange stage, of the network.

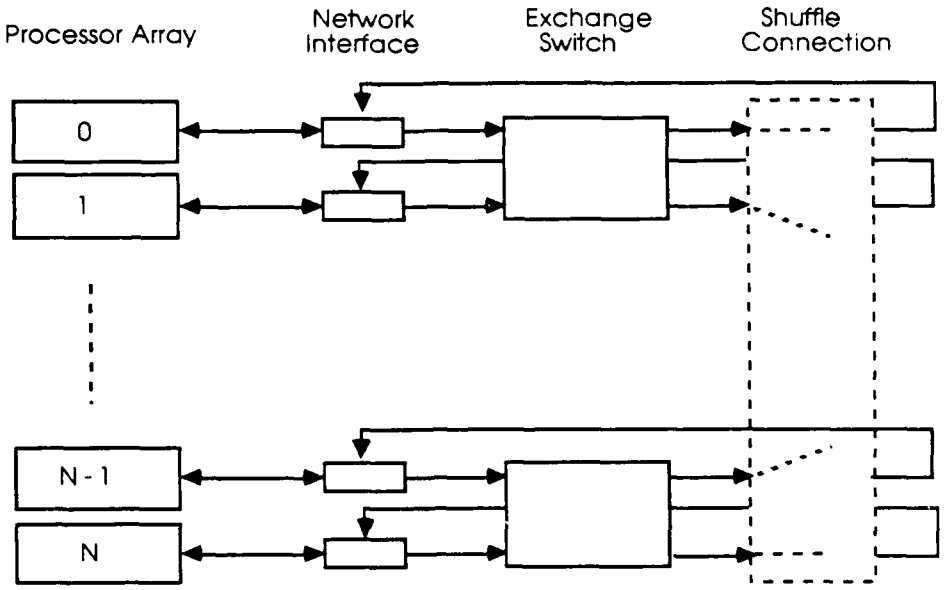



**Figure 4.1 SEN and Processor Array**

To isolate the performance of the SEN from that of the implementation of the PEs, we will define the network cycle to be the difference of the time when a message is accepted into the NI and the time when it is loaded back into NI for delivery or for recirculation.

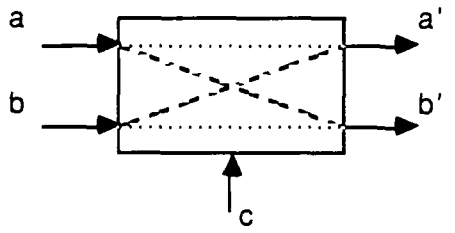
Each PE, as Figure 3.5 shows, contains two buffers for storing outgoing and ingoing messages. A message to be sent to another PE is queued at the output buffer (OB). At the beginning of a network cycle (defined as the time taken by a message to cycle once through the shuffle-exchange network), if there is a message in the OB, the PE requests access into the network through the handshake line Processor Request. The network can receive a message if there is no circulating message at the PE. The NI corresponding to the PE communicates this information to the PE via the Processor Access line. When access is granted to the PE, the message is loaded from the OB into the NI through the message data line/lines. Similarly, when the NI at the end of a network cycle has a message to be delivered to the PE, it uses the handshake signal Network Request to check if the Input Buffer (IB) of the PE is not full. If IB is not full, the PE uses the Network Access line to signal the NI to transfer the message over the data lines. Note in Figure 3.5 we have assumed that the data lines between the PEs and the NI are bidirectional ports. This has been done to reduce the number of I/O connections.

The mechanism for delivering messages from the SEN to the PEs proceeds as follows. When a circulating message is received in the NI, it is checked to see if it has completed $\log_2 N$ passes successfully. If it has, then it is transferred to the PE, otherwise it reenters the SEN. This checking can be done within the NI module or inside the SE stage.

We now focus on the Shuffle-Exchange (SE) part of the network independent of the network interface portion.

### 4.1.2. Shuffle Exchange Stage

The basic exchange switch, shown separately in the Figure 4.2, is one that is controlled by one input that produces a cross or bar connection between the input and the output. An useful exchange switch must satisfy all functional requirements listed earlier in Section 3.2. We will call such a switch the smart exchange switch (SES) described by the Boolean equation (from Section 3.3). The Boolean expression for the switch control assumes that both the deciding address bit extraction as well as the mask comparison has already been completed. There are a number of possible hardware schemes, serial and parallel, for realizing both operations. We now consider some electronic implementations.




a' = NOT(c)a + cb

b' = ca + NOT(c)b

**Figure 4.2 Basic Exchange Switch**

## 4.2. Electronic SEN Implementations

In order to uncover the critical architectural issues in designing optical SENs, we first consider a purely electronic design. A number of issues determine the nature of the electronic implementation in terms of the size and complexity as well as performance. These include pin constraints of a chip, the number of backplane interconnects between printed circuit boards or the backplane wiring constraints.

and the off-chip and off-board interconnect delay as compared to the gate delays within a chip. We examine these issues briefly as inputs to our design methodology, and then present the possible performance and size of electronic implementations.

### 4.2.1 Pin Constraints

The design approach most affected by the limits on pin counts is the issue of parallel versus serial message data transfer. Consider the parameters of the SPARO (Symbolic Processing Architecture in Optics) architecture. There are nominally 1024 PEs communicating via messages composed of five essential fields: the destination PE address, the source PE address, two data/address operands, and the instruction. For a 1024 processor architecture, the address is 10 bits wide. Although the specific application will decide the size of the data used, we assume that a 32-bit wide data would suffice. The instruction word was assumed earlier to be encoded as 1 bit per macro-instruction to simplify the instruction decoding in optics. We expect that more than 30 macroinstructions, each of which requires 2 to 3 machine cycles. In the case of an electronic design where binary decoding can be done relatively simply in the PE, we will assume 6 bits are required to encode all macro-instructions. The total width of the message is thus 90 bits (2(32) + 2(10) + 6). A message of this length presupposes that the mask for each message is generated in the network. However, to avoid increasing the complexity of network, it is preferable that the mask be generated by the processor and sent as part of the message for purposes of routing in the SEN (Figure 4.4). The total message length would then be 100 bits. Larger or smaller sizes of the messages are possible depending on the size of the data necessary. As we shall see later, the length of the message has a profound effect on the performance and implementation of the SEN.

The control lines required between each PE and the corresponding row of the SE were discussed earlier. Four handshake lines are required: Processor Request by the PE to transfer a message from the OB of PE into the NI, Processor Grant to grant the processor request, Network Request by the network for message delivery from the NI into IB of PE, and Network Access to grant the network request. Since messages are all of fixed length, we do not require acknowledge signals after message transfers have been completed.

| Destination | Mask | Source | Instruction | Operand 1 | Operand 2 |
|---|---|---|---|---|---|
| 10 | 10 | 10 | 6 | 32 | 32 |

**Figure 4.3 SPARO Message Format**

The problem of pin limitations arises when considering how the message has to be transferred between the PEs and the SEN: 104 bits of control and message information if the messages are transferred in parallel between the NI and the PE, or 5 bits for control and serial data lines. Note that in our initial electronic designs all message lines are considered to be bidirectional to reduce the space overhead. In the first case, all messages (at the beginning or at the end of the network cycle) can be transferred in one clock cycle after handshaking. In the second case, 100 clock cycles are required to transfer the message serially. While the serial option is 100 times slower, it requires 1/21th the number of pins at the output of the PE. There is thus a space-time trade off to be considered. The real question to be answered is the total space-time complexity. If 1024 PEs are placed on one board to reduce off-board delays, then the board has to accommodate 1024*104 or 106496 lines between the PE array and the SEN. By comparison, the serial transfer scheme only requires 5120 lines. How many PEs can be put on a chip or package and then on a board therefore depends on the pin limitations on the chip as well as the number of interconnection lines that can be squeezed on a single board. These issues in turn depend on the technology used to design the board.

### 4.2.2 Off-chip Interconnection Delays

Since the complete network, that is, the exchange circuitry as well as the shuffle connection, requires a multi-chip (possibly multi-board) implementation, off-chip delays will be a design concern. The problem of interconnection delay becomes severe for a large perfect shuffle where exchange stages are switched at high speeds. The network delay, or the time taken by a message bit to pass around the complete SEN, depends partially on the actual interconnection length between the output of the exchange stage and the register that delivers the message to the input of the exchange stage (see Figure 4.1). Since the shuffle connection is not modular, this length increases with the size of the network. We will examine the relative importance of the interconnection delay when examining SENs implemented in different electronic technologies.

### 4.2.3 Backplane Wiring

The problem of backplane wiring arises if a multiboard implementation is necessary when all PEs and the corresponding interconnects cannot fit on one board. The number of boards and the total delay would then be determined by the number of backplane interconnects possible. The maximum number of board-level interconnects depends on the technology used to construct the board. Thus, using thin film multilayer (TFML) boards allow for much faster and more dense interconnects than do standard PVC PCBs with edge connectors.

Both pin count and packaging constraints and limits on backplane wiring will therefore determine the nature of transfer of the message, that is, how serial or how parallel. The other factors are the complexity of each SE stage. The larger the area of a single SE stage, the fewer PEs can be fitted on a chip and thus fewer chips on a single board. We will consider these factors in more detail in the next section.

### 4.2.4 High-Speed Electronic Implementation

We have examined the complexity of the circuitry required to implement the complete shuffle-exchange network in electronics. Both GaAs and Si ECL technologies were considered for high-speed implementations. Given the large size (1024) of the network, we initially considered bit serial transfer of messages. We examine the design implications for a parallel message transfer scheme later. As we will show, the nature of the message transfer in the network is critical in determining the complexity and performance of the network implementation.

The size of each exchange stage and its controls is estimated first to determine the layout complexity and thus the total area, size, and speed of the network. The circuitry in the exchange includes the NI and its controls, the combinational logic to generate the exchange switch settings, and registers to hold the message during recirculation. Figure 4.4 shows a schematic of the electronic SEN. The operation of the network is now explained in more detail.

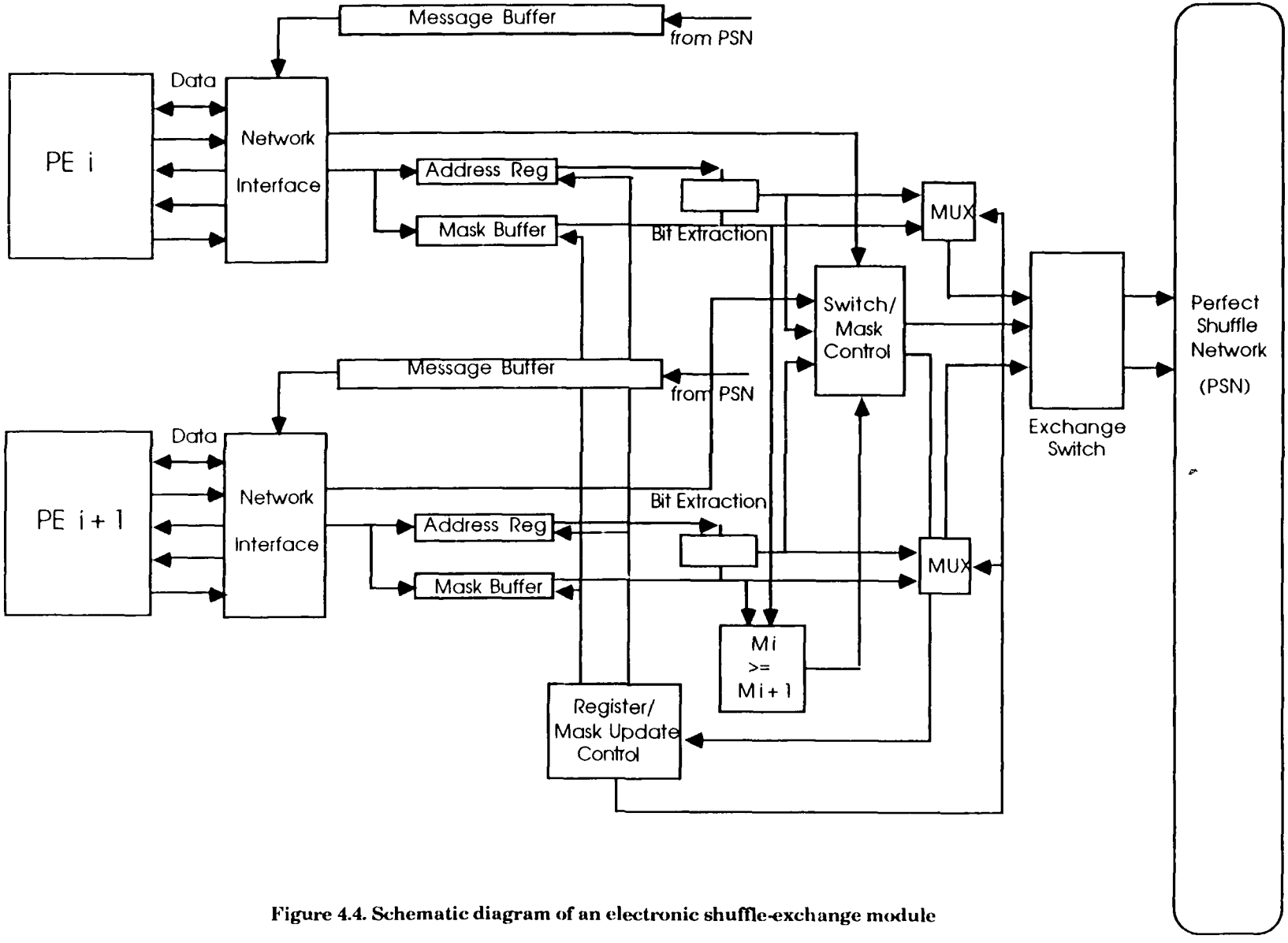



**Figure 4.4. Schematic diagram of an electronic shuffle-exchange module**

A message is accepted into the network with the destination address field entering first (as shown in Figure 4.3). The address and mask fields of the message are successively loaded into their respective registers, so that the deciding address bit can be extracted. The deciding address bit extraction is achieved by serially shifting the mask and address register contents and ANDing the output bits. While the mask register is being cyclically shifted, the mask comparison between masks of two messages can be done in parallel. The mask comparison result and the deciding address bits are fed to the switch and mask control logic. While the presence bits of messages have not been shown in Figure 4.4 to simplify the diagram, they can be derived in the NI by examining the processor request and grant lines. When the exchange switch is set, the registers are alternately emptied to serially pass their message to the shuffle stage. The message buffer holds the initial portion of the message while the remaining message portion passes through the exchange switch.

We now examine implementation of the above SE circuit in different technologies.

*Network delay for GaAs implementation*

In the case of GaAs, the total number of gates (the average gate is a two-input NOR or NAND) required for each exchange stage for every pair of PEs (Figure 4.5) is expected to be less than 1300. (The major proportion of these gates are consumed by registers and buffers.) If the fastest usable GaAs technology (from Honeywell) were employed, gates with delays of the order of 100 ps (10 GHz) could be used. Since the number levels of logic between switchable gates will be 3 to 4 on an average, the clock speed for operating the network would be about 2 GHz. At that speed, the current and near-future level of integration allows between 3000 and 4000 gates on one chip. This allows us to realistically fit in 2 exchange stages (for 4 PES) on a single chip. Using state-of-the-art multichip carriers (from Honeywell), we could fit close to 25 chips or 50 exchange stages on a rectangular multichip carrier package (MCP) measuring 3.7 by 2.4 inches. The current limit on the number of pins in a pin grid array is about 575. Since each stage requires one output message line and five input lines, a single package can be used to pack about 100 PEs. We are assuming that the data line is bidirectional. If unidirectional data lines are used, six lines are required to connect each PE to the network.

For a 1024 SEN, we would have to place at least 10 or 11 such MCPs on a PWB (printed wire board). Since standard PWBs have low dielectric constants, off-package delays will be higher than those inside the package unless polyimide substrate is used for the board. Interconnection lines would be done in copper on the polyimide. The shuffle stage would have to interconnect these 10 packages on the board. Each package would have 100 serial output lines for messages leaving the exchange stage for the shuffle connection.

Because of the non-local nature of the shuffle connection and the limit in integration, we cannot integrate the shuffle connections on the package. The off-package delays for the connections are directly proportional to the wire length. If multilayer (TFML) connections are used (currently 5 layers with 3 for ground and power), the longest vertical length between packages is 5 MCP heights, that is, 5" x 3.7" or 18.5". Since this line is large in length, there is considerable loss in the lines. It would be difficult to run GaAs at 2 GHz over long interconnection lines unless impedance matched input and output buffers are used at the input and output pins of the exchange stages. (The bigger problem is that of timing, or the distribution of clock to all parts of the SEN which is to operate synchronously.) Since the delay on copper on polyimide is 62 ps/cm, the longest interconnect delay in the shuffle stage is 2913.4 ps or nearly 3 ns. Note. that the processor to exchange stage connections are not as much of a bottleneck since they are direct local connections between PEs and the MCPs for the exchange (Figure 4.5). We assumed, in our preliminary analysis, that all PEs can be connected by the SEN on one board. As it turns out, this is not possible since packing 1024 PEs alone will consume a complete board-see note on processor complexity in the next subsection. Thus the PE array and SEN connection will be across boards and not on a single board. For purposes of determining the upper limit on the SEN performance, however, we assume that the PE and SEN connection problem can be solved, and therefore focus on the one-board SEN performance. We have assumed that load impedances will be matched at the MCP pin boundaries. Thus the delay between connections of successive stages of the network is over 3 ns given that some gate delays have be accounted for within the package. We can assume the worst case total network delay time to be about 4 ns.

In our delay computation, we have assumed that the clock can be distributed such that no clock skews occur. At Honeywell we use a star configuration in distributing

the clock within a package to ensure that the clock propagation delay is the same for all modules. If clock synchronization is a problem at the board level, and we believe it will be, we could conceivably employ a holographic optical element (HOE) to distribute the clock.

*Network Delay for ECL Implementation*

Because of the relative severity of off-chip delays in the GaAs implementation, we considered a more mature technology, ECL, as another choice for implementing the control and exchange stage of the network. The advantages of ECL over GaAs despite its slower speed is the higher level of integration as well as a relatively smaller penalty for off-chip connections. Today, we can integrate 10000 gates on an off-shelf ECL gate-array chip with relative ease. With the state-of-the-art ECL technology, maybe even 20000 gates could be put on one chip. (This would be possible since the connections in the SE stages are regular and simple with low fanouts.) We are assuming of course that the high heat dissipation problem for the on-board ECL circuitry can be solved. With such a level of integration, one could account for SE stages for 20 PEs. To completely fit all 1024 PE connections, we would need about 50 chips on a board. Each chip would have 200 pins for serially transferring data in and out of the network. This number of pins is quite feasible today. By pushing technology to its limits, the data could be clocked through at 200 MHz if delays across the board are not significant. It would be reasonable to assume that the interconnection length would be close to that of the GaAs SEN. Thus the network delay will also be close to 4 ns. We assume as before that an optical clock distribution is possible using a HOE (holographic optical interconnect) scheme. Since the ECL clock is run as much as 50 times slower than that for GaAs, the interconnection delay in the ECL SEN is a very small fraction of the cycle time.

We now examine the total cycle time of the network for each technology.

### 4.2.5 Network Cycle Time

The network cycle time is the sum of the propagation delays for the control and the exchange switch, the shuffle path delay, and the total time to transfer the message. Since the number of gates is expected to be less than 5 between clocked stages, and since the shuffle connections are compact, these delays do not slow the clock down.

The message transfer is therefore dominated by the time taken to transfer 100 bits. Since the delay across the board is not significant, the head of the message will arrive through the shuffle connection to the next exchange stage before the tail of the message has passed completely through the exchange switch. Even if the two 10-bit registers are used to alternately hold 10-bit chunks of the message (see Figure 4.4), a message buffer is required to save a portion of the message before the exchange switches can be set for the next pass.

In the case of GaAs, the message bits are pipelined out of the source PEs at 2 GHz. The network cycle time is 54 (100*500 ps + 4 ns) ns or effectively the network operates at 18.52 MHz. The network delay time is thus 108 times slower than the GaAs gate delay (500 ps) because of the serial transmission of the message. Since two 10-bit registers are used to hold portions of the message, a 80-bit message register is required. In case of ECL, the message transfer time is 504 (5 * 100 + 4 ns) ns. This implies that the network effectively operates at 2 MHz.

In case of either technology, the serial message transfer causes the network to be a bottleneck. This bottleneck is especially serious when the PEs operate on the same clock as the SEN. An average message requires $1.5*\log_2 N$ network clock [5] or $150*\log_2 N$ clock cycles to be delivered (100 bits per message), when the network load is not more than 0.25. Therefore for a 1024 SEN, a message requires 1500 clock cycles to be delivered. If a specialized reduced instruction set (for combinator graph reduction) architecture is used to implement the PE, a message can be generated at best in 3 to 5 cycles, assuming some parallel loads are allowed within the PE. Thus, the message generation rate (all 100 bits generated in parallel) is about 300 to 500 times higher than the message delivery rate if the message is transferred serially, and 3 to 5 times higher if transferred in parallel. In an ideal situation, where no bottlenecks exist, the network should deliver messages at approximately the same rate at which the PEs generate them. Thus, the message bandwidth in the network must equal the message generation bandwidth in the PE. This implies that the SEN operates on a clock that is either operating at a ridiculous 300 to 500 times faster than the PE clock when transferring messages serially, or accepts and transfers messages totally in parallel with a clock that is less than an order of magnitude faster.

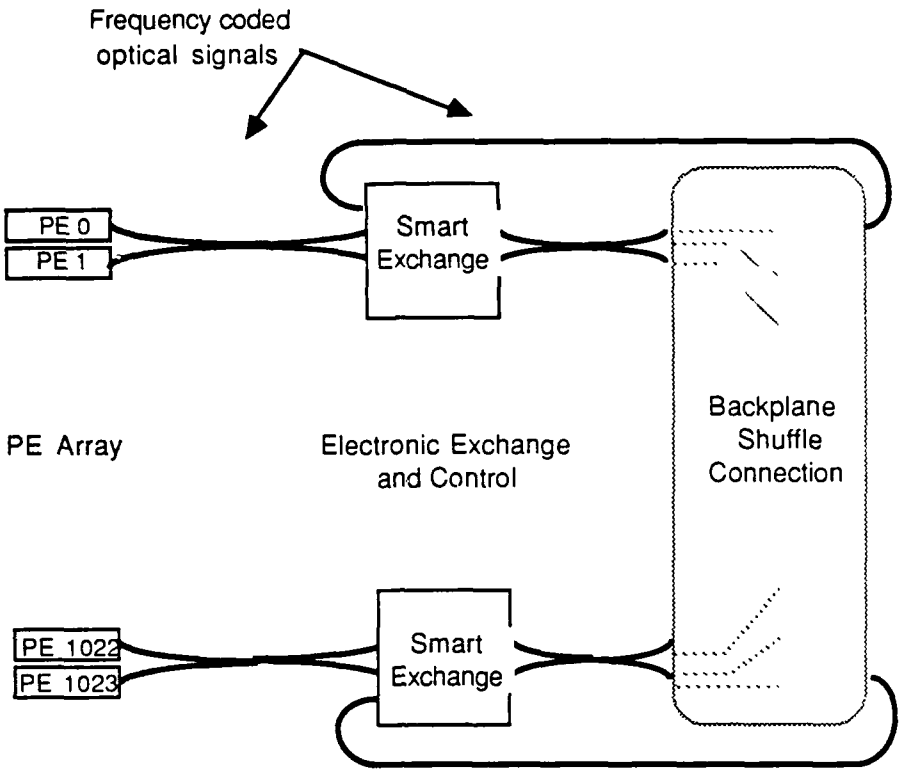



**Figure 4.5 Scheme for an electronic SEN and PE array**

4.2.6 Parallel versus Serial message transfers

The previous discussion showed that the message bandwidth in the network is inversely proportional to the message length. The increased bandwidth required in the SEN motivates us examine the complexity of a parallel implementation where messages are transferred in a parallel or quasi-parallel fashion.

For the same level of integration as described earlier, an ECL chip built with 20000 gates can accommodate exchange stages for 20 PEs. However, instead of 20*6 (5 between PE and exchange and 1 between output of the shuffle stage and exchange) or 120 I/O pins, now 20 * 204 (104 between PEs and exchange and 100 between output of the shuffle stage and exchange) or 4080 I/O pins are required, ignoring ground, power and clock connection pins. Since this is not possible, a single chip cannot be deemed to contain more than one exchange stage for 2 PEs since about 250 is the limit to the number of pins to a chip. In such a case, 500 such chips would have to be interconnected in a shuffle connection where each channel of the shuffle now has to connect 100 wires per channel or a total of 102400 wires for a 1024 shuffle. If

board-level interconnects are used, then many boards are required in the implementation of the shuffle connection since typically only 250 to 300 backplane connections are possible with standard edge connectors. Clearly, the pin limitations and size complexity makes a large shuffle-exchange for parallel message transfer impractical in electronics.

In summary, one notes that an electronic implementation, GaAs or ECL, for a large shuffle-exchange could be operated at high speeds, over 200 MHz for ECL and over 1 GHz for GaAs. However, severe limitations of the packaging technology and the level of integration of high-speed semiconductor technology forces a serial transfer of messages when a large SEN is desired. Unfortunately, when messages are transferred serially, the message throughput varies inversely as its length. For a modest message length of 100 bits, suitable for the level of fine-grained computing, the message throughput is less than 1/100th the data rate. Table 4.1 summarizes the electronic SE network cycle times and the corresponding message latencies. The message latency is defined, for our purposes, to be the average time required to deliver a message. We have assumed, using Lawrie and Padua's results [5] that a message requires an average of $1.5\log_2 N$ cycles to deliver if the network is not loaded much beyond 0.25 (that is, about 25% of all stages have messages in transit).

| Technology | Gate delay | Network latency | Message latency |
|---|---|---|---|
| GaAs | 500 ps (2 GHz) | 54 ns (18.52 MHz) | 810 ns (1.24 MHz) |
| ECL | 5 ns (200 MHz) | 504 ns (2.0 MHz) | 7.56 us (132.3 KHz) |

**Table 4.1. Performance of electronic SENs**

For comparison with the above performance, we examined the same issues for electrooptic and optical implementations.

### 4.3. Optical and electro-optical SENs

As in the electronic implementation both parallel and serial message transfers can be considered. As before a parallel message transfer scheme in optics requires that each channel be 104 (100 message lines and 4 control lines) signals wide. A total of 100K signals in a single optical system appears difficult in the near future if guided optics is used. Thus, the optical system may have to be quasi-parallel since the

worst case serial transfer cannot provide acceptable throughput unless the switching speeds in the optical SEN is hundreds of times faster than the PEs. We will explore the quasi-parallel approach in more detail later.

Since an all-optical implementation has not yet been designed, we will focus on the architecture of electrooptic and hybrid implementations. One thing is clear: if the optical network has to provide an advantage over an electronic one, using an optical serial shuffle together with an electronic exchange and control will not be an advantage. This is because the serial delay of the message in electronics is a serious bottleneck, and using an optical shuffle will only add further electron-photon conversion delays. We will therefore examine optical methods to improve the bandwidth of message transfers. Another issue that requires consideration in designing the SEN is the nature of its interface to the PE array. The network interface depends on the size of the PE and therefore on its complexity. The next subsection examines the complexity of a special-purpose graph reduction processor.

### 4.3.1 *Processor Complexity*

The nature of optical implementation depends on the level of connectivity, that is, whether the connections between the PEs are within the board or off-board. Since we are building a fine-grained parallel system, the size of each PE dictates the nature of connection. For this purpose, we examined the functional requirements and the architecture of a specialized combinator graph reduction (CGR) PE.

Our initial estimates show that a reduced instruction CGR PE will have about 30 to 35 hardwired instructions, each of which executes 2 to 3 steps. The typical sequence of operations are as follows. The PE receives a message, decodes it, operates on it, and sends out a message in response. If a simple ALU (no multiplier) is used, the PE could be implemented with about 1600 gates in Honeywell's high-speed (50 MHz), high density CMOS process. To give more power to the PEs, a larger ALU equipped with a multiplier could be shared by a pair or more PEs on the same chip. As many as 11,000 gates can be put on 1 CMOS gate array chip (400 mils X 400 mils). Thus, 6 to 7 PEs can fit on 1 chip. As many as 20 gate array chips could be squeezed on a package 3.25 " X 2.6". Thus one package would account for 120 to 140 PEs. The limit in the packaging is not in the total gates available but in the number of I/O pins. The maximum number of pins possible in such a package is 575 in a pin grid array.

Thus if 100 PEs were dedicated to a package, each would have only 5 or 6 (depending on a bidirectional or unidirectional data line) I/O pins allocated to it, ignoring common clock, power, and ground lines. This limited pin allocation per PE forces a serial message transfer scheme wherein all messages in and out of the PE have to be serial.

At the level of integration discussed, at most 10 packages can be fit on a large board. The limit at the board level would be in the number of board connections as well. If 1024 PEs were accommodated on 1 board, the number of I/O lines in the board at 5 or 6 I/O pins/PE will be over 5000. Thus, connecting a SEN to this board is not possible using standard electronic wiring connections. More importantly, it is clear that to build a scalable machine, consisting of 10,000 PEs or more (necessary for fine-grained computing), a multiboard solution is desired. Therefore, the SEN must be operational across multiple boards and not just within the board. The number of boards required for all PEs is dependent on the functionality and granularity of a PE. We now examine this important issue in more detail.

### 4.3.2 Processor Granularity

The PE granularity influences the SEN design in two ways. First, the coarser the granularity, the fewer the PEs required to solve the problem. This implies that a small number of boards will suffice to accommodate all PEs. The architecture of the processor/memory subsystem for coarse-grained processors will of course be quite different from the one chosen here. Second, coarser-grained PEs will operate on a larger problem (that is, on subgraphs rather than on individual nodes in CGR) and therefore will have less frequent communication with other PEs. The message generation frequency will therefore be considerably lower. However, the size of messages may be considerably larger. For example, the messages may contain subgraphs rather than single node information. The increased size of messages will tend to keep the bandwidth of messages high even if the message generation rate is decreased. One way to keep the message size down to the lengths that we are considering here (100 bits) is to use a radically different architecture such as shared memory and PE clusters. Using such a different architecture implies solving a different sort of PE communication problem which we will not consider here. We will instead focus on the general tradeoff of PE granularity and message bandwidth.

In the SPARO architecture that has been developed for fine-grained CGR, each PE contains and operates on a single graph node. There is no concept of memory since the registers in the PE specify a graph node completely. While this approach seemed suitable for optics where a complex processor could not be designed, when considering an electronic implementation other problems surface. First, the number of PEs required in the architecture is not defined by the size, in terms of the number of nodes, of the original combinator graph, but by the maximum number of nodes required during reduction. As recursive expansion of functions is common in CGR, we expect that the maximum number of PEs/graph nodes required for a real application may be as high as 100K to 500K [22], even when concurrent distributed garbage collection is employed as in SPARO. The maximum size of the PE array thus can be very large. For this reason, we may consider a couple of options when implementing SPARO realistically in electronics. As suggested above, we can increase the granularity of the PEs to handle subgraphs instead of single nodes, so that 1K PEs would suffice in handling reductions. However, this would reduce the maximum parallelism that can be expressed in the graph. It would also increase the memory requirements in each processor. As mentioned earlier, the message bandwidth is not expected to change much from that in SPARO since the messages will be of greater length but they may be generated less frequently.

For purposes of solving the PE interconnection problem, we can still derive a major benefit by solving the general message passing problem that features the same bandwidth as that of the messages in SPARO. We will therefore isolate the exact processing nature in the architecture used from the specification and requirements of the network, and focus on achieving a high throughput of messages between PEs in a generic parallel processing environment that uses message passing.

### 4.3.3 SEN Schemes

Since the design and analysis of Section 3.4 precludes the possibility of efficiently implementing optical exchange switches for a large SEN, we focus on SEN designs that use optical shuffles and electronic processors and network control. The implementation options in such a case is either a serial optical or a parallel optical shuffle. We use the term parallel to include both fully parallel and quasi-parallel data transfers. This broader definition is employed since it is not certain that fully

parallel (100 bits) optical data transfers (at the board level) may be possible for a large number of processors. The actual physical size and partitioning of the PEs will dictate the level of parallelism in the message transfer. An example of such (quasi-) parallelism would be to use 25 signal lines (encoded in fewer lines or non-coded data in 25 channels) to transfer the message in four periods. Note that if the messages are sent serially on the shuffle network, it has to be operated at least 100 times or faster (for a 100 bit message) than the speed at which the PEs operate. This makes the problem of interfacing the network and the processor or processor control difficult.

We examine the serial and parallel optical shuffle implementations to note the merits and demerits in both.

### 4.3.4 Serial Optical Shuffle

The messages are assumed to be generated within the PE and stored in the output buffer (OB) (Figure 3.5). Each processor has access to two clocks: the first for the electronic processor and exchange circuitry and the second faster one to clock the OB, IB, and the diode array. It is assumed that both clocks are distributed optically on the chip as well as on the board. The faster clock is assumed to be almost 100 times faster than the slower one. In such a case, the network cycle time is as long as that of the processor. Effectively, the processor and network operate at the same speed to maximize throughput. Note that since the PE can be assumed to load the OB in parallel, there is no delay in moving the message within the PE. Thus, if the PEs are high performance processors designed to run on a clock of 50 MHz (100 MHz), the network clock must operate at 5 GHz (10 GHz). The complete network cycle is then about 20 ns (10 ns), although all switching within the network occurs with a delay of 200 ps (100 ps). To avoid synchronizing problems, the slower clock would be derived from the faster clock.

There are some obvious technical obstacles to the proposal. First, the PE chip has to integrate the high-speed buffers and the laser diode array. We require, at the least, one high-speed buffer, instead of separate OB and IB, which communicates with the optical network. While the laser diode array and the buffers are required to be implemented in GaAs, the rest of the circuitry, the processor and its interface to the external world, would be in Si (ECL) (Figure 4.6). This is essential since ECL and

bipolar have much higher levels of integration for a full-scale processor design than GaAs. Using today's integration capabilities, separate (Si and GaAs) dies can be separately optimized and integrated on a single package. While placing a single laser diode on the package is not a problem, integrating a large number of such diodes at high speeds on a single package introduces severe problems of power dissipation, thermal coupling, and electrical and optical crosstalk. The limited number of lasers that can be integrated in a package has more impact on the number of message lines (and the number of PEs) that can be put on a chip. If lower speed are used to transmit data out of the PEs, a higher degree of parallelism is possible.

While the density of lasers on the package as well as on the PWB is a problem, significant advances in laser technology, specifically in reliability, process yield, threshold current and thermal degradation, will alleviate this problem. The more serious problem, however, is one of scalability of the serial approach. As the processor technology increases its basic clock speed, for example 50 MHz to 200 MHz (in GaAs), or the messages are increased in size, the synchronous loading of messages into and out of buffers would require clock speeds of hundreds of GHzs. Such switching speeds are not possible with current or near-future technology even with GaAs. Therefore, parallel transfer of messages must be employed if good message throughput is desired. Henceforth, we will only consider the parallel transfer of messages.

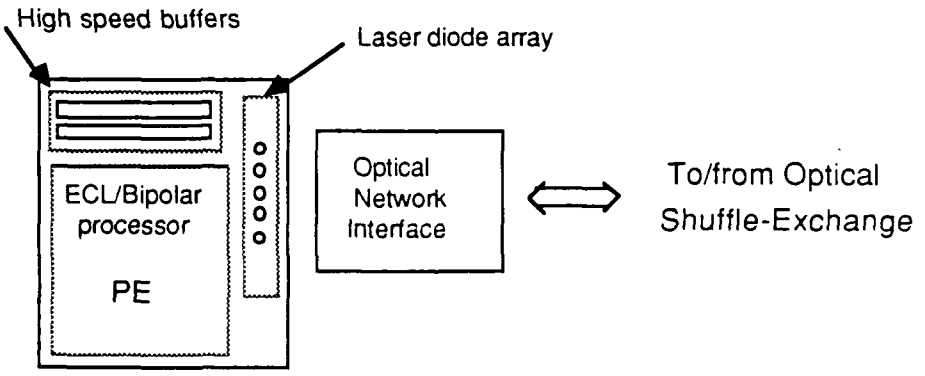



**Figure 4.6 PE and optical SEN connection**

4.3.5 Parallel Optical Shuffle

Because messages are to be transferred in parallel, the speed requirement of the optical logic is much less severe. Figure 4.5 shows the schematic layout for this SEN scheme. The PE array, most likely on multiple boards, may be connected to the shuffle network by guided or free space methods. For illustration, we consider fiber connections. (Section 6 examines the possible methods of connections.) If wavelength division multiplexing (WDM) is used to improve the parallelism in message transfer, the limit on the amount of parallelism is specified by the number laser diodes (of the required frequency) that can be integrated on-chip, the space occupied by the multiplexing and demultiplexing optics, and the required wavelength separation between adjacent channels.

Figure 4.5 shows an array of PEs connected to an array of exchange switches. In this scheme, each exchange switch contains electrooptic detectors that detect specific frequencies and polarizations of the incoming message lines. The output of the exchange modules are shuffled and fed back to the network and the PEs.

### 4.3.6 Alternate Partitioning of PEs for Parallel Shuffle

The scheme presented in Figure 4.5 appears to partition naturally in vertical slices, that is, a set of PEs on one board, a set of exchange and control logic on the same or another board, and the shuffle connection at the edge of the exchange and logic board. An alternate means of partitioning the PE array and the SEN might be more attractive in terms of implementation. Consider partitioning the complete architecture into horizontal slices, which are placed on separate boards. Each board is a slice of the architecture consisting of an even number of PEs and their corresponding exchange and control stages. A number of boards, depending on the total number of PEs in the architecture and the number of PEs that fit on a board, are connected by the shuffle connection. The advantage in this scheme is that it avoids routing the message wires between the PEs and the exchange switches across boards. The problem of designing the optical SEN is then redefined as one of distributing the shuffle across multiple boards. We will consider only this configuration in all future optical SEN designs.

The interboard shuffle will be specified by the number of channels on each board. This is determined by the off-board connection or wire density. The actual off-board wire density required is dictated by the number of PEs placed on a board. The

number of PEs placed on each board in turn is determined by the technology used to construct the board as well as the nature of board connectors used. We examine these limits in the next section.

## 5 DENSITY OF BOARD-LEVEL INTERCONNECTS

The requirements of interconnection densities depend on the interconnection network topology used. How well these density requirements can be satisfied depends on the nature of materials and devices used for implementing the interconnections. In this section we study the requirements for different topologies and how some interconnect techniques, primarily electronic, can meet them. The next section describes in greater detail the different optical techniques that can be used to support multiboard interconnection networks.

Table 5.1 shows the possible interconnect densities for different materials and technologies. Of the four technologies mentioned, only the standard edge connectors are available off-shelf. The new button board technology (using 4 mil buttons) from TRW which connects boards on the surface is the most promising high-density electronic interconnect technology possible. At present the limit on the number of I/O points on a board that can be connected using this technology is not known. The two off-board optical interconnects techniques are based on optical fibers and waveguides. Of these, the fiber optics technology is more mature. Standard fibers are usually 125 μm thick. Custom fibers could be implemented with smaller diameters, several tens of microns (20 μm without significant crosstalk). However, the yield in fabricating such fibers would be lower than for standard fibers because of the non-standard techniques required. Waveguides of dimensions less than 10 μm and 10 μm spacing provide the highest density in optical connectivity on or off-board (if alignment problems are solved). The use of waveguides, grown on polyimide substrates at Honeywell, as board level connections is still experimental. However, technology of optical waveguides is rapidly advancing. Integrated optical devices using similar technologies are now commercially available.

The different interconnect technologies can be evaluated not only on the basis of density of interconnects but also on their bandwidths. We have therefore listed the limiting speed of operations of each interconnect in Table 5.1. Complete information

on the button board operation is not available yet and is being currently compiled. The advantage in the optical techniques would be their higher available bandwidth.

| Type of board | Density/inch | Speed | Comments |
|---|---|---|---|
| PVC/Edge connector | 40 | 200 MHz | Standard/custom |
| Button Board | 170 | 150 MHz (?) | 4 mil buttons |
| Optical fibers | 2000 | Source limited | Custom |
| Polyimide waveguides | 12000 | Source limited | Experimental |

**Table 5.1. Board level interconnect density**

## 5.1 Connectivity Requirements of Some Network Topologies

The motivation for implementing the perfect shuffle in optics is driven by the needs of parallel computing. These needs were derived by a detailed examination of the I/O requirements at the board level for fine-grained processing using large scale parallelism, as evidenced in commercial machines such as the Connection Machine, a SIMD computer from Thinking Machines Corporation, and NCube, an MIMD machine from NCube Computers Inc.. We find that, not surprisingly, computer architects and system designers have made major tradeoffs in the I/O design to accommodate the limitations of electrical I/O and packaging technology. Typically, multiple (as many as 512 in case of the CM) PEs are restricted to a serial I/O line, as in the NCube, or share a serial I/O line, as in the CM. Since the communication in fine-grained processing involves packet switching, the serial communication of messages between PEs imposes a severe performance restriction. Because I/O is such a severe bottleneck, the PE performance is normally degraded and state of the art technology cannot be used. The overall effect is that the throughput of the parallel computer is limited by the I/O bandwidth at the board level.

In this section we examine the connectivity requirements for different network topologies. We can show that the wiring complexity across boards, for a multiboard system, of the shuffle-exchange is no worse than that of other interconnection architectures such as the hypercube and the crossbar when parallel message transfers are considered. In fact, the board-level connection density for SENs is less severe.

Consider, for example, a large-scale parallel architecture consisting of NM PEs distributed across M boards (or clusters, in the general case) communicating via messages. Thus each board has N PEs that need to communicate to other PEs on its board as well as others. Since we are concerned with interconnection requirements across board boundaries for different interconnection architectures, we will not consider the on-board connections that can be done by board-level routing. We will examine the total I/O channels required per board for the same density of PEs on a board for different interconnections.

### 5.1.1 Hypercube Interboard Connectivity

Examine the hypercube first. Since there are a total of NM (N and M are necessarily powers of 2) PEs, the dimension of the hypercube is $D = \log_2 NM$. Thus each PE has both input and output connections to D other PEs.

If each PE communicates a B-bit data packet or message, then each PE requires DB bits for each input or output connection. To estimate how many of the D PEs are on the same board as the source PE, we have to consider the how the PEs are partitioned.

The best case, that is, when the least number of connections are required outside the board, partition occurs when each board of the N PEs form their own smaller hypercube and D is minimum or 2. In such a case, each PE on a board is connected to $\log_2 N$ PEs on the board and to only 1 PE on the second board. In the general case when there are M boards ($M \geq 2$), each board contains N PEs in a hypercube and the number off-board unidirectional PE connections per PE in the best case is $2(\log_2 D - \log_2 N)$ or $2\log_2 M$.

Thus, the number channels required per PE in the board is $2B\log_2 M$.

The total number of channels for each board is $2NB\log_2 M$.

### 5.1.2 Crossbar Interboard Connectivity

The cross bar connection for connecting a massively parallel PE array does not really make practical sense since all PEs can talk to each other. However, for purposes of comparison and completeness, we will examine this interconnection topology. Each PE has to be connected all others. In the multiboard case, each PE has to be connected to all 2N(M -1) off-board PEs, besides the N - 1 on-board PEs.

Thus, the number (unidirectional) channels required per PE in the board is 2BN(M - 1).

The total number of channels for each board is $2BN^2(M - 1)$.

5.1.2 Shuffle-Exchange Interboard Connectivity

The computation of the I/O channels required for the shuffle-exchange is relatively simple since each PE has a fixed fanin and fanout of 1. However, in a multiboard situation the partitioning of the processors determines the number of interboard connections.

In the best case, M=2 and the 2N processors can be split up such that only N/2 PEs on each board require off-board connections to the other board. The rest of the PEs can be connected by the shuffle-exchange on-board. This is because the shuffle connection is bisymmetric (that is, the shuffle connections for the lower group of N PEs are mirror image of the connections of the upper group of N PEs) and half of each group, that is, N/2 PEs, is connected to half of the other group. This can be verified by examining the relation that describes the shuffle permutation $S(i)=(2i+\lfloor 2i/N \rfloor) \bmod N$ of the ith PE where $0 \leq i \leq N-1$.

The total board I/O required for M=2 is therefore 2BN/2 or BN since there are two connections (input and output) for each of the N/2 PEs connected to offboard PEs.

However, the SEN is not modular, so when M is increased beyond 2, each PE in the worst case partitioning may require a shuffle connection to a PE off-board. In such a case, each PE has I/O connections to two other PEs off-board, one for input and the other for output.

Thus, the worst case number channels required per PE in the board is 2B.

The total number of channels for each board in the worst case is 2BN.

| Interconnection Topology | I/O per PE | I/O per Board | Normalized I/O per Board | Board I/O (N,M,B=128,8,100) |
|---|---|---|---|---|
| Hypercube | $2B \log_2 M$ | $2NB \log_2 M$ | $\log_2 M$ | 76.8 K |
| Crossbar | $2BN(M - 1)$ | $2BN^2(M - 1)$ | $N(M - 1)$ | 22.4 M |
| SEN (M = 2) | B | BN | - | - |
| (M > 2, worst) | 2B | 2BN | 1 | 25 K |

**Table 5.2 Interboard I/O requirements for different interconnection networks assuming parallel message transfer**

Table 5.2 summarizes the I/O channel requirements for an N PE board where M boards contain a total of NM PEs. In the same table we also show the board I/O requirements for NM = 1024, M = 8 (assuming 128 PEs per board), and B = 100. Note that the columns marked as I/O per PE or I/O per board do not reflect physical fanout but rather the required connectivity. This is because, as in the hypercube operation, the PEs do not operate in a broadcast mode but rather selectively talk to individual PEs at any one time.

Figure 5.1 shows how the board-level I/O increases as a function of the number of boards for conservative values of N and B (N = 16 and B = 32). Figure 5.2 and 5.3 show graphically the total I/O requirements as a function of the number of processors for two different sets of values of message width B and the number of boards M. In each graph we have also provided two reference lines representing the total board I/O possible in two different technologies, button boards and optical fiber interconnects, assuming that a large 18" X 15" board is used. Note that since button boards have been designed for a maximum of 2000 buttons a 8" X 6" board only, we have extrapolated that figure and assumed that 5000 buttons can be placed on the larger board. In case of optical fibers, we have assumed that they are used only on one edge of the board, and not on the complete periphery like the buttons on the button board. From size and spacing considerations, 36,000 optical fibers can be fitted on the 18" side of the board. The figure is even better (216,000) if waveguide connections can be used on the edge of the board. Thus for a large number of PEs,

optical interconnects appear to hold more promise than available electronic technologies.

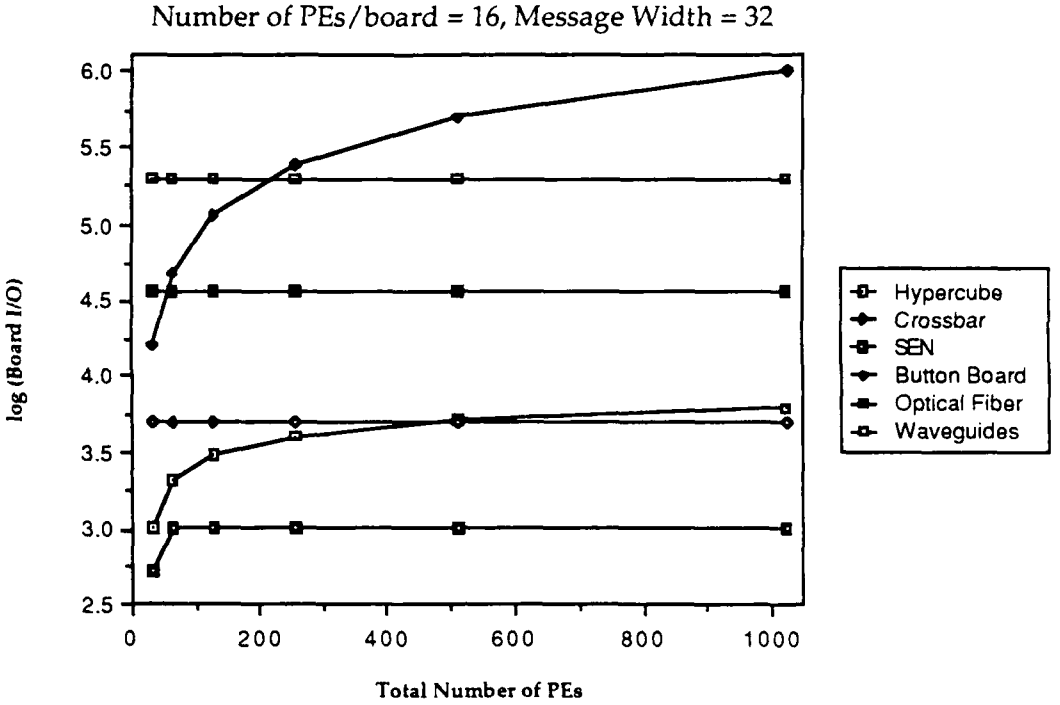



**Figure 5.1 Board-level I/O as a function of number of boards for B=32, N=16**

For reference, we have provided three (connected) points in Figure 5.3 that reflect the current and projected board I/O requirements of one 8K card rack (containing 16 cards with 512 PEs each) of the 64K CM. The lowest point (768) represents the current offboard I/O required in each board, where every 16 PEs share one serial link. The second point represents the 4K offboard connections required if each PE were allowed its own serial link. The third point represents 184K connections required if each PE were allowed 46-bit (for 4 bytes of data in a packet) parallel messages.

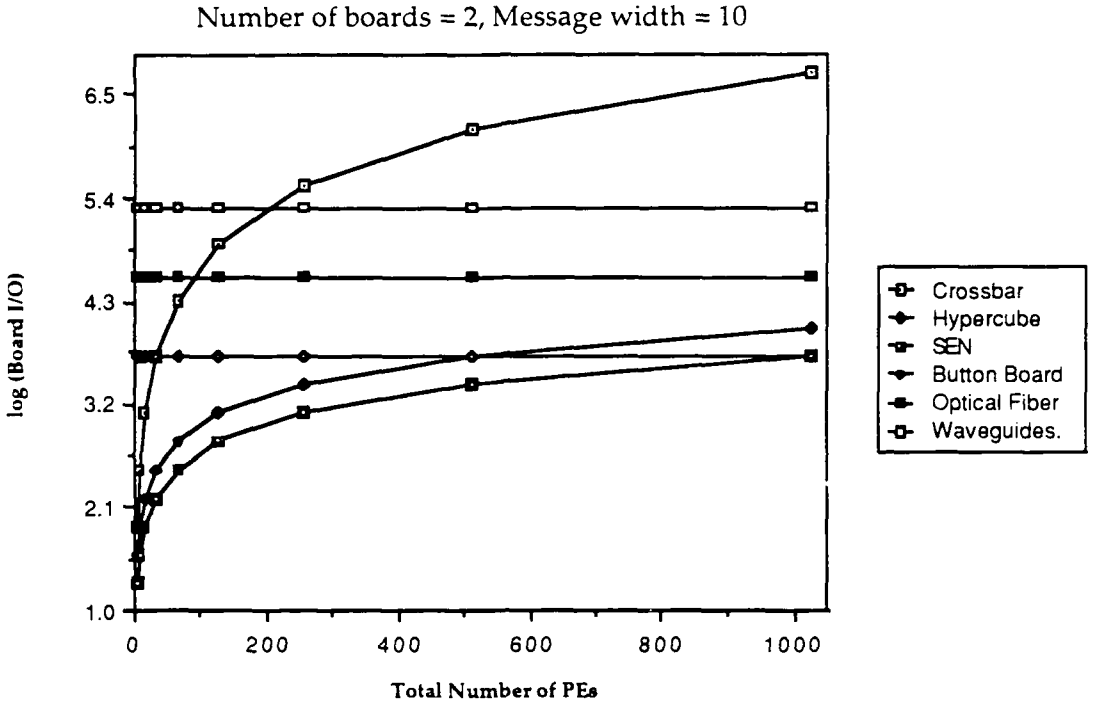



**Figure 5.2 Board-level I/O as a function of number of PEs/board for B=10, M=2**

To compare the physical limits of the different interconnect technologies, we have estimated the limits on the number of PEs that can be supported by different technologies when using a SEN. These estimates are based purely on the size and thickness of the interconnect medium.

- 400 PEs by the use of button board interconnections
- 3000 PEs by the use of optical fiber connections
- 17,000 PEs by the use of waveguides connections

When comparing the three networks, we find that the crossbar, because of the quadratic increase in the number of I/O channels is beyond practical consideration for large networks. The SEN fares better than the hypercube-an optical fiber approach can support less than 1000 PEs in a hypercube as opposed to 3000 PEs in a SEN. The higher I/O density in the hypercube is due to the larger fanout, by a factor of $\log_2 M$ over the SEN, of each PE. Further, in case of a SEN, these limits on the number of PEs per board scale only with the total number of PEs, unlike in the hypercube where the connectivity requirement of the board increases with the number of PEs as well as the number of boards. We note, in fairness, that the limited connectivity of the SEN, implies handling a smaller load, i.e., usually

around 25%. Larger loads would slow down message deliveries since more conflicts would occur. For significantly higher message traffic, replicated networks are recommended.

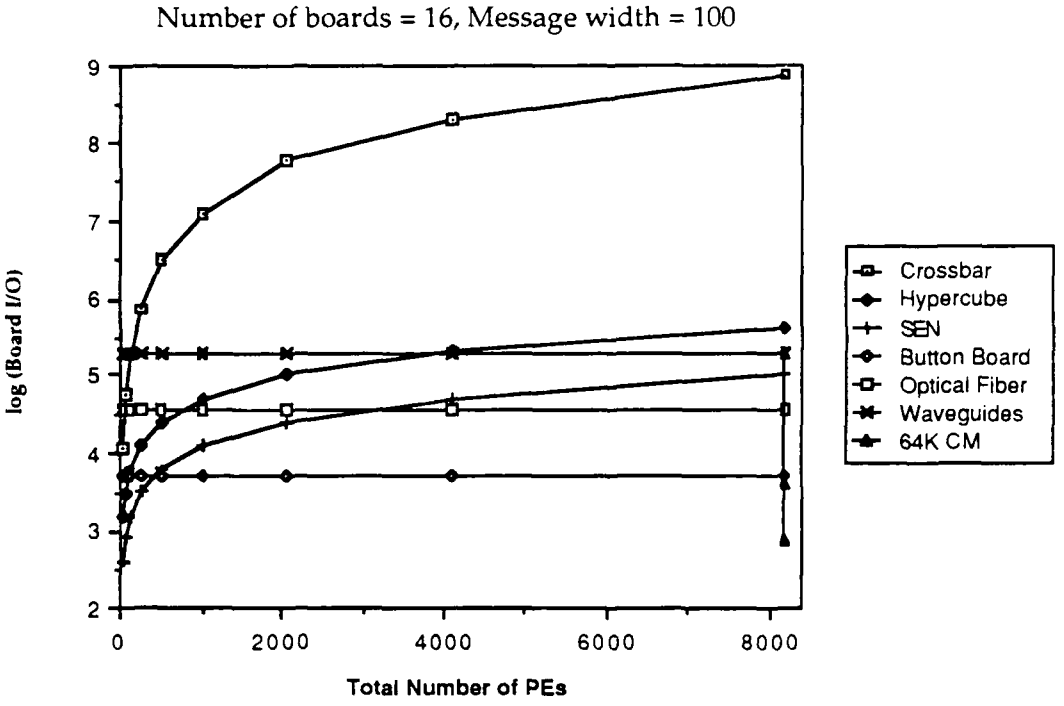



**Figure 5.3 Board-level I/O as a function of number of PEs/board for B=100, M=16**

Having examined the board I/O densities of different networks from a topological and computational perspective, we examine how different guided and free-space optical interconnect technologies can meet those requirements, and which appears most promising.

## 6. BOARD-LEVEL INTERCONNECT TECHNOLOGIES IN OPTICS

To address the demand of high I/O density at the board level that we established in the previous section, we embarked on analyzing different optical interconnect technologies as possible candidates. The key optical technologies that we investigated are:

i) Fibers

ii) Polymer Waveguides

iii) Volume Holograms
iv) Planar Holograms
v) Microoptics

To assess the relative capabilities of these technologies with the existing electrical means, we have also examined two electrical board-level interconnect approaches considered in the previous section. These are high-density conventional connectors and TRW's button boards. Before discussing the merits and demerits of the different optical approaches, we have first listed a number of issues that were used as criteria for comparing and assessing the optical approaches against the existing electrical interconnect technologies.

### 6.1 Issues for using optical interconnection

The following issues are of concern when optical interconnections are used. These considerations are important if optics is to provide a competitive edge over electronic board interconnect technologies. An evaluation of these issues provide the fundamental physical and technological limits in using optical interconnects for boards.

i) Power: Power considerations are necessary primarily for transmitters, detectors and receivers; the power consumed by the interconnect circuitry directly determines the density of optical interconnects on the board. The power budget is a function of the optical losses in the system and the power required by the drive circuitry for the sources (lasers and /or modulators) and the receivers.

ii) Size and volume of optical components: This refers to the size of optic devices (lenslets, mirrors, receivers, etc.) and the volume of hologram, if holograms are used for connecting signals at the edges of boards. The size of the optical subsystem is important since it determines the optical path length and therefore the delay in the system.

iii) Density of receivers and transmitters: The physical size and density, in case of integrated devices, determines the density of I/O connections possible. The fabrication technology also determines what densities are practically achievable.

Thermal considerations and optical and electrical crosstalk also limit the density of connections.

iv) Speed and Bandwidth: The speed and bandwidth of the receiver and transmitter determine the I/O bandwidth of the system.

v) Crosstalk: Crosstalk is important in determining signal-to-noise ratios and bit error rates in optical communications. Because of lower efficiencies in transmitting information, crosstalk is much more severe in degrading noise margins in optics than in electronics.

vi) Tolerance: The tolerance in the physical dimensions of devices and components is crucial if boards can be pulled out and reinserted into the racks. The tolerance of change in the wavelength of the signal as well as the temperature of the source is also important when ensuring correct connectivity.

vi) Reliability: The MTBF of all devices used is especially important when competing with mature electronic technologies.

vii) Cost: For practical considerations, the cost of interconnecting two boards must be close to that of electronics unless the density of optical interconnects far exceeds that possible in electronics. Cost measures for electronic interconnections are usually be expressed as $ per GHz or $ per MHz per I/O channel. In case of optical connections that are not bandwidth limited, the cost would be expressed in terms of $ per I/O channel.

### 6.2 Optical interconnect comparisons

To implement the perfect shuffle connection between PE boards optically, a variety of approaches are possible. These include volume holograms, planar holograms, waveguides, optical fibers, and microoptics. Each of the interconnect formats will be considered in turn together with their relative advantages. Table 6.1 also provides a quantitative summary of the primary features necessary to compare the different technologies.

| Interconnect Type | I/O density (SOA) /cm | I/O density (theoret) /cm | I/O density (2-yr predn) /cm | Crosstalk dB | BW-length GHz-cm | Volume | Tolerance μm | Reliability |
|---|---|---|---|---|---|---|---|---|
| Conventional Connectors | 20 | - | 32 | - | - | - | - | - |
| Button Boards | 65 | - | NA | - | - | - | - | - |
| Optical Fiber | 80 | $10^3$ *1 | $5 \times 10^2$ | 60 | $> 10^4$ | Potentially inelegant connections | ~ 10 / ~ 2 *2 (alignment) | Temperature independent medium *5 |
| Waveguides (polymer) | $10^2$ | $10^3$ | $10^3$ | 35 *3 | 60 | Planar | ~ 5 (alignment) | Temperature independent medium *5 |
| Volume Holograms | 10 | $10^2$ | $10^2$ | 20 | - *4 | Several holograms required | ~ 0.1° (λ critical) | Temperature dependent medium |
| Planar Holograms | 10 | $10^4 / 10^2$ *2 | $10^2$ | 20 | - *4 | Planar square dimensions | ~ 5 (λ critical) | Temperature dependent medium |
| Microoptics | NA | NA | NA | NA | < 60 | Bulky with vertical emittrs/recvrs | ~ 10 | - |

**Table 6.1. Comparison of optical and electrical interconnect technologies**

*1: requires non-standard manufacturing; high cost

*2: first figure for singlemode, second for multimode

*3: crosstalk mainly due to waveguide crossovers

*4: varies for each point-to-point connection; definition not applicable

*5: connections may be temperature dependent

### 6.2.1 Volume holograms

Volume holograms would appear to offer the possibility of board to board connection with high density. To most effectively use the hologram, a vertical connectivity would be employed. An array of vertically emitting sources would be mounted on the lower surface of the upper board. Immediately below it would be a volume hologram, which would serve firstly to collimate the output from each source, and to direct the signals from the sources associated with a particular exchange switch output to the relevant set of detectors. Thus the hologram is divided into a set of facets, one for each set of sources associated with an exchange switch output, and within each of these facets an array of smaller structures serving to collimate the output of each source.

Several design requirements conflict. In order to achieve high resolution of the image plane, a large hologram is required. High density in the hologram plane requires small holograms. High density in the source plane requires that the hologram plane be close, to prevent the diverging output from the source (laser or modulator) reaching the hologram associated with the adjacent channel. A study of literature evaluating volume holograms for inter-board distribution indicates that for geometries likely to be encountered in intra-board connection, densities of 100 per square cm may be attainable [28, 34].

Configurations involving multiboard connections with holograms located between boards are cumbersome, and subject to even lower densities of connectivity. Even in the two-board case, the hologram must be located accurately with respect to the source array, both positionally and with respect to angle. Although the deflection associated with each hologram facet is locally space invariant, each smaller facet also serves the purpose of collimating a source, behaving in a manner similar to a lens. Thus significant spatial misalignment of the hologram will result in angular misalignment of the beam. Smaller misalignments will result in unacceptable crosstalk. A detailed discussion on some of the tolerance issues for bulk optics that are also applicable to volume holograms is provided in Section 7.2.

Holograms used in this application impose stringent requirements on source wavelength. If the hologram is fabricated to operate at a given wavelength, the

source wavelength used must be sufficiently close to this value. Acceptable tolerances are approximately 1nm. This implies a stabilization of the source laser temperature to approximately 1°C for a simple Fabry-Perot laser and perhaps 20°C for a DFB laser. The DFB lasers are more complicated, and therefore more expensive to manufacture. LEDS are obviously unsuitable. Fabrication of vertically emitting lasers with such close tolerance has neither been demonstrated nor reported in the literature. In addition, control of individual laser wavelength within an array of uncoupled lasers is likely to be difficult, and require space-consuming circuits.

Having eliminated holograms between boards on the grounds of scalability and tolerance, one is tempted to consider the use of holograms in the backplane. Here, edge emitting structures are permissible. However, the achievable density is now only 10 per linear cm, since only one of the two orthogonal dimensions of the hologram is used.

### 6.2.2 Fiber optics

An extremely simple means of providing the optical shuffle within the interconnect medium between boards is by the use of optical fibers. Here an array of optical outputs are connected to a ribbon of optical fibers. Each ribbon corresponds to the outputs from each channel of the exchange switch, assuming no multiplexing. Thus for a total of 128 PEs on two boards, 32 ribbons would be required. High density fiber connectors are available commercially [29] with 18 fibers per connector, and the techniques used in their construction are readily scalable. However the approach is inelegant, and requires increasingly cumbersome assembly as the number of PEs increases.

The technology for fiber optics is however one of the most mature of those considered here. Fibers are low in cost, and connector technology is mature for low densities of connections, and has been demonstrated for high densities. For higher densities than are implied by the 125 mm outside diameter of most fibers, modified manufacturing technologies would be required, resulting in a higher cost. Other technologies may therefore offer lower costs per channel. A state of the art demonstration might have a density of 80 per cm. Etching of existing fibers might

yield a density of 500 per cm, while alternative fibers could be developed with very high relative refractive index differences to yield a maximum density of $10^3$ per cm.

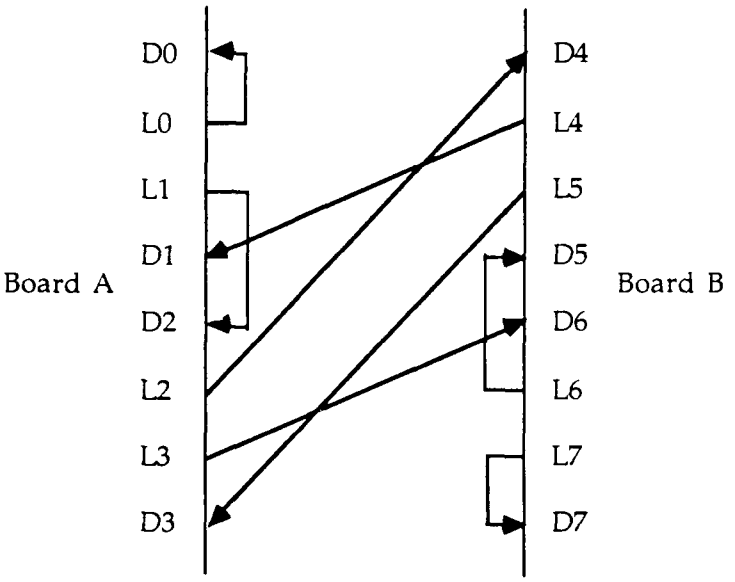



**Figure 6.1. Waveguide shuffle connection across two PWBs**
**D: detectors, L: laser sources**

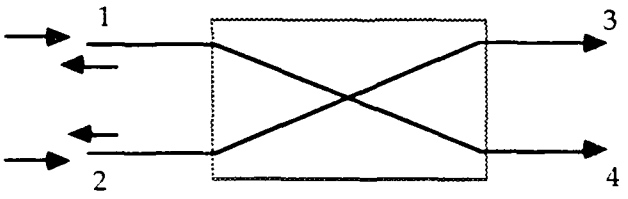
6.2.3 Polymer waveguides

These waveguides have the advantage over fiber waveguides in that the processing involves planar techniques, and the waveguides could be fabricated directly on printed circuit boards. Waveguides have been demonstrated with dimensions of approximately 10 μm in width and 4 μm in thickness. These are multimode. Adjustments to the guide dimensions are easily incorporated to increase the waveguide pitch. With such adjustments, a theoretical density of the order of $10^3$ is predicted. A state of the art demonstration would have a density of 250 per linear cm.

The crosstalk between parallel waveguides is negligibly small for waveguides with spacings comparable to the waveguide widths involved, and with the high relative refractive index differences involved when the adjacent guides are surrounded by air. Of more concern is the crosstalk associated with the intersecting waveguide junctions which will be required in the perfect shuffle implementation with waveguides on one layer. For N PEs on each of two boards, each with 100 channels,

depending on the particular configuration used, on the order of 100N crossovers will be required in the worst case. (This is proved later in Section 6.5.) Again using a value of 128 for N, and assuming worst case number of crossovers, we find that to preserve a crosstalk of -20dB at the output, each crossover must on average have a crosstalk of -60dB.

Figure 6.1 shows the shuffle connection across two boards for a network of size 8. Without making any attempt to optimize the connection layout, it is seen that two of the crossovers are associated with outgoing lines crossing lines associated with each board, while the two remaining crossovers are associated with lines between boards. Light in these intersecting junctions propagates in opposite directions. An important distinction should be made between the two types of crosstalk. In coupler terminology, a 4-port coupler illustrated in Figure 6.2 connects inputs 1 and 2 to outputs 3 and 4. In our case the ratio of the powers of the signals, P1 to P4 and P2 to P3 should be unity, while all other coefficients should be zero. Crosstalk, defined specifically as P3/P4, may arise from non-optimal designs, while a finite directivity P3/P2 may arise from a different set of imperfections. In many couplers the directivity is significantly better than the crosstalk. While sufficiently accurate data are not available for polymer waveguides, surveys of results obtained from other waveguide technologies such as high-index silica, ion-exchanged glass, or $LiNbO_3$ indicate that values of 40 or 50dB may reasonably be expected.




Ideal: P4/P1 = P3/P2 = 1

Crosstalk: P3/P4 for input into 1
P4/P3 for input into 2

Directivity: P2/P4 for input into 1
P1/P3 for input into 2

**Figure 6.2 Crosstalk across two intersecting waveguides**

Thus, the implementation of the perfect shuffle using polymer waveguides is possible, but various forms of crosstalk will limit the ultimate scalability. It should be pointed out that "engineering" techniques may result in lowering the total number of crossovers, but that no rule exists for determining the optimal layout. The situation is analogous to the CAD layout of printed circuit boards, and therefore

layout and routing algorithms from the CAD world could be profitably adapted to reduce the number of total crossovers. A simple solution is currently under consideration for the case of N PEs on each of 2 boards.

### 6.2.4 Planar holograms

An alternative approach involves the use of planar holograms. These are equivalent to modified planar waveguides, either single mode or multimode. Predicted densities are $10^2$ per square cm for the multimode planar holograms, and $10^4$ per cm for the single mode version. The geometry here differs from that of the volume holograms since planar holograms control the propagation of light from one edge of the hologram to the other. Crosstalk is predicted to be approximately -20dB at these densities. Thus, the planar holograms offer an advantage if the number of crossovers is greater than 100.

The problems associated with mechanical alignment of the holograms are not as severe as for the volume holograms since planar alignment techniques may be employed in this two-dimensional case. However, the concerns of source wavelength and its stability are still relevant.

Although an N-point to N-point distribution has not been demonstrated with this technology, reported results of a fanout demonstration indicate that a state-of-the-art demonstration might involve 10 connections [30]. The state of the art demonstration appears to be about three orders of magnitude less than the theoretical limit. Further evaluation of this technology is necessary. Issues such as the scaling of crosstalk with the number of point-to-point connections have yet to be resolved. At present this technology is not sufficiently mature to enable a cost to be associated with it.

### 6.2.5 Microoptics

Since bulk optical implementations of the perfect shuffle have been demonstrated [32], one would expect microoptic versions to be possible. Using miniature prisms and gradient index lenses ,the required mapping could be performed. Issues not yet resolved concern the effect of aberrations present in real lenses, and the input/output density attainable with this format. Microoptics does however offer

significant advantages for use in connectors between boards and backplane when used with polymer waveguides, for example.

### 6.3 Summary of optical backplane approaches

On grounds of loss, bandwidth and ultimate low-cost per channel, we infer that the most favored approach is that of the polymer waveguides. The ultimate scalability will depend upon the performance of individual crossovers and other components. Most high-quality sources under consideration are edge-emitters, while most detectors are in a planar form. The polymer waveguide approach is compatible with both, given that vertical reflecting surfaces have been reported in the literature [33].

Planar holograms deserve further investigation. Their use for the shuffle exchange, in conjunction with polymer waveguides, to transfer signals across relatively large distances may extend the scalability.

In order to demonstrate a connection scheme using such waveguides, a connector would be required between the backplane and each board. Expanded beam connectors appear to be suitable candidates, but further work would be required to design a suitable connector.

### 6.4 Sources and detectors for the optical shuffle connection

The high input/output densities described here imply the integration of many sources and detectors on one chip, rather than the assembly of discrete components. We consider first the source issue.

On grounds of yield and pwer dissipation, an approach involving a small number of discrete lasers driving a number of arrays of modulators is preferable to one involving a large, high-density array of uncoupled lasers. Issues involved in such a design are the fanout between lasers and modulators, the available power of the lasers, the effects of high power densities on the input to the waveguide fanout, loss in the fanout and the modulators, and the optical and electrical characteristics of the modulators. In either arrangement, careful design would be required to reduce electrical crosstalk and therefore optical crosstalk. Some circuitry is required to

drive the modulators -heat dissipation in such a circuit and the size of the circuit will dictate the density of the transmitters.

For logic-compatible modulators in III-V materials, devices of a few millimeter length are required. At speeds of 1 GHz or less, traveling wave configurations are not required, and the electronics drives a mainly capacitive load. The modulator design can therefore be relatively simple in comparison to modulators for telecommunication applications, since frequency chirp is not an issue in the low-dispersion short length waveguide connections.

Similarly, in the receiver array, the design of the amplifier circuits rather than the photodetector will be the limiting factor. For example in reference [31], a monolithic receiver circuit is described with a cell dimension of 75 mm x 175 mm, although the detector only occupies an area of 10 mm x 10 mm. Some reconfiguration of the layout of this circuit would enable high transverse packing densities to be achieved, up to 10mm pitch. The crosstalk which would result has not been considered. Approximately, 75 mW were dissipated in this circuit. Allowing a more conservative packing density of 200 per cm arising from a receiver pitch of 50 mm, 15W would be dissipated per linear cm, or approximately 160W per square cm. This is at the limit of what can be tolerated with the most advanced heat sinking technology. In any case, the lifetime of the devices at the resulting elevated temperatures is questionable. Tradeoffs between receiver complexity and performance may be performed. Such an optimization would include the optical performance of the backplane connection and the performance of the sources or modulators.

Our initial analysis of state of the art detectors and receivers, such as the one above, reveals three problem areas that must addressed before board-level optical interconnects can become competitive with electronic connections. These are:

i) size of the receiver circuit,
ii) crosstalk - both optical and electrical, and
iii) power budget for the receiver

Each of the size, crosstalk, and power is currently too large for implementing high-density and high bandwidth board-level interconnects.

In summary, a demonstration of a high-density interconnection cannot be performed unless high-density transmitters and receivers can be demonstrated with the required performance. We note that the problem of designing such transmitters and receivers is orthogonal to the actual approach used for accomplishing the backplane connection in optics, and that any optical approach employed must provide adequate solutions to this problem. Our analysis reveals that independent of the transmitter and detector issues, polymer waveguides and, possibly, planar holograms might be the best solution for board-level optical interconnects.

We next consider the problem of crosstalk and crossovers for the waveguide approach.

## 6.5 Crossovers in the third level interconnect for SEN and Hypercube

In considering optical interconnections for different network topologies, two different approaches to realize the connectivity between boards of PEs have been proposed: a free-space connection approach using either holograms or bulk optics, and a guided approach using waveguides in the third level interconnect. While the free-space approach is limited primarily by density of resolvable points on the board, the guided approach is limited by the amount of crosstalk that can be tolerated by the waveguide. The crosstalk results from an unavoidable number of crossovers a waveguide channel experiences in realizing the planar layout for the network connection. Here we will examine the limitations in designing a guided optical approach for the SEN and hypercube interconnection networks using polymer waveguides.

It can be seen easily that the number of crossovers in laying out any network topology varies significantly with the number of boards and the total number of PEs. Furthermore, as experience form CAD layout and routing has shown, there are many ways of minimizing the number of crossovers, but the optimal layout problem is provably NP-hard. We will therefore focus on deriving the worst case number of crossovers that can arise for an arbitrary number of PEs, N, distributed over an M boards, communicating by messages of width B bits.

### 6.5.1 Shuffle Connection

The connectivity requirements of the shuffle connection is given by the mapping S, $S(i) = (2i + \lfloor 2i/N \rfloor) \bmod N$ where $\lfloor 2i/N \rfloor$ represent the lower floor of the value 2i/N, and i ($0 \le i \le N-1$) is the index of the PE.

While there may be some heuristic techniques in partitioning the N PEs into P (=N/M) PEs in each of the M boards, we will not focus on that research problem. We will assume that contiguous pairs of PEs are placed on each board. Thus, the first board will contain the PEs 0 to (P-1), the second will contain P to (2P-1), and so on. Note that an even number of PEs is required to be placed on a board (even in case of a different partitioning) since the exchange switch required for every pair of PEs must be resident on the board.

To determine the crossovers we have to topologically represent the board on which the third level interconnects are placed. Figure 6.3 represents the placement of PEs from the M boards on 1 plane. Each row corresponds to the PE I/O connections in each board. There are thus M rows containing P PEs each. The numbering of the PEs is shown within the row: row 1 contains PEs 0 through P-1, . ., row M/2 contains PEs N/2 -P through N/2 -1, row M/2 +1 contains PEs N/2 through N/2 +P-1,. . ., and row M contains PEs N-P through N-1. The connection of the output signals of any PE is determined by its shuffle connection.

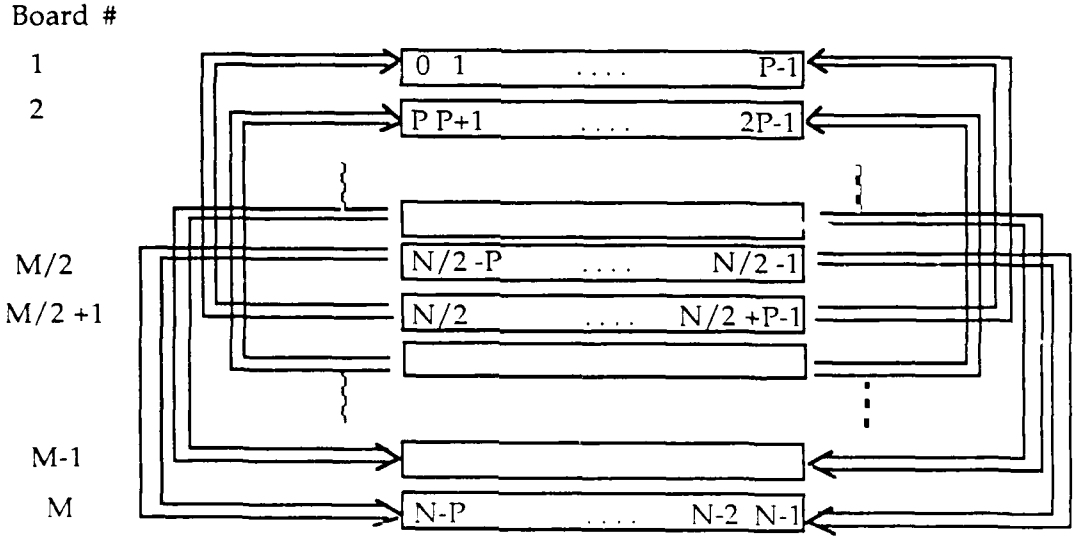



**Figure 6.3 Topological Layout for the Shuffle Connection (P = N/M)**

Using Figure 6.3 as a guide, we examine the nature of connections between different boards. Note that the output of all PEs on the first board are connected to PEs in the second board. As we go down the row to the board labeled M/2, we will notice that the output connections are to boards that are further away. Thus, PEs of board 1 connect to PEs of board 2, PEs of board 2 connect PEs of board 3 and 4, . . , and PEs of the board M/2 connect to PEs of the Mth (the last) board.

Using the symmetry of the shuffle connections we find that the output connections of the lower half boards are mirror image of the connections of the upper half boards. Thus, the output of PEs of the Mth board are connected to the M-1 th board, and the PEs of the M/2 +1 th board are connected to the first board.

The distance between connecting boards, as can be shown using the shuffle permutation, depends on the relative values of P and N. The number of crossovers of any line or channel is determined by the nature of the planar layout. To decrease the number of crossovers, we will use both sides of each row for routing the connecting channels as shown in Figure 6. To determine the number of crossovers, note that any outgoing channel from a PE (interchangeably, any incoming channel to a PE) has to possibly cross the outgoing channels of the PEs in the same row as the source, cross channels of other connections outside the rows of PEs (left or right), and the again the channels in the row of the destination PE. The worst case occurs, evident from Figure 6.3, when the source or destination PE is in row M/2 or M/2 +1. This is because PEs from these boards have to be connected to PEs furthest from the board.

To calculate the worst case number of crossovers, we consider the individual components:

Worst case number of crossovers in the source row = PB/2
Worst case number of crossovers external to the rows = (M/2 -1)PB/2 (in the lower half) + (M/2 -1)PB/2 (in the upper half) = (M/2 -1)PB
Worst case number of crossovers in the destination row = PB/2
Thus, the total number of crossovers $S_{SEN} \leq NB/2$

6.5.2 Hypercube Connection

We will use a similar approach to derive the worst case crossovers for the hypercube connection. However, we note the following differences. Each board of the hypercube connected system is its own hypercube of dimension $\log_2 P$. A PE on each board will be connected to $\log_2 M$ PEs, each on a different board.

Using the same components of crossovers as before, we determine the worst case by considering the maximum number of crossovers internal and external to a row. We obtain the following bounds.

Worst case internal crossovers in a row in the source board = $(P/2)(2\log_2 M)B$ (the factor 2 appears since unidirectional channels are assumed)
Worst case number of crossovers external to the rows = $(M-1)P/2(2\log_2 M)B$
Worst case internal crossovers in a row in the destination board = $(P/2)(2\log_2 M)B$
Thus, the total number of crossovers $S_{HYP} \leq \log_2 M(N+P)B$

Figure 6.4 shows the total number of crossovers as a function of N for both the SEN and the hypercube.

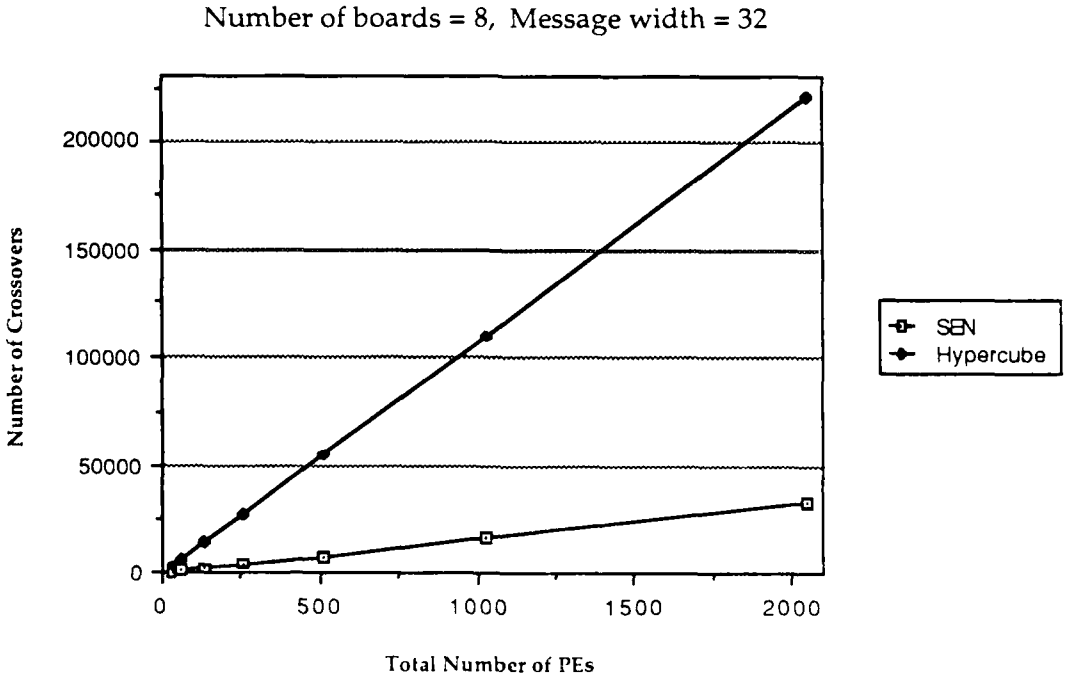



**Figure 6.4. Worst case number of crossovers for third level planar interconnects (M = 8 and B = 32)**

Thus, the number of the crossovers for the SEN and hypercube are O(N) and $O(N\log_2 M)$, respectively.

## 7. OPTICAL INTERBOARD SHUFFLE DEMONSTRATION

Based on our preliminary examination of the optical shuffle, we find at least two broad approaches possible for interconnecting multiple boards which we describe here.

In the first approach, a totally free-space approach such as a hologram (or bulk optics) is used to realize the shuffle permutation mapping of the exchange switch outputs from the edge of the boards to the input of the exchange switches also on the edge of the boards (Figure 7.1). While the limitation of this approach over the second, where the hologram is parallel to the surface, is that the number of signals (emitters) that can be accessed is significantly smaller, modifications can be made to increase the number of I/O accesses. One instance of such a modification that we are currently examining, is to use bulk imaging techniques on the surface of the board to draw out the signals to the edge and then construct the shuffle at the edge of the board as depicted schematically in Figure 7.1. The scheme presented in Figure 7.1 assumes that the shuffle is implemented on the surface of the third board using waveguides.

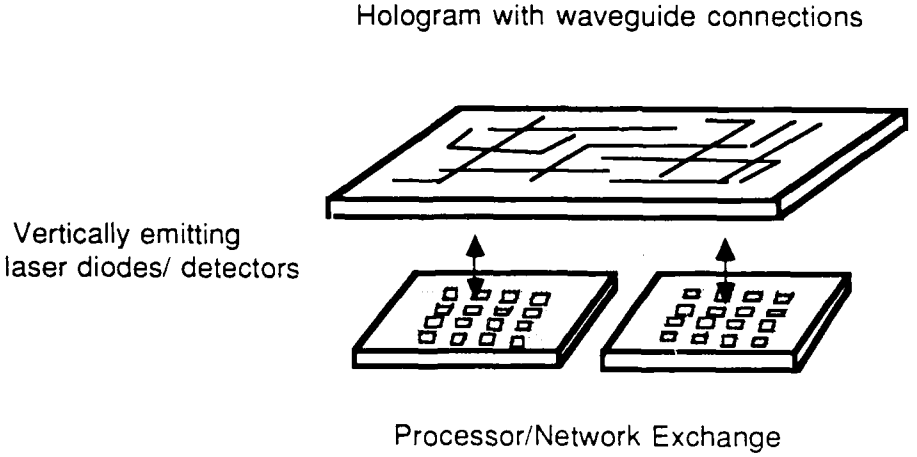



**Figure 7.1. Edge-of-the-Board Optical Shuffle**

In the second approach, the hologram (or bulk optics) can be placed parallel to the surface of the board and detects signals from vertically emitting diodes on the board (Figure 7.2). The shuffle mapping is achieved by using waveguides for routing the incoming signals to the proper point on the holographic plane. (Waveguides can also be used for realizing the backplane optical shuffle if a free-space shuffle is not desired) Although this approach is very attractive because of the larger number of signals (emitters) that can be accessed, large holograms would be required if multiple boards are necessary for the architecture. Another consideration is that the number of connection crossovers in the backplane is limited by the crosstalk in intersecting waveguides. Using holograms of larger size increases the optical path length and also increases the difficulty in implementing them practically. Therefore, this technique of designing an optical shuffle would appear impractical if the number of boards required is much greater than two. Note that the waveguide connection technique can also be applied to the edge-of-the-board connection shown in Figure 7.1. This technique is currently under investigation at Honeywell.

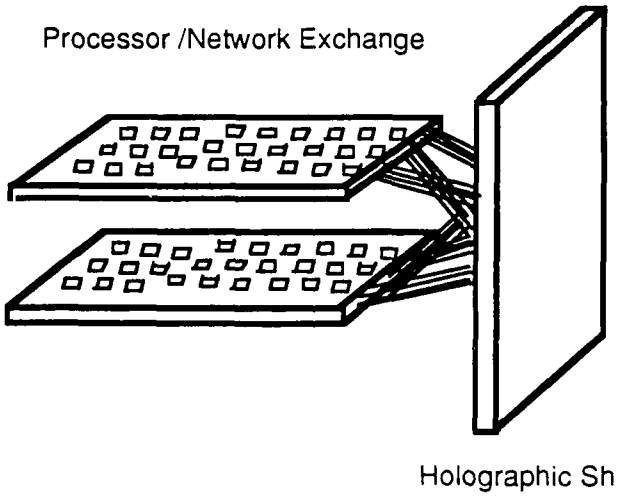



**Figure 7.2. Surface-of-the-Board Optical Shuffle**

## 7.1. Optical Connections for the Backplane

There are two types of connections that are currently under consideration for I/O connections between boards. First, a free-space approach using holograms or bulk optics. Second, a guided approach where the backplane connections are implemented using waveguides similar to wiring connections.

In either approach, the backplane connections require that the signals be directed to the edge of the board if more than two boards are to be connected. A bulk optic approach is now described.

I/O pins at the chip-level will drive LEDs or surface-emitting lasers. The optical signals from the LEDs will be collimated by a lenslet array. The collimated signals can then be directed to the edge of the board using an array of mirror strips (or, diffraction grating or hologram). Figure 7.3 shows this arrangement schematically. The optical shuffle can be performed by locating each lens in a particular position in relation to its source, or by appropriately angling each mirror strip.

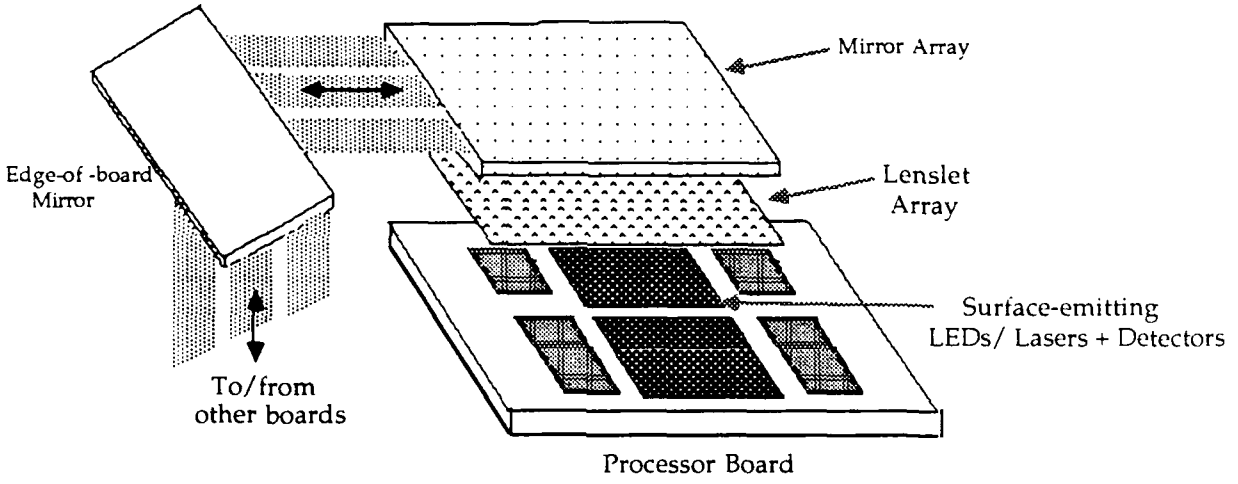



**Figure 7.3. Free-space connection for optical shuffle**

We provide some discussion on the above free-space approach, which we have examined in detail to identify the merits and demerits of the approach. The waveguide approach will be considered in detail later.

## 7.2 Free-Space Backplane Connection Using Microlenses and Bulk Optics

In this rather simple free-space connection approach, we have assumed that LEDs are used as light sources, the lenslet array aligned with the LEDs on board, and LEDs for different messages placed adjacent to each other.

To examine the scheme in more detail, one must examine the geometry of Figures 7.3 and 7.4. The lenslet array is located in the upper plane, while the PWB containing the LEDs is placed in the lower plane. The lenslet array generates the Fourier Transform of the light from each LED, so that a group of collimated (within the limits imposed by the finite source dimensions of the LED) beams is incident on the mirrors. A prism or grating can then be employed to change the direction of each beam according to its shuffle connection. If the separate bits of each word for a given processor node are aligned along the length L (see Figure 7.4) of the PWB and the lenslet array, then they can be imaged in that direction (no angular shift along the L-direction). By using a single element (prism or grating), all bits of the word can be mapped in parallel for each shuffle connection. We must note, however, that bulk optical components are expensive and such a scheme for the implementation of the shuffle would be labor-intensive.

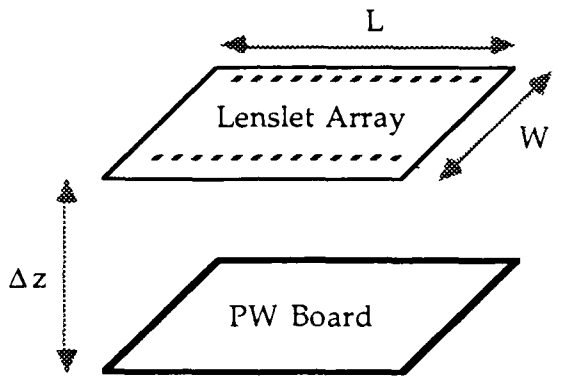



**Figure 7.4. Lenslet array and PW Board configuration**

The necessary tolerances in this scheme can be readily derived. If the emitted beam of the LED has a FWHM of 16° (a typical HP device from the catalog) then the diameter (d) of the beam at a given distance D from the LED is given as $2D\tan(8°)$. This is about D/4. The diameter d of a lenslet at a distance $\Delta z$ from the LED must be $\Delta z/4$. This means that the total length of the array must be $25\Delta z$ to accommodate 100 bits of parallel connections. In general terms of the FWHM, $\theta$, and N, the number of connections in one dimension, the lenslet-board distance is

$$\Delta z = (L/2N)/[\tan(\theta/2)]$$

The diameter of the lenslet d is given by

$$d = 2\Delta z \tan(\theta/2)$$

We can estimate the $\Delta z$ and d for some expected values of N.

(a) $N^2 = 10^4$, L = 10 cm, $\theta/2 = 8°$: $\Delta z = 0.36$ cm, d = 0.1 cm
(b) $N^2 = 10^4$, L = 1 cm, $\theta/2 = 8°$: $\Delta z = 0.36$ mm, d = 100 μm

It is the minimum separation and lens diameter that will determine the number of connections possible. Thus, for a given a minimum separation between the lenslet array and the board, the total number of connections possible in one dimension is L/d.

The number of nodes which can be fit into the other dimension of the array is similarly limited by the diameter of the lenslet and the width of the array:

$$n = W/d = W/2\Delta z \tan(\theta/2)$$

Similar tolerance limits apply to the positions of the cards in the sockets and to all mirrors in the system. The effects of the simple misalignments in the insertion of the PWB or card can be expressed in terms of angular displacements. These angular tolerances result from shifts in the position of the detector, the source, or the mirror. The requirements on angular alignment are found to be quite severe. The angular tolerances become less severe for mirrors closer to the detectors.

We now calculate the angular tolerance, a, and the positional tolerances, $\Delta p$, of the source or detector positions using the simple geometry shown in Figure 7.5b.

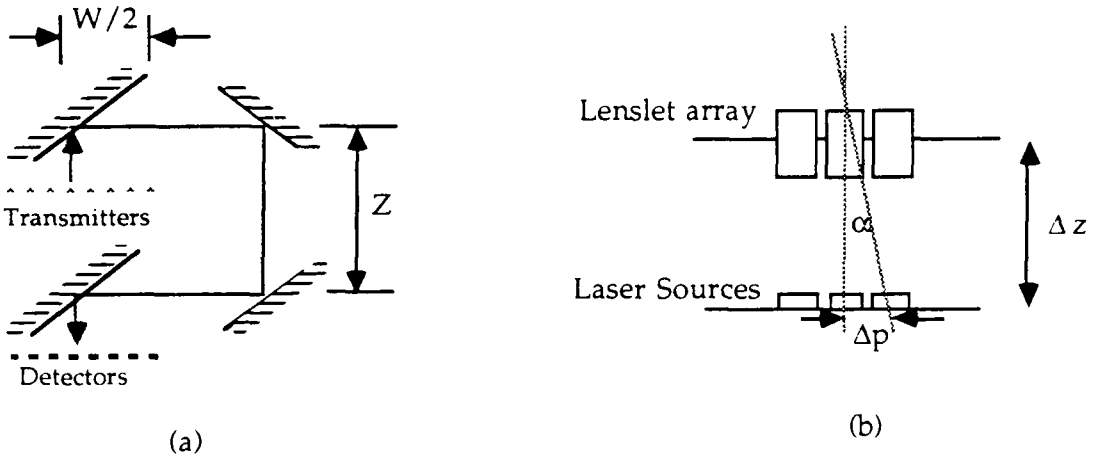



**Figure 7.5 a) Source to detector optical path. b) Source position tolerance**

Let the average total distance traveled by light from a source to its corresponding detector be T. Then T is the sum of the board width and the separation Z between the upper and lower planes, as shown in Figure 7.5a, or $T \approx W + Z$. Usually, $W \approx L$ and $Z \leq W$, so that $T \approx 2L$.

The angular tolerance $\alpha$, the angle at which a signal intended for a given detector reaches the adjacent detector. Since the separation between detectors is equal to d, the diameter of the lenslet that collimates signals from the source,

$$\tan\alpha = d/T = d/2L$$

Examining the values of L and d used in the cases (a) and (b), the tolerance in the angle is:
$L = 10$ cm, $d = 0.1$ cm, $\tan\alpha \leq 0.1/20.0$, or $\alpha = 0.29°$

We can now calculate the positional tolerance $\Delta p$ using Figure 7.5b.

$$\Delta p = \Delta z \tan\alpha$$

For cases (a) and (b), the tolerance in the source and detector position is found to be:
(a) $\Delta z = 0.36$ cm, $\tan\alpha = 0.005$, $\Delta p = 18$ μm
(b) $\Delta z = 0.036$ cm, $\tan\alpha = 0.005$, $\Delta p = 1.8$ μm

Clearly, the positional tolerances for the source, the detector and the mirror is too small for this approach to be practical unless only a small number of connections are required between the two PWBs.

Another issue to be considered is the spacing Z between the boards. If the lenslet array can be placed very close to the LEDs ($\Delta z \approx 0$), this spacing is determined by the height of the mirror from the lower board. The simplest way of directing the signals off the board to the edge is to use a mirror 45° with respect to the (lower) board. If a single mirror is used, then the spacing between the boards is as large as the width of the board W. One could reduce the angle between the board and the mirror but a lens or prism would be required to shift the reflected signals off the mirror so that they reach beyond the edge of the board. The optimum solution for reducing the spacing is to have rows of mirror strips at a minimum distance from the board. However, these mirror strips have to be aligned to tolerances of less than 0.1° as previously described.

To summarize, there are three serious problems in using either a bulk optic or a volume holographic approach to implement the shuffle network between PWBs. First, because of the requirement of realizing the shuffle connections at the edge of the board, all optical signals have to be moved to the edge of the board using mirrors that occupy a significant physical space. Second, because of the need of mirrors at angles close to 45° to the PWBs, a large optical path (equal to twice the width of the PWB) delay is introduced. Third, in case of high-density connections, the tolerance in the position of the sources, the detectors, and the mirrors become very severe and impractical.

The conclusions of this exercise in using volume holograms for high-density network interconnections further motivates the guided interconnection approach using waveguides proposed in the previous discussion on the comparison of optical interconnect technologies in Section 6.2.

## 7.3. Possible Electronic Setup for Optical Shuffle Demonstration

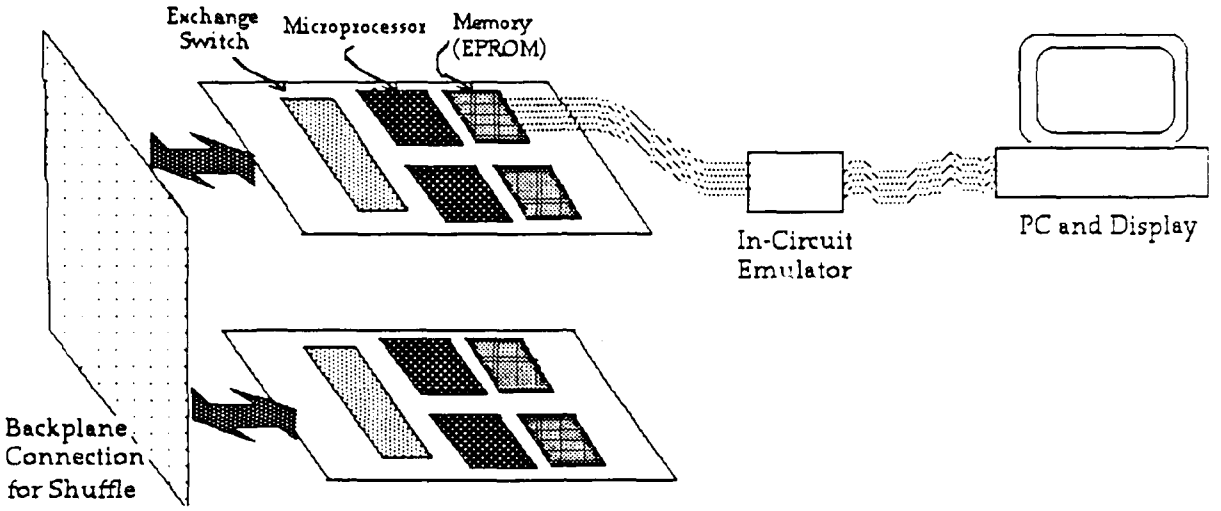



**Figure 7.6. Electronic setup for demonstrating optical backplane shuffle connection**

This section outlines a possible setup that can be used to demonstrate the backplane optical shuffle for a multiboard processing system (Figure 7.6). The setup can be built around customized two-processor boards that constitute an MIMD processor. Four microprocessors are split among two boards. The four processors on the two boards can be connected by a shuffle-exchange network. For the demonstration, the exchange can be done by a custom smart exchange switch (one for every pair of processors) while the shuffle connection would be in optics. This experimental vehicle would serve to demonstrate the feasibility of an interboard parallel shuffle connection. The shuffle connection would be achieved using either guided approach (polymer waveguides) or a free-space approach (planar holograms) where bulk optics is used to direction signals into and out of boards.

### 7.3.1 Electronic Setup

The basic electronic setup is built around some standard microprocessor, such as the Motorola 680XX or an Intel 386 running some application program. In the basic configuration, a 4-way shuffle can be demonstrated using messages that are as wide as the data path available on the chosen microprocessor. Typically, this would be 32 bits or some multiple of bytes.

The purpose of the electronic segment is to emulate fine-grained parallel processing. Since a distributed application would be executed on the computing system, each processor (PE) would access its program data and control from a dedicated memory chip, either a RAM or an EPROM. Data for the application would be transferred between PEs through the shuffle connection between the board. The initial data would be provided by the PC through the In-Circuit Emulator (ICE). The final output data would be also displayed on the PC monitor.

The network interface and exchange would be lumped into a customized Exchange Switch, built from discrete gates or a PLA.

### 7.3.2 Application

The application selected to be executed on the PEs would be one that can be executed in purely distributed fashion among the 4 PEs. The essential requirement for the demonstration is the handling of high message traffic on the shuffle network. Fine-grained applications which are typical in generating high message rates cannot be considered since only a few PEs are used. We would therefore consider a relatively simple number-crunching problem that can be executed in parallel: a parallel sort algorithm on a sequence of numbers. Since the compare and exchange operation is very simple, each PE can quickly compare numbers and send them, as message packets, to other PEs as the sorting proceeds. The final result, a list of sorted numbers can be easily displayed on the attached PC display. Thus, each PE is microcoded for sorting numbers for the parallel algorithm as well as for executing certain necessary control operations, such as initial loading (of data provided by the PC) and final despatching of data.

An in-circuit emulator (ICE) would be used for communicating data between a master PE that receives the initial list of unsorted and the final list of sorted data. The ICE would also be used to test the software that will be executed in each PE.

## 8. CONCLUSIONS

Our performance analysis of SPARO revealed that the implementation of the graph reduction processors is more feasible in high-speed GaAs or even bipolar technology rather than in opto-electronics. Using GaAs processors would facilitate the construction of the interface to the optical network. Since the bottleneck in the parallel reduction of combinator graphs is in the communication between the processors, a high-speed optical network can lead to much greater performance than obtainable in electronics.

Despite the complex issues in implementing an optical SEN, the major advantage of such an implementation is clearly in the potential increase in the overall message bandwidth. We have therefore examined the detailed issues necessary to construct the optical network.

The central problem in the all-optical SEN is the design of the smart exchange switch which can handle conflict resolution, message rerouting and delivery. Based on the examination of a number of optical techniques for realizing the basic exchange switch, a polarization encoding approach looks to be the most promising in terms of speed. However, generalizing this basic switch into a smart one that handles the conflict, delivery, etc. requires an optical method of representing and modifying the age of the message circulating in the network. Because of the space overhead that results in constructing the more complex smart switch, optical switching in the exchange elements of the SEN is currently not feasible.

There is still a great speed advantage in implementing the perfect shuffle compactly in optics. In the case of an electronic implementation of the SEN, because of the non-modular layout involved, large fanout devices and long delay paths are expected to degrade the cycle time of the network. The bottleneck is found to be not in the switching elements but in the parallel transfer of data. Based on our detailed analysis of large electronic SENs and their requirements, we find that a high-density and high-bandwidth SEN is necessary for highly parallel processing. Because of the current limitations of electronics, we believe that optics can be an advantage in implementing the interconnection network. From an evaluation of possible optical interconnect technologies, we have determined that polymer

waveguides appear to be the best choice. Theoretical interconnection densities also make planar holograms a suitable choice. However, state of the art demonstrations are two orders of magnitude poorer than the theoretical limit. Current experience with polymer waveguides and the problems of tolerances in free-space optical interconnections lead us to believe that waveguide connections appear most feasible. Orthogonal to the interconnection technology, we have to develop methods for integrating high-density optical transmitters and receivers that are critical in realizing high-density interconnections.

The tasks scheduled for the coming year are to design and demonstrate an effective method for accomplishing large perfect shuffles for parallel message passing in multiprocessor system consisting of a large number of PEs distributed across multiple boards. We plan to use polymer waveguides to implement the shuffle on the backplane.

## REFERENCES

[1] M. Derstine, A. Guha, and S. Natarajan, 'A Conceptual Design of an Optical Symbolic Computer for Combinator Graph Reduction,' to be submitted to the Journal of Optical Engineering.

[2] S. Abraham and K. Padmanabhan, 'Performance of the Direct Binary n-Cube Network for Multiprocessors,' Proc. International Conference on Parallel Processing, 1986, pp. 636 - 639.

[3] T. Kushner, et al, 'Image Processing on the ZMOB,' IEEE Transactions on Computers, October 1982, pp. 943 - 951.

[4] R. Barry and K.M. Chandy, ' Performance Models of Token Ring Local Area Networks,' SIGMETRICS 1983, pp. 266 - 274.

[5] D.H. Lawrie and D.A. Padua, 'Analysis of Message Switching with Shuffle-Exchanges in Multiprocessors,' Proc. of the Workshop on Interconnection Networks for Parallel and Distributed Processing, 1980, pp. 116 - 123.

[6] T. Lang, 'Interconnections Between Processors and Memory Modules Using the Shuffle-Exchange Network,' IEEE Transactions on Computers, May 1976, pp. 496 - 503.

[7] C.P. Kruskal and M. Snir, 'The Performance of Multistage Interconnection Networks for Multiprocessors,' IEEE Transactions on Computers, December 1983, pp. 1091 - 1098.

[8] J.H. Patel, 'Performance of Processor-Memory Interconnections for Multiprocessors,' IEEE Transactions on Computers, October 1981, pp. 771 - 780.

[9] D.M. Dias and J.R. Jump, 'Analysis and Simulation of Buffered Delta Networks,' IEEE Transactions on Computers, April 1981, pp. 273 - 282,

[10] D.H. Lawrie, 'Access and Alignment of Data in an Array Processor,' IEEE Transactions on Computers, December 1975, pp. 1145 - 1155.

[11] W.D. Hillis, 'The Connection Machine,' MIT Press, 1985.

[12] A.W. Lohmann, 'What Classical Optics can do for the Digital Optical Computer,' and 'Optical Perfect Shuffle,' Applied Optics, 15 May 1986, Vol. 25, No. 10.

[13] C.W. Stirk, R.A. Athale, C.B. Friedlander, private communication.

[14] G. Eichmann and Yao Li, 'Compact Optical Generalized Perfect Shuffle,' Applied Optics, 1 April 1987, Vol. 26, No. 7.

[15] J.E. Midwinter, 'Novel Approach to the Design of Optically Activated Wideband Switching Matrices,' IEE Proceedings, Vol. 134, Pt. J, No. 5, October 1987; "Light' Electronics, myth or reality', IEE Proceedings, Vol. 132, Pt. J, No. 6, December 1985.

[16] N. Weste and K. Eshraghian, Editors, 'Principles of CMOS VLSI Design,' Addison-Wesley, 1985.

[17] Y. J. Hsu, et al, 'A Gate-Array Design of a Shuffle-Exchange Network Switching Element,' 6th Biennial University/Government/Industry Microelectronics Symposium, 1985.

[18] J. Shamir, H. J. Caulfield, W. Micelli, and R. J. Seymour, 'Optical Computing and the Fredkin Gates,' Applied Optics, 15 May 1986.

[19] R. Cuykendall and D. McMillin, 'Control-specific Optical Fredkin Circuits,' Applied Optics, 15 May 1987

[20] E. Fredkin and T. Toffoli, 'Conservative Logic,' International Journal of Theoretical Physics, Vol. 21, Nos 3/4, 1982, pp. 219 - 253.

[21] S. C. Esener, 'One-dimensional Silicon/PLZT Spatial Light Modulators,' Optical Engineering, May 1987, Vol. 26, No. 5.

[22] M. Scheevel, 'NORMA: A Graph Reduction Processor,' 1986 ACM Conference on LISP and Functional Programming, August 1986, pp. 212 - 219. Private Communication.

[23] D.H. Hartman, 'Digital high speed interconnects: a study of the optical alternative,' Optical Engineering, October 1986, Vol. 25, No. 10, pp. 1086 - 1102.

[24] Clark, N. A., M. R. Meadows, M. A. Handschy and K. M. Johnson, "Optical Interconnections Using Ferroelectric Liquid Crystals," OSA Annual Meeting, Technical Digest, Rochester, NY, p. 119, Oct. 1987.

[25] McManus, J. B., R. S. Putnam and H. J. Caulfield, "Demonstration of Switched Holograms for Optical Interconnection", OSA Annual Meeting, Technical Digest, Rochester, NY, p. 120, Oct. 1987.

[26] A. Huang, 'Optical Digital Computers - Devices and Architectures,' private communication.

[27] R. Smolley, 'Button Board: A New Technology Interconnect for 2 and 3 Dimensional Packaging,' Proceedings of the 1985 International Symposium on Microelectronics, pp. 326 - 333.

[28] R. Kostuk, University of Arizona, private communication.

[29] AT&T, commercial literature for multifiber array connector (MAC), Spring 1988.

[30] T. Jannson, Physical Optics Corporation, private communications.

[31] C. Harder, et al, '5.2 GHz Bandwidth Monolithic GaAs Optoelectronic Receiver,' IEEE Electron Devices Letters, Vol. 9, No. 8, April 1988.

[32] K.H. Brenner, et al, 'Digital Optical Computing with Symbolic Substitution,' Applied Optics, Vol. 25, No. 18, Sept. 1986.

[33] R. Selvaraj, et al, 'Integrated Optical Waveguides in Polyimide for Wafer Scale Integration,' Journal of Lightwave Technology, Vol. 6, No. 6, July 1988.

[34] R. Kostuk, Doctoral Dissertation, Stanford University, 1986.

END

DATE

FILMED

DTIC

10-88