An Optical Multi-Mesh Hypercube: A Scalable Optical Interconnection Network for Massively Parallel Computing

Ahmed Louri *Member, IEEE,* and Hongki Sung, *Student Member, IEEE*

***Abstract*—A new interconnection network for massively parallel computing is introduced. This network is called an Optical Multi-Mesh Hypercube (OMMH) network. The OMMH integrates positive features of both hypercube (small diameter, high connectivity, symmetry, simple control and routing, fault tolerance, etc.) and mesh (constant node degree and scalability) topologies and at the same time circumvents their limitations (e.g., the lack of scalability of hypercubes, and the large diameter of meshes). The OMMH can maintain a constant node degree regardless of the increase in the network size. In addition, the flexibility of the OMMH network makes it well suited for optical implementations. This paper presents the OMMH topology, analyzes its architectural properties and potentials for massively parallel computing, and compares it to the hypercube. Moreover, it also presents a three-dimensional optical design methodology based on free-space optics. The proposed optical implementation has totally space-invariant connection patterns at every node, which enables the OMMH to be highly amenable to optical implementation using simple and efficient large space-bandwidth product space-invariant optical elements.**

***Index Terms*—Hypercube, interconnection network, optical interconnect, parallel computing, scalability, space-invariance.**

I. Introduction

It has become very clear that significant improvements in computer performance in the future can only be achieved through exploitation of parallelism at all machine design levels [1]. On the architectural side, communication among the elements of a high-performance computing system is recognized as the limiting and decisive factor in determining the performance and cost of the system [2], [3]. In recent years, there have been considerable efforts in the design of interconnection networks for parallel computers. Two of the most popular point-to-point interconnection networks for parallel computers today are the binary n-cube, also called the hypercube, and the mesh interconnection networks. Several companies, including NCUBE, Connection Machine Inc., FPS, Intel, and Ametek, are currently selling parallel machines based on the hypercube topology [1]. In a binary n-cube we have $N = 2^n$ nodes each of degree n, where the degree of a node means the number of nodes directly connected to it. A node in this paper could be a processing element (PE), a memory unit, or a switch. The attractiveness of the hypercube topology is its small diameter, which is the maximum number of links (or hops) a message has to travel to reach its final destination between any two nodes. For a binary n-cube network the diameter is identical to the degree of a node $n = \log_2 N$. Each node is numbered in such a way that there is a one binary bit difference between any node and its $\log_2 N$ neighbors that are directly connected to it. This property greatly facilitates the routing of messages through the network. In addition, the regular and symmetric nature of the network provides fault tolerance.

However, a major drawback of the hypercube network is its lack of scalability, which limits its use in building large size systems out of small size systems with little changes in the configuration. Among important parameters of an interconnection network of a multicomputer system are its scalability and modularity [2], [3]. Scalable networks have the property that the size of the system (e.g., the number of communicating nodes) can be increased with minor or no change in the existing configuration. Also, the increase in system size is expected to result in an increase in performance to the extent of the increase in size. As the dimension of the hypercube is increased by one, one more link needs to be added to every node in the network. In addition to the changes in the node configuration, at least a doubling of the size is required for the regular hypercube network to expand and to remain as a hypercube.

The second interconnection network that has been extensively studied is the mesh. Mesh networks are easily implemented because of the simple regular connection and small number of links (four) per node. Due to the constant node degree, the mesh network is highly scalable. With a network size of N nodes, the minimal incremental size is approximately $N^{1/2}$ for the perfectly balanced network. However, the mesh network also suffers from a major limitation which is its large diameter ($N^{1/2}$ for an N-node network). Moreover, a relatively small portion of algorithms for scientific and engineering problems efficiently fits the mesh topology.

On the technological side, optics, owing to its inherent parallelism, high spectral and spatial bandwidth, and low signal crosstalk, possesses the potential for a better solution to the communication problem in parallel and distributed computing [4]–[8]. Recent studies have shown that free-space optical interconnects provide far better communication bandwidth and power dissipation for sufficiently long connection

Manuscript received January 18, 1993; revised January 14, 1994. This work was supported in part by NSF Grant Number MIP 9113688, in part by NSF Grant Number MIP 9310082, and in part by a grant from US West.

The authors are with the Department of Electrical and Computer Engineering, The University of Arizona, Tucson, AZ 85721.

IEEE Log Number 9400261.

0733–8724/94$04.00 © 1994 IEEE

paths than possible with VLSI technology [9], [10]. There have already been considerable efforts in designing optical interconnection networks [5], [7], [11]–[16]. However, optical implementations of these networks often require the use of space-variant optics, which often results in low interconnection densities, and requires complex optical (active) components [17]. The degree of space-variance determines the complexity and regularity of an interconnection network [17], [7]. A totally space-invariant system has a very regular structure where all the nodes have the same connection patterns which consequently lower the design complexity. There is a fundamental trade-off between the space-bandwidth product (SBWP), the total degree of freedom in an optical interconnect (the space is considered the cross section area and the bandwidth is the highest spatial frequency handled by the system), and the degree of space-variance. A totally space-invariant system has minimal SBWP requirements, whereas a totally space-variant system has extensive SBWP requirements. Also, totally space-invariant systems are much easier to implement than totally space-variant systems.

Motivated by these limitations, we have explored a novel network topology, called *Optical Multi-Mesh Hypercube* (OMMH), which combines advantages of both the hypercube (small diameter, high connectivity, symmetry, simple control and routing, fault tolerance, etc.) and the mesh (constant node degree and scalability) topologies, while circumventing their disadvantages (lack of scalability of the hypercube, and large diameter of the mesh). We have also developed a three-dimensional (3-D) optical implementation for the OMMH. The distinctive advantages of the proposed design methodology include: 1) an efficient and scalable interconnection network, 2) better utilization of the SBWP of optical imaging systems, 3) full exploitation of the parallelism of free-space optics, 4) simple optical implementations because of the use of large SBWP space-invariant optical elements, 5) cost-efficient implementations because the beams which will be directed orthogonal to the device plane would share the same set of imaging optics for interconnects, and consequently, the cost of the optical hardware would be shared by a large amount of communicating elements, and 6) compatibility with the emerging two-dimensional (2-D) optical logic and switching, and opto-electronic integrated circuit (OEIC) technologies.

The rest of the paper is organized as follows. Section II introduces the OMMH network and its architectural properties. Section III presents the proposed (3-D) optical implementation methodology. Section IV describes possible optical hardware and settings of the physical implementation of the OMMH network. Section V concludes the paper.

II. Optical Multi-Mesh Hypercube Networks

A. Definition of OMMH Network

An OMMH is characterized by a triplet (l, m, n), where l represents the row dimension of a four-nearest-neighbor-connected mesh, m the column dimension of the mesh, and n the dimension of a binary hypercube. The total number of nodes in (l, m, n)-OMMH is $l \times m \times 2^n$. An address of a node consists of three components: (i, j, k), where $0 \leq i < l, 0 \leq j < m$, $0 \leq k < 2^n$, and i, j, k are integers. The first two components, i and j, represent the address of the node in a mesh, and the last component, k, represents the address of the node in a hypercube. Connection rules of the (l, m, n)-OMMH, for two nodes (i_1, j_1, k_1) and (i_2, j_2, k_2), are as follows:

Rule 1 There is a link, called *a hypercube link*, between two nodes if and only if (1) $i_1 = i_2$, and (2) $j_1 = j_2$, and (3) k_1 and k_2 differ by one bit position in their binary representation (Hamming distance of one). Connection rule 1 generates $l \times m$ hypercubes with dimension n and these hypercubes are separated from each other until the following connection rule is applied.

Rule 2 A link, called a *mesh link*, exists between two nodes if and only if (1) $k_1 = k_2$ and (2) two components, i and j, differ by one in one component while the other component is identical. This rule generates 2^n meshes with dimension $l \times m$. If we neglect hypercube links made by rule 1, the meshes generated by rule 2 are also separated from each other. The combination of both rule 1 and rule 2 connects hypercubes and meshes such that $l \times m$ nodes (one node from one hypercube) in the same positions of $l \times m$ hypercubes are linked together to form a mesh with dimension $l \times m$.

From the above connection rules, the interconnection functions [18], denoted by *ommh* (i, j, k) where i, j, k are three address components of a node, of the (l, m, n)-OMMH network with the wrap-around mesh can be described as follows:

- $ommh_{m_1}(i, j, k) = ((i + 1) \text{mod } l, j, k)$
- $ommh_{m_2}(i, j, k) = ((l + i - 1) \text{mod } l, j, k)$
- $ommh_{m_3}(i, j, k) = (i, (j + 1) \text{mod } m, k)$
- $ommh_{m_4}(i, j, k) = (i, (m + j - 1) \text{mod } m, k)$
- $ommh \quad {}_{c_d}(i, j, k_{n-1} \cdots \quad k_{d+1}k_d k_{d-1} \cdots k_0) = (i, j, k_{n-1} \cdots k_{d+1}\overline{k}_d k_{d-1} \cdots k_0)$, for $d = 0, 1, \cdots, n-1$, where$k_{n-1} \cdots k_{d+1}k_d k_{d-1} \cdots k_0$is a binary representation of integer k.

The first four interconnection functions, $ommh_{m_1}$, $ommh_{m_2}$, $ommh\ _{m_3}$, and $ommh_{m_4}$, are for the four-nearest-neighbor connections including wrap-around connections and $ommh_{c_d}$, for $d = 0, 1, \cdots, n-1$, determines the hypercube interconnection.

Fig. 1 shows a $(4, 4, 3)$-OMMH interconnection where solid lines represent hypercube links and dashed lines represent mesh links. Small black circles represent nodes of the OMMH network which are, in this paper, abstractions of processing elements or memory modules or switches. Both ends of mesh links, dashed lines, are connected for wrap-around connections of the mesh if they have the same labels. The size of the OMMH can grow without altering the number of links per node by expanding the size of the mesh; for example, by adding three cubes on the perimeter of the mesh in Fig. 1. This feature allows the OMMH to be scalable. More discussion on the scalability issue will follow in Section II-C. A $(4, 4, 3)$-OMMH consists of $4 \times 4 \times 2^3 = 128$ nodes. It can be viewed as eight concurrent meshes where eight nodes having identical mesh addresses form one three-cube. Alternatively, it can be viewed as 16 concurrent three-cubes in which 16 nodes having identical hypercube addresses form a 4×4 mesh. The $(4, 4, 3)$-

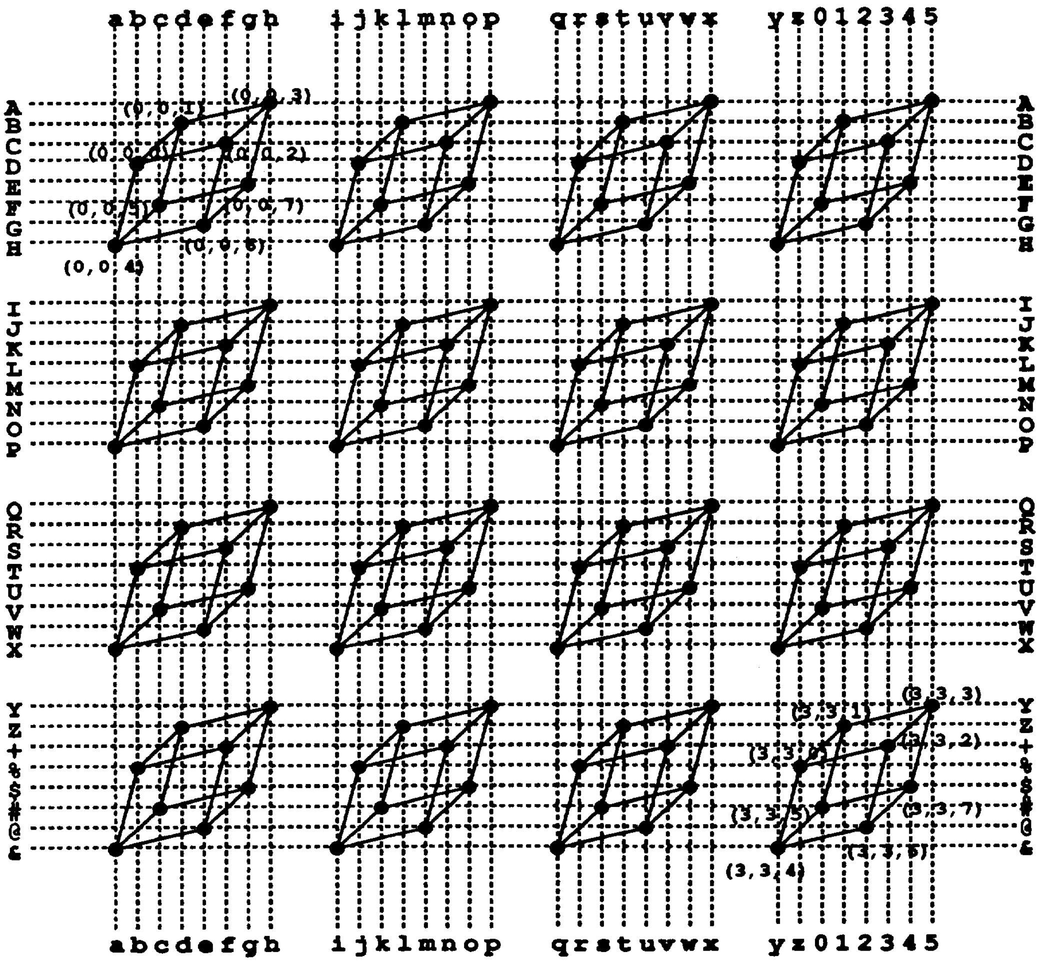


Fig. 1. An example of the optical multi-mesh hypercube network: a (4, 4, 3)-OMMH (128 nodes) interconnection is shown. Two links with the same labels are connected for the wrap-aroundconnections of the mesh. Only few addresses are shown in the parenthesis for clarity. Solid lines represent hypercube connections and dashed linesmesh connections.

OMMH in Fig. 1 looks like a three-cube-clustered 4×4 mesh.

An interesting isomorphic network is shown in Fig. 2. The same network is redrawn as a 4×4 mesh-clustered three-cube. Depending on the problems at hand, the OMMH can be configured as mesh-clustered hypercubes or hypercube-clustered meshes. This configuration flexibility is very suitable for MIMD (multiple instruction stream multiple data stream) mode of computation.

B. OMMH Network Properties

Message Routing in OMMH The distributed routing scheme for the OMMH network gives many alternative paths between any two nodes. For an (l, m, n)-OMMH network, let the addresses of two arbitrary nodes S and T be (i_s, j_s, k_s) and (i_t, j_t, k_t), respectively, where $0 \leq i_s < l, 0 \leq i_t < l, 0 \leq j_s < m, 0 \leq j_t < m, 0 \leq k_s < 2^n$, and $0 \leq k_t < 2^n$. The message routing scheme from S to T is that of an n-cube network or that of an $l \times m$ mesh network or a combination of the two depending upon the relative locations of the nodes.

1) *Routing within a hypercube:* if $i_s = i_t$ and $j_s = j_t$, then S and T are within the same hypercube. The routing scheme for this case is exactly the same as that of the regular n-cube network [18].

2) *Routing within a mesh:* if $k_s = k_t$, then S and T are within the same mesh. The routing scheme for this case is exactly the same as that of the regular $l \times m$ mesh network [19].

3) *Routing through meshes and hypercubes:* if none of the above two cases is true, S and T share neither a hypercube nor a mesh. The routing scheme for this case is first to use the hypercube routing scheme until the message arrives at the same mesh where T resides, and then to use the mesh routing scheme for the message to arrive at T. Or the mesh routing scheme can first be applied to forward the message to the same hypercube where T resides, and then the message can reach T

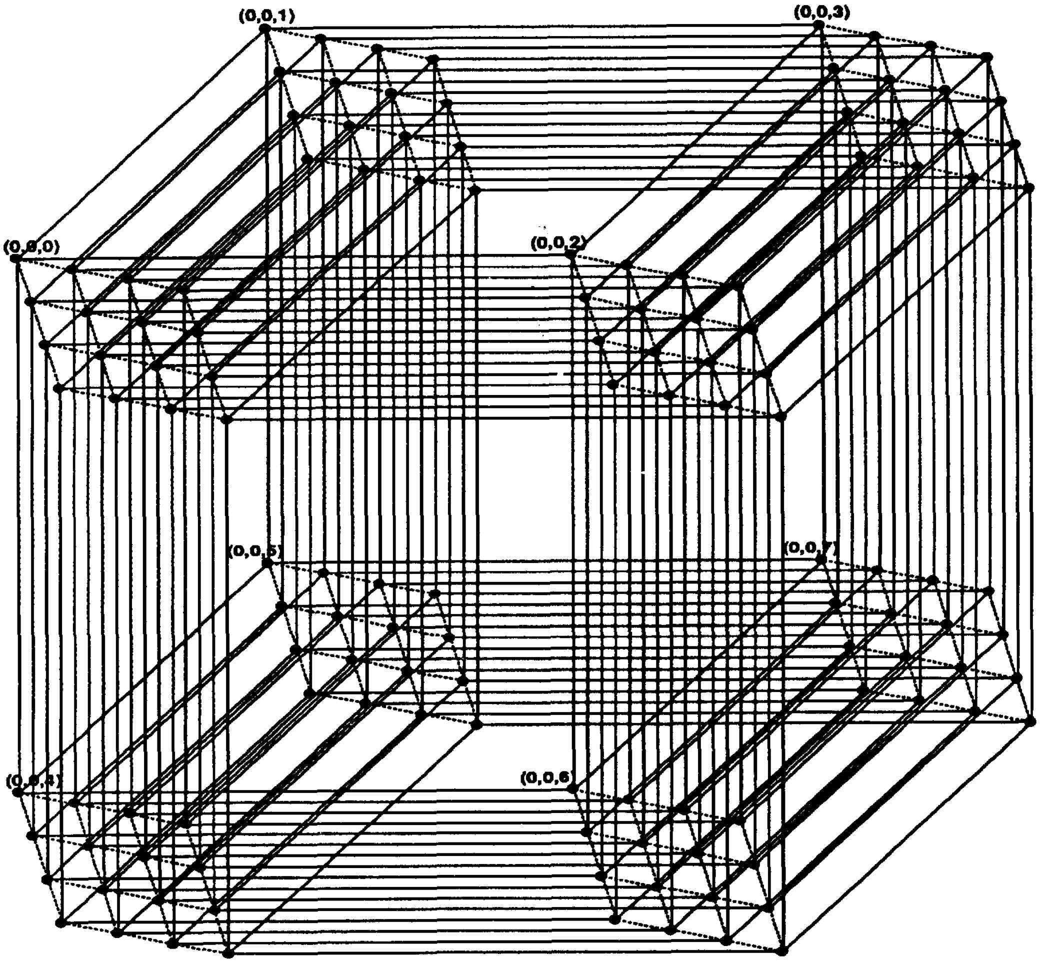


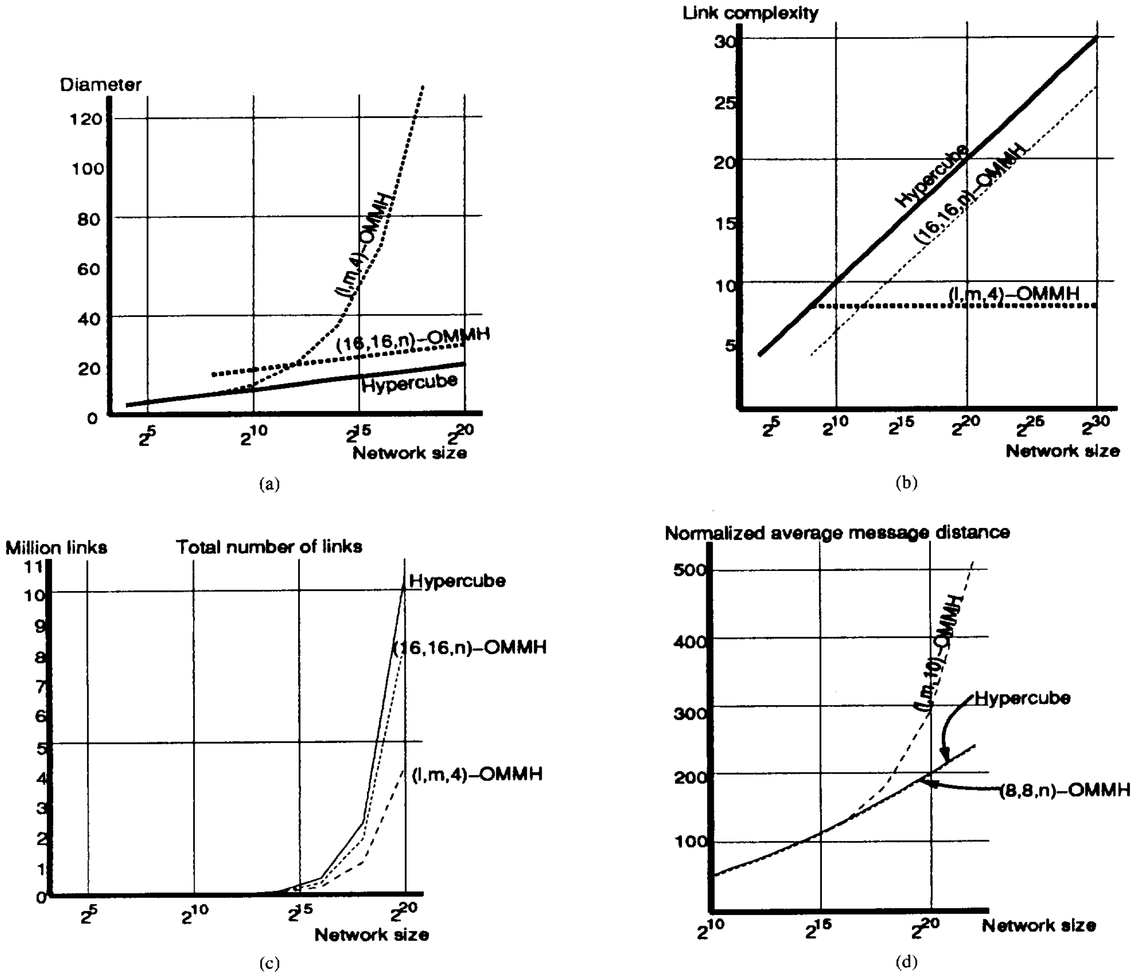
Fig. 2. A (4, 4, 3)-OMMH interconnection network, another isomorphic view. Wrap-around connections of the mesh are omitted and only a few addresses are shown in the parenthesis for clarity. Solid lines represent hypercube connections and dashed lines meshconnections.

using the hypercube routing scheme. We can also mix the hypercube and the mesh routing until the message is forwarded to the same hypercube or to the same mesh where T resides, and then we can forward the message to T using the hypercube or the mesh routing scheme, respectively.

The OMMH is less sensitive to performance degradation due to faults in links or nodes because the routing scheme in the OMMH has no preferred path, meaning all alternative paths have the same number of hops between any two nodes. This is an important advantage over other networks which have preferred paths such as Hypernet [2], Enhanced hypercube [20], or Extended hypercube [21].

Diameter and Link Complexity The distance between two nodes in a network is defined as the number of links connecting these two nodes. The diameter of a network is defined as the maximum of all the shortest distances between any two nodes. The diameter of the network is of great importance since it determines the maximum number of hops that a message may have to take. For two extreme cases, the diameter of a linear array with N nodes is $(N-1)$ while that of a completely connected network is unity. An $l \times m$ four-nearest-neighbor mesh has diameter $(\lfloor l/2 \rfloor + \lfloor m/2 \rfloor)$ if the mesh has wrapped-around connections, otherwise $(l + m - 2)$. The diameter of a hypercube with N nodes is $\log_2 N$. Thus, the diameter of (l, m, n)-OMMH is $(l + m + n - 2)$ if the mesh does not have wrapped-around connections, otherwise $(\lfloor l/2 \rfloor + \lfloor m/2 \rfloor + n)$.

Link complexity or node degree is defined as the number of links per node. The higher the link complexity, the greater is the hardware complexity and, consequently, the cost of the network. The node degree of a hypercube with N nodes is $\log_2 N$ and that of (l, m, n)-OMMH is $(n + 2)$ or $(n + 3)$ for outermost nodes, $(n + 4)$ for inner nodes if the mesh does not have wrapped-around connections. An (l, m, n)-OMMH with the wrap-around mesh has $(n + 4)$ links at every node. N is equal to $(l \times m \times 2^n)$ if the hypercube and the OMMH have the same network size. A comparison of diameters should be accompanied by a comparison of link complexity, because a higher connectivity



(a)

(b)

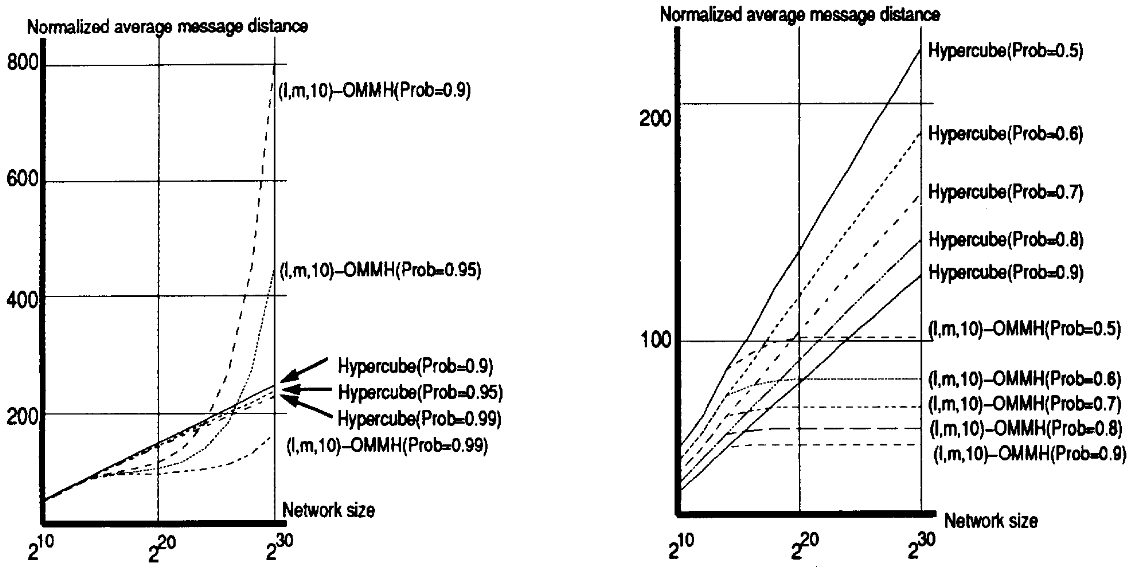
(c)

(d)

Fig. 3. Comparison of (a) diameter, (b) link complexity, (c) total number of links, and (d) normalized average message distance of the hypercube and the OMMH when the two networks have the same number of nodes.

resulting from a higher link complexity is expected to lead to smaller diameters. Fig. 3(a) compares the diameters of the hypercube and the OMMH, where $(16, 16, n)$-OMMH means the size of the mesh in the OMMH is fixed and the size of the hypercube in the OMMH is changed to have the same network size for comparison purposes. Similarly, $(l, m, 4)$-OMMH implies the size of the hypercube in the OMMH is fixed and that of the mesh is changed. Fig. 3(b) compares link complexities or node degrees of the hypercube and the OMMH. It should be noted that $(l, m, 4)$-OMMH has constant link complexity over the network size. This feature enables OMMH network to be scalable; that is, the growth of the network size does not affect the link complexity. Fig. 3(c) depicts the growth of the total number of links in the network as the network size increases. For a network size of one million nodes, the hypercube network contains about 10.5 million links while the $(l, m, 4)$-OMMH has about 4.2 million links and $(16, 16, n)$-OMMH has approximately 8.4 million links. Since one link implies one physical path, electrical or optical, between two nodes, the OMMH network is cost-efficient compared to the regular hypercube network in terms of hardware requirement.

Communication Efficiency The average message distance in a network is defined as the average number of links that a message should travel between any two nodes. It plays a key role in determining the queueing delay in a computer network [22]. In general, as the number of links per node increases, the average message distance decreases. In order to obtain a realistic comparison between different networks with different link complexity, some normalization should be made. For this purpose, it is assumed that the communication bandwidth available at a node is constant. As a consequence, the available communication bandwidth per link at a node decreases as the number of links at a node increases. In this context, the normalized average message distance was proposed as the average message distance multiplied by the number of links at the node [3]. This normalization is practical since, with no limits on the number of links, a completely connected network whose average message distance is unity could be designed. Thus, the above assumption is based on the fact that there are



(a)

(b)

Fig. 4. (a) Normalized average message distance using threshold model with 8-link threshold when probability within the threshold is 0.9, 0.95, or 0.99. (b) Normalized average message distance using geometric distribution model with four-link wide region. Probability within each region is 0.5, 0.6, 0.7, 0.8, or 0.9.

physical limitations in the number of pins and in the amount of power available to drive communication lines. Figure 3(d) plots the normalized average message distances against the network size of the hypercube and the OMMH, assuming that the message traffic is globally uniform; that is, the probability of a message being sent from any node to any other node is the same for all pairs of nodes. If the message traffic is globally uniform, the normalized average message distance of the OMMH with the fixed mesh size is no more than that of the regular hypercube.

However, it seems reasonable to assume that an efficient and realistic multicomputer system will show much heavier traffic over short distances than over long communication paths since tasks which can be partitioned into smaller subtasks would usually be assigned to neighboring processors. To characterize the locality of messages in multicomputer systems, the *Threshold Model* and the *Geometric Distribution Model* have been suggested and used to show performance of computer networks [2, 3]. The threshold model assumes that a fraction of all message destinations is uniformly distributed within some distance (threshold) of the source. The remaining destinations are uniformly distributed over the entire network. The geometric distribution model is defined as follows. For every source S, the nodes of the network are divided into regions $R_1, R_2, \cdots$ of increasing distance from S. A fraction β of all messages is destined for region R_1 of S, β of the remaining messages go to region R_2, and so on. Within each region, the distribution is uniform.

Fig. 4(a) shows the normalized average message distance of localized messages using the threshold model of eight-link threshold and Fig. 4(b) shows the the normalized average message distance using the geometric distribution model where each region is four-hop wide. We compare normalized average message distances of the hypercube and the (l, m, n)-OMMH when the two networks have the same number of nodes. With N nodes as the network size, the dimension of the hypercube is $\log_2 N$ and $l \times m \times 2^n$ nodes in the OMMH must be equal to N. The size of the mesh in the OMMH is chosen as square as possible. Fig. 4(a) indicates that as the message traffic becomes more localized, the network size within which the normalized average message distance of the $(l, m, 10)$-OMMH is shorter than that of the hypercube increases, where $(l, m, 10)$-OMMH means that the size of the hypercube in the OMMH is fixed and the size of the mesh is changed to have the same network size. Fig. 4(b) reveals that, with the geometric message distribution model, the increase of the normalized average message distance of the OMMH with constant cube with respect to the growth of the network size is negligible (constant in the graph) while that of the hypercube grows logarithmically with respect to the network size. This implies that the OMMH can be scaled up with little increase in the normalized average message distance.

C. Architectural Considerations

Scalability Scalable networks have the property that the size of the system (e.g., the number of communicating nodes) can be increased with nominal change in the existing configuration. Also, the increase in system size is expected to result in an increase in performance to the extent of the increase in size. As the dimension of the hypercube is increased by one, one more link needs to be added to every node in the network. In addition to the changes in the node configuration, at least a doubling of the size is required for the regular hypercube network to expand and remain a hypercube. This implies that the regular hypercube does not allow an incremental expansion of small sizes. Thus the regular hypercube network is not

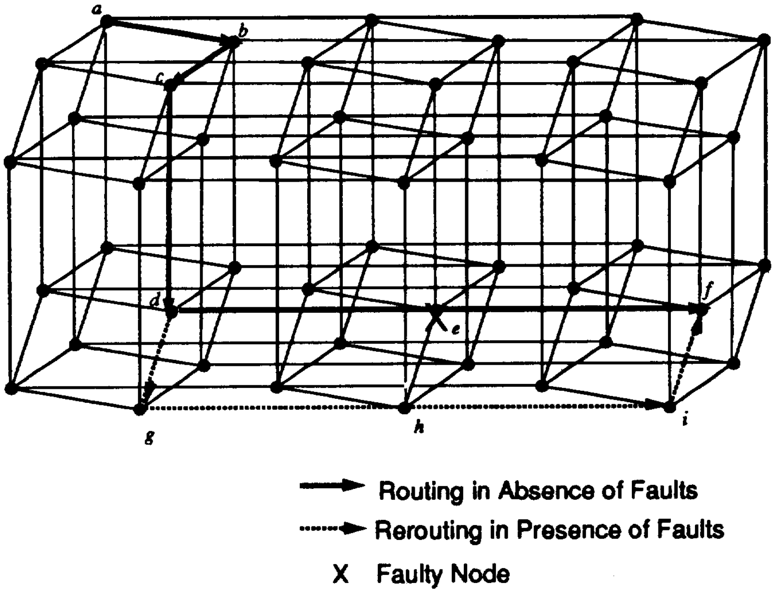


Fig. 5. Rerouting messages in the OMMH in the presence of a single fault.

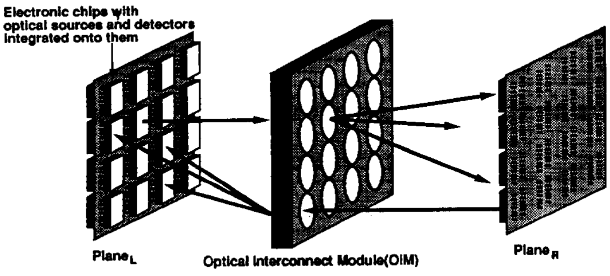

Electronic chips with optical sources and detectors integrated onto them
Plane$_L$
Optical Interconnect Module(OIM)
Plane$_R$

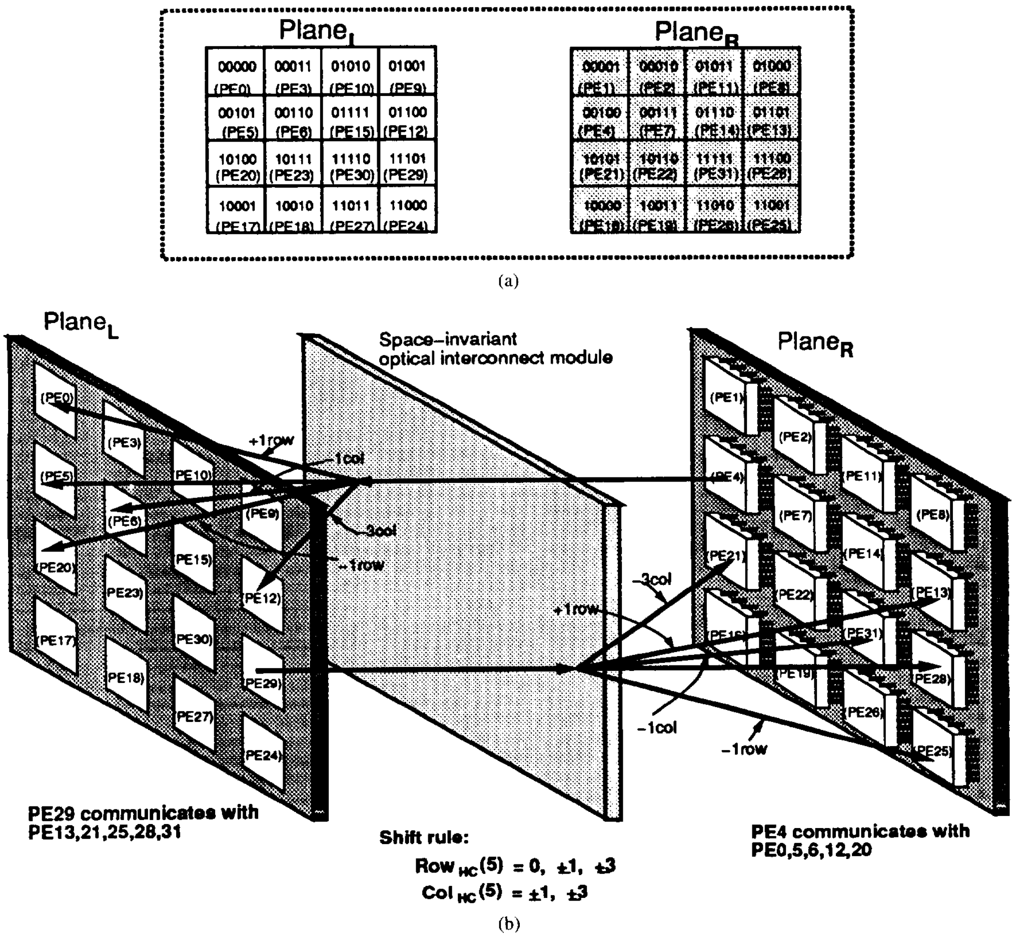
Fig. 6. . A model for 3-D optical interconnects.

scalable according to the above definition. We should note that the regular hypercube network may be scalable at a greater cost. Moreover, it is not modular [2], [3]. The lack of scalability and modularity have limited the application of the hypercube topology to large-scale high-speed data transmission systems despite the many other advantages it possesses.

This major limitation has motivated us to develop a new network topology that not only retains the many attractive properties of the hypercube network but also provides scalability. As can be seen in Fig. 3(b), the OMMH with a constant cube as a basic building block has a constant node degree, which means that the size of the OMMH is ready to be scaled up by expanding the size of the mesh without affecting the link complexity (number of links per node) of existing nodes as is the case in expanding the size of the hypercube network. However, we cannot just add one node to the OMMH. For an (l, m, n)-OMMH, we need to add at least $l \times 2^n$ nodes (if $l < m$) to have perfectly balanced mesh. In addition, in Fig. 4(b), the normalized average distance of the OMMH under geometric message distribution remains constant as the network size grows. This implies that the OMMH can be scaled up without increasing the normalized average distance. On the contrary, the regular hypercube can only be scaled up with logarithmic increase in the normalized average distance.

Fault Tolerance As the number of components in a system grows, the probability of the existence of faulty components increases. For a large-scale system, we cannot always expect that all components in such a system are free from failures. However, we need to expect such a system to continue to operate correctly in the presence of a reasonable number of failures. Due to the concurrent presence of meshes and hypercubes in the OMMH, rerouting of messages in the presence of a single faulty link or a single faulty node can easily be done with little modification of existing fault-free routing algorithms.

In the OMMH, any single faulty link or any single faulty node can be bypassed by only two additional hops as long as that particular node is not involved in the communication, namely, the node is neither the source nor the destination for any message. This can be proved as follows. As discussed in Section II-B-1), a message in the OMMH is routed using a mesh routing function if both the source and the destination of the message are in the same mesh subnetwork, or a hypercube routing function if those of the message are in the same hypercube subnetwork, or combination of these two routing functions if those of the message are neither in the same mesh nor in the same hypercube subnetwork. Consider the rerouting scheme in the presence of a single faulty link when the mesh routing function is being applied. When the message arrives at the node which is connected to the faulty link, it is forwarded to the neighboring mesh via one hop of the hypercube link (n such neighboring meshes exist in (l, m, n)-OMMH.). By applying the mesh routing function, the message arrives at a node which is one hop (one hypercube link) away from the destination since the message has been routed in



Plane$_L$

00000 (PE0)	00011 (PE3)	01010 (PE10)	01001 (PE9)
00101 (PE5)	00110 (PE6)	01111 (PE15)	01100 (PE12)
10100 (PE20)	10111 (PE23)	11110 (PE30)	11101 (PE29)
10001 (PE17)	10010 (PE18)	11011 (PE27)	11000 (PE24)

Plane$_R$

00001 (PE1)	00010 (PE2)	01011 (PE11)	01000 (PE8)
00100 (PE4)	00111 (PE7)	01110 (PE14)	01101 (PE13)
10101 (PE21)	10110 (PE22)	11111 (PE31)	11100 (PE28)
10000 (PE16)	10011 (PE19)	11010 (PE26)	11001 (PE25)

(a)

(b)

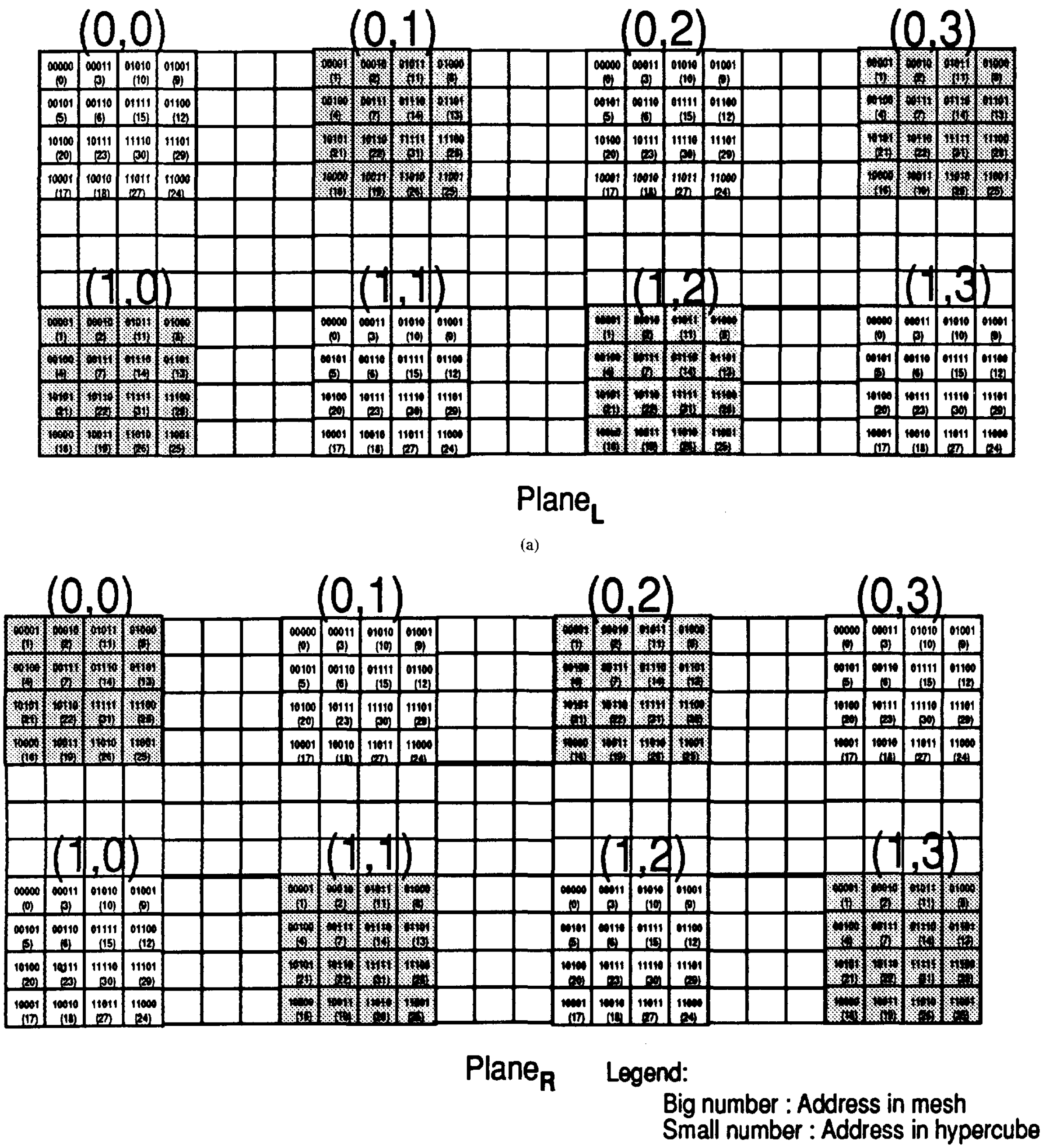
Fig. 7. Conceptual realization of a 3-D space-invariant five-cube network: (a) the 32 nodes of the five-cube network are partitioned into two partitions with totally space-invariant connections between them, (b) a conceptual optical realization of the space-invariant five-cube network. The connections of two nodes, one from each plane, are shown as an example. The shift rule defines the amount of row-wise and column-wise shifts to be performed by the optical interconnect module.

the neighboring mesh to detour the faulty link. Similarly, a single faulty link when the hypercube routing function is being applied can be bypassed by forwarding the message to the neighboring hypercube via a mesh link (four such hypercubes always exist in the OMMH). The rerouting scheme in the presence of a single faulty node is the same as that in the presence of a single faulty link but the message forwarding is done at the node located at one hop ahead of the faulty node. Thus, rerouting in the presence of a single faulty node or link can be done with two additional hops with little modification of the fault-free routing methods.

Fig. 5 shows a rerouting scheme in the OMMH network in the presence of a single faulty node. Suppose that the source of a message is node a and the destination node f. In the absence of faults, a message is forwarded from a to c by a hypercube routing scheme and from c to f by a mesh routing scheme. In the presence of a faulty node e, the message is forwarded to a neighboring mesh at node d which is one hop ahead of the faulty node. From g, the same mesh routing scheme is applied and when the message arrives at node i, it is returned to the original mesh where the final destination f belongs. Thus two additional hops are sufficient for rerouting the message to bypass the faulty node.

III. Optical Implementation of OMMH Network

Recently there has been a great deal of interest in the application of optics as an interconnection medium for high-speed computing and parallel processing [4]–[8], [23]. One of the most promising approaches is the use of free-space optical interconnects as opposed to guidewave (e.g., fibers or waveguides based on polymers) because of their tremendous spatial parallelism [5]. In this section, we first summarize a 3-D totally space-invariant optical implementation methodology of the hypercube network and, then, present a totally space-invariant implementation methodology of the proposed OMMH network. A model for 3-D optical interconnects used in this paper is shown in Fig. 6.



(a)

(b)

Fig. 8. (2,4,5)-OMMH embedding: (a) Plane$_L$ (b) Plane$_R$

A. 3-D Space-Invariant Optical Implementation of Hypercube Networks

The basic idea is derived from an observation that nodes in an interconnection network can be partitioned into two sets of nodes such that any two nodes in a set do not have a direct link. This is a well-known problem of bipartitioning a graph if the interconnection network is represented as a graph. For a binary n-cube, nodes whose addresses differ by more than one in Hamming distance can be in the same partition, since no link exists between two nodes if their Hamming distance is greater than one. Besides bipartitioning the graph, we arrange the nodes in each partition onto the plane such that interconnection between two planes becomes space-invariant.

A conceptual three-dimensional implementation of a five-cube (32 nodes) interconnection using the optical interconnect model is shown in Fig. 7. Fig. 7(a) illustrates the 3-D space-invariant embedding of a five-cube (32 nodes) network. All nodes on the left plane (Plane$_L$) (16 nodes) have the same

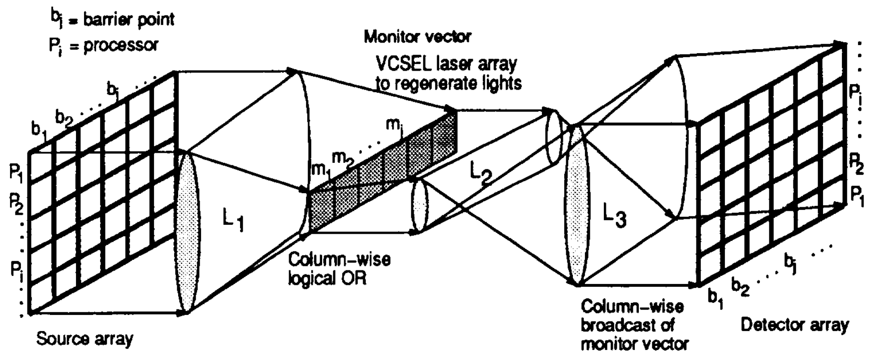


Fig. 9. Optical setup for the barrier synchronization.

connection patterns to nodes on the right plane (Plane$_R$) (16 nodes). Since the links are bidirectional, all nodes on the right plane have the same exact connection patterns to the left plane. A number in a node on the plane represents the binary address of the corresponding node. Conceptual implementation of a 3-D five-cube interconnection using the proposed model system is shown in Fig. 7(b). The required connections for a 3-D five-cube network are obtained by superimposing nine images of one plane onto the other plane (eight spatially shifted and one directly imaged onto the receiving plane). The amount of spatial shifts are $\pm 1d$ and $\pm 3d$ in both horizontal and vertical directions where d is the size of a node, and the origin is taken to be the center of the plane. Recall that communication patterns from plane$_L$ to plane$_R$ are identical to those from plane$_R$ to plane$_L$. The nine images are simultaneously incident on the receiving plane in which a receiving node gets five different optical signals representing the required hypercube connections.

The construction of an arbitrary n-cube network is carried out incrementally by putting together two $(n-1)$-cube networks, one of the two is column-wise or row-wise rotated version of the other. For more details, see [24] and [25]. The scheme in [25] is used for the implementation of the OMMH network.

B. 3-D Space-Invariant Implementation of OMMH Networks

3-D space-invariant optical implementation of the OMMH is derived in this Subsection. To facilitate the description of the embedding scheme, a few notations are defined below which have been used in [24] for the description of embedding space-invariant hypercube networks.

The embedding scheme of the (l, m, n)-OMMH using the model of Fig. 6 can be described as follows:

1. Construct layouts (two layouts per hypercube, one for Plane$_L$ and the other for Plane$_R$) of $l \times m$ hypercubes with dimension n.
2. Place hypercube layouts in the above step as building blocks in a 2-D matrix form with l rows and m columns on each plane.
3. Interchange the layout for Plane$_L$ and the layout for Plane$_R$ of hypercubes in every other row and in every other column.
4. Separate each hypercube layout in the matrix by r empty rows and by c empty columns, where $r = 0, c = 1$ if $n = 2, r = 1, c = 1$ if $n = 3$, $r = 1, c = 3$ if $n = 4, r = 3, c = 3$ if $n = 5$, and $r = \mathcal{D}_r(n) - \mathcal{D}_r(n-3)$, $c = \mathcal{D}_c(n) - \mathcal{D}_c(n-3)$ if $n > 5$.

Fig. 8 shows the 3-D implementation of $(2, 4, 5)$-OMMH network using the proposed construction algorithm (Fig. 8(a) corresponds to Plane$_L$ and Fig. 8(b) to Plane$_R$.). The required connections for the (l, m, n)-OMMH network constructed by the algorithm are as follows. Let d be the size of a node square. Shifts in the amount of $\pm[2 \times \mathcal{D}_r(n) - \mathcal{D}_r(n-3)] \times d$ in row-wise direction and $\pm[2 \times \mathcal{D}_c(n) - \mathcal{D}_c(n-3)] \times d$ in column-wise direction accomplish the required connection for the four-nearest-neighbor links in the mesh. Shifts in the amount of $\pm[2 \times \mathcal{D}_r(n) - \mathcal{D}_r(n-3)] \times (m-1) \times d$ in row-wise direction and $\pm[2 \times \mathcal{D}_c(n) - \mathcal{D}_c(n-3)] \times (l-1) \times d$ in column-wise direction accomplish the required connection for the wrap-around links in the mesh. The shift rule for an n-cube, Row$_{\mathrm{HC}}$(n) and Col$_{\mathrm{HC}}(n)$, generates required connection for the hypercube links. Thus the shift rule for an (l, m, n)-OMMH, denoted by Row$_{\mathrm{OMMH}}(l, m, n)$ and Col$_{\mathrm{OMMH}}(l, m, n)$, can be expressed as follows:

$$\begin{aligned}\mathrm{Row}_{\mathrm{OMMH}}(l,m,n) &= \mathrm{Row}_{\mathrm{HC}}(n), [2\mathcal{D}_r(n) - \mathcal{D}_r(n-3)],\\ &\quad \times [2\mathcal{D}_r(n) - \mathcal{D}_r(n-3)] \times (l-1)\\ \mathrm{Col}_{\mathrm{OMMH}}(l,m,n) &= \mathrm{Col}_{\mathrm{HC}}(n), [2\mathcal{D}_c(n) - \mathcal{D}_c(n-3)],\\ &\quad \times [2\mathcal{D}_c(n) - \mathcal{D}_c(n-3)] \times (m-1)\end{aligned} \tag{1}$$

As can be seen in Fig. 8, we can expand the size of the OMMH by adding more hypercube layouts used as basic

Plane$_L$ (or Plane$_R$) : A plane on which one of the two partitions of nodes is placed.
$\mathcal{D}_r(n)$ (or $\mathcal{D}_c(n)$) : the row (or column) dimension of the resulting n-cube on one plane.
Row$_{\mathrm{HC}}(n)$ (or Col$_{\mathrm{HC}}(n)$) : the amount of row-wise (or column-wise) shifts to be performed by the optical interconnect module to realize an n-cube network.

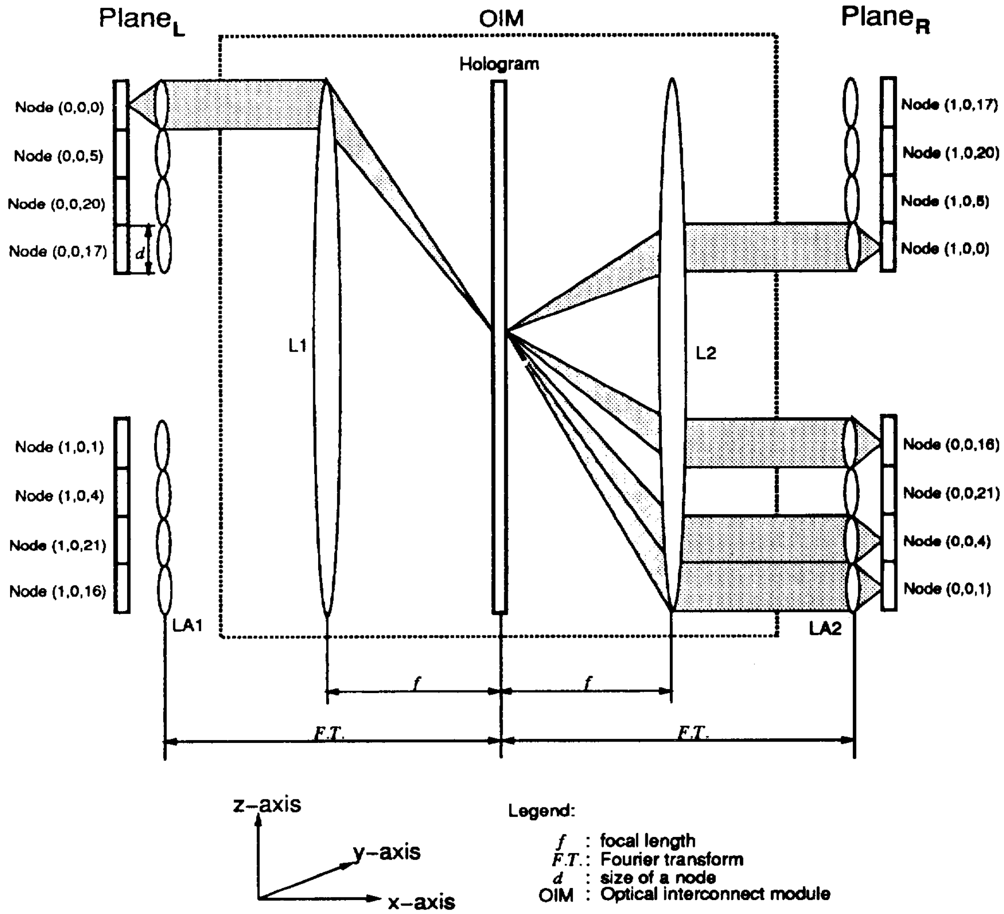


Fig. 10. A (2,4,5)-OMMH implementation using a space-invariant Fourier plane hologram: Only side view (xz-plane) for connections from node (0,0,0) to nodes (0,0,1), (0,0,4), (0,0,16), (1,0,0)is shown for clarity.

building blocks along the perimeter of the mesh. The number of shifts (number of fanouts) in the shift rule remains unchanged. If we use the OMMH with meshes having no wrap around connections, the amount of shifts in the shift rule does not change, either. This is very desirable feature because the optical interconnect module that generates the required number of shifts and the required amount of each shift remains unchanged even if the network grows in size.

C. Optical Support for the Barrier Synchronization

Architectural support for efficient process synchronization is an important aspect of the design of any MIMD multiprocessor. Message-based synchronization primitives require minimal hardware support but they would not be appropriate in a massively parallel system since lots of messages (overhead) are required in such a system whenever a barrier is encountered. Barrier synchronization is a mechanism that guarantees that all processes have reached a specified point in their execution before any are allowed to proceed. In Fig. 9, we present an optical setup which implements a barrier mechanism for fast synchronization. This setup could be used as a control subnetwork when the OMMH network is used in a massively parallel system. The source array could be a spatial light modulator illuminated by a laser where the i-th row represents processor P_i and the j-th column represents barrier point b_j. In the detector array, the rows and the columns have the same meaning as those in the source array but the rows are numbered from bottom to top due to the image inversion. Let (P_i, b_j) denote a cell where i-th row and j-th column meet. Suppose that a logical 1 is coded as the presence of light and a logical 0 as the absence of light. For a given synchronization pattern, (P_i, b_j) is set to 1 if barrier point b_j is involved in the synchronization pattern and P_i is initiated. When P_i finishes its execution, (P_i, b_j) is reset to 0. Since the monitor vector is a row vector which is the column-wise logical OR (by cylindrical lens L_1) of the source array, m_j is 0 only when all processors which need to be synchronized at barrier b_j finish their jobs. Now, the value of m_j is broadcasted to all processors through cylindrical lenses L_2 and L_3 on the detector array. Processor P_i knows the time when all other processors reach the barrier point b_j by detecting when the value of (P_i, b_j) changes from 1 to 0. A similar electronic implementation can be found in Ref. [26] where wired-NOR logic is used. The above dynamic barrier synchronization is possible only if the synchronization pattern is predicted at compile time and process preemption is not allowed. However, as discussed in Ref. [26], the above scheme along with counting semaphores can support multiprogrammed multiprocessors where preemption is allowed.

IV. Optical Hardware Required

There is a wide variety of optical components for accomplishing the basic interconnect operations, including, lenslet arrays [27] multi-split lenses [28], off-axis lenses [13], mirror arrays [14], gratings [29], and holographic techniques [30]. In order to illustrate the approach, we choose as a target network an OMMH network with five-cubes as basic building blocks (e.g., $(l, m, 5)$-OMMH where l and m are integers) and will describe an optical module for its implementation.

One particularly attractive approach for the realization of optical interconnects is the use of holographic optical elements (HOEs). HOEs offer high densities 10^4/cm^2 for space-variant and as high as 10^8/cm^2 for space-invariant interconnects using a single holographic element, while providing relatively low crosstalk. In addition, holographic approaches may be mass produced.

One simple space-invariant Fourier plane hologram would realize the entire 3-D OMMH interconnection network [7]. An envisaged implementation with HOEs of an $(2, 4, 5)$-OMMH network, for example, is illustrated in Figure 10. In the figure only the side view (xz-plane) is shown for clarity and nodes in Plane$_R$ are numbered from bottom to top because of image inversion due to the use of lenses. This figure illustrates how node (0,0,0) sends signals to node (0,0,1), (0,0,4), (0,0,16), and (1,0,0). The light beam from a source is collimated by the lenslet array ($LA1$) and incident on the hologram through a Fourier transform lens ($L1$). The hologram, in this case, is placed in the Fourier plane. The hologram splits and spatially shifts the incident beam. Multiple beams are then focused on the corresponding detectors for the required connections through another Fourier transform lens ($L2$) and a lenslet array ($LA2$). Since the hologram is not bidirectional, methods for providing bidirectional communications need to be used. A possible setup would be the use of the two orthogonal polarization states of light. The hologram can be recorded optically or can be a computer generated hologram. In either case, since the hologram is space-invariant, it is expected to be relatively simple to construct.

V. Conclusion

To overcome the lack of scalability in the regular hypercube networks, a new interconnection network topology, called an Optical Multi-Mesh Hypercube, is presented. The proposed network is a combination of hypercube and mesh topologies. The analysis and simulation results show that the new interconnection network is very scalable, meaning the configuration of the existing nodes is relatively insensitive to the growth of the network size, and more efficient in terms of communication. It is also shown that the new interconnection network is highly fault-tolerant. Any faulty node or link can be bypassed by only two additional hops with little modification of the fault-free routing scheme. Due to the concurrent existence of multiple meshes and hypercubes, the new network provides a great architectural support for parallel processing and distributed computing. In addition, a wide body of parallel algorithms that have been designed for the hypercube and the mesh interconnection are readily implementable on the proposed network.

More importantly, the proposed network is highly amenable to optical implementations. A three-dimensional optical implementation technique of the proposed network is provided. It is based on an efficient three-dimensional space-invariant implementation scheme for the regular hypercube. The proposed optical implementation technique for the new network results in totally space-invariant connection pattern at every node. Consequently, simple and cost-efficient optical implementation of the proposed network with existing optical hardware would be possible.

References

[1] K. Hwang, *Advanced Computer Architecture: Parallelism, Scalability, Programmability.* New York: McGraw-Hill, 1993.

[2] K. Hwang and J. Ghosh, "Hypernet: a communication-efficient architecture for constructing massively parallel computers," *IEEE Trans. Comput.*, vol. C-36, pp. 1450–1466, 1987.

[3] J. R. Goodman and C. H. Sequin, "Hypertree: a multiprocessor interconnection topology," *IEEE Trans. Comput.*, vol. C-30, pp. 923–933, 1981.

[4] J. W. Goodman, F. J. Leonberger, S. Y. Kung and R. A. Athale, "Optical interconnections for VLSI systems," *Proc. IEEE*, vol. 72, pp. 850–866, July 1984.

[5] A. Guha, J. Bristow, C. Sullivan and A. Husain, "Optical interconnections for massively parallel architectures," *Applied Optics*, vol. 29, pp. 1077–1093, Mar. 1990.

[6] A. Louri, "3-D optical architecture and data-parallel algorithms for massively parallel computing," *IEEE MICRO (Chips, Systems, Software, and Applications)*, vol. 11, pp. 24–68, Apr. 1991.

[7] A. A. Sawchuk and B. K. Jenkins, "Dynamic Optical interconnections for Optical Processors," in *Proc. Soc. Photo-Opt. Instr. Eng.*, vol. 625, Los Angeles, pp. 145–153, SPIE, 1986.

[8] W. T. Cathey, K. Wagner, and W. J. Miceli, "Digital computing with optics," *Proc. IEEE*, vol. 77, pp. 1558–1572, Oct. 1989.

[9] F. Kiamilev, P. Marchand, A. V. Krishnamoorthy, S. C. Esener, and S. H. Lee, "Performance comparison between optoelectronic and VLSI multistage interconnection networks," *IEEE J. Lightwave Technol.*, vol. 9, pp. 1665–1674, Dec. 1991.

[10] R. A. Nordin, A. F. J. Levi, R. N. Nottenburg, J. O'Gorman, T. Tanbun-Ek, and R. A. Logan, "A system perspective on digital interconnection Technology," *IEEE J. Lightwave Technol.*, vol. 10, pp. 811–827, Jun. 1992.

[11] M. G. Taylor and J. E. Midwinter, "Optically interconnected switching networks," *IEEE J. Lightwave Technol.*, vol. 9, pp. 791–798, June 1991.

[12] T. J. Cloonan and F. B. McCormick, "Photonic Switching Applications of 2-D and 3-D Crossover Networks Based on 2-input, 2-output Switching Nodes," *Applied Optics*, vol. 30, pp. 2309–2323, June 1991.

[13] M. W. Haney and J. J. Levy, "Optically efficient free-space folded perfect shuffle network," *Appl. Optics*, vol. 30, pp. 2833–2840, July 1991.

[14] Y. Sheng, "Space invariant multiple imaging for hypercube interconnections," *Appl. Optics*, vol. 29, pp. 1101–1105, 1990.

[15] K. H. Brenner and A. Huang, "Optical implementation of the perfect shuffle interconnection," *Appl. Optics*, vol. 27, pp. 135–137, 1988.

[16] T. S. Wailes and D. G. Meyer, "Multiple channel architecture: a new optical interconnection strategy for massively parallel computers," *IEEE J. Lightwave Technol.*, vol. 9, pp. 1702–1716, Dec. 1991.

[17] G. E. Lohman and K. H. Brenner, "Space-invariance in optical computing systems," *Optik*, vol. 89, pp. 123–134, 1992.

[18] H. J. Siegel, *Interconnection Networks for Large-scale Parallel Processing.* New York: McGraw-Hill, 1990.

[19] D. Nassimi and S. Sahni, "An optimal routing algorithm for mesh connected parallel computers," *J. ACM*, vol. 27, pp. 6–29, Jan. 1980.

[20] N. F. Tzeng and S. Wei, "Enhanced hypercubes," *IEEE Trans. Comput.*, vol. 40, pp. 284–294, Mar. 1991.

[21] J. M. Kumar and L. M. Patnaik, "Extended hypercube: a hierarchical interconnection network of hypercubes," *IEEE Trans. Parallel and Distrib. Syst.*, vol. 3, pp. 45–57, Jan. 1992.

[22] L. N. Bhuyan and D. P. Agrawal, "Generalized hypercube and hyperbus structures constructing massively parallel computers," *IEEE Trans. Comput.*, vol. C-33, pp. 323–333, 1984.

[23] A. D. McAulay, *Optical Computer Architectures: the Application of Optical Concepts to Next Generation Computers.* New York: Wiley, 1991.

[24] A. Louri and H. Sung, "A design methodology for three-dimensional space-invariant hypercube networks using graph bipartitioning," *Optics Lett.*, vol. 18, pp. 2050–2052, Dec. 1 1993.

[25] A. Louri and H. Sung, "Efficient implementation methodology for three-dimensional space-invariant hypercube-based free-space optical interconnection networks," *Appl. Optics*, vol. 32, pp. 7200–7209, Dec. 1993.

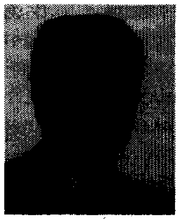
[26] K. Hwang and S. Shang, "Wired-NOR barrier synchronization for designing shared-memory multiprocessor," in *Proce. 1991 Int. Conf. on Parallel Processing*, CRC Press, Aug. 1991.

[27] I. Glaser, "Lenslet array processors," *Appl. Optics*, vol. 21, pp. 1271–1280, Apr. 1982.

[28] A. S. Kumar and R. M. Vasu, "Multiple imaging and multichannel optical processing with split lenses," *Appl. Optics*, vol. 26, pp. 5345–5349, 1987.

[29] L. P. Boivin, "Multiple imaging using various types of simple phase gratings," *Appl. Optics*, vol. 11, pp. 1782–1792, 1972.

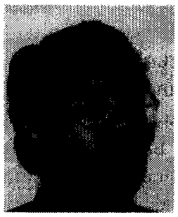
[30] H. Damman and K. Gortler, "High-efficiency in-line multiple imaging by means of phase holograms," *Opt. Commun.*, vol. 3, pp. 312–315, 1971.



Ahmed Louri (M'88) received the Engineer Degree in electrical engineering from the University of Science and Technology, and the M.S. and Ph.D. in computer engineering from the University of Southern California.

From 1986 to 1988, he worked as a Researcher with the Computer Research Institute (CRI) at USC, where he conducted research in paralllel processing, multiprocessor system design, and optical computing. Since 1988 he has been on the Faculty of the Department of Electrical and Computer Engineering at the University of Arizona, where he is now an Associate Professor and Director of the Optical Computing and Parallel Processing Laboratory. His research interests include parallel processing, optical computing, and optical interconnects. He has published numerous journal and conference articles.

In 1991, Dr. Louri received the Best Article of 1991 Award from IEEE Micro. In 1988 he was the recipient of the NSF Research Initial Award. He is a member of IEEE, ACM, OSA, and SPIE.



Hongki Sung (SM'91) received the B.S. degree in electronics engineering from Seoul National University, Seoul, Korea, in 1984, and the M.S. degree in computer engineering from the University of Southwestern Louisiana, Lafayette, LA, in 1990. He is pursuing the Ph.D. at the University of Arizona, Tucson.

From 1984 to 1989, he was a member of the Techical Staff at the Electronics and Telecommunications Research Institute in Taejon, Korea. He is now a Research Associate with the Optical Computing a Parallel Processing Laboratory at the University of Arizona, Tucson, and he expects to complete the Ph.D. degree in 1994. His research interests are in the areas of interconnection networks including optical interconnects, cache memories, and parallel computer architectures.