3D Optical Interconnects for High-Speed Interchip and Interboard Communications

Ahmed Louri and Hongki Sung

University of Arizona

Metal-based communications between subsystems and chips has become the limiting factor in high-speed computing. Maturing optics-based technologies offer advantages that may unplug this bottleneck.

Increasing volumes of data handled by today's information systems, coupled with the growing need for more processing functionality and system throughput, are placing stringent demands on communication bandwidths and processing speeds. While there has been significant progress in high-speed processing element (PE) design, high-performance interconnection network design has lagged. The primary bottleneck in today's metal-based interconnection networks is the very limited bandwidth, which results in limited communication speed.

Silicon and GaAs-based technologies are driving device speeds into the gigahertz (GHz) range and reducing rise times to fractions of a nanosecond. It is not device speeds, however, but metal-based communications between subsystems and chips at higher and higher data rates that has become the deciding factor in determining performance and cost of high-speed computing systems.[1,2] In fact, the emerging demand for greater bandwidth, data rates, and interconnect density, along with better error rates, is far beyond the capabilities of current metal interconnects and backplanes.[3]

The limitations of metal interconnects apply to both uniprocessors — high-end personal computers, workstations, and mid-range computers — and multiprocessor systems. Improved performance in uniprocessors can be generally obtained from faster clock speeds. However, to achieve improved performance from systems operating in the GHz range, metal interconnects must be treated as high-frequency transmission systems. It is the nature of metal interconnect technology that skin effect, crosstalk, interference, wave reflections, electrical noise due to current changes, and dielectric imperfections can all cause severe pulse distortions and attenuation, clock skew, and random propagation delays.[2] Multiprocessor systems based on metal interconnects experience the technological limitations of communication bandwidth constraints, low interconnect density, long network latencies, and high power requirements.

Excessive power consumption and transmission delays in high-speed systems are two of the most obvious limitations inherent with metal interconnects. As illustrated in Figure 1, electrical connections can perform relatively well where distances are less than 1 millimeter, such as within chips.[1] Where distances are greater than 1 mm (for example, between chips, multichip modules, boards, or frames), metal interconnects must, by design, occupy a larger area of the chip and consume excessive on-chip power, which drives up the cost of achieving higher levels of performance.[1,3]

The source of the power consumption problem stems from a fundamental

0018-9162/94/ $4.00 © 1994 IEEE

impedance-matching problem created when metal conductors are subjected to high data rates. Essentially, electrical transmission lines have low impedance over long distances, whereas the small logic devices that drive the electrical connections have high impedance. Hence, we cannot use high-impedance logic devices to communicate logic levels over long distances. To resolve this mismatch, we must build large drivers that also have low impedance, which takes up more room on the chip and, consequently, also increases power consumption. An additional limitation of metal interconnects is the transmission delay that results when capacitance of the metal conductors must be charged to the logic level, then discharged at the remote end. For electrical transmission, this charging effect is proportional to the length of the line and to the number of devices attached to it. The impedance-matching problem, combined with the capacitive loading effect, is the primary cause for power consumption and the intolerable delays for long-distance connections in high-speed systems.

Another inherent limitation of metal interconnects is crosstalk between adjacent conductors. Inductive/capacitive coupling increases crosstalk at high frequencies, so shielding and isolation are required. Moreover, the timing of logic signals as they arrive at a gate from different parts of a circuit becomes critical, as the timing issue gives rise to severe clock skew problems. Crosstalk, impedance matching, and timing problems will ultimately limit clock speeds, which up until now have been the primary means of improving uniprocessor system performance.

An alternative to increasing clock speeds is parallel processing, whereby the system designer increases the number of processing elements in the system. However, communication between PEs — and between PEs and memory modules — is a limiting factor in determining system performance and cost. Unfortunately, present electronic technology faces limitations in terms of power dissipation, latency, and chip area requirements.[2] Consequently, designers of future massively parallel machines cannot rely on electronic technology alone for adequate and cost-effective communication support.

The metal interconnect problem exists at various organizational levels, including frame-to-frame, chip-to-chip, and intrachip levels. In the three-dimensional model we propose later in this article, we concentrate on chip-to-chip and board-to-board interconnection levels. These two levels are of particular interest because they support many high-speed signal transmissions and are directly affected by the problems associated with wiring complexity, signal degradation due to electrical multireflections, and long delays resulting from unnecessary channel multiplexing.

As an example of the problems facing designers of such highly parallel machines as the Ncube and Thinking Machines' Connection Machine, consider the tradeoffs that have been made for the input/output design at the board level to accommodate the natural limitations of metal interconnects. The increased component density of today's integrated circuits requires literally hundreds of electrical interconnects for each chip and thousands of I/O ports for the entire system. In the case of the Ncube, as many as 512 PEs have been restricted to a serial I/O channel. In the case of the Connection Machine, the PEs have had to share a common serial I/O line. What this means is that PE communication is serialized rather than parallel, which severely restricts the machine's performance. The net effect is that system throughput is drastically limited by the I/O bandwidth at the board level, not by the speed of the individual PE.

Advantages and disadvantages of optics for high-speed and high-density interconnects

Advantages

- Inherent parallelism
- Higher temporal and spatial bandwidths
- Higher interconnection densities
- Less signal crosstalk
- Immunity from electromagnetic interference and ground loops
- Freedom from quasi-planar constraints
- Lower signal and clock skew
- Lower power dissipation
- Larger number of fan-ins and fan-outs
- Propagation speed independent of communication distance
- Potential for reconfigurable interconnects

Disadvantages

- Lower switching speeds of optical switching devices (for reasonable power levels at present time)
- Optical signal attenuation
- Difficulty in aligning optical elements
- Low conversion (electron-to-photon, photon-to-electron) efficiency of present devices
- Relatively immature technology

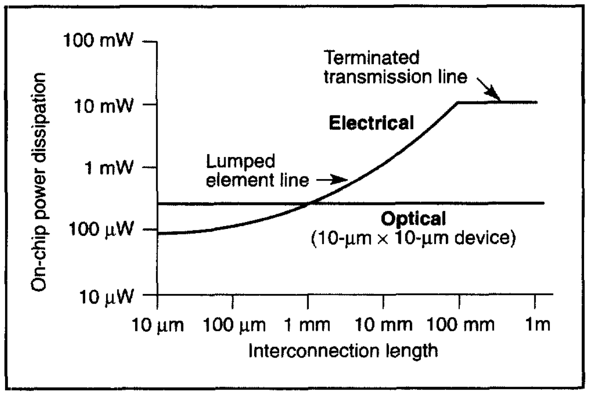


Figure 1. Minimum on-chip power dissipation at 1 gigabit per second. Courtesy of R.A. Nordin et al.[1]

Why optics for interconnects?

Optical interconnects offer high-speed computers key advantages over metal interconnects (see sidebar above). These include (1) high spatial and temporal bandwidths, (2) high-speed transmission, (3) low crosstalk independent of data rates, and (4) high interconnect densities.

With metal interconnects, the communication bandwidth is limited by the resistance, capacitance, and inductance of the path between different devices. With optics, the bandwidth is limited only by the carrier frequency — about 5×10^{14} Hz for visible light — which is very high when compared with the frequency for electronic systems — about 10 MHz to 1 GHz. In addition, because optical systems are free from planar constraints, they can exploit the third spatial dimension, and this can dramatically increase the available communication bandwidth.

Another factor favoring the use of optical interconnects is that propagation occurs independently of the number of components receiving the signal, which allows higher transmission speeds and a much larger number of fan-ins/fan-outs than with electronics. With traditional metal interconnects, signal propagation depends on the capacitance per unit length. Therefore, the speed of signal propagation decreases as more devices having capacitance are attached to the transmission line.

Despite the improvements to system performance obtainable with optical interconnects, at present it is difficult to convert information into optical form efficiently. Reliable communication requires critical alignment of optical transmitters, channel elements, and receivers. Then, too, fabrication and integration technologies of optoelectronic transmitters/receivers are still relatively immature. Optical-source wavelength stability is another major issue for communication reliability. Additionally, the switching speed of optical systems is inversely related to the input power. For high switching speed — in the nanosecond range — excessive levels of optical power are required. Ultimately, however, we want to stress that the disadvantages we cited are technology dependent and are not fundamentally insurmountable problems. Ongoing research will eventually point the way to improved performance of optical switches and converting devices.

For chip-to-chip and board-to-board interconnects, the optical approach clearly offers several advantages over the electrical approach because it

- does not require the source electronics to drive a large capacitive load created by the capacitance of the output bonding pad, the signal trace, the input bonding pad, and the input transistor,
- does not require impedance-matching terminating resistors that increase the power requirements of the source electronics,
- may permit a higher density of pinouts between chips, and
- may yield lower levels of crosstalk than a metal-based approach.

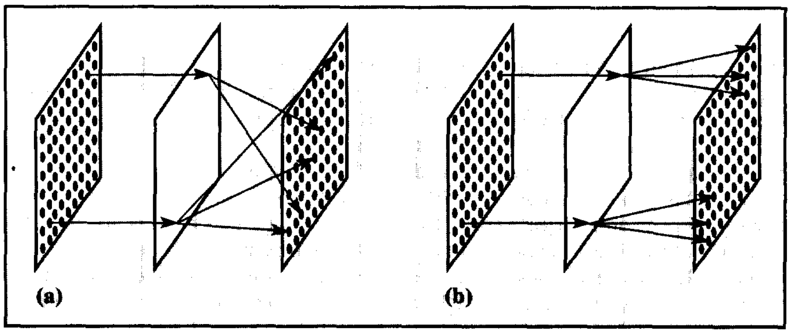


Figure 2. Types of free-space interconnects: (a) space-variant interconnection; (b) space-invariant interconnection.

Moreover, the parallel nature of optics and free-space propagation, together with their relative freedom from interference, makes them ideal for parallel communication, which is very much needed for parallel processing systems.

Types of optical interconnects. Several research projects on optical interconnection networks are now under way.[4,5] These projects address the use of free-space and waveguide interconnects.

Waveguide interconnects require physical paths for the optical signals. The paths can be constructed by integrating either fibers or waveguides into the planar substrate. The major drawbacks of fiber-based waveguides at the chip-to-chip and board-to-board levels are (1) the relatively low interconnection density, (2) lack of bending or looping flexibility, (3) crosstalk between adjacent waveguides, and (4) severe coupling losses. Densely connected networks that require many interconnection lines have a serious problem with bending and looping fibers, due to radiation losses induced by bending. For integrated waveguides, an important problem is the coupling of light into and out of the waveguides. Additionally, the density of a waveguide array is mainly limited by the coupling-induced crosstalk between adjacent waveguides. This crosstalk worsens in densely connected networks. Moreover, both fiber-based and integrated waveguide interconnects have, at the chip-to-chip and board levels, limited interconnectivity and few fan-in/fan-out channels, since they are similar to wires and therefore require physical paths between every two points. Again, however, these problems are being investigated by the optical communications community.

Free-space optical interconnects exploit free space — defined as a vacuum — for optical signal propagation. Lenses and holograms are typical optical elements that provide communication channels for this type of interconnect. Free-space interconnects have proven very suitable for chip-to-chip and board-level interconnects because (1) they have the potential to fully utilize the large spatial bandwidth and parallelism of optics, (2) they have no mutual interference (light beams can cross in space without interference), (3) they result in flexible network configurations because the interconnects are not confined to physical paths, and, most of all, (4) the interconnects can use simple imaging optics, which results in light-power efficiency, very low signal skew, and simpler architectures, and permits chip-like integration of optoelectronic circuits.

Optical interconnects are classified according to the degree of space variance,[6] which determines the network's complexity and regularity. A totally *space-variant* network allows a completely arbitrary interconnection between components, as shown in Figure 2a. A totally *space-invariant* network has a definite, regular structure whereby all the nodes have the same connection patterns, as shown in Figure 2b. There is a trade-off between the complexity of optical implementation and the resulting degree of connectivity and regularity achieved with the network. Space-invariant networks are well matched to the capabilities of optical components such as lenses, mirrors, and holograms, and are easy to implement. Space-variant networks, on the other hand, require complex optical implementations that often result in low interconnection density and high cost.

There is also a fundamental trade-off between the degree of space variance and the space-bandwidth product (SBWP). The SBWP of an optical system is 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. The SBWP of a linear optical system is equal to

$$\text{SBWP} = \frac{aA}{(\lambda f)^2} \qquad (1)$$

where λ is the wavelength of the light, f the focal length of the system, a the area of the input plane, and A the area of the spatial frequency plane. For wavelengths near visible light, this is approximately equal to 100 spatial degrees of freedom per square micron, which is about the same size as the limit for a single electronic device.

A completely space-invariant system has minimal SBWP requirements, whereas a completely space-variant system has extensive SBWP requirements. We know that the number of permissible channels in the space-variant system is proportional to the square root of the SBWP, while the number of permissible channels in the space-invariant system is directly proportional to the SBWP.[6] Therefore, space-invariant systems allow more channels — more parallelism — than space-variant systems for a given SBWP. As an example, Jenkins et al. showed that a space-invariant hologram can provide approximately 10^8 channels, whereas a space-variant hologram of the same size allows approximately 10^4 channels.[7]

3D free-space optical interconnects

Optical interconnect designs are being pursued concurrently in several different ways. Of particular interest to us here are network topologies that connect two-dimensional arrays of input nodes to 2D arrays of output nodes. These architectures exploit the high temporal, spectral, and spatial bandwidth of optical systems.

A model for optical interconnection network architectures. Figure 3 illustrates our proposed model. It consists of 2D PE arrays on facing planes with an optical interconnect module (OIM) between them, providing completely space-invariant connections.

Our model assumes that the PEs in the network are partitioned into two sets of equal size and that there is no inter-PE communication link between any two PEs on the same set (plane). Each plane contains electronic PEs with integrated optical sources and detectors. All inter-PE links between planes are enabled by free-space optics through the third dimension, which is perpendicular to the planar PE arrays. The third dimension allows circuit designers more layout flexibility than can be attained with the periphery of the board, which is all that is available with electrical interconnections technology.

Our proposed model would

1. better utilize the full SBWP of optical imaging systems;
2. exploit the parallelism of free-space optics;
3. be cost-effective because the beams, which will be directed orthogonal to the device substrate, would share the same imaging and beam-steering optics as the interconnects and, consequently, the cost of the optical hardware would be shared by many devices;
4. be compatible with recent advances in compact, 2D optical logic and switching, and optoelectronic integrated-circuit (OEIC) technologies; and
5. because of the additional, third dimension, open new possibilities for designing faster parallel processing algorithms.

Optical components for the 3D optical interconnect model. The proposed model consists of three optical components — sources, detectors, and an OIM. The source can be one of four types: (1) a directly modulated semiconductor laser, (2) a light-emitting diode, (3) a vertical-cavity surface-emitting laser,[8] which is a microlaser exceptionally well suited for the high-density source array, or (4) a modulator coupled with an external light source. The modulator can be electro-optic, absorptive, reflective, or even a spatial light modulator such as a self-electro-optic effect device.[9]

The detector is typically a photodiode or a photoconductor connected to a bias and preamplifier circuitry. Since a node might have at least one source, one detector, and some processing capability, the node could be a smart pixel, which is a hybrid optoelectronic digital processing element with three spatially separated functional units — one or more optical detectors to convert optical inputs into electronic signals, a processing unit that processes the received signal, and one or more optical sources to convert processed electronic outputs into optical signals.

The OIM in our model receives an image from the plane and generates multiple (M) images, which are simultaneously incident on the other plane. The locations of M image spots on the receiving plane are determined by the required connection patterns. Since we'd like the system to be completely space invariant, all the sources must have the same connection patterns. The connection pattern generated by the OIM determines the topology of the interconnection network.

In describing how the OIM functions, we define a *connection rule* to be the

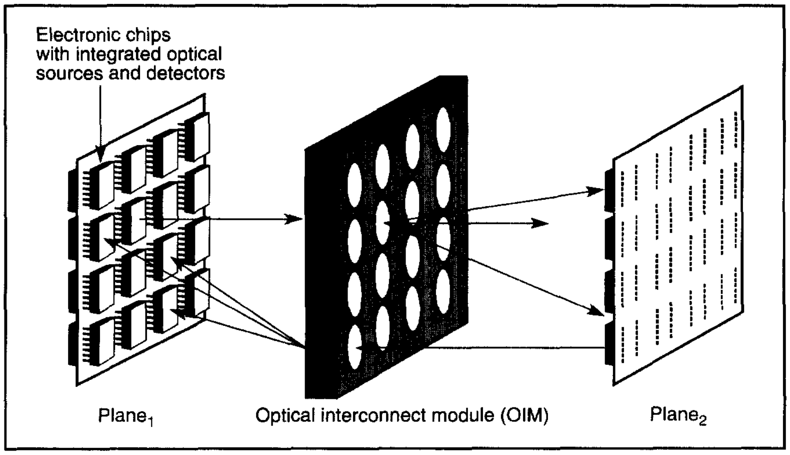


Figure 3. A model for free-space space-invariant optical interconnection network architecture.

amount of row-wise (up or down) or column-wise (left or right) spatial shift required to achieve an interconnection topology. Each connection rule for a network contains two entities, *Row* and *Col.* For example, a connection rule such as (Row = ±1 and Col = ±1) states that to implement a given network, each plane is to be replicated into four images. Each replica is then shifted as follows. For a Row = +1, the corresponding image is shifted upward by one row. Similarly, a Row = –1 indicates a downward shift by one row. A Col = +1 means a shift to the right by one column, and Col = –1 is a shift to the left by one column. There may be several shifts needed along a given direction, and that is indicated by a comma. For example, a connection rule such as Row = ±1, ±3, Col = ±1, ±3 would require eight images of each plane. These images are then shifted according to the displacement indicated. The shifted images need to be simultaneously superposed on the plane to achieve the required connections.

Multiple-image generation and spatial shifting of the OIM can be achieved through the use of beam-splitting devices or fan-out elements such as holographic optical elements. The amount of spatial shift will depend on the angle of the beam transmitted by these elements. The simultaneous superposition of the shifted images on the receiving plane, whether in whole or in part, determines the topology established by the OIM.

Network topologies considered. Interconnection density between boards depends on the number of PEs and the interconnect topologies used. A crossbar would be the fastest and most desirable network topology because it provides a dedicated channel from any PE to any other PE with only a single step. A crossbar network would also be the most demanding on the interconnection medium, since the number of required connections would vary as a square of the number of PEs involved. In terms of cost, a crossbar network is presently prohibitive for large systems, even with optical implementations. Additional research now under way may one day yield a cost-effective crossbar topology implementation.

Two other potential solutions to achieving high-density, high-bandwidth optical communication at a reduced cost are (1) a point-to-point multihop topology such as a hypercube network, and (2) a multistage interconnection network (MIN) —for example, Shuffle-Exchange, Omega, Banyan, Clos, Butterfly, or Benes, among others — that has several links and switching stages.

For the purposes of our model and this article, we examine a 3D free-space and space-invariant optical implementation of two point-to-point topologies — hypercube and mesh networks (see sidebar above). For recent work on the 3D optical implementation of MINs, see the literature.[4,5]

3D optical hypercube networks

Basis of the new design methodology. In some networks, nodes, or PEs, can be

Binary *n*-cube and mesh networks

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 network.[1] In a binary *n*-cube (*n* is referred to as the dimension of the binary cube) we have 2^n nodes each of degree *n*, where the degree of a node means the number of nodes directly connected to it. The attractiveness of the hypercube topology is its 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. Each node is numbered in such a way that there is a one-binary-digit difference between any node and its *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.[2]

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 with wraparound connections) per node. Due to the constant node degree, the mesh network is also highly scalable compared with the hypercube network.

References

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

2. J.P. Hayes and T. Mudge, "Hypercube Supercomputers," *Proc. IEEE*, Vol. 77, 1989, pp. 1,829-1,841.

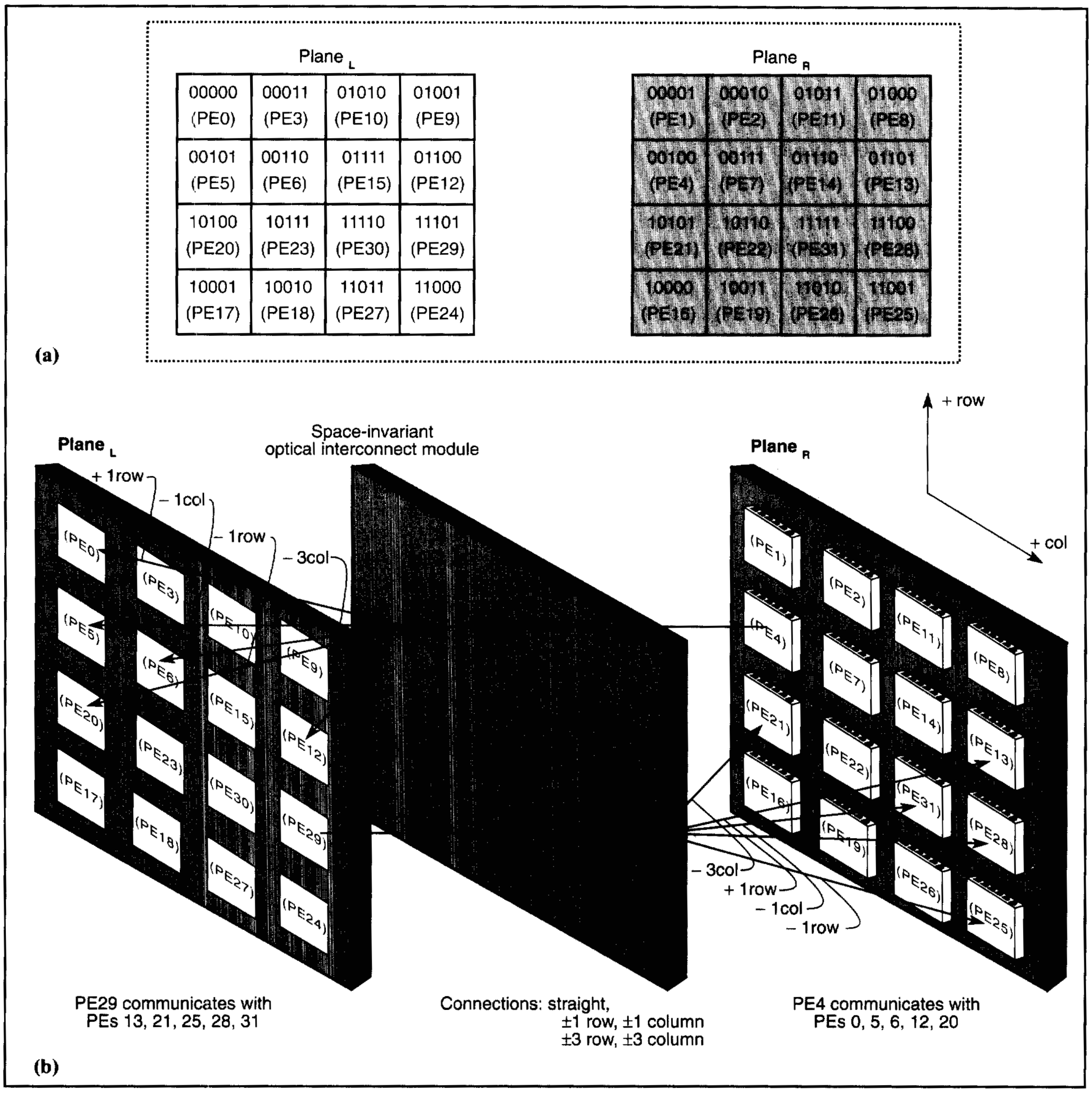
Definitions of scalability

The term *scalability* has three different meanings: problem scalability, size scalability, and generation scalability.[1] A scalable problem is one that can be made large enough to operate efficiently on a computer with a given granularity. Thus, scalability in this context is referring to the efficiency of an algorithm to solve a given problem when the number of processing elements of a computer is increased. In architectural terms, a size-scalable computer is designed to have a scaling range from few to many resource components with nominal changes in the existing system configuration, such as number of communication links per processing element. Linearly increased performance is expected as the system size grows.

Equally as important as size scalability is generation scalability, which is adaptability of the architecture to the rapid evolution of technologies. Since microprocessors become obsolete every three years and the time to find efficient algorithms for a new system is long, a significant portion of the investment in a new architecture should be preserved throughout successive generations. In a generation-scalable microprocessor, the instruction set would be carefully considered when anticipating the nature of future implementations.

Reference

1. G. Bell "Ultracomputers: A Teraflop Before Its Time," *Comm. ACM*, Vol. 35, No. 8, Aug. 1992, pp. 27-47.



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)

Figure 4. Conceptual realization of a 3D space-invariant five-cube network using the proposed model: (a) the 32 nodes of the five-cube network in 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.

partitioned into two sets such that any two nodes in the same set do not share a common link. For the binary n-cube, nodes whose addresses differ by more than one in Hamming distance (no link exists between the two nodes) can be in the same partition. Furthermore, we can arrange the nodes in each partition on the plane so that interconnections between the two planes are entirely space invariant.

A conceptual 3D implementation of a five-cube network, based on the proposed model, is shown in Figure 4. Figure 4a illustrates the space-invariant embedding of a five-cube network. Notice that the connection pattern of nodes from the left plane (plane$_L$) is identical to that of the right plane (plane$_R$). Since the links are bidirectional, all nodes on the right plane have exactly the same connection pattern to the nodes on the left plane. Another way to consider this is to say that the connection pattern from any source node to destination nodes is identical for all nodes. A number in a node on a plane represents the binary address of the corresponding node.

Figure 4b shows the conceptual interconnection of a 3D five-cube network. Here, the required connections are obtained by superposing nine images of one plane onto the other plane; eight are spatially shifted and one is directly imaged onto the receiving plane. Each optical signal emanating from a transmitting node is subject to the same spatial shifting, regardless of its position in the plane (space invariant). The amount of spatial shift is ±1d and ±3d in both row-wise and column-wise directions, where d is the size of a node in a single dimension and the origin is taken to be the center of the plane. Our connection rule for the n-cube network is

	Plane $_L$	Plane $_R$	Connection rule	Dimension
2-cube	00 (0) 11 (3)	01 (1) 10 (2)	Row(2) = 0 Col(2) = ±1	$D_r(2) = 1$ $D_c(2) = 2$
3-cube	000 (0) 011 (3) 101 (5) 110 (6)	001 (1) 010 (2) 100 (4) 111 (7)	Row(3) = 0, ±1 Col(3) = ±1	$D_r(3) = 2$ $D_c(3) = 2$
4-cube	0000 (0) 0011 (3) 1010 (10) 1001 (9) 0101 (5) 0110 (6) 1111 (15) 1100 (12)	0001 (1) 0010 (2) 1011 (11) 1000 (8) 0100 (4) 0111 (7) 1110 (14) 1101 (13)	Row(4) = 0, ±1 Col(4) = ±1, ±3	$D_r(4) = 2$ $D_c(4) = 2$
5-cube	00000 (0) 00011 (3) 01010 (10) 01001 (9) 00101 (5) 00110 (6) 01111 (15) 01100 (12) 10100 (20) 10111 (23) 11110 (30) 11101 (29) 10001 (17) 10010 (18) 11011 (27) 11000 (24)	00001 (1) 00010 (2) 01011 (11) 01000 (8) 00100 (4) 00111 (7) 01110 (14) 01101 (13) 10101 (21) 10110 (22) 11111 (31) 11100 (28) 10000 (16) 10011 (19) 11010 (26) 11001 (25)	Row(5) = 0, ±1, ±3 Col(5) = ±1, ±3	$D_r(5) = 4$ $D_c(5) = 4$

Figure 5. Embedding of binary n-cubes ($2 \le n \le 5$) onto planes, using the proposed model.

Row(n) and Col(n). Therefore, the connection rule for the five-cube network is Row(5) = 0, ±1, ±3 and Col(5) = ±1, ±3.

Recall that the communication pattern from plane$_L$ to plane$_R$ is identical to that from plane$_R$ to plane$_L$. The nine images are simultaneously incident on the receiving plane in which a receiving node acquires five different optical images (signals) that represent the required hypercube connections. The remaining four images fall outside the receiving plane and are discarded. Since each node receives several signals simultaneously, nodes must have a means of identifying the different incoming signals. Incoming signals can be separated by using wavelength-, space-, or time-division multiplexing techniques. We explain incoming-signal separation techniques in more detail elsewhere.[10]

Design methodology for 3D space-invariant hypercube networks. As described above, our proposed embedding scheme can implement hypercube networks with as many as five dimensions using the space-invariant multiple-imaging system. No empty rows or columns need be introduced on the planes. However, for networks with more than five dimensions, empty rows and columns are required to mask off unwanted connections. In doing so, we maintain a completely space-invariant connection pattern. It's doubtful whether hypercube networks can be embedded into 2D planes without empty rows and columns and still achieve entirely space-invariant connections. Through the exhaustive search for the solution of the six-cube and the seven-cube, we have come to believe that the five-cube is the maximal dimension that can be embedded using the proposed model without inserting empty rows and columns.

Below, we introduce a general methodology for constructing 3D space-invariant n-cube networks where n is greater than five. The proposed embedding scheme for a five-cube can be modified by inserting empty rows and columns to mask off unwanted interconnections for hypercubes with more than five dimensions. Construction of an arbitrary n-cube network is based on the space-invariant (n – 1)-cube network. Figure 5 presents the optical implementation of n-cube networks for n = 2, . . . , 5. These implementations represent basic modules meant to be used in constructing still larger networks. Table 1 defines the symbols we use in describing our methodology.

An algorithm to construct a 3D space-invariant n-cube using the proposed model. The following three-step algorithm constructs a 3D space-invariant n-cube (n > 5) from a space-invariant (n – 1)-cube network. We should note that an n-cube network can be constructed from two (n – 1)-cube networks.

Step 1: Given plane$_L$ and plane$_R$ of a space-invariant (n – 1)-cube, and depending on whether n is odd or even, we rotate each plane to the left by the following number of columns if n is even:

$$R_c(n) = 2^{\frac{n-4}{2}} \tag{2}$$

or we rotate each plane upward by the following number of rows if n is odd:

$$R_r(n) = 2^{\frac{n-5}{2}} \tag{3}$$

Step 2: The rotated plane is then placed at the right side of the original $(n - 1)$-cube of the opposite plane if n is even, or underneath if n is odd. During the rotation, no empty columns or rows that already exist in the $(n - 1)$-cube plane are counted as the shift amount.

We insert the following number of empty columns or rows between the two planes, the original and the rotated one:

$$\varepsilon_c(n) = 2^{\frac{n-6}{2}} + \sum_{i=1}^{\frac{n-6}{2}} \varepsilon_c(2i+4) \text{ when } n \text{ is even} \tag{4}$$

$$\varepsilon_r(n) = 2^{\frac{n-7}{2}} + \sum_{i=1}^{\frac{n-7}{2}} \varepsilon_r(2i+5) \text{ when } n \text{ is odd} \tag{5}$$

where $\sum_{i=1}^{0}$ is defined to be 0.

Note that this insertion is done for plane$_L$ of the $(n - 1)$-cube and the rotated version of plane$_R$ of the $(n - 1)$-cube, and for plane$_R$ of the $(n - 1)$-cube and the rotated version of plane$_L$ of the $(n - 1)$-cube.

Step 3: We prefix 0 as the most significant bit in all addresses of nodes on the resulting plane$_L$, and 1 as the most significant bit in all addresses of nodes on the resulting plane$_R$.

When n is even, plane$_L$ and plane$_R$ for the space-invariant n-cube have the same row dimensions as those of the $(n - 1)$-cube, and column dimensions are 2 × (column dimension of the $(n - 1)$-cube) + (the number of empty columns inserted in step 2). By row dimension and column dimension, we mean the number of rows including empty rows and the number of columns including empty columns, respectively. Thus,

Table 1. The meaning of symbols used in the generalized embedding algorithm.

Notation	Meaning
Plane$_L$	A plane located on the left side of the model, on which one of two node partitions is placed.
Plane$_R$	A plane located on the right side of the model, on which one of two node partitions is placed.
$\varepsilon_r(n)$*	The number of empty rows inserted between two layouts of $(n - 1)$-cubes on the plane to construct an n cube.
$\varepsilon_c(n)$*	The number of empty columns inserted between two layouts of $(n - 1)$-cubes on the plane to construct an n-cube.
$D_r(n)$*	The row dimension** of the resulting n-cube on the plane.
$D_c(n)$*	The column dimension** of the resulting n-cube on the plane.
$R_r(n)$*	The amount of upward rotation of an $(n - 1)$-cube layout on the plane to construct an n-cube.
$R_c(n)$*	The amount of left rotation of an $(n - 1)$-cube layout on the plane to construct an n-cube.
Row(n)	The number of row-wise shifts for implementing an n-cube.
Col(n)	The number of column-wise shifts for implementing an n cube.

* This number is for each plane (plane$_L$, plane$_R$).
** Row (column) dimension means the number of rows (columns) on the plane.

$$\begin{aligned} D_r(n) &= D_r(n-1) \\ D_c(n) &= 2 \times D_c(n-1) + \varepsilon_c(n) \end{aligned} \tag{6}$$

When n is odd, plane$_L$ and plane$_R$ for the space-invariant n-cube have row dimensions equal to 2 × (row dimension of the $(n - 1)$-cube) + (the number of empty rows inserted in step 2), and the same column dimensions as those of the $(n - 1)$-cube. Thus

$$\begin{aligned} D_r(n) &= 2 \times D_r(n-1) + \varepsilon_r(n) \\ D_c(n) &= D_c(n-1) \end{aligned} \tag{7}$$

If n is even, the connection rule of the resulting n-cube is

$$\begin{aligned} \text{Row}(n) &= \text{Row}(n-1) \\ \text{Col}(n) &= \text{Col}(n-1), \\ &\pm [D_c(n) - D_c(n-3)] \end{aligned} \tag{8}$$

If n is odd, the connection rule of the resulting n-cube is

$$\begin{aligned} \text{Row}(n) &= \text{Row}(n-1), \\ &\pm [D_r(n) - D_r(n-3)] \\ \text{Col}(n) &= \text{Col}(n-1) \end{aligned} \tag{9}$$

Since we assume bidirectional communication between two planes, the connection rule applies to both planes.

Example: Construction of 3D optical space-invariant six-cube networks using the generalized embedding algorithm. Now let's see how we would construct a six-cube network with the embedding algorithm. Figure 6 depicts the construction of a 3D space-invariant six-cube developed from the basic five-cube. First, since n is even, each basic plane shown in Figure 6a of a five-cube is rotated to the left by two columns. The amount of rotation is generally determined by Equation 2. The resulting planes are shown in Figure 6b. Next, each rotated plane is situated to the right of the original one on the opposite plane, as shown in Figure 6c. Following rotation, empty columns are inserted between the two planes (the original plane and the rotated plane). The number of empty columns is determined by Equation 4. In this example, the number of empty columns inserted is one. Finally, a 0 is prefixed as the most significant bit in every node address on plane$_L$, and a 1 is prefixed as the most significant bit in every

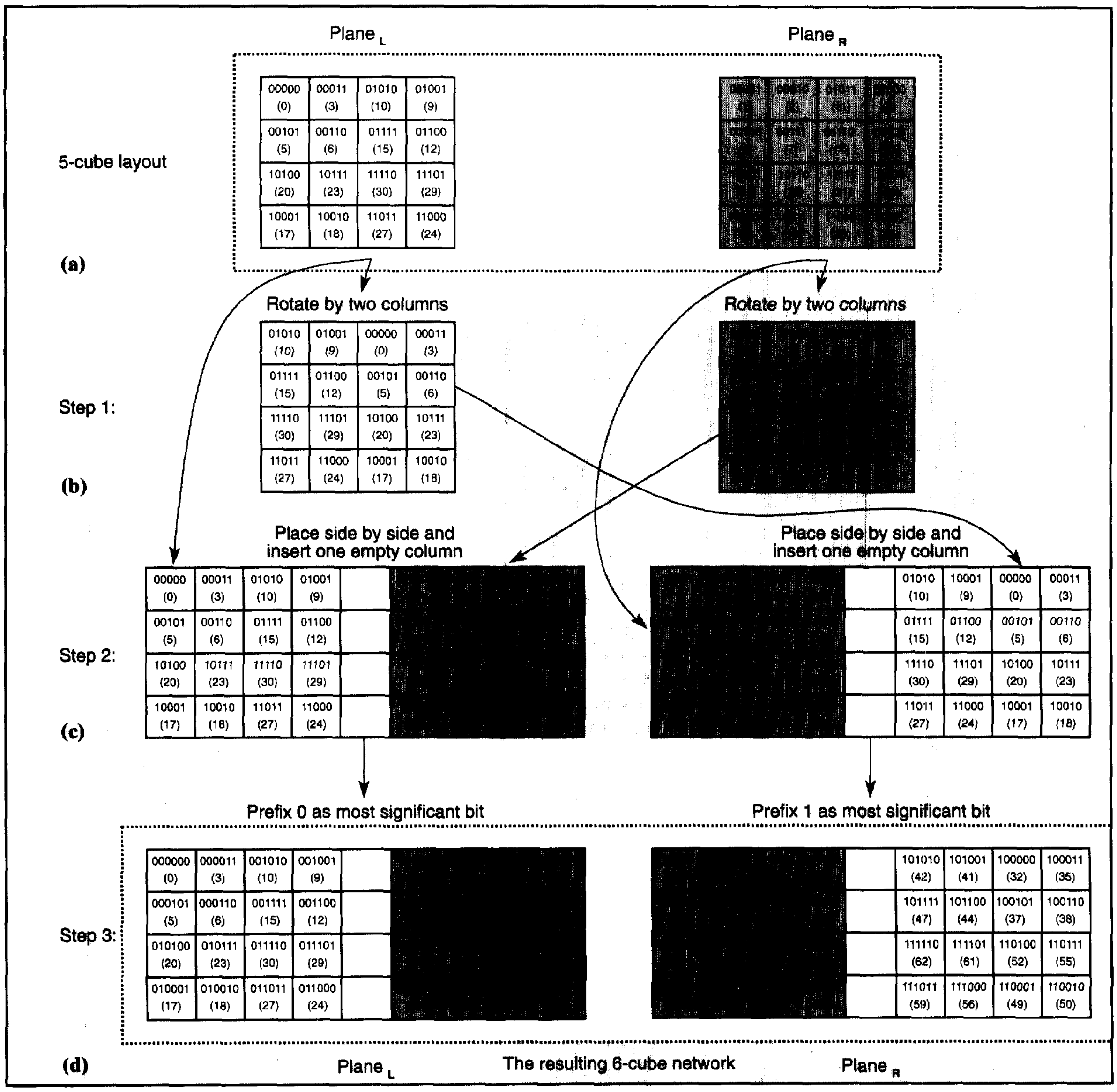


Figure 6. Construction of a 3D space-invariant six-cube network from a 3D space-invariant five-cube network. The connection rule is (Row(6) = Row(5) = 0, ±1, ±3) and (Col(6) = Col(5), ±7 = ±1, ±3, ±7).

node address on plane$_R$. The resulting 3D space-invariant six-cube is shown in Figure 6d. The connection rule for the resulting network is

$$\begin{aligned} \mathrm{Row}(6) &= \mathrm{Row}(5) &&= 0, \pm 1d, \pm 3d \\ \mathrm{Col}(6) &= \mathrm{Col}(5), \pm 7d &&= \pm 1d, \pm 3d, \pm d \end{aligned}$$

3D optical mesh networks

In this section, we show how a mesh network would be handled with a 3D space-invariant optical implementation. A d-dimensional mesh of size $k_1 \times k_2 \times \ldots \times k_d$ has nodes with d address components; $\{(a_1, \ldots, a_d) | 0 \le a_i < k_i \text{ for } 1 \le i \le d\}$, where a link exists between two nodes if and only if their address components differ by one in one component and are identical in all other components. This rule does not include wraparound connections of nodes on the edges. The first component of a node address is referred to as the dth dimensional link, the second as the $(d-1)$th-dimensional link, and the last as the first-dimensional link. For example, a three-dimensional mesh of size $2 \times 4 \times 4$ has $2 \times 4 \times 4 = 32$ nodes with three address components (k_1, k_2, k_3), where $0 \le k_1 < 2$, $0 \le k_2 < 4$, and $0 \le k_3 < 4$.

A 3D embedding of a three-dimensional mesh with size $2 \times 4 \times 4$ (32 nodes) based on the model is shown in Figure 7, where each plane consists of 16 nodes. We partition 32 nodes into two sets, each of which has 16 nodes, such that all nodes whose arithmetic sums of the three address components (that is, $k_1 + k_2 + k_3$ for node (k_1, k_2, k_3)) are even belong to one partition and the rest to the other partition. For example, nodes (0,0,0) and (1,0,3) are in the same partition (plane$_L$), since the arithmetic sums of three address components are even. Nodes (1,0,0) and (0,0,3) can also be in a common partition (plane$_R$), but this must be a different partition distinct from where nodes (0,0,0) and (1,0,3) belong, since the arithmetic sums of three components are now odd.

The figure shows that the connection rule of Row = ±1 and Col = ±1 makes two independent two-dimensional meshes with size 4 × 4, one mesh with unshaded nodes and another mesh with shaded nodes. Now, the straight connection between the two planes makes the third-dimensional links between the two meshes form a three-dimensional mesh of size 2 × 4 × 4. Note that the model can implement a three-dimensional mesh if the size of the third dimension is limited by two; however, it can realize any arbitrary size of the two-dimensional mesh.

The connection rule, which is independent of the size of the mesh, is Row = 0, ±1 and Col = ±1, where 0 means the straight connection. For the $2 \times l \times m$ mesh, the connection rule of Row = $\pm(l-1)$, Col = $\pm(m-1)$ provides wraparound connections.

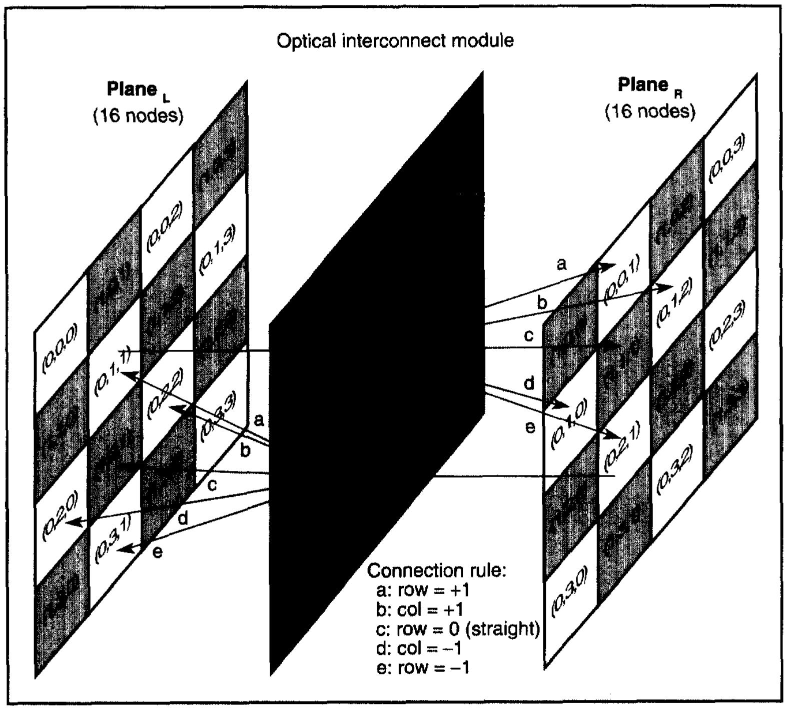


Figure 7. Embedding of a three-dimensional mesh with size 2 × 4 × 4. The figure shows an instance of communication from node (0,1,1) to nodes (1,1,1), (0,0,1), (0,1,0), (0,2,1), and (0,1,2) as well as from node (0,2,1) to nodes (1,2,1), (0,1,1), (0,2,2), (0,2,0), and (0,3,1).

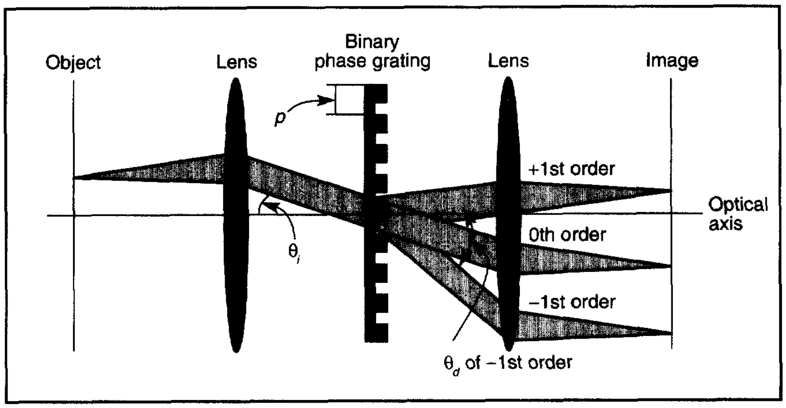


Figure 8. Space-invariant computer-generated binary phase grating.

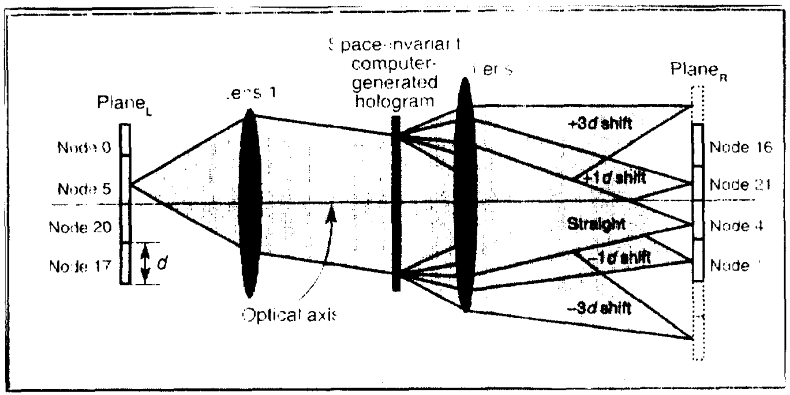


Figure 9. Side view of five-cube implementation using computer-generated holograms.

Optical implementation of 3D hypercube network OIMs

Here we discuss optical implementations of OIMs for the 3D space-invariant hypercube networks based on the design methodologies presented in this article. The implementation discussions are intended to be general in nature. A wide variety of optical techniques exist that can be used to develop the OIM. These include phase gratings, pinhole arrays, beam splitters, multisplit lenses, lenslet arrays, off-axis lens arrays, mirror arrays, and holographic techniques. We present a binary-phase-grating technique for hypercube implementation.

The binary phase grating is a subclass of the diffraction grating wherein the effect of a grating on an optical beam is well described by the laws of diffraction. It is called *binary* because the amplitude transmittance has only two levels — zero and one. Since the computer helps to design and fabricate the binary phase grating, it is also called a computer-generated hologram.[11] Due to diffraction, the optical power within a beam directed at a binary phase grating will be redirected from the grating into several orders that are symmetric about the zero-order beam, as shown in Figure 8.

The relationship between the angle of the incident beam (θ_i), the period of the grating (p), the wavelength of the light (λ), the grating order (m), and the angle of the mth order's diffracted beam (θ_d) is shown in Equation 10.[12]

$$p(\sin\theta_d - \sin\theta_i) = m\lambda \qquad (10)$$

We can alter the amount of optical power routed from the original beam into the different orders by changing the periodic structure of the grating. To have different angular spacings, we should change the period of the grating.

Figure 9 illustrates a five-cube implementation using a binary phase grating. For clarity, only a 2D side view of the five-cube is shown where an instance of communication, from node 5 to nodes 1, 4, and 21, is illustrated (a second grating oriented 90 degrees with respect to the first grating shown in the figure is not depicted due to the 2D side view). It should be noted that the address-node assignment is from bottom to top on plane$_R$ due to image inversion by the lenses.

Because the connection pattern is totally space invariant, a single hologram can provide all of the required connections for each node. According to the connection rule for the 3D space-invariant five-cube, we need one straight beam and eight diffracted beams — four diffracted beams per axis. Since the figure depicts a 2D side view, the straight beam and the four diffracted beams are shown. The 0th-order beam corresponds to communication from node 5 to node 4, +1st order from node 5 to node 21, and −1st order from node 5 to node 1. ±3rd-order diffracted beams miss the plane$_R$ and are discarded.

To estimate the theoretical upper bound of the hypercube size that can be embedded into a given plane by the proposed architecture, we need to calculate the SBWP of the proposed architecture. There are several factors that determine the upper bound, such as diffraction-limited pixel size on the receiving plane, the diffraction angle at the BPG, aberration, crosstalk, and so forth. Only diffraction-limited pixel size is considered to show the upper bound of the hypercube size.[10]

Despite many attractive features of holograms for optical interconnects — ease of beam routing, small feature size, and the ability to provide numerous fan outs — several technological problems, such as wavelength-stabilized light sources and fabrication-integration technologies, remain to be solved for cost-effective implementations.

Although faster device switching speeds will eventually be necessary for future massively parallel computing systems, the deciding factor in determining system performance and cost will be subsystem communications rather than device speed. Free-space optical interconnects, by virtue of their inherent parallelism, high data bandwidth, small size and power requirement, and relative freedom from mutual interference of signals, already show great promise in replacing metal interconnects to solve communication problems. ■

Acknowledgments

This research was supported by NSF Grant No. MIP 9310082 and a grant from USWest.

References

1. R.A. Nordin et al., "A System Perspective on Digital Interconnection Technology," *IEEE J. Lightwave Technology*, Vol. 10, June 1992, pp. 811-827.

2. H.S. Stone and J. Cocke, "Computer Architecture in the 1990s," *Computer*, Vol. 24, No. 9, Sept. 1991, pp. 30-38.

3. F. Kiamilev et al., "Performance Comparison Between Optoelectronic and VLSI Multistage Interconnection Networks," *IEEE J. Lightwave Technology*, Vol. 9, Dec. 1991, pp. 1,665-1,674.

4. "Optical Interconnections for Information Processing," special issue, *IEEE J. Lightwave Technology*, Vol. 9, Dec. 1991.

5. A. Louri and H. Sung, "An Optical Multi-mesh Hypercube: A Scalable Optical Interconnection Network for Massively Parallel Computing," *IEEE J. Lightwave Technology*, Vol. 12, No. 4, Apr. 1994, pp. 704-716.

6. G.E. Lohman and K.H. Brenner, "Space-Invariance in Optical Computing Systems," *Optik*, Vol. 89, 1992, pp. 123-134.

7. B.K. Jenkins et al., "Architectural Implications of a Digital Optical Processor," *Applied Optics*, Vol. 23, Oct. 1984, pp. 3,465-3,474.

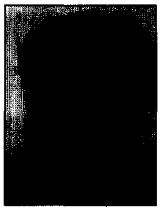
8. J.L. Jewell et al., "Surface-Emitting Microlasers for Photonic Switching and Interchip Connections," *Optical Eng.*, Vol. 29, Mar. 1990, pp. 210-214.

9. J.A. Neff, R.A. Athale, and S.H. Lee, "Two-Dimensional Spatial Light Modulators: A Tutorial," *Proc. IEEE*, Vol. 78, May 1990, pp. 836-855.

10. A. Louri and H. Sung, "Efficient Implementation Methodology for Three-Dimensional Space-Invariant Hypercube-Based Free-Space Optical Interconnection Networks," *Applied Optics*, Vol. 32, Dec. 1993, pp. 7,200-7,209.

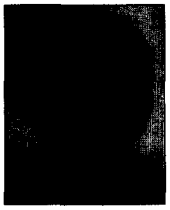
11. K.S. Urquhart et al., "Computer-Aided Design of Computer-Generated Holograms for Electron Beam Fabrication," *Applied Optics*, Vol. 28, 1989, pp. 3,387-3,396.

12. E. Hecht, *Optics*, Addison-Wesley, Reading, Mass., 1987.



Ahmed Louri is an associate professor in the Department of Electrical and Computer Engineering at the University of Arizona and director of the Optical Computing and Parallel Processing Laboratory. His research interests include computer architecture, parallel processing, optical computing, and optical interconnects. In 1988 he was the recipient of the NSF Research Initiation Award. In 1991, he received the "Best Article of 1991 Award" from *IEEE Micro*.

Louri earned the Diplome D'engenieur (engineer degree) from the University of Science and Technology, Oran, Algeria, in electrical engineering, and a master's and a PhD in computer engineering from the University of Southern California. He is a member of IEEE, ACM, the Optical Society of America, and the Society of Photooptical Instrumentation Engineers.



Hongki Sung is a research associate in the Optical Computing and Parallel Processing Laboratory at the University of Arizona, Tucson, Arizona. His research interests include interconnection networks, optical interconnects, cache memories, and parallel computer architecture. Sung received a BS in electronics engineering from Seoul National University in Korea in 1984, an MS in computer engineering from the University of Southwestern Louisiana in 1990, and a PhD in electrical engineering from the University of Arizona in 1994. He is a member of the IEEE Computer Society.

Readers can contact Louri at the University of Arizona, Dept. of Electrical and Computer Engineering, Tucson, AZ 85721. His e-mail address is louri@ece.arizona.edu.