Optical binary de Bruijn networks for massively parallel computing: design methodology and feasibility study

Ahmed Louri and Hongki Sung

The interconnection network structure can be the deciding and limiting factor in the cost and the performance of parallel computers. One of the most popular point-to-point interconnection networks for parallel computers today is the hypercube. The regularity, logarithmic diameter, symmetry, high connectivity, fault tolerance, simple routing, and reconfigurability (easy embedding of other network topologies) of the hypercube make it a very attractive choice for parallel computers. Unfortunately the hypercube possesses a major drawback, which is the complexity of its node structure: the number of links per node increases as the network grows in size. As an alternative to the hypercube, the binary de Bruijn (BdB) network has recently received much attention. The BdB not only provides a logarithmic diameter, fault tolerance, and simple routing but also requires fewer links than the hypercube for the same network size. Additionally, a major advantage of the BdB network is a constant node degree: the number of edges per node is independent of the network size. This makes it very desirable for large-scale parallel systems. However, because of its asymmetrical nature and global connectivity, it poses a major challenge for VLSI technology. Optics, owing to its three-dimensional and global-connectivity nature, seems to be very suitable for implementing BdB networks. We present an implementation methodology for optical BdB networks. The distinctive feature of the proposed implementation methodology is partitionability of the network into a few primitive operations that can be implemented efficiently. We further show feasibility of the presented design methodology by proposing an optical implementation of the BdB network.

Key words: de Bruijn network, perfect shuffle, optical implementation, parallel processing.

1. Introduction

The choice of the interconnection network structure is critical in the design of parallel computers because communication between processors (for multicomputers) and between processors and memory modules (for multiprocessors) dominate the cost of the machine, the power budget, the hardware (wiring, packaging, etc.), and the overall performance.[1–4] Many topologies have been explored for parallel computers, including multistage interconnection networks such as omega, baseline, banyan, and crossover, and point-to-point interconnection networks such as hypercube, mesh, ring, bus, and star.[3,5] Currently, two of the most popular point-to-point topologies are the binary n-cube or hypercube[6–9] and the mesh.[10–13] The hypercube topology is completely symmetric with a logarithmic diameter (the diameter of a network is defined to be the largest number of hops in the shortest path between any two nodes. For an N-node network the diameter is $\log_2 N$). It is also fault tolerant and has very simple routing algorithms. However, the shortcoming of the hypercube network is the complexity of the node structure: the number of links per node grows as the network size increases. This property makes it very difficult to construct large-scale systems with the hypercube topology. The second popular network is the mesh topology. The mesh can be implemented easily because of the simple regular connection and the small number of links (four) per node (constant node degree). Because of its constant node degree, the mesh is highly scalable. The mesh network also suffers from a major limitation, which is its large diameter ($N^{1/2}$ for an N-node network), along with its limited connectivity.

As an alternative to the hypercube and the mesh topologies, the de Bruijn topology[14,15] has recently

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

Received 13 February 1995; revised manuscript received 12 May 1995.

0003-6935/95/296714-09$06.00/0.

received much attention. Its properties and applications have been studied by several researchers.[2,16–19] Its topological properties show that the de Bruijn network is a good candidate for next-generation interconnection networks of parallel computers after the hypercube. The de Bruijn network behaves as the hypercube and retains most of its desired properties (logarithmic diameter, fault tolerance, and simple routing). The de Bruijn network possesses two major additional advantages. The first advantage is that the de Bruijn network requires fewer physical links than the hypercube for the same network size (the same number of nodes). For example, for a network of 1024 nodes, the hypercube network requires 5120 physical links, whereas the de Bruijn network requires only 2048 links. The second major property of the de Bruijn network is that the node degree is constant, whereas in the hypercube the node degree varies as $\log_2 N$ for an N-node network. For a binary de Bruijn (BdB) network the node degree is always four regardless of the network size. The node degree of the mesh network is also four, independent of the network size, but the BdB network has a much smaller diameter than the mesh for the same network size.

Recent work has also shown that most of the algorithms proposed for the hypercube network can be easily transposed onto the de Bruijn network without any increase in the complexity of the algorithms.[16] This, coupled with a constant node degree, makes the de Bruijn network a highly desirable interconnection architecture for future large-scale systems.

Despite its many attractive properties, the de Bruijn network is considerably less known compared with the hypercube network because it is much less amenable to VLSI implementations. The VLSI implementation of the de Bruijn network is nontrivial because the network is neither fully symmetric nor modular,[16,17] as is the case with other popular networks. Additionally, the de Bruijn network requires many more global connections than the hypercube and the mesh, and such global connections make its VLSI implementation more difficult. Currently, the de Bruijn topology is used in a few parallel machines: the Triton/1 computer, developed at the University of Karlsruhe,[20] and the de Bruijn VLSI network, with 8192 nodes, which is being built by NASA's Galileo project.[17]

Optics, owing to its three-dimensional (3-D) nature, global-connectivity property, and flexible signal-routing capability, seems to be very suitable for realizing nonsymmetric global connections.[21–26] In this paper we propose an implementation methodology for the optical de Bruijn network. The proposed methodology provides a partitionable optical implementation; i.e., the de Bruijn network is first decomposed into a few primitive operations, each of which can be efficiently implemented, and then these operations are combined together to realize the de Bruijn network. An optical implementation of the de Bruijn network is proposed to show feasibility of the design methodology. It is shown that a BdB network with 4096 nodes can be integrated in a 4-cm^2 area with the total power efficiency being as high as 48%.

2. Definition and Properties of Binary de Bruijn Networks

A binary de Bruijn network with 2^n nodes is denoted by n-BdB. Let node i $(0 \le i < 2^n)$ in the n-BdB be represented by an n-bit binary number, say $i = a_{n-1}a_{n-2}\cdots a_0$. Node i is connected to four neighboring nodes (i_1, i_2, i_3, and i_4) as follows:

$$i_1 = a_{n-2}a_{n-3}\cdots a_1a_0a_{n-1}$$
$$\text{(rotate the node } i \text{ address one bit to the left)}, \qquad (1)$$

$$i_2 = a_{n-2}a_{n-3}\cdots a_1a_0\bar{a}_{n-1}$$
$$\text{(rotate the node } i \text{ address one bit to the left and complement the bit)}, \qquad (2)$$

$$i_3 = a_0a_{n-1}a_{n-2}\cdots a_2a_1$$
$$\text{(rotate the node } i \text{ address one bit to the right)}, \qquad (3)$$

$$i_4 = \bar{a}_0a_{n-1}a_{n-2}\cdots a_2a_1$$
$$\text{(rotate the node } i \text{ address one bit to the right and complement the bit)}. \qquad (4)$$

Node i_1 connection from node i in Eq. (1) is obtained by rotation of the node i address to the left by one bit position, which is equivalent to the perfect-shuffle (PS) operation. The node i_2 connection from node i in Eq. (2) is obtained by rotation of the node i address to the left and then complementing of the least-significant bit, which is equivalent to a perfect-shuffle–exchange operation. Similarly, the node i_3 connection from node i in Eq. (3) is obtained by a right-rotation operation, which is equivalent to the inverse perfect-shuffle (IPS) operation, and the node i_4 connection from node i is obtained by a right rotation and a complement operation or an inverse perfect-shuffle–exchange operation. In Fig. 1 a four-BdB network is shown. It should be noted that the BdB network is not modular (i.e., we cannot build a four-BdB net-

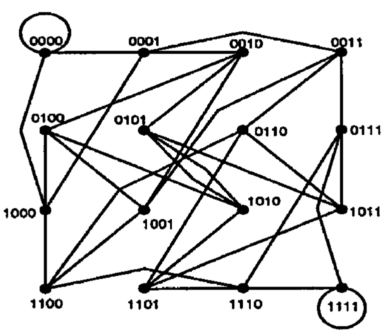

Fig. 1. Four-BdB network with 16 nodes. Node addresses are represented by binary numbers. A four-BdB network cannot be built simply by connection of two three-BdB networks because the BdB network is neither fully symmetrical nor modular.

work simply by connecting two three-BdB networks, as is the case with the hypercube network), it is not fully symmetric as the network size grows, and the connectivity is not localized (as is the case with the mesh network).

A node in the BdB network has four neighbors, as defined in Eqs. (1)–(4). Thus the node degree of an n-BdB network is always 4, which is constant and independent of the network size. In actual implementation the node degree means the number of fan-ins or fan-outs. Thus the fact that the node degree is constant greatly eases the design of large-scale systems that the BdB network compared with the hypercube-based one, whose node degree grows logarithmically with respect to the network size. As can be seen from Eqs. (1)–(4), a node in the n-BdB network can be reached from any other node in at most n hops. Thus the diameter of the n-BdB network with 2^n nodes is n (the diameter increases logarithmically with respect to the total number of nodes in the network). Table 1 compares major topological properties of the BdB network with several popular networks; the last column (constant edge length) indicates whether the given topology can be realized with edges (links) of the same length.

Table 1. Characteristics of Various Network Topologies[a]

Network	Nodes	Diameter	Constant Node Degree	Constant Edge Length[b]
1-D Mesh	n	$n - 1$	yes	yes
2-D Mesh	n^2	$2(n - 1)$	yes	yes
3-D Mesh	n^3	$3(n - 1)$	yes	yes
Binary tree	$2^n - 1$	$2(n - 1)$	yes	no
Quaternary hypertree	$2^n(2^{n+1} - 1)$	$2n$	yes	no
Pyramid	$(4n^2 - 1)/3$	$2 \log n$	yes	no
Butterfly	$(n + 1)2^n$	$2n$	yes	no
Hypercube	2^n	n	no	no
Cube-connected cycle	$n2^n$	$2n$	yes	no
Shuffle–exchange	2^n	$2n - 1$	yes	no
de Bruijn	2^n	n	yes	no

[a] Adapted from Quinn, *Parallel Computing: Theory and Practice* (McGraw-Hill, New York, 1993).

[b] A network has a constant edge length if all the edges (links) can be realized with the same length.

3. Design Methodology for Optical de Bruijn Networks

In this section we propose a design methodology for the optical implementation of the BdB networks. The presented methodology provides a partitionable optical implementation; the BdB network is decomposed into a few primitive operations that can be efficiently implemented, and then these operations are combined together to realize the BdB network. The design methodology assumes a 3-D optical-interconnect model, which consists of three parts: a two-dimensional (2-D) source array, a 2-D detector array, and an optical-interconnect module.[27] The optical-interconnect module receives an image from the source array and generates the required optical links to the detector array.

A. Decomposition of the de Bruijn Network into Primitive Optical Operations

As shown in Eqs. (1)–(4), a BdB network can be decomposed into four operations: a perfect shuffle operation, a perfect shuffle–exchange operation, an inverse perfect-shuffle operation, and an inverse perfect-shuffle–exchange operation. Because the model for the 3-D optical interconnects takes an image of the 2-D source array and generates images on the 2-D detector array, these operations and their corresponding shuffle operations should be done on 2-D arrays. There are two types of 2-D perfect shuffles[28–35]: the 2-D separable perfect shuffle (SPS) and the 2-D folded perfect shuffle (FPS). In the 2-D SPS the rows and the columns are shuffled independently, whereas in the 2-D FPS the rows and the columns of the input are obtained by folding a one-dimensional (1-D) input array. The mathematical relationship between the 2-D SPS and the 2-D FPS is shown in Fig. 36. In this subsection we first summarize the mathematical relationship between the 2-D SPS and the 2-D FPS derived in Fig. 36, and then we extend it to derive the relationship between the 2-D separable inverse perfect shuffle (SIPS) and the 2-D folded inverse perfect shuffle (FIPS). Then we identify the most fundamental three operations required for the BdB network construction.

Let us consider that N nodes ($N = 2^n$ and n is even) are arranged in a $2^{n/2} \times 2^{n/2}$ array (or a 2-D plane). A binary address of a node can be represented by $(a_{n-1}a_{n-2}\cdots a_{n/2}, a_{n/2-1}\cdots a_1a_0)$, where $a_{n-1}a_{n-2}\cdots a_{n/2}$ represents the row index and $a_{n/2-1}\cdots a_1a_0$ represents the column index. The row index and the column index are separated by a comma. A 2-D FPS (denoted as $f_{\text{2-D FPS}}$) can be expressed as

$$f_{\text{2-D FPS}}:\quad (a_{n-1}a_{n-2}\cdots a_{n/2}, a_{n/2-1}\cdots a_1a_0) = (a_{n-2}\cdots a_{n/2-1}, a_{n/2-2}\cdots a_1a_0a_{n-1}). \quad (5)$$

A 2-D SPS (denoted as $f_{\text{2-D SPS}}$) can be expressed as

$$f_{\text{2-D SPS}}:\quad (a_{n-1}a_{n-2}\cdots a_{n/2}, a_{n/2-1}\cdots a_1a_0) = (a_{n-2}\cdots a_{n/2}a_{n-1}, a_{n/2-2}\cdots a_1a_0a_{n/2-1}). \quad (6)$$

As can be seen in Eqs. (5) and (6), a 2-D FPS is obtained by rotation of the binary address as a whole to the left by one bit position, and a 2-D SPS is obtained by rotation to the left by one bit position the row address and the column address separately.

1. *Relationship between 2-D Folded Perfect Shuffle and 2-D Separable Perfect Shuffle*

From Eqs. (5) and (6), it can be seen that the 2-D FPS is equivalent to (i) the exchange of the most-significant bits (MSB's) of the row address and the column address and then (ii) performance of a 2-D SPS, as follows[36]:

(i) Exchange MSB's in
$$(a_{n-1}a_{n-2}\cdots \mathrm{a}_{n/2}, a_{n/2-1}\cdots \mathrm{a}_1a_0) = (a_{n/2-1}a_{n-2}\cdots a_{n/2}, a_{n-1}\cdots a_1a_0). \tag{7}$$

(ii)
$$f_{\text{2-D SPS}}: \quad (a_{n/2-1}a_{n-2}\cdots a_{n/2}, a_{n-1}\cdots a_1a_0) = (a_{n-2}\cdots a_{n/2-1}, a_{n/2-2}\cdots a_1a_0a_{n-1}). \tag{8}$$

If we divide the addresses of nodes placed in the source array into four quadrants, Q0, Q1, Q2, and Q3, the exchange of MSB's is equivalent to the exchange of Q1 and Q3, as depicted in Fig. 2(a).

2. *Relationship between 2-D Folded Perfect Shuffle–Exchange and 2-D Separable Perfect Shuffle*

Now we derive the 2-D folded perfect shuffle–exchange (FPS-E) (denoted as $f_{\text{2-D FPS-E}}$) from the 2-D SPS. We define a 2-D FPS-E as

$$f_{\text{2-D FPS-E}}: \quad (a_{n-1}a_{n-2}\cdots a_{n/2}, a_{n/2-1}\cdots a_1a_0) = (a_{n-2}\cdots a_{n/2-1}, a_{n/2-2}\cdots a_1a_0\bar{a}_{n-1}), \tag{9}$$

which is equivalent to (1) complementing the MSB of the row address (a_{n-1}), (2) exchanging MSB's of the row address and the column address, and (3) performing a 2-D SPS on the resulting address. Because the complement of the MSB in the row address corresponds to the exchange of quadrants Q0 and Q3 and the exchange of Q1 and Q2, steps (1) and (2) result in the clockwise rotation of quadrants by one position, as explained in Fig. 2(b).

3. *Relationship between 2-D Folded Inverse Perfect Shuffle and 2-D Separable Inverse Perfect Shuffle*

We derive the mathematical relationship between the 2-D FIPS and the 2-D SIPS. We denote a 2-D FIPS as $f_{\text{2-D FIPS}}$ and defined it as

$$f_{\text{2-D FIPS}}: \quad (a_{n-1}a_{n-2}\cdots a_{n/2}, a_{n/2-1}\cdots a_1a_0) = (a_0a_{n-1}\cdots a_{n/2+1}, a_{n/2}\cdots a_1). \tag{10}$$

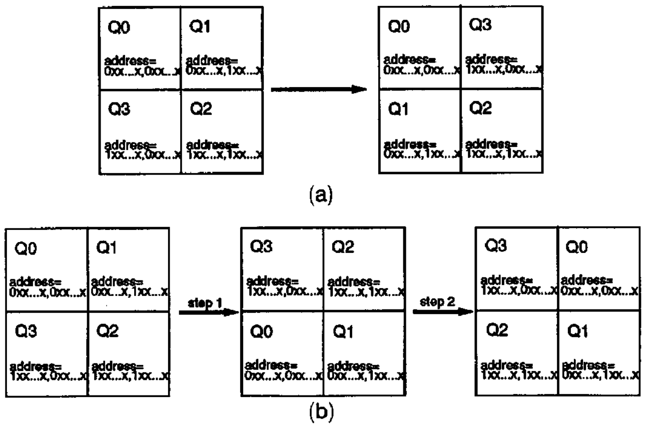

Fig. 2. In an n-bit (n even) address of a node, the most significant $n/2$ bits represent the row index, and the rest represent the column index. An x in the address represents a don't-care bit. (a) The exchange of MSB's in the row index and the column index is equivalent to the exchange of quadrants Q1 and Q1. (b) The complement of the MSB in the row index (a_{n-1}) (step 1), followed by the exchange of MSB's in the row index and the column index (step 2), results in clockwise rotation of quadrants by one position.

Similarly, we denote a 2-D SIPS as $f_{\text{2-D SIPS}}$ and define it as

$$f_{\text{2-D SIPS}}: \quad (a_{n-1}a_{n-2}\cdots a_{n/2}, a_{n/2-1}\cdots a_1a_0) = (a_{n/2}a_{n-1}\cdots a_{n/2+1}, a_0, a_{n/2-1}\cdots a_1). \tag{11}$$

Equations (10) and (11) show that the 2-D FIPS is equivalent to (1) performing a 2-D SIPS and (2) exchanging the MSB's in the row address and the column address of the resulting node address. The latter is equivalent to the exchange of quadrants Q1 and Q3.

4. *Relationship between 2-D Folded Inverse Perfect Shuffle–Exchange and 2-D Separable Inverse Perfect Shuffle*

Finally, we derive the relationship between the 2-D folded inverse perfect shuffle–exchange (FIPS-E) and the 2-D SIPS. A 2-D FIPS-E (denoted as $f_{\text{2-D FIPS-E}}$) is defined as

$$f_{\text{2-D FIPS-E}}: \quad (a_{n-1}a_{n-2}\cdots a_{n/2}, a_{n/2-1}\cdots a_1a_0) = (\bar{a}_0a_{n-1}\cdots a_{n/2+1}, a_{n/2}\cdots a_1), \tag{12}$$

which is equivalent to (1) performing the 2-D SIPS, (2) exchanging MSB's in the row index and the column index, and (3) complementing the MSB in the row index of the resulting source assay. As shown in Fig. 2(b), steps (2) and (3) correspond to the clockwise rotation of the quadrants by one position.

Figure 3 is a decomposition tree of the BdB network that summarizes the relationships derived so far. The optical BdB network based on the 3-D optical-interconnect model consists of four operations, 2-D FPS, 2-D FPS-E, 2-D FIPS, and 2-D FIPS-E operations. The 2-D FPS operation illustrated as the leftmost branch of the decomposition tree is further divided into three operations in sequence: quadrant exchange (QE), followed by a columnwise 1-D PS, followed by rowwise 1-D PS

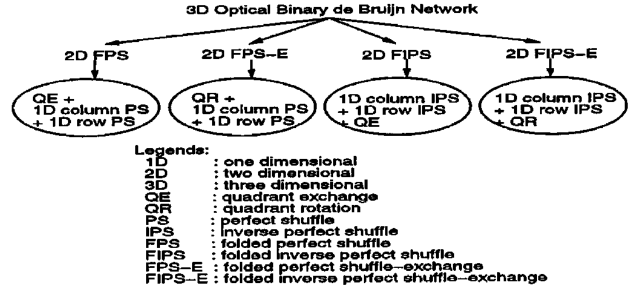

Fig. 3. Decomposition of the 3-D optical BdB network. The most fundamental three operations are identified to be 1-D PS (or IPS), QE, and QR operations because the IPS operation can be obtained from the PS operation by swapping inputs and outputs or vice versa.

operations. Similarly, 2-D FPS-E, 2-D FIPS, and 2-D FIPS-E operations are further divided into three operations, as shown in Fig. 3. The IPS operation can be obtained from the PS operation by swapping inputs and outputs or vice versa. Thus we can conclude that the most fundamental three operations are QE, quadrant rotation (QR), and PS (or IPS) for constructing the BdB network.

B. Construction of the Binary de Bruijn Network by Use of the Primitive Operations

The construction of the BdB network by use of fundamental operations is the reverse process of the decomposition, as shown in Fig. 4. At stage 1, four images (fan-outs) of the $N \times N$ input array are generated. Four images undergo FPS, FPS-E, FIPS, and FIPS-E operations, as indicated by branches 1), 2), 3), and 4), respectively. For example, QE, columnwise 1-D PS, and rowwise 1-D PS operations are performed in sequence to accomplish FPS operation. Stage 5 combines four images to give the BdB connection pattern between the input array and the output array.

4. Feasibility Study for Optical Implementations

In this section we apply the presented design methodology to the implementation of the optical BdB network, and then we analyze the proposed implementation to show the feasibility of the design methodology. An optical implementation of each fundamental operation is first presented, and then the integration of these fundamental operations is shown to construct the optical BdB network. For cascadability we restrict beam angles entering and leaving each module (an implementation of an operation) to be normal to the surface.

A. Implementation of Fundamental Optical Operations

1. *Implementation of Quadrant-Exchange Operation*

Figure 5 shows the geometry for the implementation of the QE operation. Deflecting optical components, e.g., obtained by diffractive gratings or volume holo grams, are fabricated both on the top and on the bottom of the substrate. Beams incident upon quadrants Q0 and Q2 pass through directly, whereas beams on quadrant Q1 get deflected toward Q3, and beams on Q3 get deflected toward Q1. Thus the net effect of the QE operation becomes the swapping of quadrants Q1 and Q3. The deflection-angle requirement can be calculated by use of Fig. 5(b). Because Q1 and Q3 are swapped, the beam deflection occurs along line *XX–YY*, and the angle (θ) is equal to $\tan^{-1}[(L/\sqrt{2})/t]$, where t is thickness of the substrate and L is the size of the input array in a single dimension. Suppose that we use diffractive gratings for beam deflection. From the grating equation we can derive the grating period (p) required for the QE operation on the given light wavelength (λ) as follows:

$$p = \frac{\lambda}{\sin\left(\tan^{-1}\dfrac{L/\sqrt{2}}{t}\right)}. \tag{13}$$

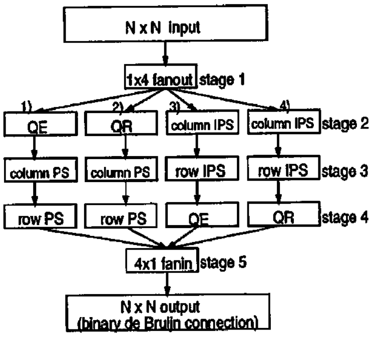

Fig. 4. Construction of the BdB network by use of fundamental operations. Fanouts indicated by 1), 2), 3), and 4) correspond to FPS, FPS-E, FIPS, and FIPS-E operations, respectively. These four operations are combined together to realize a BdB network.

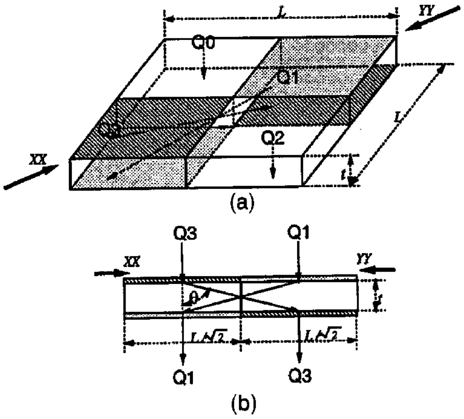

Fig. 5. Optical implementation of the QE operation: (a) a 3-D view and (b) a cross-sectional view along the *XX–YY*.

We can also use two copies of an identical volume hologram in implementing the QE operation because holograms on the Q1 facet and on the Q3 facet can have the same structure but with different orientations.

2. *Implementation of Quadrant-Rotation Operation*

Figure 6 illustrates an implementation of the QR operation. We construct four volume holograms or gratings on the facets of Q0, Q1, Q2, and Q3 for the required beam deflections. As shown in Fig. 6(b), the deflection angle (θ) is equal to $\tan^{-1}(L/2t)$, where L is the 1-D size of the input array and t is the thickness of the substrate. All four holograms will have identical structure but with different orientations. The hologram on Q0 deflects incident beams along the $+x$ direction, Q1 along the $-y$ direction, Q2 along the $-x$ direction, and Q3 along the $+y$ direction.

3. *Implementation of Perfect-Shuffle Operation*

Several implementations of permutation interconnects, including the PS operation, have been demonstrated by use of holographic optical elements,[37,38] diffractive lenslets,[39] and refractive lenslets.[40] We

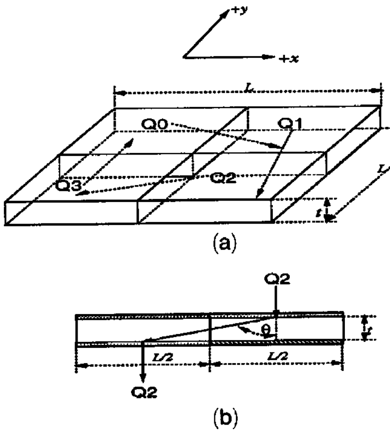

Fig. 6. Optical implementation of the QR operation: (a) a 3-D view and (b) a side view.

can easily extend such methods to implement 1-D rowwise (or columwise) PS operations.

Figure 7 shows a rowwise (or columnwise) 1-D PS implementation on eight rows (or columns). Let d be the node size along a single dimension, t be the thickness of the substrate, and θ_i be the deflected angle at node i. For a k-node 1-D PS (k is a power of two), $\theta_i = \tan^{-1}(id/t)$ if $i \leq k/2$ or $\theta_i = \tan^{-1}\{[(i - k + 1)d]/t\}$ if $i > k/2$. As discussed above, these angular requirements determine the period of the grating when we use diffractive gratings for the beam deflection. We should note that the 1-D PS operation on k nodes requires $k/2$ distinct deflecting components because deflection angles of the first $k/2$ nodes are symmetric to those of the remaining $k/2$ nodes. Also, an implementation of the 1-D IPS operation can be easily achieved by swapping inputs and outputs of the 1-D PS implementation in Fig. 7.

B. Integration of Fundamental Operations to Construct the Optical Binary de Bruijn Network

As shown in Fig. 4, we need a 1 × 4 fan-out element (a 4 × 1 fan-in element as well) to construct the BdB network in addition to the implementations of fundamental operations discussed so far.

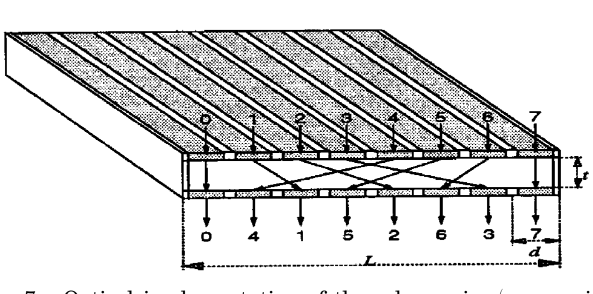

Fig. 7. Optical implementation of the columnwise (or rowwise) 1-D PS operation on eight columns (or eight rows). If we swap inputs and outputs, it can perform the columnwise (or rowwise) 1-D inverse PS operation.

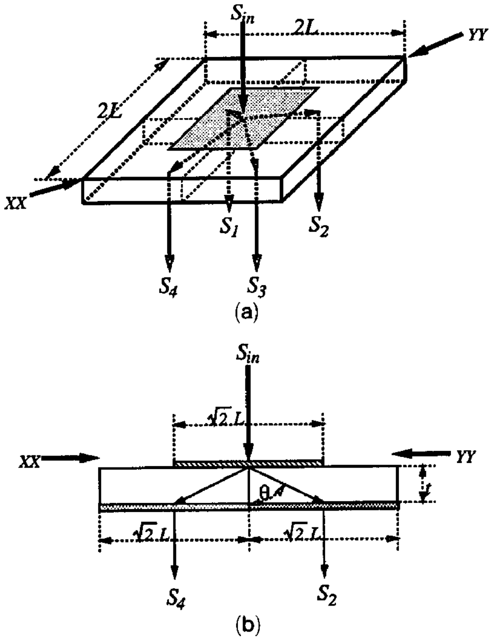

Fig. 8. Optical implementation of the 1 × 4 fan-out element (or the 4 × 1 fan-in element if we swap the input and the output): (a) a 3-D view and (b) a cross-sectional view along line *XX*–*YY*.

1. *Implementation of 1 × 4 Fan-out/Fan-In Elements*

Several implementations of fan-out elements have been demonstrated.[41–44] We discuss geometry requirements of fan-out elements to implement the BdB network. Figure 8 illustrates an implementation of the fan-out element that uses a multiplexed volume hologram. Using the geometry given in Fig. 8(b), we can see that the deflection angle of each beam is equal to $\tan^{-1}[L/(\sqrt{2}t)]$. A 4 × 1 fan-in element can be achieved by swapping inputs and outputs of the 4 × 1 fan-out element.

2. *Construction of the Binary de Bruijn Network*

The construction of the BdB network by use of implementations of fundamental operations is the process of integration, as shown in Fig. 4. Figure 9 shows a 3-D view of the constructed BdB network obtained by implementations of the fundamental operations presented in Subsection 4A. Beams gen-

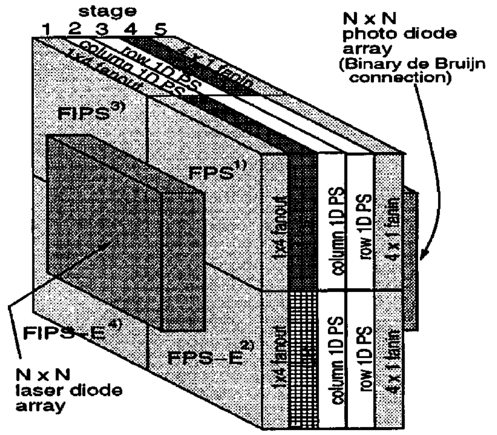

Fig. 9. Integration of fundamental optical operations to correct the BdB network. Fanouts indicated by 1), 2), 3), and 4) correspond to FPS, FPS-E, FIPS, and FIPS-E operations, respectively.

erated from an $N \times N$ laser diode array are first split into four images by the 1×4 fan-out element. Four images of the input array will undergo FPS, FPS-E, FIPS, and FIPS-E operations, respectively. Each operation is achieved by performance of a sequence of three fundamental operations. At the last stage a fan-in element combines four images, which result in the BdB connections between the input and the output arrays.

An index-matching fluid might be used between stages to ensure that the traveling light is not disturbed optically. Also, we can use the space-division multiplexing technique with multiple detectors per node to avoid the fan-in problem[45] at the last stage of the BdB network integration. An incoming signal distinction scheme, which encodes spatial positions of the sources at the nodes, would allow us to use an affordable number of detectors per node.[46]

C. Analysis of the Proposed Implementation

1. *Integration Density*

The optical implementation of the network cannot be made arbitrarily small because of the limitation in the achievable deflection angle, resulting from the limited resolution of the diffractive grating. In addition, the integration density is limited owing to diffraction spreading. Both constraints must be fulfilled at the same time.

We use Gaussian beam optics[47] to analyze the integration density limited by diffraction spreading. In Gaussian beam optics the dependency of the beam radius on the beam traveling distance is governed by

$$w(z) = w_0\left[1 + \left(\frac{\lambda z}{\pi {w_0}^2}\right)^2\right]^{1/2}, \qquad (14)$$

where z is the distance propagated, λ is the wavelength of the light, w_0 is the original beam radius, and $w(z)$ is the beam radius after the beam propagates a distance z. For the BdB network implementation depicted in Fig. 9, we calculate the longest distance that beams may propagate. Assume that L is the size of the input array along a single dimension and that t is the thickness of each substrate. The longest beam traveling distance in the entire BdB network is equivalent to the accumulation of the longest path in each module. From Fig. 8(b) the fan-out and fan-in modules make the beam travel a distance $2[t^2 + (L^2/2)]^{1/2}$. From Figs. 5, 6, and 7 we can see that the QE, QR, and PS operations require the longest distance, $[t^2 + (L^2/2)]^{1/2}$, $[t^2 + (L^2/4)]^{1/2}$, and $[t^2 + (L^2/4)]^{1/2}$, respectively. Thus the longest distance ($z_{\max}$) in the BdB network implementation that a beam may propagate is equal to

$$z_{\max} = 3\left(t^2 + \frac{L^2}{2}\right)^{1/2} + 2\left(t^2 + \frac{L^2}{4}\right)^{1/2}. \qquad (15)$$

Therefore, from Eq. (14) the source beam radius w_0 will expand to $w(z_{\max})$ on the detector side.

We calculate the achievable integration density on the input array size ($L \times L$) to be 10 mm × 10 mm. Suppose that the thickness (t) is 7 mm, the source beam size is 100 µm in diameter, and the source light wavelength is 785 nm. The maximum deflection angle required in realizing the optical BdB network occurs in the implementation of the QE operation, as discussed in Subsection 4A. From Eq. (13) the highest grating resolution for the given input array size is 276 nm if a four-level grating is used. Such resolution can be achieved with current electron-beam writing technology.[48] Now we consider the effect of diffractive spreading on the integration density. The longest distance ($z_{\max}$) becomes approximately 48 mm, resulting in the beam size of 157 µm in diameter on the detector array. With this beam size we can integrate approximately 4096 sources (detectors) in a 10 mm × 10 mm input (detector) array size. Thus a 12-BdB network (4096 nodes) can be easily integrated. The total area required for the 12-BdB network is 20 mm × 20 mm because of to the fan-out module.

2. *Implementation Complexity*

As discussed in Subsection 4.A, an implementation of the N-node BdB network requires implementations of QE, QR, and PS operations, in addition to the 1×4 fan-out/fan-in elements. The number of distinct optical deflecting components in implementing the QE, QR, and fan-out/fan-in elements is independent of the network size N. However, the PS operation requires $\sqrt{N/2}$ distinct deflecting components.

3. *Signal Skew*

The longest path in the proposed BdB implementation is derived in Eq. (15). The shortest path occurs if beams propagate without any deflection in the QE and the PS operations. Thus the shortest path ($z$min) is equal to

$$z_{\min} = 2\left(t^2 + \frac{L^2}{2}\right)^{1/2} + 2t + \left(t^2 + \frac{L^2}{4}\right)^{1/2}. \qquad (16)$$

With the same sizes of L and t used in the previous calculation the shortest distance becomes approximately 43 mm. Thus the inherent signal skew owing to the path difference in the proposed implementation is no greater than 167 ps.

4. *Power Efficiency*

In this analysis we assume the use of index-matching fluid between stages and the use of antireflection coating on each surface for enhancing power efficiency. The fan-out/fan-in module that uses a multiplexed volume hologram can be made with a transmission power efficiency as high as 90%.[43] The QE, QR, and PS operations require only beam-deflecting components. With volume holograms a deflection efficiency of 95% can easily be achieved.[44] The transmission power efficiency of each operation is 90% because a pair of holograms is used along the beam path. Thus, overall power efficiency of the proposed implementation can be as high as 48%.

5. Cascadability

Suppose that we use volume holograms in implementing each operation. The proposed optical implementation uses a pair of identical volume holograms along the beam path for each operation. This configuration tolerates the dependency of the beam angles that leave the module on the instability of the light wavelength. In other words, the beam will leave the module normal to the surface regardless of the light wavelength as long as the beams are incident normal to the surface. Thus cascadability can be well maintained by use of a pair of identical holograms.

5. Conclusions

The binary de Bruijn topology as an interconnection network for parallel computers has been studied recently as an alternative to the hypercube-based or the mesh-based interconnection network. The binary de Bruijn network has the node degree of a two-dimensional mesh and the diameter of a hypercube. The de Bruijn network retains most of the desired properties of the hypercube network such as a small diameter, on easy message-routing scheme, fault tolerance, and efficient mapping of many scientific and engineering problems. In addition, the de Bruijn network has a constant node degree independent of the network size, which is very desirable in constructing large-scale systems. Unfortunately, the de Bruijn network is not fully symmetric, and the connection patterns are not localized. This makes its VLSI implementation nontrivial, though not impossible. However, free-space optics, owing to its flexibility and three-dimensional connectivity, can easily realize the nonsymmetric global connections.

In this paper we have proposed a design methodology for the optical implementation of the de Bruijn network. The methodology first decomposes the binary de Bruijn network into a few basic operations that can be efficiently implemented. Then it integrates these basic operations to construct the network. To show the feasibility of the design methodology, we proposed an optical implementation of the binary de Bruijn network. The developed design methodology is good for bulk optics, holographic optics, or planar optics because the methodology does not assume any specific optical technologies. A 4096-node binary de Bruijn network was analyzed and found to be feasible for optical implementations. Such implementation demonstrates good feasibility by showing a reasonable optical power efficiency and a volume size capable of fitting within the case of a massively parallel computer.

This research was supported by National Science Foundation grant MIP 9310082.

References

1. H. J. Siegel, *Interconnection Networks for Large-Scale Parallel Processing* (McGraw-Hill, New York, 1990).
2. J.-C. Bermond and C. Peyrat, "De Bruijn and Kautz networks: a competitor for the hypercube?" *in Proceedings of the First European Workshop on Hypercube and Distributed Computers* (Elsevier, New York, 1989), pp. 279–293.
3. K. Hwang, *Advanced Computer Architecture: Parallelism, Scalability, Programmability* (McGraw-Hill, New York, 1993).
4. T.-Y. Feng, "A survey of interconnection networks," IEEE Computer (December 1981), pp. 12–27.
5. A. Louri and H. Sung, "An optical multi-mesh hypercube: a scalable optical interconnection network for massively parallel computing," J. Lightwave Technol. **12,** 704–716 (1994).
6. J. P. Hayes and T. Mudge, "Hypercube supercomputers," Proc. IEEE **77,** 1829–1841 (1989).
7. W. D. Hillis, *The Connection Machine* (MIT Press, Cambridge, Mass., 1985).
8. R. Berrendorf and J. Helin, "Evaluating the basic performance of the Intel iPSC/860 parallel computer," Concurrency: Pract. Exper. **4**(5), 223–240 (1992).
9. G. Ramanathan and J. Oren, "Survey of commercial parallel machines," Comput. Archit. News **21**(6), 13–33 (1993).
10. Supercomputer Systems Division, *Paragon XP/S Product Overview* (Intel Corporation, Beaverton, Oregon, 1991).
11. "Cray/MPP Announcement" (Cray Research, Inc., Eagan, Minnesota, 1992).
12. C. L. Seitz, "Mosaic C: an experimental fine-grain multicomputer," Tech. Rep. 55 (California Institute of Technology, Pasadena, California, 1992).
13. D. Lenoski, J. Laudon, K. Gharachorloo, W. D. Weber, A. Gupta, J. Hennessy, M. Horowitz, and M. Lam, "The Stanford Dash multiprocessor," IEEE Computer (March 1992), pp. 63–79.
14. N. G. de Bruijn, "A combinatorial problem," K. Ned. Akad. Wet. Proc. Ser. A **49,** 758–764 (1946).
15. H. Fredricksen, "A new look at the de Bruijn graph," Discrete Appl. Math. **37,** 193–203 (1992).
16. M. R. Samatham and D. K. Pradhan, "The de Bruijn multiprocessor network: a versatile parallel processing and sorting network for VLSI," IEEE Trans. Comput. **38,** 567–581 (1989); also, see corrections, **40,** 122 (1991).
17. O. Collins, S. Dolinar, R. McEliece, and F. Pollara, "A VLSI decomposition of the de Bruijn graph," J. ACM **39,** 931–948 (1992).
18. M. A. Sridhar and C. S. Raghavendra, "Fault-tolerant networks based on the de Bruijn graph," IEEE Trans. Comput. **40,** 1167–1174 (1991).
19. S. Okugawa, "Characteristics of the de Bruijn (DB) network for massively parallel computers," Trans. Inst. Electron. Inf. Commun. Eng. **J75D-I,** 592–599 (1992).
20. C. G. Herter, W. F. Warschko, and M. Philippsen, "Triton/1: a massively parallel mixed-mode computer designed to support high level languages," in *Proceedings of the IEEE International Parallel Processing Symposium* (Institute of Electrical and Electronics Engineers, New York, 1993).
21. A. Louri, "3-D Optical architecture and data-parallel algorithms for massively parallel computing," IEEE Micro **11**(4), 24–68 (1991).
22. G. E. Lohman and K. H. Brenner, "Space-invariance in optical computing systems," Optik **89,** 123–134 (1992).
23. A. Guha, J. Bristow, C. Sullivan, and A. Husain, "Optical interconnections for massively parallel architectures," Appl. Opt. **29,** 1077–1093 (1990).
24. 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. **9,** 1665–1674 (1991).
25. H. S. Hinton, J. R. Erickson, T. J. Cloonan, and G. W. Richards, "Space-division switching," in *Photonics in Switching: Systems,* J. E. Midwinter, ed. (Academic, New York, 1993), Vol. 2, chap. 3.
26. F. B. McCormick, "Free-space interconnection techniques," in

Photonics in Switching: Systems, J. E. Midwinter, ed. (Academic, New York, 1993), Vol. 2, chap. 4.

27. A. Louri and H. Sung, "3-D optical interconnects for high-speed interchip and interboard communications," Computer **27**(10), 27–37 (1994).
28. A. W. Lohman, W. Stork, and G. Stucke, "Optical perfect shuffle," Appl. Opt. **25,** 1530–1531 (1986).
29. K. H. Brenner and A. Huang, "Optical implementation of the perfect shuffle interconnection," Appl. Opt. **27,** 135–137 (1988).
30. C. W. Stirk, R. A. Athale, and M. W. Haney, "Folded perfect shuffle optical processor," Appl. Opt. **27,** 202–203 (1988).
31. M. W. Haney and J. J. Levy, "Optically efficient free-space folded perfect shuffle network," Appl. Opt. **30,** 2833–2840 (1991).
32. J. M. Wang, L. Cheng, and A. A. Sawchuk, "Optical two-dimensional perfect shuffles based on a one-copy algorithm," Appl. Opt. **31,** 5464–5467 (1992).
33. Y. Zhan, "Optical implementation of the folded perfect shuffle interconnection network using quadrant-encoded gratings," Opt. Eng. **32,** 1657–1661 (1993).
34. S. Bian, K. Xu, and J. Hong, "Optical perfect shuffle using Wollaston prisms," Appl. Opt. **30,** 173–174 (1991).
35. S. Lai and D. Hsu, "Perfect-shuffle implementation by using reflecting prisms," Appl. Opt. **31,** 6191–6192 (1992).
36. L. Cheng and A. A. Sawchuk, "Three-dimensional omega networks for optical implementation," Appl. Opt. **31,** 5468–5479 (1992).
37. H. Kobolla, F. Sauer, and R. Volkel, "Holographic tandem arrays," in *Holographic Optics II: Principles and Applications,* G. M. Morris, Proc. Soc. Photo-Opt. Instrum. Eng. **1136,** 146–149 (1989).
38. B. Robertson, E. J. Restall, M. R. Taghizadeh, and A. C. Walker, "Space-variant holographic optical elements in dichromated gelatin," Appl. Opt. **30,** 2368–2375 (1991).
39. J. Jahns and W. Daschner, "Optical cyclic shifter using diffractive lenslet arrays," Opt. Commun. **79,** 407–410 (1990).
40. F. Sauer, J. Jahns, and C. R. Nijander, "Refractive–diffractive micro-optics for permutation interconnects," Opt. Eng. **33,** 1550–1560 (1990).
41. M. Kato, Y.-T. Huang, and R. K. Kostuk, "Multiplexed substrate-mode holograms," J. Opt. Soc. Am. A **7,** 1441–1447 (1990).
42. S. K. Case, "Coupled-wave theory for multiple exposed thick holographic gratings," J. Opt. Soc. Am. **65,** 724–729 (1975).
43. M. R. Wang, G. J. Sonek, R. T. Chen, and T. Jannson, "Large fan-out optical interconnects using thick holographic gratings and substrate wave propagation," Appl. Opt. **31,** 236–249 (1992).
44. D. Prongue, "Diffractive optical elements for interconnections," Ph.D. dissertation (Universite de Neuchâtel, Neuchâtel, Switzerland, 1992).
45. J. W. Goodman, "Fan-in and fan-out with optical interconnections," Opt. Acta **32,** 1489–1496 (1985).
46. A. Louri, H. Sung, Y. Moon, and B. P. Zeigler, "An efficient signal distinction scheme for large-scale free-space optical networks using genetic algorithms," presented at the OSA Photonics in Switching, Salt Lake City, Utah, Topical Meeting on 12–17 March 1995.
47. B. E. A. Saleh and M. C. Teich, *Fundamentals of Photonics* (Wiley, New York, 1991).
48. H. P. Herzig, M. T. Gale, H. W. Lehmann, and R. Morf, "Diffractive components: computer generated elements," in *Perspectives for Parallel Optical Interconnects,* P. Lalanne and P. Chavel, eds. (Springer-Verlag, New York, 1991), Chap. 5.