An Efficient 3D Optical Implementation of Binary de Bruijn Networks with Applications to Massively Parallel Computing [1]

Ahmed Louri and Hongki Sung

Department of Electrical and Computer Engineering
The University of Arizona
Tucson, AZ 85721

Abstract – *As an alternative to the hypercube, the binary de Bruijn (BdB) network is recently receiving 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, due to its asymmetrical nature and global connectivity it is posing 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. In this paper, 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 efficiently implemented. We further show feasibility of the presented design methodology by proposing an optical implementation of the BdB network.*

1 Introduction

The recent quest for massively parallel computing systems is placing a major emphasis on scalable networks with small diameters and bounded node degrees[1]. As an alternative to the hypercube and the mesh topologies, the de Bruijn topology[2, 3] has recently been receiving much attention. Its properties and applications have been studied by several researchers[4, 5, 6, 7, 8]. Its topological properties show that the de Bruijn network is a good candidate for interconnection networks of the next generation of parallel computers after the hypercube. The de Bruijn network behaves like the hypercube, and retains most of the same 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, to construct a network of 1,024 nodes, the hypercube network requires 5,120 physical links whereas the de Bruijn network requires only 2,048 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 network, the node degree is always four regardless of the network size. It should be noted that the node degree of the mesh network is also four independent of the network size, but the binary de Bruijn network has 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[5]. This coupled with a constant node degree, makes the de Bruijn network a highly desirable architecture for future large-scale systems.

Despite its many attractive properties, the de Bruijn network is considerably less known compared to the hypercube network because it is much less amenable to VLSI implementations. The VLSI implementation of the de Bruijn network is nontrivial since the network is neither fully symmetric nor modular[5, 6] 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 couple of parallel machines: The Triton/1 computer developed at the University of Karlsruhe[9], and the de Bruijn VLSI network with 8192 nodes being built by NASA's Galileo project[6].

Optics, owing to its three-dimensional (3-D) nature, global connectivity property, and its flexible signal routing capability, seems to be very suitable for realizing non-symmetric global connections[10, 11]. 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 oper-

[1]This research was supported by an NSF grant No. MIP 9310082.

0-8186-7101-7/95 $4.00 © 1995 IEEE

ations 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 $4cm^2$ area with the total power efficiency being as high as 48%. We will provide the detailed analysis that led to these figures at the conference.

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 ($0 \leq i < 2^n$) 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.

$$i_1 = a_{n-2}a_{n-3}\cdots a_1 a_0 a_{n-1} \quad (1)$$
$$i_2 = a_{n-2}a_{n-3}\cdots a_1 a_0 \overline{a}_{n-1} \quad (2)$$
$$i_3 = a_0 a_{n-1} a_{n-2}\cdots a_2 a_1 \quad (3)$$
$$i_4 = \overline{a}_0 a_{n-1} a_{n-2}\cdots a_2 a_1 \quad (4)$$

Node i_1 connection from node i in Eq. 1 is obtained by rotating node i address to the left by one bit position, which is equivalent to the perfect shuffle (PS) operation. Node i_2 connection from node i in Eq. 2 is obtained by rotating node i address to the left and then complementing the least significant bit which is equivalent to a perfect shuffle-exchange (PS-E) operation. Similarly, 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 node i_4 connection from node i is obtained by a right rotation and complement operation or an inverse perfect shuffle-exchange (IPS-E) 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 network by simply connecting two three-BdB networks as is the case with the hypercube network), 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 using the BdB network compared to 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). A detailed comparison of the n-BdB network with the two most popular networks, the binary n-cube and the mesh networks, is summarized in Table 1.

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 [12]. The optical interconnect module receives an image from the source array and generates the required optical links to the detector array.

3.1 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 PS operation, a PS-E operation, an IPS operation, and an IPS-E operation. Since 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 kinds of 2-D perfect shuffles[13, 14, 15]: 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 columns of the input are obtained by folding a 1-D input array. The mathematical relationship between the 2-D SPS and the 2-D FPS has been shown in Ref. [16]. In this subsection, we first summarize the mathematical relationship between the 2-D SPS and the 2-D FPS derived in Ref. [16], and then 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 2-D plane). A binary address of a node can be represented by $(a_{n-1}a_{n-2}\cdots a_{n/2},\ a_{n/2-1}\cdots a_1 a_0)$, where $a_{n-1}a_{n-2}\cdots a_{n/2}$ represents the row index and $a_{n/2-1}\cdots a_1 a_0$ represents the column index. Note that 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:

$$f_{\text{2-D FPS}}: (a_{n-1}a_{n-2}\cdots a_{n/2},\ a_{n/2-1}\cdots a_1 a_0) = (a_{n-2}\cdots a_{n/2-1},\ a_{n/2-2}\cdots a_1 a_0 a_{n-1}) \quad (5)$$

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

$$f_{\text{2-D SPS}}: (a_{n-1}a_{n-2}\cdots a_{n/2},\ a_{n/2-1}\cdots a_1 a_0) =$$

$$(a_{n-2}\cdots a_{n/2}a_{n-1},\ a_{n/2-2}\cdots a_1a_0a_{n/2-1}) \quad (6)$$

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

Relationship Between 2-D Folded Perfect Shuffle(FPS) and 2-D Separable Perfect Shuffle(SPS)

From Eqs. 5 and 6, it can be seen that the 2-D FPS is equivalent to (i) exchanging most significant bits (MSBs) of the row address and the column address, and then (ii) performing a 2-D SPS as follows[16].

(i) Exchange MSBs in

$$(a_{n-1}a_{n-2}\cdots a_{n/2}, a_{n/2-1}\cdots a_1a_0) =$$
$$(a_{n/2-1}a_{n-2}\cdots a_{n/2}, a_{n-1}\cdots a_1a_0) \quad (7)$$

(ii) $f_{2\text{-D SPS}}$:

$$(a_{n/2-1}a_{n-2}\cdots a_{n/2}, a_{n-1}\cdots a_1a_0) =$$
$$(a_{n-2}\cdots a_{n/2-1}, a_{n/2-2}\cdots a_1a_0a_{n-1}) \quad (8)$$

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

Relationship Between 2-D Folded Perfect Shuffle -Exchange (FPS-E) and 2-D Separable Perfect Shuffle (SPS)

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

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

which is equivalent to (1) complementing the MSB of the row address (a_{n-1}), (2) exchanging MSBs of the row address and the column address, and (3) performing a 2-D SPS on the resulting address. Since 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 clockwise rotation of quadrants by one position as explained in Fig. 2.b.

Relationship Between 2-D Folded Inverse Perfect Shuffle (FIPS) and 2-D Separable Inverse Perfect Shuffle (SIPS)

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

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

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

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

Eqs. 10 and 11 show that the 2-D FIPS is equivalent to (1) performing 2-D SIPS, and (2) exchanging most significant bits (MSBs) 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.

Relationship Between 2-D Folded Inverse Perfect Shuffle-Exchange (FIPS-E) and 2-D Separable Inverse Perfect Shuffle (SIPS)

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

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

which is equivalent to (1) performing the 2-D SIPS, (2) exchanging most significant bits (MSBs) in the row index and the column index, and (3) complementing the MSB in the row index of the resulting source array. 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 which summarizes the relationships derived so far. The optical BdB network based on the 3-D optical interconnect model consists of four operations; 2D FPS, 2D FPS-E, 2D FIPS, and 2D FIPS-E operations. The 2D FPS operation illustrated as the leftmost branch of the decomposition tree is further divided into three operations in sequence; QE followed by columnwise 1D PS, and followed by rowwise 1D PS operations. Similarly, 2D FPS-E, 2D FIPS, and 2D FIPS-E operations are further divided into three operations as shown in Fig. 3. It should be noted that 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 identified to be QE, QR, and PS (or IPS) for constructing the BdB network.

3.2 Construction of the Binary de Bruijn Network Using the Primitive Operations

The construction of the BdB network using fundamental operations is the reverse process of the decomposition as shown in Fig. 4. At stage 1, four images (fanouts) of the $N \times N$ input arrary are generated. Four images undergo FPS, FPS-E, FIPS, FIPS-E operations as indicated by branches 1), 2), 3), and 4) respectively. For example, QE, columnwise 1D PS, and

rowwise 1D PS operations are performed in sequence to accomplish FPS operation. Stage 5 combines four images to give the binary de Bruijn 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 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.

4.1 Implementation of Fundamental Optical Operations

Implementation of Quadrant Exchange Operation

Figure 5 shows the geometry for the implementation of the QE operation. Deflecting optical components, e.g., using diffractive gratings or volume holograms, are fabricated both on the top and on the bottom of the substrate. Beams incident on 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 using Fig. 5.b. Since Q1 and Q3 are swapped, the beam deflection occurs along line $XX - YY$ and the angle (θ) is equal to $\tan^{-1}\frac{L/\sqrt{2}}{t}$ 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:

$$p = \frac{\lambda}{\sin(\tan^{-1}\frac{L/\sqrt{2}}{t})} \qquad (13)$$

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

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 $\tan^{-1}\frac{L}{2t}$, where L is the one dimensional size of the input array and t is the thickness of the substrate. It should be noted that all four holograms will have identical structure but with different orientations. The hologram on Q0 deflects incident beams along $+x$ direction, Q1 along $-y$ direction, Q2 along $-x$ direction, and Q3 along $+y$ direction.

Implementation of Perfect Shuffle Operation

Several implementations of permutation interconnects, including the PS operation, have been demonstrated so far using holographic optical elements [17, 18], diffractive lenslets [19], and refractive lenslets [20]. We can easily extend such methods to implement 1D rowwise (or columwise) PS operations.

Figure 7 shows a rowwise (or columnwise) 1D PS implementation on 8 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), $\theta_i = \tan^{-1}\frac{id}{t}$ if $i \leq k/2$, or $\theta_i = \tan^{-1}\frac{(i-k+1)d}{t}$ if $i > k/2$. As discussed earlier, these angular requirements determine the period of the grating when we use diffrative gratings for the beam deflection. We should note that the 1-D PS operation on k nodes requires $k/2$ distinct deflecting components since deflection angles of the first $k/2$ nodes are symmetric to those of the remaining $k/2$ nodes. Also note that 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.

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

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

Implementation of 1×4 fanout/fanin elements

Several implementations of fanout elements have been demonstrated [21, 22, 23, 24]. We discuss geometry requirements of fanout elements to implement the BdB network. Figure 8 illustrates an implementation of the fanout element using 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 $\tan^{-1}\frac{L}{\sqrt{2}\times t}$. It should be noted that a 4×1 fanin element can be achieved by swapping inputs and outputs of the 4×1 fanout element.

Construction of the binary de Bruijn network

The construction of the BdB network using 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 using implementations of fundamental operations presented in Subsec. 4.1. Beams generated from an $N \times N$ laser diode array are first split into four images by the 1×4 fanout element. Four images of the input array will undergo FPS, FPS-E, FIPS, and FIPS-E operations, respectively. Each operation is achieved by performing a sequence of three fundamental operations. At the last stage a fanin element combines four images,

which result in the BdB connections between the input and output arrays.

It should be noted that an index-matching fluid might be used between stages to ensure that the traveling light is not disturbed optically. Also note that we can use the space division multiplexing technique with multiple detectors per node to avoid fanin problem [25] 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 [26].

4.3 Analysis of the Proposed Implementation

Due to page limitations, we provide the detailed performance analysis of the proposed implementation at the conference.

5 Conclusions

The binary de Bruijn topology as an interconnection network for parallel computers has been recently studied 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 small diameter, 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.

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. And then, it integrates these basic operations to construct the network. To show feasibility of the design methodology, we proposed an optical implementation of the binary de Bruijn network. It should be noted that the developed design methodology is good for bulk optics, holographic optics, or planar optics, etc., since 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.

References

[1] G. Bell, "Ultracomputers: A Teraflop Before Its Time," *Communications of the ACM*, vol. 35, pp. 27–47, Aug 1992.

[2] N. G. de Bruijn, "A Combinatorial Problem," *Koninklije Nedderlandse Academie van Wetenshappen Proc*, vol. Ser A49, pp. 758–764, 1946.

[3] H. Fredricksen, "A New Look at the de Bruijn Graph," *Discrete Applied Mathematics*, vol. 37-38, pp. 193–203, Jul. 1992.

[4] 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*, Rennes, France, pp. 279–293, Elsevier Science Publishers, Oct. 4-6 1989.

[5] M. R. Samatham and D. K. Pradhan, "The De Bruijn Multiprocessor Network: A Versatile Parallel Processing and Sorting Network for VLSI," *IEEE Trans. on Computers*, vol. 38, pp. 567–581, Apr 1989. Also see corrections appeared in Vol. 40, pp. 122, Jan. 1991.

[6] O. Collins, S. Dolinar, R. McEliece, and F. Pollara, "A VLSI decomposition of the deBruijn graph," *Journal of the ACM*, vol. 39, pp. 931–948, Oct. 1992.

[7] M. A. Sridhar and C. S. Raghavendra, "Fault-tolerant networks based on the de Bruijn graph," *IEEE Transactions on Computers*, vol. 40, pp. 1167–1174, Oct. 1991.

[8] S. Okugawa, "Characteristics of the de Bruijn (DB) Network for Massively Parallel Computers," *Trans. of the Institute of Electronics, Information and Communication Engineers*, vol. J75D-I, pp. 592–599, Aug. 1992.

[9] 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 International Parallel Processing Symposium*, 1993.

[10] A. Louri, "3-D Optical Architecture and Data-parallel Algorithms for Massively Parallel Computing," *IEEE MICRO (Chips, Systems, Software, and Applications)*, vol. 11, pp. 24–68, April 1991.

[11] F. B. McCormick, "Free-space Interconnection Techniques," in *Photonics in Switching: Volume II, Systems* (J. E. Midwinter, ed.), ch. 4, New York, New York: Academic Press, 1993.

[12] A. Louri and H. Sung, "3D Optical Interconnects for High-Speed Interchip and Interboard Communications," *Computer*, vol. 27, pp. 27–37, Oct. 1994.

[13] A. W. Lohman, W. Stork, and G. Stucke, "Optical Perfect Shuffle," *Applied Optics*, vol. 25, pp. 1530–1531, 1986.

[14] C. W. Stirk, R. A. Athale, and M. W. Haney, "Folded Perfect Shuffle Optical Processor," *Applied Optics*, vol. 27, pp. 202–203, Jan 1988.

[15] J. M. Wang, L. Cheng, and A. A. Sawchuk, "Optical Two-dimensional Perfect Shuffles Based on a One-copy Algorithm," *Applied Optics*, vol. 31, pp. 5464–5467, Sep. 1992.

[16] L. Cheng and A. A. Sawchuk, "Three-dimensional Omega Networks for Optical Implementation," *Applied Optics*, vol. 31, pp. 5468–5479, Sep. 1992.

[17] H. Kobolla, F. Sauer, and R. Volkel, "Holographic Tandem Arrays," in *Proc. SPIE, vol. 1136*, pp. 146–149, 1989.

[18] B. Robertson, E. J. Restall, M. R. Taghizadeh, and A. C. Walker, "Space-variant Holographic Optical Elements in Dichromated Gelatin," *Applied Optics*, vol. 30, pp. 2368–2375, 1991.

[19] J. Jahns and W. Daschner, "Optical Cyclic Shifter Using Diffrative Lenslet Arrays," *Optics Communications*, vol. 79, pp. 407–410, 1990.

[20] F. Sauer, J. Jahns, and C. R. Nijander, "Refractive-diffractive Micro-optics for Permutation Interconnects," *Optical Engineering*, vol. 33, pp. 1550–1560, 1990.

[21] M. Kato, Y.-T. Huang, and R. K. Kostuk, "Multiplexed Substrate-mode Holograms," *J. Opt. Soc. Am. A*, vol. 7, pp. 1441–1447, 1990.

[22] S. K. Case, "Coupled-wave Theory for Multiple Exposed Thick Holographic Gratings," *Journal of the Optical Society of America*, vol. 65, pp. 724–729, Jun. 1975.

[23] M. R. Wang, G. J. Sonek, R. T. Chen, and T. Jannson, "Large Fanout Optical Interconnects Using Thick Holographic Gratings and Substrate Wave Propagation," *Applied Optics*, vol. 31, pp. 236–249, Jan 1992.

[24] D. Prongue, "Diffractive Optical Elements for Interconnections," Ph.D. dissertation, Universite de Neuchatel, 1992.

[25] J. W. Goodman, "Fan-in and fan-out with Optical Interconnections," *Optica Acta*, vol. 32, pp. 1489–1496, 1985.

[26] 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," in *Photonics in Switching: Topical Meeting, OSA*, pp. 90–92, Salt Lake City, Utah, March 12–17 1995.

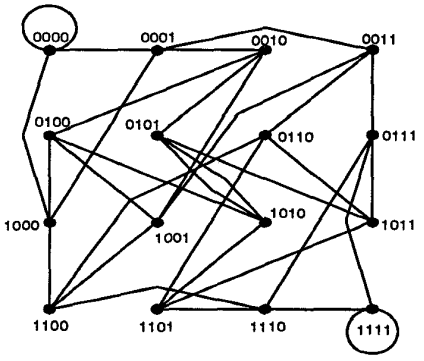

Figure 1: A four-BdB network with 16 nodes. Node addresses are represented by binary numbers.

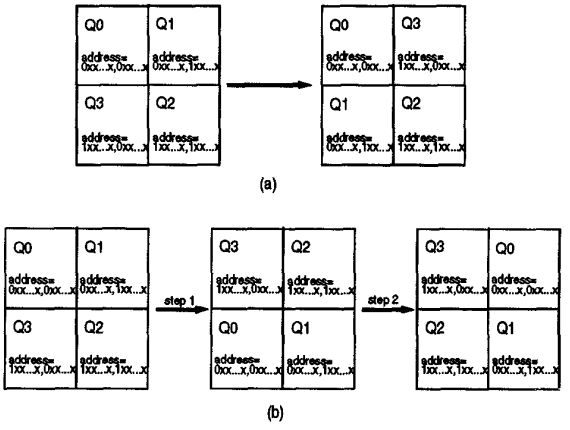

Figure 2: In an n-bit (n even) address of a node, the most significant $n/2$ bits represent the row index and the rest represent the column index. An 'x' in the address represents *don't care* bit. (a) The exchange of MSBs in the row index and the column index is equivalent to the exchange of quadrants Q1 and Q3, and (b) the complement of MSB in the row index (a_{n-1}) (step 1), followed by the exchange of MSBs in the row index and the column index (step 2) results in clockwise rotation of quadrants by one position.

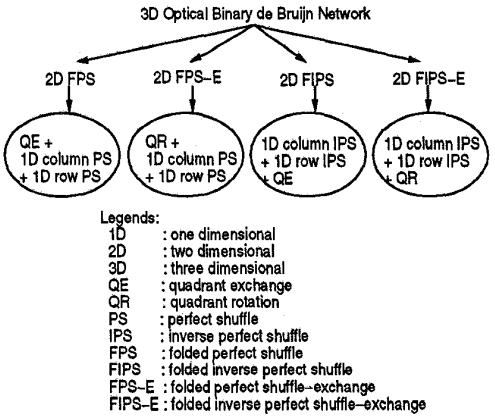

Figure 3: Decomposition of the 3-D optical binary de Bruijn network. The most fundamental three operations are identified to be 1-D perfect shuffle (or inverse perfect shuffle), quadrant exchange, and quadrant rotation operations since the inverse perfect shuffle operation can be obtained from the perfect shuffle operation by swapping inputs and outputs or vice versa.

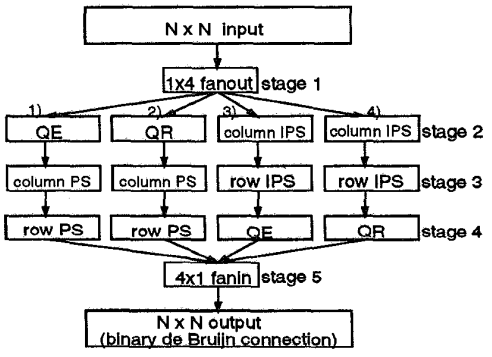

Figure 4: Construction of the binary de Bruijn network using fundamental operations. Fanouts indicated by 1), 2), 3), and 4) correspond to FPS, FPS-E, FIPS, and FIPS-E operations, respectively. These four operations are combined together to realize a binary de Bruin network .

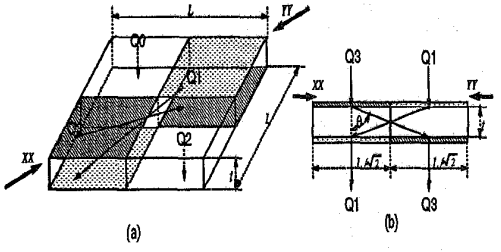

Figure 5: An optical implementation of the quadrant exchange (QE) operation. (a) a 3-D view, (b) a cross sectional view along line XX-YY.

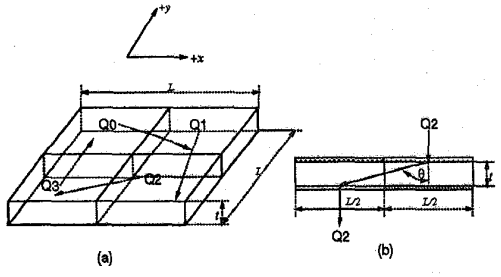

Figure 6: An optical implementation of the quadrant rotation (QR) operation. (a) a 3-D view, (b) a side view.

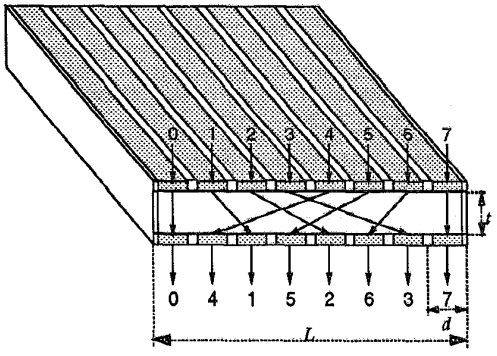

Figure 7: An optical implementation of the column-wise (or row-wise) 1D perfect shuffle (PS) operation on 8 columns (or 8 rows). If we swap inputs and outputs, it can perform the column-wise (or row-wise) 1D inverse perfect shuffle operati on.

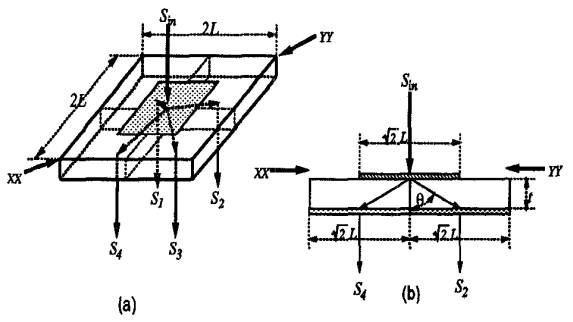

Figure 8: An optical implementation of the 1 x 4 fanout element (or 4 x 1 fanin element if we swap the input and the output). (a) a 3-D view, (b) a cross sectional view along line XX-YY.

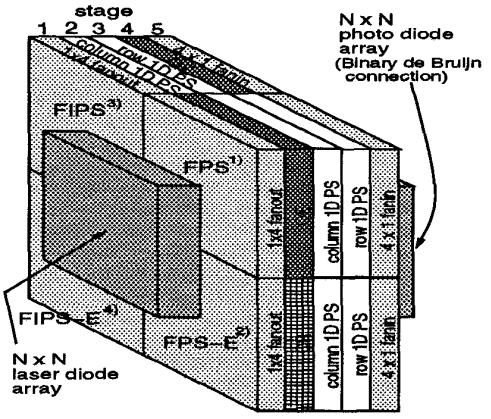

Figure 9: Integration of fundamental optical operations to contruct the BdB network. Fanouts indicated by 1), 2), 3), and 4) correspond to FPS, FPS-E, FIPS, and FIPS-E operations, respectively.

	binary n-cube	$2^{n/2} \times 2^{n/2}$ mesh	n-BdB
total number of nodes	2^n	2^n	2^n
node degree	n	constant (4)	constant (4)
total number of links	$n/2 \times 2^n$	2×2^n	2×2^n
diameter	n	$2^{n/2}$	n
fault tolerance	Yes	Limited	Yes
routing	easy self-routing	easy self-routing	easy self-routing
detours around faults	difficult	difficult	easy
embeddability of other topologies	linear array, mesh	binary tree, tree machine	linear array, binary tree shuffle-exchange, linear array

Table 1: Characteristics of the binary n-cube, the two-dimensional mesh, and the n-BdB networks. It is assumed that the mesh network has wraparound connections in the rows and columns. The table shows that the n-BdB network is an attractive alternative over binary n-cube and the mesh network.