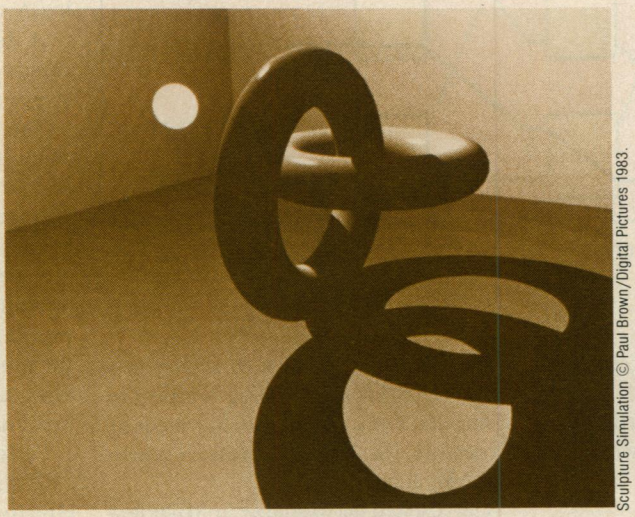
Guest Editor's Introduction

Interconnection Networks for Parallel and Distributed Processing

Laxmi N. Bhuyan

University of Southwestern Louisiana

With the rapid advances in technology, it is now feasible to build a system consisting of hundreds or thousands of processors. Processors in such a parallel/distributed system may spend a considerable amount of time just communicating among themselves unless an efficient interconnection network (IN) connects them. A complete interconnection such as crossbar may be cost prohibitive, but a shared-bus interconnection may be inefficient and unreliable. We need to design INs with both cost and performance between these two extremes.



Sculpture Simulation © Paul Brown/Digital Pictures 1983.

Parallel or distributed computers can generally be divided into two categories: multiprocessors and multicomputers. The main difference between the two lies in the level at which interactions between the processors occur.

A multiprocessor system must permit all processors to directly share main memory, as shown in Figure 1a. In a multicomputer system, however, each processor has its own local memory. Sharing between the processors occurs at a higher level, through a complete file or data set. As shown in Figure 1b, a processor cannot directly access another processor's local memory.

An IN is a complex connection of switches and links that permits data communication between processors and memories in a multiprocessor system or between the processors in a multicomputer system.

A multiprocessor or a multicomputer architecture is further characterized by the topology of the IN it uses. Current multiprocessor organizations are based on crossbar, multistage interconnection networks (MINs) and multiple-bus networks, as shown by the first four articles in this special issue. Multicomputer architectures include topologies such as star, ring, tree, and hypercube, as shown by the last article in this issue.

Classification of INs

Generally speaking, any IN has three operational characteristics. These are based on its timing, switching, and overall control.

The timing control of an IN can be either synchronous or asynchronous. Synchronous systems are characterized by a central global clock that broadcasts the clock signal to all devices on the IN so that they operate in a lockstep fashion. Asynchronous systems, on the other hand support independent operation of the devices without a global clock.

An IN transfers data using either circuit switching or packet switching. In circuit switching, once a device is granted a path in the IN it will occupy that path for the duration of the data transfer. In packet switching, the information is broken into small packets that individually compete for a path in the IN.

Based on the overall control of the network, an IN may be classified as centralized or decentralized. In centralized control, a global controller receives all requests and transmits the messages in the IN. In a decentralized system, requests are handled independently by different devices in the IN.

0018-9162/87/0600-0009$01.00©1987IEEE

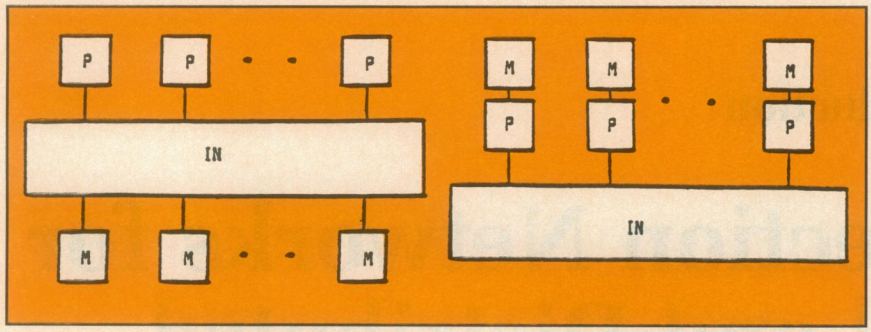


Figure 1. A multiprocessor architecture (a) and a multicomputer architecture (b).

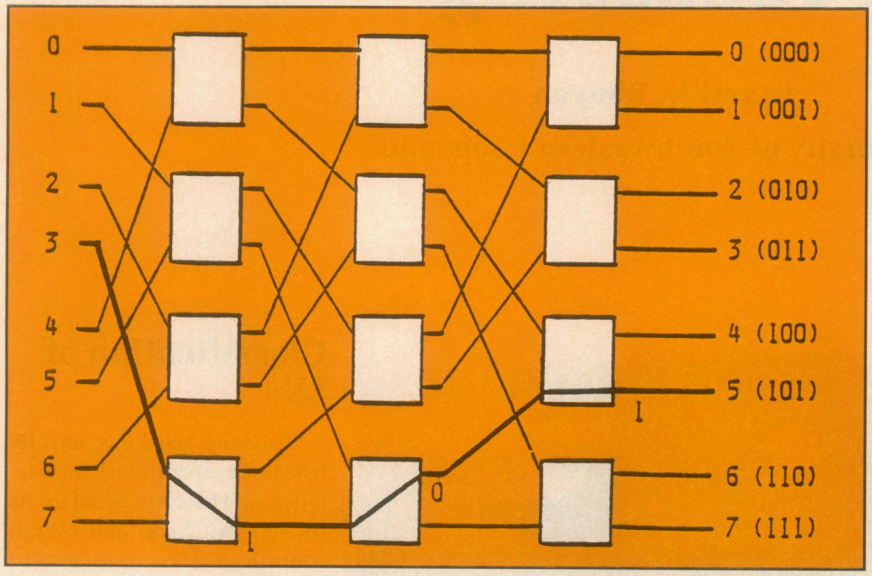


Figure 2. An 8 × 8 omega network.

These three operational characteristics with the topology define an IN.[1] For example, the Butterfly parallel processor uses an asynchronous, packet-switched, decentralized MIN.

Multiprocessor INs

Any processor in a multiprocessor system should be able to directly address every shared memory module through the IN. As a result, the performance of a multiprocessor rests primarily on the design of its IN.

A shared-bus interconnection is the least complex, but it does not allow more than one processor at a time to access a shared memory. A large number of processors means a long wait for the bus.

On the other hand, a crossbar supports all possible distinct connections between the processors and memories simultaneously. However, the cost of such a network is $O(N^2)$ for connecting N inputs to N outputs. For a system with hundreds of processors, the cost of such an IN is prohibitively high.

The cost and performance of MINs and multiple-bus networks hit a reasonable balance between those of a shared bus and a crossbar. Design, analysis, and development of MINs during the last decade have made them the most current technology.

The multiple-bus interconnection is somewhat new in the IN arena, as described in detail in the third article in this special issue.

An $N \times N$ MIN connects N processors to N memories. For N a power of two, it employs $\log_2 N$ stages of 2×2 switches with $N/2$ switches per stage. Each switch has two inputs and two outputs. The connection between an input and an output depends on a control bit c provided by the input. When $c = 0$, the input is connected to the upper output. When $c = 1$, it is connected to the lower output.

An omega network,[2] shown in Figure 2, is characterized by a perfect shuffle interconnection preceding every stage of switches. The requesting processor generates a tag that is the binary representation of the destination. The connection of a switch at the ith stage is then accomplished by the ith bit of this binary tag counted from the most significant bit. This self-routing property of an MIN avoids the need for a central controller, making it suitable for multiprocessors.

The connection between input 3 and output 5 (101_2) is shown by a bold line in Figure 2. Many significant MINs, such as SW Banyan, generalized cube, and base line,[1] have been proposed, perhaps confusing the community at large. However, all these networks are similar except for the interconnection between the adjacent stages.

The switch size in an MIN need not be restricted to 2×2. In fact, the Butterfly parallel processor connects N inputs to N outputs using 4×4 crossbar switches and $\log_4 N$ stages with $N/4$ switches per stage.

A delta network can connect $M = a^n$ inputs to $N = b^n$ outputs through n stages of $a \times b$ crossbar switches.[1]

The generalized shuffle network can connect any $M = m_1 * m_2 * \ldots * m_r$ to $N = n_1 * n_2 * \ldots * n_r$ outputs through r stages of switches.[3] The ith stage employs $m_i \times n_i$ crossbar switches and is preceded by a generalized shuffle interconnection that is essentially a superset of the omega and delta interconnections. This is the most generalized version of an MIN reported so far, in that it allows different input and output sizes.

Performance analysis shows that an MIN using 4×4 switches is more cost effective than one with 2×2 switches.[3]

Researchers recognized the advantages of MINs and started many academic and industrial research projects. University projects include TRAC at Texas, PASM at Purdue, Ultra-Computer at New York, and Cedar at Illinois. RP3 is a notable industry project at IBM and Butterfly is a successfully marketed product from BBN Laboratories Inc.

As these projects were starting, a serious drawback of MINs surfaced. There is only one path from an input to an output. It was necessary to incorporate some fault-tolerance into these networks so that at least a single fault in a switch or a link could be tolerated. This has given rise to an abundance of research during the past

few years devoted to the design and evaluation of fault-tolerant MINs. The first two articles in this special issue are devoted to this relatively new area.

Multicomputer INs

Several structures for connecting a large multicomputer have been proposed in the literature.[1] These networks usually fall into the asynchronous, decentralized, packet-switched category. A dedicated link exists between two processors (nodes) over which the communication between them occurs. A message generated at a source node may go through a number of intermediate nodes in a store-and-forward manner before reaching the final destination.

Although several measures have been suggested in the literature, the characteristic of a symmetric IN can be easily expressed in terms of the degree of a node and the diameter of the IN. The degree of a node is the number of links per node. This reflects the cost associated with the network. The diameter is the maximum number of links a message has to travel between any source and any destination along the shortest path. Usually the networks that have a lower degree for each node give rise to a higher diameter, which means more delay in communication. A higher degree node reduces the diameter, but at a higher cost.

A bidirectional loop and completely connected structures represent these extreme examples, respectively.[1] The goal of research in the multicomputer IN area is, therefore, to design a network that has a low degree as well as a low diameter. The boolean n-cube or hypercube[4,5] represents a well-balanced structure between the above two extremes. It is described below.

A three-dimensional hypercube computer is shown in Figure 3. In general, a network with $N = 2^n$ nodes is arranged as an $n = \log_2 N$ dimensional hypercube with two nodes in each dimension. The nodes are represented by binary equivalents of the decimal numbers between 0 and $N-1$ where adjacent nodes differ by one bit. The distance between the source $X = x_{n-1}x_{n-2} \ldots x_1x_0$ and destination $Y = y_{n-1}y_{n-2} \ldots y_1y_0$ is equal to the number of bits by which X and Y differ in their binary representations. The diameter of this network is n and there exist n disjoint paths between a source and a destination. For example, in Figure 3 a message from 0 to 7 can be routed through any one of the following paths:

$000 \rightarrow 001 \rightarrow 011 \rightarrow 111$
$000 \rightarrow 010 \rightarrow 110 \rightarrow 111$
$000 \rightarrow 100 \rightarrow 101 \rightarrow 111$

Hence the structure is highly fault-tolerant. The number of I/O ports per node is again $n = \log_2 N$, which many critics consider too high for large values of N. However, the structure is easily implementable, as evidenced by many hypercube products, such as the Intel iPSC, Ametek S/14, and NCUBE NCUBE/ten hypercube computers. Benchmarks on some of these computers are given in the last article of this special issue.

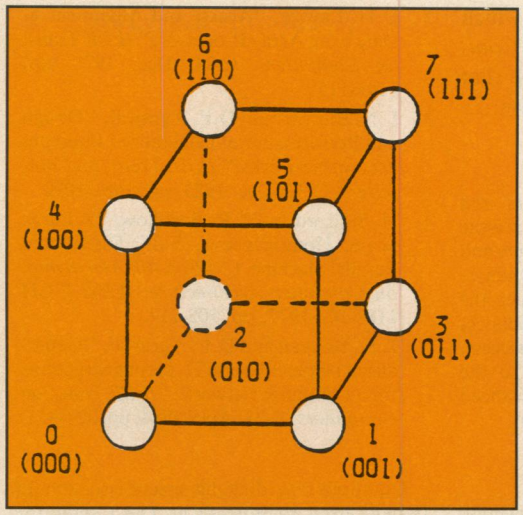


Figure 3. A hypercube network with eight nodes.

Bhuyan and Agrawal[5] extended the hypercube design to any number of nodes N, in contrast to $N = 2^n$ for traditional hypercubes. If $N = m_{n-1}*m_{n-2}* \ldots * m_1*m_0$ for integer values of m and n, an n-dimensional hypercube can be obtained with the ith dimension containing m_i nodes.

Analysis of such a generalized hypercube shows that it is more cost effective to build a system with four nodes in each dimension, compared to two in normal hypercubes.[4]

About this issue

The article by Adams, Agrawal, and Siegel in this issue is a first attempt to survey fault-tolerant networks. It compares the properties of different networks and provides the reader with the state of the art in this area. The second article, by Kumar and Reddy, focuses on a particular fault-tolerance technique that can be applied to shuffle exchange MINs. The reader, therefore, is exposed to the basic concept of fault-tolerance as applicable to MINs.

During the past three to four years, a lot of interest has developed in multiple-bus INs. Bus structures, in general, are easily understood. Multiple-bus systems can be viewed as an incremental expansion of many single-bus multiprocessor architectures. In the third article of this issue, Mudge, Hayes, and Winsor provide a tutorial on this new type of IN, including a hypothetical design of a multiple-bus multiprocessor.

The use of optics for communication purposes has been increasing dramatically, and its use in local area networks has received considerable attention. The article by Sawchuk, Jenkins, Raghavendra, and Varma provides different designs and discusses various trade-offs for the implementation of optical crossbar networks. Unlike their electronic counterparts, it seems feasible to build moderately large optical crossbars in a physically compact unit.

A number of structures or topologies such as loop, tree, full connection, and hypercube have been proposed to connect a network of computers. In the last article of this issue, Reed and Grunwald present a comparative study of these multicomputer networks. They also supplement their theoretical studies with benchmarks they obtained on a hypercube computer.

The development of this issue was guided by a desire to concentrate on interconnection network research that has emerged during the past five years. Thus we intend to bring the reader up-to-date since the last special

issue on interconnection networks, which appeared in the December 1981 *Computer*, edited by C. L. Wu. I hope that readers will find the issue interesting. □

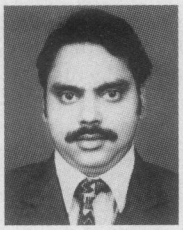
Acknowledgments

I would like to thank the authors and reviewers who helped to make this special issue a reality. They had severe time constraints and worked even during the Christmas holidays. Special thanks are due to Bruce D. Shriver, the editor-in-chief, for his constant guidance, encouragement, and help in the review process. Finally, my deep appreciation goes to the secretarial staff at The Center for Advanced Computer Studies for their timely help.

References

1. T.Y. Feng, "A Survey of Interconnection Networks," *Computer*, special issue on Interconnection Networks, Dec. 1981, pp. 12-27.
2. D.H. Lawrie, "Access and Alignment of Data in an Array Processor," *IEEE Trans. on Computers*, C-24, Dec. 1975, pp. 1145-1155.
3. L.N. Bhuyan and D.P. Agrawal, "Design and Performance of Generalized Interconnection Networks," *IEEE Trans. on Computers*, C-32, Dec. 1983, pp. 1081-1090.
4. H. Sullivan and T.R. Bashkow, "A Large Scale, Homogeneous, Fully Distributed Parallel Machine, I," *Proc. 4th Ann. Symp. on Computer Architecture*, IEEE 77CH 1182-5C, 1977, pp. 105-117.
5. L.N. Bhuyan and D.P. Agrawal, "Generalized Hypercube and Hyperbus Structures for a Computer Network," *IEEE Trans. on Computers*, C-33, April 1984, pp. 323-333.

Questions regarding this special issue should be addressed to Bhuyan at The Center for Advanced Computer Studies, University of Southwestern Louisiana, PO Box 44330, Lafayette, LA 70504.



Laxmi N. Bhuyan is an associate professor in The Center for Advanced Computer Studies at the University of Southwestern Louisiana, in Lafayette. His present research interests include parallel and distributed computer architecture, performance and reliability evaluation, and local area networks.

Bhuyan received his BS and MS in electrical engineering from the Regional Engineering College, Rourkela under Sambalpur University, India. He received the PhD in computer engineering from Wayne State University in Detroit in 1982.

Bhuyan is a senior member of the Computer Society.