BANYAN NETWORKS
FOR PARTITIONING MULTIPROCESSOR SYSTEMS

By

Louis Rodney Goke

A DISSERTATION PRESENTED TO THE GRADUATE COUNCIL OF
THE UNIVERSITY OF FLORIDA
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

1976

ACKNOWLEDGMENTS

I wish to thank my advisor, Dr. G. J. Lipovski, for his inspiration and guidance concerning both this dissertation and computer architecture in general. It was largely for the opportunity to work with and to learn from Dr. Lipovski that I chose to continue at the University of Florida beyond the master's program, and I feel that his influence has contributed greatly to my technical and professional preparation. The research presented here began as an outgrowth of his earlier work with SW switching structures (Lipovski, 69, 70), and was undertaken originally as thesis research for the degree of Engineer. It was developed, instead, into a doctoral dissertation largely as a result of Dr. Lipovski's encouragement.

Portions of this dissertation have evolved piecemeal over a period of years, and I am indebted to a large number of typists and others who have assisted in document preparation. Most notable has been the contribution of Ms. Sylvia Hansing, who has had the perseverance and the skill to type the entire final manuscript in Mag Card form.

Special thanks go to my wife, Mary Goke, for her tolerance and understanding and for the variety of ways in which she has assisted me during the preparation of this dissertation.

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

LIST OF TABLES

LIST OF TABLES (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

Abstract of Dissertation Presented to the Graduate Council
of the University of Florida in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy

BANYAN NETWORKS FOR PARTITIONING MULTIPROCESSOR SYSTEMS

By

Louis Rodney Goke

June, 1976

Chairman: Gerald J. Lipovski
Major Department: Electrical Engineering

There is a strong and growing need for switching structures suitable for interconnecting numerous processors and other resource modules in large, general purpose computing systems. For this purpose, "banyan" network structures are defined and analyzed with the use of graph theory, and their cost-performance characteristics are compared with those of alternative networks.

Techniques are proposed for utilizing banyan networks in large, general purpose, partitionable systems containing numerous microprocessors or other resources. A banyan network used in this manner can partition system resources into a wide variety of task-oriented subsystems and, when necessary, can be multiplexed to realize any possible partition. Techniques are also presented for constructing banyan networks in modular form and for controlling them in a rapid and potentially fault-tolerant manner using distributed logic in the networks themselves.

Banyan partitioning networks are shown to have significant advantages over alternative crossbar, or multiple-bus, structures for use in large

systems. It is shown that banyan cost functions tend to grow more slowly with network size and that banyan networks can be expanded without limit using fixed-fanout devices. Statistical simulation results are presented indicating that banyans have a potential cost-performance advantage over large crossbar-based partitioning networks and that this advantage tends to increase with network size.

Graph theory and APL vector operations are used in characterizing theoretical banyan properties. Various subclasses of banyans are defined, providing a taxonomy of network structures and permitting the derivation of additional useful properties. The analysis is oriented towards the use of banyans as partitioning networks, but it is noted that a variety of networks proposed previously for other purposes are structurally equivalent to special cases of banyans, suggesting that the theory of banyan graphs could have much broader applications.

SECTION 1

INTRODUCTION

1.1 The Trend Towards Numerous-Module Systems

There is reason to believe that large data processing systems of the future will tend more and more to contain numerous small resource modules instead of a few large ones. For example, a computing facility requiring a large amount of processing power might contain a number of miniprocessors or microprocessors instead of a single large processor. Similarly, its primary memory might be provided by numerous low capacity modules rather than a few high capacity modules.

It has long been recognized that modular multiprocessor systems have major potential advantages over single processors in the areas of throughput, reliability, availability, and expandability. Potential throughput advantages are obvious in applications which lend themselves to parallel processing. By automatically reassigning tasks so as to bypass faulty modules, excellent system reliability and availability can be achieved, even if individual module failures are common. Furthermore, it is possible to expand the capacity of a highly modular system quickly and efficiently by simply adding more modules.

Economically, the attractiveness of multiple-processor architectures has grown with progress in integrated circuit technology. With early component technologies, large processors tended to provide the most computing power for the money, and the high costs of processors tended

to make numerous-processor systems prohibitively expensive. Accordingly, early multiprocessor systems tended to contain small numbers of processors and generally were constructed only as research projects or for special applications where adequate performance could not be obtained with conventional machines (Comptre, 74).

When MSI technology and volume production made miniprocessors available for several thousand dollars each, systems with multiple miniprocessors began to be viewed as alternatives to large, single-processor, time-shared systems. These miniprocessors found considerable use as elements of geographically distributed, special purpose systems, and efforts were undertaken to develop reconfigurable, general purpose, multi-miniprocessor systems at two major universities (Baskin et al., 69, 72; Wulf and Bell, 72). Simple cost-effectiveness measures published recently by Bhandarkar and Juliussen (75) indicate that multiminiprocessor systems with up to several tens of processors could have cost-effectiveness advantages over comparable single-processor computers.

More recently, the availability of extremely low cost LSI microprocessors has accelerated interest in multiple-processor architectures, and the trend is expected to continue (Searle and Freberg, 75). Microprocessors became recognized for their excellent cost-performance potential shortly after they were introduced (Schultz et al., 73), and improvements in both cost and performance have progressed rapidly. General purpose processors are now available for several tens of dollars, and substantial further reductions are projected. The excellent cost-effectiveness of these processors suggests that multiple-microprocessor systems are destined to replace large single-processor computers in a number of applications.

The cost, size, and power requirements of microprocessors are now low enough that systems with hundreds or even thousands of processors are reasonable to consider. In fact, a general purpose computer containing up to 512 microprocessors already is being marketed by one manufacturer (Frank, 75).

The trend towards numerous small modules is not limited to processors. Advancing LSI technologies are now challenging core technology for primary memory applications. MOS/LSI memories are beginning to compete with core memories in speed and cost; and faster, but more costly, bipolar memories are finding applications in high-performance machines such as the Texas Instruments Advanced Scientific Computer. Before long, charge coupled devices may be widely used where higher capacity and lower speed are required. Unlike core technology, the newer LSI semiconductor technologies are well suited to the fabrication of numerous small modules.

Clearly, strong incentives now exist for developing general purpose multiprocessor systems containing large numbers of resource modules. Potentially, at least, such systems could have significant advantages over large single-processor computers in the areas of throughput, reliability, availability, expandability, and cost.

1.2 Problems with Previous Interconnection Schemes

Although there are good reasons for developing general purpose multiprocessors with numerous modules, most previously used interconnection schemes are practical only for small or specialized systems. Problems with module interconnection schemes used previously will be discussed briefly in this section. More detailed surveys of these schemes have been published by Comptre (74) and by Searle and Freberg (75).

Crossbar switching structures can support high data rates but are not practical for interconnecting large numbers of modules. The number of "contacts", or switching devices, needed for a crossbar increases with the square of the number of modules connected to it, making the crossbar prohibitively expensive for very large systems. Since the fanout of switching devices in a crossbar increases linearly with the number of resource modules, this too can be a serious problem in large systems, especially if expandability is not to be limited.

A single time-shared bus can provide flexible, inexpensive communication among a small number of modules, but bus contention problems make this approach impractical for large systems. As the number of modules on the bus increases, bus utilization increases, causing the resource modules to waste more and more of their time waiting for a nonbusy bus.

Multiple time-shared busses can be used to alleviate bus contention problems, but this configuration has problems similar to those of a crossbar. In fact, the switching devices which enable each module to be connected with any bus are arranged in a crossbar configuration. Since the maximum data rate that can be handled by a bus is fixed, the number of busses required grows proportionally with the total number of modules,

and the number of switching devices required grows as the square of this number.

Both single and multiple time-shared busses have fanout problems in large systems, because each module on a bus must be capable of driving all other modules attached to that bus. Thus, as is the case with a crossbar, fanout requirements grow linearly with the number of modules. In a very large system, fanout limitations can be overcome by dividing each bus into segments interfaced by bidirectional amplifiers, but this further increases network cost and increases the time required for signals to propagate across a bus.

"Multiport" systems, in effect, use crossbar structures to interconnect various classes of modules, and hence, they have cost and fanout problems like those of crossbars.

A number of vector and array organizations have been devised in which each resource module (usually containing a processor with memory) can communicate directly with only a fixed number of "nearest neighbors." These organizations generally can be extended to very large sizes without severe cost or fanout problems but are not well suited to general purpose use. Since each module can communicate directly with only a few neighbors, considerable overhead is required whenever logical data-flow patterns do not correspond closely with the hardware interconnection pattern. It has been found, generally, that efficient software is difficult to produce for these machines unless the application "fits" the architecture. Also, when a module fails, there may be no efficient way for another module to take its place, since the substitute module is not likely to have the same neighbors as the module that failed. The ILLIAC IV (Barnes et al., 68) is typical of this kind of organization.

Various distributed architectures have been used successfully in dedicated-function applications but are not well suited to large-scale, general purpose use. Efficiency in a large distributed system usually derives from the fact that each module or subsystem performs a specialized function and needs to communicate with other modules or subsystems only in very limited ways known at design time. This, of course, is not feasible in a general purpose system, which must be designed for a variety of applications not necessarily known at design time.

Thus, interconnection schemes used previously in multiple-processor systems generally are undesirable for generalpurpose systems with very large numbers of modules, because their cost and fanout functions grow too rapidly with system size, because performance degrades with system size, or because they perform well only in specialized applications.

1.3 Banyan Partitioning Networks

A class of connecting networks suitable for interconnecting large numbers of resource modules in a general purpose, multiple-processor system will be defined and analyzed in this dissertation. These networks, called banyans, will be analyzed for their ability to partition the resources of a modular system into task-oriented subsystems. The work presented here concentrates on the use of banyans as partitioning networks, because this mode of operation is particularly applicable to large, general purpose, multiple-processor systems. Banyan networks also can be used in other ways, but nonpartitioning applications are beyond the scope of this dissertation and will be discussed only in relating this to previous work.

Banyan partitioning networks offer a great deal of flexibility for general purpose use and have major practical advantages for use in large systems. They can economically partition the resources of large modular systems into a wide variety of subsystems. Any possible partition can be realized by paralleling several networks or by multiplexing a single network in a manner to be described later. Banyans are potentially much more economical than crossbar-based structures for large systems, because their "cost" functions increase more slowly with system size and also because they have easily satisfied fanout requirements that are independent of system size. Results will be given indicating that a cost-performance advantage over crossbar-based structures can be achieved for large systems and that a crossbar structure actually can be considered a nonoptimal special case of a banyan structure. Propagation delays through a banyan network grow only logarithmically with system size, and high intrasubsystem data transfer rates can be sustained regardless of

the number of subsystems realized by the network. Banyan advantages also include failsoft capabilities, isolation of independent jobs, and rapid control algorithms which can be performed largely by distributed logic in the network itself.

A graph theoretic analysis of banyan structures will be presented. Useful theoretical results will be derived concerning the structure, cost, and performance of different classes of banyan networks. The analysis is oriented towards the use of banyans as partitioning networks, but it is expected that the theory of banyan graphs could have much broader significance. Structures graphically equivalent or similar to restricted classes of banyans have been proposed for a variety of data manipulation functions, including sorting, shifting, and permuting; so banyan theory potentially could be applied in these areas and could tie together a number of previously unrelated works.

For the reader's convenience, all proofs and other tedious theoretical material will be separated from the main text and presented in an appendix.

Simulation results will be presented concerning certain performance-related issues not resolved analytically. These results, together with those derived theoretically, will be used in assessing the cost-performance potential of banyan partitioning networks.

Most major results presented in this dissertation were published in an earlier paper (Goke and Lipovski, 73).

SECTION 2

PARTITIONING NETWORK CONCEPTS AND REQUIREMENTS

The purpose of this section is to explain how a partitioning network could be used in a general purpose, multiple-processor system and to identify requirements this application would be likely to impose on a partitioning network. A partitionable system architecture will be presented in Section 2.1. The ways in which a system of this type could be applied to different classes of problems will be discussed in Section 2.2. Requirements imposed on a partitioning network by this kind of use will be identified in Section 2.3.

2.1 Architecture of a Partitionable System

The basic architecture of a partitionable data processing system is illustrated in Figure 2.1-1. The system contains a number of resource modules, such as processors, memory modules, mass storage devices, I/O devices, or even complete computer systems. Each module is connected to a partitioning network through one or more ports. These ports generally would be bidirectional so that a module could both transmit and receive signals through the same port. For example, an input device might receive requests for data and then transmit the data requested.

The purpose of the partitioning network is to establish necessary communication paths by connecting ports together to form subsystems. For each subsystem, the partitioning network, in effect, provides a separate time-shared bus to which all ports in that subsystem are connected. Thus, the partitioning network partitions the ports of system resource modules into subsystems, and a particular resource module may belong to as many subsystems as it has ports.

The connections established by the partitioning network typically would be controlled by an operating system or executive and would be modified automatically to accommodate different job needs. The operating system, like any other job, could be executed by a set of resource modules linked as a subsystem.

To facilitate operating system functions, some facility for passing system control messages outside the partitioning network would probably be needed. For example, user subsystems might send messages to the operating system in order to request additional resources, to release resources no longer needed, to signal job completion, or to request other changes in system configuration. Similarly, the operating system might

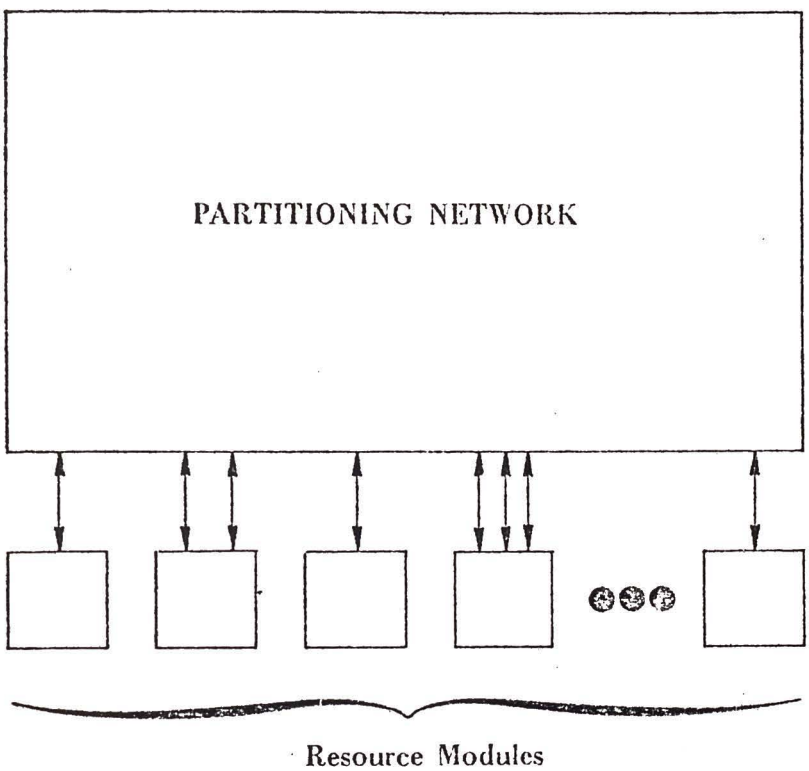


Figure 2.1-1. Basic Architecture of a Partitionable System

direct system configuration changes by sending commands to the partitioning network, and it might send messages to user subsystems in order to control execution and to acknowledge service requests. Control messages such as these would likely be short and infrequent but generally should be transmitted quickly. Communication of this sort could be provided by a single time-shared bus or other simple facility linking together all processor modules and the control inputs of the partitioning network as illustrated in Figure 2.1-2. This facility for passing system control messages would tend to be a critical, but relatively inexpensive, part of a large partitionable system; so the straightforward use of redundant hardware in this facility could prevent it from limiting system reliability and would add little to overall system cost.

The architecture presented here has a number of attractive features. By effectively providing a separate bidirectional bus for each subsystem, the partitioning network allows data transfers within each subsystem to take place at high rates and with little delay. Single time-shared busses are now widely used for interconnecting resource modules in small computing systems, and a variety of devices now on the market can be interfaced in this manner.

Security problems in a multiprogramming environment are greatly simplified by the fact that disjoint subsystems readily can be established to execute independent jobs. Since each subsystem bus functions independently from all others, subsystems or sets of interconnected subsystems cannot interfere with each other unless they share one or more resource modules.

Excellent potential exists for achieving high system reliability and availability, because each module potentially can be connected directly with any set of other modules by grouping them together as a

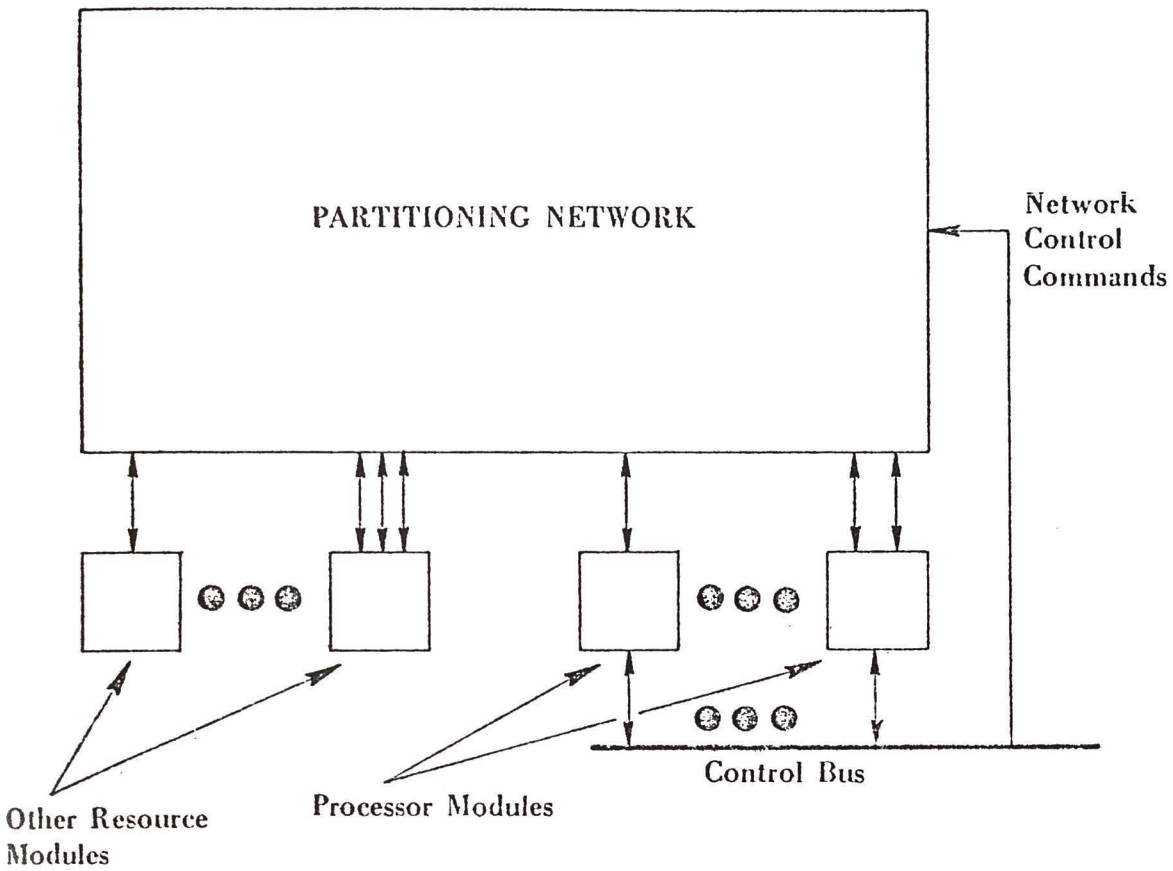


Figure 2.1-2. A Partitionable System with Special Bus for Control Messages

subsystem. Thus, if a module failed, the partitioning network could connect an equivalent spare module in its place, and the repaired subsystem could continue processing as efficiently as before.

The flexibility of this architecture makes it attractive for general purpose use. As will be shown in Section 2.2, it is suitable for most applications of large computers and, for most purposes, could be programmed in a very straightforward manner.

2.2 Utilization of a Partitionable System

The system architecture described in Section 2.1 is extremely flexible and could be used in many of the application areas currently dominated by large single-processor computers. It allows each job to be executed by a set of resource modules interconnected to emulate a system architecture appropriate for that job. Conventional batch jobs and small real-time jobs could be executed by isolated subsystems, each configured to function as a single-processor computer dedicated to its own job. More demanding jobs encountered in many real-time and near-real-time applications could be handled by sets of subsystems linked via shared resources to form distributed processing networks or parallel processing arrays. Potentially, any of these configurations could coexist with others and could be assembled, disassembled, or modified under operating system control to satisfy changing job requirements. In the remainder of this section, several basic techniques will be discussed for applying a partitionable system to common types of data processing problems.

The simplest way to use a partitionable system is to configure an isolated subsystem for each job. An isolated subsystem is one whose resource modules are functionally independent from those of other subsystems. Except for possible communication with a central operating system, each isolated subsystem would function independently as a separate computer system. Different numbers and types of resource modules could be assigned to different subsystems according to job requirements, and if necessary, modules of a subsystem could be added or deleted by the operating system in response to service calls from the subsystem.

Figure 2.2-1 shows a typical isolated subsystem which might be used to execute a small batch job. The subsystem bus connects the

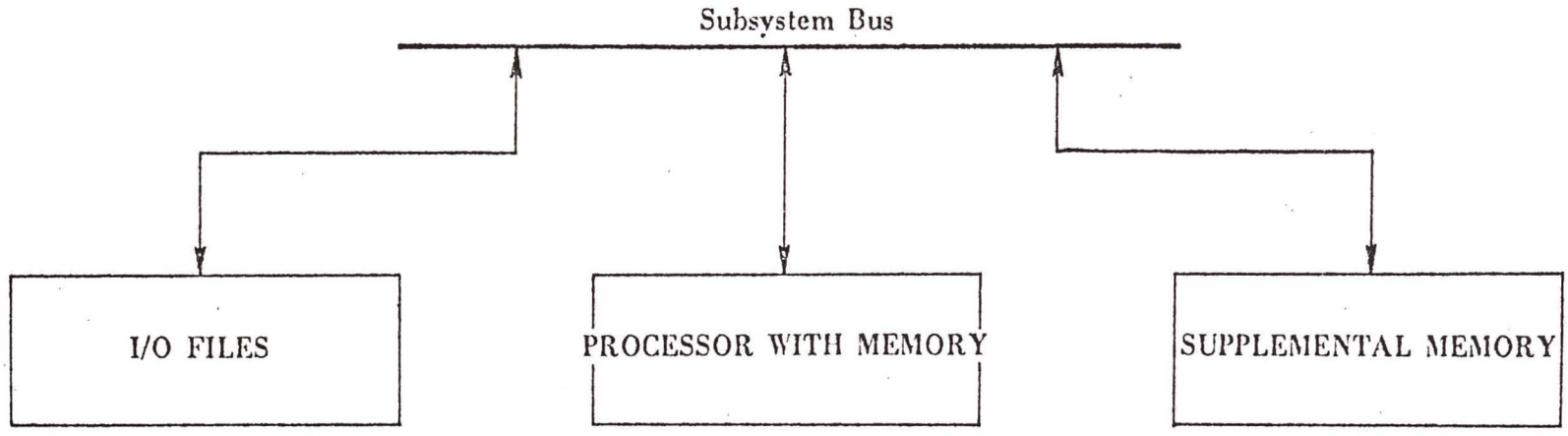


Figure2.2-1. Example of an Isolated Subsystem

necessary resource modules so that they can function together as a small single-processor computer. Although the resource modules are functionally independent from those of other subsystems, some of them may be physically combined with those of other subsystems. For example, the module labeled "I/O Files" might actually be one of several ports to a larger file management system, which makes its ports appear to be independent by restricting the files accessible through each port.

An isolated subsystem could be used effectively for any job not requiring a large amount of processing power. Batch programs and small real-time programs could be written in a conventional manner and could be executed by single-processor subsystems so that, in effect, each program would have its own small computer. High system throughput could be achieved by executing many jobs concurrently with different subsystems.

Multiple processors could be interconnected for jobs too demanding for a single processor. This might be the case in certain real-time or near-real-time applications or for very large batch jobs that would take too long to execute otherwise. In some cases, two or more processors could be assigned to an isolated subsystem, but severe bus contention problems would be likely if many processors shared the same subsystem bus.

A more practical approach for unusually demanding jobs would be to link two or more subsystems together via shared resource modules as illustrated in Figure 2.2-2. A number of multiport modules might be included in a system for this purpose. For example, a number of multiport memory modules with arbitration hardware could be provided. To increase versatility, the memory areas accessible through each port could be limited

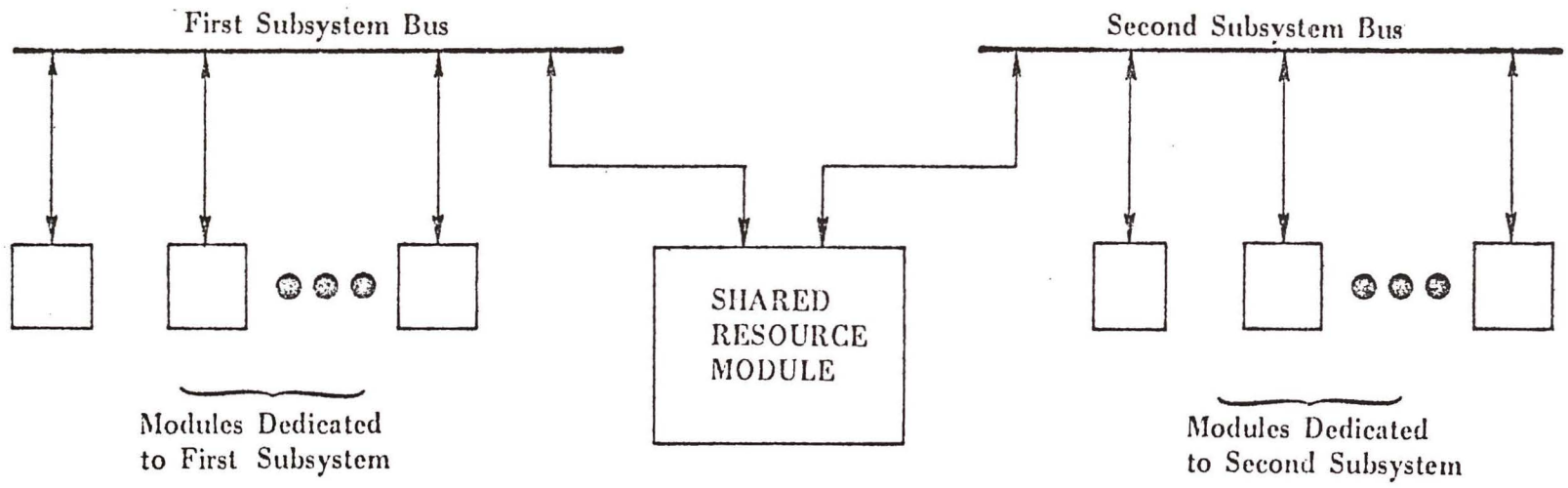


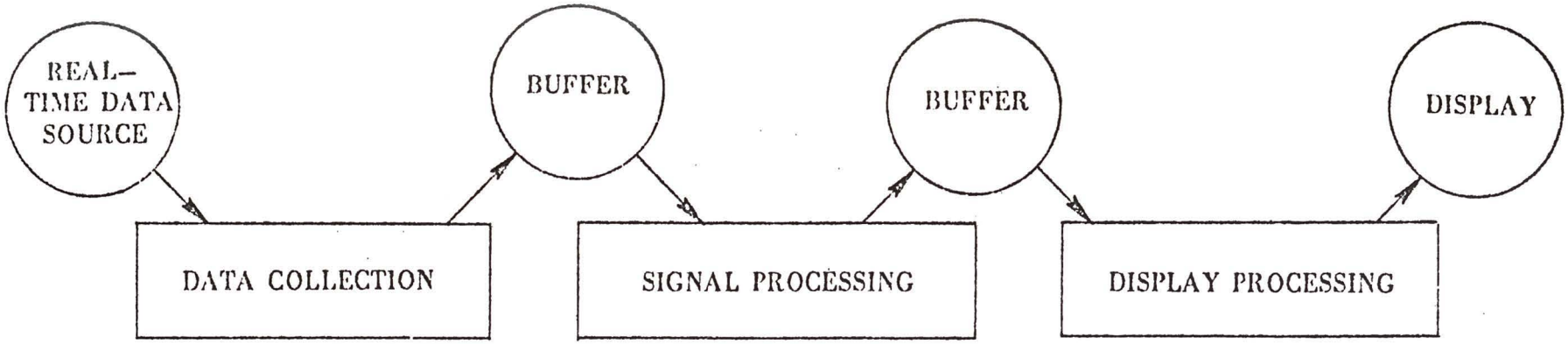
Figure 2.2-2. Subsystems Linked by a Shared Resource Module

by registers associated with the port so that each port could be used as a separate memory module when shared memory was not required.

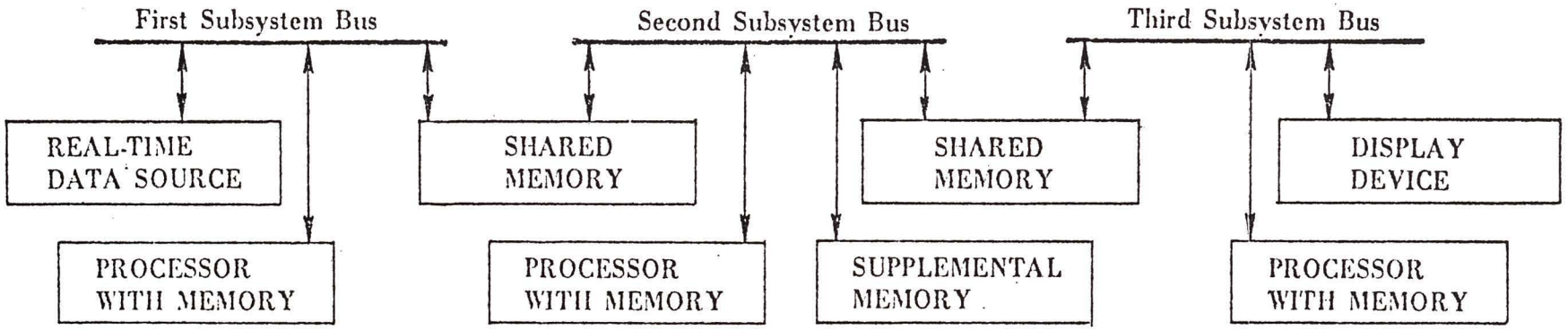
Each set of linked subsystems could be configured to function as a distributed network or parallel processing array suited to the needs of its associated job. This technique would enable application programmers to run unusually demanding jobs on partitionable systems, and also could provide a fast, economical means for system designers to emulate special purpose distributed networks and array processors prior to construction.

Figure 2.2-3 illustrates how a set of subsystems might be linked for distributed processing in a simple near-real-time application. The subsystems are cascaded in assembly line fashion with each subsystem performing a different kind of processing function. Thus, throughput capacity of the distributed system is enhanced by pipelining. A data-flow diagram of the distributed system is shown in Figure 2.2-3a, and the organization of resource modules into linked subsystems is shown in Figure 2.2-3b. The first subsystem collects data from a real-time source, does some initial processing on it, and writes the results into a buffer in shared memory. The second subsystem reads data from this buffer, applies a signal processing algorithm, and writes its results into a second buffer. The third subsystem takes data from the second buffer, arranges it into the desired output format, and writes it to a display device.

Distributed configurations like this, which process data in assembly line fashion, should be relatively easy to design and program. Data-flow diagrams for more complex systems could be designed in a systematic, top-down manner much as sequential programs are in certain "structured



a) Data-Flow Diagram



b) Resource Module Interconnections Established by Partitioning Network

Figure 2.2-3. Example of Subsystems Linked for Distributed Processing

programming" methodologies. Once a data-flow structure has been fully defined, each processing node could be implemented with a single-processor subsystem programmed like a conventional computer. For example, a subsystem which continually reads data from one buffer and writes its results into another could be programmed as if it were a conventional computer reading data from an input device and writing results to an output device. Any mutual exclusion protocols required for accessing shared buffers could be built into I/O routines and need not necessarily concern the application programmer. These basic design principles could be applied even to complex systems in which some processing nodes and buffers might have multiple inputs and outputs.

The practicability of distributed-system design is evidenced by the growing use of distributed processing in specialized data processing systems. Continuation of this trend, no doubt, will lead to improved design techniques and to more widespread familiarity with distributed systems among designers and programmers.

Subsystems also could be linked for parallel processing of array data. With this approach, a number of subsystems would be interconnected in a regular pattern, and each subsystem would perform the same processing function on a different data stream.

One of many possible array processing configurations is shown in Figure 2.2-4. In this example, four subsystems are linked in a rectangular pattern for multiplying a pair of large matrices. A $(2\times I)$ by J matrix $\underline{A}$ is to be multiplied times a J by $(2\times K)$ matrix $\underline{B}$ to obtain a $(2\times I)$ by $(2\times K)$ matrix $\underline{C}$. The matrices $\underline{A}$ and $\underline{B}$ initially are segmented and loaded into the shared memory modules as shown. Each subsystem then computes one

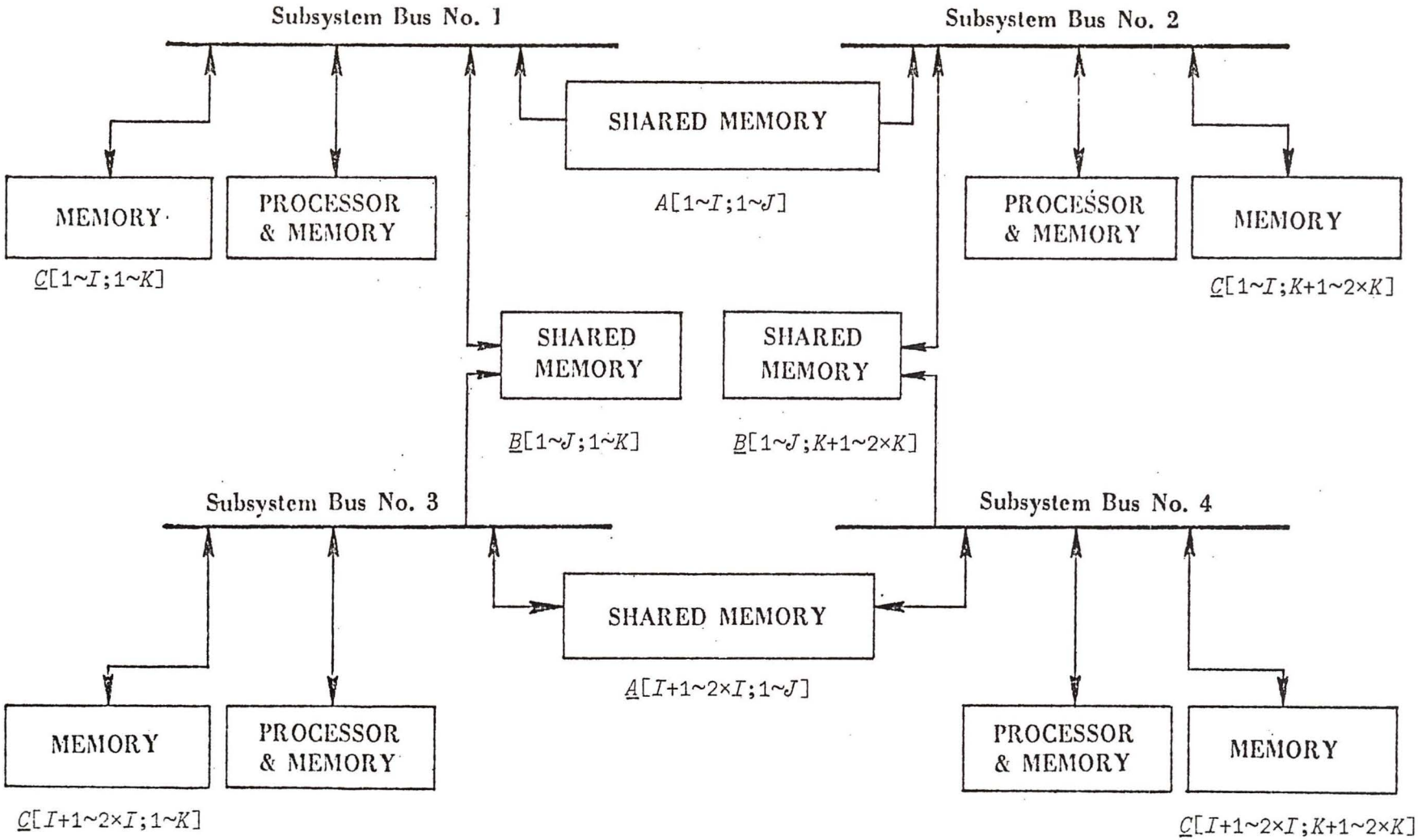


Figure 2.2-4. Example of Subsystems Linked for Array Processing

quadrant of the resulting matrix C and stores it in the memory module shown. By working in parallel, the four processors should be able to perform the matrix multiplication about four times as fast as a single processor.

A number of fixed-configuration array processors have been proposed but generally have been built only as research projects or for specialized applications.[1] There is no doubt that these machines can perform certain kinds of computations at extremely high speed, but efficient software for most of them has been notoriously difficult to produce except for specialized applications where the problem "fits" the machine. A frequent source of programming difficulty and software inefficiency is the fact that processing elements are interconnected in a fixed or nearly fixed configuration, which may not have the best dimensions or interconnection pattern for a given problem. In such situations, one must either use the available processing array inefficiently or employ complicated software techniques to "fit" the problem to the machine.

Parallel array processing in a partitionable system similarly could provide very high speed array computation but need not involve some of the software problems associated with existing array processors. A processing array in a partionable system could be configured with whatever interconnection pattern and dimensions were needed for a particular problem. Consequently, relatively straightforward software could be executed with high efficiency by array processing subsystems.

In a partitionable system, parallel array processing and other forms of processing could be used together easily and efficiently for

---

[1] A number of existing array processors have been surveyed by Comptre (74).

any job with varied computational requirements. For example, a parallel processing array for signal processing might be used as a processing node of a larger distributed system configured for a real-time application; or a large batch job might involve several steps, each of which would run with a different configuration of subsystems. Such flexibility is desirable because there are many applications in which parallel array processing would be useful only for part of the required computations.

In almost any practical situation, an operating system would be necessary for effective utilization of a partitionable system. The kinds of functions performed by such an operating system would be similar to those performed by a conventional multiprogramming operating system except that partitioning network control functions would be performed instead of time-sharing overhead functions. The overhead and complexity normally required for swapping tasks in and out of execution on a single processor would be unnecessary in many situations, because each job or task could be run to completion on its own subsystem or set of subsystems. The software complexity and overhead required for controlling a partitioning network would depend on the kind of network used. As will be shown later, much of the work required for controlling a banyan partitioning network can be performed very rapidly by distributed logic in the network itself.

2.3 Requirements of a Partitioning Network

Partitionable systems have a number of attractive features for a wide range of large-computer applications, but the practicability of this approach depends on the availability of a suitable class of partitioning networks. In this section, we will identify properties that a class of partitioning networks should have for use in partitionable systems.

Cost. Reasonable cost is perhaps the most obvious practical requirement. Generally, a network will be economically practical so long as its cost per resource module port is below some application-dependent limit. Since network cost per port increases with the number of ports, this effectively limits the maximum network size that is economically practical. Thus, in order for a class of networks to be practical for large systems, its cost function should grow slowly with system size. Actual network cost, of course, depends on many factors, including component technology and packaging techniques. For comparing network structures, however, it is common practice to use the number of "contacts", or switching devices, required as a measure of network cost.

Fanout. All major families of electronic switching devices have fanout limitations; that is, each device is capable of driving only a limited number of similar devices. Thus, the number of other devices to which each switching device in a network is connected should grow very slowly, or not at all, with system size in order for a network structure to be practical for large systems.

Bidirectional switching. Data paths, and hence the devices used for switching them, must be bidirectional in a partitioning network so that data can be transferred from any resource module to any other resource module connected to the same subsystem bus. Some bidirectional

electronic switching devices suitable for this purpose have been described by Vice et al. (73). Some additional bidirectional switching circuits using standard TTL and ECL gates are described in Appendix C.

Priority hardware. Whenever several devices communicate over a bidirectional, time-shared bus, some mechanism is needed to prevent more than one device from trying to transmit on the same line at the same time. Priority hardware built into a bus is probabably the fastest and most desirable mechanism for arbitrating simultaneous requests for bus use. For this reason, priority hardware is likely to be needed in a partitioning network to arbitrate conflicting requests for use of subsystem busses. The ease with which suitable priority hardware can be built into a partitioning network is, thus, an important consideration.

Speed requirements. There are three basic response times of interest in a partitioning network: the time required to rearrange connections (probably one subsystem at a time), the time required for a resource module to gain control of its subsystem's bus, and the rate at which a module can transfer data over this bus after obtaining control.

The time required to rearrange connections in a partitioning network depends largely on the complexity of the control algorithms involved and the extent to which these algorithms can be performed by hardware in the network itself, as opposed to sequential execution in an external processor. In a partitionable system used as described in Section 2.2, a subsystem generally would exist long enough for numerous messages to be transferred within that subsystem. Consequently, the time required to establish a new subsystem might be substantially greater than a typical message transfer time without significantly degrading overall performance,

especially if new subsystems can be established without disrupting communication in existing subsystems. In a large system with many subsystems, however, frequent reconfiguration may be necessary even if the average subsystem life is long. In this case, it may be necessary that new subsystems be connected very quickly or that they be connected without disrupting communication in existing subsystems or both. This is an application dependent issue. Fast, simple control algorithms clearly are more desirable than slow, complicated ones, but the importance of this depends largely on the frequency with which the system must be reconfigured.

The time required for a resource to request and receive bus control (assuming that the bus is available) depends primarily on the speed of priority hardware used to arbitrate bus control requests. Since this must be done prior to each transmission, propagation delays in the priority hardware can significantly affect the rate at which short messages can be transmitted. In designing partitioning networks for large expandable systems, one must be careful that neither the propagation delay nor the cost of priority hardware grows unreasonably with system size.

The maximum rate at which a resource module can transfer data after gaining control of its subsystem's bus tends to be inversely proportional to the number of bidirectional switches through which the signal propagates on its way through the network. If the network is multiplexed in a manner to be described later, this maximum rate also tends to be inversely proportional to the number of multiplexed "layers". The maximum rate referred to here is the limit imposed by the partitioning network. The rate at which data is actually transferred, of course, is

limited by the resource modules also. The maximum data transfer rate allowed by the partitioning network should be high enough not to unduly slow down the resource modules. For large expandable systems, the propagation delay and the number of multiplexed layers (if used) should grow slowly with system size.

Fault tolerance. A major advantage of partitionable systems is their potential for fault tolerance. If this potential is to be realized, a system must be able to tolerate hardware failures in its partitioning network as well as in its resource modules. Thus, a partitioning network should be able to continue functioning to at least some extent in spite of limited hardware failures. It is desirable that there be more than one possible way in which to connect any given subsystem, because otherwise a single failure in the network could make certain subsystems impossible to connect, even if no demands are placed on the network by other subsystems. It also should be possible to employ control algorithms adaptable enough to bypass faulty portions of the network when establishing new subsystems.

Modularity and expandability. Modularity and expandability also are advantages of partitionable systems, and it is desirable that a partitioning network share these properties. To minimize production cost and to facilitate maintenance, it should be possible to built a partitioning network by connecting together a number of identical modules, perhaps supplied by a manufacturer as "off-the-shelf" items. It also should be possible to expand a partitioning network such that the old network becomes part of the new one instead of being replaced by it.

Partitioning flexibility. Ideally, we would like for a partitioning network to be able to partition system resource ports into subsystems

in any conceivable way, but complete flexibility in this regard may be unnecessary in practice. Since greater flexibility generally requires greater cost and complexity, it is useful to determine just how much flexibility really is needed in a partitioning network. The ways in which a network actually needs to be able to partition a system depend on the kinds of system resource modules and on the ways in which the system is to be used.

In a practical system, certain kinds of subsystems might be incapable of performing any useful function and, hence, need never exist. For example, a subsystem containing only memories might be unable to function. Similarly, for certain kinds of multiport modules, it may be pointless to ever connect two or more ports of the same module to the same subsystem bus.

In applications requiring only isolated subsystems, such as batch execution of conventional programs, a partitioning network should be able to configure any reasonable subsystem by itself but need not necessarily be able to configure all reasonable combinations of subsystems. If subsystems required for a particular set of jobs cannot all be configured at the same time, then some of the jobs simply can be executed at different times. Thus, jobs can be scheduled to avoid conflicting demands on a partitioning network just as they must be scheduled to avoid conflicting requirements for other system resources. Rescheduling jobs because of partitioning network limitations might result in less efficient resource module utilization, but would allow all jobs to execute eventually, so long as the network could configure each subsystem individually. Isolated subsystems only need to coexist sufficiently for efficient utilization of resource modules.

Linked subsystems, on the other hand, interact during execution and hence must exist concurrently. Consequently, greater partitioning flexibility is likely to be required if a system is to accommodate large sets of linked subsystems. Additional flexibility for configuring arbitrary sets of linked subsystems can be achieved either by inherent properties of a network structure or by multiplexing or paralleling certain less flexible network structures in a manner that will be described later.

SECTION 3

SOME ALTERNATE REALIZATIONS OF PARTITIONING NETWORKS

Two types of partitioning networks, based on crossbars and permutation networks, respectively, will be described in this section. These networks are presented for their conceptual significance in relating partitioning networks to other structures and also to provide a basis of comparison for the banyan networks described in following sections. As will be explained, the networks described in this chapter have certain characteristics which tend to make them impractical for very large systems.

3.1 Crossbar Networks

The network shown in Figure 3.1-1a is perhaps the most straightforward partitioning structure.[1] It contains a number of busses, which are linked with all of the resource modules by bidirectional switching devices. Partitioning is accomplished by assigning one bus to each subsystem and connecting resources to them accordingly. For N resource modules, up to $\lfloor N \div 2$ busses may be required since this is the maximum number of nontrivial subsystems possible at any one time. A subsystem with only one resource module is trivial in the sense that it need not use the partitioning network for intrasubsystem communication.

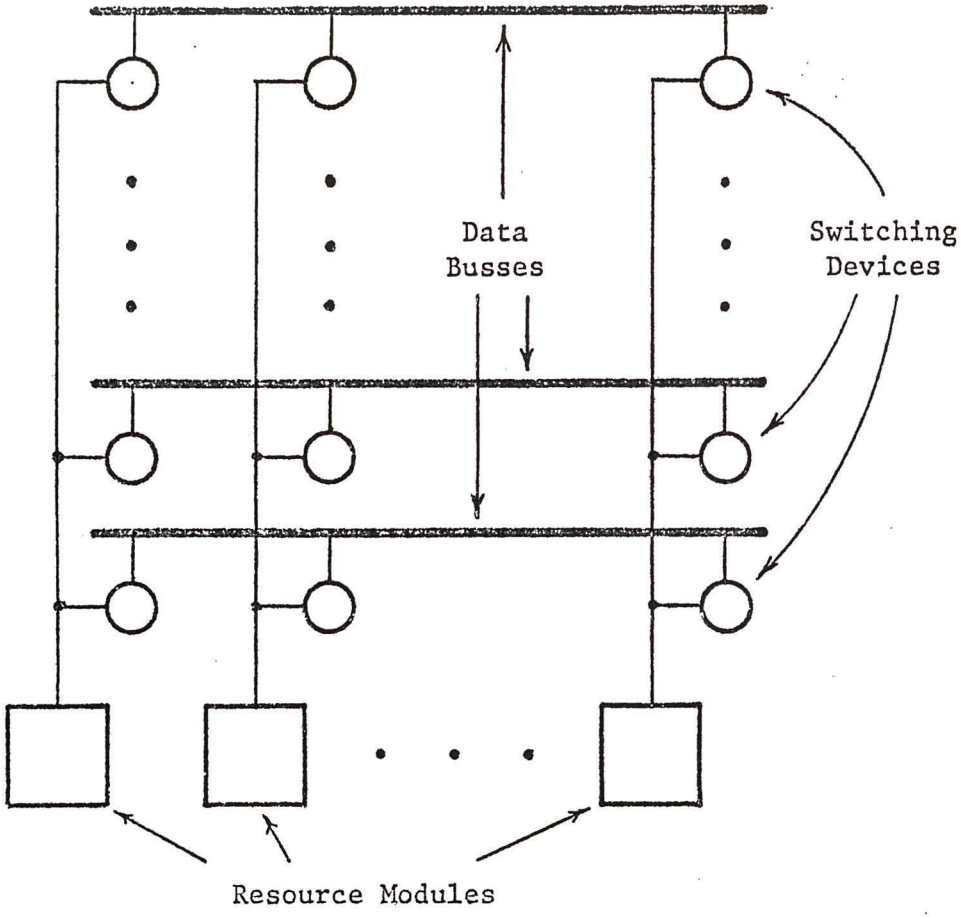
Figure 3.1-1b is a graph representing the structure of this network. This representation of network structure is similar to that used by Benes (62). It uses vertices to represent data busses, or links, and uses edges to represent the switching devices, or "contacts", connecting them. Graph representations of this kind will be used with other structures later.

The network shown is represented by a biparte graph with an edge connecting each bus with every resource module. Graphically, this structure is equivalent to a crossbar switch.

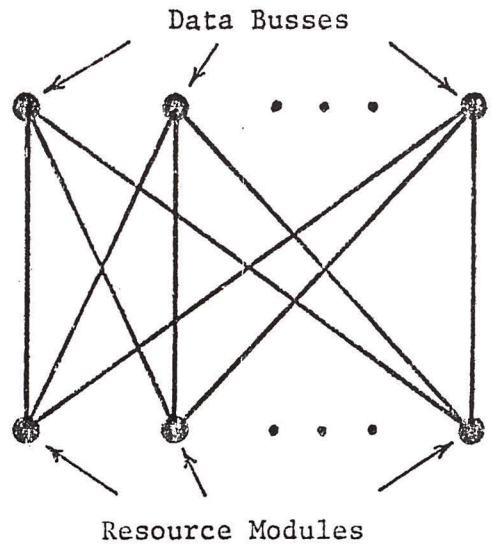
Crossbar partitioning networks have a simple, regular structure. They also have potentially low propagation delay for data transmission since data must propagate through only two switches regardless of network size. Propagation delay for priority hardware, however, would grow logarithmically with N, assuming the use of methods similar to those described by Foster (68). Although faster priority hardware is possible,

---

[1] This structure is equivalent ot the "multiple, system-wide, functionally and physically non-dedicated busses" described by Thurber et al. (72).



a) Block Diagram



b) Graph Representation

Figure 3.1-1. Crossbar Partitioning Network

substantial improvement would most likely be prohibitively expensive in very large systems.

The principal drawbacks of large crossbar networks are their cost and fanout requirements. A network with $\lfloor N \div 2$ busses for partitioning N resource modules would require $N \times \lfloor N \div 2$ switches. Thus, the cost in terms of switching devices required tends to grow as the square of N. Since each switching device in this network is connected to $N-1$ similar devices on the same bus, the fanout required of the devices tends to grow linearly with N. Similarly, each resource module port is connected to $\lfloor N \div 2$ switches; so the fanout capability required of resource modules grows linearly with N also.

3.2 Permutation Networks

It is possible to build a partitioning network from a permutation network by supplying the external links shown in Figure 3.2-1. A permutation network can connect, in pairs, a set of input terminals to a set of output terminals of equal size so that any desired permutation of inputs onto outputs can be realized. These connections allow transmission in either direction when bidirectional switches are used in the network. In the configuration of Figure 3.2-1, the network permutes the set of resource modules onto itself, allowing connected subsystems to correspond to the cycles of the permutation. By choosing a permutation with the appropriate cycles, any desired partition can be connected.

This result is theoretically significant because it implies that a minimal partitioning network for N resource modules requires no more switching devices than an N-input, N-output permutation network. It has been shown that when N is a power of 2, such a permutation network can be built with as few as $4 \times (N \times 2 \circledast N) - N-1$ switching devices (Goldstein and Leibholz, 67; Joel, 68; Waksman, 68). Thus, the cost of this network tends to grow as $N \times \circledast N$, a substantial improvement over that of a crossbar for large N. Further, the fanout required of switches in such networks is independent of N. This too is a substantial improvement over crossbar structures.

The partitioning structure of Figure 3.2-1 is of limited practical value, however, for several reasons. Propagation delay tends to be excessive for data transmission in large subsystems. A signal in a subsystem bus connecting I resource modules may have to propagate through the network as many as $\lfloor I \div 2$ times to reach its destination. Each time through, it must propagate through as many as $(2 \times 2 \circledast N)-1$ switching

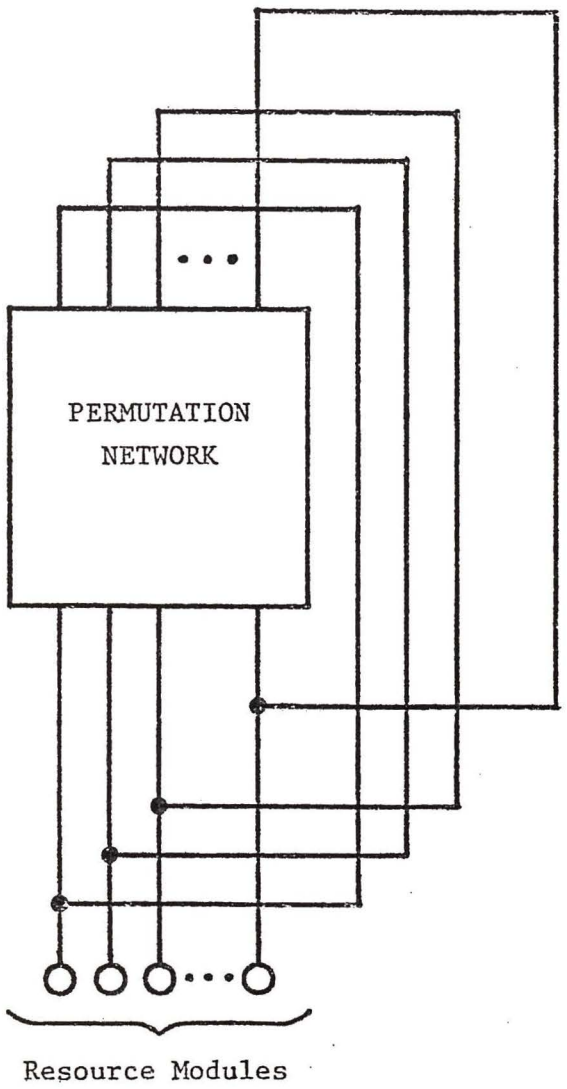


Figure 3.2-1. Permutation Network Used as a Partitioning Network

devices in a minimal-cost permutation network. Network reconfiguration would be hampered by the complexity of control algorithms and by the fact that existing connections may have to be rerouted whenever a new subsystem is added. The difficulty of incorporating priority hardware into this structure also appears to be a serious drawback.

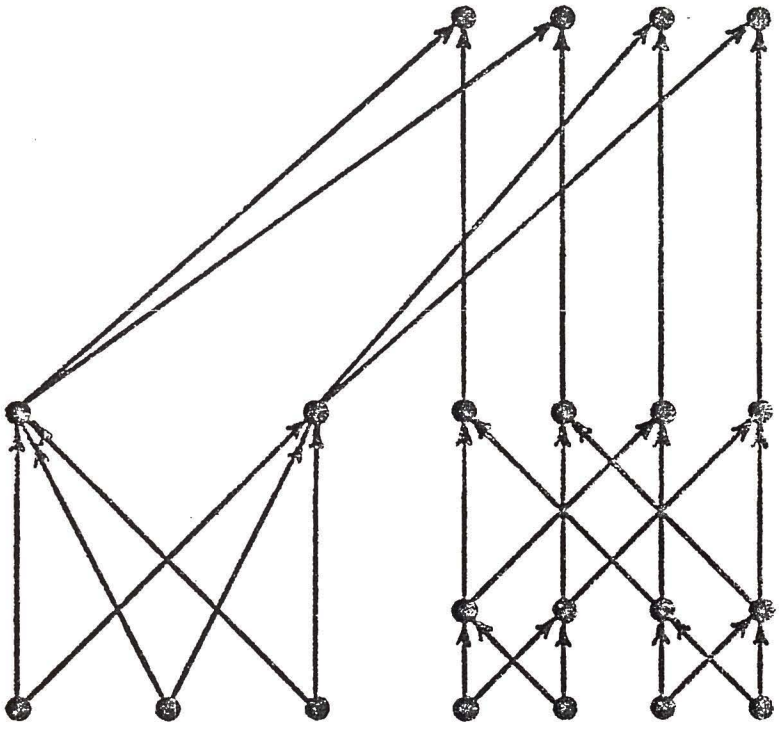
SECTION 4

BANYANS

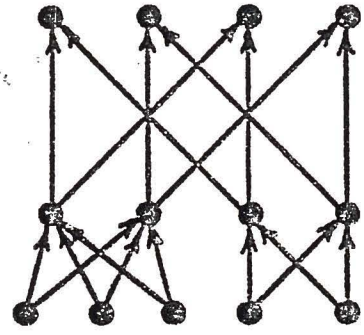
Banyan networks, named for the East Indian fig trees of somewhat similar structure, are defined in terms of their graph representations in Definition 1.1.[1] A banyan graph is a Hasse diagram of a partial ordering in which there is one and only one path from any base to any apex. A base is defined as any vertex having no arcs incident into it, an apex is any vertex with no arcs incident out from it, and all other vertices are called intermediates. When a banyan is used as a partitioning network, its bases are connected to resource modules, but its apexes and intermediates are internal to the network. Some examples of banyans are shown in Figure 4-1. We use a directed graph representation because it is useful for specifying the structure and its control algorithms, but the switching devices represented by edges are still bidirectional. Frequently, we will omit the arrow heads from banyan graph diagrams and let it be understood that all arcs point up.

Useful properties common to all banyan partitioning networks will be presented in Sections 4.1 through 4.5. The general class of banyan networks is quite broad, but it is expected that most useful banyan partitioning networks will be included in the more specialized categories described in Sections 5 through 7.

[1] All definitions and theorems discussed in this paper appear in Appendix B.



a) Irregular Banyan



b) L-Level Banyan

Figure 4-1. Examples of Banyans

4.1 Tree-Shaped Connections

In a banyan the data path established to connect the resource modules of any subsystem always forms a tree rooted at some apex. By definition, there is a unique path from each base to each apex. A subsystem bus is formed by selecting an apex and then closing all switches along the path from each desired base to the selected apex. Since each path is unique, the resulting data path forms a tree rooted at the selected apex (Th. 1.1.1). Algorithms for locating eligible apexes and establishing the connections will be presented in Section 4.4.

Tree-shaped connections are significant because they lend themelves well to the inclusion of priority hardware and because they can afford low propagation delay with limited fanout. A method for building priority hardware into a banyan partitioning network will be described next. Propagation delay and fanout requirements of certain types of banyans will be discussed in subsequent sections.

4.2 Priority Hardware in Tree-Shaped Data Paths

The need for priority hardware in a partitioning network was described in Section 2.3. The tree-shaped nature of subsystem busses in a banyan network allows suitable priority hardware to be built into the network using the basic approach outlined in this section. Implementation will not be discussed in detail since a number of variations are possible. Although designed for use in banyan partitioning networks, the technique described here is applicable to any tree-shaped data path, and is somewhat similar to that described by Foster (68). It allows priority levels to be associated with requests for bus control by resource modules and, in the event of simultaneous requests, grants control to the module with the highest priority request. Various tie breaking schemes are possible.

In the proposed priority scheme, each resource module desiring control of its tree-shaped subsystem bus transmits a "bus request signal" apexward toward the tree's root along a set of "bus request lines." A bus request signal is an encoded number representing the priority level of the corresponding request for bus control. Modules not desiring bus control transmit bus request signals at priority level zero. Priority hardware in the network compares these signals and sends "request denial signals" to all resource modules except the one to which it grants bus control.

Figure 4.2-1 illustrates how the priority scheme would function. Since the priority hardware for each subsystem functions independently, only one subsystem bus is illustrated. A, B, C, and D are the bases, or resource module ports, included in this subsystem. X and Y are intermediate vertices used in the corresponding tree-shaped connection,

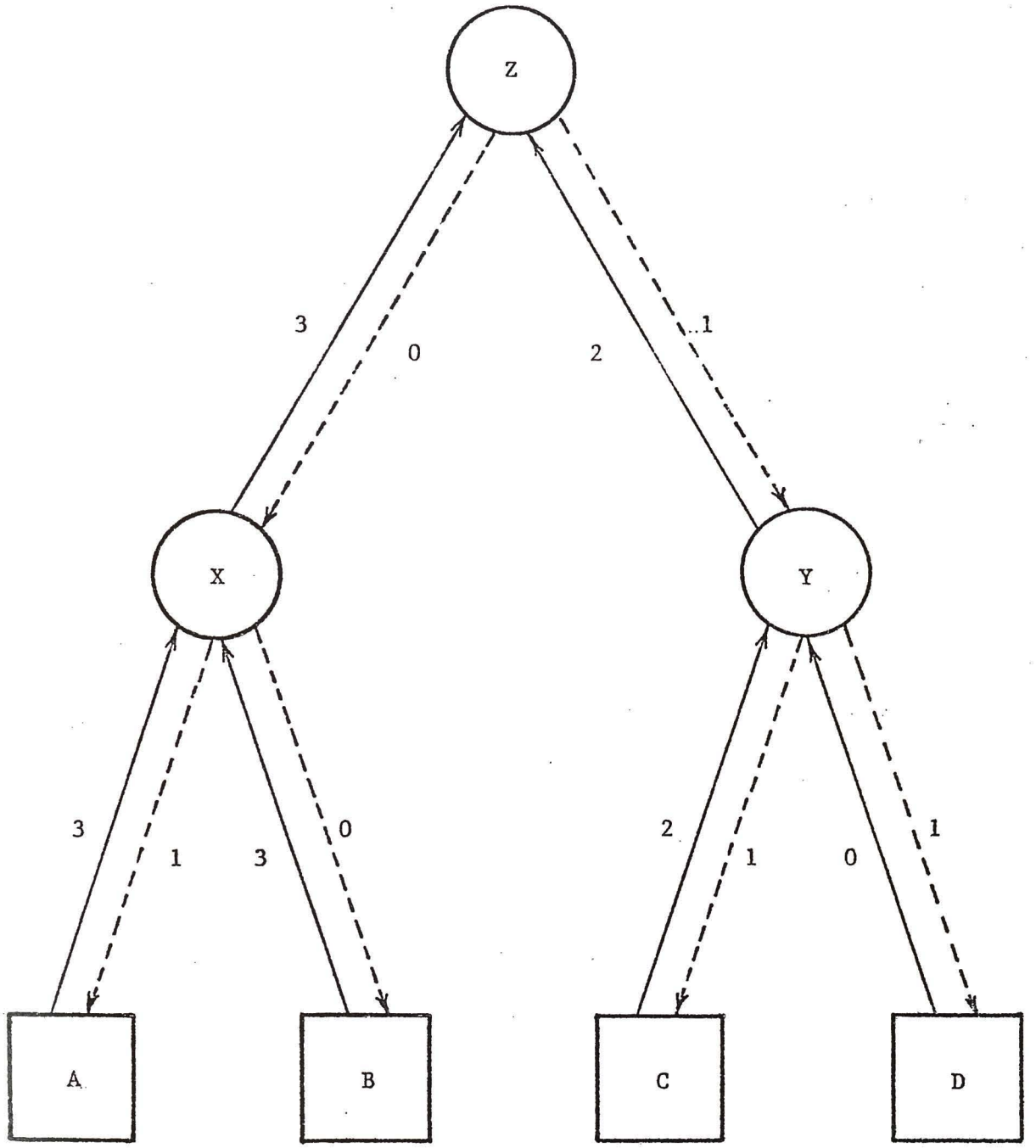


Figure 4.2-1. Example of Priority Vie

and Z is the apex at its root. Solid lines in the diagram represent bus request lines and dash lines represent "request denial lines," which are used to convey request denial signals. Numbers beside these lines indicate the signal values they carry in the example.

To simplify the explanation, we assume that bus request and request denial lines are physically distinct from those used to convey data. These lines are switched by switching devices in the banyan just as are the data lines so that their connection patterns are correspondingly tree-shaped. They differ from the data lines, however, in that they transfer signals in only one direction and interface with priority hardware at each intermediate and apex vertex. Bus request signals originate at bases and propagate only in an apexwardly direction. Request denial signals are generated by the priority hardware at apex and intermediate vertices and propagate baseward.

Priority hardware at each intermediate vertex, such as X or Y, compares its incoming bus request signals and forwards the maximum on to the vertex above. It also generates a request denial signal, represented by a logical "1", on all but one of the request denial lines below it. The request denial line over which it sends no denial signal, logical "0", corresponds to the incoming bus request signal with highest priority. Ties may be broken in various ways. In the example shown, the right-most branch with highest request priority is selected. Other tie-breaking schemes are possible, however. For example, a fairer but more complex scheme might be to select the branch whose past requests have been least recently granted.

When an intermediate receives a request denial signal from the vertex above, however, it transmits request denial signals to all

vertices below it in the connection regardless of their request priorities.

An apex, such as Z, functions exactly the same as an intermediate except that it can neither receive request denials from nor transmit bus requests to a vertex above.

In the example of Figure 4.2-1, bases A, B, and C are making bus requests at priority levels 3, 3, and 2, respectively. Base D does not desire bus control so, consequently, produces a request signal at level zero. Intermediates X and Y compare their incoming request signals and forward the maximums on to apex Z. The apex compares these requests, sends a "request granted" signal (logical "0") to the branch with highest request priority (X), and sends request denial signals (logical "1") to the other branch (Y). Vertex Y transmits denial signals to both C and D because it receives a denial from Z above. Vertex X may grant the request of either A or B since they have tied for highest priority. In this case, the tie is broken by granting B's request.

In explaining this priority scheme, we have assumed the existence of physically separate lines for data, bus request signals, and request denial signals. This allows the priority vie to be performed in parallel by combinational logic with a worst case propagation delay approximately proportional to the longest path length from a base to the apex in the connection. Further, it allows the priority vie for one message transfer to overlap data transmission from the previous one. This could be desirable to achieve high transfer rates for short messages.

For applications in which lower rates are acceptable, however, serial implementations using fewer lines may be more economical. A number of variations are possible. For example, the same lines might

be used for the priority vie as for data transfers. During a priority vie, the bus request and request denial lines would function as described. Once the new bus master has been selected, the network, or at least that part of it used by a given subsystem, would change its operating mode and would allow these lines to function as a bidirectional data path during the message transfer.

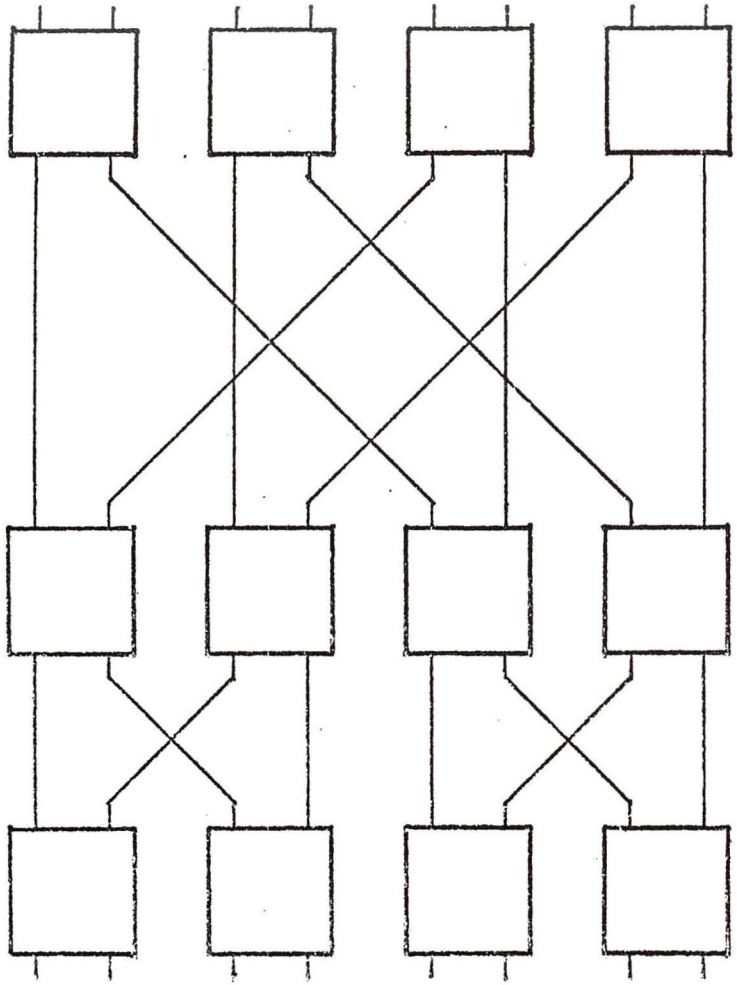
Another possibility is to perform the priority vie sequentially in two phases. First the bus request signals would be transmitted apexward as described. Actual transmission of these signals could be either serial or parallel. During this phase, a small register is set in each intermediate and apex vertex to record the branch from which the highest priority request was received. These registers then contain sufficient information to control routing of the request denial signals. For the second phase, the operating mode of the priority hardware is changed and request denial signals are propagated baseward using one of the lines previously used for the bus request signals.

4.3 Synthesizing Large Banyans from Smaller Ones

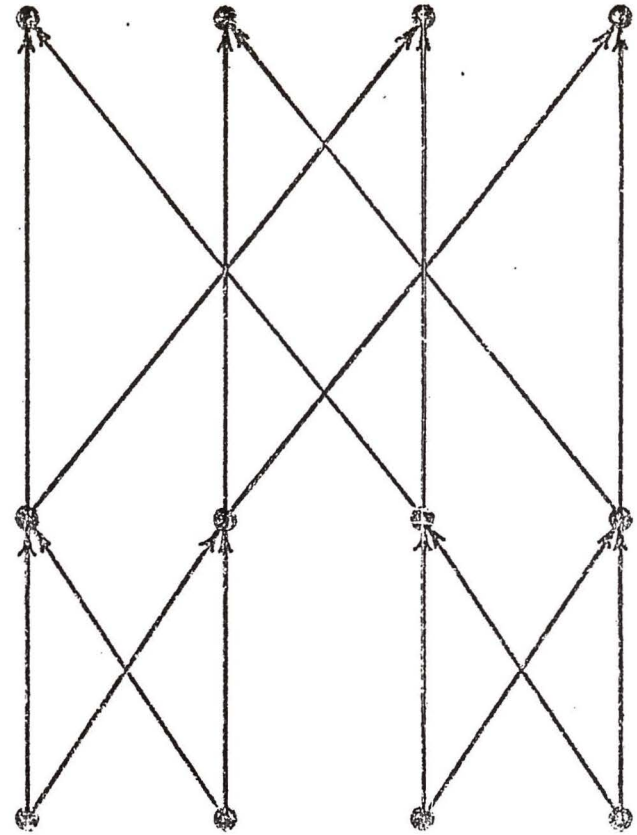
Large banyan networks can be synthesized recursively from smaller ones. Suppose that one has available a number of small banyan networks, perhaps supplied by a manufacturer as standard components, and one wishes to synthesize a larger network. This can be done as illustrated in Figure 4.3-1a by connecting the apexes of some banyans to the bases of others.

The interconnections of these banyans can be represented by a graph, as illustrated in Figure 4.3-1b. In this graph, called an interconnection graph, each vertex represents a banyan network. An arc from any vertex $V1$ to another vertex $V2$ means that one apex of banyan $V1$ is directly connected to one base of banyan $V2$. We assume that if there are any arcs incident into a vertex, then the corresponding banyan has exactly one base for each incident arc. Similarly, the number of apexes equals the number of arcs incident out from the corresponding vertex unless there are none. When there are no arcs incident into a vertex, the bases of the corresponding banyan become the bases of the synthesized network. Similarly, the apexes of the synthesized network are those of the component banyans with no arcs incident out.

Theorem 1.2.3 states that when banyan networks, called the component banyans, are interconnected as described, the resulting synthesized network will be a banyan if and only if the corresponding interconnection graph is a banyan. This implies that once one or more banyan structures are known, these structures can be recursively expanded to arbitrarily large sizes. Using this principle, one might construct a large banyan network by systematically interconnecting a number of smaller component



a) Synthesized Network



b) Interconnection Graph

Figure 4.3-1. Banyan Synthesis

banyan networks in a pattern which is itself a banyan. Suitable component banyans could be manufactured as standardized modules.

The SW structure, discussed later, is characterized by applying this kind of recursive expansion to a crossbar, which is one of the simplest banyan structures.

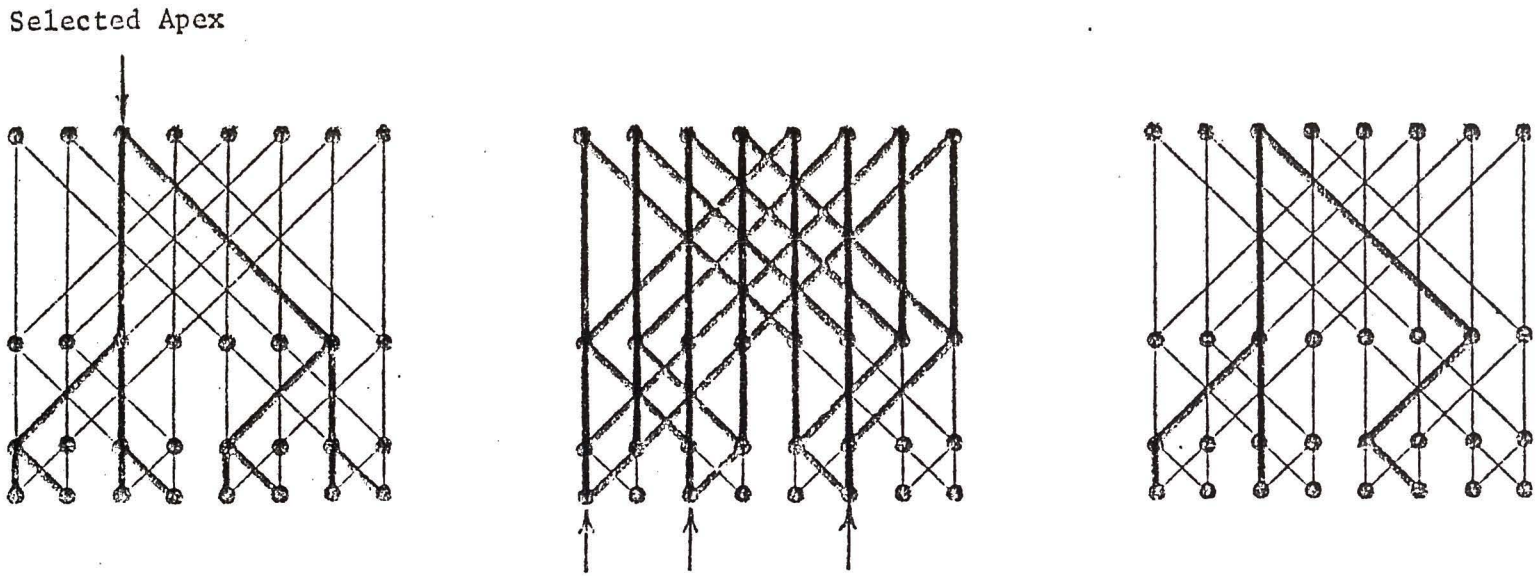
4.4 Control of Connections

The tree-shaped subsystem connections in a banyan network can be established very rapidly and in a potentially fault-tolerant manner using distributed control hardware within the network. Control is accomplished by means of a set-up algorithm, which establishes a subsystem connection, and a search algorithm, which locates eligible apexes. In setting up the first subsystem, any apex may be used as the root of its tree-shaped subsystem bus. Prior to setting up each additional subsystem, however, a search algorithm must be employed to select an apex such that the new connection will not interfere with those already existing.

A two-step set-up algorithm is illustrated in Figure 4.4-1 and is justified theoretically in Theorem 1.3.1. Set-up is facilitated by a single control line provided in each link of the network. First, a "one" signal is broadcast baseward from the selected apex over the control line, as illustrated in Figure 4.4-1a. The signal fans baseward at each vertex so that the "one" propagates to all bases. This signal sets a flip-flop in each intermediate and apex through which it passes.

In the second step, "ones" are broadcast apexward from each base in the desired subsystem, as illustrated in Figure 4.4-1b. In this step, the signal is OR'ed apexward at each vertex. As illustrated in Figure 4.4-1c, the desired connection is made by closing every switch that receives this signal from below and has a set flip-flop in the adjacent vertex above. These are the links through which control signals propagated in both steps one and two.

As described, this set-up algorithm would require two steps but only one control line in each link. Unlike the data lines, this control



a) Step 1

b) Step 2

c) Final Connection

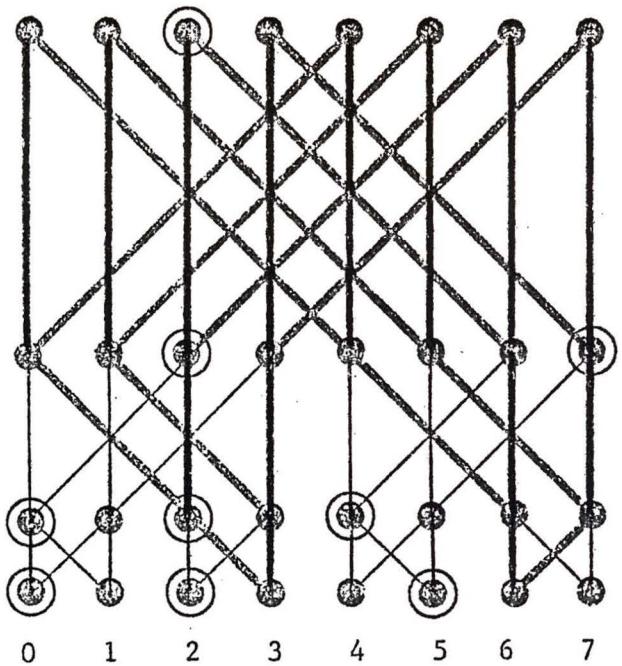
Figure 4.4-1. Set-Up Algorithm

line is always connected between vertices and does not require a bi-directional switch for each edge of the graph. Switching for the control line occurs at the vertices, where the signal is either OR'ed up or OR'ed down. With the use of two control lines rather than one, the two steps of this algorithm could be combined into one, eliminating the need for the flip-flop at each intermediate and apex.

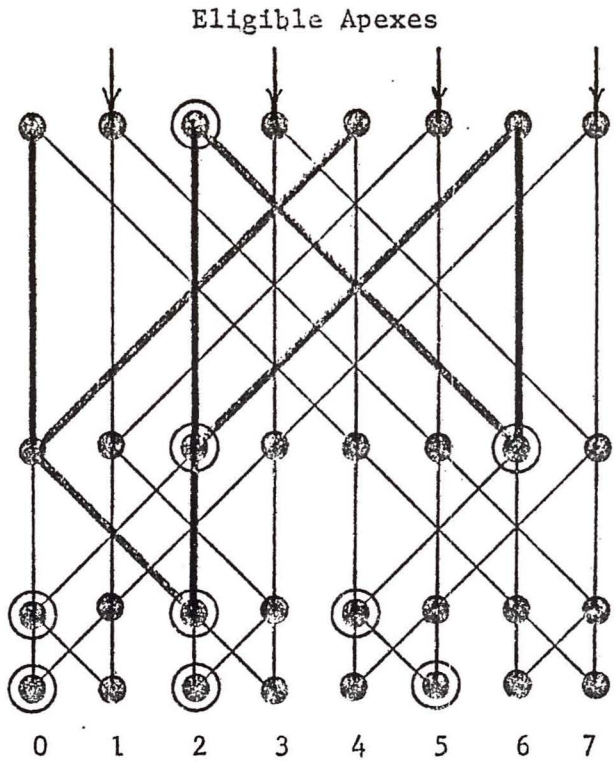
A two-step search algorithm is illustrated in Figure 4.4-2 and is justified theoretically in Theorem 1.3.2. The purpose of the search algorithm is to locate those apexes which are suitable for connecting a given subsystem (i.e., set of bases) without interfering with any existing connections. Two subsystem connections can interfere if and only if they have some vertex in common.

In the example illustrated, the circled vertices represent those already in use, and bases 3 and 6 are to be connected as a new subsystem. As shown in Figure 4.4-2a, control signals are first broadcast apexward simultaneously from all bases in the desired subsystem and are then OR'ed upward using the same control line used in set-up. During this step, a flip-flop is set in every intermediate and apex which receives this control signal and is already in use.

In the second step, illustrated in Figure 4.4-2b, the control signals from the bases are turned off, and each vertex with a set flip-flop broadcasts a "one", which is OR'ed apexward on the same line used in step one. All apexes not receiving a "one" during this step are eligible. Final selection could then be performed by a priority circuit attached to the apexes.



a) Step 1



b) Step 2

Figure 4.4-2. Search Algorithm

Steps one and two of this algorithm, like those of the set-up algorithm, could be combined using a second control line. With four control lines, search and set-up could all be combined in one step.

Another possible variation is to provide no dedicated control lines at all and use instead the same lines that are used for data and priority signals within a subsystem. Since these lines must be treated differently by the network, the entire network could be switched between two modes as needed to facilitate either reconfiguration or intrasubsystem communication. This implementation of the control algorithms could reduce the required pin counts in network hardware, but, unlike other implementations, would require that communication be temporarily suspended in all existing subsystems whenever new ones are being set up.

The most desirable control algorithm implementation would depend on the cost-performance tradeoffs of a particular application.

The control algorithms described are inherently fault-tolerant when faulty vertices in the network can be made to appear like those already in use to the search algorithm. New connections would then be routed around faulty portions of the network just as they are routed around those portions already in use. Using the search algorithm described, however, a portion of the control circuitry in faulty cells would still need to function properly in order to make them appear like those in use. A slower search algorithm that avoids this problem has been described by Lipovski (70). Alternatively, a software search algorithm could replace the faster hardware algorithm in the event of hardware failure.

4.5 Parallel and Multiplexed Networks

Search and set-up algorithms may be repeated until all desired subsystems have been connected or until no more eligible apexes can be found. Practical banyan networks allow numerous combinations of subsystems to be configured in this way but are not necessarily capable of realizing all possible partitionings of system resources. As was discussed in Section 2.3, this degree of partitioning flexibility may suffice for certain kinds of applications. When necessary, a banyan network can always configure conflicting isolated subsystems at different times.

When greater partitioning flexibility is required, there are two solutions which potentially will allow configuration of all possible partitionings. First, several banyans can be connected in parallel. The parallel networks would function independently but their bases would be connected to the same set of resource modules. As many subsystems as possible would be connected using the first network. Those left over would be connected in as many additional networks as required.

The other solution is to multiplex a single network so that it periodically rearranges itself to connect first one set of subsystems, then another, and so on, so that each subsystem has some time slot during which it can communicate. A partitioning network, as considered here, acts as a rearrangeable set of time-shared busses. A resource module attached to the network must request and receive control of its bus before transmitting data, and must be prepared to wait whenever the bus is not immediately available. Normally, the bus would be unavailable only when used by other resources in the same subsystem;

but should it ever become temporarily unavailable for other reasons, the only effect would be to delay data transmission within the subsystem. This situation makes multiplexing possible with little or no modification of the resource modules. The system need only be designed so that any resource not currently connected by the network would "see" it as a busy bus.

Multiplexing requires that a small amount of memory be associated with each switch in the network to store the state of the switch during each time slot. With LSI, this could be done at reasonable cost by associating a small register with each switch and synchronizing all state changes from a central clock.

The techniques of parallel networks and multiplexing may be mixed to balance cost and performance. Whether a network structure is space shared with parallel hardware or time shared with multiplexing, the parallel networks and/or time slots share many properties and are called layers. The number of layers required depends on a number of factors and will be discussed later.

There also is a partial solution which could provide some increase in partitioning flexibility. Sometimes it might be possible to connect additional subsystems in a single layer by rearranging the connections of existing subsystems. This, however, would require more complex algorithms, would interrupt processing in existing subsystems during reconfiguration, and would provide only a limited increase in partitioning flexibility. Consequently, it is believed that parallel networks or multiplexing would be of much more practical value.

SECTION 5

L-LEVEL BANYANS

An L-level banyan, defined formally in Definition 2.1, is a banyan whose vertices are arranged in levels so that switches, or arcs of the graph, exist only between vertices in adjacent levels. For example, the graphs in Figures 4-1b, 4.4-1, and 4.4-2 are L-level banyans, but 4-1a is not. There are actually L+1 levels of vertices in an L-level banyan. They are, by convention, numbered apexward from 0 through L so that all bases are in level 0 and all apexes are in level L. When we say that a banyan has L levels, however, it will be understood that it is an L-level banyan rather than an (L-1)-level banyan.

The class of L-level banyans is a proper subset of the general banyan networks discussed in Section 4, but is still broad enough to include most practical designs. As will be explained in this section, L-level banyans have additional useful properties, which make them attractive as partitioning networks.

Any path from a base to an apex in an L-level banyan has exactly L arcs; thus, the propagation delay of data through the network cannot exceed that of $2\times L$ switches, since in the worst case, data must travel from base to apex to base.

Base and apex "distance" functions can be associated with L-level banyans and will be discussed in Section 5.1. Theoretical results concerning these functions can be used to improve the performance of an L-level banyan network.

A class of L-level banyans called "uniform" banyans will be discussed in Section 5.2. Special cases of uniform banyans called "regular" and "rectangular" banyans also will be discussed. It will be shown that measures of the size and cost of a uniform banyan can be expressed as functions of certain parameters called "fanout" and "spread". The orderly structure of the networks discussed in Section 5.2 makes them likely candidates for modular construction, and it is expected that most practical designs would fall into these categories.

5.1 Base and Apex Distance

The dyadic operators $\boxtriangle$ and $\boxtriangledown$, called base distance and apex distance, respectively, are defined in Definition 2.1.2 for any L-level banyan. The base distance $B1 \boxtriangle B2$ specifies the minimum number of levels up into the banyan a connection must extend to connect two bases $B1$ and $B2$. Similarly, the apex distance $A1 \boxtriangledown A2$ specifies the minimum number of levels down from the top of the banyan a connection must extend to connect two apexes $A1$ and $A2$.

Figure 5.1-1 illustrates the concepts of base and apex distance. The darkened paths represent minimal connections. The connection of apexes is presented only as a conceptual aid in explaining apex distance and would not actually occur in a banyan partitioning network.

The definitions of base and apex distance are extended to sets of bases and apexes, respectively, in the same way that point distances often are extended to sets of points. That is, the base distance between any two sets of bases $SB1$ and $SB2$ is defined to be the minimum of all distances $B1 \boxtriangle B2$ such that $B1 \in SB1$ and $B2 \in SB2$. The analogous extention applies to apex distance.

Base and apex distance functions are used in Theorem 2.1.7 to characterize a way of avoiding conflicts in connections established within an L-level banyan. Theorem 2.1.7 tells us that if

$$L < (SB1 \boxtriangle SB2)+(A1 \boxtriangledown A2),$$

then subsystems $SB1$ and $SB2$ can be connected without conflict in the same layer using tree-shaped connections rooted at apexes $A1$ and $A2$, respectively.

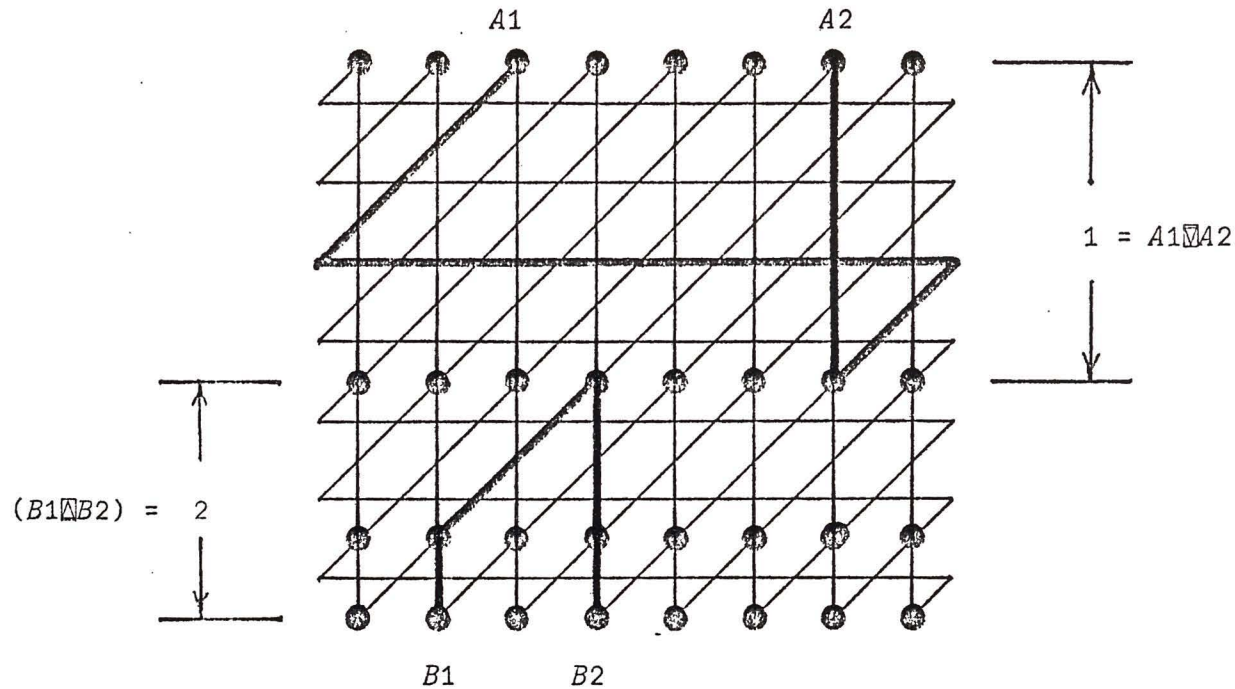


Figure 5.1-1. Base and Apex Distances in an L-Level Banyan

There are two potentially useful interpretations of Theorem 2.1.7 which suggest ways of enhancing the performance of an L-level banyan partitioning network. First, subsystems close to each other place more stringent requirements on the separation of apexes used than do widely separated subsystems, suggesting that closely spaced subsystems are less likely to be connected in the same layer. Thus, if it is known at design time which resources of a system are most likely to be connected, one might improve performance by gerrymandering the assignment of resources to bases so that bases most likely to be connected tend to be closest. For example, each processor module might be placed close to some memory module port, and multiple ports to the same memory module might be widely separated. An operating system also could take advantage of this result by allocating closely spaced resource modules to a subsystem whenever possible. The amount of improvement thus obtainable is not estimated here since this would be highly problem dependent, but one can easily contrive extreme examples in which more than one layer would seldom or never be needed.

The second interpretation concerns the selection of apexes. The search procedure described earlier locates all apexes eligible for connecting a new subsystem in a partially occupied layer, but does not determine which of the eligible apexes is the best choice. Theorem 2.1.7 now suggests a plausible selection criterion. According to the theorem, any new subsystem can be connected if we can find some apex sufficiently distant from those already in use. Thus, apexes most distant from those in use are the most valuable in the sense that they are likely to be eligible for connecting the greatest variety of subsystems.

More subsystems might then be connected in a layer by selecting each new eligible apex so as to leave as many "valuable" apexes as possible for subsequent connections. This criterion is ambiguous in some cases, but, nevertheless, is the conceptual basis for a priority rule found to improve performance in network simulations discussed in Section 8.

5.2 Fanout and Spread

A class of L-level banyans called uniform banyans is defined in Definition 2.2.1. Within each level of a uniform banyan, all vertices are alike in that each has the same number of arcs incident into it and the same number of arcs incident out from it. The arcs incident out from the vertices of a uniform banyan are characterized by an L-component vector $\underline{F}$, called the fanout vector. Similarly, the arcs incident in are characterized by an L-component spread vector $\underline{S}$. When $\underline{S} = \underline{F}$, the banyan has the same number of vertices in each level (Corollary 2.2.1b) and is called rectangular (Definition 2.2.2).

A regular banyan (Definition 2.2.3) is a special case of a uniform banyan, in which all vertices throughout the network are alike except, of course, for the fact that bases have no arcs incident into them and apexes have no arcs incident out. All components of a regular banyan's fanout vector, thus, are equal and can be characterized by a single scalar parameter F, called fanout. Similarly, all components of its spread vector are equal and are characterized by a spread parameter S.

Regular banyans probably would be the most economical to fabricate, because they can be built from a number of identical cells, each containing the circuitry associated with a vertex and the arcs incident into it. The fanout and fanin requirements of these cells are determined by F and S. Such modular construction could be used for other uniform banyans as well except that different kinds of cells might be needed for different levels.

Theorems 2.2.1 and 2.3.1 and their corollaries show how the numbers of arcs and vertices in various kinds of uniform banyans are related to fanout and spread. The expressions derived are summarized in Table 5.2-

1. The total number of arcs in a banyan graph can be used as a measure of network cost since these arcs represent the bidirectional switching devices in the corresponding network.

It is shown in Theorem 2.4.2 that, for a given number of bases, the "cost" (number of arcs) per base of a regular rectangular banyan is minimized with respect to fanout when $F = 3$. Further, the cost of such a network is the same when $F = 4$ as it is when $F = 2$. Similarly, it is shown that a crude cost-performance measure, obtained by multiplying this cost function by a measure of maximum data propagation delay, is optimized when $F = 7$ and is very near optimal when $F = 8$. The cost and performance aspects of banyan partitioning networks will be discussed in greater detail in Section 9.

TABLE 5.2-1. Size and Cost Functions for Uniform Banyans

Type of Banyan	Vertices in Level I	Bases	Apexes	Arcs (Cost Measure)
Uniform	$\times/(I\uparrow\underline{S}),I\downarrow\underline{F}$ (Th. 2.2.1)	$\times/\underline{F}$ (Cor. 2.2.1a)	$\times/\underline{S}$ (Cor. 2.2.1a)	$[I=1\sim L]\underline{F}[I]\times\times/(I\uparrow\underline{S}),I\downarrow\underline{F}$ (Th. 2.3.1)
Rectangular	$\times/\underline{F}$ (Cor. 2.2.1a & 2.2.1b)	$\times/\underline{F}$ (Cor. 2.2.1a)	$\times/\underline{F}$ (Cor. 2.2.1a & 2.2.1b)	$(\times/\underline{F})\times/\underline{F}$ (Cor. 2.3.1a)
Regular and Rectangular	$F*L$ (Cor. 2.2.1b & 2.2.1c)	$F*L$ (Cor. 2.2.1c)	$F*L$ (Cor. 2.2.1c)	$(F*L)\times L\times F$ (Cor. 2.3.1b)
Regular	$(S*I)\times F*L-I$ (Cor. 2.2.1c)	$F*L$ (Cor. 2.2.1c)	$S*L$ (Cor. 2.2.1c)	$F\times[I=1\sim L](S*I)\times F*L-I$ (Cor. 2.3.1c)
Regular and Nonrectangular	$(S*I)\times F*L-I$ (Cor. 2.2.1c)	$F*L$ (Cor. 2.2.1c)	$S*L$ (Cor. 2.2.1c)	$(F*L)\times S\times(((F*L)\times S)-F)\div S-F$ (Cor. 2.3.1d)

SECTION 6

SW BANYANS

SW banyans (Definition 3.1.2) are a particularly interesting class of L-level banyans which can be synthesized recursively from crossbars (Definition 3.1.1) using the synthesis principle discussed in Section 4.3. SW banyans are probably the most attractive banyans for partitioning networks because they are the best understood theoretically, because they lend themselves exceptionally well to modular construction and, because they are a broad class of networks which can be varied in a number of ways to meet the needs of different applications. Additionally, the analysis of SW banyans might have much broader implications, because a number of connecting networks originally proposed for other applications are actually special cases of SW banyans.

6.1 Previous Special Cases

SW banyans are actually a generalization of a number of network structures considered previously for a variety of applications. The term "SW structure" was originally used by Lipovski (69, 70), who first proposed them for partioning applications in a large associative processor. The structures he defined are equivalent to regular SW banyans, and the possibility of uniform (but nonregular) SW banyans was implied (Lipovski, 69). Structures graphically equivalent to regular rectangular banyans with fanout and spread equal to 2 had been proposed earlier by Batcher (68) for use as "bitonic sorters."

More recently, networks graphically equivalent to rectangular SW banyans were proposed by Lawrie (73, 75) for memory-processor communication. Lawrie defined these networks using "omega-base" representations of integers and analyzed them for his application using number theory. Lawrie also noted that interconnections between stages of an "omega network" (i.e., between levels of a rectangular banyan) are equivalent to the "perfect shuffle" connection discussed by Pease (68) and by Stone (71). Additional material on the control and applications of networks of this type was published by Lang and Stone (76).

A variety of permutation networks also have been proposed which contain special cases of SW banyans as major subgraphs. These networks are intended to permute a set of input lines onto an equal number of output lines in any desired fashion and were studied largely for telephone switching applications. Clos networks, proposed by Clos (53) and discussed further by Benes (62), contain three stages of crossbars interconnected symmetrically. If the last stage (or alternately the first, since the network is symmetrical) were removed from an n by m

by r Clos network, then the remainder would be a uniform 2-level SW banyan with fanout vector n r and spread vector m r. Benes (64a, 64b, 65) analyzed a similar, but more general, class of permutation networks containing an odd number (not less than 3) of stages of crossbars interconnected symmetrically. These Benes networks, like Clos networks, are symmetrical about the center stage. If one were to remove all stages from one side of a Benes network, the remainder would be a rectangular SW network (or, equivalently, an omega network as noted by Lawrie (73)). It was subsequently shown by Goldstein and Liebholz (67) and by Waksman (68) that certain crossbars can be removed from one side of a Benes network without destroying its ability to connect all possible permutations of input lines onto output lines. The other side of such a network (including its center stage) is identical to one side of a Benes network, and hence is a rectangular SW banyan. Another class of permutation networks, called "nested tree" networks, were proposed by Joel (68) with little supporting theory. Each stage of a "nested tree" network is built from two-by-two crossbars and apparently is a regular, rectangular SW banyan with fanout and spread equal to 2.

Such common structures as crossbars and homogeneous trees[1] are also special cases of SW banyans. Crossbars are simply 1-level SW banyans and homogeneous trees are uniform SW banyans in which each component of the fanout vector equals 1.

---

[1]The term "homogeneous tree" is used here in the sense of Iverson (62, p. 58). All leaves of a homogeneous tree lie in the same level, and within each level, all vertices have the same degree.

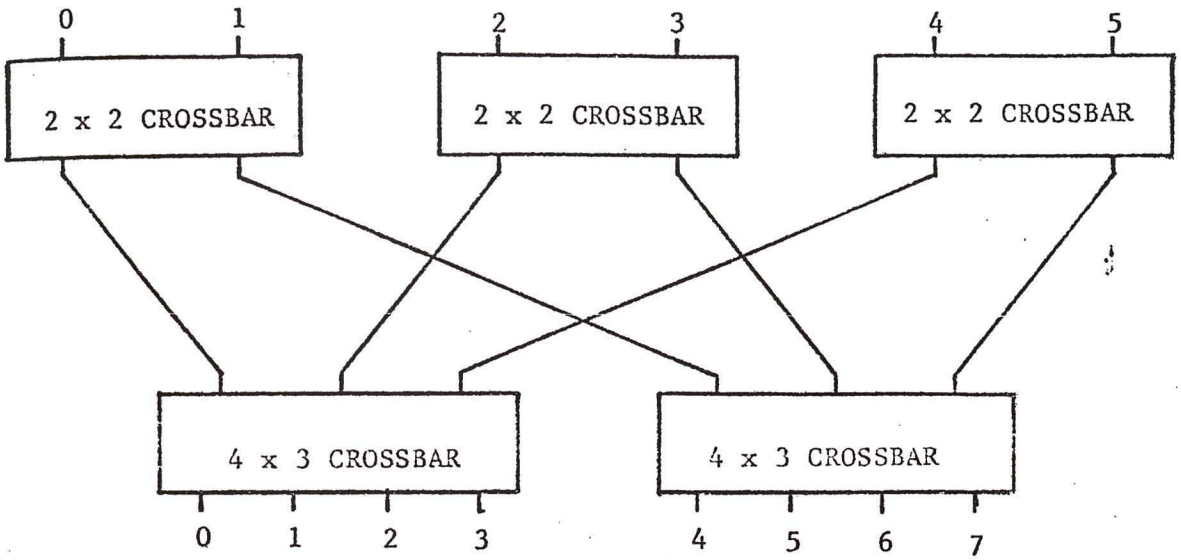
We are presently concerned with SW banyans as partitioning networks, but this diversity of applications suggests that theoretical results concerning SW banyans also could be useful in other areas.

6.2 Structure

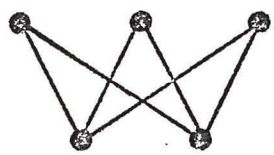
SW banyans (Definition 3.1.2) are defined recursively in terms of crossbars (Definition 3.1.1), which are simply 1-level banyans. All crossbars are SW banyans. Additionally, a synthesized banyan is an SW banyan if its interconnection graph is an SW banyan and its component banyans are all crossbars. Crossbars and synthesized SW banyans are the only SW banyans. This definition of an SW banyan is somewhat simpler and more general than that published previously by the author (Goke and Lipovski, 73). Unlike the earlier definition, it does not necessarily require an SW banyan to be uniform.

Figure 6.2-1 illustrates the synthesis of an SW banyan. Figure 6.2-1a shows the interconnection of component crossbars, 6.2-1b is the corresponding interconnection graph, and 6.2-1c is the resulting synthesized SW banyan graph. Figure 6.2-2 shows some additional examples of synthesized SW banyans and their corresponding interconnection graphs, which, of course, are also SW banyans.

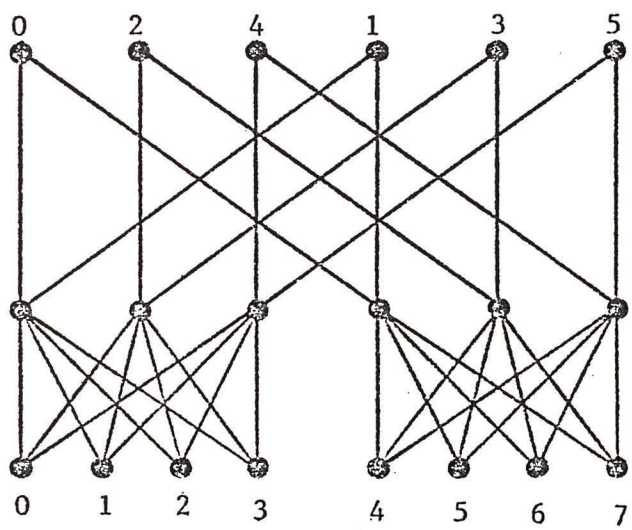
Properties of a synthesized SW banyan are related to those of its interconnection graph in a number of ways. All SW banyans are L-level banyans, and the number of levels in a synthesized SW banyan is one greater than the number of levels in its interconnection graph (Theorem 3.1.3). A uniform synthesized SW banyan with fanout vector $\underline{F}$ and spread vector $\underline{S}$ has a uniform interconnection graph with fanout vector $1\downarrow\underline{F}$ and spread vector $(-1)\downarrow\underline{S}$ (Theorem 3.1.5). Similarly, a uniform SW interconnection graph with fanout vector $\underline{F}'$ and spread vector $\underline{S}'$ can be used to synthesize a uniform SW banyan with fanout vector $B,\underline{F}'$ and spread vector $\underline{S}',A$, where B is the number of bases in each bottom-level component crossbar and A is the number of apexes in each top-level



a) Interconnection of Crossbars

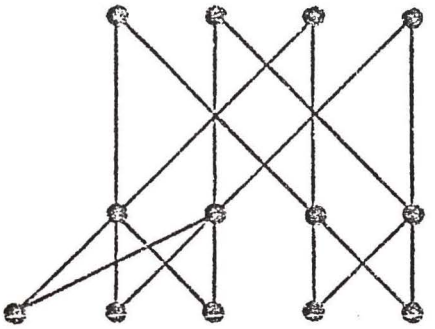


b) Interconnection Graph

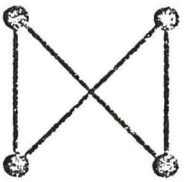


c) Synthesized SW Banyan

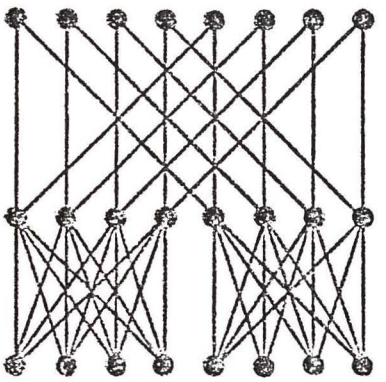
Figure 6.2-1. Synthesis of an SW Banyan



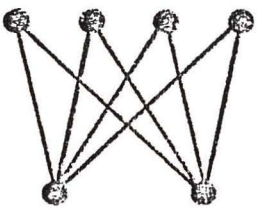
a) Nonuniform SW Banyan



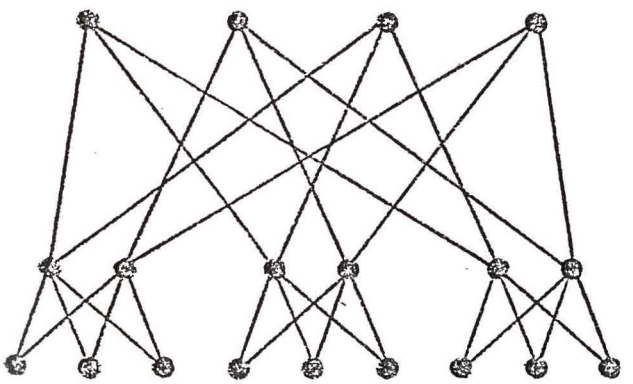
b) Interconnection Graph of a



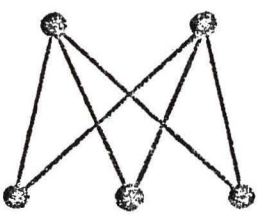
c) Rectangular SW Banyan



d) Interconnection Graph of c

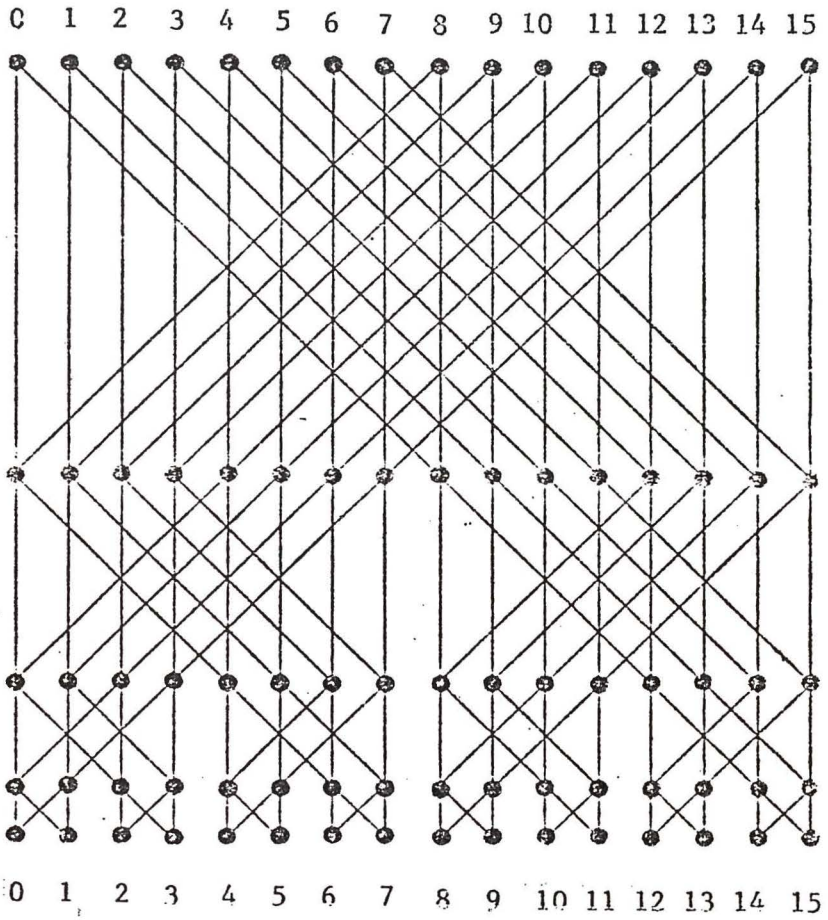


e) Regular SW Banyan

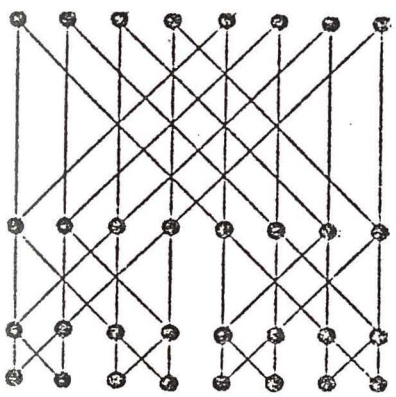


f) Interconnection Graph of e

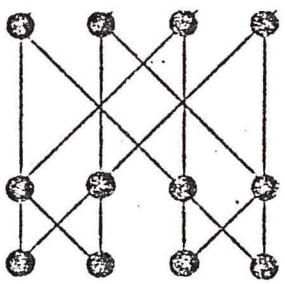
Figure 6.2-2. Examples of SW Banyans



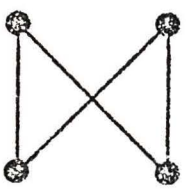
g) Regular, Rectangular SW Banyan



h) Interconnection Graph of g



i) Interconnection Graph of h



j) Interconnection Graph of i

Figure 6.2-2 (Continued)

component crossbar (Theorem 3.1.4). Also, the bases and apexes of a synthesized SW banyan can be mapped into those of its interconnection graph such that distances in the synthesized banyan are one greater than the corresponding distances in the interconnection graph (Theorems 3.1.6 and 3.1.7).

Synthesized SW banyans lend themselves especially well to modular construction. By definition, a synthesized SW banyan can be constructed by interconnecting crossbars in a pattern which is itself a simpler SW banyan. Crossbars are, thus, natural building blocks for SW banyans. A regular SW banyan has the advantage that its component crossbars are all identical. At most L different kinds of crossbar modules are needed for a uniform SW banyan with L levels. A manufacturer, thus, could mass produce a few standardized kinds of crossbar modules in sizes where crossbars are practical, and these modules could be interconnected in an SW pattern to produce larger networks.

We observe that it is also possible to synthesize large SW banyans by interconnecting smaller SW banyan modules in an SW pattern. For example, Figure 6.2-3 shows how an SW banyan equivalent to that of Figure 6.2-2g can be synthesized from eight component SW banyans interconnected in a crossbar pattern. In Figure 6.2-3, graphs of the component SW banyans are drawn in the usual manner using solid lines as arcs, and broken lines show how the component banyans are interconnected. Base and apex numbers correspond with those in Figure 6.2-2g. In this manner, SW banyan modules can be used as building blocks for larger SW banyan networks. This would be useful if, for example, available packaging technology made it desirable to use modules larger than the largest practical crossbar.

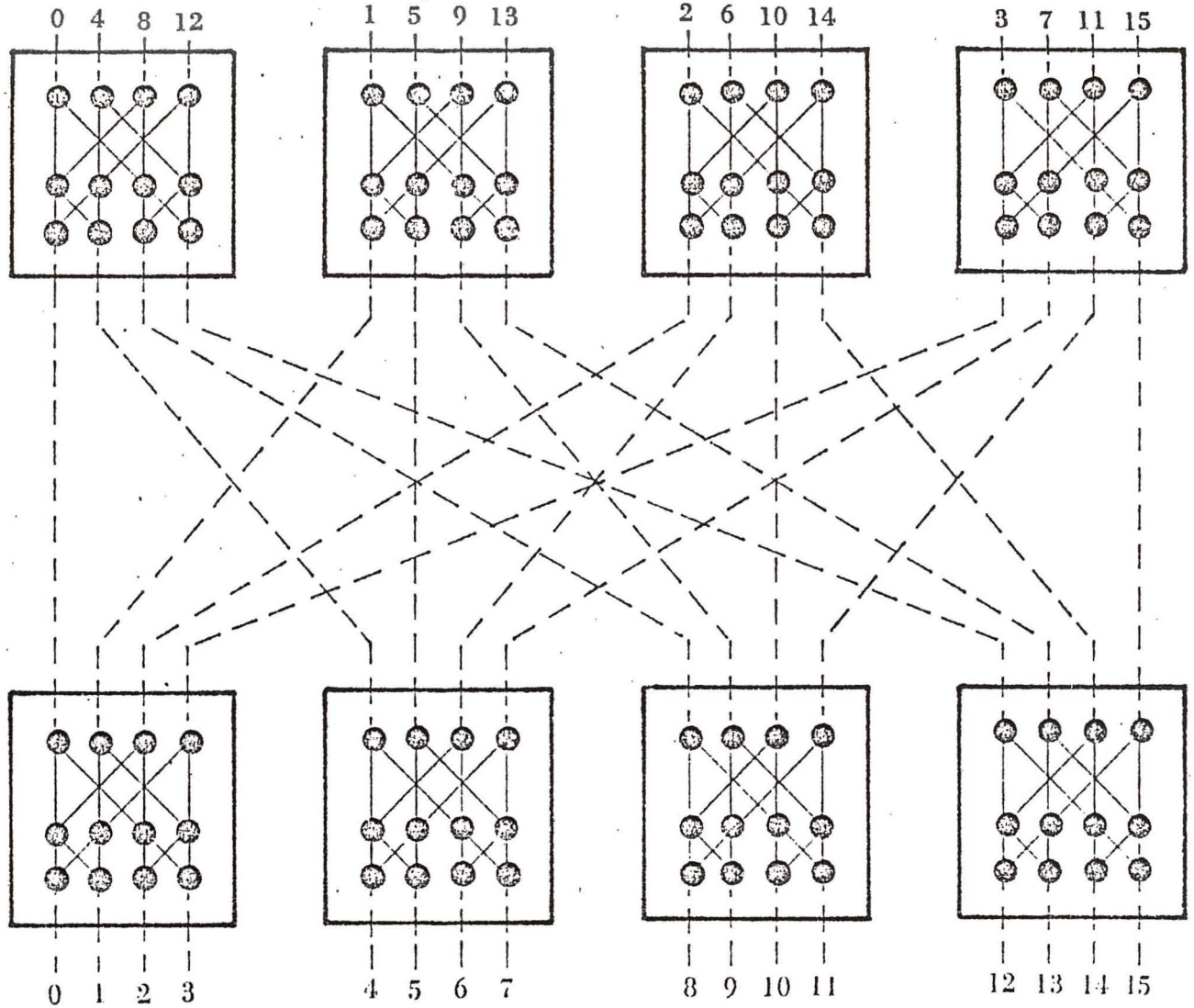


Figure 6.2-3. Synthesis of an SW Banyan From Smaller Component SW Banyans

6.3 Distance Properties

The base and apex distance functions of an SW banyan are metrics on its bases and apexes, respectively (Corollaries 3.2.3b and 3.2.4b). In an L-level SW banyan, each of these functions is characterized by $L+1$ equivalence relations with nested equivalence classes (Corollaries 3.2.3a and 3.2.4a and Theorems 3.2.5 and 3.2.6). Base distance is characterized by relations $\mathbb{\Delta}_0,\mathbb{\Delta}_1,\ldots,\mathbb{\Delta}_L$ where $B1\ \mathbb{\Delta}_I\ B2$ if and only if $(B1\mathbb{\Delta}B2) \leq I$ (Definition 3.2.1). Similarly, apex distance is characterized by relations $\mathbb{\nabla}_0,\mathbb{\nabla}_1,\ldots,\mathbb{\nabla}_L$ where $A1\ \mathbb{\nabla}_I\ A2$ if and only if $(A1\mathbb{\nabla}A2) \leq I$ (Definition 3.2.2). The equivalence classes of these relations are listed in Table 6.3-1 for the SW banyan in Figure 6.2-2g. In a uniform SW banyan with fanout vector $\underline{F}$, the relation $\mathbb{\Delta}_I$ partitions the networks bases into $\times/I\downarrow\underline{F}$ equivalence classes with $\times/I\uparrow\underline{F}$ elements each (Theorem 3.2.7). Similarly, relation $\mathbb{\nabla}_I$ partitions the network's apexes into $\times/(-I)\downarrow\underline{S}$ equivalence classes with $\times/(-I)\uparrow\underline{S}$ elements each (Theorem 3.2.8).

For reasons explained in Section 5.1, it is desirable to assign resources to bases such that resources most likely to be in the same subsystem are closest to each other. Thus, with an SW banyan, resources most likely to be in the same subsystem should be assigned to bases in the same small equivalence class. For example, suppose that eight processors and eight memory modules are to be attached to the 16 bases of the network in Figure 6.2-2g, and suppose it is known that a typical subsystem will require about as many processors as it does memories. Then chances are that more subsystems could be connected per layer if processors were attached to bases $0,2,4,\ldots,14$ and memories to bases $1,3,5,\ldots,15$ than if processors were attached to bases $0,1,\ldots,7$ and memories to bases $8,9,\ldots,15$.

TABLE 6.3-1. Base and Apex Equivalence Classes for SW Banyan in Figure 6.2-2g

Relation	Equivalence Classes of Bases
$\mathbb{A}_0$	[0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15]
$\mathbb{A}_1$	[0 , 1] [2 , 3] [4 , 5] [6 , 7] [8 , 9] [10 , 11] [12 , 13] [14 , 15]
$\mathbb{A}_2$	[0 , 1 , 2 , 3] [4 , 5 , 6 , 7] [8 , 9 , 10 , 11] [12 , 13 , 14 , 15]
$\mathbb{A}_3$	[0 , 1 , 2 , 3 , 4 , 5 , 6 , 7] [8 , 9 , 10 , 11] [12 , 13 , 14 , 15]
$\mathbb{A}_4$	[0 , 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 , 10 , 11 , 12 , 13 , 14 , 15]

Relation	Equivalence Classes of Apexes
$\mathbb{V}_0$	[0] [8] [4] [12] [2] [10] [6] [14] [1] [9] [5] [13] [3] [11] [7] [15]
$\mathbb{V}_1$	[0 , 8] [4 , 12] [2 , 10] [6 , 14] [1 , 9] [5 , 13] [3 , 11] [7 , 15]
$\mathbb{V}_2$	[0 , 8 , 4 , 12] [2 , 10 , 6 , 14] [1 , 9 , 5 , 13] [3 , 11 , 7 , 15]
$\mathbb{V}_3$	[0 , 8 , 4 , 12 , 2 , 10 , 6 , 14] [1 , 9 , 5 , 13 , 3 , 11 , 7 , 15]
$\mathbb{V}_4$	[0 , 8 , 4 , 12 , 2 , 10 , 6 , 14 , 1 , 9 , 5 , 13 , 3 , 11 , 7 , 15]

Base and apex distance functions are even more informative for SW banyans than they are for other L-level banyans. In an L-level banyan, these functions specify the minimum number of levels into the banyan that a connection must extend to connect two bases or apexes (Section 5.1). In an SW banyan, these functions also specify the maximum number of levels into the banyan that such a connection may extend before its two branches join. For example, if the distance between two bases of an SW banyan is 3, then any tree-shaped connection joining them will fork precisely at level 3 rather than just somewhere in level 3 or above. Accordingly, the necessary condition for a conflict in an L-level banyan (Theorem 2.1.7) is both a necessary and sufficient condition for a conflict in an SW banyan (Theorem 3.3.1).

SECTION 7

CC BANYANS

CC banyans (Definition 4.1.1) are a class of rectangular banyans which are potentially useful as partitioning networks. They differ from SW banyans in that multilevel CC banyans are not synthesized from smaller banyans. The distance functions of a CC banyan differ from those of an SW banyan in that bases or apexes appear to be arranged in a circle rather than in nested equivalence classes. The distance between two bases or apexes is then determined by their separation on the circle.

Relatively few examples of the CC banyan structure are known to exist in earlier networks. The "barrel switch" of the ILLIAC IV Processing Element is graphically equivalent to a 3-level, regular CC banyan with fanout and spread equal to 4. In this application, it is used to shift 64 bits an arbitrary number of places left or right (Davis, 69).

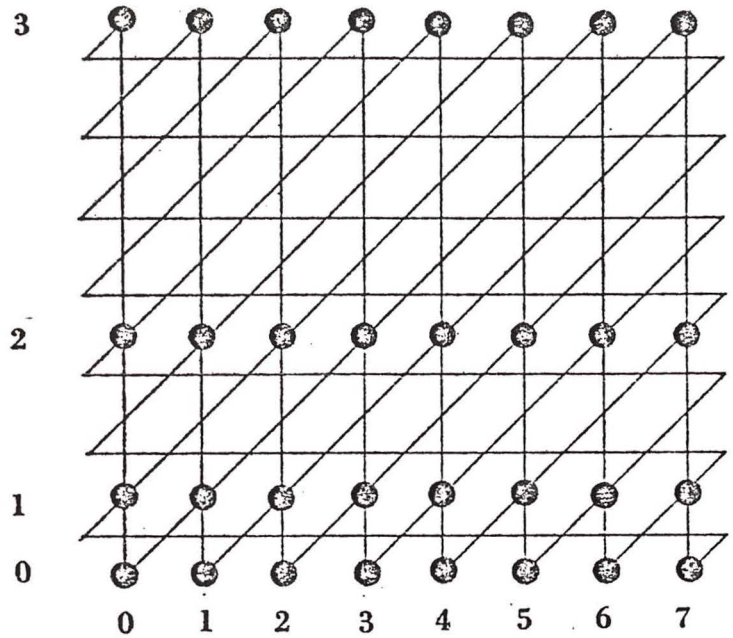
CC banyans also are related to the "line manipulator" networks proposed by Feng (74) for performing a variety of data manipulation functions. A line manipulator is not itself a banyan because it contains multiple paths from any given base to an apex, but it contains both a CC banyan and an SW banyan as partial graphs. These partial graphs are both regular, rectangular banyans with fanout and spread equal to 2.

7.1 Structure

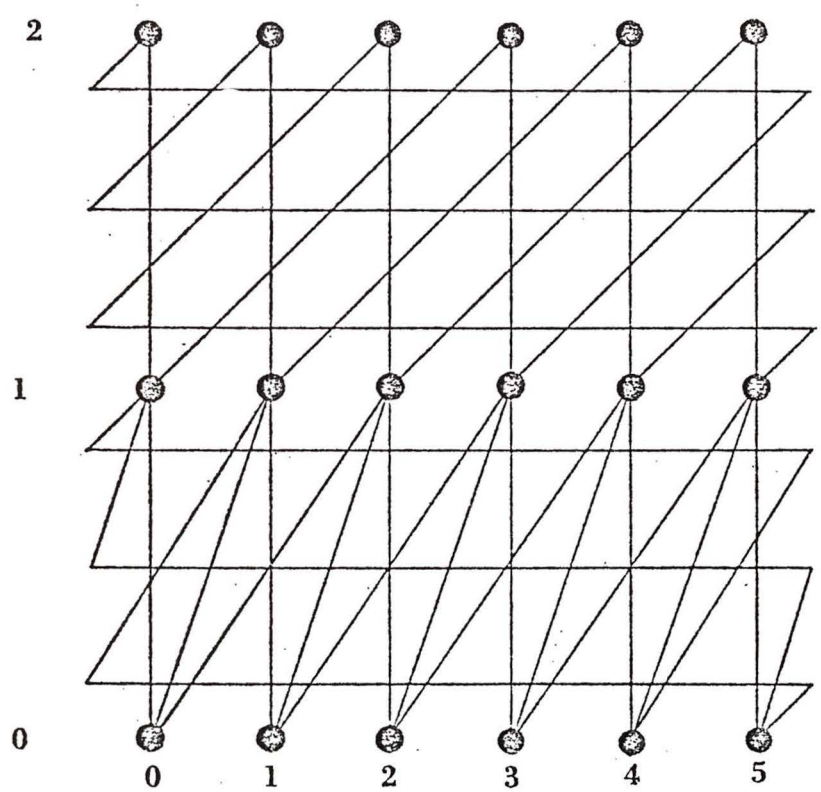
A CC banyan (Definition 4.1.1) is rectangular (Theorem 4.1.3) and, hence, has the same number of vertices in each level. For convenient identification, we can index these vertices as $\underline{V}[0\sim L;0\sim N-1]$ where $\underline{V}[I;0\sim N-1]$ are the N vertices of level I. Hence, $\underline{V}[0;0\sim N-1]$ are bases, and $\underline{V}[L;0\sim N-1]$ are apexes. Let $\underline{S}[1\sim L]$ be the fanout and spread vector of this rectangular banyan. Then from each vertex $\underline{V}[I;J]$ where $0 \leq I < L$, there is an arc to each of the vertices $\underline{V}[I1;J]$, $\underline{V}[I1;J\oplus(\times/I\uparrow\underline{S})]$, $\underline{V}[I1;J\oplus 2\times(\times/I\uparrow\underline{S})]$, ..., $\underline{V}[I1;J\oplus(\underline{S}[I+1]-1)\times(\times/I\uparrow\underline{S})]$, where $\oplus$ denotes addition modulo N. Some examples of CC banyans are shown in Figure 7.1-1. With the vertices in each level numbered from left to right as shown, each arc from a level I to level $I+1$ shifts circularly to the right $M\times(\times/I\uparrow\underline{S})$ places for some integer M where $0 \leq M < S[I1]$.

Note that if either of the CC banyans in Figure 7.1-1 were drawn on the surface of a vertical cylinder instead of in a plane, the horizontal lines would disappear, and a comparatively simple crosshatch pattern would remain. The term "CC" is an acronym for "cylindrical crosshatch" and is based upon this conceptualization of CC banyan structure.

CC banyans can be constructed in a modular fashion using a physical configuration similar to that of an ILLIAC IV "barrel switch," which has been described by Davis (69). The "barrel switch" layout is applicable directly to any regular, 3-level CC banyan with fanout and spread equal to 4, and can be extended in a straightforward manner for other CC banyans.



a) A CC Banyan with $L = 3$ and $\underline{S} = 2\ 2\ 2$.



b) A CC Banyan with $L = 2$ and $\underline{S} = 3\ 2$.

Figure 7.1-1. Examples of CC Banyans

7.2 Distance Properties

The base distance function in a CC banyan is characterized in terms of minimum circular distance. Consider the integers $0 \sim N-1$ arranged in a circle as illustrated in Figure 7.2-1. The minimum circular distance between two numbers $J1$ and $J2$ is denoted by $J1 \boxminus J2$ and is defined to be the minimum number of steps, either clockwise or counterclockwise, which separate $J1$ from $J2$ on this circle. For example, if $N \geq 4$ then $((N-1)\boxminus 1) = (1\boxminus(N-1)) = 2$. This function is a metric on the integers $0 \sim N-1$ (Theorem 4.2.3).

The distance between two bases of a CC banyan can be determined from the minimum circular distance between their indices. Any base distance, of course, must be an integer in the range $0 \sim L$. For any integer I in this range, the base distance between two bases $\underline{V}[0;J1]$ and $\underline{V}[0;J2]$ will be equal to or less than I if and only if $(J1\boxminus J2) < \times/I\uparrow\underline{S}$, where $\underline{S}$ is the fanout and spread vector of the CC banyan and where $N = \times/\underline{S}$. (Theorem 4.2.4). Thus, the base distance between $\underline{V}[0;J1]$ and $\underline{V}[0,J2]$ is simply the smallest value of $I = 0 \sim L$ such that $(J1\boxminus J2) < \times/I\uparrow\underline{S}$. Base distance is a metric on the bases of a CC banyan except possibly for degenerate CC banyans in which one or more components of $\underline{S}$ are less than 2 (Theorem 4.2.6).

It is apparent from this characterization that bases are closest in terms of base distance when their indices are closest in terms of minimum circular distance. Consequently, bases in a CC banyan can be thought of as arranged in a circle like the numbers in Figure 7.2-1. Hence, resource module ports most likely to be assigned to the same subsystem should be attached to adjacent bases, and those least likely to be connected should be attached to bases opposite each other on the circle.

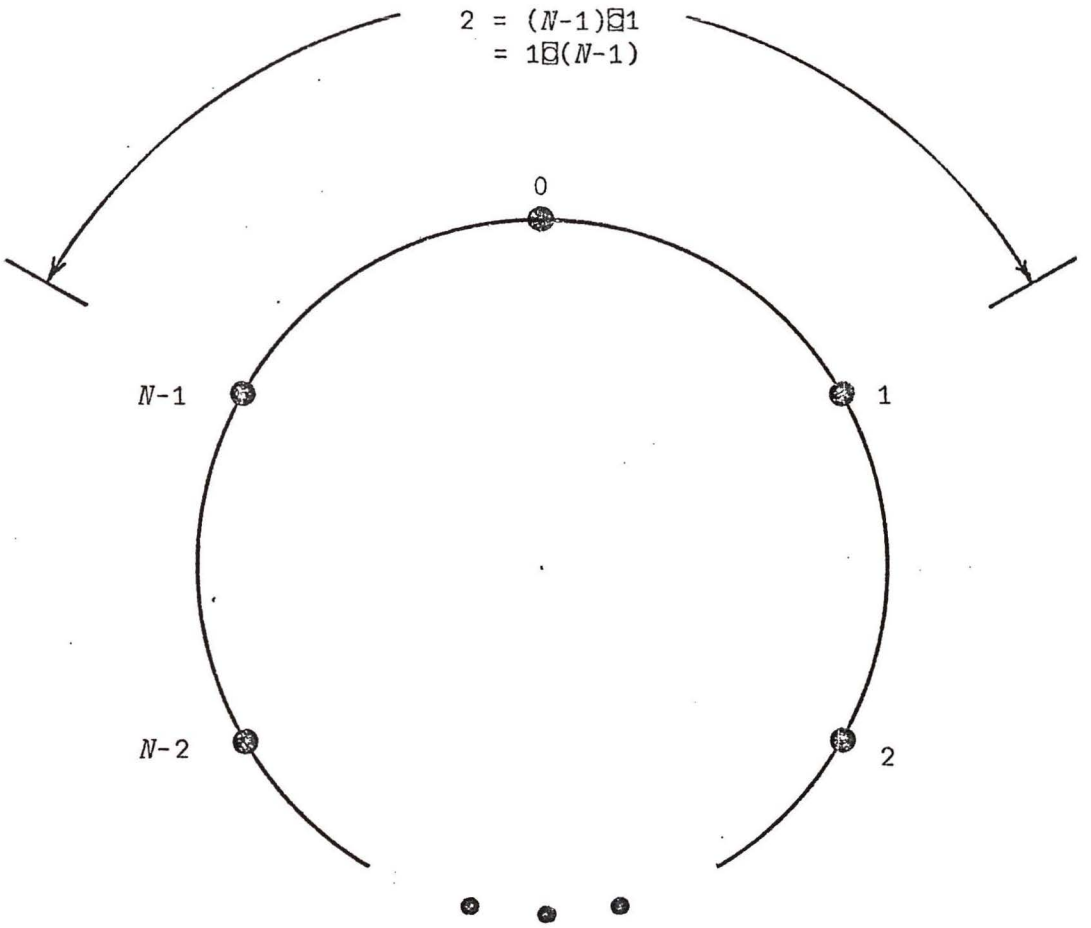


Figure 7.2-1. Conceptualization of Minimum Circular Distance

Apex distance in a CC banyan is characterized differently from base distance, but it is still useful to think of a CC banyan's apexes as being arranged in a circle. The apex distance between two apexes $\underline{V}[L,J1]$ and $\underline{V}[L,J2]$ is the smallest integer $I = 0 \sim L$ such that $0 = (+/(-I)\downarrow\underline{S})|J2-J1$; that is, such that $\times/(-I)\downarrow\underline{S}$ divides $J2-J1$ (Corollary 4.2.5a). It may be observed, however, that since $\times/(-I)\downarrow\underline{S}$ divides N, all of the following are equivalent.

$$0 = (\times/(-I)\downarrow\underline{S})|J2-J1$$

$$0 = (\times/(-I)\downarrow S)|N|J2-J1$$

$$0 = (\times/(-I)\downarrow\underline{S})|N|J1-J2$$

$$0 = (\times/(-I)\downarrow\underline{S})|J2\boxminus J1$$

If one thinks of apexes $\underline{V}[L;0]$ through $\underline{V}[L;N-1]$ arranged in a circle similar to that shown in Figure 7.2-1, then the distance between any two apexes can be determined from their separation on the circle. Starting with any apex $\underline{V}[L;J]$ and proceeding in either direction, one can find an apex I or closer to $\underline{V}[L;J]$ every $\times/(-I)\downarrow\underline{S}$ steps around the circle. For example, Figure 7.2-2 shows the apexes of the CC banyan in Figure 7.1-1a arranged in a circle. The number in parentheses next to each apex is that apex's distance from $\underline{V}[L,0]$. This pattern of numbers simply may be rotated clockwise J places to determine distances from any other apex $\underline{V}[L,J]$.

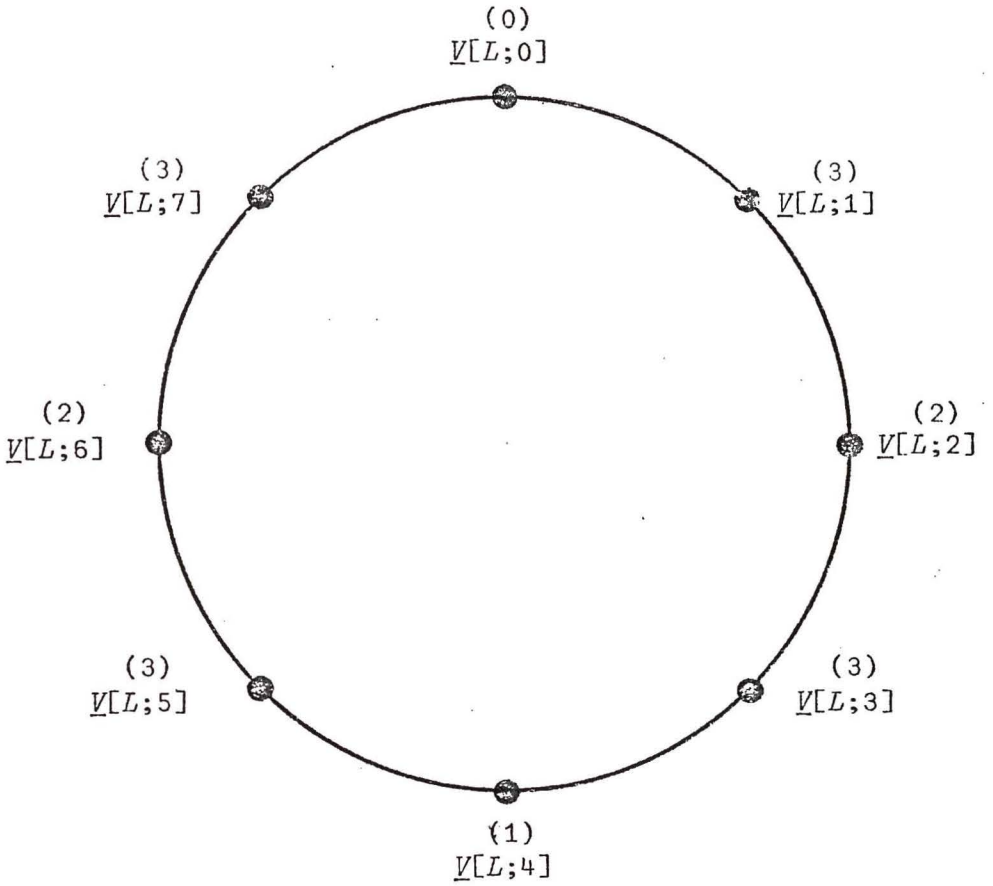


Figure 7.2-2. Apex Distances for CC Banyan in Figure 7.1-1a

SECTION 8

BANYAN NETWORK SIMULATIONS

Although theoretical analysis of banyan structures has been fruitful in many respects, it has thus far failed to yield good quantitative measures of partitioning flexibility. Consequently, a number of banyan networks were simulated on a digital computer in order to study network performance characteristics and to assess the effects of certain design options.

The tests performed were intended primarily for comparing the effects of design variations on the partitioning flexibility of a banyan network. The simulated test conditions were not based on any particular application or job mix. They were designed to exercise a network's partitioning capabilities thoroughly, but in a conceptually simple manner. As will be explained later, these test conditions tended to be contrived "worst case" conditions in several respects and probably were more severe than normal conditions in any practical application.

The simulations tested the ability of networks to connect randomly selected partitions of system resources. Statistics were gathered concerning the numbers of parallel or multiplexed layers (Section 4.5) required and concerning the number of subsystems connected in each layer. The nature of these simulations will be described in greater detail in Section 8.1.

The simulation results, tabulated in Appendix D and discussed in Section 8.2, demonstrate how performance measures for a banyan network tend to be affected by its size, fanout-spread, and structure type and by certain control options. As will be explained in Section 9, these simulation results also indicate that banyan networks could have significant cost-performance advantages over crossbar-based networks in large systems.

8.1 Nature of Simulations

Kinds of networks simulated. Both SW and CC banyan partitioning networks were simulated. All simulated networks were regular and rectangular and had fanout-spread parameters ranging from 2 to 8. The number of bases ranged from 4 to 256, but due to computer time limitations, most tests were performed using networks with at most 64 bases.

Apex selection rules. As was discussed in Section 5.1, Theorem 2.1.7 suggests that more subsystems might be connected in a given layer if apexes for new subsystems were selected as near as possible to apexes used for existing subsystems. To assess the significance of this selection criterion, two apex selection rules were used, one of which tended to select new apexes far from those in use and the other of which tended to select new apexes near to those in use. We call these the "far rule" and the "near rule", respectively.

Both selection rules were simple, fixed-priority rules. The only difference was the way in which selection priorities were assigned to apexes.

The far rule simply selected the left-most elligible apex, assuming that a network was layed out in the usual manner as illustrated in Figures 4.3-1b, 4.4-1, 4.4-2, 5.1-1, 6.2-2g, 6.2-2h, 6.2-2i, and 7.1-1a. For example, if the far rule were applied to the SW banyan in Figure 6.2-2g, apex 0 would be first choice, apex 1 would be second choice, etc. Apex 15 would be selected only if it were the only apex elligible. Thus, with apexes numbered in this manner, apex I-1 would be the Ith choice according to the far rule. This rule tended to select apexes very distant from those already in use, because consecutive choices generally were the most widely separated apexes in terms of apex distance.

In contrast, the near rule tended to select new apexes close to those already in use, because its consecutive choices tended to be the nearest apexes in terms of apex distance. With apexes numbered in the conventional manner, apex number $(\phi\underline{S})\perp\phi(\underline{S}\mathrm{T}(I\text{-}1))$ would be the Ith choice according to the near rule, where $\underline{S}$ is the spread vector. For example, the apexes of the SW banyan in Figure 6.2-2g would be assigned priorities as shown below in Table 8.1-1. This rule would simply select the left-most eligible apex if the banyan were laid out as shown in Figure 8.1-1. Similarly, the near rule would select apexes of the CC banyan in Figure 7.1-1b according to the priorities listed in Table 8.1-2. This would be equivalent to selecting the left-most eligible apex if the CC banyan were laid out as shown in Figure 8.2-2.

Note that our apex numbering conventions are such that the same far and near rules are applicable to both SW and CC banyans.

TABLE 8.1-1. Application of Near Apex Selection Rule to the Banyan in Figure 6.2-2g

CHOICE NUMBER (I)	$\underline{S}\mathrm{T}(I\text{-}1)$	$\phi(\underline{S}\mathrm{T}(I+1))$	APEX NUMBER
1	0 0 0 0	0 0 0 0	0
2	0 0 0 1	1 0 0 0	8
3	0 0 1 0	0 1 0 0	4
4	0 0 1 1	1 1 0 0	12
5	0 1 0 0	0 0 1 0	2
6	0 1 0 1	1 0 1 0	10
7	0 1 1 0	0 1 1 0	6
8	0 1 1 1	1 1 1 0	14
9	1 0 0 0	0 0 0 1	1
10	1 0 0 1	1 0 0 1	9
11	1 0 1 0	0 1 0 1	5
12	1 0 1 1	1 1 0 1	13

TABLE 8.1-1. (Continued)

CHOICE NUMBER (I)	$\underline{S}_T(I-1)$	$\phi(\underline{S}_T(I-1))$	APEX NUMBER
13	1 1 0 0	0 0 1 1	3
14	1 1 0 1	1 0 1 1	11
15	1 1 1 0	0 1 1 1	7
16	1 1 1 1	1 1 1 1	15

TABLE 8.1-2. Application of Near Apex Selection Rule to the Banyan in Figure 7.1-1b

CHOICE NUMBER (I)	$\underline{S}_T(I-1)$	$\phi(\underline{S}_T(I-1))$	APEX NUMBER
1	0 0	0 0	0
2	0 1	1 0	3
3	1 0	0 1	1
4	1 1	1 1	4
5	2 0	0 2	2
6	2 1	1 2	5

Set-up rules. Two set-up rules were simulated, the standard set-up rule described in Section 4.4 and a modified rule in which the "trunk" portion of a tree-shaped connection was disconnected immediately after set-up. The standard rule sometimes produced tree-shaped connections, like that denoted by heavy lines Figure 8.1-3a, in which no branching existed at the apex or root. In such cases, the portion of a connection between the apex and the highest-level branch point was superfluous once the connection had been established. The modified set-up rule

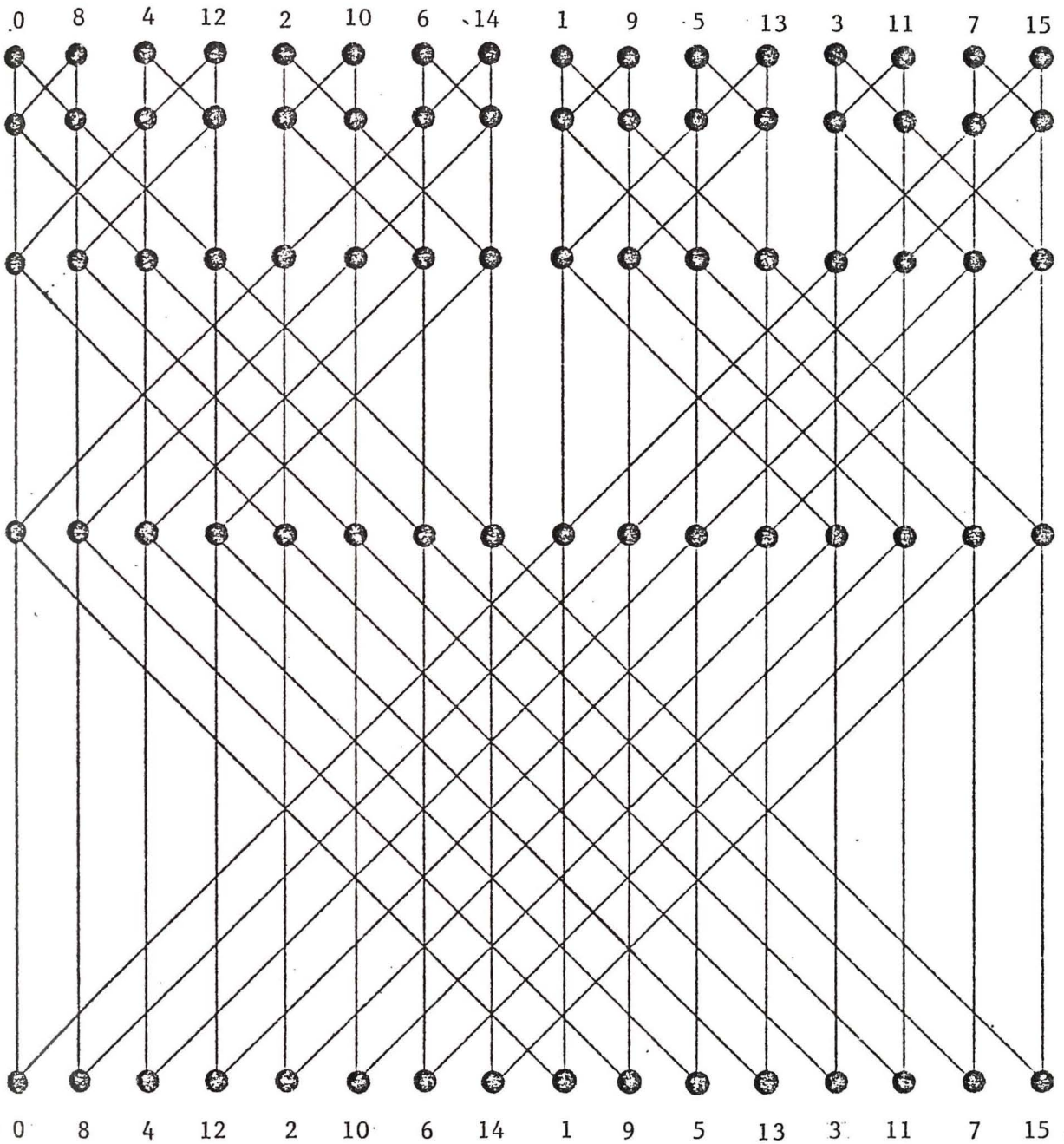


Figure 8.1-1. Redrawn Version of SW Banyan in Figure 6.2-2g

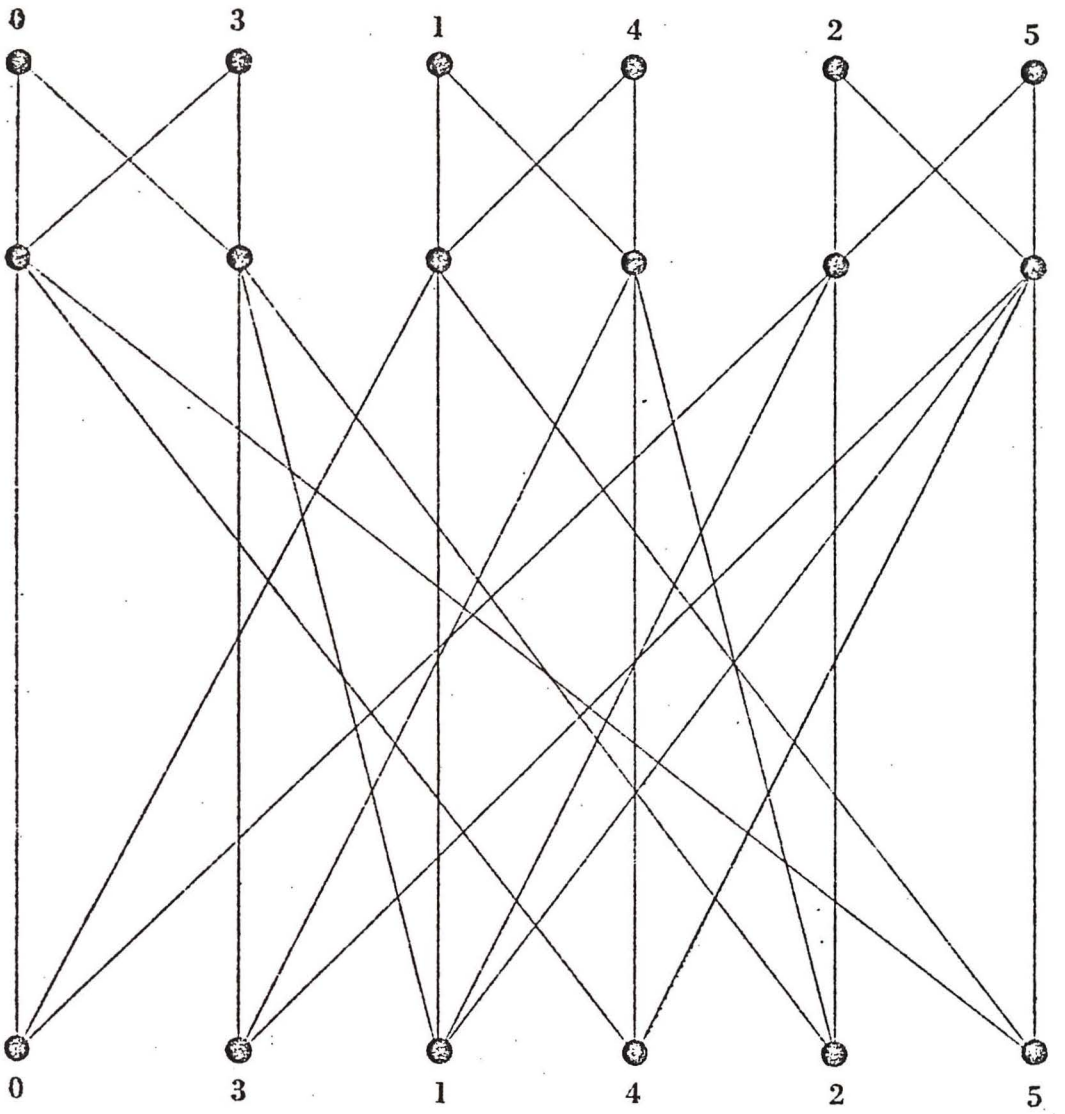
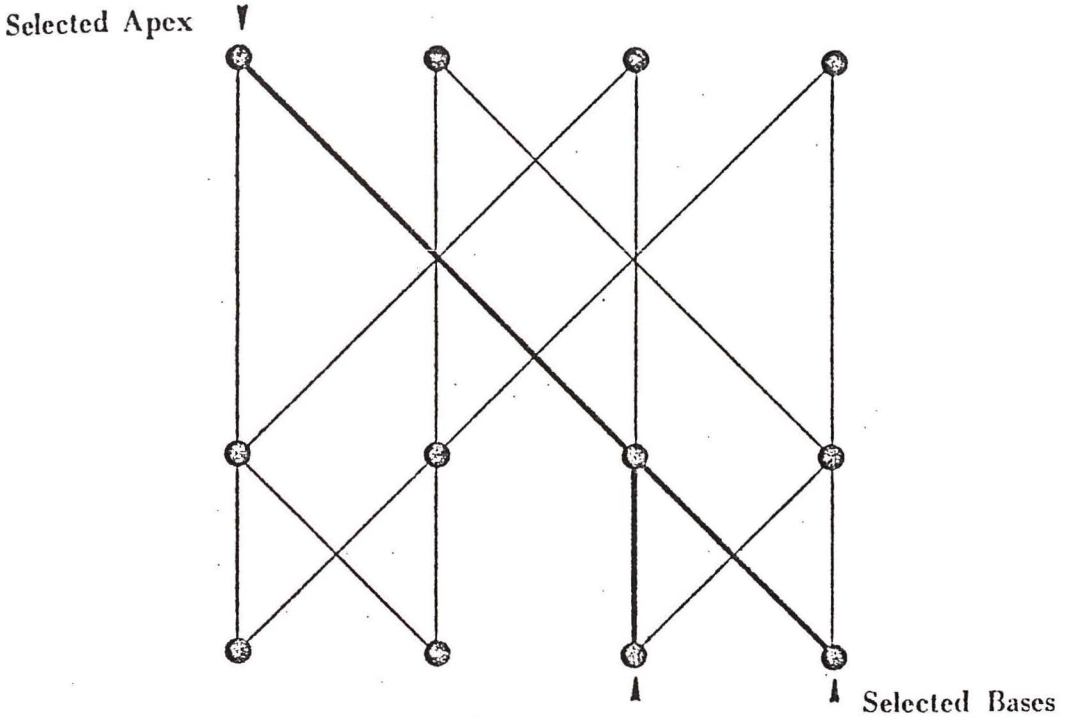
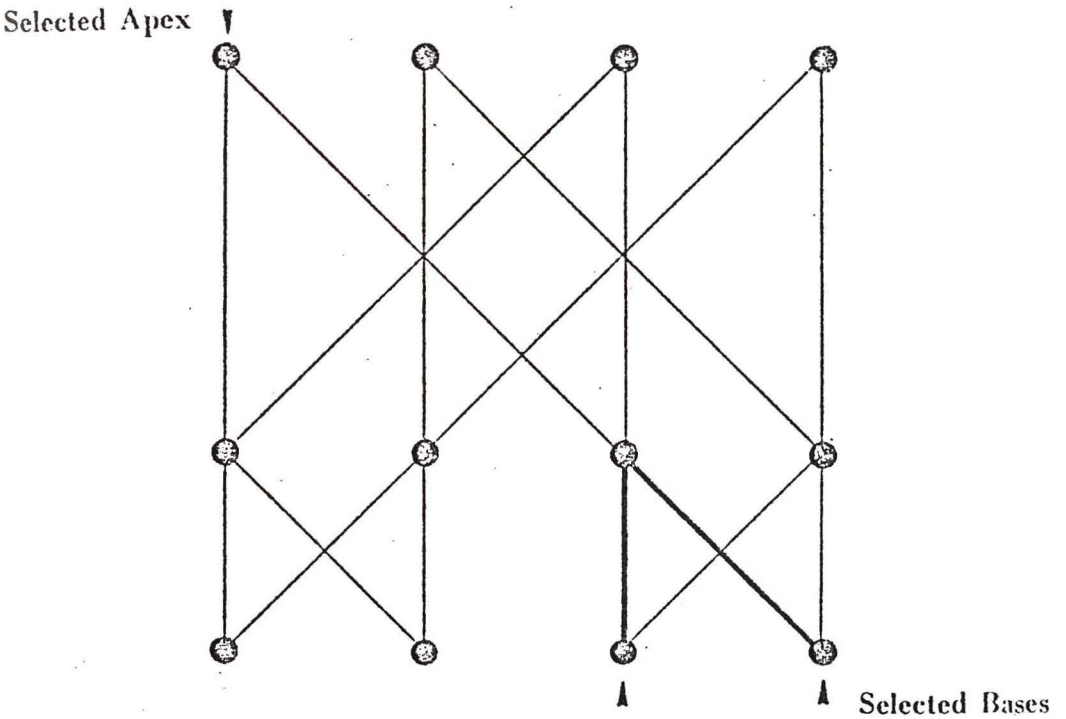


Figure 8.1-2. Redrawn Version of CC Banyan
in Figure 7.1-1b



a) Connection Initially Established



b) Connection Remaining After Set-Up Using Modified Rule

Figure 8.1-3. Set-Up Rule Modification

employed the same search and set-up algorithms as the standard rule (Section 4.4), but disconnected the superfluous portion of a connection immediately after set-up, as illustrated in Figure 8.1-3b. The purpose of this modification was to achieve more efficient network utilization by leaving as much of the partitioning network as possible for connecting other subsystems.

Test case generation. Complete partitions of a network's bases were generated pseudorandomly. First, the number of subsystems in a partition was selected as a pseudorandom number uniformly distributed from 1 to the number of bases in the network. Then each base was assigned pseudorandomly to one of these subsystems such that all subsystems were equally probable. Thus, the number of bases assigned to any subsystem could vary and could even be zero in some cases. Subsystems then were connected one at a time, placing each in the first available layer. All subsystems of the partition were connected in this manner using as many layers as were required. Then, all subsystems were dissolved and the entire procedure was repeated for a total of 100 partitions.

To determine if the number of subsystems in a partition had any substantial effect on network loading, a few simulations also were performed in which the number of subsystems was fixed in advance instead of being selected pseudorandomly for each partition. These simulations were like those described above in all other respects, including the pseudorandom assignment of bases to subsystems.

In certain respects, the test conditions simulated tended to be "worst case" conditions more demanding than those likely to be encountered in practical applications. By assigning every base to some subsystem in each partition, we effectively simulated a situation in which

every port of every resource module was always needed by some subsystem. Also, trivial one-base subsystems were treated just like those with multiple bases in the simulations, even though partitioning network connections for one-base subsystems would be entirely superfluous from a practical standpoint. Further, by assigning bases to subsystems in the manner described, we simulated a situation in which no knowledge of base distance was used to enhance network performance. In most practical situations, however, knowledge of a network's base distance function could be used to enhance performance as suggested in Section 5.1.

Kinds of data collected. Several kinds of data were collected during each simulation. The average number of layers required for fully connecting all partitions was computed along with an estimate of the standard error of this mean. For networks multiplexed as described in Section 4.5, the average number of layers required is a useful performance measure, because it indicates how much the maximum allowable intrasubsystem data rates typically would be diminished due to network time sharing. In interpreting the values obtained, however, one should remember that the test conditions were extremely severe in that all subsystems of each partition were required to exist at the same time.

The maximum number of layers required for fully connecting all partitions was recorded also, because this indicates the maximum number of layers a network should be capable of providing when operated under comparable conditions. This empirically observed maximum was based on a limited sample, however, and should not be taken as a theoretical upper bound.

The distribution of subsystems among layers was also recorded and was expressed as a cumulative percentage of nonempty subsystems con-

nected versus number of layers. That is, we determined the percentage of all nonempty subsystems that were connected in layer 1, the percentage connected in the first 2 layers, the percentage connected in the first 3 layers, etc. The percentage of subsystems connected in a given number of layers can be taken as an indication of how well a network limited to that number of layers would perform, assuming that the subsystems of a partition were isolated subsystems capable of existing at different times.

8.2 Simulation Results

Effects of varying the number of subsystems in a partition. A series of simulations was performed to determine if the number of subsystems in a partition had any substantial effect on the ease with which the subsystem could be connected by a banyan network. This was of interest during the planning of subsequent simulations because we wished to generate test cases that would seriously challenge a network's connecting abilities.

Results of this series of tests are shown in Table 8.2-1. The same network was used in all tests. For each test, 100 partitions were generated as described in Section 8.1 using a fixed number of subsystems per partition. Since this generation procedure made it possible for the actual number of nonempty subsystems in a partition to be somewhat less than the specified number, the total number of nonempty subsystems in all 100 partitions is listed in the right-most column for each test.

These results indicate that, except when the number of subsystems per partition is extremely small, variations in this number have little effect on the average or maximum number of layers required. The percentage of subsystems connected in the first layer grew slowly but steadily with the number of subsystems per partition, except when the number of subsystems was so small that all could be connected in the first layer. Clearly, partitions are easiest for such a structure to connect when the number of subsystems is extremely small. In fact, it has been observed that a partition can always be connected in a single-layer SW or CC banyan if it contains no more than $\underline{S}[1]$ subsystems, where $\underline{S}$ is the network's spread vector. Aside from such extreme cases, however, the number of subsystems per partition appeared to have relatively little

TABLE 8.2-1 Effects of Varying the Number of Subsystems in a Partition

Structure: SW Apex Selection Rule: Far Set-Up Rule: Standard

Fanout and Spread: 2 Number of Bases: 32

Subsystems in Each Partition (including empty subsystems)	Layers Required			Nonempty Subsystems Connected (percent)				Total Nonempty Subsystems
	Mean	Standard Error of Mean	Maximum	1 Layer	2 Layers	3 Layers	4 Layers	
2	1.0	0.0	1	100.00				200
4	1.96	.020	2	60.25	100.00			400
8	2.10	.030	3	69.03	98.74	100.00		791
16	2.04	.020	3	75.96	99.72	100.00		1406
32	2.05	.022	3	84.42	99.71	100.00		2054

effect on the ease with which the partition could be realized, at least by the network tested.

Subsequent simulations. In subsequent simulations, the number of subsystems in each partition was selected pseudorandomly as described in Section 8.1. The pseudorandom number generator used was initialized with the same seed at the beginning of each simulation so that different networks with the same number of bases were tested with the same set of partitions. Complete results of these simulations are tabulated in Appendix D. Intepretations of these results will be discussed in the following paragraphs, and relevant portions of the results will be presented in different forms where necessary.

Effects of network size and fanout-spread on average layers required. Figures 8.2-1 through 8.2-4 show how the average number of layers required varied with the number of bases and with a network's fanout-spread parameter (F). Each mean is plotted along with a confidence interval of plus and minus two standard errors of the mean, which corresponds to a confidence level of approximately 95 percent. Notice that semilog graphs are shown so that straight-line plots represent logarithmic functions. The lines shown were drawn visually based on the points plotted.

It is apparent from these graphs that, for each type of network, the average layers required increased with increasing network size and decreased with increasing fanout-spread. Further, wherever three or more data points were plotted for the same fanout-spread, the average layers required appear to have grown no more rapidly than a logarithmically function of the number of bases. This is evidenced by the fact that most such plots either closely approximated straight lines or else

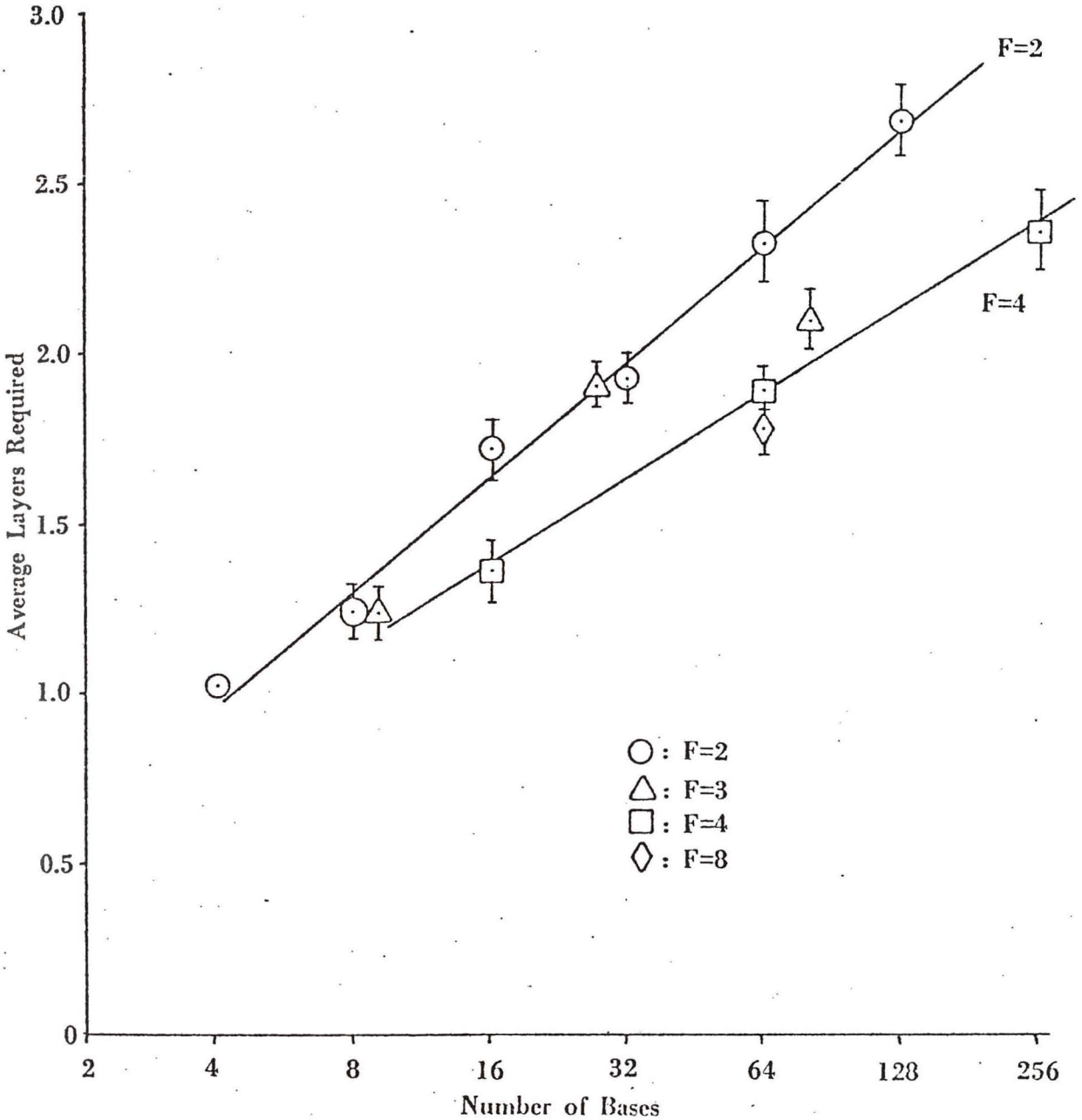


Figure 8.2-1. Average Layers Required for SW Banyans Using Far Apex Selection Rule and Standard Set-Up Rule

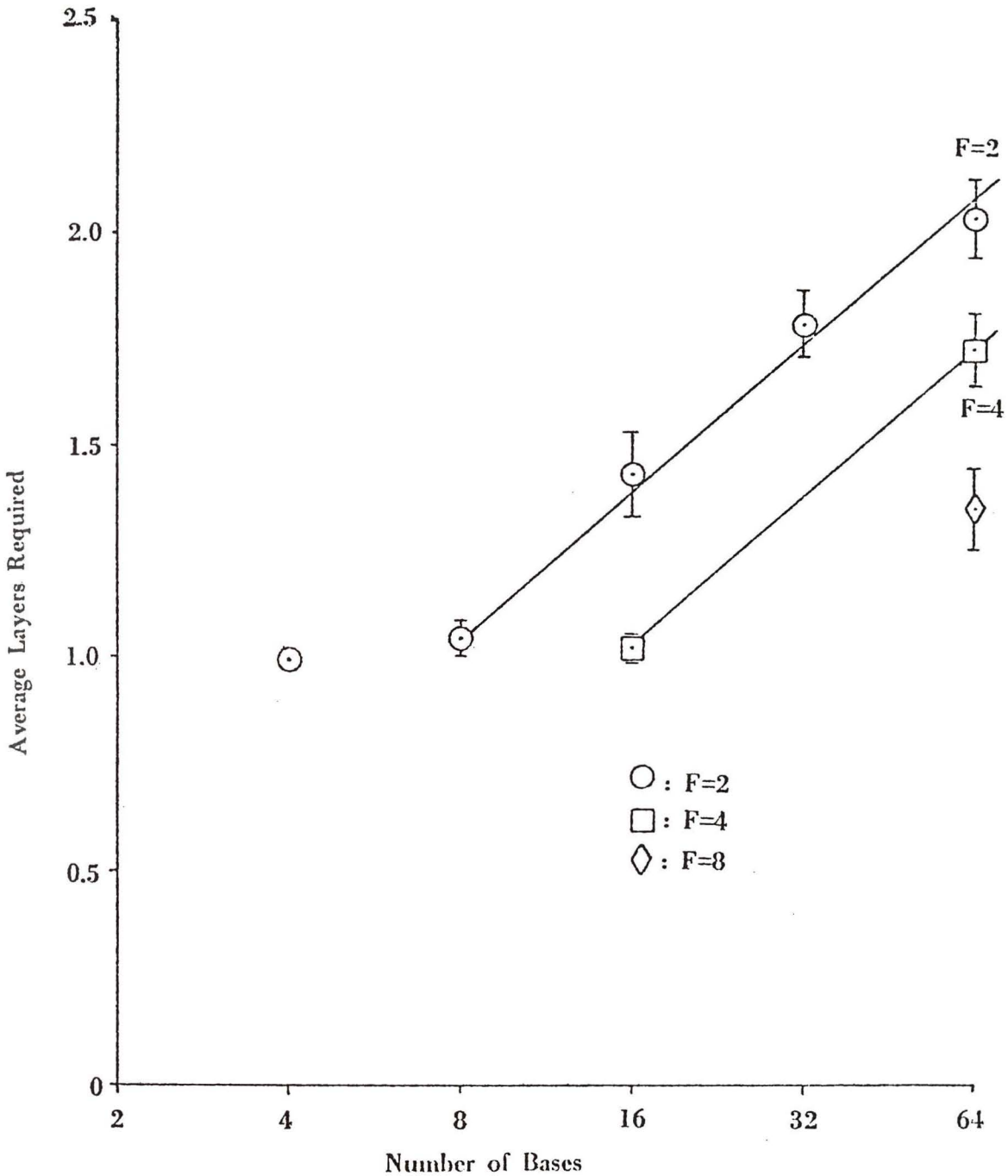


Figure 8.2-2. Average Layers Required for SW Banyans Using Near Apex Selection Rule and Modified Set-Up Rule

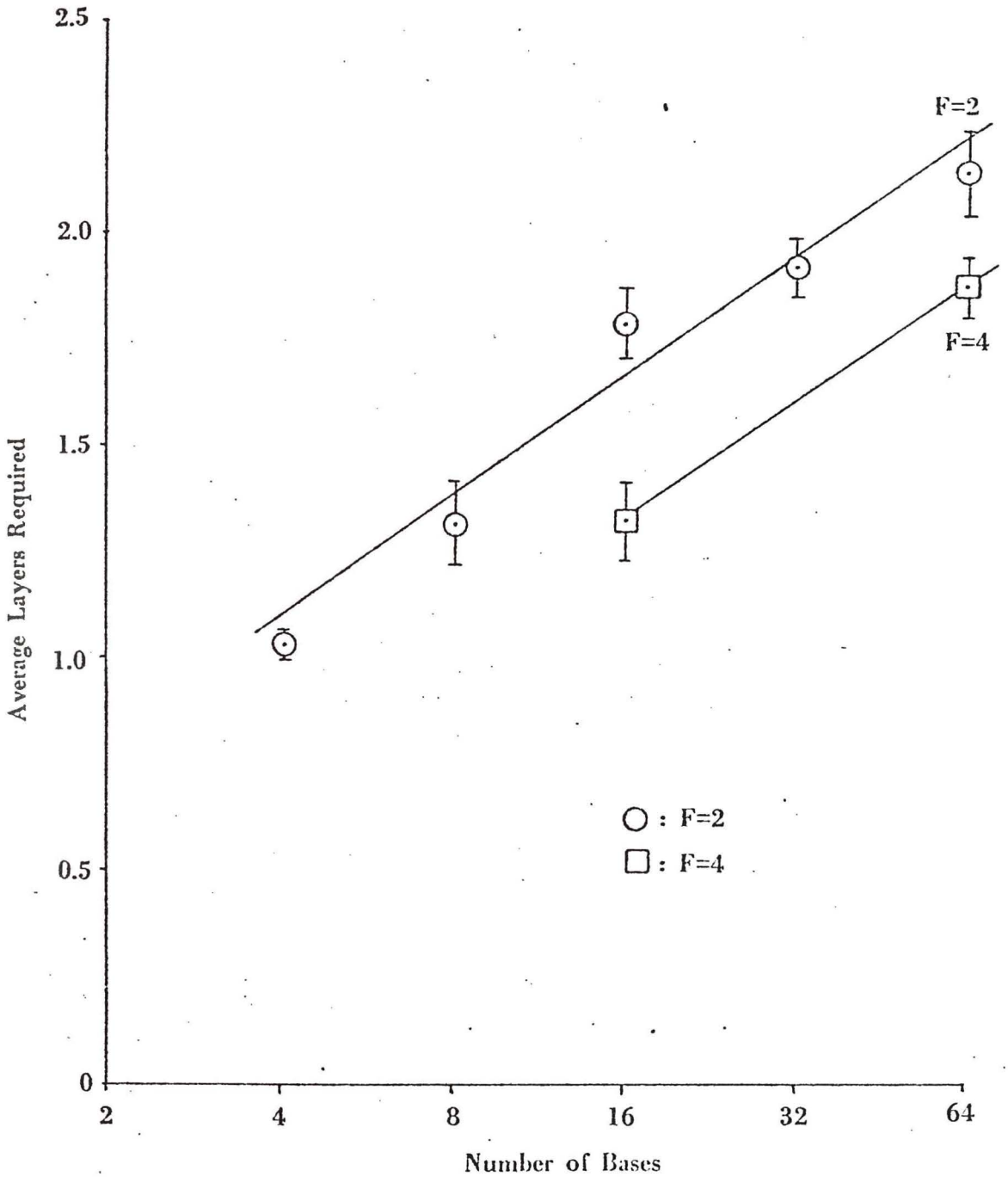


Figure 8.2-3. Average Layers Required for CC Banyans Using Near Apex Selection Rule and Standard Set-Up Rule

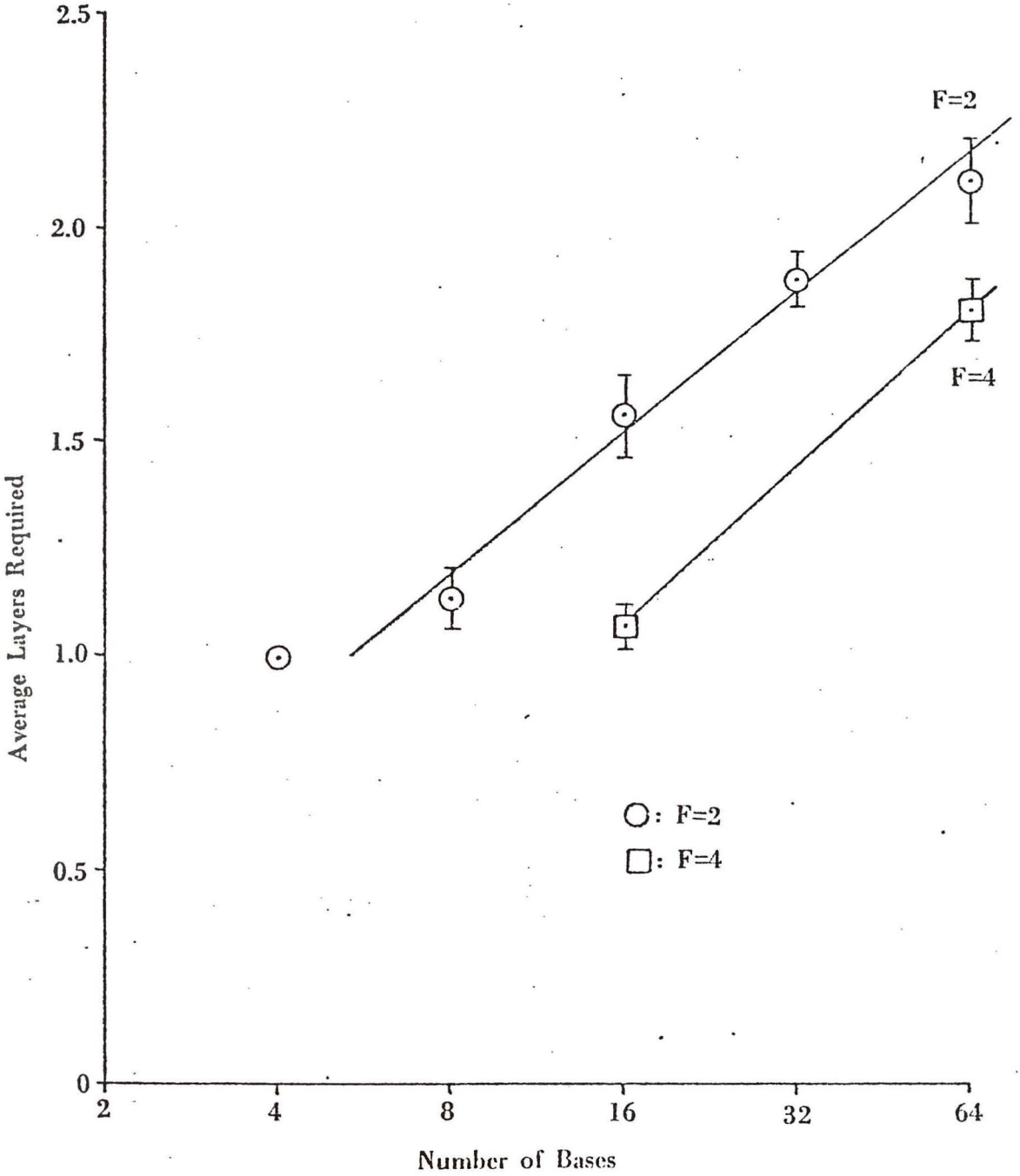


Figure 8.2-4. Average Layers Required for CC Banyans Using Near Apex Selection Rule and Modified Set-Up Rule

curved downward indicating that the average layers required grew less rapidly than a logarithmic function. The only notable exceptions to this appear at the low ends of some plots (e.g., for fanout and spread equal to 2 in Figures 8.2-2 and 8.2-4) where the plot becomes nearly horizontal as the average layers required approach 1. This anomaly is to be expected, however, since the layers required to connect a partition can never be less than 1.

Effects of network size and fanout-spread on maximum layers required. It is apparent from the tables in Appendix D that the maximum number of layers required generally was related to network size and to fanout-spread much as was the average number of layers required. That is, it tended to increase with increasing network size and tended to decrease with increasing fanout. The maximum number of layers required ranged from 1 through 4 for the various networks simulated. Since only a small range of integer values were covered, it is difficult to assess how rapidly the maximum number of layers grows with system size, but logogarithmic growth appears plausible.

Distribution of subsystems among layers. It is also apparent from the tables in Appendix D that nearly all subsystems were connected in the first layer or two, even in cases where comparatively large values were obtained for the maximum and average layers required. For example, in the largest network simulated, an SW banyan with 256 bases (Table D-1), over 87 percent of the subsystems were connected in the first layer and over 99 percent were connected in the first two, even through an average of 2.39 and a maximum of 4 layers were required to fully connect all partitions. This indicates that only one or a very few layers might

provide sufficient partitioning flexibility for applications involving mostly isolated subsystems.

Comparison of SW and CC networks. Simulation results for comparable SW and CC networks are shown in Table 8.2-2. To facilitate comparison, table entries are paired so that each row for an SW banyan is followed immediately by a row for an otherwise identical CC banyan. Network structure and control rule options are abbreviated as follows:

SW - SW banyan structure

CC - CC banyan structure

F - Far apex selection rule

N - Near apex selection rule

S - Standard set-up rule

M - Modified set-up rule

Although performance differences between the two types of networks were generally minor, SW banyans always performed as well as or better than their corresponding CC banyans. This was true for all performance measures, including average layers required, maximum layers required, and the percentage of subsystems connected in any given number of layers.

Comparison of far and near apex selection rules. Simulation results for comparable networks using far and near apex selection rules are presented in Table 8.2-3. Abbreviations used are the same as for Table 8.2-2. As predicted in Section 5.1, the near rule consistently outperformed the far rule except for one small network for which identical results were obtained with the two rules.

Comparison of standard and modified set-up rules. Simulation results for comparable networks using standard and modified set-up rules are presented in Table 8.2-4. Abbreviations used are the same as for

Table 8.2-2. As expected, the modified rule consistently outperformed the standard rule.

TABLE 8.2-2 Comparison of SW and CC Network Structures

Network Structure and Control Rules	Fanout and Spread	Number of Bases	Layers Required			Subsystems Connected (percent)			
			Mean	Standard Error of Mean	Maximum	1 Layer	2 Layers	3 Layers	4 Layers
SW,F,S	2	64	2.35	.061	3	76.04	98.05	100.00	
CC,F,S	2	64	2.41	.064	4	70.38	97.52	99.96	100.00
SW,N,S	2	8	1.07	.026	2	97.95	100.00		
CC,N,S	2	8	1.32	.047	2	90.35	100.00		
SW,N,S	2	64	2.08	.046	3	81.77	99.35	100.00	
CC,N,S	2	64	2.17	.053	3	73.31	98.94	100.00	
SW,N,S	4	16	1.19	.039	2	97.24	100.00		
CC,N,S	4	16	1.33	.047	2	94.78	100.00		
SW,N,S	4	64	1.89	.031	2	88.61	100.00		
CC,N,S	4	64	1.90	.030	2	85.68	100.00		
SW,N,M	2	4	1.00	0.0	1	100.00			
CC,N,M	2	4	1.00	0.0	1	100.00			
SW,N,M	2	8	1.05	.022	2	98.54	100.00		
CC,N,M	2	8	1.14	.035	2	95.91	100.00		
SW,N,M	2	16	1.44	.050	2	93.32	100.00		
CC,N,M	2	16	1.57	.050	2	89.70	100.00		

TABLE 8.2-2 (Continued)

Network Structure and Control Rules	Fanout and Spread	Number of Bases	Layers Required			Subsystems Connected (percent)			
			Mean	Standard Error of Mean	Maximum	1 Layer	2 Layers	3 Layers	4 Layers
SW,N,M	2	32	1.80	.040	2	87.59	100.00		
CC,N,M	2	32	1.89	.034	3	84.35	99.92	100.00	
SW,N,M	2	64	2.05	.046	3	86.05	99.43	100.00	
CC,N,M	2	64	2.12	.050	3	81.53	99.19	100.00	
SW,N,M	4	16	1.03	.017	2	99.56	100.00		
CC,N,M	4	16	1.07	.026	2	98.84	100.00		
SW,N,M	4	64	1.74	.044	2	92.19	100.00		
CC,N,M	4	64	1.82	.039	2	91.83	100.00		

TABLE 8.2-3 Comparison of Far and Near Apex Selection Rules

Network Structure and Control Rules	Fanout and Spread	Number of Bases	Layers Required			Subsystems Connected (percent)			
			Mean	Standard Error of Mean	Maximum	1 Layer	2 Layers	3 Layers	4 Layers
SW,F,S	2	8	1.25	.044	2	92.69	100.00		
SW,N,S	2	8	1.07	.026	2	97.95	100.00		
SW,F,S	2	64	2.35	.061	3	76.04	98.05	100.00	
SW,N,S	2	64	2.08	.046	3	81.77	99.35	100.00	
SW,F,S	3	27	1.92	.027	2	84.36	100.00		
SW,N,S	3	27	1.82	.039	2	89.55	100.00		
SW,F,S	4	16	1.37	.049	2	93.76	100.00		
SW,N,S	4	16	1.19	.039	2	97.24	100.00		
SW,F,S	4	64	1.91	.032	3	83.40	100.00		
SW,N,S	4	64	1.89	.031	2	88.61	100.00		
SW,F,M	2	8	1.05	.022	2	98.54	100.00		
SW,N,M	2	8	1.05	.022	2	98.54	100.00		
SW,F,M	2	64	2.15	.052	3	84.17	99.02	100.00	
SW,N,M	2	64	2.05	.046	3	86.05	99.43	100.00	
SW,F,M	4	64	1.85	.036	2	90.15	100.00		
SW,N,M	4	64	1.74	.044	2	92.19	100.00		
CC,F,S	2	64	2.41	.064	2	70.38	97.52	99.96	100.00
CC,N,S	2	64	2.17	.053	3	73.31	98.94	100.00	

TABLE 8.2-4 Comparison of Standard and Modified Set-Up Rules

Network Structure and Control Rules	Fanout and Spread	Number of Bases	Layers Required			Subsystems Connected (percent)			
			Mean	Standard Error of Mean	Maximum	1 Layer	2 Layers	3 Layers	4 Layers
SW,F,S	2	8	1.25	.044	2	92.69	100.00		
SW,F,M	2	8	1.05	.022	2	98.54	100.00		
SW,F,S	2	64	2.35	.061	3	76.04	98.05	100.00	
SW,F,M	2	64	2.15	.052	3	84.17	99.02	100.00	
SW,F,S	4	64	1.91	.032	3	82.40	99.96	100.00	
SW,F,M	4	64	1.85	.036	2	90.15	100.00		
SW,N,S	2	8	1.07	.026	2	97.95	100.00		
SW,N,M	2	8	1.05	.022	2	98.54	100.00		
SW,N,S	2	64	2.08	.046	3	81.77	99.35	100.00	
SW,N,M	2	64	2.05	.046	3	86.05	99.43	100.00	
SW,N,S	4	16	1.19	.039	2	97.24	100.00		
SW,N,M	4	16	1.03	.017	2	99.56	100.00		
SW,N,S	4	64	1.89	.031	2	88.61	100.00		
SW,N,S	4	64	1.74	.044	2	92.19	100.00		
CC,N,S	2	4	1.03	.017	2	98.46	100.00		
CC,N,M	2	4	1.00	0.0	1	100.00			

TABLE 8.2-4 (Continued)

Network Structure and Control Rules	Fanout and Spread	Number of Bases	Layers Required			Subsystems Connected (percent)			
			Mean	Standard Error of Mean	Maximum	1 Layer	2 Layers	3 Layers	4 Layers
CC,N,S	2	8	1.32	.047	2	90.35	100.00		
CC,N,M	2	8	1.14	.035	2	95.35	100.00		
CC,N,S	2	16	1.80	.040	2	80.84	100.00		
CC,N,M	2	16	1.57	.050	2	89.70	100.00		
CC,N,S	2	32	1.94	.034	3	76.72	99.76	100.00	
CC,N,M	2	32	1.89	.034	3	84.35	99.92	100.00	
CC,N,S	2	64	2.17	.053	3	73.31	98.94	100.00	
CC,N,M	2	64	2.12	.050	3	81.53	99.19	100.00	
CC,N,S	4	16	1.33	.047	2	94.78	100.00		
CC,N,M	4	16	1.07	.026	2	98.84	100.00		
CC,N,S	4	64	1.90	.030	2	85.68	100.00		
CC,N,M	4	64	1.82	.039	2	91.83	100.00		

SECTION 9

COST AND PERFORMANCE FUNCTIONS

Certain quantitative measures of a banyan network's cost and performance will be discussed in this section. In Section 9.1, we will discuss functional relationships between these measures and network size, and will compare banyan cost and performance functions with those of the crossbar partitioning structure discussed in Section 3.1. Cost and performance measures for a specific banyan network will be compared with those for alternative crossbar networks in Section 9.2. In Section 9.3, it will be shown that the fanout-spread parameter of a regular, rectangular banyan can be selected to optimize a given cost or cost-performance measure and that optimum fanout-spread values are constant with respect to network size.

9.1 Functions of Interest

Number of Arcs. The number of bidirectional switching devices, or "contacts", required is a commonly used measure of the "cost" of connecting network. This "cost" measure is proportional to the number of arcs in a banyan graph and is given by the formulas in Table 5.2-1. Notice that a regular, rectangular banyan with fanout and spread F requires only $(F*L)\times L\times F$ arcs for a network with $F*L$ bases. Thus, such a network with N bases requires only $(F\circledast N)\times F$ arcs per base, which is a logarithmic function of N. A comparable crossbar-based network like that shown in Figure 3.1-1 would require $\lfloor(N\div 2)$ arcs per base, a function which tends to grow linearly with N. Thus, the cost advantage of a regular, rectangular banyan network over a crossbar-based partitioning network improves without limit as network size increases.

Required fanout capabilities. A bidirectional switching device, corresponding to an arc of a banyan graph, must be capable of driving all other switching devices attached to the same vertex. A vertex in level I of a uniform banyan has $\underline{F}[I]$ arcs incident into it and $\underline{S}[I+1]$ arcs incident out from it implying that each of bidirectional switching devices attached to that vertex must be capable of driving $\underline{F}[I]+\underline{S}[I+1]-1$ similar devices. Thus, the fanout capabilities required of the bidirectional switching devices used in a uniform banyan network depend only on the fanout and spread values selected for the network and do not depend on network size. Hence, arbitrarily large banyan networks can be built using switching devices with a limited fanout capability. Similarly, each resource module port, or base, of the banyan network need only drive $\underline{S}[1]$ bidirectional switching devices regardless of network size.

An N-port crossbar network like that in Figure 3.1-1, however, requires each bidirectional switching device to drive as many as N-1 similar devices. Further, each resource module drives $\lfloor(N \div 2)$ bidirectional switching devices. Thus, for any given family of switching devices, the maximum size of a straightforward crossbar partitioning network is limited by device fanout capabilities. This limitation can be overcome by subdividing each bus of a crossbar network into a number of segments interfaced with each other using bidirectional amplifiers. As will be exemplified in the next section, however, such modification of a crossbar partitioning network increases network cost and data propagation delays.

Priority propagation delay. For reasons explained in Section 2.3, priority hardware is likely to be needed to resolve conflicting requests for use of a subsystem bus. If priority hardware is built into an L-level banyan network as described in Section 4.2, then the propagation delay for priority signals will be proportional to the number of levels L. In a regular banyan, there are $F*L$ bases, so the propagation delay for priority signals need grow only logarithmically with network size, assuming constant fanout F. Practically the same propagation delay for priority signals could be achieved in a crossbar network using methods described by Foster (68), and it is unlikely that a better than logarithmic growth rate could be achieved without drastically increasing cost.

Thus, propagation delays for priority signals are likely to be approximately the same for both banyan and crossbar partitioning networks. With either structure it should be easy to achieve short delays which grow only logarithmically with network size.

Data propagation delay. The time required for a data signal to propagate through a connecting network is approximately proportional to the number of switching devices through which the signal must pass. The longest possible signal path in a banyan network is from a base to an apex to another base. Therefore, in an L-level banyan, a data signal would have to propagate through at most $2\times L$ bidirectional switching devices. Hence, the worst case data signal propagation delay in a regular banyan with N bases would be that of $2\times(F\circledast N)$ bidirectional switches, which is a logarithmic function of N.

In a simple crossbar network like that in Figure 3.1-1, a data signal must propagate through only 2 bidirectional switches regardless of network size. This, however, is feasible only in small networks which do not exceed device fanout capabilities. To construct larger crossbar networks with limited-fanout devices, one must divide large busses into small segments interfaced by bidirectional amplifiers. This modification, which will be illustrated in greater detail in Section 9.2, causes data propagation delays to grow logarithmically with system size much as they do for banyan networks.

Thus, data propagation delays, like priority propagation delays are likely to be approximately the same for both banyan and crossbar partitioning networks when large network must be constructed using limited-fanout devices. With either structure, short delays which grow only logarithmically with network size are achievable.

Average layers required. Unlike a full crossbar network, a banyan network may require multiple layers in some applications as was discussed in Section 4.5. If parallel networks are employed, then network cost

can be expected to rise in proportion to the number of parallel networks. If, on the other hand, a single network is multiplexed, then bus acquisition and data transmission times will effectively increase in proportion to the average number of "time slots", or layers, used. Preliminary empirical results presented in Section 8 indicate that the average number of layers required under artificially severe test conditions tends to grow no more rapidly than a logarithmic function of the number of bases in a network. In many banyan network applications, fewer layers might suffice or multiple layers might be entirely unnecessary.

Cost-delay product. Cost-delay products are commonly used cost-performance measures for digital circuits. Similar measures can be useful in comparing the cost-performance potentials of different connecting network structures. A simple cost-delay product for a partitioning network can be obtained by multiplying the number of arcs, or bidirectional switching devices, required for each base of the network times the maximum number of switching devices through which a data signal must propagate. If a network is multiplexed to provide multiple layers, then the rate at which data can be transferred within a subsystem is effectively divided by the number of layers used. To take this into account, the cost-delay product of a network can be multiplied times the average number of layers required to obtain a cost-delay-layer product.

Cost-delay and cost-delay-layer products grow more slowly with network size for regular, rectangular banyans than they do for crossbar partitioning networks. Let N be the number of bases of a partitioning network. For a simple crossbar like that described in Section 3.1, the number of bidirectional switching devices required per base grows linearly

with N, the data propagation delay is a constant 2, and the number of layers required is a constant 1. Hence, the cost-delay and cost-delay-layer products for simple crossbars grow linearly with N. To build large crossbars with limited-fanout devices, however, one must insert a number of bidirectional amplifiers into the network's busses, causing the data propagation delay to grow logarithmically with N. Thus, the cost-delay and products for large crossbar networks can be expected to grow as

$$N \times \circledast N.$$

In regular, rectangular banyans, however, both the data propagation delay and the number of bidirectional switches required per base grow logarithmically with N so that the cost-delay product for such banyan networks grows as

$$(\circledast N) * 2.$$

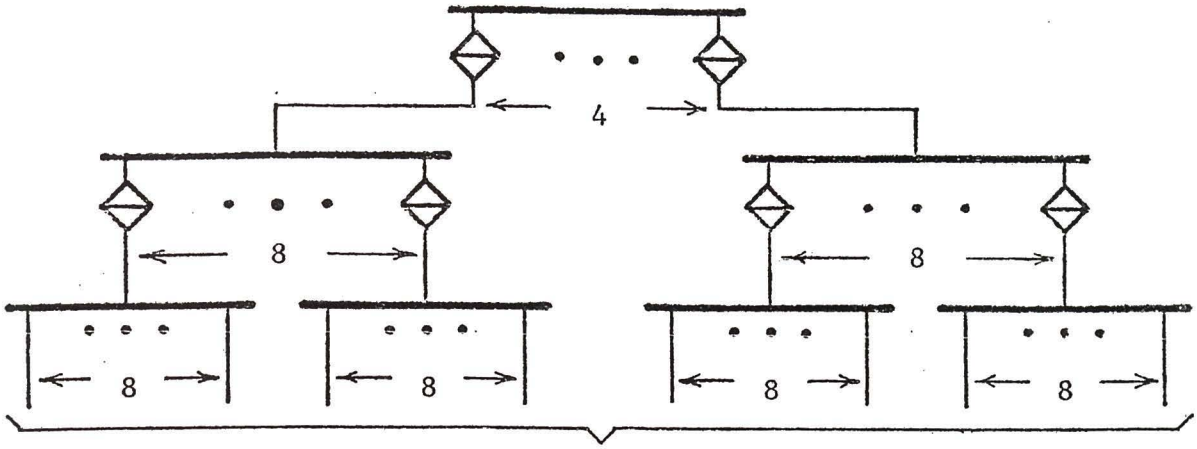
Empirical data discussed in Section 8 indicates that the average number of layers required tends to grow no more rapidly than a logarithmic function of N. This indicates that the cost-delay-layer product for regular, rectangular banyans tends to grow no more rapidly than a function proportional to

$$(\circledast N) * 3.$$

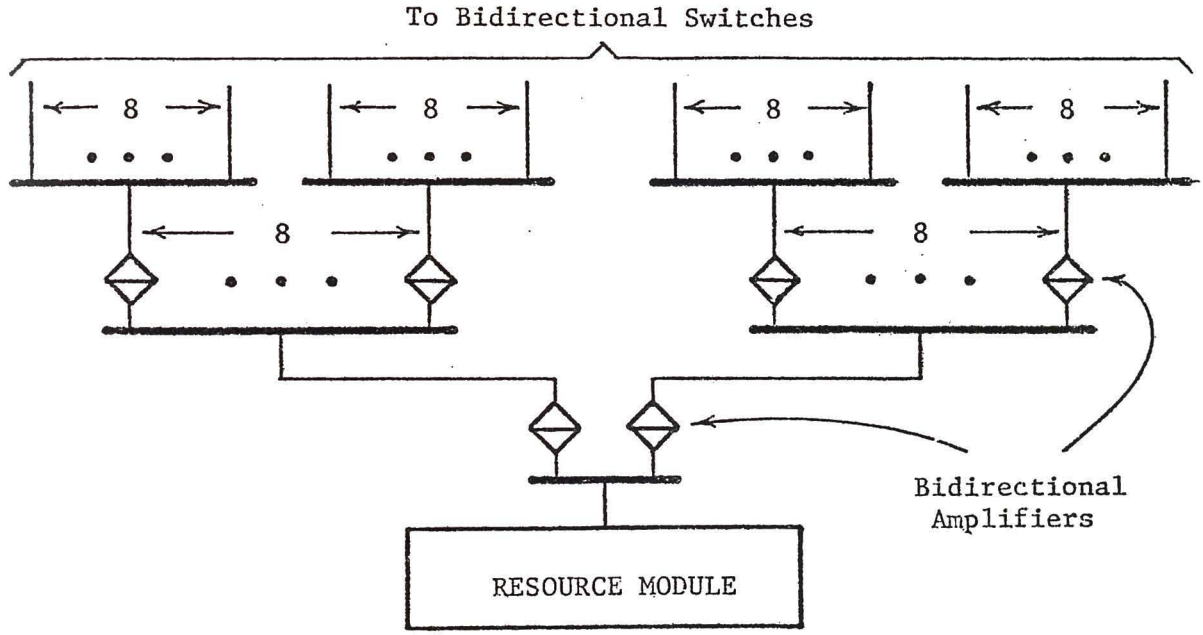
9.2 A Comparative Example

The cost and performance measures discussed in Section 9.1 were evaluated for three 256-base networks and are summarized in Table 9.2-1. The first network listed is a regular, rectangular, 4-level SW banyan with fanout and spread equal to 4. This is not necessarily the optimum banyan structure for 256 bases, but it was selected for comparison because it is the largest banyan for which simulation results were available. The second network listed is a straightforward crossbar partitioning network as described in Section 3.1. This network places severe requirements on the fanout capabilities of both the bidirectional switches and on the resource modules used. To provide a fairer basis for cost-performance comparison, a third network is listed. It is a modified crossbar partitioning network in which a number of bidirectional amplifieres have been inserted to achieve fanout requirements comparable to those of the banyan network.

The modified crossbar network is assumed to be constructed as shown in Figure 9.2-1. To limit the fanout required of the bidirectional switches and amplifiers, each of the 128 data busses in the network is divided into 32 bus segments as illustrated in Figure 9.2-1a. These bus segments are linked by a tree-shaped arrangement of bidirectional amplifiers so that they function together as a single bus. Similarly, each of the 256 resource module ports is connected to 128 bidirectional switches by a tree-shaped structure of bidirectional amplifiers as illustrated in Figure 9.2-1b. This modification requires each port of a resource module to drive only 2 bidirectional amplifiers and requires each input/output of a bidirectional amplifier or switch to drive no more than 8 similar



a) Modified Structure of Each Data Bus



b) Modified Interface with Each Resource Module

Figure 9.2-1. Modification of a Crossbar Partitioning Network to Limit Fanout Requirements

TABLE 9.2-1 Cost and Performance Measures for Three Alternative Networks

Cost or Performance Measure	SW Banyan $F = S = 4$ $L = 4$	Simple Crossbar	Modified Crossbar with Limited Fanout
Total Number of Bidirectional Switches or Amplifiers Required	4,096	32,768	41,984
Number of Bidirectional Switches or Amplifiers Required per Base	16	128	164
Fanout Capability Required of Switches and/or Amplifiers	7	256	8
Fanout Capability Required of Resource Modules	4	128	2
Data Propagation Delay (maximum number of bidirectional switches or amplifiers through which data must propagate)	8	2	10
Average Layers Required	2.39*	1	1
Cost-Delay Product	128	256	1640
Cost-Delay-Layer Product	305.92*	256	1640

*Based on simulation of network using "far" apex selection rule and "standard" set-up rule as described in Section 8.

devices. It may be observed that this modified crossbar network is structurally equivalent to a 5-level banyan with fanout vector 1 1 8 8 4 and with spread vector 2 8 8 1 1, assuming that its bidirectional amplifiers are counted as switching devices.

Entries in Table 9.2-1 were computed as described in Section 9.1. Bidirectional amplifiers in the modified crossbar network were counted together with switches since bidirectional amplifiers and switches require almost identical circuits and would contribute similarly to network cost. The average number of layers required using the SW banyan was obtained from the simulation results reported in Section 8 and Appendix D. The network was simulated using the "far" apex selection rule and the "standard" set-up rule. For reasons explained in Section 8, it is likely that the average number of layers required would have been less if the "near" and "modified" rules had been used instead. Also, it should be remembered that the figure shown represents severe test conditions and might be significantly less in some applications.

The table indicates that the banyan network would cost about an order of magnitude less to build than either crossbar and that it has the smallest cost-delay product. The cost-delay-layer products indicate that the simple crossbar network should be slightly more cost-effective than the banyan network, but this ignores the fact that the simple crossbar requires both bidirectional switches and resource modules to have much greater fanout capabilities. When the banyan network is compared with a crossbar modified to overcome these fanout problems, the cost-delay-layer products indicate better than a five-to-one cost-performance advantage for the banyan. Since these cost and cost-performance measures grow more slowly for banyans than for crossbar-

based structures, they can be expected to favor banyans more strongly for larger networks and less strongly, or not at all, for smaller networks.

This example is intended only to illustrate certain measures of cost and performance and to demonstrate the potential cost-performance advantages of large banyan networks. A much more detailed analysis, of course, would be needed to accurately assess the cost-effectiveness of a particular network in a specific application and system environment.

9.3 Optimum Fanout and Spread

It is shown in Theorems 2.4.1 and 2.4.2 that the fanout-spread parameter of a regular, rectangular banyan can be selected to optimize the network's cost or cost-delay product.

The number of arcs per base in an N-base regular, rectangular banyan is given by

$$\underline{CM}[1] = F \times F \circledast N,$$

which is a measure of network cost per resource module port. The cost-delay product for such a network is given by

$$\begin{aligned} \underline{CM}[2] &= 2 \times L \times \underline{CM}[1] \\ &= 2 \times L \times F \times F \circledast N \\ &= 2 \times (F \circledast N) \times F \times (F \circledast N) \\ &= 2 \times F \times (F \circledast N) * 2. \end{aligned}$$

Each of these functions is of the form

$$\underline{CM}[I] = C \times F \times (F \circledast N) * I,$$

where C is the appropriate proportionality constant, 1 or 2.

A function of this form is minimized with respect to F when $F = e * I$ (Theorem 2.4.1). When F is restricted to positive integer values, as it must be for real networks, the cost measure $\underline{CM}[1]$ is minimized when $F = 3$, and the cost-delay product $\underline{CM}[2]$ is minimized when $F = 7$ (Theorem 2.4.2). The cost-delay product is increased by only about 0.08%, however, when $F = 8$ (proof of Theorem 2.4.2). Also, network "cost" is increased by the same amount when $F = 4$ as it is when $F = 2$ (Theorem 2.4.2). The optimum values for the fanout-spread parameter F are independent of network size, except, of course, for the fact that N must be a power of F in

any regular banyan.

No optimum fanout-spread value has been determined for minimizing a network's cost-delay-layer product, because the effects of fanout-spread on the average number of layers required are not precisely known and may be application dependent. The simulation results indicate that somewhat fewer layers are required when F is large, suggesting that the cost-delay-layer product would tend to be minimized by a fanout-spread value somewhat larger than that required for minimizing the cost-delay product.

SECTION 10

CONCLUSIONS

There is a strong and growing need for switching structures suitable for interconnecting numerous processors and other resource modules in large, general purpose computing systems. Banyan partitioning networks can be used as described in Section 2 to fulfill this need.

Banyan structures have been presented which satisfy the requirements identified in Section 2.3 and which offer potentially large cost-performance advantages over conventional "crossbar", or "multiple-bus", structures for large systems. Regular, rectangular banyans have been described, whose fanout requirements are constant and whose cost-per-port functions grow only logarithmically with the number of resource module ports. This is a significant improvement over crossbar-based structures, whose fanout and cost-per-port functions both grow linearly with network size. Worst-case propagation delays in regular, rectangular banyan networks grow only logarithmically with network size, which is as good as can be achieved with any network using limited-fanout devices. Banyan partitioning networks can be controlled very rapidly and in a potentially fault-tolerant manner using distributed hardware in the network itself, as described in Section 4.4. Priority hardware for resolving bus request conflicts can be built into a banyan network easily, using the technique described in Section 4.2. As noted in Sections 6.2 and 7.1, SW and CC banyans can be constructed in modular fashion by interconnecting identical

"building blocks" suitable for use in any size network. Expansion of a banyan can be accomplished by using the old network as a component banyan of a new synthesized network as described in Section 4.3.

Adequate partitioning flexibility for many applications might be provided by single-layer banyan networks. Simulation of regular, rectangular SW and CC banyans with up to 256 bases has indicated that most subsystems of randomly selected partitions can be connected with only one layer and that all, or nearly all, can be connected with two, even under severe test conditions.

In applications where greater flexibility is needed, any partitioning of system resources into subsystems can be achieved with a multiplexed banyan network. In the simulated networks, the average number of layers required to fully connect random partitions appears to have grown no more rapidly than a logarithmic function of the number of resource module ports. As explained in Section 9, this still allows the potential cost-performance advantage of banyans over crossbar-based networks to improve without limit as network size increases.

The research reported here has focused on the use of banyan networks for partitioning applications. Hence, banyan networks were compared in Section 9 with alternative partitioning networks rather than with networks designed for different functions, such as permuting or store-and-forward message switching. It is felt, however, that the adaptation of banyan structures for such applications warrants further study.

Banyan structures have been defined and analyzed formally using graph theory. A number of useful and mathematically interesting properties of banyans have been identified. This analysis has been oriented towards

the use of banyans as partitioning networks, but as was noted in Sections 6.1 and 7, networks graphically equivalent to special cases of banyans have been proposed or used previously for many different applications. The taxonomy of network structures and the mathematical tools presented here are thus applicable to networks for a variety of other data manipulation functions, such as permuting, shifting, and sorting. Because of its generality, the theory of banyan graphs tends to tie together a number of previous works in addition to establishing useful properties of banyan partitioning networks.

Like most research endeavors, this work has left certain questions unanswered and suggests directions for further research. For banyan partitioning networks, the most pressing need for further research is in the area of performance evaluation. Simulation results reported here indicate that banyans can be much more cost-effective than crossbar-based partitioning networks in very large systems, but caution must be used in extrapolating these results to networks or operating conditions other than those actually simulated. It would be useful to extend these simulations to include larger networks and networks with different fanout and spread vectors. As explained in Section 8.1, however, the test conditions simulated may have been much more severe than those likely to be encountered in practical systems. To accurately estimate banyan performance for any specific purpose, more realistic test conditions will most likely be needed and should take operating system scheduling and resource allocation strategies into account. For example, it is suspected that the number of layers required in most practical applications could be cut drastically if base distance properties were fully exploited as suggested in Section 5.1.

It is hoped that further mathematical research will provide useful results concerning average and maximum layer requirements. A tight theoretical upper bound on the number of layers required to fully connect an arbitrary partition in an SW banyan would be very useful and is being sought by the author. Useful results concerning layer requirements tend to be difficult to derive mathematically because of the multiplicity of connection possibilities, constraints, and options that must be considered. It is felt that this is a high risk, but potentially high payoff, area for further research.

The application of banyan graph theory to other areas also has interesting possibilities. Particularly, the author suspects that banyan theory could prove useful in the study of permutation networks.

Finally, there is a need for further engineering to develop network building blocks which can be produced economically as integrated circuits. If a banyan partitioning network were fabricated today using commercially available components, then bidirectional switches, like those discussed in Appendix C, would have to be assembled from small-scale integrated circuit gates. The TTL circuit in Figure C-3a, for example, would require one IC package per switch. To reduce the cost and physical size of a network, one would like to use large-scale integrated circuits containing many bidirectional switches plus control logic. Technology for doing this already exists, and basic techniques for constructing large banyans in modular form have been proposed here. Further work is needed, however, to design one or more specific module types which could be manufactured efficiently as standard ICs and which would be versatile enough to warrant volume production.

APPENDIX A

MATHEMATICAL NOTATION AND TERMINOLOGY

Graph theoretic terms used in this dissertation were taken mostly from Berge (62), where formal definitions can be found. For the reader's convenience, these terms are explained briefly below.

A graph consists of a set of vertices and a set of arcs. When a diagram is drawn depicting a graph, vertices and arcs are represented by dots and arrows respectively. An arc is incident out from its initial vertex and incident into its terminal vertex.

A path is a sequence of arcs such that the terminal vertex of one arc is always the initial vertex of the next. A path is said to be from the initial vertex of its first arc and to the terminal vertex of its last arc, assuming that the path is finite. A path from a vertex back to that same vertex is called a circuit. A graph without circuits has a partial ordering associated with it and sometimes is referred to as a Hasse diagram. The partial ordering associated with such a graph is a relation on the graph's vertices. It asserts that the two vertices are equal or else that there exists a path from the first to the second.

A graph is said to be nontrivial if it has two or more vertices and is called finite if it has only a finite number of vertices. A subgraph of a graph is formed by deleting zero or more of the graph's vertices and then deleting precisely those arcs whose initial and/or terminal vertices no longer exist. A partial graph of a graph is formed by deleting zero or more of the graph's arcs without changing its vertices. A partial subgraph of a graph G is a partial graph of some subgraph of G.

Sometimes we wish to consider a graph without regard to the direction, or orientation, of its arcs. For this, we define an edge between two vertices to be the set containing those two vertices. We say that a graph contains an edge between two vertices $V1$ and $V2$ if it contains either an arc from $V1$ to $V2$ or an arc from $V2$ to $V1$ or both. A chain is a sequence of edges in which each edge has one vertex in common with the preceding edge and the other vertex in common with the succeeding edge. A finite chain which begins and ends with the same vertex is called a cycle. A connected graph is one in which each pair of vertices can be linked by some chain. A nontrivial, finite, connected graph with no cycles is called a tree.

The notation used for mathematical expressions in this dissertation is an extension of that used in the APL programming language (Gilman and Rose, 70; IBM, 68). APL notation was adopted because it is a compact, standardized, and reasonably well-known notation encompassing a number of powerful vector operations encountered frequently in the theory of banyan graphs. Without these APL operators, it is believed that many of the results derived in Appendix B would have been notably more cumbersome to express and prove. Where necessary, standard APL notation has been extended to include sets, quantifiers, and other constructs needed for mathematical proofs. For consistency, the same APL-based notation is used for all mathematical expressions in this dissertation, even when vector operations are not involved.

APL notation differs from common mathematical notation in that expressions are always evaluated from right to left without regard for operator precedence, except where parentheses explicitly designate a different order of evaluation. For example,

$$A \times B + C$$

equals

$$A \times (B + C)$$

rather than $(A \times B) + C$. Similarly,

$$A - B + C$$

equals

$$A - (B + C)$$

rather than $(A - B) + C$. Some expressions in this dissertation contain redundant parentheses in order to make certain subexpressions more conspicuous and to help prevent misinterpretation by readers unaccustomed to APL notation.

Standard APL operators used in this dissertation are summarized in Table A-1. More detailed explanations of these operators can be found in any book or manual explaining the APL programming language, such as that by Gilman and Rose (70) or that by IBM (68). APL extensions used in this dissertation are summarized in Table A-2. Operators peculiar to banyan theory are not listed but are defined as needed in Appendix B. To make it easier for readers to distinguish between vectors and scalars, vector names are underlined consistently throughout the dissertation.

Additionally, standard APL conventions have been modified with respect to relational operators, such as $=$, $<$, $\leq$, $\geq$, and $>$, so that a conjunction of several relations can be abbreviated in the customary way.

TABLE A-1 Standard APL Notation Used in Dissertation

NOTATION	MEANING
$A+B$	A plus B.
$A-B$	A minus B.
$-A$	Negative A.
$A\times B$	A times B.
$A\div B$	A divided by B.
$A*B$	A raised to the B power.
$A\circledast B$	Log B base A.
$\circledast A$	Natural logarithm of A.
$A\lfloor B$	Minimum of A and B.
$\lfloor A$	Greatest integer not exceeding A.
$A\lceil B$	Maximum of A and B.
$\lceil A$	Least integer not less than A.
$A\mid B$	Residue of B mod A. Result is nonnegative number less than A and is congruent to B mod A.
$P\wedge Q$	P and Q.
$P\vee Q$	P or Q.
$A < B$	A is less than B.
$A \leq B$	A is less than or equal to B.
$A > B$	A is greater than B.
$A \geq B$	A is greater than or equal to B.
$A = B$	A is equal to B.
$A \neq B$	A is not equal to B.
$\underline{V}[I]$	Component I of vector $\underline{V}$.
$\underline{M}[I;J]$	Component in row I and column J of matrix $\underline{M}$.

TABLE A-1 (Continued)

NOTATION	MEANING
$A \in \underline{V}$	A equals some component of vector $\underline{V}$.
$+/\underline{V}$	Sum of all components of vector $\underline{V}$. Result is 0 if $\underline{V}$ has no components.
$\times/\underline{V}$	Product of all components of vector $\underline{V}$. Result is 1 if $\underline{V}$ has no components.
$\lceil/\underline{V}$	Maximum of all components of vector $\underline{V}$.
$\lfloor/\underline{V}$	Minimum of all components of vector $\underline{V}$.
$I\uparrow\underline{V}$	First I components of vector $\underline{V}$. Assuming that I is nonnegative, result is an I-component vector whose components equal the first I components of $\underline{V}$.
$(-I)\uparrow\underline{V}$	Last I components of vector $\underline{V}$. Assuming that I is nonnegative, result is an I-component vector whose components equal the final I components of $\underline{V}$.
$I\downarrow\underline{V}$	All except the first I components of vector $\underline{V}$. Assuming that I is nonnegative, result is vector formed by deleting the first I components of $\underline{V}$.
$(-I)\downarrow\underline{V}$	All except last I components of vector $\underline{V}$. Assuming that I is nonnegative, result is vector formed by deleting the final I components of $\underline{V}$.
$\phi\underline{V}$	Reverse of vector $\underline{V}$. Result is vector identical to $\underline{V}$ except that the order of components is reversed.
$\underline{V},\underline{W}$	Vector formed by catenating $\underline{V}$ and $\underline{W}$. The components of $\underline{V}$ become the first components of result, and the components of $\underline{W}$ become final components of result.
$\underline{V}\perp\underline{W}$	The integer obtained by decoding vector $\underline{W}$ with respect to the mixed base $\underline{V}$. The components of $\underline{W}$ are interpreted as digits of a number in a mixed-base number system. The first components of $\underline{W}$ are interpreted as the most significant digits, the last as least significant. Vector $\underline{V}$ identifies the mixed base such that its first components correspond to the most significant digits. See APL reference for details.

TABLE A-1 (Continued)

NOTATION	MEANING
$\underline{V}\top I$	The vector whose components represent I in the mixed-base number system specified by vector $\underline{V}$. For a given mixed base $\underline{V}$, $\underline{V}\top$ is the inverse of function $\underline{V}\perp$ above. See APL reference for details.

TABLE A-2 Extensions to APL Notation Used in Dissertation

NOTATION	MEANING
$E1 \leftrightarrow E2$	Expression $E1$ is defined to be equivalent to expression $E2$.
$\underline{V}[I\sim J]$	This denotes the vector consisting of components I through J of vector $\underline{V}$, provided that $\underline{V}$ is already defined. When used in the definition vector $\underline{V}$, however, "$\underline{V}[I\sim J]$" means that the components of $\underline{V}$ are to be indexed by the integers I through J. This convention extends in the obvious way to arrays of more than one dimension.
ϕ	The null set.
$[X: P(X)]$	The set of all X such that $P(X)$ is true, where $P(X)$ is some expression denoting a logical function of X.
$A \in S$	A is an element of set S.
$S \subseteq T$	Set S is a subset of set T.
$S \subset T$	Set S is a proper subset of set T.
$S \supseteq T$	Set S contains set T.
$S \supset T$	Set S contains set T as a proper subset.
$S \times T$	Cartesian product of sets S and T. $S \times T \leftrightarrow [\underline{X}[1\sim 2]: (\underline{X}[1] \in S) \wedge (\underline{X}[2] \in T)]$
$(\exists X)$	There exists an X such that...
$+[I = J \sim N]$	Summation from $I = J$ to N. This is equivalent to $\sum_{I=J}^{N}$ in conventional notation.
$\times[I = J \sim N]$	Product from $I = J$ to N. This is equivalent to $\prod_{I=J}^{N}$ in conventional notation.
$\underline{V}\perp\underline{W}$	$\leftrightarrow (\phi\underline{V})\perp(\phi\underline{W})$. This is like $\underline{V}\perp\underline{W}$ except that first components of vectors $\underline{V}$ and $\underline{W}$ correspond to the least significant digits instead of most significant.
$\underline{V}\top I$	$\leftrightarrow \phi(\phi\underline{V})\top I$. For a given mixed base $\underline{V}$, $\underline{V}\top$ is the inverse of function $\underline{V}\perp$ above.

APPENDIX B

THE THEORY OF BANYAN GRAPHS

Banyans are defined and analyzed with the use of graph theory in this appendix. The graph theoretic terms used are explained in Appendix A and are mostly taken from Berge (62). The mathematical notation used is also explained in Appendix A and is basically an extension of that used in the APL programming language.

Definitions, theorems, and corollaries are numbered hierarchically. The first part of each number identifies the subsection of this appendix in which the definition, theorem, or corollary appears. For example, Theorem 3.2.1 is the first theorem in Section B.3.2. Lemmas are numbered as theorems. Corollaries of a theorem are numbered by appending letters to the corresponding theorem number.

The outline below summarizes each lemma, theorem, and corollary presented in this appendix and identifies the terms or symbols defined in each definition. It is included for the reader's convenience to simplify reference.

B.1 Banyans

Def. 1.1. Base, apex, intermediate.

Def. 1.2. Banyan, above, below.

Def. 1.3. ⊴, ⊵.

B.1.1 Connecting Trees

Def. 1.1.1. Connecting tree.

Th. 1.1.1. Connecting trees are trees.

B.1.2 Banyan Synthesis

Def. 1.2.1. Synthesized graph, component set, component banyan, interconnection graph.

Th. 1.2.1. In an interconnection graph, the arcs incident into a component banyan correspond to the component banyan's bases and the arcs incident out from a component banyan correspond to its apexes.

Th. 1.2.2. If $V1'$ and $V2'$ are distinct component banyans and $V1$ is a base of $V1'$ and $V2$ is an apex of $V2'$, then there exists a path from $V1$ to $V2$ in the synthesized graph iff there exists a path from $V1'$ to $V2'$ in its interconnection graph.

Cor. 1.2.2a. If $V1$ is an apex of component banyan $V1'$ and $V2$ is a base of component banyan $V2'$, then $V1 \trianglelefteq V2$ in the synthesized graph iff $V1' \trianglelefteq V2'$ in its interconnection graph.

Th. 1.2.3. A synthesized graph is a banyan iff its interconnection graph is a banyan.

B.1.3 Control of Connections

Th. 1.3.1. The set of vertices in the connecting tree connecting an apex A with a set of bases SB in a banyan is $(\mathbb{B}A)\cap\mathbb{A}SB$.

Th. 1.3.2. If SB is a set of bases and SV is a set of vertices of a banyan, then the connecting tree connecting SB with an apex A contains one or more of the vertices in SV iff $A \in \mathbb{A}(SV\cap\mathbb{A}SB)$.

B.1.4 Connectability

Def. 1.4.1. Subsystem.

Def. 1.4.2. Call, callset.

Def. 1.4.3. Conflict.

Def. 1.4.4. Connectable.

Def. 1.4.5. K-Connectable.

B.2 L-Level Banyans

Def. 2.1. L-level banyan, level.

B.2.1 Base and Apex Distances

Def. 2.1.1. $\mathbb{D}$.

Def. 2.1.2. Base distance $\mathbb{A}$, apex distance $\mathbb{V}$.

Th. 2.1.1. Base and apex distance operators are commutative.

Th. 2.1.2. No base or apex distance can be less than zero or greater than L.

Th. 2.1.3. If $B1$ and $B2$ are bases, then $0 = B1\mathbb{A}B2$ iff $B1 = B2$. If $A1$ and $A2$ are apexes, then $0 = A1\mathbb{V}A2$ iff $A1 = A2$.

Th. 2.1.4. If $SB1$ and $SB2$ are subsystems, then $0 = SB1\mathbb{A}SB2$ iff $\phi \neq SB1\cap SB2$. If $SA1$ and $SA2$ are sets of apexes, then $0 = SA1\mathbb{A}SA2$ iff $\phi \neq SA1\cap SA2$.

Lemma 2.1.5. If $A1$ and $A2$ are apexes (or sets of apexes), $B1$ and $B2$ are bases (or subsystems), and SC is a set of calls from $B1$ (or bases in $B1$) to $A1$ (or apexes in $A1$), then $(SC\boxed{\vee}A2) \geq A1\boxed{\vee}A2$ and $(SC\boxed{\wedge}B2) \geq B1\boxed{\wedge}B2$.

Lemma 2.1.6. Let $SC1$ be a callset, let A be an apex (or set of apexes), let B be a base (or subsystem), and let $SC2$ be a set of calls from B (or bases in B) to A (or apexes in A). Then $SC1$ and $SC2$ will not conflict with each other if $L < (SC1\boxed{\wedge}B)+SC1\boxed{\vee}A$.

Th. 2.1.7. Let $A1$ and $A2$ be apexes (or sets of apexes), let $B1$ and $B2$ be bases (or subsystems), let $SC1$ be a set of calls from $B1$ (or bases in $B1$) to $A1$ (or apexes in $A1$), and let $SC2$ be a set of calls from $B2$ (or bases in $B2$) to $A2$ (or apexes in $A2$). Then $SC1$ and $SC2$ will not conflict with each other if $L < (B1\boxed{\wedge}B2)+A1\boxed{\vee}A2$.

B.2.2 Fanout and Spread

Def. 2.2.1. Uniform, fanout vector, spread vector.

Def. 2.2.2. Rectangular.

Def. 2.2.3. Regular, fanout, spread.

Th. 2.2.1. There are $\times/(I\uparrow\underline{S}),I\downarrow\underline{F}$ vertices in level I of a uniform banyan with fanout vector $\underline{F}$ and spread vector $\underline{F}$.

Cor. 2.2.1a. There are $\times/\underline{F}$ bases and $\times/\underline{S}$ apexes in a uniform banyan with fanout vector $\underline{F}$ and spread vector $\underline{S}$.

Cor. 2.2.1b. A rectangular banyan has the same number of vertices in each level.

Cor. 2.2.1c. A regular banyan with fanout F, spread S, and levels L has $F*L$ bases, $S*L$ apexes, and $(S*I)\times F*L-I$ vertices in level I.

Th. 2.2.2. Let V be a vertex in level I of a uniform banyan with fanout vector $\underline{F}$ and spread vector $\underline{S}$. There are $\times/J\downarrow I\uparrow\underline{F}$ vertices below V in level I if $J<I$, and there are $\times/(J-I)\uparrow I\downarrow\underline{S}$ vertices above V in level J if $J>I$.

Cor. 2.2.2a. If V is a vertex of a uniform banyan with fanout vector $\underline{F}$ and spread vector $\underline{S}$, then there are $\times/I\uparrow\underline{F}$ bases below V and $\times/I\downarrow\underline{S}$ apexes above V.

B.2.3 Cost Functions

Th. 2.3.1. In a uniform banyan, the number of arcs (measure of cost) is given by $CM = +[I=1\sim L]\underline{F}[I]\times\times/(I\uparrow\underline{S}),I\downarrow\underline{F}$.

Cor. 2.3.1a. In a rectangular banyan, $CM = (\times/\underline{F})\times+/\underline{F}$.

Cor. 2.3.1b. In a regular, rectangular banyan, $CM = ((F*L)\times L\times F) = F\times N\times F\circledast N$, where $N = F*L$.

Cor. 2.3.1c. In a regular banyan, $CM = F\times+[I=1\sim L](S*I)\times F*L-I$.

Lemma 2.3.2. If X is a real number other than 1 and L is a positive integer, then $(+[I=1\sim L]X*I) = X\times((X*L)-1)\div X-1$.

Cor. 2.3.1d. In a regular, nonrectangular banyan, $CM = N[0]\times S\times((N[0]\times S)-F)\div S-F$, where $N[0] = F*L$.

B.2.4 Optimum Fanouts

Th. 2.4.1. A cost or cost/performance measure given by $\underline{CM}[I] = F\times N\times(F\circledast N)*I$ for a regular, rectangular banyan is minimized with respect to F when $F = e*I$.

Th. 2.4.2. Optimum integer values of fanout F for a regular, rectangular banyan are 3 using cost measure $\underline{CM}[1]$ above and 7 using cost/performance measure $\underline{CM}[2]$. Also, the value of measure $\underline{CM}[1]$ is the same for $F = 2$ as for $F = 4$.

B.3 SW Banyans

B.3.1 Synthesis

Def. 3.1.1. Crossbar.

Th. 3.1.1. Crossbars are regular. A crossbar with F bases and S apexes has fanout F and spread S.

Def. 3.1.2. SW banyan, synthesized SW banyan, component crossbar.

Th. 3.1.2. SW banyans are banyans.

Th. 3.1.3. SW banyans are L-level banyans. If an SW banyan G' with L' levels is the interconnection graph of a synthesized SW banyan G with L levels, then $L = L'+1$, the bases of the component crossbars in level I of G' are the vertices of level I of G, and the apexes of the component crossbars in level I of G' are the vertices of level $I+1$ of G.

Cor. 3.1.3a. If V is a base or an intermediate of a synthesized SW banyan G with interconnection graph G', then there exists a unique component crossbar V' in level $\lfloor V$ of G' such that V is a base of V'.

Cor. 3.1.3b. If V is an apex or an intermediate of a synthesized SW banyan G with interconnection graph G', then there exists a unique component crossbar V' in level $(\lfloor V)-1$ of G' such that V is an apex of V'.

Th. 3.1.4. If a sythesized SW banyan G has an interconnection graph G' with fanout vector $\underline{F}'$ and spread vector $\underline{S}'$, then G is uniform with fanout vector $B,\underline{F}'$ and spread vector $\underline{S}',A$.

Th. 3.1.5. If a synthesized SW banyan is uniform with fanout vector $\underline{F}$ and spread vector $\underline{S}$, then its interconnection graph is uniform with fanout vector $1\downarrow\underline{F}$ and spread vector $(-1)\downarrow\underline{S}$.

Th. 3.1.6. If $B1$ and $B2$ are distinct bases of a synthesized SW banyan G with interconnection graph G' and if $B1'$ and $B2'$ are the bases of G' that contain $B1$ and $B2$ respectively as bases, then $(B1' \boxtimes B2') = (B1 \boxtimes B2)-1$.

Th. 3.1.7. If $A1$ and $A2$ are distinct apexes of a synthesized SW banyan G with interconnection graph G' and if $A1'$ and $A2'$ are the apexes of G' that contain $A1$ and $A2$ respectively as apexes, then $(A1' \boxtimes A2') = (A1 \boxtimes A2)-1$.

B.3.2 Distance Properties

Th. 3.2.1. If $B1$ and $B2$ are bases of an SW banyan and if V is a vertex such that $(B1 \boxtimes B2) \leq \boxtimes V$, then $B1 \boxleft V$ iff $B2 \boxleft V$.

Th. 3.2.2. If $A1$ and $A2$ are apexes of an SW banyan with L levels and if V is a vertex such that $(A1 \boxtimes A2) \leq L-\boxtimes V$, then $V \boxleft A1$ iff $V \boxleft A2$.

Def. 3.2.1. $\boxtimes_X$.

Def. 3.2.2. $\boxtimes_X$.

Th. 3.2.3. The relation $\boxtimes_X$ is both transitive and symmetric.

Cor. 3.2.3a. If $X \geq 0$ then $\boxtimes_X$ is an equivalence relation.

Cor. 3.2.3b. The base distance operation $\boxtimes$ is a metric on the bases of an SW banyan.

Th. 3.2.4. The relation $\boxtimes_X$ is both transitive and symmetric.

Cor. 3.2.4a. If $X \geq 0$ then $\boxtimes_X$ is an equivalence relation.

Cor. 3.2.4b. The apex distance operation $\boxtimes$ is a metric on the bases of an SW banyan.

Th. 3.2.5. If $0 \le X \le Y$ then the equivalence classes of relation $\boxtimes_X$ are subsets of those of relation $\boxtimes_Y$.

Th. 3.2.6. If $0 \le X \le Y$ then the equivalence classes of relation $\boxtimes_X$ are subsets of those of relation $\boxtimes_Y$.

Th. 3.2.7. If G is a uniform SW banyan with L levels and fanout vector $\underline{F}$ and if I is an integer such that $0 \le I \le L$, then the relation $\boxtimes_I$ partitions the bases of G into $\times/I\uparrow\underline{F}$ equivalence classes containing $\times/I\uparrow\underline{F}$ bases each.

Cor. 3.2.7a. Let G be a uniform SW banyan with L levels and fanout vector $\underline{F}$, and let I be an integer such that $1 \le I \le L$. Then the relation $\boxtimes_{I-1}$ has $\underline{F}[I]$ equivalence classes that are subsets of any given equivalence class of $\boxtimes_I$.

Th. 3.2.8. If G is a uniform SW banyan with L levels and spread vector $\underline{S}$ and if I is an integer such that $0 \le I \le L$, then the relation $\boxtimes_I$ partitions the apexes of G into $\times/(-I)\uparrow\underline{S}$ equivalence classes containing $\times/(-I)\uparrow\underline{S}$ apexes each.

Cor. 3.2.8a. Let G be a uniform SW banyan with L levels and spread vector $\underline{S}$, and let I be an integer such that $1 \le I \le L$. Then the relation $\boxtimes_{I-1}$ has $\underline{S}[L-(I-1)]$ equivalence classes that are subsets of any given equivalence class of $\boxtimes_I$.

B.3.3 Connectability

Th. 3.3.1. Let $SA1$ and $SA2$ be sets of apexes and let $SB1$ and $SB2$ be subsystems of an SW banyan with L levels. Let $SC1 = SB1 \ltimes SA1$ and let $SC2 = SB2 \ltimes SA2$. Then the callsets $SC1$ and $SC2$ conflict with each other iff $L \ge (SB1 \boxtimes SB2) + SA1 \boxtimes SA2$.

B.4 CC Banyans

B.4.1 Structure

Def. 4.1.1. CC Banyan.

Lemma 4.1.1. Every CC banyan is the graph a partial order.

Th. 4.1.2. In a CC banyan, there is exactly one path from a vertex $\underline{V}[I1;J1]$ to vertex $\underline{V}[I2;J2]$ if $I1 < I2$ and $(J2 \ominus J1) < \times/I2\uparrow\underline{S}$ and $0 = (\times/I1\uparrow\underline{S})\mid J2-J1$; otherwise, there is no path.

Cor. 4.1.2a. In a CC banyan $\underline{V}[I1;J1) \boxleftarrow \underline{V}[I2;J2]$ iff $I1 \leq I2$ and $(J2 \ominus J1) < \times/I2\uparrow\underline{S}$ and $0 = (\times/I1\uparrow\underline{S})\mid J2-J1$.

Cor. 4.1.2b. In a CC banyan $\underline{B}[J1] \boxleftarrow \underline{V}[I;J2]$ iff $(J2 \ominus J1) < \times/I\uparrow\underline{S}$.

Cor. 4.1.2c. In a CC banyan $\underline{V}[I;J1] \boxleftarrow \underline{A}[J2]$ iff $0 = (\times/I\uparrow\underline{S})\mid J2-J1$.

Th. 4.1.3. A CC banyan is a rectangular banyan with L levels and with fanout/spread vector $\underline{S}$. Also, $\underline{B}[0\sim N-1]$ are its bases and $\underline{A}[0\sim N-1]$ are its bases.

B.4.2 Distance Properties

Def. 4.2.1. Minimum circular distance $\boxminus$.

Lemma 4.2.1. If $(X \ominus Y) \leq M$ and $(Z \ominus Y) \leq M$, then $(X \boxminus Z) \leq M$, where X, Y, Z, and M are arbitrary integers.

Lemma 4.2.2. If $(Y \ominus X) \leq M$ and $(Y \ominus Z) \leq M$, then $(X \ominus Z) \leq M$, where X, Y, Z, and M are arbitrary integers.

Th. 4.2.3. $\boxminus$ is a metric on the integers $0\sim N-1$.

Th. 4.2.4. If $\underline{B}[J1]$ and $\underline{B}[J2]$ are bases of a CC banyan and I is an integer such that $0 \leq I \leq L$, then $(\underline{B}[J1] \boxtimes \underline{B}[J2]) \leq I$ iff $(J1 \boxminus J2) < \times/I\uparrow\underline{S}$.

Th. 4.2.5. If $\underline{A}[J1]$ and $\underline{A}[J1]$ are apexes of a CC banyan and I is an integer such that $0 \leq I \leq L$, then $(\underline{A}[J1] \boxtimes \underline{A}[J2]) \leq L-1$ iff $0 = (\times/I\uparrow\underline{S})\mid J2-J1$.

Cor. 4.2.5a. If $\underline{A}[J1]$ and $\underline{A}[J2]$ are apexes of a CC banyan and I is an integer such that $0 \leq I \leq L$, then $(\underline{A}[J1]\boxtimes\underline{A}[J2]) \leq I$ iff $0 = (\times/(-I)\downarrow\underline{S})\mid J2-J1$.

Th. 4.2.6. $\boxtimes$ is a metric on the bases of a CC banyan, provided that $2 \leq \underline{S}[I]$ for every $I = 1\sim L$.

Th. 4.2.7. $\boxtimes$ is a metric on the apexes of a CC banyan.

B.1 Banyans

A banyan is defined to be a certain kind of graph.

Definition 1.1. A vertex of a directed graph G is called a base of G iff there are no arcs incident into it in G, is called an apex of G iff there are no arcs incident out from it in G, and is called an intermediate of G otherwise.

Definition 1.2. A banyan is a nontrivial finite graph such that its associated weak ordering $\trianglelefteq$ is a partial ordering[1] and such that for every base B and apex A, there is one and only one path from B to A. If $V1$ and $V2$ are vertices of a banyan and $V1 \trianglelefteq V2$, then $V1$ is said to be below $V2$, and $V2$ is said to be above $V1$.

Notice that by this definition, a vertex is always both above and below itself. For notational convenience, we also define the operators $\blacktriangle$ and $\blacktriangledown$ which give us, respectively, the vertices above and below a vertex or set of vertices.

Definition 1.3. In a banyan with partial order $\trianglelefteq$, let V be a vertex and let SV be a set of vertices. $\blacktriangle$ and $\blacktriangledown$ are monadic operators defined as follows:

$$\blacktriangle V \leftrightarrow [X:\ V \trianglelefteq X]$$

$$\blacktriangle SV \leftrightarrow [X:\ (\exists Y)\ (Y \in SV) \wedge Y \trianglelefteq X]$$

$$\blacktriangledown V \leftrightarrow [X:\ X \trianglelefteq V]$$

$$\blacktriangledown SV \leftrightarrow [X:\ (\exists Y)\ (Y \in SV) \wedge X \trianglelefteq Y]$$

---

[1] A directed graph is associated with a partial order iff it contains no circuits. Such graphs are sometimes called Hasse diagrams. See Berge (62, p. 12).

B.1.1 Connecting Trees

Next, we show that the partial subgraph formed by all paths from a set of bases to a single apex is a tree. This is the basis for asserting in Section 4.1 that subsystems are connected by tree-shaped connections in a banyan network.

Definition 1.1.1. Let G be a banyan, let A be an apex, and let SB be a nonempty set of bases. The partial subgraph of G consisting of all arcs and vertices of all paths from bases in SB to apex A is called a connecting tree.

Theorem 1.1.1. Every connecting tree is a tree.

Proof. Let T be a connecting tree connecting apex A with a set of bases SB in banyan G. Since every vertex of T is a vertex of some path to A, T is connected.

Now suppose that there exists in T a cycle C. T contains no circuits, because it is a partial subgraph of a graph G of a partial order $\trianglelefteq$. Since C is a cycle but not a circuit, it must have some vertex V that is the initial vertex of at least 2 arcs. These 2 or more arcs must be parts of 2 or more nonidentical paths from V to A. Thus, there exists some base B (any base below V will do) such that there are 2 or more paths from B to apex A. But this contradicts the definition of banyan G. Therefore, T contains no cycles.

Since T is connected and contains no cycles, it is a tree.

Q.E.D.

In a similar manner, it can be shown that the paths connecting a single base to a set of apexes is likewise a tree. Also, it is apparent that any banyan with only one apex or with only one base is a tree. These corollaries are simply pointed out for mathematical interest and will not be treated formally.

B.1.2 Banyan Synthesis

The recursive method discussed in Section 4.3 for synthesizing large banyans from smaller ones will be proven in this section.

Definition 1.2.1. Let *CS* be a set of banyan subgraphs of a directed graph *G* such that the arc sets of the banyans in *CS* together form a partitioning of the arcs of *G* and also such that for any $X \in CS$,

1) no intermediate of *X* is a vertex of any other element of *CS*,
2) no base of *X* is a vertex of any other element of *CS* unless each base of *X* is an apex of a different element of *CS*, and
3) no apex of *X* is a vertex of any other element of *CS* unless each apex of *X* is a base of a different element of *CS*.

Any such graph *G* is said to be synthesized, *CS* is called a component set of *G*, and the elements of *CS* are called component banyans of *G*. Now let *G'* be a directed graph on *CS* (i.e., the component banyans of *G* are the vertices of *G'*) such that for any component banyans *V1'* and *V2'* in *CS*, there exists an arc in *G'* from *V1'* to *V2'* iff some apex of *V1'* is also a base of *V2'*. *G'* is called an interconnection graph of *G*.

It is possible for a synthesized graph *G* to have more than one component set, and hence, more than one interconnection graph. Strictly speaking, a subgraph of *G* is a component banyan of *G* if it is an element of any component set of *G*; i.e., if it is a vertex of any interconnection graph of *G*. In subsequent theorems, however, we will generally select a particular interconnection graph of *G* and use the term "component banyan" to refer to its vertices only.

Theorem 1.2.1. Let *G* be a synthesized graph with interconnection graph *G'*, and let *V'* be a component banyan (i.e., a vertex of *G'*). Then, if

there are any arcs incident into V' in G', there is exactly one such arc for each base of V'. Also, if there are any arcs incident out from V' in G', then there is exactly one such arc for each apex of V'.

Proof. Suppose that there is an arc incident into V' in G', so that some base of V' is an apex of another component banyan. By part 2 of Definition 1.2.1, each base of V' is an apex of a different component banyan. By the definition of an interconnection graph, these component banyans correspond one-for-one with the arcs incident into V' in G'. Thus, there is one such arc for each base of V'.

Now suppose that there is an arc incident out from V' in G', so that some apex of V' is a base of another component banyan. By part 3 of Definition 1.2.1, each apex of V' is a base of a different component banyan. By the definition of an interconnection graph, these component banyans correspond one-for-one with the arcs incident out from V' in G'. Thus, there is one such arc for each apex of V'.

Q.E.D

Theorem 1.2.2. Let G be a synthesized graph with interconnection graph G', let $V1'$ and $V2'$ be distinct vertices of G', and let $V1$ and $V2$ be vertices of G such that $V1$ is a base of $V1'$ and $V2$ is an apex of $V2'$. Then there exists a path from $V1$ to $V2$ in G iff there exists a path from $V1'$ to $V2'$ in G'.

Proof. First, suppose that there exists a path from $V1$ to $V2$ in G, and let $A[1],A[2],\ldots,A[N]$ be the sequence of arcs in this path. Since the arc sets of the vertices of G' (component banyans) form a partitioning of the arcs of G, there exists a sequence of indices $I[0],I[1],\ldots,I[N']$ such that

1) $1 < \underline{I}[0] < \underline{I}[2] < \ldots < \underline{I}[N'] = N$,

2) $\underline{A}[1],\ldots,\underline{A}[\underline{I}[0]]$ are arcs of the same component banyan,

3) for each $J = 0,\ldots,N'-1$, the arcs $\underline{A}[\underline{I}[J]+1],\underline{A}[\underline{I}[J]+2],\ldots,$ $\underline{A}[\underline{I}[J+1]]$ belong to the same component banyan but $\underline{A}[\underline{I}[J]]$ does not belong to the same component banyan as a $\underline{A}[\underline{I}[J+1]]$.

Now let $\underline{W}[0],\underline{W}[1],\ldots,\underline{W}[N']$ be the component banyans containing arcs $\underline{A}[\underline{I}[0]],\underline{A}[\underline{I}[1]],\ldots,\underline{A}[\underline{I}[N']]$ respectively. Then $\underline{W}[0]$ is also the component banyan containing $\underline{A}[1]$, which is an arc incident out from $V1$ in G. But since $V1$ is a base of $V1'$, all arcs incident out of $V1$ in G are arcs of $V1'$. Therefore, $\underline{W}[0] = V1'$. Similarly, $\underline{W}[N']$ contains arc $\underline{A}[\underline{I}[N']] = \underline{A}[N]$, which is incident into $V2$ in G. Then since $V2$ is an apex of $V2'$, $\underline{W}[N'] = V2'$. Now for each $J = 0,\ldots,N'-1$, $\underline{A}[\underline{I}[J]]$ and $\underline{A}[\underline{I}[J]+1]$ are consecutive arcs of a path in G. Thus, the terminal vertex of $\underline{A}[\underline{I}[J]]$ is the initial vertex of $\underline{A}[\underline{I}[J]+1]$. But since $\underline{A}[\underline{I}[J]]$ is an arc of $\underline{W}[J]$ and $\underline{A}[\underline{I}[J]+1]$ is an arc of $\underline{W}[J+1]$, this vertex is common to both $\underline{W}[J]$ and $\underline{W}[J+1]$. By Definition 1.2.1, this common vertex must either be an apex of $\underline{W}[J]$ and a base of $\underline{W}[J+1]$ or else a base of $\underline{W}[J]$ and an apex of $\underline{W}[J+1]$. It cannot be a base of $\underline{W}[J]$, however, because there exists an arc $\underline{A}[\underline{I}[J]]$ incident into it in $\underline{W}[J]$. Consequently, the common vertex is an apex of $\underline{W}[J]$ and a base of $\underline{W}[J+1]$, implying that there is an arc from $\underline{W}[J]$ to $\underline{W}[J+1]$ in G'. Therefore, $\underline{W}[0]$, $\underline{W}[1],\ldots,\underline{W}[N']$ are the vertices of a path from $V1'$ to $V2'$ in G'.

Now suppose there exists a path P' from $V1'$ to $V2'$ in G', and let $\underline{W}[0],\underline{W}[1],\ldots,\underline{W}[N']$ be the sequence of vertices of G' along this path. Thus, $\underline{W}[0] = V1'$ and $\underline{W}[N'] = V2'$. Since P' is a path in an interconnection graph, each component banyan $\underline{W}[I]$ such that $1 \leq I \leq N'$ has some base that

is an apex of $\underline{W}[I-1]$. Let $\underline{B}[0] = V1$, let $\underline{B}[N'+1] = V2$, and for each $I = 1,\ldots,N'$, let $\underline{B}[I]$ be a base of $\underline{W}[I]$ which is also an apex of $\underline{W}[I-1]$. Then for each $J = 0,\ldots,N'$, there exists a path from $\underline{B}[J]$ to $\underline{B}[J+1]$ in G, because $\underline{B}[J]$ is a base and $\underline{B}[J+1]$ is an apex of component banyan $\underline{W}[J]$. Therefore, there exists a path from $\underline{B}[0]$ to $\underline{B}[N'+1]$ in G. Since $\underline{B}[0] = V1$ and $\underline{B}[N'+1] = V2$, there exists a path from $V1$ to $V2$ in G.

Q.E.D.

Corollary 1.2.2a. Let G be a synthesized graph with interconnection graph G', let $V1'$ and $V2'$ be any vertices of G' (not necessarily distinct), and let $V1$ and $V2$ be vertices of G such that $V1$ is a base of $V1'$ and $V2$ is an apex of $V2'$. Then $V1 \trianglelefteq V2$ in G iff $V1' \trianglelefteq V2'$ in G'.

Proof. First, suppose that $V1' = V2'$. Then $V1' \trianglelefteq V2'$. Also, since $V1$ is a base and $V2$ is an apex of the same component banyan, there exists a path from $V1$ to $V2$ in G and hence $V1 \trianglelefteq V2$. Therefore, $V1 \trianglelefteq V2$ iff $V1' \trianglelefteq V2'$.

On the other hand, suppose that $V1' \neq V2'$. Then $V1 \trianglelefteq V2$ iff $V1' \trianglelefteq V2'$, by Theorem 1.2.2.

Q.E.D.

Theorem 1.2.3. If G is a synthesized graph with an interconnection graph G', then G is a banyan iff G' is a banyan.

Proof. Let G be a synthesized graph with interconnection graph G'.

There are three ways a directed graph like G or G' can not be a banyan: it could contain a circuit, there could be more than one path from some base to some apex, or there might be no path from some base to some apex. We will show that the existence of any of these conditions in G would cause the same condition to exist in G' and vice versa.

First, suppose that G contains a circuit C. Since no circuit can exist wholly within any component banyan, C must pass through at least two component banyans $V1'$ and $V2'$. Thus, there exists a base $V1$ of $V1'$ and an apex $V2$ of $V2'$ such that both $V1$ and $V2$ are vertices of C. Since there exists a path from $V1$ to $V2$ in G, there exists a path from $V1'$ to $V2'$ in G' by Theorem 1.2.2. If $V1 = V2$ then G' contains an arc from $V2'$ to $V1'$. If $V1 \neq V2$ then G contains a path from $V2$ to $V1$, implying by

Theorem 1.3.2 that G' contains a path from $V1'$ to $V2'$. Thus, in either case, $V1'$ and $V2'$ are vertices of a circuit in G'.

Second, suppose that G contains at least two paths $P1$ and $P2$ from some base B to an apex A. Then B is a base of some base B' of G', and A is an apex of some apex A' of G'. Let $\underline{V1}[0],\underline{V1}[1],\ldots,\underline{V1}[N1]$ be the sequence of vertices along $P1$, and let $\underline{V2}[0],\underline{V2}[1],\ldots,\underline{V2}[N2]$ be the sequence of vertices along $P2$. Let I be the integer such that the first $I+1$ vertices of $P1$ equal the first $I+1$ vertices of $P2$; i.e., $\underline{V1}[I+1] \neq \underline{V2}[I+1]$ but $0 \leq J \leq I$ implies that $\underline{V1}[J] = \underline{V2}[J]$. (Thus, $\underline{V1}[I] = \underline{V2}[I]$.) Let VI' be the component banyan containing $\underline{V1}[I]$ and the arcs incident out from $\underline{V1}[I]$. Thus, $\underline{V1}[I+1]$ and $\underline{V2}[I+1]$ are also in VI'. Since $\underline{V1}[I+1] \neq \underline{V2}[I+1]$ and there cannot be two distinct paths from the same base to the same apex in VI', there exist two distinct apexes $W1$ and $W2$ of VI' such that $W1$ is a vertex of $P1$ and $W2$ is a vertex of $P2$. Then $W1$ and $W2$ must be bases of two distinct component banyans $W1'$ and $W2'$, and hence, there exist arcs from VI' to $W1'$ and from VI' to $W2'$ in G'. Finally, let B' be the base of G' containing B as a base. Then by Theorem 1.2.2, there exist paths in G' from B' to VI', from $W1'$ to A' and from $W2'$ to A'. Therefore, G' contains two distinct paths from B' to A', one of which pases through vertex $W1'$ and the other through $W2'$.

Third, suppose that G contains no path from some base B to some apex A. Let B' be the base of G' that contains B as a base and let A' be the apex of G' that contains A as an apex. Then by Theorem 1.2.2, G' contains no path from B' to G'.

Thus far, we have shown that if G is not a banyan, then neither is G'. The converse is proven next.

First, suppose that G' contains a circuit C'. Let $V1'$ and $V2'$ be the initial and terminal vertices respectively of some arc in C'. Hence, there exists a vertex V of G which is both an apex of $V1'$ and a base of $V2'$. Since V cannot be both a base and an apex of the same component banyan, $V1' \neq V2'$. Thus, by Theorem 1.2.2, there exists a path from V to V in G, implying that G contains a circuit.

Second, suppose that G' contains two distinct paths $P1'$ and $P2'$ from some base B' to an apex A', and let $V1'$ and $V2'$ be distinct vertices of $P1'$ and $P2'$, respectively. Let B be a base of B' and let A be an apex of A'. Thus, B is also a base of G and A is an apex of G. Let $V1A$ be an apex of $V1'$, and let $V2A$ be an apex of $V2'$. Then by Theorem 3, there exist paths in G from B to $V1A$ and from B to $V2A$. The path from B to $V1A$ must pass through some base $V1B$ of $V1'$, and likewise, that from B to $V2A$ must pass through some base $V2B$ of $V2'$. Then by Theorem 1.2.2, there exist paths in G from $V1B$ to A and from $V2B$ to A. Further, $V1B \neq V2B$ since component banyans $V1'$ and $V2'$ cannot have a common base. Therefore, G contains two distinct paths from A to B, one through vertex $V1B$ and the other through $V2B$.

Third, suppose that G' contains no path from some base B' to some apex A'. Let A be an apex of A', and let B be a base of B'. Thus, A is an apex of G, and B is a base of G. By Theorem 1.2.2, G contains no path from B to A.

Q.E.D.

B.1.3 Control of Connections

The two-step setup method described in Section 4.4 can be justified theoretically. When we broadcast a "one" baseward from an apex A, it propagates to all vertices below A. Likewise, when we broadcast "ones" apexward from a set of bases SB, they propagate to all vertices above any bases in SB. The set of vertices selected by the procedure is therefore

$$(\boxtimes A) \cap \boxtimes SB.$$

The next theorem shows that these are precisely the vertices of the connecting tree connecting apex A with the bases in SB.

Theorem 1.3.1. The set of vertices in the connecting tree connecting an apex A with a set of bases SB in a banyan is

$$(\boxtimes A) \cap \boxtimes SB.$$

Proof. The vertices of a path from any base B to apex A are

$$[V\colon (B \boxtimes V) \wedge (V \boxtimes A)].$$

The set of vertices in the desired tree connecting a set of bases SB to apex A is thus

$$[V\colon (\exists B)\ (B \in SB) \wedge (B \boxtimes V) \wedge (V \boxtimes A)]$$

$$= [V\colon (V \boxtimes A) \wedge (\exists B)\ (B \in SB) \wedge (B \boxtimes V)]$$

$$= (\boxtimes A) \cap \boxtimes SB.$$

Q.E.D.

We will next prove that the two-step search procedure described in Section 4.4 correctly selects those apexes which can be used to connect a new block without interfering with those already connected.

Suppose that SV is the set of all vertices that are already in use and, hence, must not be part of the new connecting tree. Suppose also that SB is the set of bases in the subsystem to be connected. In step one

of the procedure, a control signal is broadcast to all vertices above any base in SB; i.e., to $\blacktriangle SB$. A flip-flop is set in each vertex in SV that receives this signal. The vertices thus selected are $SV \cap \blacktriangle SB$.

In step two, each of these vertices broadcasts a signal apexward so that the vertices receiving this signal are

$$\blacktriangle SV \cap \blacktriangle SB.$$

The following theorem will show that the apexes in this set are precisely those which cannot be used to connect subsystem SB.

Theorem 1.3.2. Let SB be a set of bases and let SV be a set of vertices of a banyan. The connecting tree connecting the bases in SB with an apex A contains one or more vertices in SV iff $A \in \blacktriangle SV \cap \blacktriangle SB$.

Proof. The set of vertices in the connecting tree connecting SB with A is $(\blacktriangle SB) \cap \blacktriangledown A$. The intersection of this set with SV is nonempty iff:

$$(\exists V)\ V \in SV \cap (\blacktriangle SB) \cap \blacktriangledown A$$

$$(\exists V)\ (V \in SV \cap \blacktriangle SV) \wedge V \in \blacktriangledown A$$

$$(\exists V)\ (V \in SV \cap \blacktriangle SB) \wedge V \trianglelefteq A$$

$$A \in \blacktriangle SV \cap \blacktriangle SB.$$

Q.E.D.

B.1.4 Connectability

It was pointed out in Section 4.5 that a single banyan network cannot necessarily connect all desired subsystems of an arbitrary partition simultaneously. When considering any particular banyan, it is important to determine how often and under what circumstances multiplexing or additional networks are likely to be needed. We will therefore define several terms concerning the ability of a banyan network to connect multiple subsystems.

Definition 1.4.1. A subsystem is a set of bases.

Definition 1.4.2. If C is a two-element vector, or ordered pair, such that $C[1]$ is a base and $C[2]$ is an apex, then C is said to be a call from base $C[1]$ to apex $C[2]$. A callset is a set of calls.

Calls and callsets are used to characterize connections one might wish to make in a banyan network. For any call C, there corresponds a unique path through the network from base $C[1]$ to apex $C[2]$. A connecting tree connecting a subsystem SB with an apex A is characterized by a callset containing a call from each base in SB to apex A.

Two or more connections can exist simultaneously without interference in a banyan network if and only if no two of them require the same vertex. Connections which cannot coexist in a given network are said to conflict with each other. Similarly, a subsystem is said to conflict with connections or other subsystems if all possible connections (connecting trees) for that subsystem conflict with the connections or other subsystems. A set of connections and/or subsystems is said to be connectable if no two of them conflict.

Definition 1.4.3. Two calls $C1$ and $C2$ conflict with each other iff, in a given banyan, the path from $C1[1]$ to $C1[2]$ has some vertex in common with the path from $C2[1]$ to $C2[2]$. A call C and a callset SC conflict with each other iff some element of SC conflicts with C. A subsystem SB and a call C conflict with each other iff for every apex A, the callset $SB \ltimes [A]$ conflicts with C. A call, callset, or subsystem X and a set of calls, callsets, and/or subsystems SX conflict with each other iff X conflicts with some element of SX. If the elements of two sets $SX1$ and $SX2$ are calls, callsets, and/or subsystems, then the sets conflict with each other iff some element of $SX1$ conflicts with some element of $SX2$.

Definition 1.4.4. A set of calls, callsets, and/or subsystems is connectable iff no two of its elements conflict with each other.

Connections or subsystems which cannot exist simultaneously in a given network can be connected in different layers as described in Section 4.5. A set of connections and/or subsystems is called K-connectable if it can be connected using K or fewer layers.

Definition 1.4.5. A set of calls, callsets, and/or subsystems is K-connectable iff it can be partitioned into K or fewer connectable subsets.

B.2 L-Level Banyans

An L-level banyan, discussed in Section 5, is simply a banyan whose vertices are arranged in levels so that direct connections can only exist between vertices in adjacent levels.

Definition 2.1. An *L*-level banyan is a banyan whose vertices can be partitioned into $L + 1$ subsets $SV[0],\ldots,SV[L]$ such that for any two vertices $V1$ and $V2$, an arc can exist from $V1$ to $V2$ only if $V1 \in SV[I-1]$ and $V2 \in SV[I]$ for some $I = 1 \sim L$. The vertices in each subset $SV[I]$ are said to be in level I.

B.2.1 Base and Apex Distances

The base and apex distance functions discussed in Section 5.1 will be defined formally here. The formal definition is somewhat more general that that of 5.1, because it allows distance to be measured from calls and callsets as well as from bases, subsystems, apexes and sets of apexes.

The monadic operator $\square$, which gives the level of a vertex is also defined for notational convenience.

Definition 2.1.1. Let V be a vertex and let SV be a set of vertices in an L-level banyan. We define the monadic operator $\square$ so that $\square V$ is the level of V and so that $\square SV$ is the set $[\square W: W \in SV]$.

Definition 2.1.2. In an L-level banyan, let $B1$ and $B2$ each be bases or subsystems, and let $A1$ and $A2$ each be apexes or sets of apexes. Also let C be a call or callset and let SV be the set of all vertices in the corresponding paths. The diadic operators $\boxtimes$ and $\boxtimes$ are defined as follows:

$$B1\boxtimes B2 \leftrightarrow L/\square(\boxplus B1)\cap\boxplus B2$$

$$A1\boxtimes A2 \leftrightarrow L - \Gamma/\square(\boxplus A1)\cap\boxplus A2$$

$$B1\boxtimes C \leftrightarrow C\boxtimes B1 \leftrightarrow L/\square\, SV\cap\boxplus B1$$

$$A1\boxtimes C \leftrightarrow C\boxtimes A1 \leftrightarrow L - \Gamma/\square\, SV\cap\boxplus A1$$

Since $\boxtimes$ is defined for bases and $\boxtimes$ for apexes, these operators are called base distance and apex distance respectively. Either may simply be called "distance" when the distinction is clear from context.

The following theorems express fundamental properties of base and apex distances.

Theorem 2.1.1. Base and apex distance operations are both commutative.

Proof. Let $B1$, $B2$, $A1$, $A2$, and C be as defined in Definition 2.1.2. Then

$$(B1\boxtimes B2) = \lfloor/\square(\boxplus B1)\cap\boxplus B2$$

$$= \lfloor/\square(\boxplus B2)\cap\boxplus B1$$

$$= (B2\boxtimes B1)$$

and

$$(A1\boxtimes A2) = L-\Gamma/\square(\boxplus A1)\cap\boxplus A2$$

$$= L-\Gamma/\square(\boxplus A2)\cap\boxplus A1$$

$$= (A2\boxtimes A1).$$

By Definition 2.1.2,

$$(B1\boxtimes C) = C\boxtimes B1$$

and

$$(A1\boxtimes C) = C\boxtimes A1.$$

Q.E.D.

Theorem 2.1.2. In an L-level banyan, no base or apex distance can be less than zero or greater than L.

Proof. Let $B1$, $B2$, $A1$, $A2$, C, and SV be as defined in Definition 2.1.2.

Then each of

$$(\boxplus B1)\cap\boxplus B2$$

$$(\boxminus A1)\cap\boxminus A2$$

$$SV\cap\boxplus B1$$

$$SV\cap\boxminus A1$$

is a set of vertices of an L-level banyan. But since each vertex of an L-level banyan has a non-negative level number not exceeding L,

$$0 \leq (\lfloor/\square(\boxplus B1)\cap\boxplus B2) \leq L$$

$$0 \leq (L-\lceil/\square(\boxminus A1)\cap\boxminus A2) \leq L$$

$$0 \leq (\lfloor/\square SV\cap\boxplus B1) \leq L$$

$$0 \leq (L-\lceil/\square SV\cap\boxminus A1) \leq L$$

Q.E.D.

Theorem 2.1.3. Let $B1$ and $B2$ be bases and let $A1$ and $A2$ be apexes of an L-level banyan G. Then $0 = B1\boxtimes B2$ iff $B1 = B2$, and $0 = A1\boxtimes A2$ iff $A1 = A2$.

Proof. First, suppose that $0 = B1\boxtimes B2$. Then $0 = \lfloor/\square(\boxplus B1)\cap\boxplus B2$, so the set $(\boxplus B1)\cap\boxplus B2$ contains some vertex V in level 0 of G. Hence, $V \in \boxplus B1$ and $V \in \boxplus B2$. But the only element of $\boxplus B1$ in level 0 is $B1$, and the only element of $\boxplus B2$ in level 0 is $B2$. Thus, $V = B1$ and $V = B2$, so $B1 = B2$.

Next, suppose that $B1 = B2$. Then $B1 \in (\boxplus B1)\cap\boxplus B2$. Since $0 = \square B1$, $0 = \lfloor/\square(\boxplus B1)\cap\boxplus B2$, and hence, $0 = B1\boxtimes B2$.

Therefore, $0 = B1\boxtimes B2$ iff $B1 = B2$.

Now, suppose that $0 = A1\boxtimes A2$. Then $0 = L-\lceil/\square(\boxminus A1)\cap\boxminus A2$, so the set $(\boxminus A1)\cap\boxminus A2$ contains some vertex V in level L of G. Hence, $V \in \boxminus A1$ and $V \in \boxminus A2$. But the only element of $\boxminus A1$ in level L is $A1$, and the only element of $\boxminus A2$ in level L is $A2$. Thus, $V = A1$ and $V = A2$, so $A1 = A2$.

Next, suppose that $A1 = A2$. Then $A1 \in (\boxplus A1) \cap \boxplus A2$. Since $L = \boxtimes A1$,

$$L = \Gamma / \boxtimes (\boxplus A1) \cap \boxplus A2$$

$$0 = L - \Gamma / \boxtimes (\boxplus A1) \cap \boxplus A2$$

$$0 = A1 \boxbslash A2.$$

Therefore, $0 = A1 \boxbslash A2$ iff $A1 = A2$.

Q.E.D.

Theorem 2.1.4. Let $SB1$ and $SB2$ be subsystems and let $SA1$ and $SA2$ be sets of apexes of an L-level banyan G. Then $0 = SB1 \boxbar SB2$ iff $\emptyset \neq SB1 \cap SB2$, and $0 = SA1 \boxbslash SA2$ iff $\emptyset \neq SA1 \cap SA2$.

Proof. First, suppose that $0 = SB1 \boxbar SB2$. Then there exist bases $B1 \in SB1$ and $B2 \in SB2$ such that $0 = B1 \boxbar B2$. By Theorem 2.1.3, $B1 = B2$. Therefore, $B1 \in SB1 \cap SB2$, so $\emptyset \neq SB1 \cap SB2$.

Next, suppose that $\emptyset \neq SB1 \cap SB2$. Then there exists a base B such that $B \in SB1$ and $B \in SB2$. But $0 = B \boxbar B$, so $0 = SB1 \boxbar SB2$.

Therefore, $0 = SB1 \boxbar SB2$ iff $\emptyset \neq SB1 \cap SB2$.

Now, suppose that $0 = SA1 \boxbslash SA2$. Then there exist apexes $A1 \in SA1$ and $A2 \in SA2$ such that $0 = A1 \boxbslash A2$. By Theorem 2.1.3, $A1 = A2$. Therefore, $A1 \in SA1 \cap SA2$, so $\emptyset \neq SA1 \cap SB2$.

Next, suppose that $\emptyset \neq SA1 \cap SB2$. Then there exists an apex A such that $A \in SA1$ and $A \in SA2$. But $0 = A \boxbslash A$, so $0 = SA1 \boxbslash SA2$.

Therefore, $0 = SA1 \boxbslash SA2$ iff $\emptyset \neq SA1 \cap SA2$.

Q.E.D.

Theorem 2.1.7, discussed in Section 5.1, will be proven next. Actually, the theorem here will be proven in a more general form that applies to arbitrary callsets, not just those of connecting trees. The proof given will be a trivial consequence of two not-so-trivial lemmas. The lemmas

are separated from the proof of Theorem 2.1.7 since they are felt to be of some theoretical interest in themselves.

Lemma 2.1.5. In an L-level banyan, let $A1$ and $A2$ be apexes (or sets of apexes) and let $B1$ and $B2$ be bases (or subsystems). Let SC be any set of calls from $B1$ (or bases in $B1$) to $A1$ (or apexes in $A1$). Then,

$$(SC\nabla A2) \geq A1\nabla A2$$

and

$$(SC\Delta B2) \geq B1\Delta B2.$$

Proof. Let SV be the set of all vertices of the paths corresponding to callset SC. Hence,

$$SV \subseteq (\Delta B1)\cap\nabla A1.$$

Therefore,

$$SV \subseteq \Delta B1$$

$$(SV\cap\Delta B2) \subseteq (\Delta B1)\cap\Delta B2$$

$$(\square SV\cap\Delta B2) \subseteq \square(\Delta B1)\cap\Delta B2$$

$$(\lfloor/\square SV\cap\Delta B2) \geq \lfloor/\square(\Delta B1)\cap\Delta B2$$

$$(SC\Delta B2) \geq B1\Delta B2.$$

Similarly,

$$SV \subseteq \nabla A1$$

$$(SV\cap\nabla A2) \subseteq (\nabla A1)\cap\nabla A2$$

$$(\square SV\cap\nabla A2) \subseteq \square(\nabla A1)\cap\nabla A2$$

$$(\lceil/\square SV\cap\nabla A2) \leq \lceil/\square(\nabla A1)\cap\nabla A2$$

$$(L\text{-}\lceil/\square SV\cap\nabla A2) \geq L\text{-}\lceil/\square(\nabla A1)\cap\nabla A2$$

$$(SC\nabla A2) \geq A1\nabla A2.$$

Q.E.D.

Lemma 2.1.6. In an L-level banyan, let $SC1$ be a callset, let A be an apex (or set of apexes), let B be a base (or subsystem), and let $SC2$ be a set of calls from B (or bases in B) to A (or apexes in A). Then $SC1$ and $SC2$ will not conflict with each other if $L < (SC1\boxtimes B)+SC1\boxtimes A$.

Proof. Let $SV1$ be the set of all vertices of the paths corresponding to $SC1$, and let $SV2$ be a similar vertex set for $SC2$. Suppose that $SC1$ conflicts with $SC2$. Then there must exist some vertex V such that

$$V \in SV1 \cap SV2.$$

But,

$$SV2 \subseteq \boxtimes B.$$

Therefore,

$$V \in SV1 \cap \boxtimes B$$

$$(\boxtimes V) \geq \lfloor / \boxtimes SV1 \cap \boxtimes B.$$

By Definition 2.1.2,

$$(\boxtimes V) \geq SC1\boxtimes B. \tag{1}$$

Similarly,

$$SV2 \subseteq \boxtimes A$$

$$V \in SV1 \cap \boxtimes A$$

$$(\boxtimes V) \leq \lceil / \boxtimes SV1 \cap \boxtimes A$$

$$(\boxtimes V) \leq L\text{-}SC1\boxtimes A$$

$$L \geq (\boxtimes V)+SC1\boxtimes A. \tag{2}$$

From (1) and (2) above,

$$L \geq (SC1\boxtimes B)+SC1\boxtimes A.$$

Q.E.D.

Theorem 2.1.7. In an L-level banyan, let $A1$ and $A2$ be apexes or sets of apexes, let $B1$ and $B2$ be bases or subsystems, let $SC1$ be a set of calls

from $B1$ (or bases in $B1$) to $A1$ (or apexes in $A1$), and let $SC2$ be a set of calls from $B2$ (or bases in $B2$) to $A1$ (or apexes in $A1$). Then $SC1$ and $SC2$ will not conflict with each other if $L < (B1 \boxtimes B2)+A1 \boxtimes A2$.

Proof. Suppose,

$$L < (B1 \boxtimes B2)+A1 \boxtimes A2.$$

By Lemma 2.1.5

$$((B1 \boxtimes B2)+A1 \boxtimes A2) \leq (SC1 \boxtimes B2)+SC1 \boxtimes A2.$$

Therefore,

$$L < (SC1 \boxtimes B2)+SC1 \boxtimes A2.$$

By Lemma 2.1.6, $SC1$ and $SC2$ are consistent.

Q.E.D.

B.2.2 Fanout and Spread

We next define several terms dealing with the number of arcs incident into and out from vertices in the various levels. These properties, discussed in Section 5.2, are of practical interest because they determine the fanout and fanin requirements of circuits used to realize a network. Also, as will be shown, they can be used to specify the size and shape of an L-level banyan.

Definition 2.2.1. An L-level banyan is uniform iff there exist vectors $\underline{F}[1\sim L]$ and $\underline{S}[1\sim L]$ such that for each vertex V in each level I:

1) the number of arcs incident into V is 0 if $I = 0$ and is $\underline{F}[I]$ otherwise, and
2) the number of arcs incident out from V is 0 if $I = L$ and is $\underline{S}[I+1]$ otherwise.

$\underline{F}$ is called the fanout vector, and $\underline{S}$ is called the spread vector.

Definition 2.2.2. A rectangular banyan is a uniform banyan whose fanout vector $\underline{F}$ is equal to its spread vector $\underline{S}$.

Definition 2.2.3. A uniform banyan is said to be regular with fanout F and spread S iff for each I=1~L, F=$\underline{F}[I]$, and S=$\underline{S}[I]$, where $\underline{F}$ and $\underline{S}$ are the fanout and spread vectors, respectively.

We will next show how the number of vertices in each level of a uniform banyan is related to the fanout and spread vectors.

Theorem 2.2.1. In a uniform banyan with fanout vector $\underline{F}$ and spread vector $\underline{S}$, the number of vertices $\underline{N}[I]$ in level I is given by

$$\underline{N}[I] = \times/(I\uparrow\underline{S}),I\downarrow\underline{F}.$$

Proof. We will use finite induction. First, consider the case where I = 0. Let A be some apex. There are $\times/\underline{F}$ choices of paths from various bases to A. By the definition of a banyan, there is one and only one path from each base to A. Therefore,

$$\underline{N}[0] = \times/\underline{F}$$

$$= \times/(0\uparrow\underline{S}),0\downarrow\underline{F}.$$

Next let $I\epsilon\iota L$ and suppose that $\underline{N}[I-1] = \times/((I-1)\uparrow\underline{S}),(I-1)\downarrow\underline{F}$. Since there are $\underline{S}[I]$ arcs incident out from each vertex in level I-1, the total number of arcs from level I-1 to level I is $\underline{N}[I-1]\times\underline{S}[I]$, which equals $\times/(I\uparrow\underline{S}),(I-1)\downarrow\underline{F}$. Since $\underline{F}[I]$ of these arcs are incident into each vertex in level I,

$$\underline{N}[I] = (\times/(I\uparrow\underline{S}),(I-1)\downarrow\underline{F})\div\underline{F}[I]$$

$$= \times/(I\uparrow\underline{S}),I\downarrow\underline{F}.$$

Q.E.D.

Corollary 2.2.1a.

$$\underline{N}[0] = \times/\underline{F}$$

$$\underline{N}[L] = \times/\underline{S}$$

Proof. Follows from Theorem 2.2.1 by substituting 0 and L respectively for I.

Corollary 2.2.1b. A rectangular banyan has the same number of vertices in each level.

Proof. In a rectangular banyan, the fanout vector $\underline{F}$ equals the spread vector $\underline{S}$; so by substitution into Theorem 2.2.1, we have

$$\underline{N}[I] = \times/(I\uparrow\underline{F}),I\uparrow\underline{F}$$

$$= \times/\underline{F}$$

for each level I.

Q.E.D.

Corollary 2.2.1c. In a regular banyan with fanout F and spread S, the number of vertices in each level I is given by

$$\underline{N}[I] = (S*I)\times F*L-I,$$

the number of bases is

$$\underline{N}[0] = F*L,$$

and the number of apexes is

$$\underline{N}[L] = S*L.$$

Proof. Follows from Theorem 2.2.1 and Definition 2.2.3.

Theorem 2.2.2. Let G be a uniform banyan with L levels, with fanout vector $\underline{F}$, and with spread vector $\underline{S}$, and let V be a vertex in some level I of G. Let J be an integer. If $0 \leq J < I$, then there are exactly $\times/J\downarrow I\uparrow\underline{F}$ vertices below V in level J of G. If $I < J \leq L$, then there are exactly $\times/(J-I)\uparrow I\downarrow\underline{S}$ vertices above V in level J of G.

Proof. For every $0 \leq K \leq L$, let $\underline{SV}[K]$ be the set of all vertices in level K that are either above or below V. Thus, the vertices in $\underline{SV}[K]$ are below V iff $K \leq I$ and are above V iff $I \leq K$.

First, consider the case where $0 \leq J = I-1$. Since G is an L-level banyan, there is exactly one arc incident into V from each vertex in $\underline{SV}[J]$. Since G is uniform, there are $\underline{F}[I]$ such arcs. Thus, the number

of vertices in $\underline{SV}[J]$ is

$$\underline{F}[I]$$

$$= \times/(I-1)\downarrow I\uparrow\underline{F}$$

$$= \times/J\downarrow I\uparrow\underline{F}.$$

Thus, the theorem is true when $0 \leq J = I-1$.

Next, consider the case where $0 \leq J \leq I-2$, and suppose that there are $\times/(J+1)\downarrow I\uparrow\underline{F}$ vertices in $\underline{SV}[J+1]$. It is apparent that $\underline{SV}[J]$ is precisely the set of initial vertices of the arcs incident into the vertices in $\underline{SV}[J+1]$. Suppose that some vertex $V1$ in $\underline{SV}[J]$ were below two distinct vertices $V2$ and $V3$ in $\underline{SV}[J+1]$. Then for some base B below $V1$ and for some apex A above V, there would exist two paths from B to A in G, one passing through $V1$, $V2$, and V and the other passing through $V1$, $V3$, and V. Since this is impossible in a banyan, we conclude that each vertex in $\underline{SV}[J]$ is below only one vertex in $\underline{SV}[J+1]$. Therefore, there is exactly one vertex in $\underline{SV}[J]$ for each arc incident into a vertex in $\underline{SV}[J+1]$. But each vertex in $\underline{SV}[J+1]$ has $\underline{F}[J+1]$ arcs incident into it, so the number of vertices in $\underline{SV}[J]$ is

$$\underline{F}[J+1] \times \times/\ (J+1)\downarrow I\uparrow\underline{F}$$

$$=\times/J\downarrow I\uparrow\underline{F}.$$

Thus, the theorem is also true when $0 \leq J \leq I-2$.

Now consider the case where $(I+1) = J \leq L$. Since G is an L-level banyan, there is exactly one arc incident out from V to each vertex in $\underline{SV}[J]$. Since G is uniform, there are $\underline{S}[I+1]$ such arcs. Thus, the number of vertices in $\underline{SV}[J]$ is

$$\underline{S}[I+1]$$

$$= \times/1\uparrow I\downarrow\underline{S}$$

$$= \times/(J-I)\uparrow I\downarrow\underline{S}.$$

Thus, the theorem is true when $(I+1) = J \leq L$.

Next, consider the case where $(I+2) \leq J \leq L$, and suppose that there are $\times/((J-1)-I)\uparrow I\downarrow \underline{S}$ vertices in $\underline{SV}[J-1]$. It is apparent that $\underline{SV}[J]$ is precisely the set of terminal vertices of the arcs incident out from the vertices in $\underline{SV}[J-1]$. Suppose that some vertex $V1$ in $\underline{SV}[J]$ were above two distinct vertices $V2$ and $V3$ in $\underline{SV}[J-1]$. Then for some base B below V and for some apex A above $V1$, there would exist two paths from B to A in G, one passing through V, $V2$, and $V1$ and the other through V, $V3$, and $V1$. Since this is impossible in a banyan, we conclude that each vertex in $\underline{SV}[J]$ is above only one vertex in $\underline{SV}[J-1]$. Therefore, there is exactly one vertex in $\underline{SV}[J]$ for each arc incident out from a vertex in $\underline{SV}[J-1]$. But each vertex in $\underline{SV}[J-1]$ has $\underline{S}[J]$ arcs incident out from it, so the number of vertices in $\underline{SV}[J]$ is

$$\begin{aligned}
&\underline{S}[J] \times \times/\ ((J-1)-I)\uparrow I\downarrow \underline{S} \\
&= \underline{S}[J] \times \times/\ \underline{S}[(I+1) \sim I+(J-1)-I] \\
&= \underline{S}[J] \times \times/\ \underline{S}[(I+1) \sim J-1] \\
&= \times/\ \underline{S}[(I+1) \sim J] \\
&= \times/\ (J-I)\uparrow I\downarrow \underline{S}.
\end{aligned}$$

Thus, the theorem is also true when $(I+2) \leq J \leq L$.

Q.E.D.

Corallary 2.2.2. Let V be a vertex in level I of a uniform banyan G with L levels, with fanout vector $\underline{F}$, and with spread vector $\underline{S}$. There are exactly $\times/I\uparrow \underline{F}$ bases below V and exactly $\times/I\downarrow \underline{S}$ apexes above V.

Proof. Follows from Theorem 2.2.2 by substituting 0 and L respectively for J.

Q.E.D.

B.2.3 Cost Functions

The "cost" of a connecting network is often measured in terms of the number of contacts, or bidirectional switching devices, required. In the graph of a banyan network, each arc represents a contact (or set of contacts if parallel data paths are used); so the number of arcs is a measure of network "cost" in contacts. The number of arcs in a uniform banyan is a function of its fanout and spread vectors.

Theorem 2.3.1. In a uniform banyan with fanout vector $\underline{F}$, spread vector $\underline{S}$ and L levels, the number of arcs is given by

$$CM = +[I=1\sim L]\ \underline{F}[I] \times \times/(I\uparrow\underline{S}),I\downarrow\underline{F}$$

Proof. For each $I=1\sim L$, there are $\underline{F}[I]$ arcs incident into each vertex in level I, so that the total number of arcs is given by

$$CM = +[I=1\sim L]\ \underline{F}[I] \times \underline{N}[I],$$

where $\underline{N}[I]$ is the number of vertices in level I. By Theorem 2.2.1,

$$CM = +[I=1\sim L]\ \underline{F}[I] \times \times/(I\uparrow\underline{S}),I\downarrow\underline{F}.$$

Q.E.D.

Corollary 2.3.1a. The number of arcs in a rectangular banyan with fanout/spread vector $\underline{F}$ is given by

$$CM = (\times/\underline{F}) \times +/\underline{F}.$$

Proof. In a rectangular banyan $\underline{F} = \underline{S}$. Therefore,

$$\begin{aligned} CM &= +[I=1\sim L]\ \underline{F}[I] \times \times/(I\uparrow\underline{F}),I\downarrow\underline{F} \\ &= +[I=1\sim L]\ \underline{F}[I] \times \times/\underline{F} \\ &= (\times/\underline{F}) \times +[I=1\sim L]\ \underline{F}[I] \\ &= (\times/\underline{F}) \times +/\underline{F}. \end{aligned}$$

Q.E.D.

Corollary 2.3.1b. The number of arcs in a regular, rectangular banyan is given by

$$CM = (F*L) \times L \times F$$

$$= F \times N \times F \circledast N,$$

where F is the fanout, L is the number of levels, and N equals the number of bases $F*L$.

Proof. Follows immediately from Corollary 2.3.1a.

Corollary 2.3.1c. The number of arcs in a regular banyan is given by

$$CM = F \times +[I=1\sim L]\ (S*I)\times F*L-I$$

where F is the fanout, S is the spread, and L is the number of levels.

Proof. By Theorem 2.3.1.

$$CM = +[I=1\sim L]\ F\times(S*I)\times F*L-I$$

$$= F \times +[I=1\sim L]\ (S*I)\times F*L-I.$$

Q.E.D.

Lemma 2.3.2. Let X be a real number other than 1, and let L be a positive integer. Then

$$(+[I=1\sim L]X*I) = X\times((X*L)-1)\div X-1$$

Proof. Let

$$Z = +[I=1\sim L]\ X*I. \tag{1}$$

Then

$$(Z+1) = +[I=0\sim L]\ X*I$$

$$(X\times Z+1) = +[I=1\sim L+1]\ X*I. \tag{2}$$

Subtracting (1) from (2) yields

$$((X\times Z+1)-Z) = X*L+1$$

$$((X\times Z)-Z) = (X*L+1)-X$$

$$(Z\times X-1) = (X*L+1)-X$$

Since $X \neq 1$,

$$Z = ((X*L+1)-X)\div-1$$

$$= X\times((X*L)-1)\div X-1$$

Q.E.D.

Corollary 2.3.1d. The number of arcs in a regular, nonrectangular banyan is given by

$$CM = \underline{N}[0]\times S\times((\underline{N}[0]\times S)-F)\div S-F$$

where F is the fanout, S is the spread, and $\underline{N}[0]$ is the number of bases $F*L$.

Proof. Let

$$K = (\circledast S) \div \circledast F$$

Then,

$$(\circledast S) = K \times \circledast F$$

$$S = F*K. \tag{1}$$

By Corollary 2.3.1c,

$$CM = F \times +[I=1\sim L]\ (S*I)\times F*L-I. \tag{2}$$

Substituting (1) into (2) yields

$$CM = F \times +[I=1\sim L]\ (F*K\times I)\times F*L-I$$

$$= F \times +[I=1\sim L]\ F*(K\times I)+L-I$$

$$= (F*1+L) \times +[I=1\sim L]\ F*(K\times I)-I$$

$$= (F*1+L) \times +[I=1\sim L]\ (F*K-1)*I. \tag{3}$$

Since the banyan is not rectangular,

$$S \neq F$$

$$(\circledast S) \neq \circledast F$$

$$K \neq 1$$

$$(F*K-1) \neq 1.$$

Thus, by Lemma 2.3.2,

$$(+[I=1\sim L]\ (F*K-1)*I) = (F*K-1)\times((F*L\times K-1)-1)\div(F*K-1)-1. \tag{4}$$

Substituting (4) into (3) yields

$$\begin{aligned} CM &= (F*1+L)\times(F*K-1)\times((F*L\times K-1)-1)\div(F*K-1)-1 \\ &= (F*L)\times(F*K)\times(((F*L)\times F*K-1)-1)\div(F*K-1)-1 \\ &= (F*L)\times(F*K)\times(((F*L)\times F*K)-F)\div(F*K)-F. \end{aligned} \tag{5}$$

From Corollary 2.2.1c,

$$(F*L) = \underline{N}[0] \tag{6}$$

Substituting (1) and (6) into (5) yields

$$CM = \underline{N}[0]\times S\times((\underline{N}[0]\times S)-F)\div S-F.$$

Q.E.D.

B.2.4 Optimum Fanouts

Theorems 2.4.1 and 2.4.2 show how the fanout-spread parameter of a regular, rectangular banyan can be selected to minimize either the cost or the cost-delay product discussed in Section 9.

Theorem 2.4.1. Let F be the fanout of a regular, rectangular banyan with N bases, and let I be a positive integer. Any cost or cost-performance function of the form

$$\underline{CM}[I] = C \times F \times (F \circledast N) * I$$

is minimized with respect to F when

$$F = e * I,$$

where e is the natural logarithm base, 2.71828..., and where C is a positive constant.

Proof. To minimize $\underline{CM}[I]$ with respect to F, we set its partial with respect to F equal to zero and solve for F.

$$\begin{aligned}
0 &= \frac{\partial}{\partial F} C \times F \times (F \circledast N) * I \\
&= \frac{\partial}{\partial F} C \times F \times ((\circledast N) \div \circledast F) * I \\
&= C \times ((\circledast N) * I) \times \frac{\partial}{\partial F} F \div (\circledast F) * I \\
0 &= \frac{\partial}{\partial F} F \div (\circledast F) * I \\
&= (((\circledast F) * I) - F \times \frac{\partial}{\partial F} (\circledast F) * I) \div (\circledast F) * I \times 2 \\
0 &= ((\circledast F) * I) - F \times \frac{\partial}{\partial F} (\circledast F) * I \\
&= ((\circledast F) * I) - F \times I \times ((\circledast F) * I - 1) - \frac{\partial}{\partial F} \circledast F \\
&= ((\circledast F) * I) - F \times I \times ((\circledast F) * I - 1) \div F \\
0 &= (\circledast F) - I \\
(\circledast F) &= I \\
F &= e * I
\end{aligned}$$

Q.E.D.

Theorem 2.4.2. Let $CM[I]$ be defined as in Theorem 2.4.1. When F is restricted to positive integer values, $CM[1]$ is minimized with respect to F when $F = 3$, and $CM[2]$ is minimized with respect to F when $F = 7$. Further, the value of $CM[1]$ is the same for $F = 2$ as for $F = 4$.

Proof.

$$\begin{aligned} CM[I] &= C\times F\times(F\circledast N)*I \\ &= C\times F\times((\circledast N)\div\circledast F)*I \\ &= (C\times(\circledast N)*I)\times F\div(\circledast F)*I. \end{aligned}$$

In Theorem 2.4.1, $CM[1]$ was found to have a single minimum at $F = e$, which lies between integers 2 and 3. Consequently, the optimal integer value for F is either 2 or 3. But,

$$((C\times(\circledast N)*1)\times 2\div(\circledast 2)*1) = (C\times\circledast N)\times 2.885,$$

which is greater than

$$((C\times(\circledast N)*1)\times 3\div(\circledast 3)*1) = (C\times\circledast N)\times 2.731.$$

Therefore, $CM[1]$ is minimized with respect to F when $F = 3$. Similarly, $CM[2]$ has a single minimum at $F = e*2$, which lies between integers 7 and 8. The optimal integer value for F is then either 7 or 8. But,

$$((C\times(\circledast N)*2)\times 7\div(\circledast 7)*2) = (C\times(\circledast N)*2)\times 1.8486.$$

which is less than

$$((C\times(\circledast N)*2)\times 8\div(\circledast 8)*2) = (C\times(\circledast N)*2)\times 1.8501.$$

Therefore, $CM[2]$ is minimized with respect to F when $F = 7$. (Note, however, that $CM[2]$ is increased only by about 0.08% when $F = 8$.)

Finally, we show that $CM[1]$ has the same value when $F = 2$ as when $F = 4$. When $F = 2$,

$$CM[1] = (C\times(\circledast N))\times 2\div\circledast 2.$$

When $F = 4$,

$$\underline{CM}[1] = (C\times(\circledast N))\times 4\div\circledast 4$$

$$= (C\times(\circledast N))\times 4\div\circledast 2*2$$

$$= (C\times(\circledast N))\times 4\div 2\times\circledast 2$$

$$= (C\times(\circledast N))\times 2\div\circledast 2$$

Q.E.D.

B.3 SW Banyans

The class of L-level banyans called SW banyans, discussed in Section 6, will be defined and analyzed formally in this section.

B.3.1 Synthesis

SW banyans are synthesized recursively from crossbars using the synthesis principle discussed in Sections 4.3 and B.1.2. In this section, we will define crossbars, define SW banyans recursively in terms of crossbars, prove that SW banyans are indeed banyans, and then analyze relationships between a synthesized SW banyan and its interconnection graph and component banyans.

Definition 3.1.1. A crossbar is a 1-level banyan.

Theorem 3.1.1. Every crossbar is regular. A crossbar with F bases and S apexes has fanout F and spread S.

Proof. Let C be a crossbar with F bases and S apexes. By Definition 3.1.1, C is an L-level banyan. From each vertex in level 0 (i.e., from each base), there must exist exactly one path, and, hence one arc, to each vertex in level 1 (i.e., to each apex). Since there are S apexes, there are S arcs incident out from each vertex in level 0. Similarly, to each apex, there must exist one arc from each base; so there are F arcs incident into each vertex in level 1. C is therefore uniform, and since $L = 1$, it is also regular with fanout F and spread S.

Q.E.D.

Definition 3.1.2. A graph G is called an SW banyan iff either

1) G is a crossbar or

2) G is a synthesized graph whose component banyans are all crossbars and whose interconnection graph is an SW banyan. We call SW banyans formed in this second manner synthesized SW banyans, and we call their component banyans component crossbars.

From this definition, it is apparent that if every crossbar has a property P and if $P(G')$ implies $P(G)$ for every synthesized SW banyan with interconnection graph G', then P is true of every SW banyan. This induction principle will be used in proving many of the following theorems.

Theorem 3.1.2. Every SW banyan is a banyan.

Proof. Every crossbar is an SW banyan by Definition 3.1.1. Now let G be a synthesized SW banyan with interconnection graph G', and suppose that G' is a banyan. Then, by Theorem 1.2.3, G is a banyan.

Q.E.D.

Theorem 3.1.3. Every SW banyan is an L-level banyan. Further, if an SW banyan G' with L' levels is the interconnection graph of a synthesized SW banyan G with L levels and if $0 \leq I \leq L'$, then

1) $L = L'+1$,

2) the bases of the component crossbars in level I of G' are the vertices of level I of G, and

3) the apexes of the component crossbars in level I of G' are the vertices of level $I+1$ of G.

Proof. Every crossbar is a 1-level banyan by definition.

Now suppose that G is a synthesized SW banyan with interconnection graph G'. Suppose further that G' is an L'-level banyan. For each $I = 0 \sim L'$, let $SA[I]$ be the set of all apexes and let $SB[I]$ be the set

of all bases of the component crossbars in level I of G'. Thus, $\underline{SB}[I] = \underline{SA}[I-1]$ for all $I = 1 \sim L'$, and the vertices of G' can be partitioned into $L+2$ subsets as follows.

$$\underline{SV}[0] = \underline{SB}[0]$$

$$\underline{SV}[I] = \underline{SB}[I] = \underline{SA}[I-1] \qquad (I = 1 \sim L)$$

$$\underline{SV}[L+1] = \underline{SA}[L]$$

Since the arcs of G are those of its component crossbars, each arc in G is from some vertex in $\underline{SV}[I]$ to some vertex $\underline{SV}[I+1]$ for some $0 \leq I \leq L$. Thus, G is an $(L+1)$-level banyan and $\underline{SV}[0],\ldots,\underline{SV}[L+1]$ as defined above are its levels.

Q.E.D.

Corollary 3.1.3a. Let G be a synthesized SW banyan with interconnection graph G', and let V be either a base or an intermediate of G. Then there exists a unique component crossbar V' in level $\square V$ of G' such that V is a base of V'.

Proof. By part 2 of Theorem 3.1.3, there exists a component crossbar V' in level $\square V$ of G' such that V is a base of V'. There cannot exist more than one such component crossbar, since by part 2 of Definition 1.2.1, V cannot be a base of more than one component banyan.

Q.E.D.

Corollary 3.1.3b. Let G be a synthesized SW banyan with interconnection graph G', and let V be either an apex or an intermediate of G. Then there exists a unique component crossbar V' in level $(\square V)-1$ of G' such that V is an apex of V'.

Proof. By part 3 of Theorem 3.1.3, there exists a component crossbar V' in level $(\square V)-1$ of G' such that V is a base of V'. There cannot exist

more than one such component crossbar, since by part 3 of Definition 1.2.1, V cannot be an apex of more than one component banyan.

Q.E.D.

Theorem 3.1.4. Let G be a synthesized SW banyan, and suppose that its interconnection graph G' is uniform with L' levels, fanout vector $\underline{F}'$, and spread vector $\underline{S}'$. Suppose also that each apex of G' has A apexes and each base of G' has B bases. Then G is uniform with fanout vector

$$B,\underline{F}'$$

and spread vector

$$\underline{S}',A.$$

Proof. By Theorem 3.1.3, each vertex in level 1 of G is an apex of a base of G'. But each base of G' is a crossbar with B bases, and hence, has fanout B by Theorem 3.1.1. Therefore, each vertex in level 1 of G has B arcs incident into it.

Now let I be a level number such that $2 \leq I \leq L$, where L is the number of levels in G. By Theorem 3.1.3, each vertex in level I of G is an apex of a component crossbar in level I-1 of G'. But each component crossbar in level I-1 of G' has $\underline{F}[I-1]$ bases by Theorem 1.2.1, and hence, has fanout $\underline{F}[I-1]$ by Theorem 3.1.1. Therefore, each vertex in level I of G has $\underline{F}[I-1]$ arcs incident into it.

Similarly, by Theorem 3.1.3, each vertex in level L-1 of G is a base of an apex of G'. But each apex of G' is a crossbar with A apexes, and, hence, has spread A by Theorem 3.1.1. Therefore, each vertex in level L-1 of G has A arcs incident out from it.

Now let J be a level number such that $0 \leq J \leq L-2$. By Theorem 3.1.3, each vertex in level J of G is a base of a component crossbar in level J of G'. But each component crossbar in level J of G' has $\underline{S}[J+1]$ apexes by

Theorem 1.2.1, and hence, has spread $\underline{S}[J+1]$ by Theorem 3.1.1. Therefore, each vertex in level J of G has $\underline{S}[J+1]$ arcs incident out from it.

Thus, G is uniform with fanout vector $B,\underline{F}$ and spread vector $\underline{S},A$.

Q.E.D.

Theorem 3.1.5. If a synthesized SW banyan is uniform with fanout vector $\underline{F}$ and spread vector $\underline{S}$, then its interconnection graph is uniform with fanout vector

$$1\downarrow\underline{F}$$

and spread vector

$$(-1)\downarrow\underline{S}.$$

Proof. Let G be a uniform synthesized SW banyan with fanout vector $\underline{F}$, spread vector $\underline{S}$, and L levels. Let G' be the interconnection graph of G. By Theorem 3.1.3, G' has $L-1$ levels.

Let V' be any component crossbar in level I of G' such that $1 \leq I \leq L-1$. By Theorem 3.1.3, the apexes of V' are in level $I+1$ of G. Thus, the fanout of V' is $\underline{F}[I+1]$. By Theorem 3.1.1, V' has $\underline{F}[I+1]$ bases; so by Theorem 1.2.1, V' has $\underline{F}[I+1]$ arcs incident into it in G'.

Similarly, let W' be any component crossbar in level J of G' such that $0 \leq J \leq L-2$. By Theorem 3.1.3, the bases of W' are in level J of G. Thus, the spread of W' is $\underline{S}[J+1]$. By Theorem 3.1.1, W' has $\underline{S}[J+1]$ apexes; so by Theorem 1.2.1, W' has $\underline{S}[J+1]$ arcs incident out from it in G'.

Therefore, G' is uniform with fanout vector $1\downarrow\underline{F}$ and spread vector $(-1)\downarrow\underline{S}$.

Q.E.D.

Theorem 3.1.6. Let $B1$ and $B2$ be two distinct bases of a synthesized SW banyan G with interconnection graph G'. Let $B1'$ and $B2'$ be the bases of G' that contain $B1$ and $B2$ respectively as bases. Then $(B1'\boxtimes B2') = (B1\boxtimes B2)-1$.

Proof. Let $I = B1\boxtimes B2$. Then there exists a vertex V in level I of G such that $B1 \boxleft V$ and $B2 \boxleft V$. Let V' be the component crossbar in level I-1 of G' that contains V as an apex. By Corollary 1.2.2a, $B1' \boxleft V'$ and $B2' \boxleft V'$. Therefore, $(B1'\boxtimes B2') \leq I-1$.

Now let $J = B1'\boxtimes B2'$. Then there exists a component crossbar W' in level J of G' such that $B1' \boxleft W'$ and $B2' \boxleft W'$. Let W be an apex of W'. By Theorem 3.1.3, W is in level J+1 of G. By Corollary 1.2.2a, $B1 \boxleft W$ and $B2 \boxleft W$. Therefore, $(B1\boxtimes B2) \leq J+1$ and hence, $(B1'\boxtimes B2') \geq (B1\boxtimes B2)-1$.

Thus, $(B1'\boxtimes B2') = (B1\boxtimes B2)-1$.

Q.E.D.

Theorem 3.1.7. Let $A1$ and $A2$ be two distinct apexes of a synthesized SW banyan G with interconnection graph G'. Let $A1'$ and $A2'$ be the apexes of G' that contain $A1$ and $A2$ respectively as apexes. Then $(A1'\boxbar A2') = (A1\boxbar A2)-1$.

Proof. Let $I = A1\boxbar A2$ and let L be the number of levels in G. Then there exists a vertex V in level L-I of G such that $V \boxleft A1$ and $V \boxleft A2$. Let V' be the component crossbar in level L-I of G' that contains V as a base. By Corollary 1.2.2a, $V' \boxleft A1'$ and $V' \boxleft A2'$. Therefore, $(A1'\boxbar A2') \leq L'-(L-I)$ where L' is the number of levels in G'. But by Theorem 3.1.3, $L' = L-1$. Thus,

$$(A1'\boxbar A2') \leq (L-1)-(L-I)$$

$$(A1'\boxbar A2') \leq I-1.$$

Now let $J = A1' \boxtimes A2'$. Then there exists a component crossbar W' in level $L'-J$ of G' such that $W' \boxtimes A1'$ and $W' \boxtimes A2'$. Let W be a base of W'. By Theorem 3.1.3, W is in level $L'-J$ of G. By Corollary 1.2.2a, $W \boxtimes A1$ and $W \boxtimes A2$. Therefore,

$$(A1 \boxtimes A2) \leq L-(L'-J)$$

$$(A1 \boxtimes A2) \leq L-((L-1)-J)$$

$$(A1 \boxtimes A2) \leq 1+J$$

$$(A1' \boxtimes A2') \geq (A1 \boxtimes A2)-1$$

Thus, $(A1' \boxtimes A2') = (A1 \boxtimes A2)-1$.

Q.E.D.

B.3.2 Distance Properties

In this section, useful and mathematically interesting properties of an SW banyan's base and apex distance functions will be derived formally. The properties derived here include those discussed in Section 6.2.

Theorem 3.2.1. Let $B1$ and $B2$ be any two bases of an SW banyan G, and let V be a vertex in some level I of G. If $(B1 \boxtimes B2) \leq I$ then $B1 \boxleft V$ iff $B2 \boxleft V$.

Proof. Let L be the number of levels in G and suppose that $(B1 \boxtimes B2) \leq I$.

Suppose first that G is a crossbar. If $B1 = B2$, then obviously $B1 \boxleft V$ iff $B2 \boxleft V$. If $B1 \neq B2$, then V is an apex of G, in which case $B1 \boxleft V$ and $B2 \boxleft V$. Thus, in either case, $B1 \boxleft V$ iff $B2 \boxleft V$.

Suppose next that G is a synthesized SW banyan with interconnection graph G', and suppose that Theorem 3.2.1 holds for G'. If $I=0$, then $B1 = B2$, in which case $B1 \boxleft V$ is equivalent to $B2 \boxleft V$ by substitution. Now suppose $I > 0$, and let V' be the component crossbar in level I-1 of G' which contains V as an apex. Also, let $B1'$ and $B2'$ be the bases of G' which contain $B1$ and $B2$ respectively as bases. Since $(B1 \boxtimes B2) \leq I$, it follows by Theorem 3.1.6 that $(B1' \boxtimes B2') \leq I$-1. Thus, by the supposition that Theorem 3.2.1 holds for G', $B1' \boxleft V'$ iff $B2' \boxleft V'$. But by Corollary 1.2.2a, $B1 \boxleft V$ iff $B1' \boxleft V'$ and $B2' \boxleft V'$ iff $B2 \boxleft V$. Therefore, $B1 \boxleft V$ iff $B2 \boxleft V$.

Thus, if $(B1 \boxtimes B2) \leq I$, then $B1 \boxleft V$ iff $B2 \boxleft V$.

Q.E.D.

Theorem 3.2.2. Let $A1$ and $A2$ be any two apexes of an SW banyan G with L levels, and let V be a vertex in some level I of G. If $(A1 \boxbslash A2) \leq L$-1 then $V \boxleft A1$ iff $V \boxleft A2$.

Proof. Suppose that $(A1 \overline{\vee} A2) \leq L\text{-}I$.

Suppose first that G is a crossbar. If $A1 = A2$, then obviously $V \trianglelefteq A1$ iff $V \trianglelefteq A2$. If $A1 \neq A2$, then V is a base of G, in which case $V \trianglelefteq A1$ and $V \trianglelefteq A2$. Thus, in either case, $V \trianglelefteq A1$ iff $V \trianglelefteq A2$.

Suppose next that G is a synthesized SW banyan with interconnection graph G', and suppose that Theorem 3.2.2 holds for G'. If $I = 0$, then $A1 = A2$, in which case $V \trianglelefteq A1$ is equivalent to $V \trianglelefteq A2$ by substitution. Now suppose $I > 0$, and let V' be the component crossbar in level I of G' which contains V as a base. Also, let $A1'$ and $A2'$ be the apexes of G' which contain $A1$ and $A2$ respectively as apexes. Since $(A1 \overline{\vee} A2) \leq L\text{-}I$, it follows by Theorem 3.1.7 that

$$(A1' \overline{\vee} A2') \leq (L\text{-}I)\text{-}1$$

$$(A1' \overline{\vee} A2') \leq (L\text{-}1)\text{-}1.$$

But L-1 is the number of levels in G'. Thus, by the supposition that Theorem 3.2.2 holds for G', $V' \trianglelefteq A1'$ iff $V' \trianglelefteq A2'$. But by Corollary 1.2.2a, $V \trianglelefteq A1$ iff $V' \trianglelefteq A1'$ and $V' \trianglelefteq A2'$ iff $V \trianglelefteq A2$. Therefore, $V \trianglelefteq A1$ iff $V \trianglelefteq A2$.

Thus, if $(A1 \overline{\vee} A2) \leq L\text{-}I$, then $V \trianglelefteq A1$ iff $V \trianglelefteq A2$.

Q.E.D.

Definition 3.2.1. For any real number X we use $\overline{\wedge}_X$ to denote the relational operation defined by

$$B1 \overline{\wedge}_X B2 \leftrightarrow (B1 \overline{\wedge} B2) \leq X,$$

where $B1$ and $B2$ are arbitrary bases of an SW banyan.

Definition 3.2.2. For any real number X we use $\overline{\vee}_X$ to denote the relational operation defined by

$$A1 \overline{\vee}_X A2 \leftrightarrow (A1 \overline{\vee} B2) \leq X,$$

where $A1$ and $A2$ are arbitrary bases of an SW banyan.

Theorem 3.2.3. For any real number X, the relation $\boxtimes_X$ is both transitive and symmetric.

Proof. Let $B1$, $B2$, and $B3$ be bases of an SW banyan G with L levels, and suppose that $B1 \boxtimes_X B2$ and $B2 \boxtimes_X B3$. Then there exists an integer I such that $(B1 \boxtimes B2) \leq I$, $(B2 \boxtimes B3) \leq I$, $I \leq X$, and $I \leq L$. Since $(B1 \boxtimes B2) \leq I$, it follows that $I \leq X$ and $I \leq L$. Since $(B1 \boxtimes B2) \leq I$, there exists a vertex V in level I of G such that $B1 \trianglelefteq V$ and $B2 \trianglelefteq V$. By Theorem 3.2.1, $B3 \trianglelefteq V$. Since $B1 \trianglelefteq V$ and $B3 \trianglelefteq V$, it follows that $(B1 \boxtimes B3) \leq I$. But $I \leq X$, so $B1 \boxtimes_X B2$. Thus, $\boxtimes_X$ is transitive.

Symmetry of $\boxtimes_X$ follows from Definition 3.2.1 and Theorem 2.1.1.

Q.E.D.

Corollary 3.2.3a. If X is a non-negative real number, then $\boxtimes_X$ is an equivalence relation.

Proof. Let B be a base. By Theorem 2.1.3, $0 = B \boxtimes B$. Since X is non-negative, $(B \boxtimes B) \leq X$. Thus, $B \boxtimes_X B$, and hence, $\boxtimes_X$ is reflexive. $\boxtimes_X$ is transitive and symmetric by Theorem 3.2.3. Therefore, $\boxtimes_X$ is an equivalence relation.

Q.E.D.

Corallary 3.2.3b. The base distance operation $\boxtimes$ of an SW banyan G is a metric on the bases of G.

Proof. Let $B1$, $B2$, and $B3$ be bases of G. By Theorem 2.1.2, $0 \leq B1 \boxtimes B2$. By Theorem 2.1.3, $0 = B1 \boxtimes B2$ iff $B1 = B2$. By Theorem 2.1.1, $(B1 \boxtimes B2) = B2 \boxtimes B1$.

Now let $X = (B1 \boxtimes B2) \lceil (B2 \boxtimes B3)$. Thus, $B1 \boxtimes_X B2$ and $B2 \boxtimes_X B3$. By Theorem 3.2.3, $(B1 \boxtimes B3) \leq X$. But since base distances are non-negative, $X \leq (B1 \boxtimes B2) + (B2 \boxtimes B3)$. Therefore, $(B1 \boxtimes B3) \leq (B1 \boxtimes B2) + (B2 \boxtimes B3)$.

Thus, $\boxtimes$ is a metric on the bases of G.

Q.E.D.

Theorem 3.2.4. For any real number X, the relation $\mathbb{V}_X$ is both transitive and symmetric.

Proof. Let $A1$, $A2$, and $A3$ be apexes of an SW banyan G with L levels, and suppose that $A1\ \mathbb{V}_X\ A2$ and $A2\ \mathbb{V}_X\ A3$. Then there exists an integer I such that $(A1\mathbb{V}A2) \leq I$, $(A2\mathbb{V}A3) \leq I$, $I \leq X$, and $0 \leq L\text{-}I$. Since $(A1\mathbb{V}A3) \leq I$, there exists a vertex V in level I of G such that $V \trianglelefteq A1$ and $V \trianglelefteq A2$. By Theorem 3.2.2, $V \trianglelefteq A3$. Since $V \trianglelefteq A1$ and $V \trianglelefteq A3$, it follows that $(A1\mathbb{V}A3) \leq I$. But $I \leq X$, so $A1\ \mathbb{V}_X\ A2$. Thus, $\mathbb{V}_X$ is transitive.

Symmetry of $\mathbb{V}_X$ follows from Definition 3.2.2 and Theorem 2.1.1.

Q.E.D.

Corollary 3.2.4a. If X is a non-negative real number, then $\mathbb{V}_X$ is an equivalence relation.

Proof. Let A be an apex. By Theorem 2.1.3, $0 = A\mathbb{V}A$. Since X is non-negative, $(A\mathbb{V}A) \leq X$. Thus, $A\ \mathbb{V}_X\ A$, and hence, $\mathbb{V}_X$ is reflexive. $\mathbb{V}_X$ is transitive and symmetric by Theorem 3.2.4. Therefore, $\mathbb{V}_X$ is an equivalence relation.

Q.E.D.

Corollary 3.2.4b. The apex distance operation $\mathbb{V}$ of an SW banyan G is a metric on the apexes of G.

Proof. Let $A1$, $A2$, and $A3$ be apexes of G. By Theorem 2.1.2, $0 \leq A1\mathbb{V}A2$. By Theorem 2.1.3, $0 = A1\mathbb{V}A2$ iff $A1 = A2$. By Theorem 2.1.2, $(A1\mathbb{V}A2) = A2\mathbb{V}A1$.

Now let $X = (A1\mathbb{V}A2)\lceil(A2\mathbb{V}A3)$. Thus, $A1\ \mathbb{V}_X\ A2$ and $A2\ \mathbb{V}_X\ A3$. By Theorem 3.2.4, $(A1\mathbb{V}A2) \leq X$. But since apex distances are non-negative, $X \leq (A1\mathbb{V}A2)+(A2\mathbb{V}A3)$. Therefore, $(A1\mathbb{V}A2) \leq (A1\mathbb{V}A2)+(A2\mathbb{V}A3)$.

Thus, $\mathbb{V}$ is a metric on the bases of G.

Q.E.D.

Theorem 3.2.5. Let X and Y be real numbers such that $0 \leq X \leq Y$. Then the equivalence classes of relation $\boxed{\Delta}_X$ are subsets of those of relation $\boxed{\Delta}_Y$.

Proof. Let SBX be one of the equivalence classes determined by the relation $\boxed{\Delta}_X$, and let SBY be an equivalence class of the relation $\boxed{\Delta}_Y$ such that $\emptyset \neq SBX \cap SBY$. Let $BC \in SBX \cap SBY$ and let $BX \in SBX$. Since both BC and BX are in SBX, $(BC\boxed{\Delta}BX) \leq X$. But $X \leq Y$, so $(BC\boxed{\Delta}BX) \leq Y$. Since $BC \in SBY$, this implies that $BX \in SBY$. Therefore, $SBX \subseteq SBY$.

Q.E.D.

Theorem 3.2.6. Let X and Y be real numbers such that $0 \leq X \leq Y$. Then the equivalence classes of the relation $\boxed{\nabla}_X$ are subsets of those of relation $\boxed{\nabla}_Y$.

Proof. Let SAX be one of the equivalence classes determined by the relation $\boxed{\nabla}_X$, and let SAY be an equivalence class of the relation $\boxed{\nabla}_Y$ such that $\emptyset \neq SAX \cap SAY$. Let $AC \in SAX \cap SAY$ and let $AX \in SAX$. Since both AC and AX are in SAX, $(AC\boxed{\nabla}AX) \leq X$. But $X \leq Y$, so $(AC\boxed{\nabla}AX) \leq Y$. Since $AC \in SAY$, this implies that $AX \in SAY$. Therefore, $SAX \subseteq SAY$.

Q.E.D.

Theorem 3.2.7. Let G be a uniform SW banyan with L levels and fanout vector F, and let I be an integer such that $0 \leq I \leq L$. The relation $\boxed{\Delta}_I$ partitions the bases of G into $\times/I\uparrow\underline{F}$ equivalence classes containing $\times/I\uparrow\underline{F}$ bases each.

Proof. Let $SB1$ be one of the equivalence classes of the relation $\boxed{\Delta}_I$. Also, let V be a vertex in level I such that V is above some base in $SB1$, and let $SB2$ be the set of all bases below V. By Corollary 2.2.2a, $SB2$ contains $\times/I\uparrow\underline{F}$ elements. By Theorem 3.2.1, every base in $SB1$ is

below V, so $SB1 \subseteq SB2$. Now let $B2$ be an arbitrary element of $SB2$ and let $B1 \in SB1$. Both $B1$ and $B2$ are elements of $SB2$, and hence, are below V. Therefore, $(B1 \boxtimes B2) \leq I$. Since $B1 \in SB1$ this implies that $B2 \in SB1$. Thus, $SB2 \subseteq SB1$, $SB1 = SB2$, and $SB1$ contains $\times/I\uparrow\underline{F}$ elements. Therefore, each equivalence class of $\boxtimes_I$ contains $\times/I\uparrow\underline{F}$ bases.

By Corollary 2.2.1a, there are $\times/\underline{F}$ bases total. Since the equivalence classes of $\boxtimes_I$ form a partitioning of these bases, the number of equivalence classes is

$$(\times/\underline{F}) \div \times/I\uparrow\underline{F}$$

$$= \times/I\downarrow\underline{F}.$$

Q.E.D.

Corollary 3.2.7a. Let G be a uniform SW banyan with L levels and fanout vector $\underline{F}$, and let I be an integer such that $1 \leq I \leq L$. Then the relation $\boxtimes_{I-1}$ has $\underline{F}[I]$ equivalence classes that are subsets of any given equivalence class of $\boxtimes_I$.

Proof. Let SB be an equivalence class of $\boxtimes_I$. By Theorem 3.2.7, SB contains $\times/I\uparrow\underline{F}$ bases. It is apparent from Theorem 3.2.5 that the equivalence classes of $\boxtimes_{I-1}$ that are subsets of SB form a partitioning of SB. But by Theorem 3.2.7, each of these subsets has $\times/(I-1)\uparrow\underline{F}$ elements. Therefore, the number of equivalence classes of $\boxtimes_{I-1}$ that are subsets of SB is

$$(\times/I\uparrow\underline{F}) \div \times/(I-1)\uparrow\underline{F}$$

$$= \underline{F}[I].$$

Q.E.D.

Theorem 3.2.8. Let G be a uniform SW banyan with L levels and fanout vector $\underline{F}$, and let I be an integer such that $0 \leq I \leq L$. The relation $\overline{\boxtimes}_I$ partitions the apexes of G into $\times/(-I)\downarrow\underline{S}$ equivalence classes containing

$\times/(-I)\uparrow\underline{S}$ apexes each.

Proof. Let $SA1$ be one of the equivalence classes of the relation $\overline{\mathbb{V}}_I$. Also let V be a vertex in level $L\text{-}I$ such that V is below some base in $SA1$, and let $SA2$ be the set of all apexes above V. By Corollary 2.2.2a, the number of elements in $SA2$ is

$$\times/(L\text{-}I)\downarrow\underline{S}$$

$$= \times/(-I)\uparrow\underline{S}.$$

By Theorem 3.2.2, every apex in $SA1$ is above V, so $SA1 \subseteq SA2$. Now let $A2$ be an arbitrary element of $SA2$ and let $A1 \in SA1$. Both $A1$ and $A2$ are elements of $SA2$ and, hence, are above V. Therefore, $(A1\overline{\mathbb{V}}A2) \leq (L\text{-}(L\text{-}I)) = I$. Since $A1 \in SA1$, this implies that $A2 \in SA1$. Thus, $SA2 \subseteq SA1$, $SA1 = SA2$, and $SA1$ contains $\times/(-I)\uparrow\underline{S}$ elements. Therefore, each equivalence class of $\overline{\mathbb{V}}_I$ contains $\times/(-I)\uparrow\underline{S}$ apexes.

By Corollary 2.2.1a, there are $\times/\underline{S}$ apexes total. Since the equivalence classes of $\overline{\mathbb{V}}_I$ form a partitioning of these apexes, the number of equivalence classes is

$$(\times/\underline{S})\div \times/(-I)\uparrow\underline{S}$$

$$= \times/(-I)\downarrow\underline{S}.$$

Q.E.D.

Corollary 3.2.8a. Let G be a uniform SW banyan with L levels, with fanout vector $\underline{F}$, and with spread vector $\underline{S}$; and let I be an integer such that $1 \leq I \leq L$. Then the relation $\overline{\mathbb{V}}_{I-1}$ has $\underline{S}[L\text{-}(I\text{-}1)]$ equivalence classes that are subsets of any given equivalence class of $\overline{\mathbb{V}}_I$.

Proof. Let SA be an equivalence class of $\overline{\mathbb{V}}_I$. By Theorem 3.2.8, SA contains $\times/(-I)\uparrow\underline{S}$ apexes. It is apparent from Theorem 3.2.6 that the equivalence classes of $\overline{\mathbb{V}}_{I-1}$ that are subsets of SA form a partitioning

of *SA*. But by Theorem 3.2.8, each of these subsets has $\times/(-(I-1))\uparrow\underline{S}$ elements. Therefore, the number of equivalence classes of $\boxtimes_{I-1}$ that are subsets of *SA* is

$$(\times/(-I)\uparrow\underline{S})\div\times/(-(I-1))\uparrow\underline{S}$$

$$= \underline{S}[(L-(I-1))\sim L]\div\underline{S}[(L-(I-2))\sim L]$$

$$= \underline{S}[L-(I-1)].$$

Q.E.D.

B.3.3 Connectability

The next theorem is a strengthened version of Theorem 2.1.7 applicable to SW banyans.

Theorem 3.3.1. Let $SA1$ and $SA2$ be sets of apexes and let $SB1$ and $SB2$ be subsystems of an SW banyan with L levels. Let $SC1 = SB1 \times SA1$ and let $SC2 = SB2 \times SA2$. Then the callsets $SC1$ and $SC2$ conflict with each other iff

$$L \geq (SB1 \boxtimes SB2) + SA1 \boxtimes SA2.$$

Proof. If $L < (SB1 \boxtimes SB2) + SA1 \boxtimes SA2$ then $SC1$ does not conflict with $SC2$ by Theorem 7.

Now suppose that $L \geq (SB1 \boxtimes SB2) + SA1 \boxtimes SA2$. Then there exists a level number I such that $(SB1 \boxtimes SB2) \leq I \leq L-(SA1 \boxtimes SA2)$. Let $B1$, $B2$, $A1$, and $A2$ be elements of $SB1$, $SB2$, $SA1$, and $SA2$ respectively such that $(B1 \boxtimes B2) = SB1 \boxtimes SB2$ and $(A1 \boxtimes A2) = SA1 \boxtimes SB2$. Thus, $(B1 \boxtimes B2) \leq I$ and $(A1 \boxtimes A2) \leq L-I$. Let $C1$ be the call from $B1$ to $A1$ and let $C2$ be the call from $B2$ to $A2$. There exists a vertex V in level I such that V lies along the path from $B1$ to $A1$. Thus, $B1 \trianglelefteq V$ and $V \trianglelefteq A1$. But by Theorems 3.2.1 and 3.2.2 respectively, $B2 \trianglelefteq V$ and $V \trianglelefteq A2$, implying that V lies along the path from $B2$ to $A2$. Since V is common to both paths, $C1$ conflicts with $C2$. But $C1 \in CS1$ and $C2 \in CS2$, so $SC1$ conflicts with $SC2$.

Q.E.D.

B.4 CC Banyans

The class of rectangular banyans called CC banyans, discussed in Section 7, will be defined and analyzed formally in this section.

B.4.1 Structure

In this section, we will define CC banyans, characterize the existence of paths in them, and show that they are indeed rectangular banyans.

Definition 4.1.1. Let L be a positive integer, let $\underline{S}[1\sim L]$ be a vector of positive integers, and let $N = \times/\underline{S}$. Also let $\underline{V}[0\sim L;0\sim N-1]$ be the vertices of a graph G such that for any two vertices $\underline{V}[I1;J1]$ and $\underline{V}[I2;J2]$, there exists an arc from $\underline{V}[I1;J1]$ to $\underline{V}[I2;J2]$ iff both $I2 = I1+1$ and $J2 = N|J1+M\times(\times/I1\uparrow\underline{S})$ for some $M = 0\sim\underline{S}[I2]-1$. Any such graph G is called a CC banyan.

Throughout this section, it will be understood that G, L, $\underline{S}$, N, and $\underline{V}$ are as defined above. The particular CC banyan to which these quantities refer will be clear from context, because we will not deal with more than one CC banyan at a time. It will also be understood that $\underline{B}[0\sim N-1] = \underline{V}[0;0\sim N-1]$ and $\underline{A}[0\sim N-1] = \underline{V}[L;0\sim N-1]$. The modulo N arithmetic operators defined below will be used frequently in this section.

$$X\oplus Y \leftrightarrow N|X+Y$$

$$\ominus X \leftrightarrow N|-X$$

$$X\ominus Y \leftrightarrow X\oplus\ominus Y$$

Lemma 4.1.1. Every CC banyan is the graph of a partial order.

Proof. Since each arc of G is from some vertex $\underline{V}[I;J1]$ to a vertex $\underline{V}[I+1;J2]$, G cannot contain a circuit and, hence, is the graph of a partial order.

Q.E.D.

Theorem 4.1.2. Let $\underline{V}[I1;J1]$ and $\underline{V}[I2;J2]$ be arbitrary vertices of a CC banyan G. If $I1 < I2$ and $(J2 \ominus J1) < \times/I2\uparrow\underline{S}$ and $0 = (\times/I1\uparrow\underline{S})\mid J2-J1$, then G contains exactly one path from $\underline{V}[I1;J1]$ to $\underline{V}[I2;J2]$. Otherwise, G contains no path from $\underline{V}[I1;J1]$ to $\underline{V}[I2;J2]$.

Proof. First suppose that $I1 < I2$. Then it is apparent from Definition 4.1.1 that any path from $\underline{V}[I1;J1]$ to $\underline{V}[I2;J2]$ must contain exactly $I2-I1$ arcs. It is also apparent that any $(I2-I1)$-arc path from $\underline{V}[I1;J1]$ must pass through the vertex sequence

$$\underline{V}[I1;J1],$$

$$\underline{V}[I1+1;J1\oplus(\times/I1\uparrow\underline{S})\times\underline{M}[1]],$$

$$\underline{V}[I1+2;J1\oplus(\times/I1\uparrow\underline{S})\times\underline{M}[1]\oplus\underline{M}[2]\times\underline{S}[I1+1]],$$

$$\underline{V}[I1+3;J1\oplus(\times/I1\uparrow\underline{S})\times\underline{M}[1]\oplus(\underline{M}[2]\times\underline{S}[I1+1])\oplus\underline{M}[3]\times\underline{S}[I1+1]\times\underline{S}[I1+2]],$$

$$\ldots,$$

$$\underline{V}[I1+K;J1\oplus(\times/I1\uparrow\underline{S})\times(I1\downarrow\underline{S})\circledast K\uparrow\underline{M}],$$

$$\ldots,$$

$$\underline{V}[I2;J1\oplus(\times/I1\uparrow\underline{S})\times(I1\downarrow\underline{S})\circledast(I2-I1)\uparrow\underline{M}]$$

for some vector of integers $\underline{M}[1\sim I2-I1]$ where for each K, $0 \leq \underline{M}[K] < \underline{S}[I1+K]$. Since $((I2-I1)\uparrow\underline{M}) = \underline{M}$, the last vertex in this sequence is simply $\underline{V}[I2;J1\oplus(\times/I1\uparrow\underline{S})\times(I1\downarrow\underline{S})\circledast\underline{M}]$. But since $0 \leq \underline{M}[K] < \underline{S}[I1+K]$, there are $\times/I1\downarrow I2\uparrow\underline{S}$ possible values of $\underline{M}$, and the possible values of $(I1\downarrow\underline{S})\circledast\underline{M}$ are $0 \sim (\times/I1\downarrow I2\uparrow\underline{S})-1$. Thus, there are $\times/I1\downarrow I2\uparrow\underline{S}$ paths of length $I2-I1$ from vertex $\underline{V}[I1;J1]$, and the terminal vertices of these paths are

$\underline{V}[I2;J1]$, $\underline{V}[I2;J1\oplus(\times/I1\uparrow\underline{S})]$, $\underline{V}[I2;J1\oplus(\times/I1\uparrow\underline{S})\times 2]$,..., $\underline{V}[I2;J1\oplus(\times/I1\uparrow\underline{S})\times(\times/I1\downarrow I2\uparrow\underline{S})-1]$. But $((\times/I1\uparrow\underline{S})\times(\times/I1\downarrow I2\uparrow\underline{S})-1) = (\times/I2\uparrow\underline{S})-(\times/I1\uparrow\underline{S})$, so this set of terminal vertices is precisely $[\ \underline{V}[I2;X] : (0 = (\times/I1\uparrow\underline{S})|X\ominus J1) \wedge (X\ominus J1) < \times/I2\uparrow\underline{S}\ ]$. This set equals $[\ \underline{V}[I2;X] : (0 = (\times/I1\uparrow\underline{S})|X-J1) \wedge (X\ominus J1) < \times/I2\uparrow\underline{S}]$, however, because $0 = (\times/I1\uparrow\underline{S})|N$ and hence $((\times/I1\uparrow\underline{S})|X\ominus J1) = ((\times/I1\uparrow\underline{S})|N|X-J1) = (\times/I1\uparrow\underline{S})|X-J1$. Thus, if there exists a path from $\underline{V}[I1;J1]$ to $\underline{V}[I2;J2]$, then $0 = (\times/I1\uparrow\underline{S})|J2-J1$ and $(J2\ominus J1) < \times/I2\uparrow\underline{S}$. Likewise, if $0 = (\times/I1\uparrow\underline{S})|J2-J1$ and $(J2\ominus J1) < \times/I2\uparrow\underline{S}$, then there exists a path from $\underline{V}[I1;J1]$ to $\underline{V}[I2;J2]$. Further, since $((\times/I1\uparrow\underline{S})\times(\times/I1\downarrow I2\uparrow\underline{S})-1) = (\times/I2\uparrow\underline{S})-(\times/I1\uparrow\underline{S}) < \times/\underline{S} = N$, the terminal vertices described above are all distinct; so there can be at most one path from $\underline{V}[I1;J1]$ to $\underline{V}[I2;J2]$.

Suppose, on the other hand, that $I1 \geq I2$. Then it is obvious from Definition 4.1.1 that G contains no path from $\underline{V}[I1;J1]$ to $\underline{V}[I2;J2]$.

Q.E.D.

Corollary 4.1.2a. Let $\underline{V}[I1;J1]$ and $\underline{V}[I2;J2]$ be arbitrary vertices of a CC banyan G, and let $\trianglelefteq$ be the partial order associated with G. Then $\underline{V}[I1;J1] \trianglelefteq \underline{V}[I2;J2]$ if $I1 \leq I2$ and $(J2\ominus J1) < \times/I2\uparrow\underline{S}$ and $0 = (\times/I1\uparrow\underline{S})|J2-J1$.

Proof. First suppose that $I1 \leq I2$ and $(J2\ominus J1) < \times/I2\uparrow\underline{S}$ and $0 = (\times/I1\uparrow\underline{S})|J2-J1$. If $I1 < I2$ then, by Theorem 4.1.2, there is a path from $\underline{V}[I1;J1]$ to $\underline{V}[I2;J2]$, implying that $\underline{V}[I1;J1] \trianglelefteq \underline{V}[I2;J2]$. If, on the other hand, $I1 = I2$ then $0 = (\times/I2\uparrow\underline{S})|J2-J1$. Since $(\times/I2\uparrow\underline{S})|N$,

$$0 = (\times/I2\uparrow\underline{S})|N|J2-J1$$
$$= (\times/I2\uparrow\underline{S})|J2\ominus J1.$$

But,

$$(J2\ominus J1) < \times/I2\uparrow\underline{S}.$$

Therefore,

$$(J2\ominus J1) = 0$$
$$J1 = J2$$

Now, suppose that $\underline{V}[I1;J1] \trianglelefteq \underline{V}[I2;J2]$. Then either $\underline{V}[I1;J1] = \underline{V}[I2;J2]$ or there exists a path from $\underline{V}[I1;J1]$ to $\underline{V}[I2;J2]$. If there exists a path from $\underline{V}[I1;J1]$ to $\underline{V}[I2;J2]$, then it follows from Theorem 4.1.2 that $I1 \leq I2$ and $(J2 \ominus J1) < \times/I2\uparrow\underline{S}$ and $0 = (\times/I1\uparrow\underline{S})|J2-J1$. If, on the other hand, $\underline{V}[I1;J1] = \underline{V}[I2;J2]$, then $I1 = I2 \leq I2$ and $(J2 \ominus J1) = (J2 \ominus J2) = 0 < \times/I2\uparrow\underline{S}$ and $0 = ((\times/I1\uparrow\underline{S})|0) = ((\times/I1\uparrow\underline{S})|J2-J2) = (\times/I1\uparrow\underline{S})|J2-J1$.

Q.E.D.

Corollary 4.1.2b. Let $\underline{B}[J1]$ be a base and let $\underline{V}[I;J2]$ be a vertex of a CC banyan G with partial order $\trianglelefteq$. Then $\underline{B}[J1] \trianglelefteq \underline{V}[I;J2]$ iff $(J2 \ominus J1) < \times/I\uparrow\underline{S}$.

Proof. If $\underline{B}[J1] \trianglelefteq \underline{V}[I,J2]$ then it follows immediately from Corollary 4.1.2a that $(J2 \ominus J1) < \times/I\uparrow\underline{S}$.

Now, suppose that $(J2 \ominus J1) < \times/I\uparrow\underline{S}$. Clearly, $0 \leq I$. Also, $0 = (1|J2-J1) = (\times/0\uparrow\underline{S})|J2-J1$. Since $\underline{B}[J1] = \underline{V}[0;J1]$, it follows from Corollary 4.1.2a that $\underline{B}[J1] \trianglelefteq \underline{V}[I;J2]$.

Q.E.D.

Corollary 4.1.2c. Let $\underline{V}[I;J1]$ be a vertex and let $\underline{A}[J2]$ be an apex of a CC banyan G with partial order $\trianglelefteq$. Then $\underline{V}[I;J1] \trianglelefteq \underline{A}[J2]$ iff $0 = (\times/I\uparrow\underline{S})|J2-J1$.

Proof. If $\underline{V}[I;J1] \trianglelefteq \underline{A}[J2]$ then it follows immediately from Corollary 4.1.2a that $0 = (\times/I\uparrow\underline{S})|J2-J1$.

Now suppose that $0 = (\times/I\uparrow\underline{S})|J2-J1$. Clearly $I \leq L$. Also, $(J2 \ominus J1) < N = (\times/\underline{S}) = \times/L\uparrow\underline{S}$. Since $\underline{A}[J2] = \underline{V}[L;J2]$, it follows from Corollary 4.1.2a that $\underline{V}[I;J1] \trianglelefteq \underline{A}[J2]$.

Q.E.D.

Theorem 4.1.3. A CC banyan G is a rectangular banyan with L levels and with fanout/spread vector $\underline{S}$. Also, $\underline{B}[0],\ldots,\underline{B}/N\text{-}1]$ are the bases of G, and $\underline{A}[0],\ldots,\underline{A}[N\text{-}1]$ are the apexes of G.

Proof. It is apparent from Definition 4.1.1 that $\underline{B}[0],\ldots,\underline{B}[N\text{-}1]$ have no arcs incident into them and, hence, are bases. Further, these are the only base of G, because for any $I = 1\sim L$ and any $J = 0\sim N\text{-}1$, vertex $\underline{V}[I;J]$ has an arc incident into it from $\underline{V}[I\text{-}1;J]$. Also it is obvious from Definition 4.1.1 that $\underline{A}[0],\ldots,\underline{A}[N\text{-}1]$ are precisely those vertices of G which have no arcs incident out from them, and, hence, are the apexes of G.

Next we will show that G is a banyan. By Lemma 4.1.1, G is the graph of a partial order. Next, consider an arbitrary base $\underline{V}[0;J1]$ and an arbitrary apex $\underline{V}[L;J2]$. Clearly, $0 < L$, and $(J2\ominus J1) < N = (\times/\underline{S}) = \times/L\uparrow\underline{S}$, and $0 = (1|J2\text{-}J1) = (\times/0\uparrow\underline{S})|J2\text{-}J1$. Thus, by Theorem 4.1.2, G contains exactly one path from $\underline{V}[0;J1]$ to $\underline{V}[L;J2]$. Therefore, G is a banyan.

It is apparent from Definition 4.1.1 that G has L levels and that for each $I = 0\sim L\text{-}1$, there are $\underline{S}[I+1]$ arcs incident out from any vertex $\underline{V}[I;J]$ in level I. Now consider the arcs incident into an arbitrary vertex $\underline{V}[I2;J2]$ in level $I2$ where $1 \le I2 \le L$. Let $\underline{V}[I1;J1]$ be the initial vertex of one such arc. Then by Definition 4.1.1, $I1 = I2\text{-}1$ and $J2 = J1\oplus M\times\times/I1\uparrow\underline{S}$ for some integer M where $0 \le M \le \underline{S}[I2]\text{-}1$. Therefore, $J1 = J2\ominus M\times\times/I1\uparrow S$. There are $\underline{S}[I2]$ possible values of M. Since $0 \le M \le \underline{S}[I1+1]\text{-}1$ and $0 \le I1 \le L\text{-}1$, it follows that $0 \le (M\times\times/I1\uparrow\underline{S}) < N$. Consequently, each possible value of M corresponds to a different possible value of J, implying that there are $\underline{S}[I2]$ arcs incident into $\underline{V}[I2;J2]$.

Therefore, G is rectangular with fanout vector $\underline{S}$ and with spread vector $\underline{S}$.

Q.E.D

B.4.2 Distance Properties

In this section, useful and mathematically interesting properties of a CC banyan's base and apex distance functions will be derived formally. The properties derived here include those discussed in Section 7.2.

Definition 4.2.1. The dyadic operator $\boxdot$, called minimum circular distance, is defined by $J1 \boxdot J2 \leftrightarrow (J2 \ominus J1) \lfloor (J1 \ominus J2)$ where $J1$ and $J2$ are integers.

Lemma 4.2.1. Let X, Y, Z, and M be integers. If $(X \ominus Y) \leq M$ and $(Z \ominus Y) \leq M$, then $(X \boxdot Z) \leq M$.

Proof. Suppose $(X \ominus Y) \leq M$ and $(Z \ominus Y) \leq M$. But $0 \leq (X \ominus Y)$ and $0 \leq (Z \ominus Y)$, so $((X \ominus Y) - (Z \ominus Y)) \leq M$ and $((Z \ominus Y) - (X \ominus Y)) \leq M$.

First, consider the case where $(X \ominus Y) \leq (Z \ominus Y)$. Then $0 \leq ((Z \ominus Y) - (X \ominus Y)) < N$ and hence $((Z \ominus Y) \ominus (X \ominus Y)) = (Z \ominus Y) - (X \ominus Y)$. Therefore,

$$\begin{aligned} (X \boxdot Z) &= (Z \ominus X) \lfloor (X \ominus Z) \\ &= Z \ominus X \\ &= (Z \ominus Y) \ominus (X \ominus Y) \\ &= (Z \ominus Y) - (X \ominus Y) \\ &\leq M. \end{aligned}$$

Next consider the case where $(Z \ominus Y) < (X \ominus Y)$. Then $0 \leq ((X \ominus Y) - (Z \ominus Y)) < N$ and hence $((X \ominus Y) \ominus (Z \ominus Y)) = (X \ominus Y) - (Z \ominus Y)$. Therefore,

$$\begin{aligned} (X \boxdot Z) &= (Z \ominus X) \lfloor (X \ominus Z) \\ &\leq X \ominus Z \\ &= (X \ominus Y) \ominus (Z \ominus Y) \\ &= (X \ominus Y) - (Z \ominus Y) \\ &= M. \end{aligned}$$

Thus, in either case, $(X \boxdot Z) \leq M$.

Q.E.D.

Lemma 4.2.2. Let X, Y, Z, and M be integers. If $(Y \ominus X) \leq M$ and $(Y \ominus Z) \leq M$, then $(X \boxdot Z) \leq M$.

Proof. Suppose $(Y \ominus X) \leq M$ and $(Y \ominus Z) \leq M$. But $0 \leq (Y \ominus X)$ and $0 \leq (Y \ominus Z)$, so $((Y \ominus X) - Y \ominus Z)) \leq M$ and $((Y \ominus Z) - (Y \ominus X)) \leq M$.

First, consider the case where $(Y \ominus X) \leq Y \ominus Z$. Then $0 \leq ((Y \ominus Z) - (Y \ominus X)) < N$ and hence $((Y \ominus Z) \oplus (Y \ominus X)) = (Y \ominus Z) - (Y \ominus X)$. Therefore,

$$\begin{aligned}(X \boxdot Z) &= (Z \ominus X) \lfloor (X \ominus Z) \\ &\leq X \ominus Z \\ &= (Y \ominus Z) \oplus (Y \ominus X) \\ &= (Y \ominus Z) - (Y \ominus X) \\ &\leq M.\end{aligned}$$

Next consider the case where $(Y \ominus Z) < Y \ominus X$. Then $0 \leq ((Y \ominus X) - (Y \ominus Z)) < N$ and hence $((Y \ominus X) \oplus (Y \ominus Z)) = (Y \ominus X) - (Y \ominus Z)$. Therefore,

$$\begin{aligned}(X \boxdot Z) &= (Z \ominus X) \lfloor (X \ominus Z) \\ &\leq Z \ominus X \\ &= (Y \ominus X) \oplus (Y \ominus Z) \\ &= (Y \ominus X) - (Y \ominus Z) \\ &= M.\end{aligned}$$

Thus, in either case, $(X \boxdot Z) \leq M$.

Q.E.D.

Theorem 4.2.3. The minimum circular distance operator $\boxdot$ for any CC banyan is a metric on the integers $0 \sim N-1$.

Proof. Let X, Y, and Z be integers in the range $0 \sim N-1$. Then,

$$(X \boxdot Y) = ((Y \ominus X) \lfloor (X \ominus Y)) \geq 0,$$

and

$$(X\boxdot Y) = (Y\ominus X)\lfloor(X\ominus Y)$$

$$= (X\ominus Y)\lfloor(Y\ominus X)$$

$$= Y\boxdot X.$$

If $(X\boxdot Y) = 0$ then

$$0 = (Y\ominus X)\lfloor(X\ominus Y)$$

$$(0 = Y\ominus X) \vee (0 = X\ominus Y)$$

$$(Y=X) \vee (X=Y)$$

$$X = Y.$$

If $X = Y$ then

$$(X\boxdot Y) = (X\ominus X)\lfloor(X\ominus X) = 0.$$

Therefore, $(X\boxdot Y) = 0$ iff $X = Y$.

Finally, to show that $(X\boxdot Z) \leq (X\boxdot Y)+(Y\boxdot Z)$, consider first the case where $(Y\ominus X) \leq X\ominus Y$ and $(Z\ominus Y) \leq Y\ominus Z$. Then $(X\boxdot Y) = Y\ominus X$ and $(Y\boxdot Z) = Z\ominus Y$. Therefore,

$$(X\boxdot Z) = (Z\ominus X)\lfloor(X\ominus Z)$$

$$= Z\ominus X$$

$$\leq Z\ominus X$$

$$= (Y\ominus X)\oplus(Z\ominus Y)$$

$$\leq (Y\ominus X)+(Z\ominus Y)$$

$$= (X\boxdot Y)+(Y\boxdot Z)$$

Second, consider the case where $(X\ominus Y) < Y\ominus X$ and $(Y\ominus Z) < Z\ominus Y$. Then $(X\boxdot Y) = X\ominus Y$ and $(Y\boxdot Z) = Y\ominus Z$. Therefore,

$$(X\boxdot Z) = (Z\ominus X)\lfloor(X\ominus Z)$$

$$\leq X\ominus Z$$

$$= (X\ominus Y)\oplus(Y\ominus Z)$$

$$\leq (X\ominus Y)+(Y\ominus Z)$$

$$= (X\boxdot Y)+(Y\boxdot Z)$$

Third, consider the case where $(X\ominus Y) < Y\ominus X$ and $(Z\ominus Y) = Y\ominus Z$. Then $(X\boxdot Y) = X\ominus Y$ and $(Y\boxdot Z) = Z\ominus Y$. Let $M = (X\ominus Y)\lceil(Z\ominus Y)$. By Lemma 4.2.1, $(X\boxdot Z) \leq M$. But,

$$\begin{aligned} M &= (X\ominus Y)\lceil(Z\ominus Y) \\ &\leq (X\ominus Y)+(Z\ominus Y) \\ &= (X\boxdot Y)+(Y\boxdot Z) \end{aligned}$$

so $(X\boxdot Z) \leq (X\boxdot Y)+(Y\boxdot Z)$.

Finally, consider the case where $(Y\ominus X) \leq X\ominus Y$ and $(Y\ominus Z) < Z\ominus Y$. Then $(X\boxdot Y) = Y\ominus X$ and $(Y\boxdot Z) = Y\ominus Z$. Let $M = (Y\ominus X)\lceil)Y\ominus Z)$. By Lemma 4.2.2, $(X\boxdot Z) \leq M$. But,

$$\begin{aligned} M &= (Y\ominus X)\lceil(Y\ominus Z) \\ &= (Y\ominus X)+(Y\ominus Z) \\ &= (X\boxdot Y)+(Y\boxdot Z) \end{aligned}$$

so $(X\boxdot Z) \leq (X\boxdot Y)+(Y\boxdot Z)$.

Thus, in every case, $(X\boxdot Z) \leq (X\boxdot Y)+(Y\boxdot Z)$.

Q.E.D.

Theorem 4.2.4. Let $\underline{B}[J1]$ and $\underline{B}[J2]$ be bases of a CC banyan G, and let I be an integer such that $0 \leq I \leq L$. Then $(\underline{B}[J1]\boxtimes\underline{B}[J2]) \leq I$ iff $(J1\boxdot J2) < \times/I\uparrow\underline{S}$.

Proof. First, suppose that $(\underline{B}[J1]\boxtimes\underline{B}[J2]) \leq I$. Then there exists a vertex $\underline{V}[I;J3]$ in level I of G such that $\underline{B}[J1] \boxminus \underline{V}[I;J3]$ and $\underline{B}[J2] \boxminus \underline{V}[I;J3]$. By Corollary 4.1.2b, $(J3\ominus J1) < \times/I\uparrow\underline{S}$, and similarly, $(J3\ominus J2) < \times/I\uparrow\underline{S}$. Therefore, $(J3\ominus J1) \leq (\times/I\uparrow\underline{S})-1$ and $(J3\ominus J1) \leq (\times/I\uparrow\underline{S})-1$. By Lemma 4.2.2, $(J1\boxdot J2) \leq ((\times/I\uparrow\underline{S})-1 < (\times/I\uparrow\underline{S})$.

Nest suppose that $(J1\boxdot J2) < \times/I\uparrow\underline{S}$. Consider the case where $(J2\ominus J1) \leq J1\ominus J2$. Then $(J2\ominus J1) = (J1\boxdot J2) < \times/I\uparrow\underline{S}$. Also $(J2\ominus J2) \leq 0$

$< \times/I\uparrow\underline{S}$. By Corollary 4.1.2b, $\underline{B}[J1] \boxleft \underline{V}[I;J2]$ and $\underline{B}[J2] \boxleft \underline{V}[I;J2]$. Therefore, $(\underline{B}[J1]\boxtimes\underline{B}[J2]) \leq I$.

On the other hand, consider the case where $(J1\ominus J2) < J2\ominus J1$. Then $(J1\ominus J2) = (J1\boxminus J2) < \times/I\uparrow\underline{S}$. Also, $(J1\ominus J1) = 0 < \times/I\uparrow\underline{S}$. By Corollary 4.1.2b, $\underline{B}[J2] \boxleft \underline{V}[I;J1]$ and $\underline{B}[J1] \boxleft [I;J2]$. Therefore, $(\underline{B}[J1]\boxtimes\underline{B}[J2]) \leq I$.

Thus, in either case where $(J1\boxminus J2) < \times/I\uparrow\underline{S}$, we obtain $(\underline{B}[J1]\boxtimes\underline{B}[J2]) \leq I$.

Q.E.D.

Theorem 4.2.5. Let $\underline{A}[J1]$ and $\underline{A}[J2]$ be apexes of a CC banyan G, and let I be an integer such that $0 \leq I \leq L$. Then $(\underline{A}[J1]\boxtimes\underline{A}[J2]) \leq L\text{-}1$ iff $0 = (\times/I\uparrow\underline{S})\mid J2\text{-}J1$.

Proof. First, suppose that $(\underline{A}[J1]\boxtimes\underline{A}[J2]) \leq L\text{-}I$. Then there exists a vertex $\underline{V}[I;J3]$ in level I of G such that $\underline{V}[I;J3] \boxleft \underline{A}[J1]$ and $\underline{V}[I;J3] \boxleft \underline{A}[J2]$. By Corollary 4.1.2c, $0 = (\times/I\uparrow\underline{S})\mid J1\text{-}J3$, and similarly, $0 = (\times/I\uparrow\underline{S})\mid J2\text{-}J3$. Therefore,

$$\begin{aligned} 0 &= (\times/I\uparrow\underline{S})\mid(J2\text{-}J3)\text{-}(J1\text{-}J3) \\ &= (\times/I\uparrow\underline{S})\mid J2\text{-}J1. \end{aligned}$$

Next, suppost that $0 = (\times/I\uparrow\underline{S})\mid J2\text{-}J1$. Then by Corollary 4.1.2c, $\underline{V}[I;J1] \boxleft \underline{A}[J2]$. But $0 = ((\times/I\uparrow\underline{S})\mid 0) = \times/I\uparrow\underline{S}\mid J1\text{-}J1$, so it also follows from Corollary 4.1.2c that $\underline{V}[I;J1] \boxleft \underline{A}[J1]$. Therefore, $(\underline{A}[J1]\boxtimes\underline{A}[J2]) \leq L\text{-}I$.

Q.E.D.

Corollary 4.2.5a. Let $\underline{A}[J1]$ and $\underline{A}[J2]$ be apexes of a CC banyan G, and let I be an integer such that $0 \leq I \leq L$. Then $(\underline{A}[J1]\boxtimes\underline{A}[J2]) \leq I$ iff $0 = (\times/(\text{-}I)\downarrow\underline{S})\mid J2\text{-}J1$.

Proof. Since $I = L\text{-}(L\text{-}I)$, it follows from Theorem 4.2.5 that $(\underline{A}[J1]\boxtimes\underline{A}[J2])$

$\leq I$ iff $0 = (\times/(L-I)\uparrow\underline{S})|J2-J1$. But $((L-I)\uparrow\underline{S}) = (-I)\downarrow\underline{S}$, so $(\underline{A}[J1]\boxtimes\underline{A}[J2])$ $\leq I$ iff $0 = (\times/(-I)\downarrow\underline{S})|J2-J1$.

Q.E.D.

Theorem 4.2.6. Let G be a CC banyan such that $2 \leq \underline{S}[I]$ for every $I = 1 \sim L$. Then the base distance operator $\boxtimes$ is a metric on the bases of G.

Proof. Let $\underline{B}[J1]$, $\underline{B}[J2]$, and $\underline{B}[J3]$ be bases of G. By Theorem 2.1.2, $0 \leq \underline{B}[J1]\boxtimes\underline{B}[J2]$. By Theorem 2.1.3, $0 = \underline{B}[J1]\boxtimes\underline{B}[J2]$ iff $\underline{B}[J1] = \underline{B}[J2]$. By Theorem 2.1.1, $(\underline{B}[J1]\boxtimes\underline{B}[J2]) = \underline{B}[J2]\boxtimes\underline{B}[J1]$.

To prove the triangle inequality, let $I1 = \underline{B}[J1]\boxtimes\underline{B}[J2]$, let $I2$ $= \underline{B}[J2]\boxtimes\underline{B}[J3]$, and let $I3 = I1+I2$. If $I1 = 0$ then $\underline{B}[J1] = \underline{B}[J2]$ and $(\underline{B}[J1]\boxtimes\underline{B}[J3]) = I2 = I3$. If $I2 = 0$ then $\underline{B}[J2] = \underline{B}[J3]$ and $(\underline{B}[J1]\boxtimes\underline{B}[J3])$ $= I1 = I3$. If $I3 \geq L$ then $(\underline{B}[J1]\boxtimes\underline{B}[J2]) \leq I3$ by Theorem 2.1.2.

Suppose, on the other hand, that $0 < I1$ and $0 < I2$ and $I3 < L$. Then by Theorem 4.2.4, $(J1\boxtimes J2] < \times/I1\uparrow\underline{S}$ and $(J2\boxtimes J3) < \times/I2\uparrow\underline{S}$. Consequently, $((J1\boxtimes J2)+J2\boxtimes J3) < (\times/I1\uparrow\underline{S}) + \times/I2\uparrow\underline{S}$. Since $\boxtimes$ is a metric, $(J1\boxtimes J3)$ $\leq (J1\boxtimes J2)+J2\boxtimes J3$. Therefore,

$$(J2\boxtimes J3) < (\times/I1\uparrow\underline{S}) + \times/I2\uparrow\underline{S}$$

$$\leq 2 \times \times/(I1\lceil I2)\uparrow\underline{S}.$$

Since $2 \leq \underline{S}[I3]$,

$$(J1\boxtimes J3) < \underline{S}[I3] \times \times/(I1\lceil I2)\uparrow\underline{S}.$$

Since $0 < I1$ and $0 < I2$, $I3 = (I1+I2) > I1\lceil I2$. Therefore,

$$(J1\boxtimes J3) < \times/I3\uparrow\underline{S}.$$

By Theorem 4.2.4,

$$(\underline{B}[J1]\boxtimes\underline{B}[J3]) \leq I3.$$

Thus, in every case,

$$(\underline{B}[J1]\mathbb{M}\underline{B}[J3]) \leq I3 = (\underline{B}[J1]\mathbb{M}\underline{B}[J2])+\underline{B}[J2]\mathbb{M}\underline{B}[J3].$$

Q.E.D.

Theorem 4.2.7. The apex distance operator $\mathbb{M}$ is a metric on the apexes of a CC banyan.

Proof. Let $\underline{A}[J1]$, $\underline{A}[J2]$, and $\underline{A}[J3]$ be apexes of G. By Theorem 2.1.2, $0 \leq \underline{A}[J1]\mathbb{M}\underline{A}[J2]$. By Theorem 2.1.3, $0 = \underline{A}[J1]\mathbb{M}\underline{A}[J2]$ iff $\underline{A}[J1] = \underline{A}[J2]$. By Theorem 2.1.1, $(\underline{A}[J1]\mathbb{M}\underline{A}[J2]) = \underline{A}[J2]\mathbb{M}\underline{A}[J1]$.

To prove the triangle inequality, let $I1 = L\text{-}\underline{A}[J1]\mathbb{M}\underline{A}[J2]$ and let $I2 = L\text{-}\underline{A}[J2]\mathbb{M}\underline{A}[J3]$. Hence,

$$(\underline{A}[J1]\mathbb{M}\underline{A}[J2]) = L\text{-}I1$$

and

$$(A[J2]\mathbb{M}\underline{A}[J3]) = L\text{-}I2.$$

Then, by Theorem 4.2.5,

$$0 = (\times/I1\uparrow\underline{S})|J2\text{-}J1$$

and

$$0 = (\times/I2\uparrow\underline{S})|J3\text{-}J2.$$

Since $(I1\lfloor I2) \leq I1$,

$$0 = (\times/(I1\lfloor I2)\uparrow\underline{S})|J2\text{-}J1.$$

Since $(I1\lfloor I2) \leq I2$,

$$0 = (\times/(I1\lfloor I2)\uparrow\underline{S})|J3\text{-}J2.$$

Therefore,

$$\begin{aligned} 0 &= (\times/(I1\lfloor I2)\uparrow\underline{S})|(J3\text{-}J2)+(J2\text{-}J1) \\ &= (\times/(I1\lfloor I2)\uparrow\underline{S})|J3\text{-}J1. \end{aligned} \tag{1}$$

Since apex distances are nonnegative, $L \geq I1$ and $L \geq I2$. Thus,

$$L \geq I1\lceil I2$$

$$0 \geq (I1\lceil I2)\text{-}L$$

$$(I1\lfloor I2) \geq (I1\lfloor I2)+(I1\lceil I2)-L$$

$$(I1\lfloor I2) \geq I1+I2-L$$

$$0 = (\times/(I1+I2-L)\uparrow \underline{S})|(\times/(I1\lfloor I2)\uparrow \underline{S}). \qquad (2)$$

From equations (1) and (2) above,

$$0 = (\times/(I1+I2-L)\uparrow \underline{S})|J3-J1.$$

By theorem 4.2.5,

$$(\underline{A}[J1]\mathbb{V}\underline{A}[J2]) \leq L-(I1+I2-L)$$

$$(\underline{A}[J1]\mathbb{V}\underline{A}[J2]) \leq (L-I1)+(L-I2)$$

$$(\underline{A}[J1]\mathbb{V}\underline{A}[J2]) \leq (\underline{A}[J1]\mathbb{V}\underline{A}[J2])+\underline{A}[J2]\mathbb{V}\underline{A}[J3].$$

Q.E.D.

APPENDIX C

BIDIRECTIONAL SWITCHING CIRCUITS

A switching device is said to be "bidirectional" if it can pass a signal in either direction rather than in just one direction. Mechanical switches and relay contacts are probably the simplest bidirectional devices. They will pass signals equally well in both directions when closed and will block signals in both directions when open. Figure C-1 shows a simple relay used as a bidirectional switch.

In contrast, the AND gate shown in Figure C-2 is a unidirectional switch since it controls signal propagation in one direction only. This circuit has the advantage, however, that it can be realized using a small, high-speed electronic circuit. Further, with just about any logic family other than diode logic, the AND gate switch can amplify the signal it controls. Power amplification of data signals can be very important in a large switching network so that data from a low-power source can be routed to many possible destinations without exceeding fanout limitations.

There are a number of simple electronic circuits which can both switch and amplify bidirectional binary signals. Standard logic families such as DTL, TTL, ECL, and I^2L can be used. Most any family can be adapted for bidirectional switching so long as it supports either "wired AND" or "wired OR" connections. Figures C-3 and C-4 show bidirectional switching circuits which can be built from standard TTL gates and ECL gates, respectively. Figure C-5 shows a comparable I^2L circuit suitable

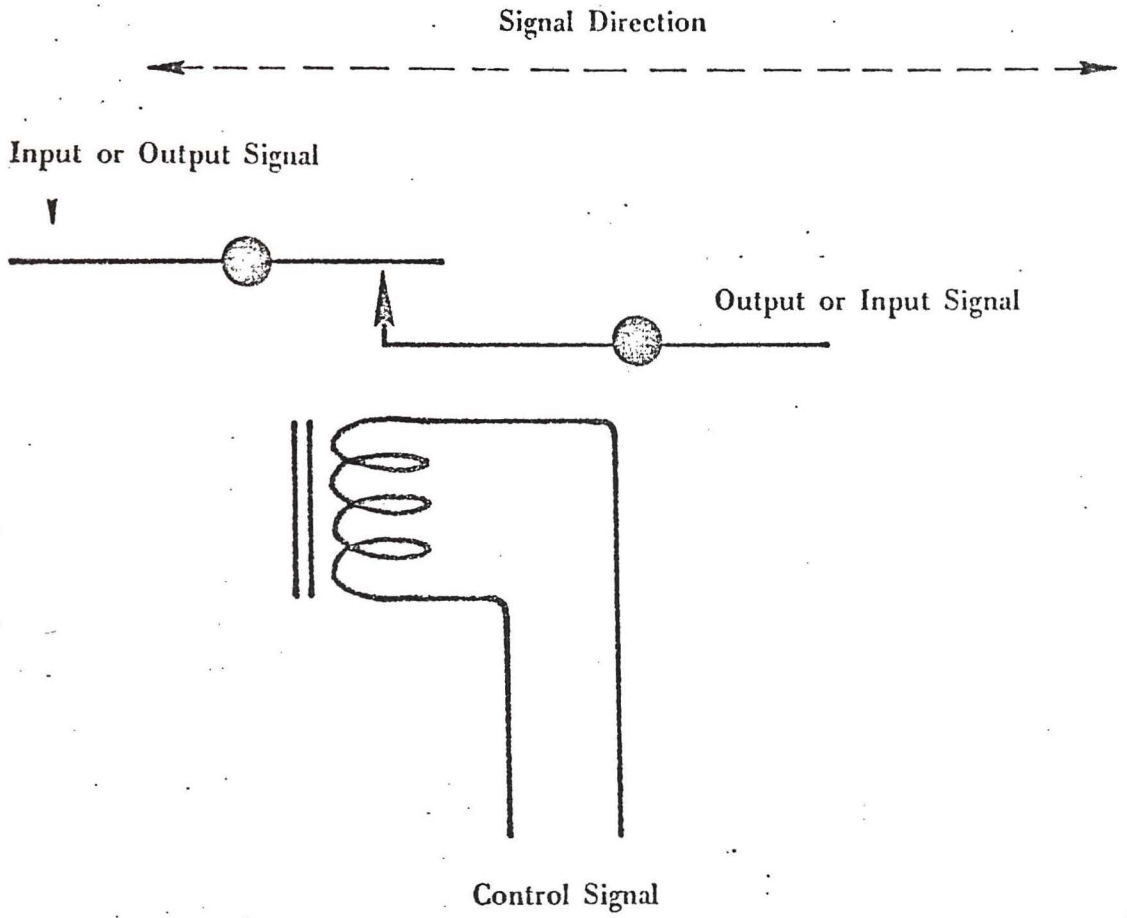
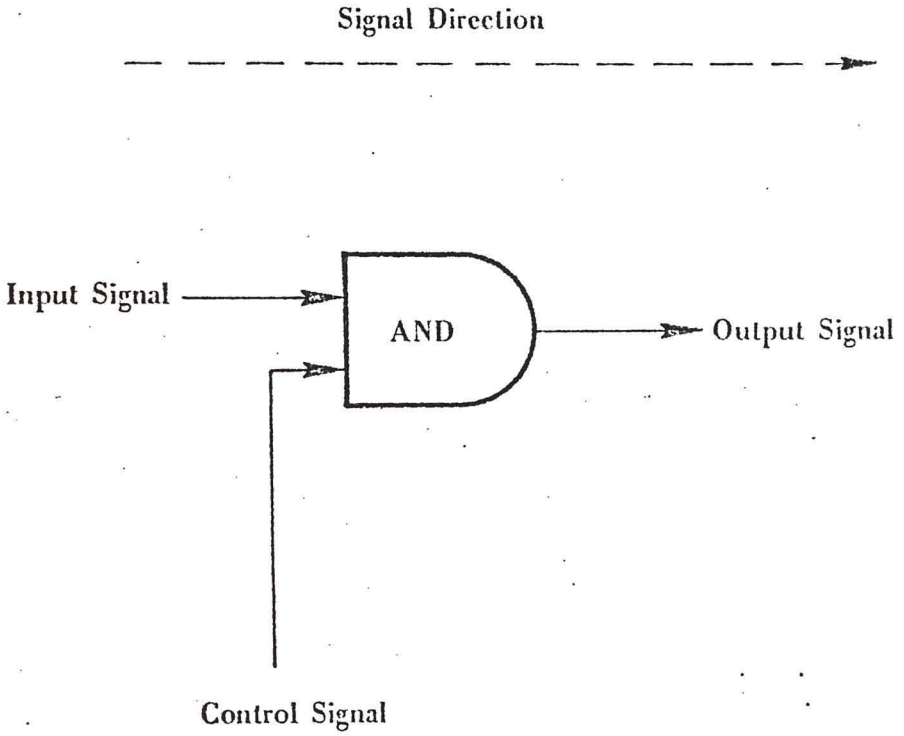
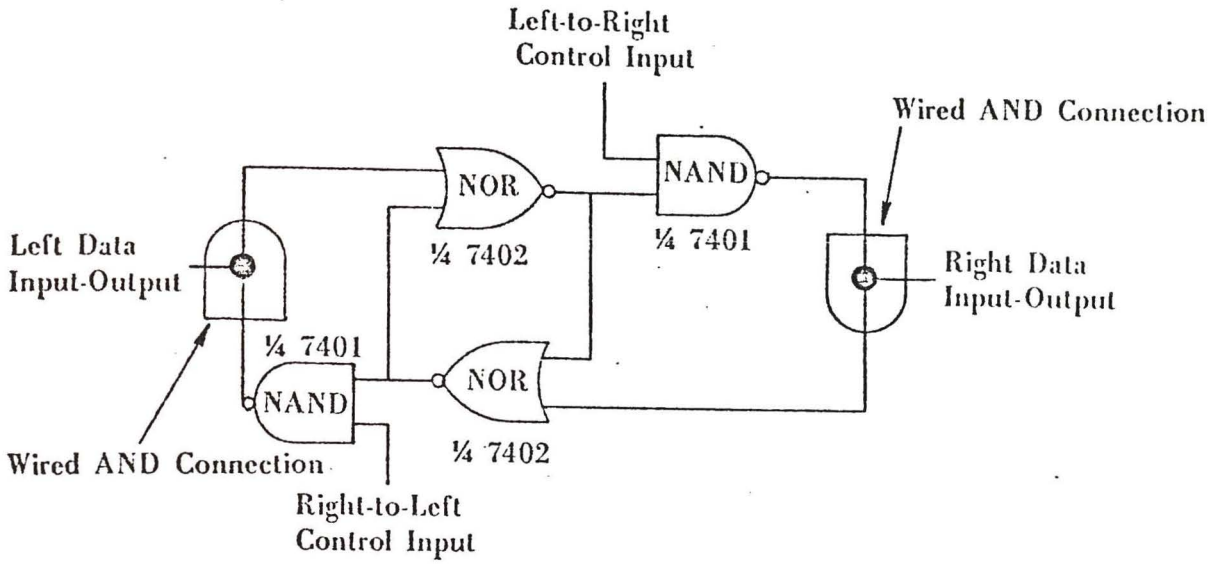


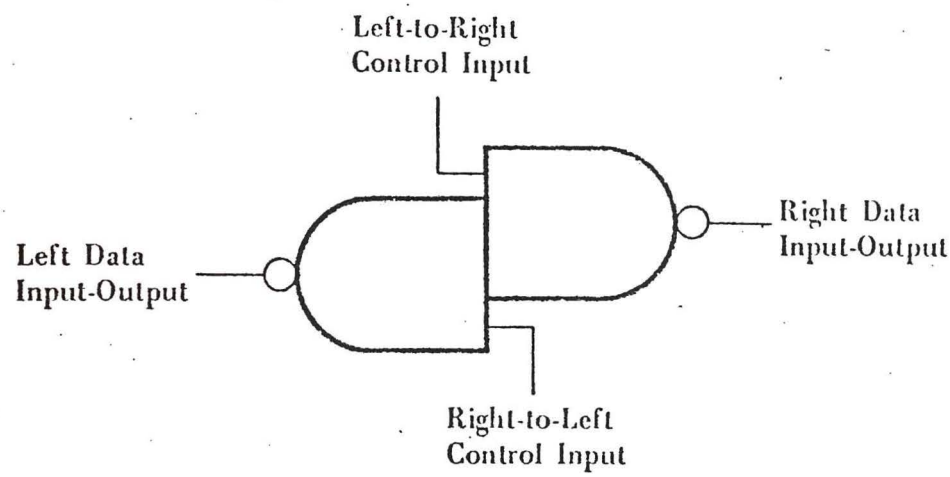
Figure C-1. Relay Used as a Bidirectional Switch



Fgure C-2. AND Gate Used as a Unidirectional Switch

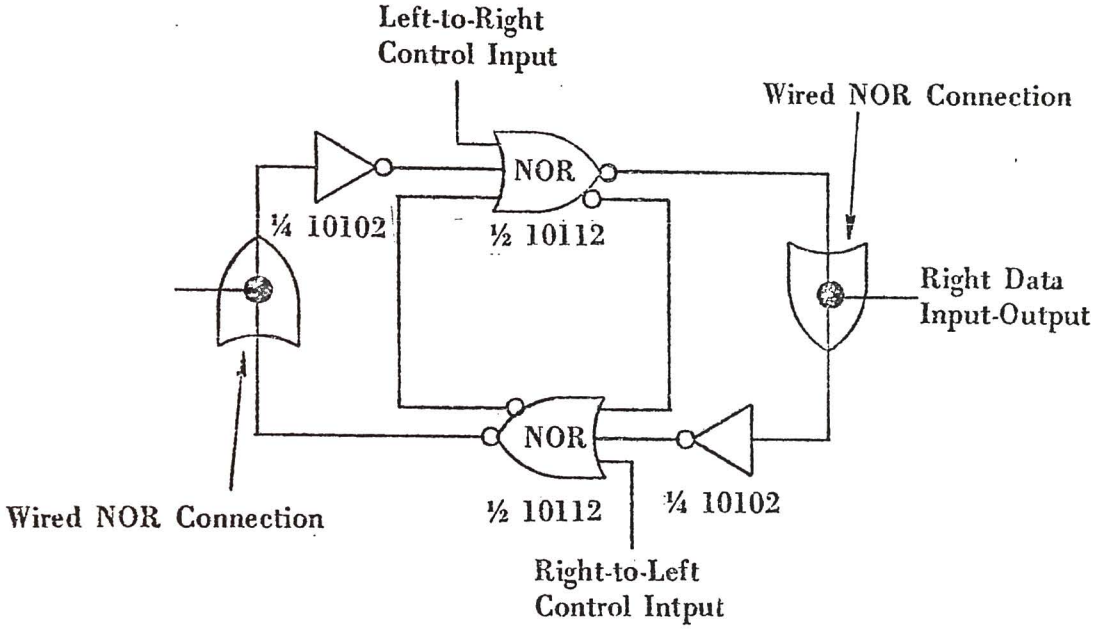


a) Logic Diagram

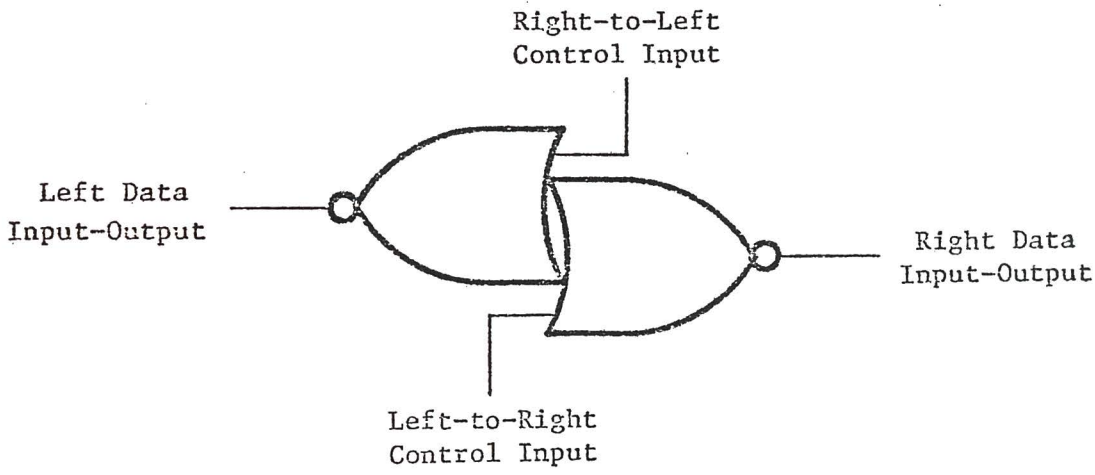


b) Suggested Symbol for Circuit Above

Figure C-3. Bidirectional Switch Using Standard TTL Gates

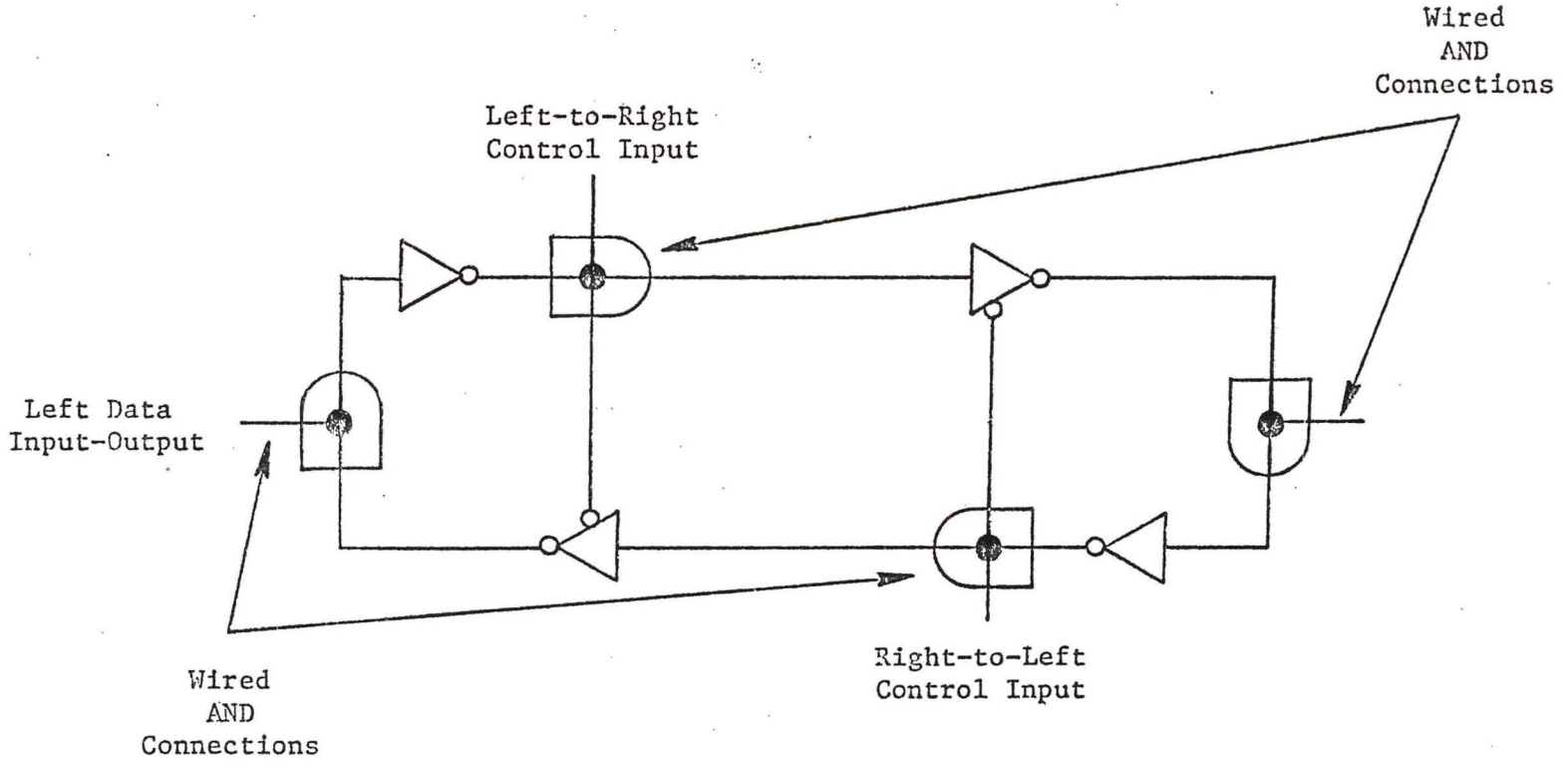


a) Logic Diagram



b) Suggested Symbol for Circuit Above

Figure C-4. Bidirectional Switch Using Standard ECL Gates



NOTE: Suggested symbol for this circuit is same as for TTL circuit in Figure C-3.

Figure C-5. Bidirectional Switch for LSI Using I^2L Gates

for large scale integration. Integrated DTL bidirectional switches were constructed by Vice et al. (73), who also surveyed a number of earlier bidirectional circuits.

APPENDIX D

COMPLETE SIMULATION DATA

Statistical data collected from the simulation experiments discussed in Section 8 is shown in Tables D-1 through D-7. Figures 8.2-1 through 8.2-4 and Tables 8.2-2 through 8.2-4 were derived from this data.

TABLE D-1 Simulation Results for SW Banyans Using Far Apex Selection Rule and Standard Set-Up Rule

Fanout and Spread	Number of Bases	Layers Required			Nonempty Subsystems Connected (percent)				Total Nonempty Subsystems
		Mean	Standard Error of Mean	Maximum	1 Layer	2 Layers	3 Layers	4 Layers	
2	4	1.03	.017	2	98.46	100.00			195
2	8	1.25	.044	2	92.69	100.00			342
2	16	1.73	.045	2	85.34	100.00			689
2	32	1.95	.036	3	79.24	99.68	100.00		1233
2	64	2.35	.061	3	76.04	98.05	100.00		2458
2	128	2.72	.057	4	73.33	96.76	99.92	100.00	4999
3	9	1.24	.043	2	93.88	100.00			392
3	27	1.92	.027	2	84.36	100.00			1215
3	81	2.12	.043	3	80.48	99.37	100.00		3197
4	16	1.37	.049	2	93.76	100.00			689
4	64	1.91	.032	3	83.40	99.96	100.00		2458
4	256	2.39	.060	4	87.42	99.09	99.95	100.00	12931
8	64	1.80	.040	2	90.15	100.00			2458

TABLE D-2 Simulation Results for SW Banyans Using Near Apex Selection Rule and Standard Set-Up Rule

Fanout and Spread	Number of Bases	Layers Required			Nonempty Subsystems Connected (percent)				Total Nonempty Subsystems
		Mean	Standard Error of Mean	Maximum	1 Layer	2 Layers	3 Layers	4 Layers	
2	8	1.07	.026	2	97.95	100.00			342
2	64	2.08	.046	3	81.77	99.35	100.00		2458
3	27	1.82	.039	2	89.55	100.00			1215
4	16	1.19	.039	2	97.24	100.00			689
4	64	1.89	.031	2	88.61	100.00			2458

TABLE D-3 Simulation Results for SW Banyans Using Far Apex Selection Rule and Modified Set-Up Rule

Fanout and Spread	Number of Bases	Layers Required			Nonempty Subsystems Connected (percent)				Total Nonemtpy Subsystems
		Mean	Standard Error of Mean	Maximum	1 Layer	2 Layers	3 Layers	4 Layers	
2	8	1.05	.022	2	98.54	100.00			342
2	64	2.15	.052	3	84.17	99.02	100.00		2458
4	64	1.85	.036	2	90.15	100.00			2458

TABLE D-4 Simulation Results for SW Banyans Using Near Apex Selection Rule and Modified Set-Up Rule

Fanout and Spread	Number of Bases	Layers Required			Nonempty Subsystems Connected (percent)				Total Nonempty Subsystems
		Mean	Standard Error of Mean	Maximum	1 Layer	2 Layers	3 Layers	4 Layers	
2	4	1.00	0.0	1	100.00				195
2	8	1.05	.022	2	98.54	100.00			342
2	16	1.44	.050	2	93.32	100.00			689
2	32	1.80	.040	2	87.59	100.00			1233
2	64	2.05	.046	3	86.05	99.43	100.00		2458
4	16	1.03	.017	2	99.56	100.00			689
4	64	1.74	.044	2	92.19	100.00			2458
8	64	1.37	.049	2	97.48	100.00			2458

TABLE D-5 Simulation Results for CC Banyans Using Far Apex Selection Rule and Standard Set-Up Rule

Fanout and Spread	Number of Bases	Layers Required			Nonempty Subsystems Connected (percent)				Total Nonempty Subsystems
		Mean	Standard Error of Mean	Maximum	1 Layer	2 Layers	3 Layers	4 Layers	
5	64	2.41	.064	4	70.38	97.52	99.96	100.00	1458

TABLE D-6 Simulation Results for CC Banyans Using Near Apex Selection Rule and Standard Set-Up Rule

Fanout and Spread	Number of Bases	Layers Required			Nonempty Subsystems Connected (percent)				Total Nonempty Subsystems
		Mean	Standard Error of Mean	Maximum	1 Layer	2 Layers	3 Layers	4 Layers	
2	4	1.03	.017	2	98.46	100.00			195
2	8	1.32	.047	2	90.35	100.00			342
2	16	1.80	.040	2	80.84	100.00			684
2	32	1.94	.034	3	76.72	99.76	100.00		1233
2	64	2.17	.053	3	73.31	98.94	100.00		2458
4	16	1.33	.047	2	94.78	100.00			689
4	64	1.90	.030	2	85.68	100.00			2458

TABLE D-7 Simulation Results for CC Banyans Using Near Apex Selection Rule and Modified Set-Up Rule

Fanout and Spread	Number of Bases	Layers Required			Nonempty Subsystems Connected (percent)				Total Nonempty Subsystems
		Mean	Standard Error of Mean	Maximum	1 Layer	2 Layers	3 Layers	4 Layers	
2	4	1.00	0.0	1	100.00				195
2	8	1.14	.035	2	95.91	100.00			342
2	16	1.57	.050	2	89.70	100.00			689
2	32	1.89	.034	3	84.35	99.92	100.00		1233
2	64	2.12	.050	3	81.53	99.19	100.00		2458
4	16	1.07	.026	2	98.84	100.00			689
4	64	1.82	.039	2	91.83	100.00			2458

REFERENCES

Barnes, G. H., R. M. Brown, M. Kato, D. J. Kuck, D. L. Slotnick, and R. A. Stokes, "The ILLIAC IV Computer," IEEE Transactions on Computers, Vol. C-17, pp. 746-757, Aug., 1968.

Baskin, H. B., E. B. Horowitz, R. D. Tennison, and L. E. Rittenhouse, "A Modular Computer Sharing System," Communications of the ACM, Vol. 12, No. 10, pp. 551-559, Oct., 1969.

_____, B. R. Borgerson, and R. Roberts, "Prime - A Modular Architecture for Terminal-Oriented Systems," AFIPS Proc. SJCC, Vol. 40, pp. 431-437, 1972.

Batcher, K. E., "Sorting Networks and Their Applications," AFIPS Proc. SJCC, Vol. 32, pp. 307-314, 1968.

Benes, V. E. (62), "Algebraic and Topological Properties of Connecting Networks," Bell System Technical Journal, pp. 1249-1274, July, 1962.

_____ (64a), "Permutation Groups, Complexes, and Rearrangeable Connecting Networks," Bell System Technical Journal, pp. 1619-1640, July, 1964.

_____ (64b), "Optimal Rearrangeable Multistage Connecting Networks," Bell System Technical Journal, pp. 1641-1656, July, 1964.

_____ (65), Mathematical Theory of Connecting Networks and Telephone Traffic, Academic Press, New York, 1965.

Berge, Claude, The Theory of Graphs, John Wiley and Sons, Inc., New York, 1962.

Bhandarkar, D. P. and J. E. Juliussen, "A Comparative Evaluation of the Cost Effectiveness of Computer Systems," in J. D. White, ed., Proc. Annual Conference of the ACM, pp. 323-324, Oct. 20-22, 1975.

Clos, C., "A Study of Non-Blocking Switching Networks," Bell System Technical Journal, pp. 406-424, March, 1953.

Comptre Corporation, Enslow, P. H., Jr., ed., Multiprocessors and Parallel Processing, John Wiley and Sons, Inc., New York, 1974.

Davis, R. L., "The ILLIAC IV Processing Element," IEEE Transactions on Computers, Vol. C-18, No. 9, pp. 800-816, Sept., 1969.

Feng, Tse-Yun, "Data Manipulating Functions in Parallel Processors and Their Implementations," IEEE Transactions on Computers, Vol. C-23, No. 3, pp. 309-318, March, 1974.

Foster, C. C., "Determination of Priority in Associative Memories," IEEE Transactions on Computers, Vol. C-17, No. 8, pp. 788-789, Aug., 1968.

Frank, R. A., "Imsai Arrays Micros for Low-Cost Power," Computerworld, Vol. IX, No. 44, pp. 1 and 3, Oct. 29, 1975.

Gilman, L. and A. J. Rose, APL/360 an Interactive Approach, John Wiley and Sons, Inc., New York, 1970.

Goke, L. R. and G. J. Lipovski, "Banyan Networks for Partitioning Multiprocessor Systems," in Lipovski, G. J. and S. A. Szygenda, ed., Proc. First Annual Symposium on Computer Architecture, IEEE Catalog no. 73CH0824-3C, pp. 21-28, Dec. 9-11, 1973.

Goldstein, L. J. and S. W. Leibholz, "On the Synthesis of Signal Switching Networks with Transient Blocking," IEEE Transactions on Electronic Computers, Vol. EC-16, No. 5, pp. 637-641, Oct., 1967.

IBM, APL/360 User's Manual, Publication GH20-0683-1, IBM Corp., 1968.

Iverson, K. E., A Programming Language, John Wiley and Sons, 1962.

Joel, A. E., Jr., "On Permutation Switching Networks," Bell System Technical Journal, pp. 813-822, May/June, 1968.

Lang, T. and H. S. Stone, "A Shuffle-Exchange Network with Simplified Control," IEEE Transactions on Computers, Vol. C-25, pp. 55-65, Jan., 1976.

Lawrie, D. H. (73), "Memory-Processor Connection Networks," University of Illinois Report no. UIUCDCS-R-73-557, Feb., 1973.

_____ (75), "Access and Alignment of Data in an Array Processor," IEEE Transactions on Computers, Vol. C-24, No. 12, pp. 1145-1155, 1975.

Lipovski, G. J. (69), The Architecture of a Large Distributed Logic Associative Processor, Coordinated Science Laboratory Report R-424, University of Illinois, July, 1969.

_____ (70), "The Architecture of a Large Associative Processor," AFIPS Proc. SJCC, Vol. 36, pp. 385-396, 1970.

Pease, M. C., "An Adaptation of the Fast Fourier Transform for Parallel Processing," Journal of the ACM, Vol. 15, pp. 252-264, April, 1968.

Schultz, G. W., R. M. Holt, and H. L. McFarland, "A Guide to Using LSI Microprocessors," Computer, pp. 13-19, June, 1973.

Searle, B. C. and D. E. Freberg, "Tutorial: Microprocessor Applications in Multiple Processor Systems," Computer, Vol. 8, No. 10, pp. 22-30, Oct., 1975.

Stone, H. S., "Parallel Processing with the Perfect Shuffle," IEEE Transactions on Computers, Vol. C-20, No. 2, pp. 153-161, Feb., 1971.

Thurber, K. J., E. D. Jensen, L. A. Jack, L. L. Kinney, P. C. Patton, and L. C. Anderson, "A Systematic Approach to the Design of Digital Bussing Structures," AFIPS Proc. FJCC, Vol. 41, pp. 719-740, 1972.

Vice, W. E., A. J. Brodersen, and G. J. Lipovski, "On Integrated Circuit Bidirectional Amplifiers," Journal of Solid State Circuits, Oct., 1973.

Waksman, A., "A Permutation Network," Journal of the ACM, Vol. 15, No. 1, pp. 159-163, Jan., 1968.

Wulf, W. A. and C. G. Bell, "C.mmp - A Multi-mini-processor," AFIPS Proc. FJCC, Vol. 41, pp. 765-777, 1972.

BIOGRAPHICAL SKETCH

Louis Rodney Goke was born December 6, 1946, in Memphis, Tennessee. In 1968, he received the B.S. degree in Mathematics from Christian Brothers College at Memphis. In 1971, he received the M.S.E. degree in Electrical Engineering from the University of Florida. While at the University of Florida, he held a College of Engineering Fellowship from 1968 to 1969, a teaching assistantship in the Department of Electrical Engineering from 1969 through 1970, and a research assistantship in the Department of Clinical Psychology from 1971 to 1973. Since 1973, he has been employed as a Member of the Technical Staff with Texas Instruments, where he has performed research and design work concerning operating systems, programming languages, software reliability, and software engineering methodologies. He is a member of the Institute of Electrical and Electronic Engineers, the IEEE Computer Society, the Association for Computing Machinery, and the ACM Special Interest Group on Programming Languages. He participated in the IEEE Region 3 Student Paper Competition in 1968 and subsequently has published three technical papers, two with Dr. Keith L. Doty concerning his master's research and one with Dr. G. J. Lipovski concerning his doctoral research.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Gerald J. Lipovski, Chairman
Associate Professor of Electrical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Keith L. Doty
Associate Professor of Electrical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Stanley Y. W. Su
Associate Professor of Electrical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

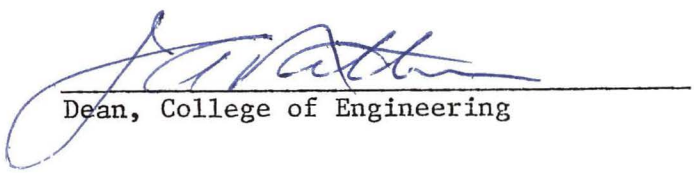
Alexander R. Bednarek
Professor of Mathematics

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Frank D. Vickers
Associate Professor of Computer Science

This dissertation was submitted to the Graduate Faculty of the College of Engineering and to the Graduate Council, and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

June, 1976



Dean, College of Engineering

Dean, Graduate School