8523095

Peyravi, Mohammad-Hassan

STUDY OF INTERCONNECTION NETWORKS

The University of Oklahoma PH.D. 1985

University
Microfilms
International 300 N. Zeeb Road, Ann Arbor, MI 48106

THE UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

STUDY OF INTERCONNECTION NETWORKS

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

degree of

DOCTOR OF PHILOSOPHY

BY

MOHAMMAD HASSAN PEYRAVI

Norman, Oklahoma

1985

STUDY OF INTERCONNECTION NETWORKS

A DISSERTATION

APPROVED FOR THE SCHOOL OF ELECTRICAL ENGINEERING

AND COMPUTER SCIENCE

BY

S. Lakshmivarahan

[illegible]

A.R. Hurson

M. Breen

John T. Minor

ACKNOWLEDGMENTS

I would like to express my sincere gratitude and appreciation to my dissertation advisor, Dr. S. Lakshmivarahan for his inspiration, constant encouragement and enthusiastic guidance throughout this dissertation and my entire graduate work. It was largely for the opportunity to work and learn from him that I chose to continue at the University of Oklahoma beyond the master's program and I feel that his influence has contributed greatly to my technical and professional preparation.

I also wish to express my appreciation to Dr. S. Kothari for his constructive help.

My special thanks go to Dr. S. K. Kahng, Director of the School of Electrical Engineering and Computer Science, for his support and thoughtful advice throughout my doctoral program.

I would like to take this opportunity to thank my dissertation committee members, Dr. M. Breen, Dr. S. Dhall, Dr. A. Hurson and Dr. J. Minor for their contribution and advise.

I am sincerely grateful to my brother and sisters for their continuing moral support, understanding and encouragement over many years. And finally, the warmest thanks and

most affectionate appreciation are extended to my parents who suffered the most throughout the several years of graduate school and brought me up and made me what I am with their endless love. With love and gratitude, I wish to dedicate this work to them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure | Page

ABSTRACT

A multi-stage NxN interconnection network is said to be universal if it realizes the set of all permutations on N objects. A new bound on the number of stages required for the universality of shuffle-exchange network as well as the analysis of the combinatorial power for the block-structured networks are given. Finally, the complexity of the verification of a new sufficient condition for rearrangeability due to Benes[B5] is analyzed.

Chapter I

INTRODUCTION

1.1 PARALLEL COMPUTERS:

The computing power of a machine is often defined as the number of floating point operations it can perform in a second. In principle, there are two ways to increase the computing power, first by increasing the speed of the basic hardware and second using parallelism by replication of hardware. However, there are two fundamental limiting factors which determine the speed of the hardware namely switching delay of the basic components and the signal propagation delay. These two factors have been decreasing steadily from one generation to the next generation (from 10^{-6} to 10^{-9}) by using a variety of technological innovations such as bipolar integrated circuits on silicon chip, large scale integration etc. A tremendous reduction in overall size of the hardware components and a dramatic decrease in cost of the hardware are two major aspects of this large scale integration. Based on the concept of computer on a chip and using VLSI, the prospect of further reduction in the cost of hardware have only become better. While further reduction in the switching delay is not impossible, it is clear that the signal propagation delay dominates the overall speed of the hardware. However, there is a growing consensus among the leading experts that the limit is seen for shrinking the size of the integrated circuit elements [W1]. Based on the above argument, we can conclude that until further significant advances in VLSI are made the only way to increase the computing power is through the replication of the hardware while having the present level of switching delay. In other words, parallel computation is perhaps the only feasible and viable solution to increase the computing power [A3,H5].

The term parallel processing or parallel computation refers to performing more than one operation at a time. Accordingly any computational device which is capable of computing in parallel is is known as a parallel computer. In the early days, parallelism was introduced at the level of basic arithmetic operations and parallel input/output operations. Over the past decade, the term parallel computers have been used exclusively to denote machines with certain architectural features. In section 1.1 several different classes of machines are identified based on their architecture and their functionality [H6,K5].

1.1.1 Multiprocessors:

In a multi-processor [K5], more than one computer share a common bank of memory units. This is the highest level for obtaining parallelism. Several copies of one program or different programs can operate in parallel on different data set. The communication between processors can be done directly or indirectly through a common memory bank using an interconnection network as shown in Figure 1.1. Some popular examples of this type are c.mmp, cm* (of the Carnegie Mellon University) and HEP(Hetrogeneous Element Processor) of Denelcor Inc.. The c.mmp has 16 minicomputers (PDP 11/70's) sharing a bank of 16 memory units through a 16?16 complete cross-bar switch.

1.1.2 Array Processors:

This is an intermediate level of parallelism [K5] which consists of an array of identical processing elements and they are controlled by a master computer. Each processing unit has its own local memory (set of registers) as shown in Figure 1.2(a). The array itself is obtained by interconnecting the processor elements in a network. Each processing unit has a limited number of instruction sets. The master computer broadcasts the same instruction to the processing units and they operate simultaneously on different data sets. ILLIAC IV, STARAN, MPP(massive parallel processor), DAP(distributed array processor of ICL)and BSP(Burrough's scientific processor) are examples of array processors, as shown in Figure 1.2(b). Array processors have gained wide acceptance as a viable parallel computer.

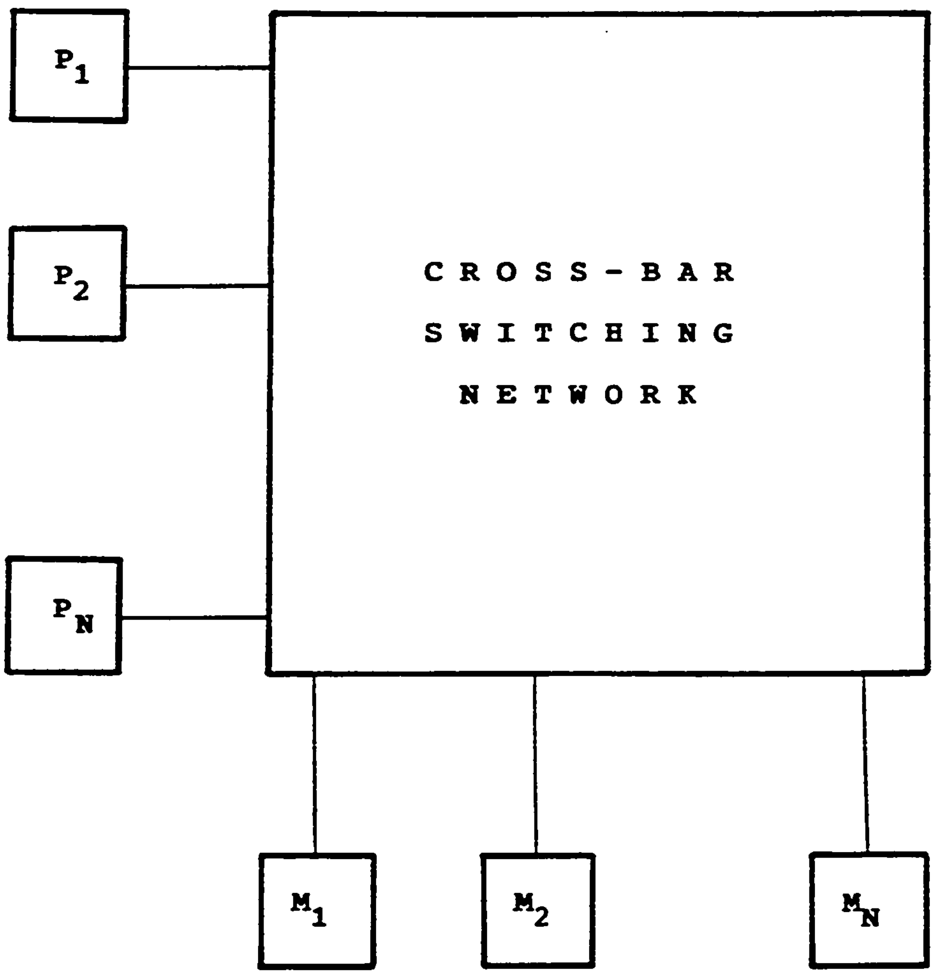


Figure 1.1: An example of a multi-processor c.mmp. P_i and M_i are the i^{th} processor and memory processing elements, respectively, $i=1,2, \ldots ,N$.

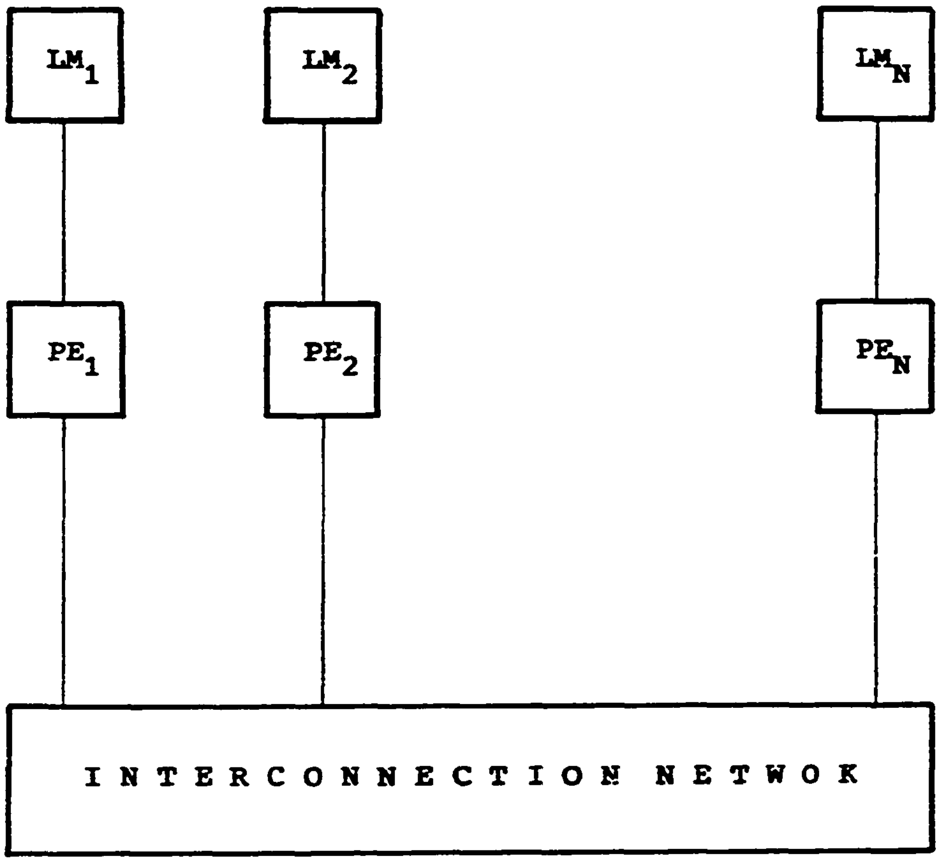


Figure 1.2(a): An example of an array of processing elements. Each PE_i with local memory LM_i, $i=1,2, \ldots, N$, interconnected through a network.

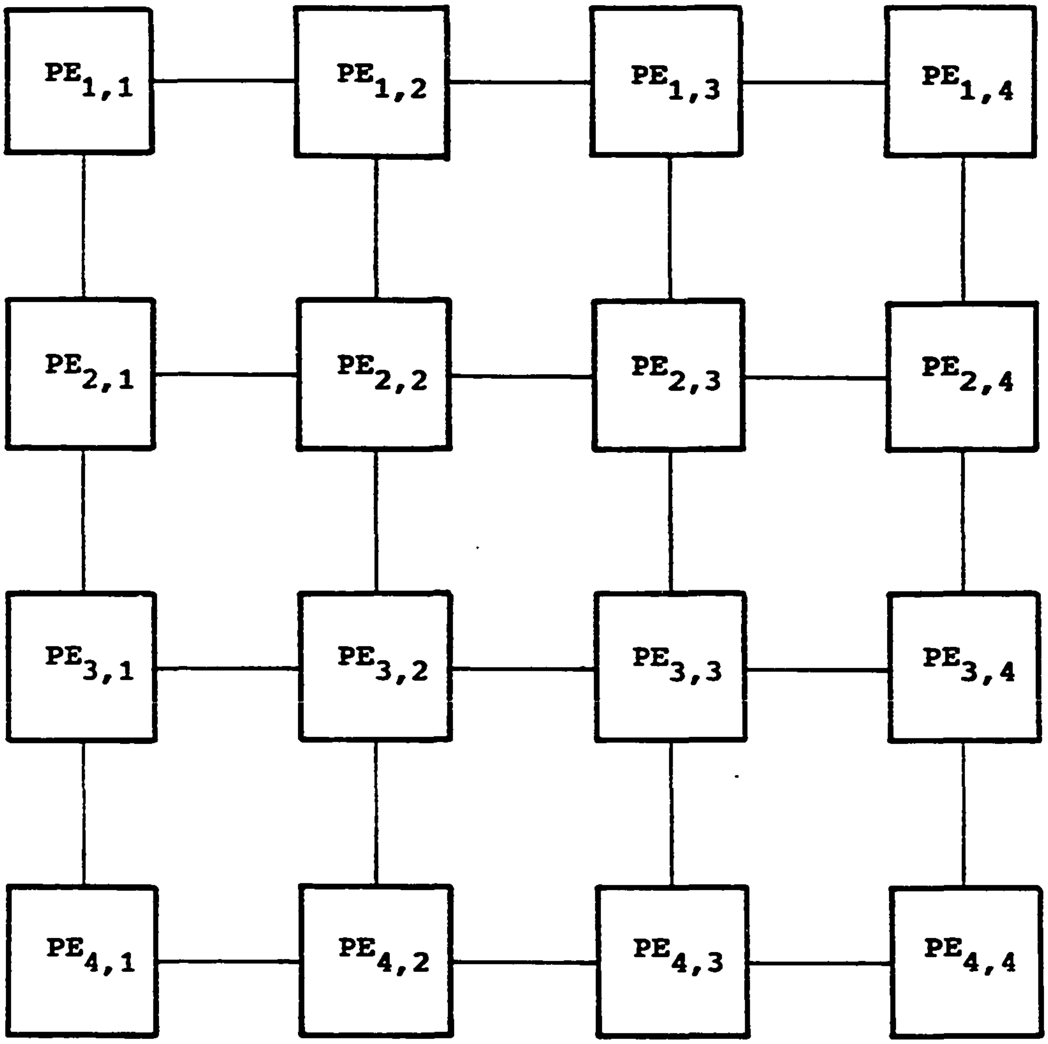


Figure 1.2(b): An example of a two-dimensional array of processing elements. $PE_{i,j}$ is the processing element in the i^{th} row and j^{th} column, i,j=1,2,3,4. For convenience, local memories attached to these processors are not explicitly shown.

1.1.3 Multi-Functional Units:

Providing multiple functional units to perform different operations in parallel on different data sets such as addition, multiplication, logic and index computation for instruction fetch, etc. ATLAS, HEP and CRAY-1 are examples of this kind.

1.1.4 Pipelining:

Processes running on a pipelined processor are decomposed into a series of sequential subprocesses [F2]. For example, floating point addition can be broken into four stages: Comparison of exponents, shifting mantissa, addition and normalizing. Each subprocess is executed on a dedicated facility. e.g. Amdahl, Cray-1 and Cyber- 205.

In the literature, the term parallel computer refers either to the multi-processors(level 1) or array processors(level 2). The above classification of the parallel computers depends on the architecture of these machines. Flynn(1972) [F2] classified these machines based on their functionality as follows.

a) SISD machines(Single Instruction stream, Single Data stream): Conventional(serial) machines may be characterized at the functional level as SISD machines. e.g. IBM 360, PDP 11/70 and Vax 11/780.

b) SIMD machines(Single Instruction stream, Multiple Data Stream): Parallel computers such as array processors are characterized at this level.

c) MIMD machines(Multiple Instruction stream, Multiple Data stream): Parallel computer such as multi-processors are classified in this class.

d) MISD machines(Multiple Instruction stream, Single Data stream): Pipeline machines may be classified in this class. However, Flynn did not give any example of this category.

In order to design a parallel computer, we have to have a communication medium between the processing elements. In parallel machines, an interconnection network either connects processing units within themselves or connects processors to memory units. In other words the interconnection network in parallel computers dictates the communication capabilities and hence the computing power of the system. This dissertation is concerned with the study of certain classes of interconnection networks that play a crucial role in parallel computers. Over the years various interconnection networks have been developed in the literature [L4]. In the next section we begin by describing the topology, functionality and control of various classes of useful networks.

1.2 CLASSIFICATION OF NETWORKS:

The basic building block of an interconnection network is a cross-bar switch. An NxN complete cross-bar switch is a connecting network of N inputs and N outputs as shown in Figure 1.3(a). The intersection of the i^{th} input line and j^{th} output line is a cross-point switch. Each switch has two states, Figure 1.3(b). If the switch state is "on", then there is a connecting link between input i and output j, $i \neq j$. If the switch state is "off", then input i is connected to output i. It is obvious that in a complete NxN cross-bar switch there are N^2 cross-point switches.

Definition 1.1:

Let T be a set of terminals, $T = \{0,1, \ldots ,N-1\}$. A permutation P on the set T is a 1-1 and onto function

$$P: T \text{-----} > T.$$

A switching network realizes the permutation P if the input terminal i can be connected to the output terminal P(i) by proper setting of the switches, where i, and P(i) belong to T.

Consider any permutation P. By proper setting of the switches at the intersection of the i^{th} input line and $P(i)^{th}$ output line of a cross-bar, $i=0,2, \ldots ,N-1$,P can be realized.

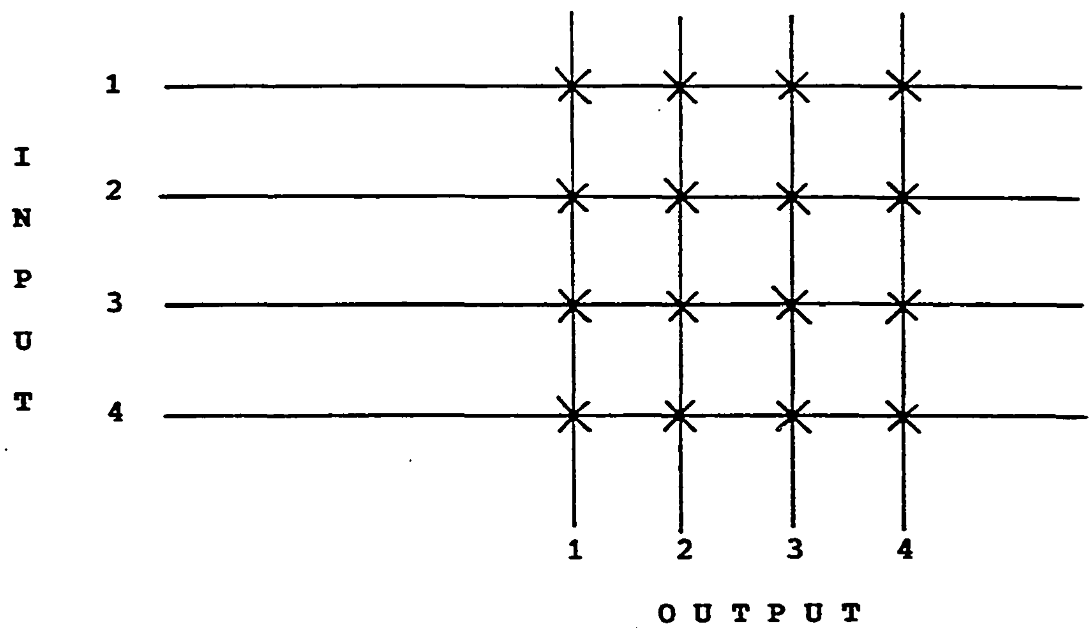
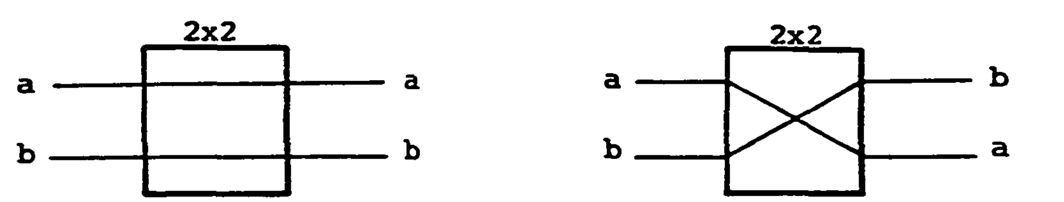


Figure 1.3(a): An example of a cross-bar switch with N=4.
X represents the cross-point switching elements.



Through-state or "0"-state

Crossed-state or "1"-state

Figure 1.3(b): A representation of the 2x2 cross-bar switch.

Graphically, an NxN complete cross-bar switch can be represented as a complete bipartite graph $K_{N,N}$, Figure 1.4, refers to a complete 4x4 cross-bar switch and Figure 1.5, refers to a complete 3x2 cross-bar switch and the corresponding graph. Similarly, an incomplete cross-bar switch is one which does not have a switching element at every intersection point of the input/output lines. Given a specific permutation P the problem of finding the setting for the switches in a cross-bar is called control or routing routine. Given a permutation, the routing algorithm is trivial on a cross-bar switch. Furthermore each input-output path goes through only one switch and there is only one switching delay for a pair of input and output. These are the two principal advantages of the cross-bar. Regardless of supporting a high data rate, a cross-bar switch is not practical for interconnecting a large number of input/output ports. The number of cross-points needed for a cross-bar increases with the square of the number of modules connected to it and hence the cross-bar is very expensive for very large systems. A cross-bar would probably cost more than the rest of the system components combined. Therefore, it is very difficult to justify the use of cross-bars for large systems.

Another building block in the design of interconnection networks is the time-shared bus. A single time-shared bus can provide flexible, inexpensive communication among a

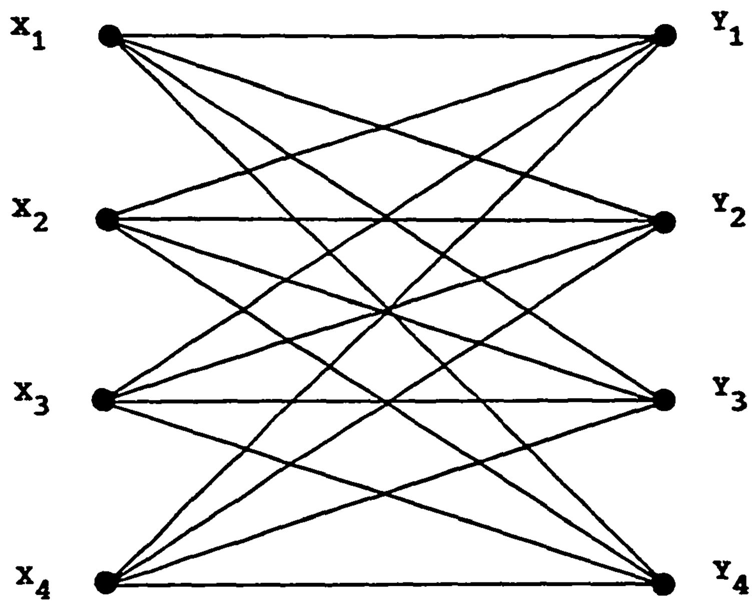


Figure 1.4: A graph model of 4x4 complete cross-bar switch. The node X_i (Y_i) correspond i^{th} input(output) terminal of cross-bar switch, i=1,2,3,4.

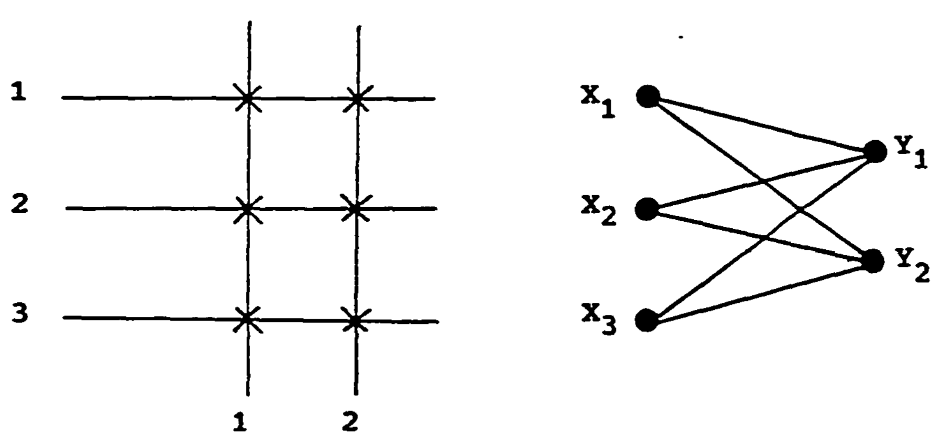


Figure 1.5: An example of complete 3x2 cross-bar switch and its graph.

small number of modules but bus connection problems make this approach impractical for large systems. As the number of modules on the bus increases, bus utilization increases causes more waiting time for a module to use a nonbusy bus.

A multiple time-shared bus on the other hand has problems similar to those of the cross-bar switch. In other words, the switching points which enable each module to be connected with any bus are arranged in a cross-bar configuration. Since the maximum data rate in a bus is fixed, the number of buses grows proportionally with the total number of modules and the number of switching points increases with the square of this number. This further increases the network cost and the time required for signals to propagate across a bus.

These interconnection schemes, i.e. the cross-bar and time-shared bus, are not desirable for general purpose systems with very large number of sources because the cost grows rapidly with the size of the system. Between these extremes there are many interesting classes of cost efficient networks. In the following subsection we briefly summarize their classification by system size. Having the cross-bar as the basic building block, we now classify the interconnection networks as follows [A2,F1,L5,S1,T1,T2,W5].

1.2.1 Topological Classification:

A network can be represented as a graph of nodes(input/output terminals and switches) and edges (interconnecting links). The topology of a network defines various physical components of the network such as the number of input/output terminals, the number and size of the switches and the way of interconnection between these components.

An interconnection network is called a STATIC network if there is no cross-bar switch involved as a part of the network. Thus, in a static network all units are connected through dedicated links. An interconnection network is called a DYNAMIC network if there is one or more cross-bar switches involved.

Static networks can be classified further based on the layout of the network. Figure 1.6, illustrates a one dimensional or linear array. This type of interconnection is used in SIMD machines and pipelined computers. Figures 1.7(a,b,c,d,e) illustrate two-dimensional arrays of standard patterns such as trees, stars, rings, mesh connected, and the systolic arrays, respectively. These types of networks are used in the design of special purpose machines. Figure 1.8, illustrates multi-dimensional array. Multi-dimensional cubes [Pease 1977] and cube connected cycles (Preparatta and Vuillemin 1979) have also been used to develope a number of parallel algorithms.

Figure 1.6: An example of a linear array of processing elements PE_i, i=1,2, ... ,N.

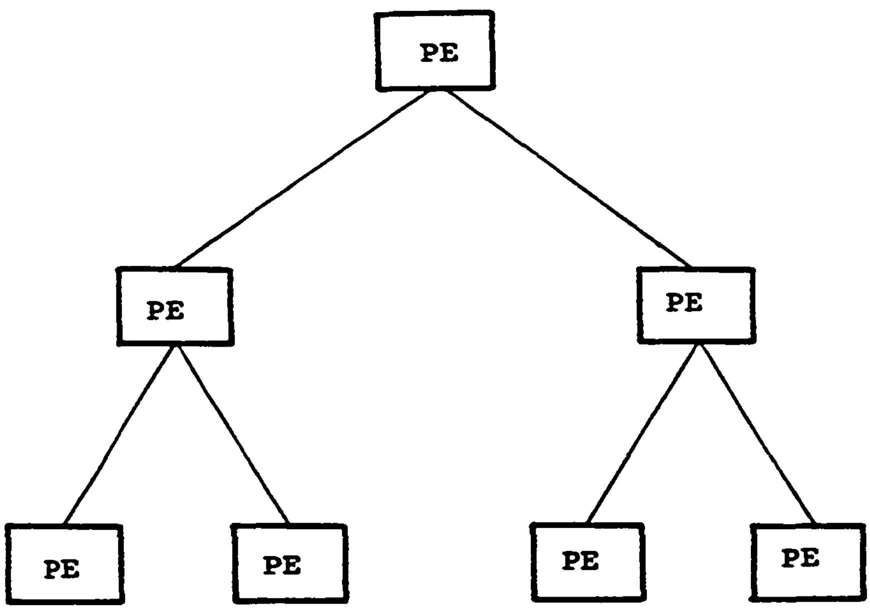


Figure 1.7(a): TREE, An example of a two dimensional array of processing elements.

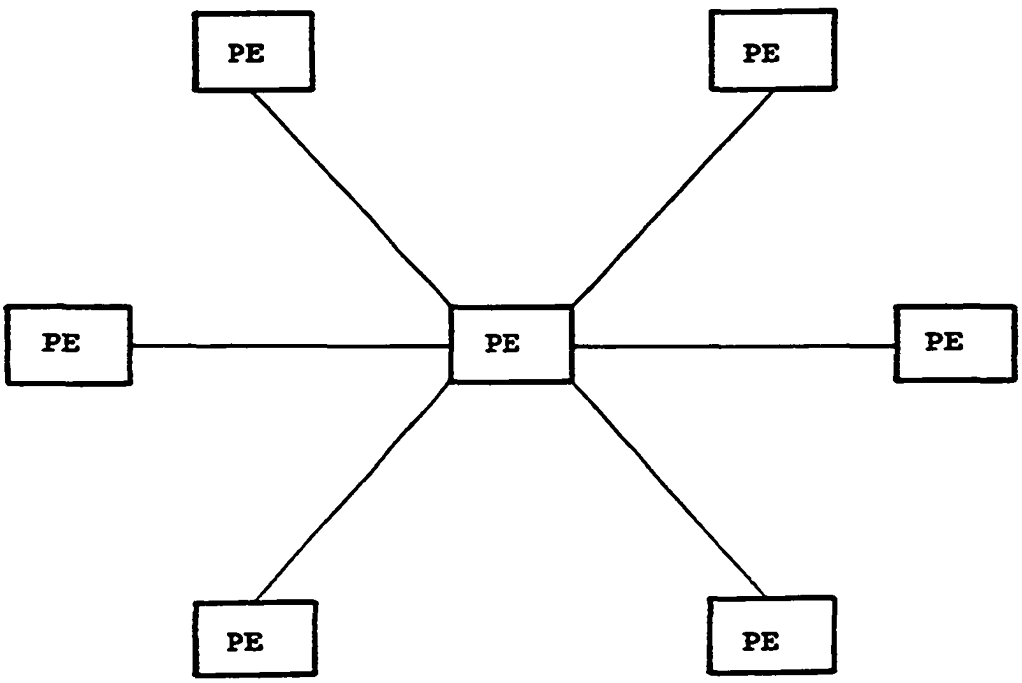


Figure 1.7(b): STAR, An example of a two dimensional array of processing elements.

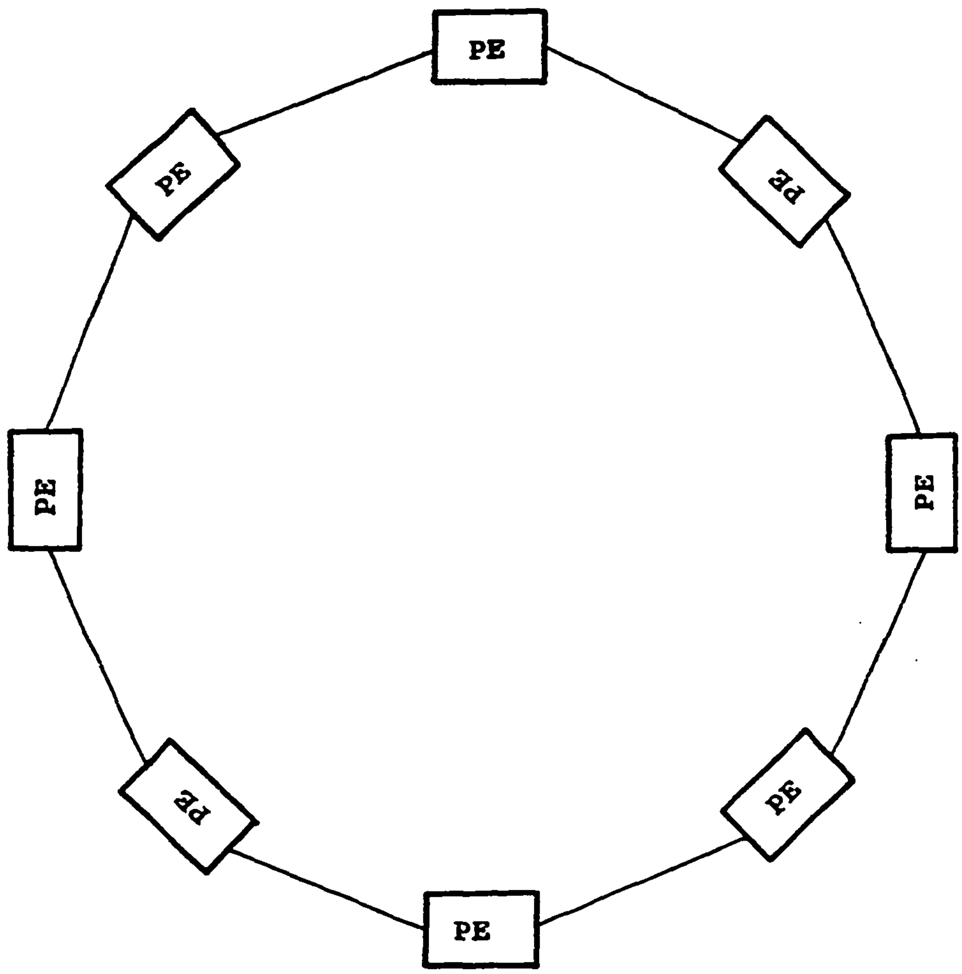


Figure 1.7(c): RING, An example of a two dimensional array of processing elements.

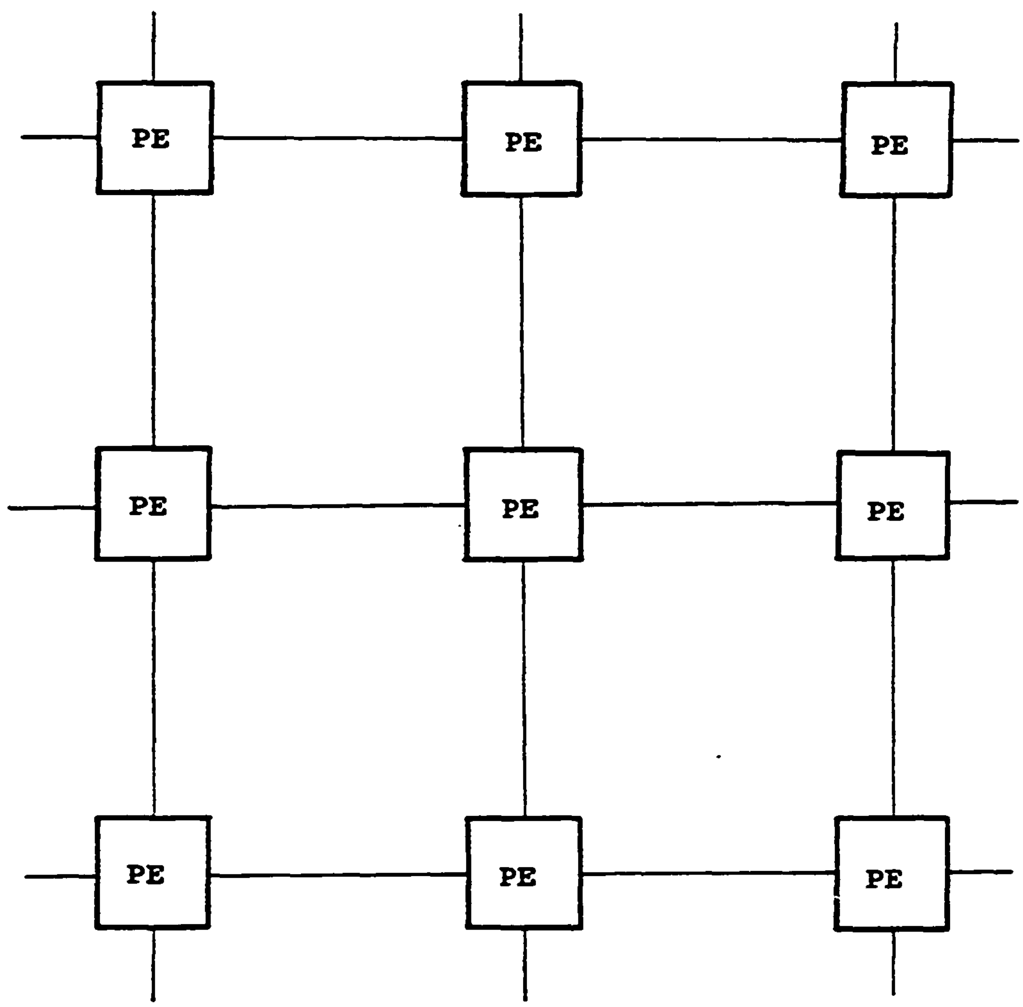


Figure 1.7(d): MESH-CONNECTED, An example of a two dimensional array of processing elements.

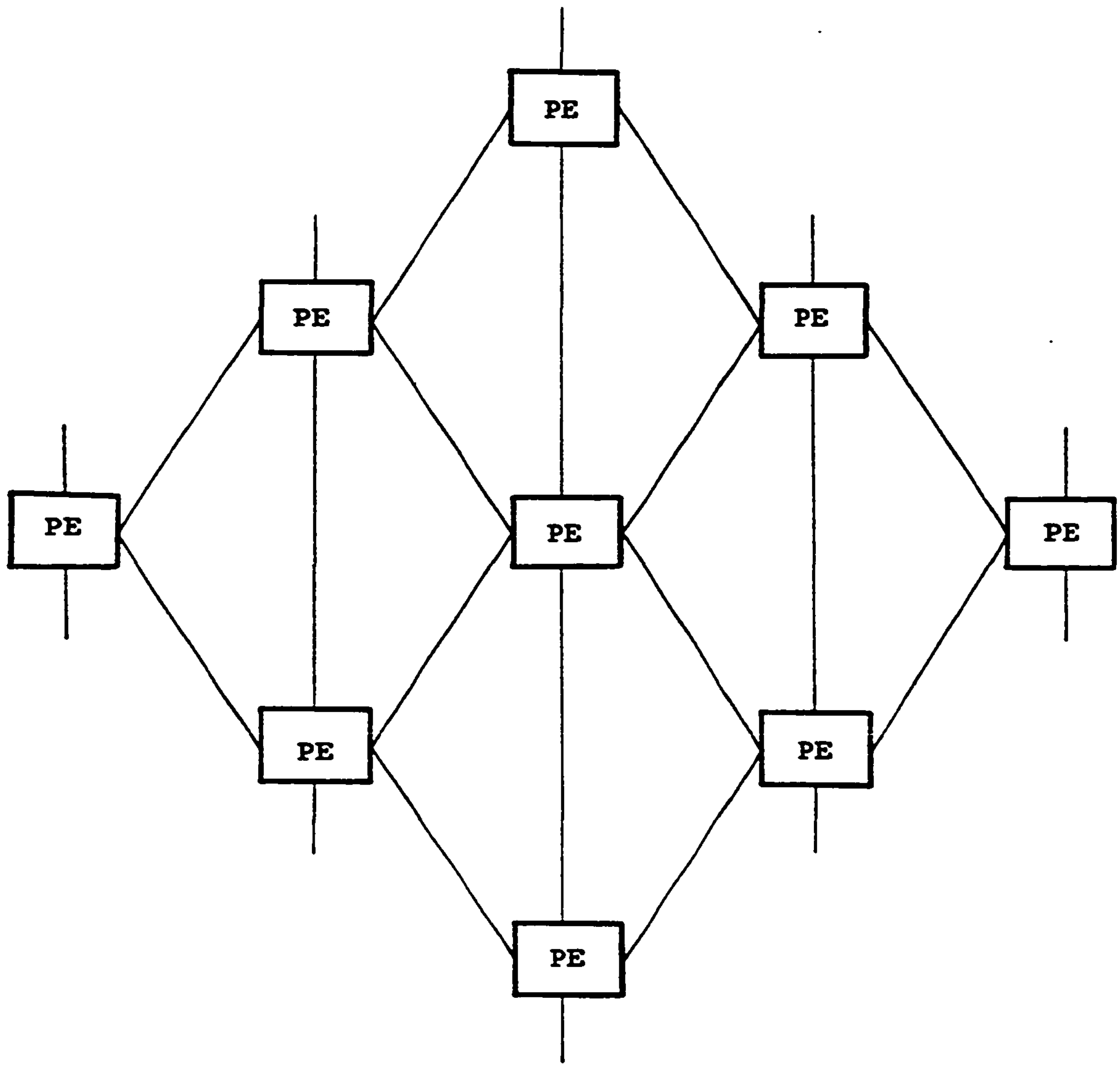


Figure 1.7(e): SYSTOLIC ARRAY, An example of a two dimensional array of processing elements.

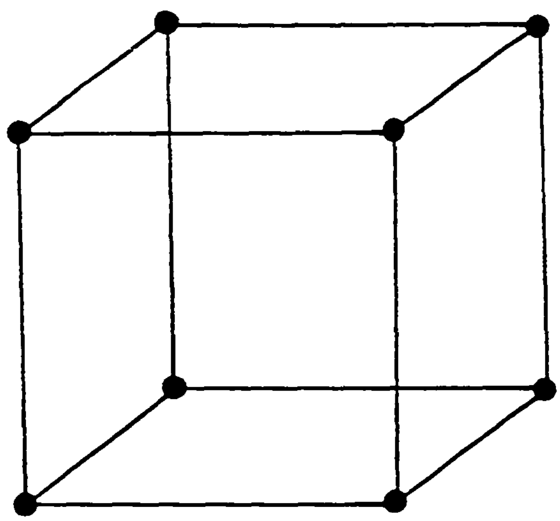


Figure 1.8(a): An example of a 3-cube.

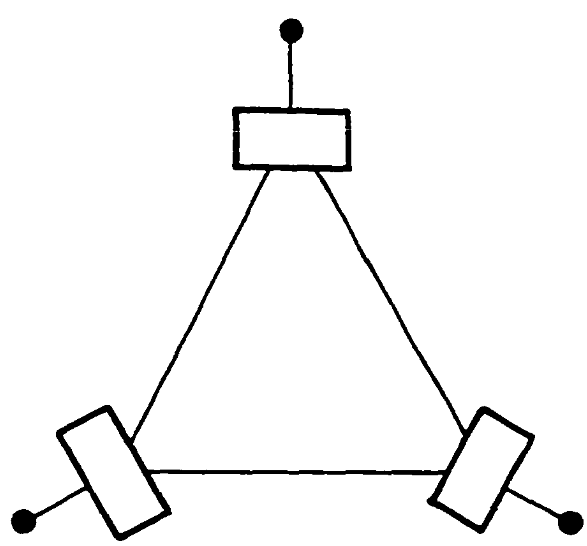


Figure 1.8(b): An example of a 3-cycle.

Figure 1.8: If each vertex in the cube is replaced by a 3-cycle, it becomes a 3-dimensional cube connected cycle.

Dynamic networks can be classified into two subgroups namely single stage and multi-stage networks. Figure 1.9, illustrates a single stage and its associated graph. A single stage network has a bank or stage of (small size such as 2x2) cross-bar switches. An NxN full cross-bar switch is a trivial example of a single stage network.

A multi-stage dynamic network contains more than one stage. The outputs of stage i are connected to the inputs of stage i+1 through a link permutation. Figure 1.10 and Figure 1.11 illustrate examples of one stage and multi-stage networks and their associated graphs.

1.2.2 Technological Classification:

The interconnection networks can be further classified based on their switching modes, namely, CIRCUIT and PACKET switching. In circuit switching a circuit or dedicated path makes a connection between a source and a destination by proper setting of the switches. In packet switching the set of data is divided into a number of slices called packets of certain fixed size. Each packet has its destination address and goes through the network until it reaches the destination. Circuit switching is the most popular switching mode for computer communication and telephone networks. Packet switching is very commonly used in computer communication networks.

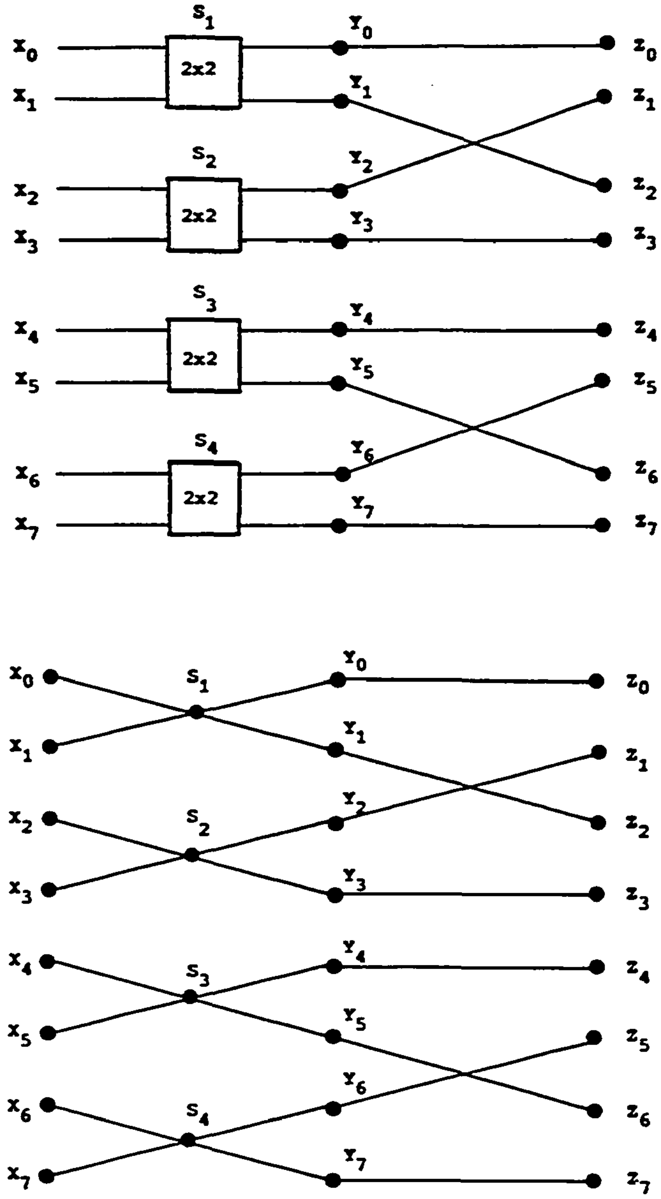


Figure 1.9: An example of a single stage dynamic network without output permutation and its graph. S_i is a 2x2 cross-bar switch for i=1,2,3,4.

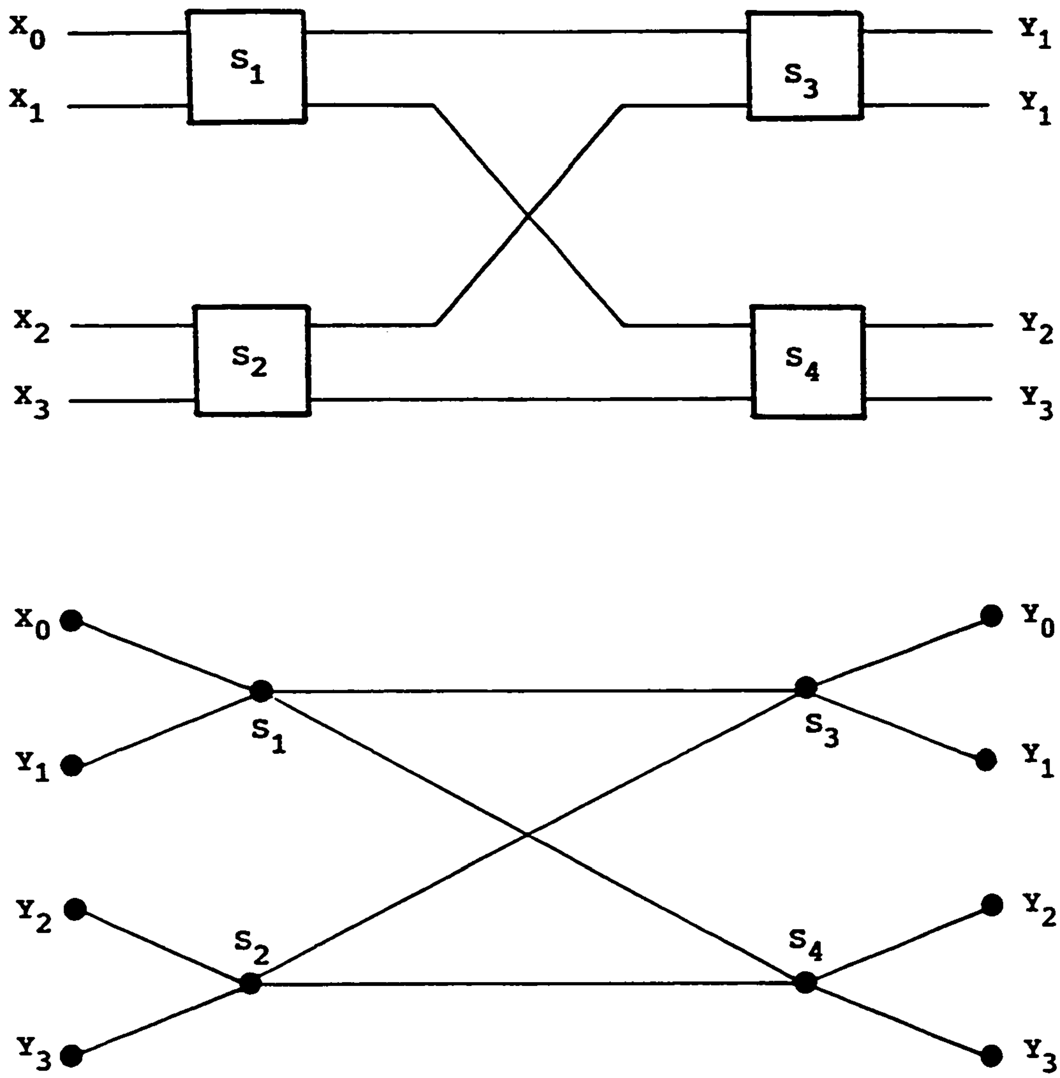


Figure 1.10: An example of a two stage dynamic network without output permutation and its graph. S_i is a 2x2 cross-bar switch for i=1,2,3,4.

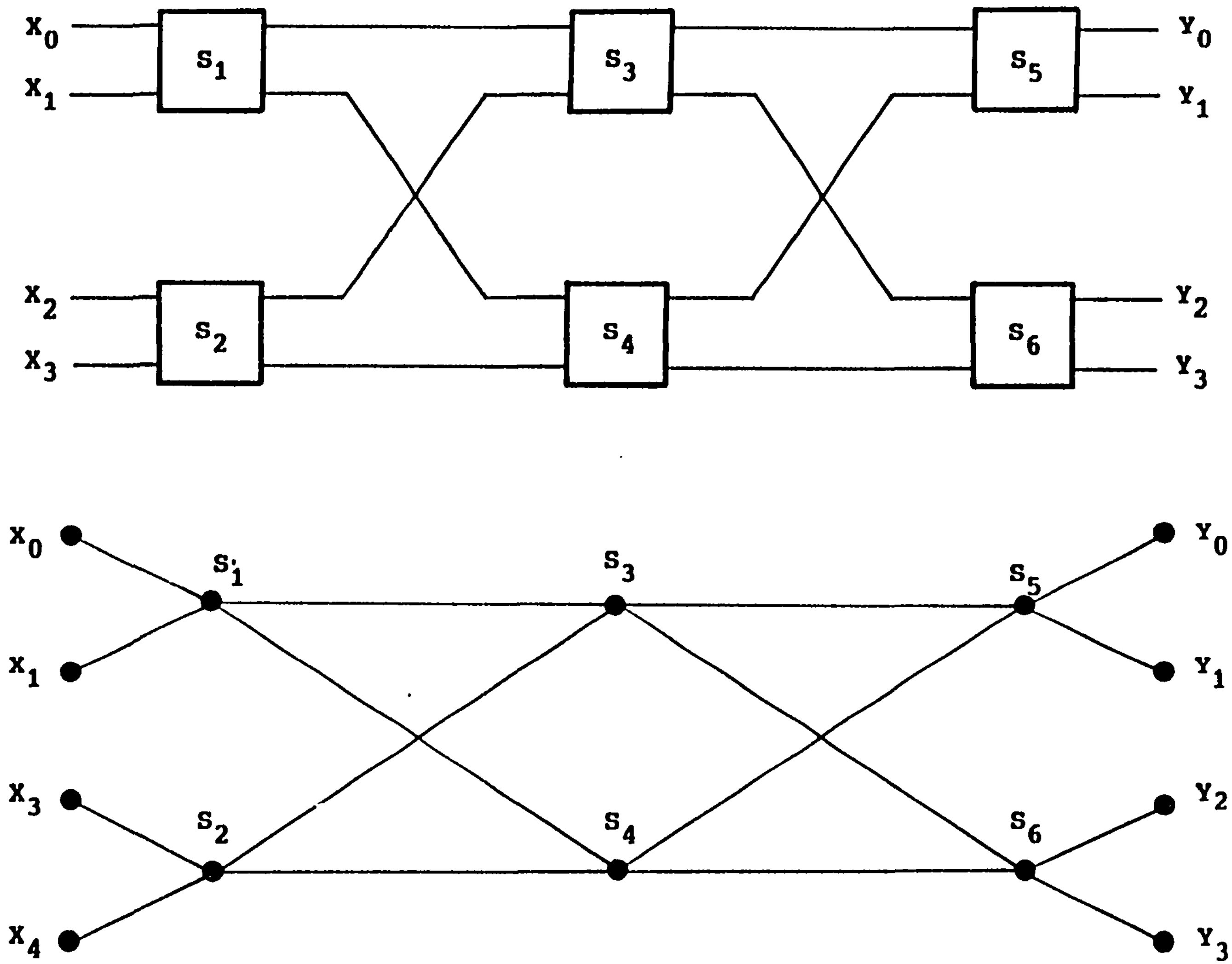


Figure 1.11: An example of a three stage dynamic network without output permutation and its graph. S_i is a 2x2 cross-bar switch for i=1 to 6.

1.2.3 Functional Classification:

The functionality of a multi-stage network depends on the number of edge disjoint paths between various pairs of sources and destinations. This relates to the different interconnection patterns that can be realized by the network.

Definition 1.2:

Let $\Psi(N)$ be the set of all permutations of N objects, clearly $|\Psi(N)| = N!$, and let $K \subseteq \Psi(N)$. An NxN interconnection network is called a K-permutation network if for each K there exists at least one set of N edges-disjoint paths between the input terminal i and the output terminal P(i) for $i=1,2, \ldots ,N$.

Definition 1.3:

A K-permutation network is referred to asa blocking network if $|K| < N!$. In other words if there is at least one permutation which is not realized by the network. On the other hand, a K-permutation network is referred to as a non-blocking network if $|K|=N!$, as in Figure 1.12. Clearly, a non-blocking network realizes all possible permutations. Figure 1.13, shows that the permutation

$$p=\begin{pmatrix} 0 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 \end{pmatrix}$$

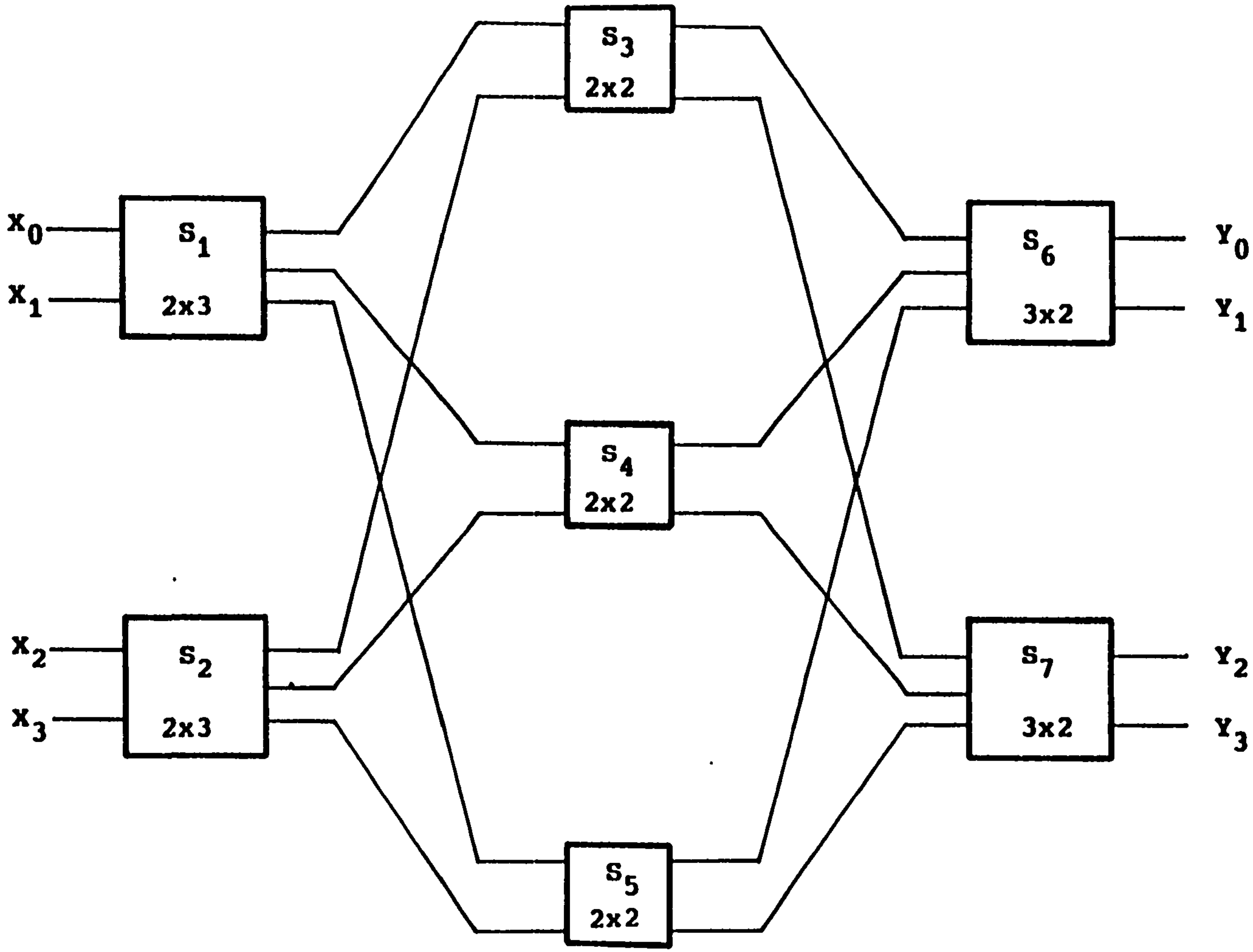


Figure 1.12: An example of non-blocking interconnection network. Each component switch is a complete cross-bar switch of specified size.

cannot be realized by that network. Blocking networks of various kind, Banyan[G1], Omega[L3], Delta[P2], and Baseline[W3] have been extensively studied.

Assume that the input terminal i is already connected to the output terminal p(i) for some i, and suppose we now want to connect terminal j to p(j). It may be necessary to reroute or rearrange the previous setting to make a new connection. This need for rearrangement leads to the following definition.

Definition 1.4:

If a permutation network realizes all possible permutations(perhaps with some rearrangement) then it is called a <u>rearrangeable network</u>.

A non-blocking network does not require any rearrangement of paths for setting up all permutations. Figure 1.11, is a rearrangeable network and Figure 1.12, is a non-blocking network. It is clear that every non-blocking network is rearrangeable but not conversely. Rearrangeable networks require less number of switching points compared to non-blocking networks but the control of the rearrangeable networks is more involved compared to that of non-blocking networks. [C1,L5,O1,P5]. These three and other classes of switching networks are discussed by Feng[F1], Seigel[S2] and Thurber[T2].

Definition 1.5:

Let K be the number of permutations realized by a network. The combinatorial power (CP) is defined as CP=K/N!.

A multi-stage NxN network is said to be UNIVERSAL if it realizes the set of all permutations on N objects. Clearly, for rearrangeable networks and non-blocking networks CP=1 and for blocking networks CP $<$ 1. Examples of blocking networks include base-line networks[F1,W2], Figure 1.14, Omega networks[L3], Figure 1.15, Indirect Binary n-Cube networks[P3], Figure 1.16, etc. A Benes network[B6] is an example of a rearrangeable network, Figure 1.17, and Clos network[C1] is an example of non-blocking network, Figure 1.18.

1.3 SCOPE OF THE DISSERTATION:

An NxN (N inputs and N outputs) multi-stage switching network is an arrangement of switches and connecting links in which a set of N input terminals can be connected to a set of N output terminals according to some permutation. A number of papers have focused on the universality of a cascade of two or more blocking networks, in general, and shuffle-exchange networks such as Omega networks, in particular. For $N=2^k$ and $k \geq 4$, Parker [P1] showed that a cascade of 3k shuffle-exchange stages is universal . Wu and Feng [W4] later proved the same result using only 3k-1 stages. An open question in this context is whether a cascade of 2k shuffle-exchange stages is universal. In chapter two we derive a new

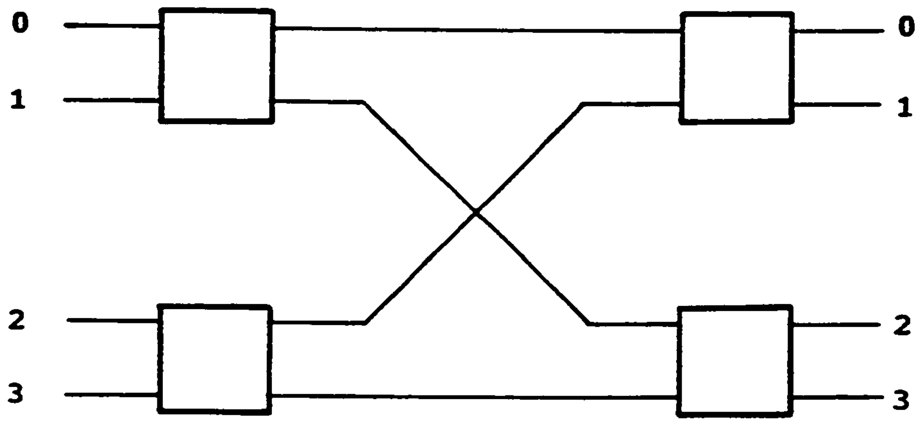


Figure 1.13: An example of a blocking network which does not realize the permutation $\begin{pmatrix} 0 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 \end{pmatrix}$.

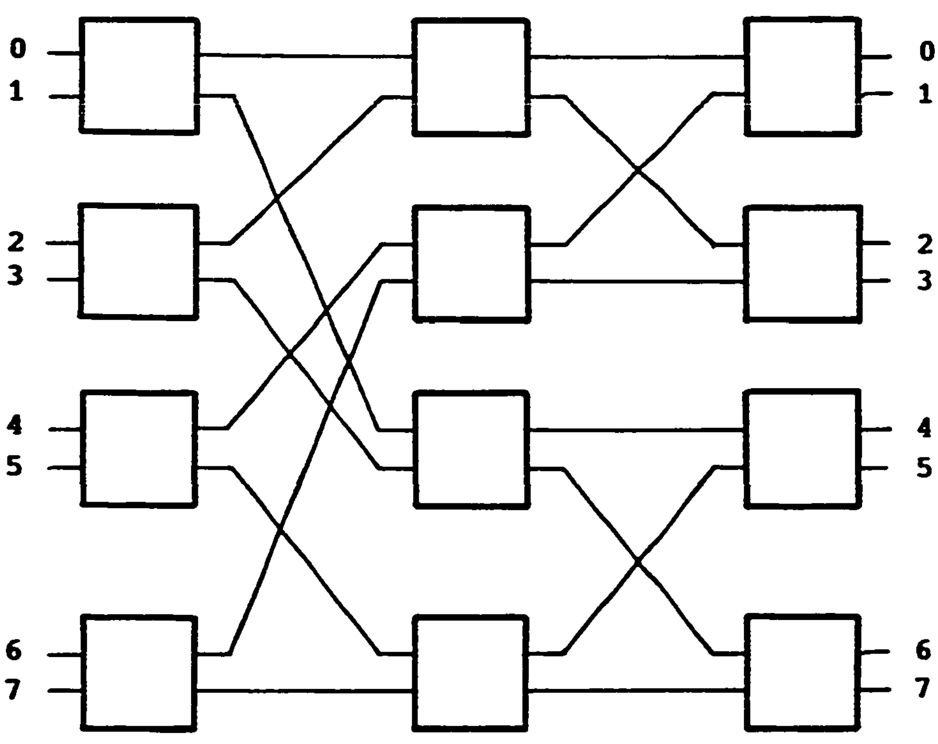


Figure 1.14: A configuration of 8x8 base line network.

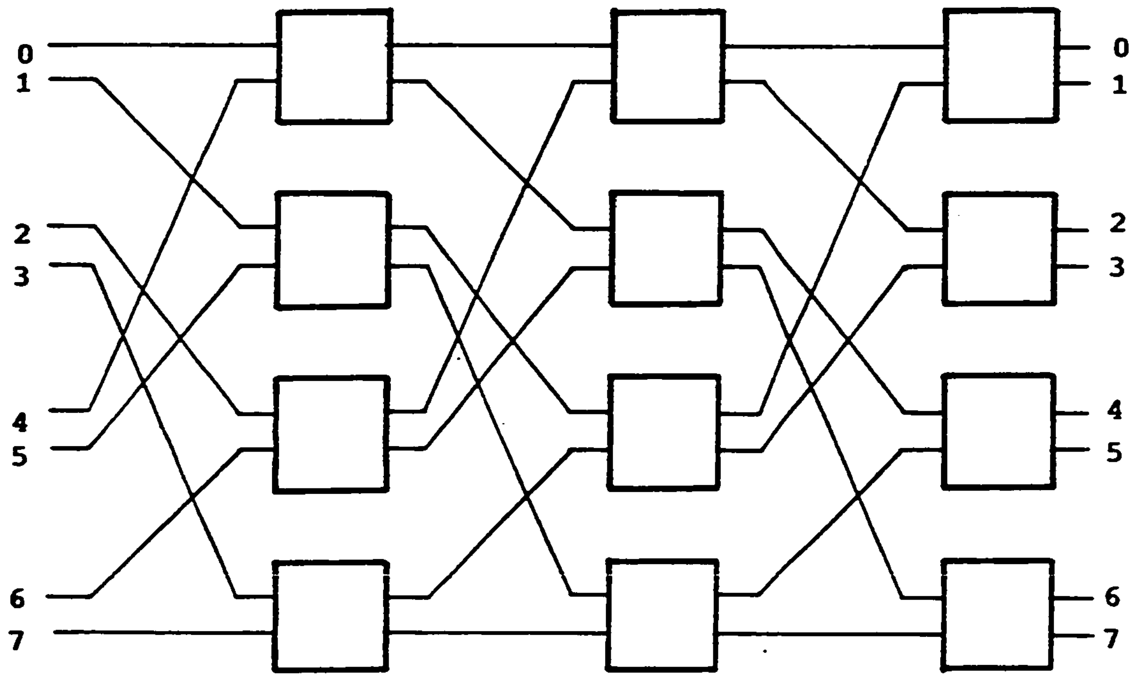


Figure 1.15: An example of 8x8 Omega network.

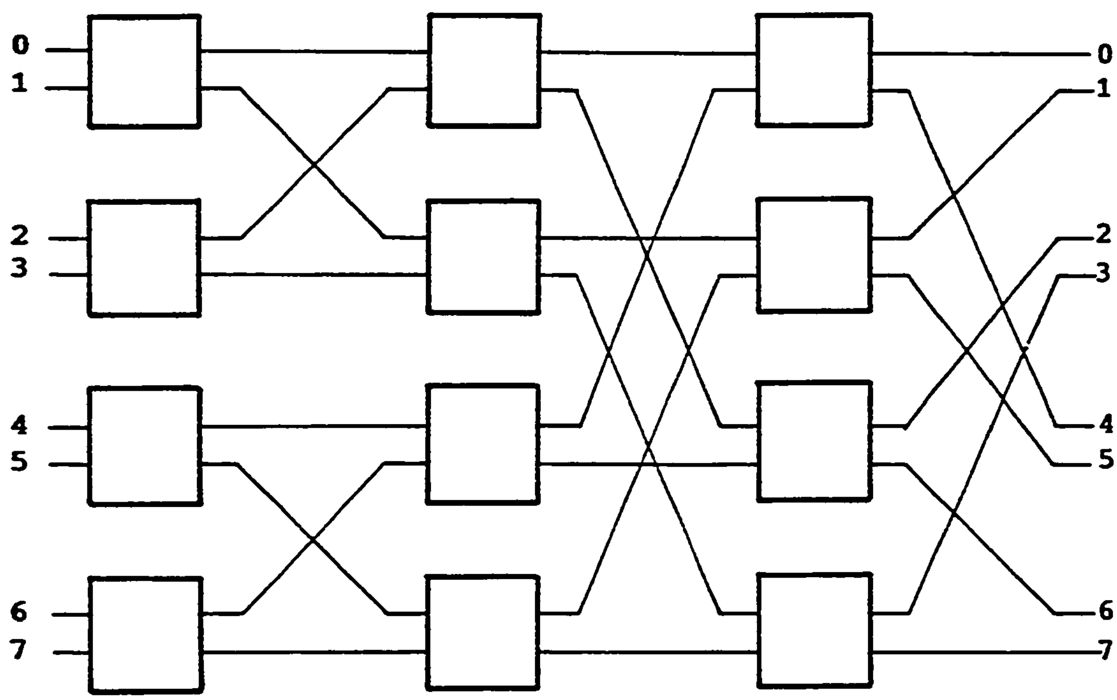


Figure 1.16: A configuration of 8x8 binary n-cube network.

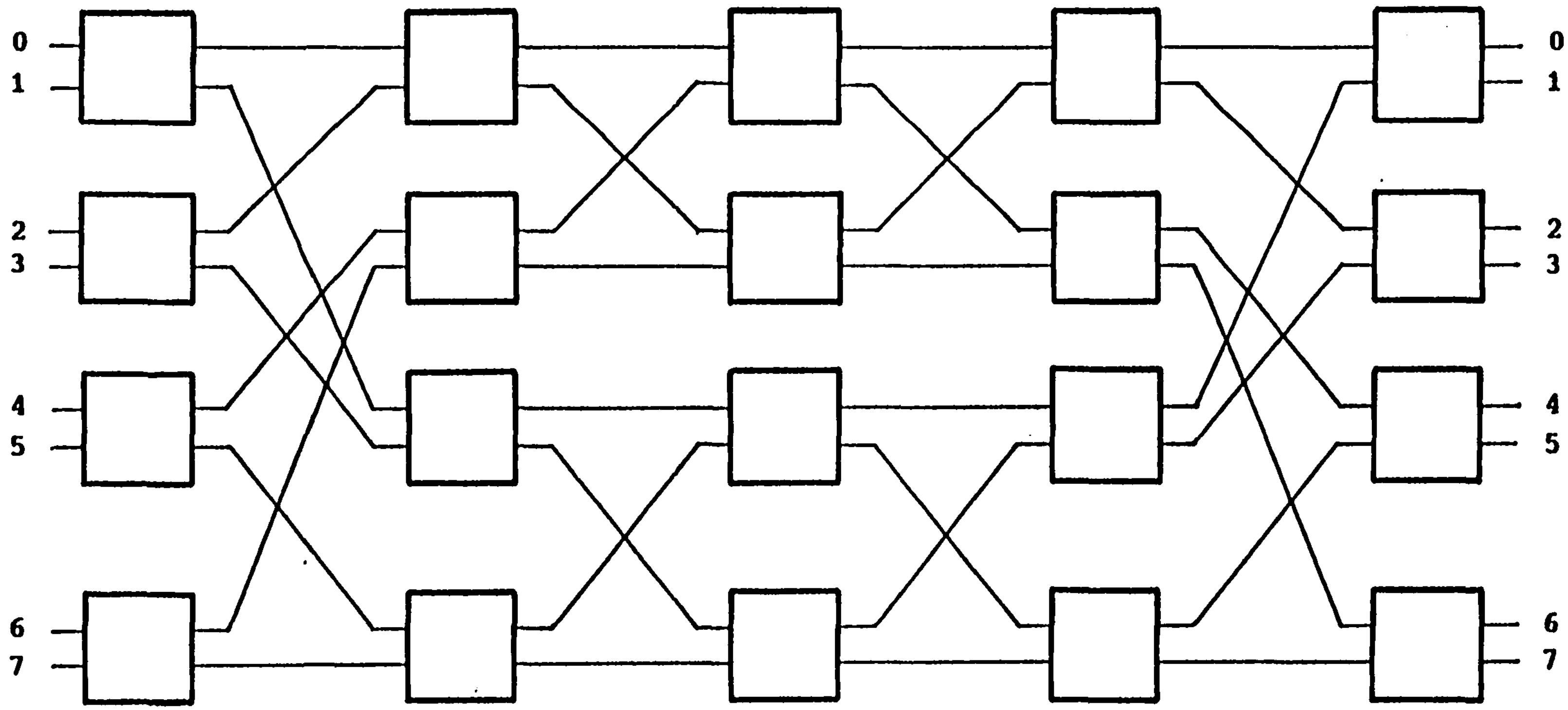


Figure 1.17: An example of 8x8 Benes network.

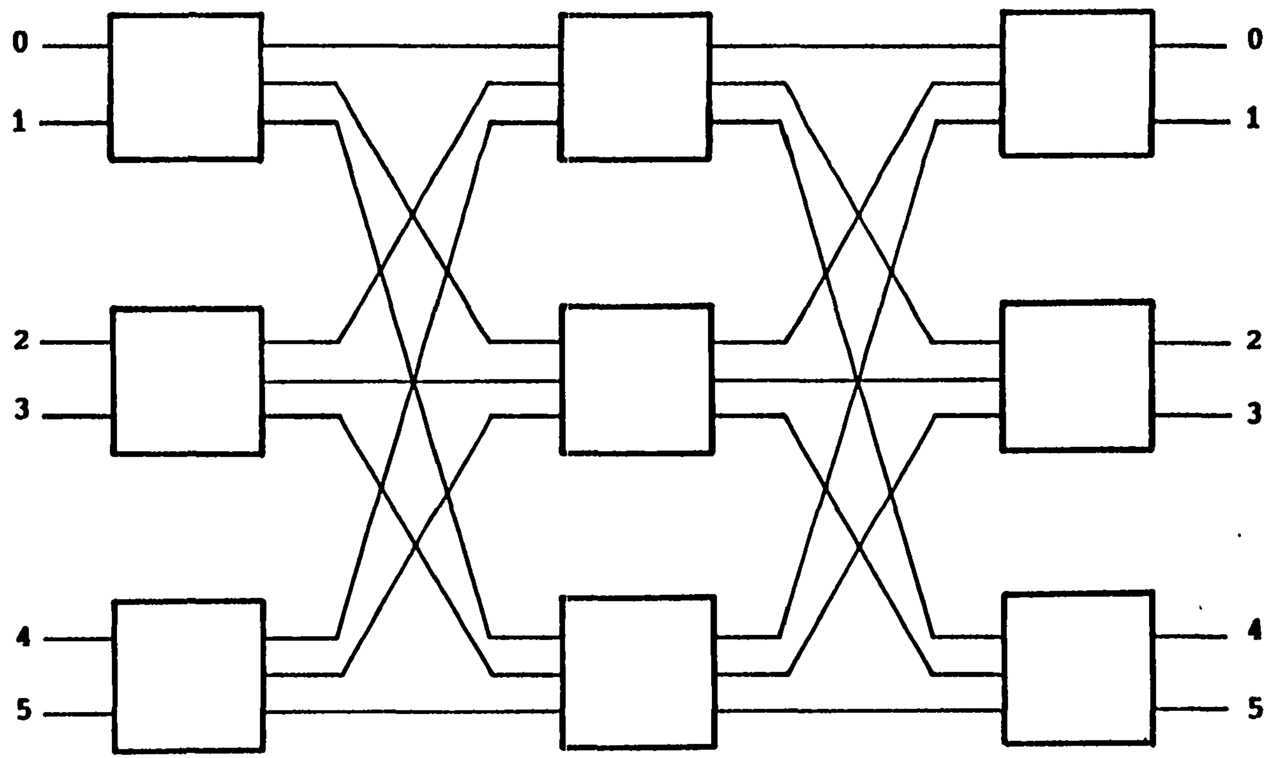


Figure 1.18: A configuration of 6x6 Clos network.

upper bound by showing that 3k-3 shuffle-exchange stages are indeed sufficient for universality. Further, it is shown that the cascade of 2k shuffle-exchange stages cannot be transformed to the Benes type network. This in turn implies that the universality of a cascade of 2k shuffle-exchange stages must be settled outside the theory of Benes type symmetric networks [K3].

Chapter three introduces the concept of L-stage block-structured networks where the link permutations between stages satisfy a fundamental property called the distributive property. It is shown that there exists an intimate relation between this class of networks and the well known SW-Banyan networks[G1] and Delta networks[P2]. This class of block-structured networks with distributive property includes the expanding and contracting SW-Banyan networks. Computing the number of permutation realized by an L-stage block-structured NxM network is by no means trivial and often gives rise to an interesting class of enumeration problems. The trade-off between the number of permutations realized, the path blockage and the cost(measured in the term of the switching elements)for the set of all block-structured NxN networks with distributive property is illustrated through an example when N=16 [K2].

Chapter four analyzes certain new sufficient conditions due to Benes [B5] for the rearrangeability of switching networks. The complexity of this algorithm is analyzed and it is shown that these conditions are excessively sufficient in the sense that there are a number of simple networks which are rearrangeable but do not satisfy the above sufficient conditions. Also, a counter example to the theorem due to Afshar[A1] is given in chapter four. This example relates to the existence of feed-back free networks which are not rearrangeable.

Concluding observations are given in chapter five.

Chapter II

THE UNIVERSALITY OF SHUFFLE-EXCHANGE NETWORKS

2.1 SHUFFLE-EXCHANGE PROPERTIES:

The Shuffle-Exchange networks have been shown to be a very good interconnection network between memory modules and processors [G3,L1,L2,L3,P3,S5]. These networks are constructed of repeated copies of a "perfect shuffle" connection followed by a column of switches of size 2x2. A shuffle-exchange network of size N=8 is shown in Figure 2.1. Each switch can have straight connection, as shown in Figure 2.2(a) or crossed connection as shown in Figure 2.2(b).

We focus our attention on networks made of 2x2 switches. The input /output terminals are numbered 0,1, ... ,N-1 and the switches are numbered 0,1, ... ,N/2 -1. Let P be a permutation on N objects. A network is said to realize the permutation P if there is a proper setting of the switches such that the input terminal i can be connected to the output terminal P(i), for i=0,1, ... ,N-1. Recall that the combinatorial power (CP) of a switching network is defined as a ratio of the number of distinct permutations realized by a given network to the total number of permutations of N objects [B4]. It is clear $0 \leq CP \leq 1$ [B4].

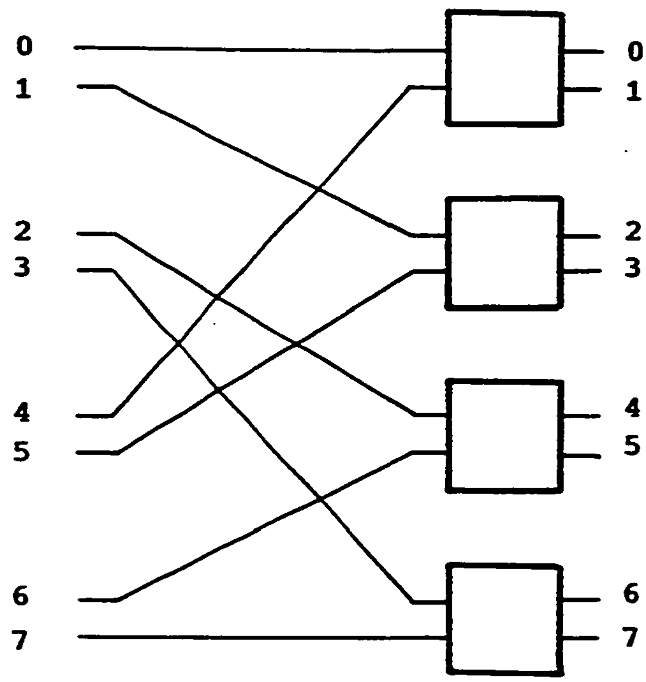


Figure 2.1: A configurartion of Shuffle-exchange network.

a) Straight connection b) Crossed connection

Figure 2.2: Switching connection.

The universality of a cascade of two or more passes of blocking networks has received considerable attention [M1,S3,W4]. Parker [P1] showed that for $N = 2^k$ and $k \geq 4$ a cascade of 3k shuffle-exchange stages is universal. Later WU and Feng [W4] improved the above result by showing that for $k \geq 4$ indeed (3k-1) shuffle-exchange stages are sufficient. For special cases of k=2, 3 shuffle-exchange stages are necessary for universality, while for k=3, 6 shuffle-exchange stages are necessary [P1,L3,W6]. One of the open questions in this context is that for $N = 2^k$ and $k \geq 4$, whether 2k shuffle-exchange stages are universal or not [P1].

In this chapter we focus our attention to the universality of cascades of shuffle-exchange stages. It is shown that for $k \geq 4$, 3k-3 shuffle-exchange stages are universal instead of presently known bound which is 3k-1. Further, since 3k-3 = 2k for k = 3, the method explains why a cascade of two passes of omega network is universal. It is also shown that for $k \geq 4$ a cascade of 2k shuffle-exchange stages is not transformable to Benes type networks. As a result, the universality of cascades of 2k shuffle-exchange stages must be settled outside of the framework of Benes theory of rearrangeable symmetric networks [B4,K3].

2.2 NEW UPPER BOUND:

Our approach is based on transforming a given cascade of shuffle-exchange stages to a Benes type network. The derivation would need the structure of the target which is a Benes network. Let $N = 2^k$, $k \geq 2$. The Benes network is made of $2k-1$ stages. The middle three stages consist of 2^{k-2} independent 4x4 Benes networks, called 4x4 Benes blocks. Figure 2.3(a) shows a typical block and Figure 2.4 shows a Benes network of size N=16. Any such typical 4x4 Benes block is topologically equivalent to the graph shown in Figure 2.3(b). A 4x4 Benes block has a distinguishing feature such that it has three associated pairs of switches connected as shown in Figure 2.3(a).

The transformation consists of finding the number of shuffle-exchange stages needed in forming the required number of independent 4x4 Benes blocks. We establish the following notations. Stages of the network are numbered in ascending order from input stage to the output stage. The switches in each stage are numbered 0 through $N/2 -1$. The inputs and outputs of a switch at position x are numbered 2x and 2x+1, respectively, where $0 \leq x \leq N/2 -1$. Clearly, the even numbered terminals are the top terminals and the odd numbered terminals are bottom terminals of the 2x2 switches. In binary notation, if $x=[x_{k-1}\ x_{k-2} \cdots x_2\ x_1]$ is the position of a switch at a given stage, then the inputs and outputs of this switch are numbered $[x_{k-1}\ x_{k-2} \cdots x_2\ x_1\ 0]$ and

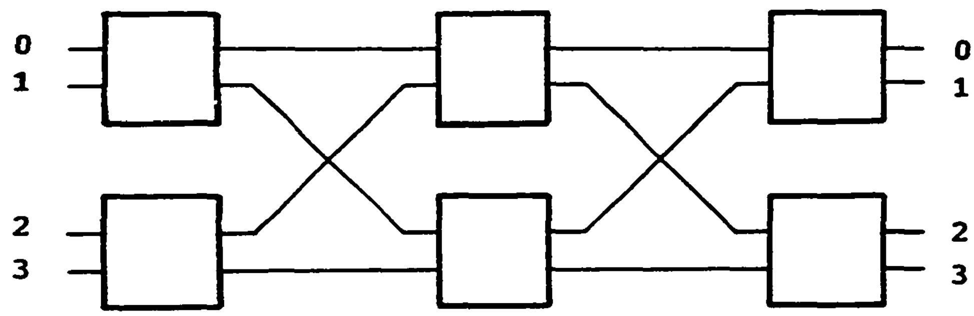


Figure 2.3(a): An example of 4x4 Benes network.

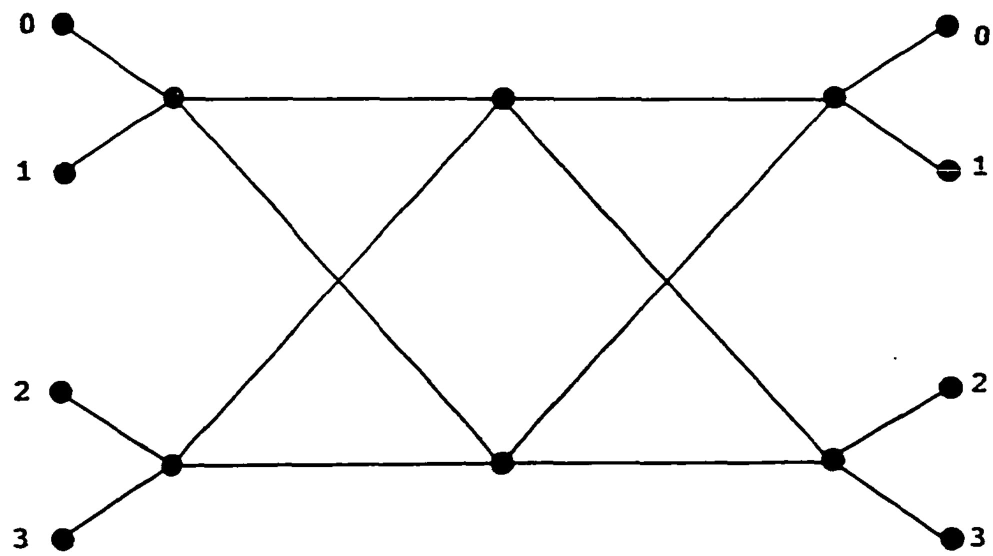


Figure 2.3(b): The graph of a typical block of Benes network.

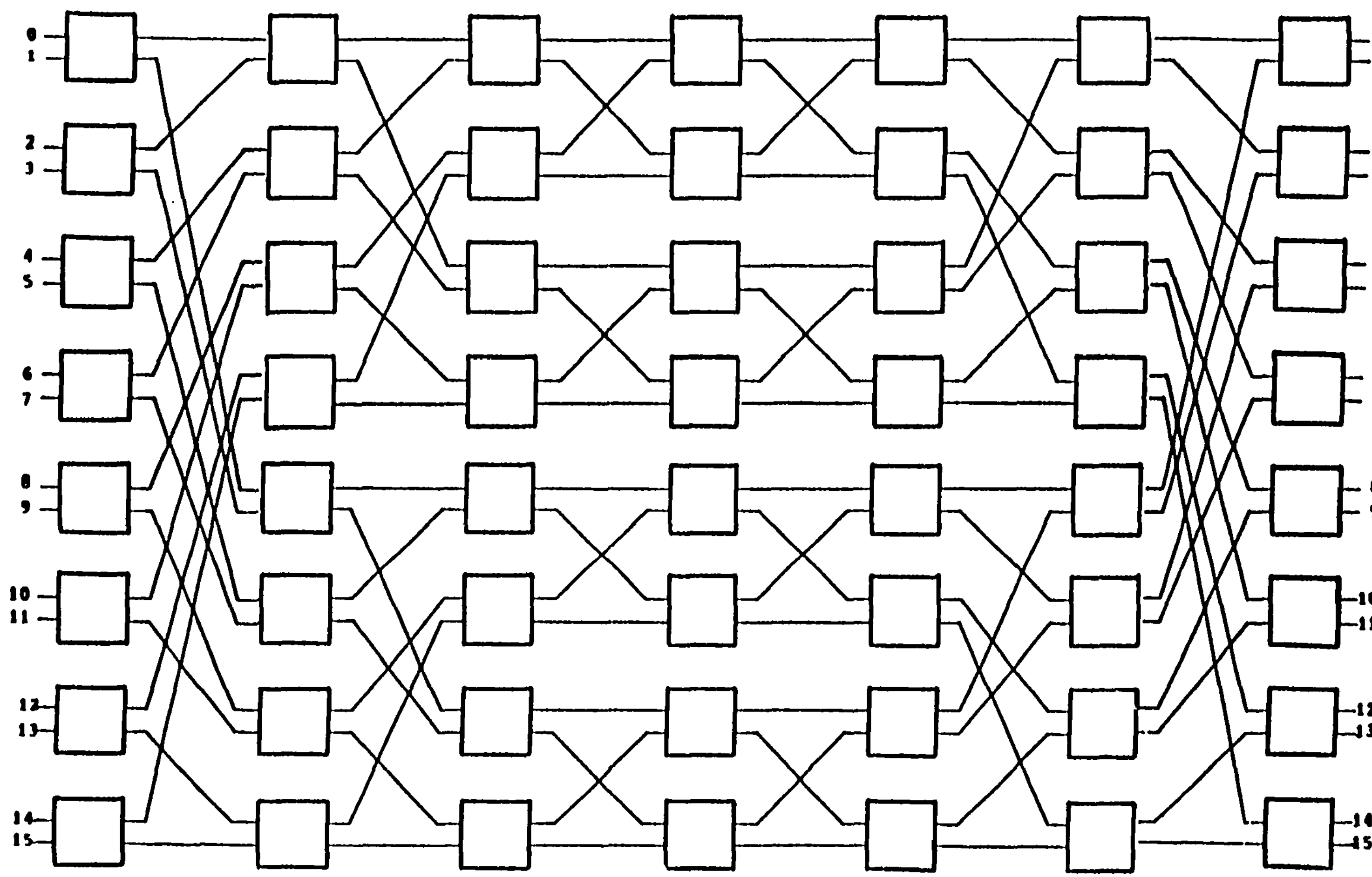


Figure 2.4: An example of 16x16 Benes network consists of 4 Benes blocks.

$[x_{k-1}\ x_{k-2}\ \cdots\ x_2\ x_1\ 1]$. In addition, we need to define the following terminology.

2.2.1 Perfect Shuffle Permutation:

Definition 2.1:

The perfect shuffle permutation σ is defined [S3] by

$$\sigma(x) = \left(2x + \frac{2x}{N}\right) \text{Mod } N$$

then the shuffling corresponds to a circular left shift is $\sigma\,([x_{k-1}\ x_{k-2}\ \cdots\ x_1\ x_0]) = [x_{k-2}\ x_{k-3}\ \cdots\ x_1\ x_0\ x_{k-1}]$, where x is the index of some input line. The unshuffle corresponds to a circular right shift and is given by:

$$\sigma^{-1}([x_{k-1}\ x_{k-2}\ \cdots\ x_1\ x_0]) = [x_0\ x_{k-1}\ \cdots\ x_3\ x_2\ x_1].$$

2.2.2 Exchange Permutation:

Definition 2.2:

The Exchange permutation E is defined as:

$$E([x_{k-1}\ x_{k-2}\ \cdots\ x_1\ x_0\ ]) =$$

$$[x_{k-1}\ x_{k-2}\ \cdots\ x_2\ x_1\ x_0\ ] \quad \text{if x is in through state}$$

or

$$[x_{k-1}\ x_{k-2}\ \cdots\ x_2\ x_1\ \bar{x}_0\ ] \quad \text{if x is in crossed state}$$

where $\bar{x}_0$ is 0 or 1 if x_0 is 1 or 0.

Let $x=[x_{k-1}\ x_{k-2}\ \ldots\ x_2\ y_1\ y_0\ ]$ be an output terminal of a switch at stage i. x is the top output terminal of the switch at position $[x_{k-1}\ x_{k-2}\ \ldots\ x_2\ y_1\ ]$ if $y_0 = 0$ and the bottom output if $y_0 = 1$.

The set of all input terminals at stage i+m of Shuffle-exchange network, $m \geq 1$, which are reachable from x is denoted by:

$$R_m(x) = \sigma(E.\ \sigma)^{m-1}\ (x).$$

Clearly, the number of terminals in this set is 2^{m-1}. Similarly, the set of all input terminals at stage i+k-1 of Shuffle-exchange network, $k \geq 2$, which are reachable from x is

$$R_{k-1}(x) = \sigma(E.\ \sigma)^{k-2}\ (x).$$

which is

$$\{[y_0\ b_1\ b_2\ \ldots\ b_{k-2}\ y_1\ ]\ |\ b_i\ \varepsilon\{0,1\},\ 1 \leq i \leq k-2\}.$$

These 2^{k-2} input terminals are incident on the set of S of 2^{k-2} switches where,

$$S=\{[y_0\ b_1\ b_2\ \ldots\ b_{k-2}\ ]\ |\ b_i\ \varepsilon\{0,1\},\ 1 \leq i \leq k-2\}.$$

Thus, if $y_0 = 0$ the switches in S correspond to the top half of switches at stage (i+k-1) that are numbered 0 through N/4-1, and if $y_0= 1$ the switches in S correspond to the bottom half of switches at stage (i+k-1) that are numbered N/4 through N/2 -1. Furthermore, any input terminal in $R_{k-1}(x)$

is the top input to a switch in S if $y_1 = 0$ and it is a bottom input if $y_1 = 1$. Consequently, this leads us to the following observation.

The two top output terminals $[x_{k-1}\ x_{k-2}\ \ldots\ x_2\ 0\ 0]$ and $[x_{k-1}\ x_{k-2}\ \ldots\ x_2\ 1\ 0]$ of two adjacent switches with numbers $[x_{k-1}\ x_{k-2}\ \ldots\ x_2\ 0]$ and $[x_{k-1}\ x_{k-2}\ \ldots\ x_2\ 1]$ at stage i, respectively, after (k-2) shuffle-exchange stages, can be made as the two inputs to any single switch numbered 0 to N/4 -1 at stage i+k-1. Likewise, the two bottom output terminals $[x_{k-1}\ x_{k-2}\ \ldots\ x_2\ 0\ 1]$ and $[x_{k-1}\ x_{k-2}\ \ldots\ x_2\ 1\ 1]$ of the same set of two adjacent switches can be made as the two inputs to any switch numbered N/4 through N/2 -1 at stage (i+k-1). This can be done by suitably fixing the state of the switches at stages (i+1) through (i+k-2). Two consecutive switches x and x+1, for x=0,2,4, ... ,N/2, can be connected to two switches at stage (i+k-1), as required in the 4x4 Benes block, Figure 2.3(a).

To obtain the first part of the 4x4 Benes block, consider two consecutive switches x and x+1, for x=0,2,4, ... ,N/2, at stage i, they are connected under an un-shuffle permutation to the switches x/2 and $x/2\ +2^{k-2}$, which are from switches of the top part numbered (0 to N/4 -1) and switches of the bottom part(N/4 to N/2 -1), respectively. Thus, by considering a sequence of k+1 stages, one can easily identify the required number of independent 4x4 Benes blocks. The above discussion naturally leads us to the following theorem.

2.2.3 Constructing Independent Benes Blocks

Theorem 2.1:

For $N=2^k$ and $k \geq 4$, a cascade of k+1 shuffle-exchange stages are necessary to obtain all the 2^{k-2} independent 4x4 Benes blocks.

Proof:

It follows from the above argument that the two top terminals, namely, $[x_{k-1}\ x_{k-2}\ \cdots\ x_2\ 0\ 0\ ]$ and $[x_{k-1}\ x_{k-2}\ \cdots\ x_2\ 1\ 0\ ]$, of two consecutive switches $[x_{k-1}\ x_{k-2}\ \cdots\ x_2\ 0\ ]$ and $[x_{k-1}\ x_{k-2}\ \cdots\ x_2\ 1\ ]$, (for x=0,2,4, ... ,N/2) in stage i cannot reach the two input terminals of a switch in stage i+m unless $m \geq k-1$. The desired connections are obtained by fixing the states of the switches in at least (k-2) intermediate stages. Therefore, (k-2)+3=k+1 stages of a shuffle-exchange are necessary for construction of 4x4 Benes blocks. Consider the five stages of shuffle-exchange of Figure 2.5, by setting the switches at stage 3 and 4, four independent Benes blocks of size 4x4 can be formed, namely A, B, C, and D.

2.2.4 Universality of two passes of Omega Networks

Corollary 2.1:

For $N = 2^k$ and $k \geq 4$, a cascade of 2k shuffle-exchange stages cannot be transformed into Benes type network.

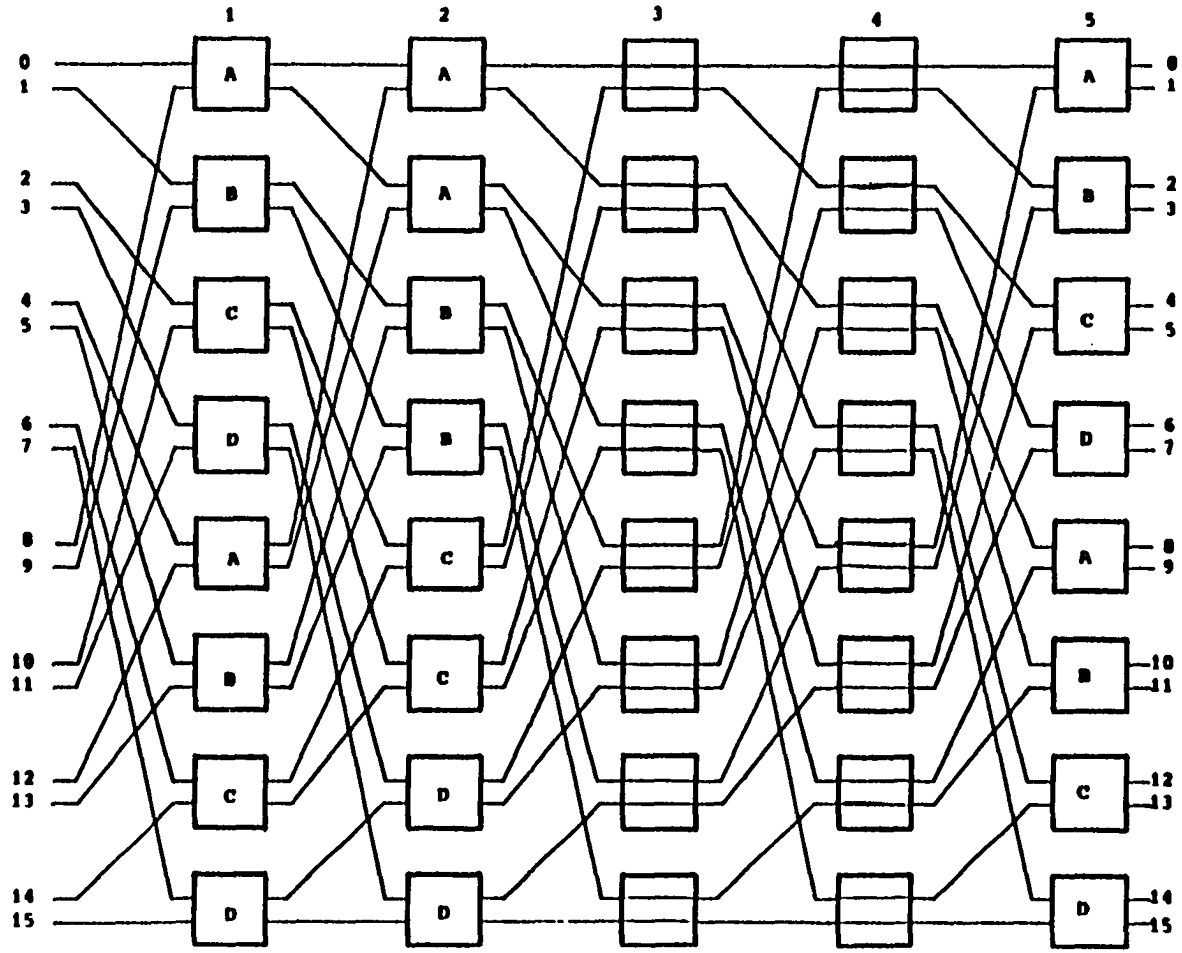


Figure 2.5: Illustration of theorem 2.1 for k=4.

Proof:

(k+1) shuffle-exchange stages are necessary to construct 2 independent 4x4 Benes blocks, which have three stages. Thus, we are left with only 2k-(k+1)+3=k+2 stages. But a Benes network has 2k-1 stages and for $k \geq 4$, $(k+2) < (2k-1)$, the corollary follows.

To this date, the only method known for proving the rearrangeability of a cascade of blocking networks is to transform it into a Benes type network [K1,P1,S2,W2,W4]. The above corollary has a direct impact on the well known open question namely whether a cascade of 2k shuffle-exchange stages is universal or not. Since it cannot be transformed into a Benes type network, the universality of such a network must be settled outside of the framework of the theory of symmetric rearrangeable Benes type networks.

From the above discussion it follows that more than 2k shuffle-exchange stages are necessary for universality. In the following section, we show that indeed 3k-3 shuffle-exchange stages are sufficient.

2.3 A TRANSFORMATION TO BENES NETWORK

Given 3k-3 stages of shuffle-exchange, the procedure has two phases. The first phase defines a renumbering of the switches in some stages and the second phase sets the states of the switches in k-2 stages to obtain all independent (4x4) blocks. The derivation would need the following terminology.

2.3.1 Sub-Shuffle, Super-Shuffle and Un-Shuffle Permutations:

Definition 2.3:

Let $[x_{k-1}\ x_{k-2}\ \cdots\ x_2\ x_1]$ be the binary representation of x, where $0 \leq x < N/2$, then the bit-reversal permutation, ρ , is defined by,

$$\rho([x_{k-1}\ x_{k-2}\ \cdots\ x_2\ x_1]) = [x_1\ x_2\ \cdots\ x_{k-2}\ x_{k-1}].$$

The sub-shuffle permutation , $\sigma_{(i)}$, is defined by:

$$\sigma_{(i)}(x) = \sigma_{(i)}([x_{k-1}\ x_{k-2}\ \cdots\ x_{i+1}\ x_i\ x_{i-1}\ \cdots\ x_2\ x_1]) = [x_{k-1}\ x_{k-2}\ \cdots\ x_{i+1}\ x_{i-1}\ x_{i-2}\ \cdots\ x_2\ x_1\ x_i].$$

The super-shuffle permutation of x is defined by:

$$\sigma^{(i)}(x) = \sigma^{(i)}([x_{k-1}\ x_{k-2}\ \cdots\ x_{k-i+1}\ x_{k-i}\ x_{k-i-1}\ \cdots\ x_2\ x_1]) = [x_{-2}\ x_{k-3}\ \cdots\ x_{k-i+1}\ x_{k-i}\ x_{k-1}\ x_{k-i-1}\ \cdots\ x_2\ x_1].$$

Clearly, the super-unshuffle permutation of x is defined by:

$$\sigma^{(i)^{-1}}(x) = \sigma^{(i)^{-1}}([x_{k-1}\ x_{k-2}\ \cdots\ x_{k-i+1}\ x_{k-i}\ x_{k-i-1}\ \cdots\ x_2\ x_1] = [x_{k-i}\ x_{k-1}\ x_{k-2}\ \cdots\ x_{k-i+1}\ x_{k-i-1}\ \cdots\ x_2\ x_1]).$$

In other words, the sub-unshuffle and super-unshuffle permutation of x is the unshuffle permutation on trailing and leading i bits of x.

Definition 2.4:

Let S be a permutation of m objects, i.e.

$$S=\begin{pmatrix} 0 & 1 & 2 & \cdots & m-1 \\ S(0) & S(1) & S(2) & \cdots & S(m-1) \end{pmatrix}$$

for $0 \le x \le m-1$, then we define a composition $P_1(S(x))$ as:

$P_1(S(x))=S(x)$ if x is even.

$P_1(S(x))=S((x+m/2) \text{Mod } m)$ if x is odd.

In other words, $P_1(S(x))$ swaps $S(x)$ with $S(x+m/2)$ if x is odd, and $1 \le x < m/2$, for example,

x	0	1	2	3	4	5	6	7
S(x)	0	1	6	7	2	3	4	5
$P_1(S(x))$	0	3	6	5	2	1	4	7

Define a permutation $P_2(S(x))$ for $0 \le x \le m-1$ as follows,

For $0 \le x$ lt $m/2$

$P_2(S(x))=S(x)$ if $\lfloor x/2 \rfloor$ is even.

$P_2(S(x))=S((x+m/4) \text{Mod } m/2)$ if $\lfloor x/2 \rfloor$ is odd.

For $m/2 \le x$ lt $m-1$

$P_2(S(x))=S(x)$ if $\lfloor x/2 \rfloor$ is even.

$P_2(S(x))=S(m/2+(x+m/4) \text{Mod } m/2)$ if $\lfloor x/2 \rfloor$ is odd.

For example:

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S(x)	0	1	8	9	4	5	12	13	2	3	10	11	6	7	14	15
$P_2(S(x))$	0	1	12	13	4	5	8	9	2	3	14	15	6	7	10	11

Similarly, define a permutation $P_4(S(x))$ for $0 \leq x \leq m-1$ by:

$P_4(S(x))=S(x)$ if $\lfloor x/4 \rfloor$ is even.

$P_4(S(x))=S((x+m/2) \text{Mod } m)$ if $\lfloor x/4 \rfloor$ is odd.

We now describe the transformation using the following algorithm. In all the following $0 \leq x < N/2$.

Algorithm 2.1:

Step 1: Divide 3k-3 shuffle-exchange stages network into three groups:

-group I consists of the first k stages,

-group II consists of the next k-2 stages,

-group III consists of the last k-1 stages,

Step 2: Renumbering of the switches in group I.

a) for stage 1 to 2 do

$$S(x) := x$$

b) for stage i:=3 to k-1 do

$S(x) := \sigma_{(i)}(\sigma_{(i+1)}(\sigma_{(i+2)} \quad \cdots \quad \sigma_{(k-1)}(x)))$.

c) for stage k do

$$S(x) := x$$

<u>Step 3</u>: Renumbering of the switches in group III.

a) for stages 3k-3 downto 3k-4 do

$$S(x) := x$$

b) for i:=2 to k-2; in stages (3k-3-i) through (2k-1) rename switches as follows

$$S(x) := \sigma^{-1}_{(3k-3-i)}(\sigma^{-1}_{(3k-4-i)}(\;\ldots\; \sigma^{-1}_{(2)}(x)))\,.$$

c) for stage 2k do

$$S(x) := P_4\ (S(x))$$

d) for stage 2k-1 do

$$S(x) := P_1\ (P_2\ (S(x)))$$

This completes the renumbering phases of the algorithm for group I and group III. Figure 2.6 and Figure 2.7 illustrate the renumbering phases for networks of size 16 and 32, respectively.

The second phase consists of setting the switches in the (k-2) stages in group II in such a way so as to obtain all the 2^{k-2} of independent 4x4 Benes blocks. The k^{th} stage, k-2 stages of group II and the $(2k-1)^{th}$ stage, together constitute one pass of Omega network [H6]. The problem of obtaining all 4x4 Benes blocks is in fact equivalent to the

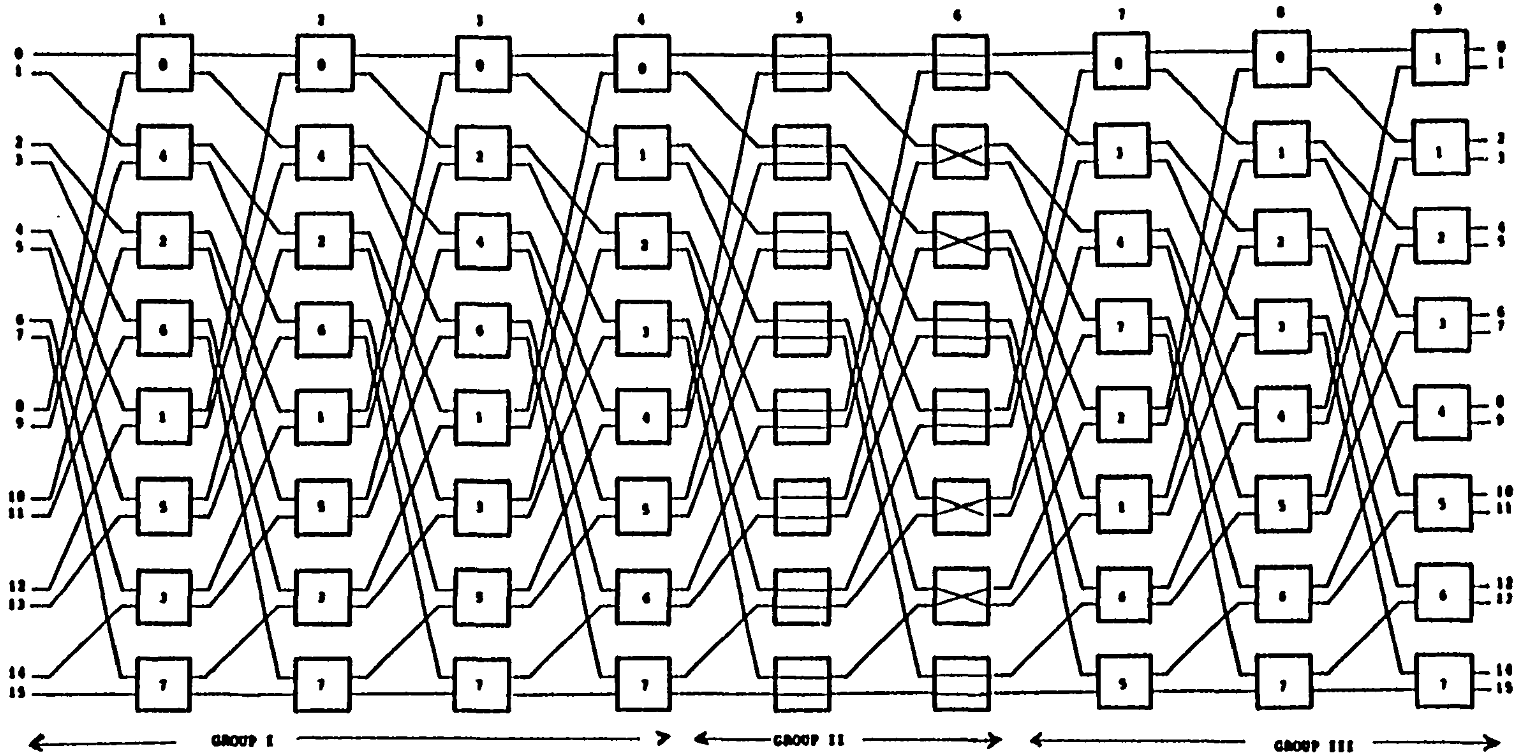


Figure 2.6: Renumbering scheme for N=16.

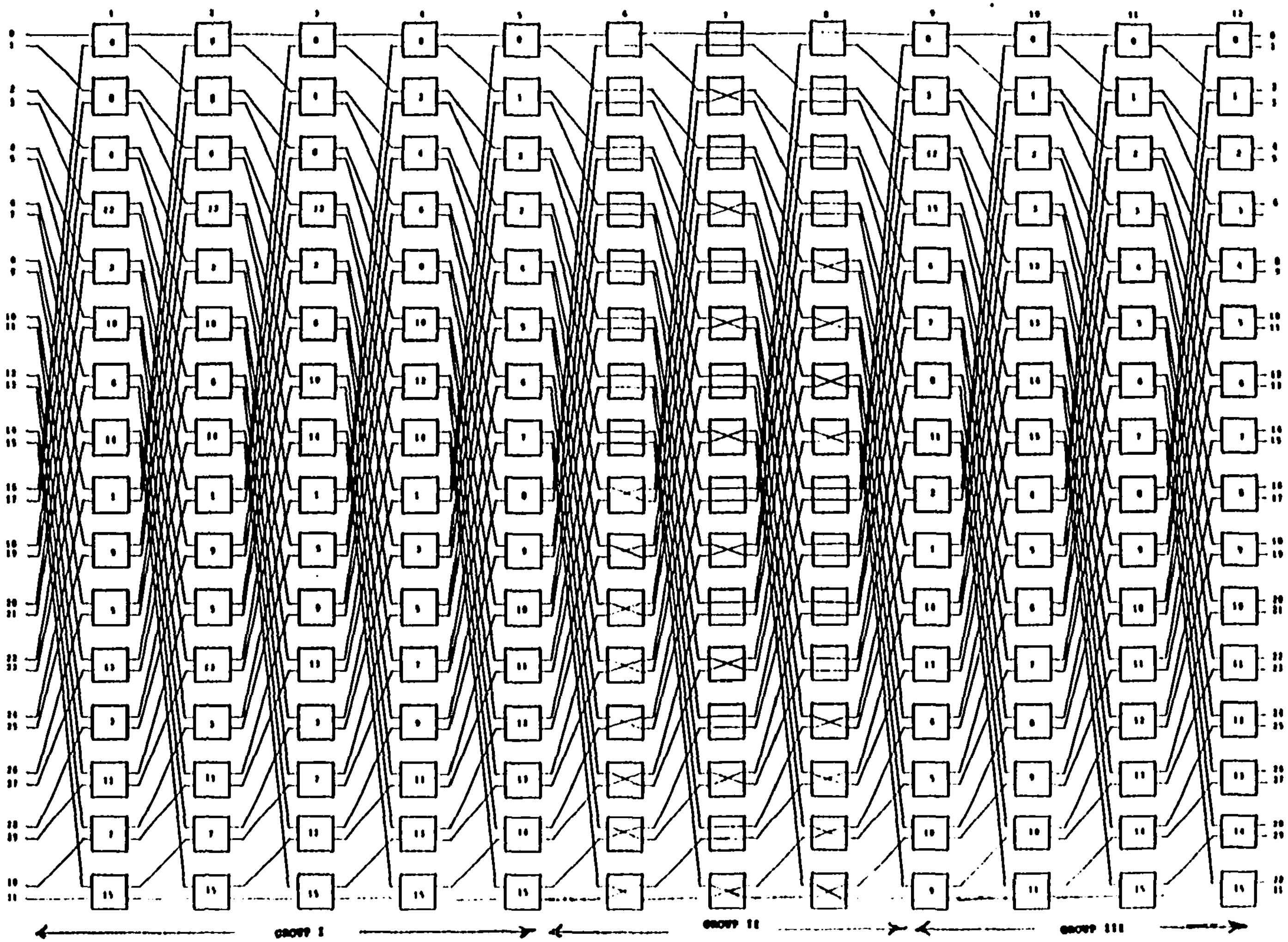


Figure 2.7: Renumbering scheme for N=32.

partitioning of an Omega network formed of stages k through (2k-1) [S4]. Combining theorems 1 and 6 in [S4], it follows that such a partitioning exists. In the following theorem we explicitly prove the existance of such a partitioning.

2.3.1.1 Partitioning of Omega Networks

Theorem 2.2:

An NxN Omega network with $N=2^k$ is partitionable into 2^{k-2} Benes blocks of size 4x4 blocks.

Proof:

Let $N=2^k$ and consider an Omega network of size N. Based on the algorithm 2.1, the rank of a switch x, in binary representation, in group III is as follows:

Stage	Renumbering Scheme
3k-3	$[x_{k-2}\ x_{k-3}\ \cdots\ x_2\ x_1\ x_0]$.
3k-4	$[x_{k-2}\ x_{k-3}\ \cdots\ x_2\ x_1\ x_0]$.
3k-5	$[x_{k-3}\ x_{k-2}\ \cdots\ x_2\ x_1\ x_0]$.
3k-6	$[x_{k-4}\ x_{k-3}\ \cdots\ x_2\ x_1\ x_0]$.
.	.
.	.
.	.
2k-2	$[\ x_2\ x_3\ \cdots\ x_{k-2}\ x_1\ x_0\ ]$.
2k-1	$[x_1\ x_2\ \cdots\ x_{k-1}\ x_{k-2}\ x_0]$.

The composition $P_2(x)$ can be written as follows:

If $x_{k-2}=0$ (first half) and $x_1=0$ ($\lfloor x/2 \rfloor$ is even), then

$$P_2([x_{k-2}\ x_{k-3}\ \cdots\ x_1\ x_0]) = [x_1\ x_2\ \cdots\ x_{k-2}\ x_0]$$

therefore,

$$P_2([x_{k-2}\ x_{k-3}\ \cdots\ x_1\ x_0]) = [x_1\ x_2\ \cdots\ x_{k-2}\ x_0].$$

If $x_{k-2}=0$ (first half) and $x_1=0$ ($\lfloor x/2 \rfloor$ is odd), then

$$P_2(([x_{k-2}\ x_{k-3}\ \cdots\ x_1\ x_0]+2^{k-3}) \text{ Mod } 2^{k-2})$$
$$=[x_1\ x_2\ \cdots\ x_{k-2}\ x_0]$$

therefore,

$$P_2([0\ (1 \oplus x_{k-3})\ \cdots\ x_1\ x_0]) = [x_1\ x_2\ \cdots\ x_{k-2}\ x_0].$$

If $x_{k-2}=1$ (second half) and $x_1=0$ ($\lfloor x/2 \rfloor$ is even), then

$$P_2([x_{k-2}\ x_{k-3}\ \cdots\ x_1\ x_0])$$
$$=[x_1\ x_2\ \cdots\ x_{k-2}\ x_0]$$

therefore,

$$P_2([x_{k-2}\ x_{k-3}\ \cdots\ x_1\ x_0]) = [x_1\ x_2\ \cdots\ x_{k-2}\ x_0].$$

If $x_{k-2}=1$ (second half) and $x_1=1$ ($\lfloor x/2 \rfloor$ is odd), then

$$P_2(2^{k-2}+(2^{k-3}+[x_{k-2}\ x_{k-3}\ \cdots\ x_1\ x_0])\text{Mod } 2^{k-2})=$$
$$[x_1\ x_2\ \cdots\ x_{k-2}\ x_0]$$

therefore,

$$P_2(2^{k-2}+([01000\ \ldots\ 00]+[\ x_1\ x_2\ \cdots\ x_{k-2}\ x_0])\text{ Mod } 2^{k-2})=$$
$$[x_1\ x_2\ \cdots\ x_{k-2}\ x_0] \text{ and}$$

$$P_2([0\ (1\oplus x_{k-3})\ \ldots\ x_1\ x_0]) = [x_1\ x_2\ \ldots\ x_{k-2}\ x_0].$$

After some simplification, function P_2 becomes:

$$P_2([0\ x_{k-3}\ \ldots\ 0\ x_0]) \longrightarrow [0\ x_2\ \ldots\ 0\ x_0].$$

$$P_2([0\ (1\oplus x_{k-3})\ \ldots\ 1\ x_0]) \longrightarrow [1\ x_2\ \ldots\ 0\ x_0]$$

$$P_2([1\ x_{k-3}\ \ldots\ 0\ x_0]) \longrightarrow [0\ x_2\ \ldots\ 1\ x_0].$$

$$P_2([1\ (1\oplus x_{k-3})\ \ldots\ 1\ x_0]) \longrightarrow [1\ x_2\ \ldots\ 1\ x_0]$$

An Example for N=16 is shown in Table 2.1.

$\underline{x}$	$\underline{P_2}(\underline{x})$
0000	0000
0001	0001
0010	1100
0011	1101
0100	0100
0101	0101
0110	1000
0111	1001
1000	0010
1001	0011
1010	1110
1011	1111
1100	0110
1101	0111
1110	1010
1111	1011

Table 2.1

Binary Representation for a permutation $P_2(x)$.

Define P_1 over P_2 as follows,

1. If $x_{k-2}=0$, $x_1=0$, and $x_0=0$ then

$$P_1(P_2([x_{k-2}\ x_{k-3}\ \cdots\ x_1\ x_0]))=$$
$$P_1([x_1\ x_2\ \cdots\ x_{k-2}\ x_0])=$$
$$[x_1\ x_2\ \cdots\ x_{k-2}\ x_0].$$

2. If $x_{k-2}=0$, $x_1=0$, and $x_0=1$ then

$$P_1(P_2([x_{k-2}\ x_{k-3}\ \cdots\ x_1\ x_0]))=$$
$$P_1([x_1\ x_2\ \cdots\ x_{k-2}\ x_0])$$
$$P_1(P_2([x_{k-2}\ x_{k-3}\ \cdots\ x_1\ x_0]\ +2^{k-2})\mathrm{Mod}\ 2^{k-1})=$$
$$[x_1\ x_2\ \cdots\ x_{k-2}\ x_0].$$
$$P_1(P_2([x_{k-2}\ x_{k-3}\ \cdots\ x_1\ x_0]\ +[1000\ \ldots\ 00])\mathrm{Mod}\ 2^{k-1})=$$
$$[x_1\ x_2\ \cdots\ x_{k-2}\ x_0]$$
$$P_1(P_2([1\ x_{k-3}\ \cdots\ x_1\ x_0]\mathrm{Mod}\ 2^{k-1}\ ))=[x_1\ x_2\ \cdots\ x_{k-2}\ x_0])$$
$$P_1(P_2([1\ x_{k-3}\ \cdots\ x_1\ x_0]))\ =[x_1\ x_2\ \cdots\ x_{k-2}\ x_0])$$

3. If $x_{k-2}=0$, $x_1=1$, and $x_0=0$ then

$$P_1(P_2([0\ (1\oplus x_{k-3})\ \cdots\ x_1\ x_0]))=$$
$$P_1([x_1\ x_2\ \cdots\ x_{k-1}\ x_0])\ =[x_1\ x_2\ \cdots\ x_{k-2}\ x_0].$$

4. If $x_{k-2}=0$, $x_1=1$, and $x_0=1$ then

$$P_1(P_2([0\ (1\oplus x_{k-3})\ \cdots\ x_1\ x_0]))$$
$$P_1(P_2([0\ (1\oplus x_{k-3})\ \cdots\ x_1\ x_0]\ +2^{k-2})\mathrm{Mod}\ 2^{k-1})$$
$$P_1(P_2([0\ (1\oplus x_{k-3})\ \cdots\ x_1\ x_0]\ +[1000\ \ldots\ 00])\mathrm{Mod}\ 2^{k-1}$$
$$P_1(P_2([0\ (1\oplus x_{k-3})\ \cdots\ x_1\ x_0])\ \mathrm{Mod}\ 2^{k-1})=$$
$$[x_1\ x_2\ \cdots\ x_{k-1}\ x_0].$$
$$P_1(P_2([0\ (1\oplus x_{k-3})\ \cdots\ x_1\ x_0]))\ =[x_1\ x_2\ \cdots\ x_{k-2}\ x_0].$$

5. If $x_{k-2}=1$, $x_1=0$, and $x_0=0$ then

$P_1(P_2([x_{k-2}\ x_{k-3}\ \cdots\ x_1\ x_0]))=$

$P_1([x_1\ x_2\ \cdots\ x_{k-2}\ x_0]) = [x_1\ x_2\ \cdots\ x_{k-2}\ x_0]$.

6. If $x_{k-2}=1$, $x_1=0$, and $x_0=0$ then

$P_1(P_2([x_{k-2}\ x_{k-3}\ \cdots\ x_1\ x_0]))= P_1([x_1\ x_2\ \cdots\ x_{k-2}\ x_0])$

$P_1(P_2([x_{k-2}\ x_{k-3}\ \cdots\ x_1\ x_0] +2^{k-2})\text{Mod } 2^{k-1})=$

$[x_1\ x_2\ \cdots\ x_{k-2}\ x_0])$

$P_1(P_2([x_{k-2}\ x_{k-3}\ \cdots\ x_1\ x_0] +[1000\ \ldots\ 00])\text{Mod } 2^{k-1})=$

$[x_1\ x_2\ \cdots\ x_{k-2}\ x_0]$

$P_1(P_2([(1\oplus x_{k-2})\ x_{k-3}\ \cdots\ x_1\ x_0])) = [x_1\ x_2\ \cdots\ x_{k-2}\ x_0]$.

7. If $x_{k-2}=1$, $x_1=0$, and $x_0=1$ then

$P_1(P_2([0\ (1\oplus x_{k-3})\ \cdots\ x_1\ x_0]))=$

$P_1([x_1\ x_2\ \cdots\ x_{k-1}\ x_0]) = [x_1\ x_2\ \cdots\ x_{k-2}\ x_0]$.

8. If $x_{k-2}=1$, $x_1=1$, and $x_0=1$ then

$P_1(P_2([1\ (1\oplus x_{k-3})\ \cdots\ x_1\ x_0]))=$

$P_1(P_2([1\ (1\oplus x_{k-3})\ \cdots\ x_1\ x_0] +2^{k-2})\text{Mod } 2^{k-1})$

$P_1(P_2([0\ (1\oplus x_{k-3})\ \cdots\ x_1\ x_0] +[1000\ \ldots\ 00])\text{Mod } 2^{k-1}$

$P_1(P_2([1\ (1\oplus x_{k-3})\ \cdots\ x_1\ x_0])\ \text{Mod } 2^{k-1})=$

$[x_1\ x_2\ \cdots\ x_{k-1}\ x_0]$.

$P_1(P_2([0\ (1\oplus x_{k-3})\ \cdots\ x_1\ x_0])) = [x_1\ x_2\ \cdots\ x_{k-2}\ x_0]$.

Therefore the composite function $P_1(P_2(x))$ is:

$$[0\ x_{k-3}\ \dots\ x_2\ 0\ 0] \text{ -----> } [0\ x_2\ \dots\ x_{k-2}\ 0]$$
$$[1\ x_{k-3}\ \dots\ x_2\ 0\ 1] \text{ -----> } [0\ x_2\ \dots\ x_{k-2}\ 1]$$
$$[0\ (1\oplus x_{k-3})\ \dots\ x_2\ 1\ 0] \text{ -----> } [1\ x_2\ \dots\ x_{k-2}\ 0]$$
$$[1\ (1\oplus x_{k-3})\ \dots\ x_2\ 1\ 1] \text{ -----> } [1\ x_2\ \dots\ x_{k-2}\ 1]$$
$$[x_{k-2}\ x_{k-3}\ \dots\ x_2\ 0\ 0] \text{ -----> } [0\ x_2\ \dots\ x_{k-2}\ 0]$$
$$[(1\oplus x_{k-2})\ x_{k-3}\ \dots\ x_2\ 0\ 1] \text{ -----> } [0\ x_2\ \dots\ x_{k-2}\ 1]$$
$$[1\ (1\oplus x_{k-3})\ \dots\ x_2\ 1\ 0] \text{ -----> } [1\ x_2\ \dots\ x_{k-2}\ 0]$$
$$[0\ (1\oplus x_{k-3})\ \dots\ x_2\ 1\ 1] \text{ -----> } [1\ x_2\ \dots\ x_{k-2}\ 1]$$

and in general,

$$[x_{k-2}\ x_{k-3}\ \dots\ x_1\ x_0] \text{ -----> } [x_1\ x_2\ \dots\ x_{k-3}\ x_{k-2}\ x_0], \quad \text{if } x_1=0 \text{ and } x_0=0.$$

$$[x_{k-2}\ x_{k-3}\ \dots\ x_1\ x_0] \text{ -----> } [x_1\ x_2\ \dots\ x_{k-3}\ x_{k-2}\ x_0], \quad \text{if } x_1=0 \text{ and } x_0=1.$$

$$[x_{k-2}\ x_{k-3}\ \dots\ x_1\ x_0] \text{ -----> } [x_1\ x_2\ \dots\ x_{k-3}\ x_{k-2}\ x_0], \quad \text{if } x_1=1 \text{ and } x_0=0.$$

$$[x_{k-2}\ x_{k-3}\ \dots\ x_1\ x_0] \text{ -----> } [x_1\ x_2\ \dots\ x_{k-3}\ x_{k-2}\ x_0], \quad \text{if } x_1=1 \text{ and } x_0=1.$$

An example for N=16 is shown in Table 2.2.

$\underline{x}$	$\underline{P}_1(\underline{P}_2(\underline{x}))$
0000	0000
0001	0011
0010	1100
0011	0101
0100	0100
0101	0111
0110	1000
0111	1011
1000	0010
1001	0001
1010	1110
1011	1101
1100	0110
1101	0101
1110	1010
1111	1001

Table 2.2

Binary Representation for a Composed permutation $P_1(P_2(x))$.

Construction of Benes blocks of size 4x4 requires the partitioning scheme of Siegel[S4]. To define such a partitioning, the following definitions are introduced.

Definitions 2.5: Let:

1. $P=\{0,1, \ldots ,N-1\}$, the set of output terminals at stage 2k-1.
2. $l_i=\{l_{i0}, l_{i1}, \ldots ,l_{i(w_i-1)}\}$, the set of input terminals in the i^{th} partition.
3. w_i is the size of l_i (that is $|l_i|=w_i$), where $0 < w_i \leq N$ and w_i is a power of two. In this case $w_i=4$.
4. v is the number of partitions which is 2^{k-2} in this case.

5. $L=l_0 \cup l_1 \cup \ldots \cup l_{(v-1)}$, the set of input terminals at stage k.

6. m is a bijection from P to L such that if $m(p_k)=l_{ij}$ then $m^{-1}(l_{ij}) = p_k$, where $p_k \in P$ and $l_{ij} \in L$.

Siegel[S4] stated the following theorem.

Theorem:

In terms of the cycle structure of the cube interconnection function, the network will be partitioned into independent sub-networks if and only if m is such that for all i, $0 \leq i < v$, for each of $\log_2 w_i$ distinct cube functions exactly $w_i/2$ of the cycles contain only elements of P which are mapped to elements of l_i by m. In addition, for $0 \leq r < \log_2 w_i$, if

$$\mathrm{Cube}_i(m^{-1}(l_{ij}))=m^{-1}(l_{ik}) \qquad (4.3)$$

then j and k can differ in only the r^{th} bit position for j,k, $0 \leq j,k < w_i$, where

$$\mathrm{Cube}_i([s_{n-1} \ldots s_1\ s_0])= [s_{n-1} \ldots s_{i+1}\ \bar{s}_i\ s_{i-1} \ldots s_0].$$

In order to verify the criteria of the above theorem, consider two consecutive switches at stage k, namely $[x_1\ x_2 \ldots x_{k-2}\ 0]$ and $[x_1\ x_2 \ldots x_{k-2}\ 1]$ which are preceded by shuffle permutations. The input terminals to switch $[x_1\ x_2 \ldots x_{k-2}\ 0]$ are $[0\ x_1\ x_2 \ldots x_{k-2}\ 0]$ and $[1\ x_1\ x_2 \ldots x_{k-2}\ 0]$. The input terminals to switch $[x_1\ x_2 \ldots x_{k-2}\ 1]$ are [0

$x_1\ x_2 \ldots x_{k-2}\ 1]$ and $[1\ x_1\ x_2 \ldots x_{k-2}\ 1]$. After renumbering the switches at stage 2k-1 the physical address of the switch $[x_1\ x_2 \ldots x_{k-2}\ 0]$ will be at $[x_{k-2}\ x_{k-3} \ldots x_1\ 0]$ if $x_1=0$ and $[x_{k-2}\ \bar{x}_{k-3} \ldots x_1\ 0]$ if $x_1=1$. Likewise, the physical address of switch $[x_1\ x_2 \ldots x_{k-2}\ 1]$ will be at $[x_{k-2}\ x_{k-3} \ldots x_1\ 1]$ if $x_1=0$ and $[\bar{x}_{k-2}\ \bar{x}_{k-3} \ldots x_1\ 1]$ if $x_1=1$, as shown in Figure 2.8. Consequently, the output terminals of switch $[x_1\ x_2 \ldots x_{k-2}\ 0]$ at stage 2k-1 is $[x_{k-2}\ x_{k-3} \ldots x_1\ 0\ 0]$ and $[x_{k-2}\ x_{k-3} \ldots x_1\ 0\ 1]$ if $x_1=0$ and it is $[x_{k-2}\ x_{k-3} \ldots x_1\ 0\ 1]$ and $[x_{k-2}\ \bar{x}_{k-3} \ldots x_1\ 0\ 1]$ if $x_1=1$. The output terminals of the switch $[x_1\ x_2 \ldots x_{k-2}\ 1]$ at stage 2k-1 is $[\bar{x}_{k-2}\ x_{k-3} \ldots x_1\ 1\ 0]$ and $[\bar{x}_{k-2}\ x_{k-3} \ldots x_1\ 1\ 1]$ if $x_1=0$ and it is $[\bar{x}_{k-2}\ \bar{x}_{k-3} \ldots x_1\ 1\ 0]$ and $[\bar{x}_{k-2}\ \bar{x}_{k-3} \ldots x_1\ 1\ 1]$ if $x_1=1$, as shown in Figure 2.8.

One of the possible correct choice of m is

$$m([x_{k-2}\ x_{k-3} \ldots x_1\ 0\ 0]= [0\ x_1\ x_2 \ldots x_{k-2}\ 0] = l_{i0}$$

$$m([x_{k-2}\ x_{k-3} \ldots x_1\ 0\ 1]= [0\ x_1\ x_2 \ldots x_{k-2}\ 1] = l_{i1}$$

$$m([x_{k-2}\ x_{k-3} \ldots x_1\ 1\ 0]= [1\ x_1\ x_2 \ldots x_{k-2}\ 0] = l_{i2}$$

$$m([x_{k-2}\ x_{k-3} \ldots x_1\ 1\ 1]= [1\ x_1\ x_2 \ldots x_{k-2}\ 1] = l_{i3}$$

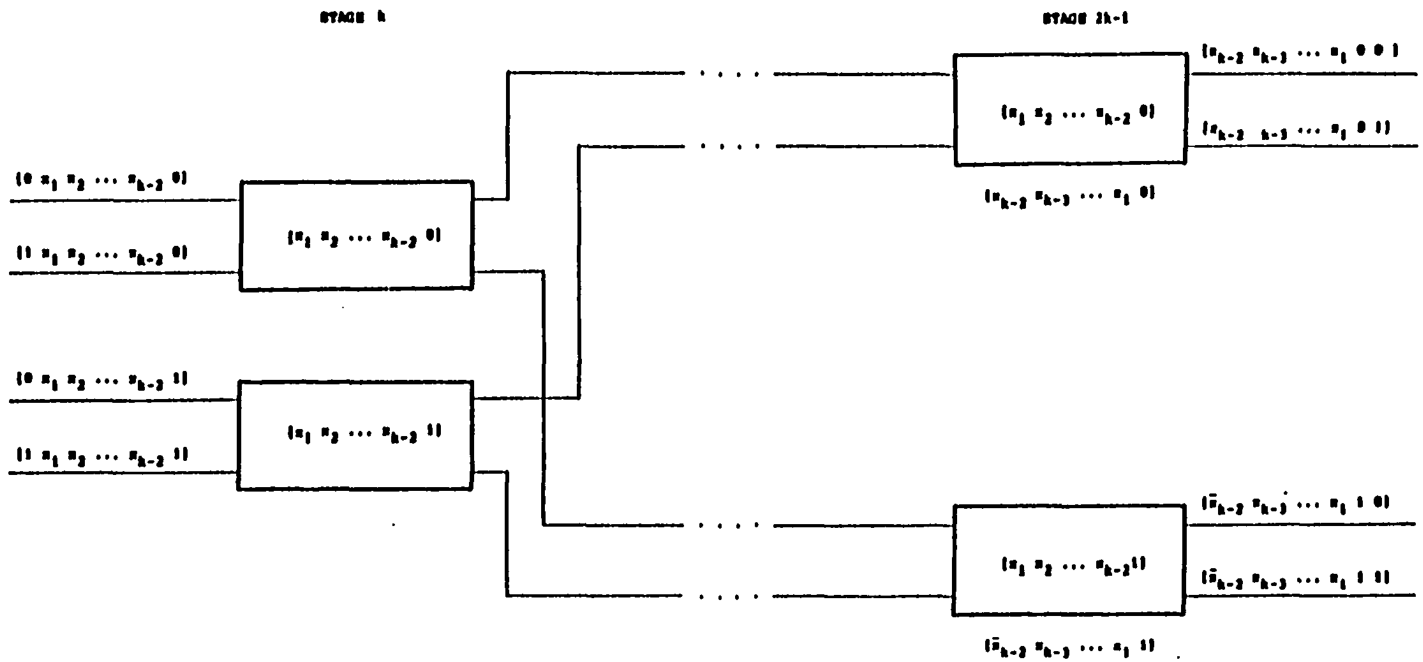


Figure 2.8: A configuration of two consecutive switches at stage k and the corresponding switches at stage 2k-1.

As an example, one possible choices of m for N=16 is shown in Table 2.3.

```
m([0000])=[0000]= 0
m([0001])=[0001]= 1
m([0010])=[1010]=10
m([0011])=[1011]=11
m([0100])=[0100]= 4
m([0101])=[0101]= 5
m([0110])=[1110]=14
m([0111])=[1111]=15
m([1000])=[0010]= 2
m([1001])=[0011]= 3
m([1010])=[1000]= 8
m([1011])=[1001]= 9
m([1100])=[0110]= 6
m([1101])=[0111]= 7
m([1110])=[1100]=12
m([1111])=[1101]=13
```

Table 3.2: One possible choice of partitioning for N=16.

Clearly, the criterion of theorem 2.5 holds good for partitioning one pass of an Omega network for the special case where $v=2^{k-2}$, $w_i=4$, and m is a bijection from P to L as above.

2.4 CONCLUSION

While the open problem whether the cascade of 2k shuffle-exchange stages is universal is not yet settled, theorem 2.1 implies that the universality of this cascade must be settled outside of the framework of the Benes network. Curiously enough it has been verified by exhaustive enumeration that for k=3, a cascade of five shuffle-exchange stages indeed realizes the set of all permutations over eight objects. Benes[B5] recently gave a set of sufficient conditions in the form of a new factorization of the symmetric groups of composite degree. Even this new criterion for rearrangeability is too sufficient a condition that is not often satisfied by many rearrangeable networks. For example, it can be easily verified that a cascade of five shuffle-exchange stages is universal, yet it does not satisfy the new set of Benes conditions [B5]. Another example of a rearrangeable network that does not satisfy the condition in [B5] is the 4x4 Benes network corresponding to the graph in Figure 2.3(a). All these results indicate that the resolution of the above open problem must await the development of newer techniques for proving universality of interconnection networks.

Chapter III

COMPUTING THE NUMBER OF PERMUTATIONS REALIZED BY SW-BANYAN NETWORKS

3.1 INTRODUCTION:

Goke and Lipovski[Gl] introduced a fundamental class of networks, in the context of multi-processor systems, called the Banyan network. A Banyan network is defined a by certain kind of directed graph G. A vertex of G is called a base of G if and only if there are no arcs incident into it in G and is called an apex of G if and only if there are no arcs incident out of it. The useful property of a Banyan network is that there is one and only one path from any base to any apex (unique path network). The Banyan class contains a very rich and useful subclass of networks called L-stage Banyan networks[Gl,Dl]. A variety of special cases of L-stage Banyan networks have received considerable attentions in the design and in the literature. Large Banyan networks can be synthesized from smaller ones. This is illustrated in Figure 3.1(a). The interconnections of these Banyans can be represented by a graph such as in Figure 3.1(b). A cross-bar is a trivial Banyan network. L-stage Banyans are synthesized recursively from cross-bar switches as in Figure 3.1(b). An example of a non-L-level Banyan is

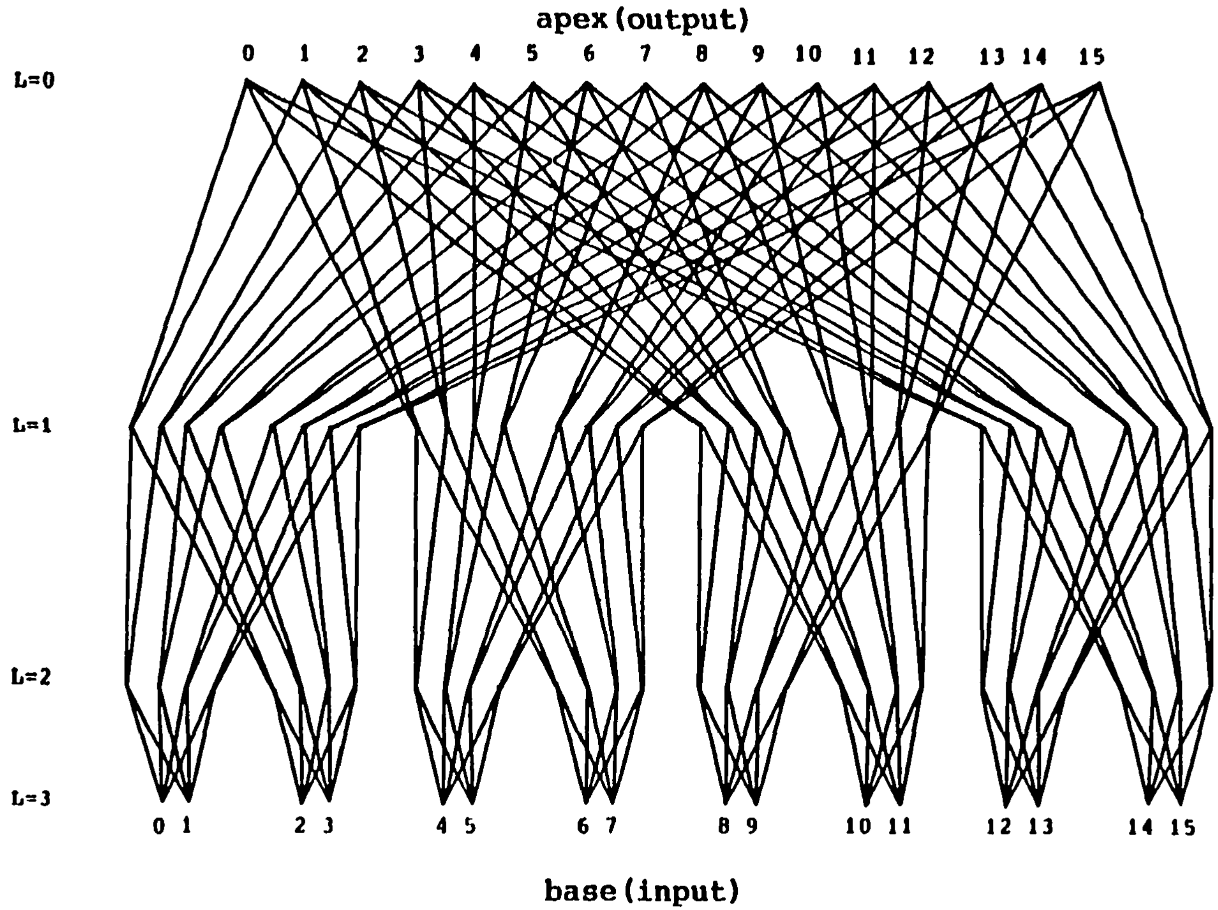


Figure 3.1(a): An example of L-level banyan, where F=(2,2,4) and S=(4,2,2).

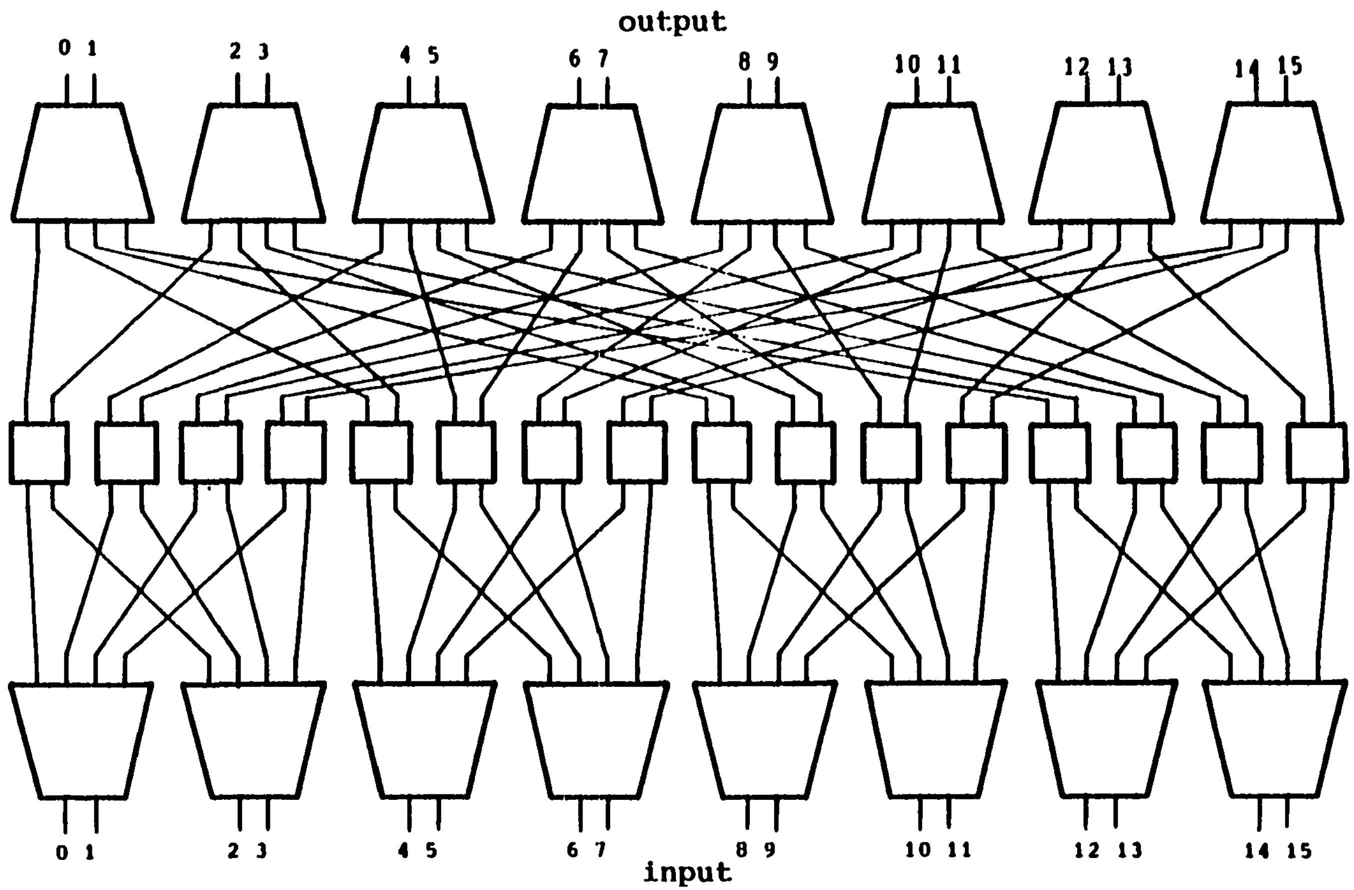


Figure 3.1(b): The corresponding network of Figure 3.1(a).

given in Figure 3.2. One of the most notable subclasses of an L-stage Banyan network is an SW-Banyan[G2] and Delta Network[D2,K4,P2].

The well known networks such as Omega networks[L3], Indirect Binary n-cube networks[P4] and Base-line networks[W2] are all special cases of Delta networks. The characteristic property of a Delta network is that it permits simple decentralized routing based on "destination" tags.

The primary emphasis of this chapter is to compute the number of permutations realizeable by a class of SW-Banyan networks. It is shown that the method due to Bhuyan and Agrawal [B7] is incorrect and a new method is suggested. This method gives rise to an interesting class of enumeration problems. Our notations follow those given in DeGroot[D1,K2].

3.2 PROPERTIES OF BANYAN NETWORKS:

An L-level Banyan is a Banyan graph in which the path between each base to apex(or apex to base) has length L. Therefore, in an L-level Banyan, there are L+1 level of nodes and L-level of edges. The apex is considered to be at level 0 and the base is at level L.

SW-Banyan is the proper subset of L-level Banyan networks. A Banyan is an SW-Banyan if and only if for any two bases b_1 and b_2 (or any apexes a_1 and a_2), their level-x reachability sets are disjoint or identical, where the lev-

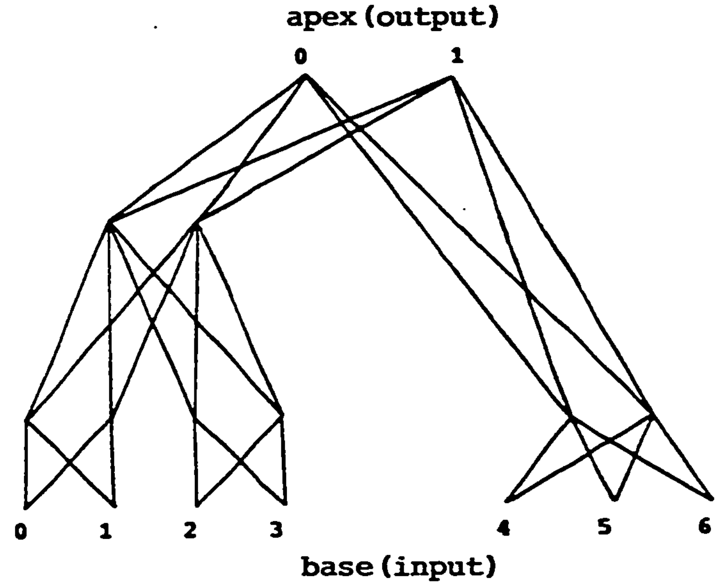


Figure 3.2(a): An example of non-L-level banyan.

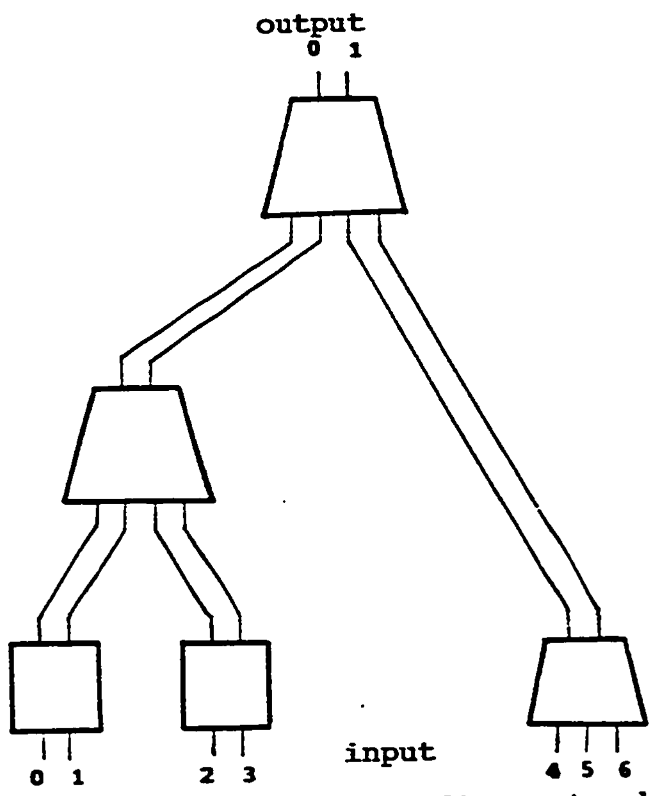


Figure 3.2(b): The corresponding network of Figure 3.2(a).

el-x reachability set for any base b is defined as the set of all nodes at level x ($0 \leq x \leq L$) that can be reached by directed paths from b.

Graph Representation of Banyan Networks:

Definition 3.1:

A base of a Banyan graph is any vertex with in-degree zero and an apex is any vertex with out-degree zero. All other vertices are called intermediate vertices, where the direction is taken from base to apex. In our treatment, base is taken to be the input and apex is taken to be the output.

Definition 3.2:

In a Banyan network, the spread of a vertex is the out-degree of a node and the fanout of a vertex is the in-degree of a node (the direction is from base to apex).

Definition 3.3:

If all vertices within the same level of a Banyan network have identical spread and fanout values then the Banyan is called uniform Figure 3.3, otherwise it is called non-uniform, Figure 3.4.

In a uniform Banyan network, the fanout values and the spread values may be characterized by L component vectors $F=(f_0,f_1, \ldots ,f_{L-1})$ and $S=(s_1,s_2, \ldots ,s_L)$ as fanout vector

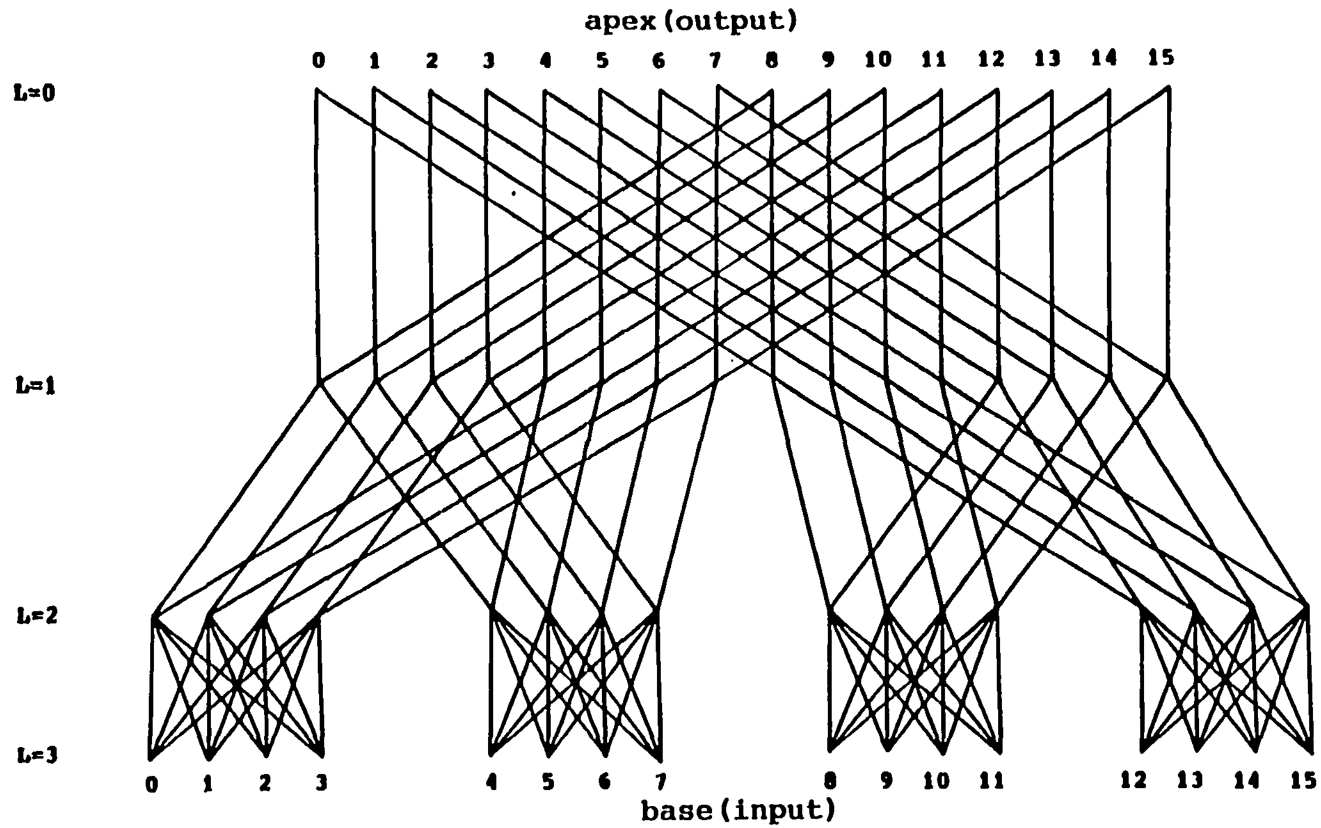


Figure 3.3(a): An example of Uniform banyan, where F=(2,2,4) and S=(2,2,4).

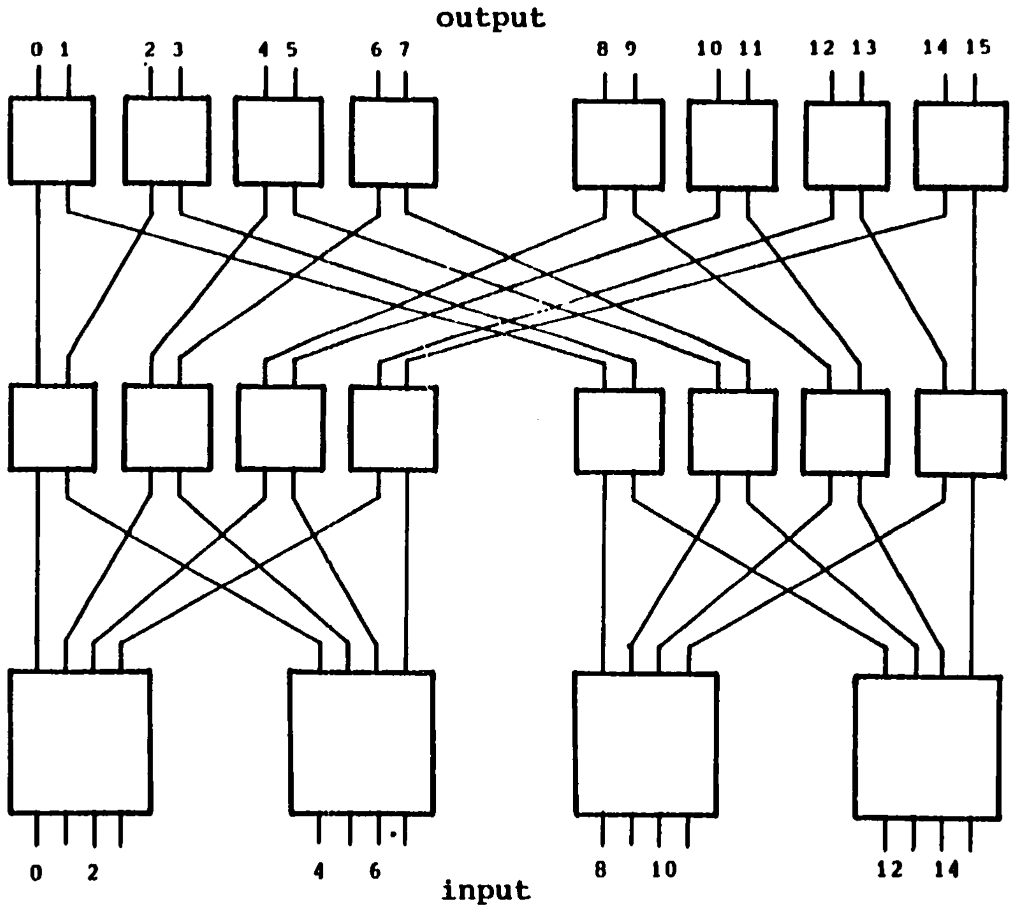


Figure 3.3(b): The corresponding network of Figure 3.3(a).

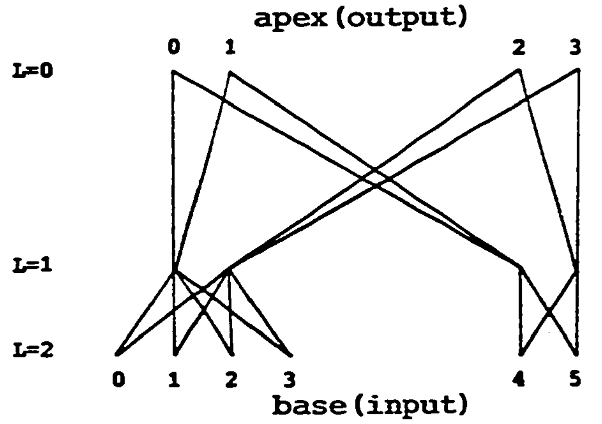


Figure 3.4(a): An example of non-Uniform banyan.

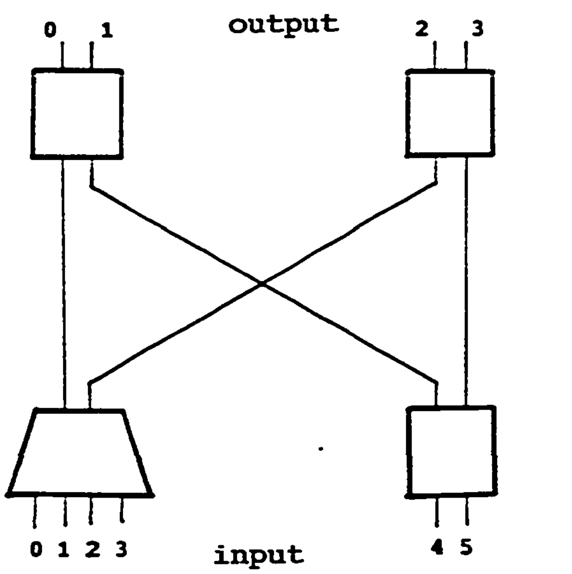


Figure 3.4(b): The corresponding network of Figure 3.4(a).

and spread vector, respectively, where s_i and f_i denote the spread and the fanout of a node at level i. Note that s_ix f_{i-1} is the size of the switches at level i, for $1 \leq i \leq L$. Clearly, $f_L=s_0=0$.

Definition 3.4:

If $s_{i+1}=f_i$ for $1 \leq i \leq L-1$, that is S=F then the Banyan network is called <u>rectangular</u>, Figure 3.5. If $s_{i+1} \neq f_i$ for some $0 \leq i \leq L-1$, then it is called <u>non-rectangular</u>, Figure 3.6.

Definition 3.5:

If every component of S is equal to some constant s and every component of F is equal to some constant f then the Banyan is called <u>regular</u>, Figure 3.7, otherwise it is <u>irregular</u>, Figure 3.8.

If s=f this implies that F=S and the Banyan network is both regular and rectangular which is called <u>strongly rectangular</u>. Figure 3.9 and Figure 3.10 show strongly and weakly rectangular Banyan networks, respectively.

Recently DeGroot[D1] introduced an interesting class of expanding and contracting SW-Banyan networks. Given a non-prime integer N, a variety of SW-Banyan networks can be built by considering different sets of factors of N. It is shown through examples[D1] that an expanding and contracting SW-Banyan has less path blockage, i.e. when one base is connected to one apex by a directed communication path, no oth-

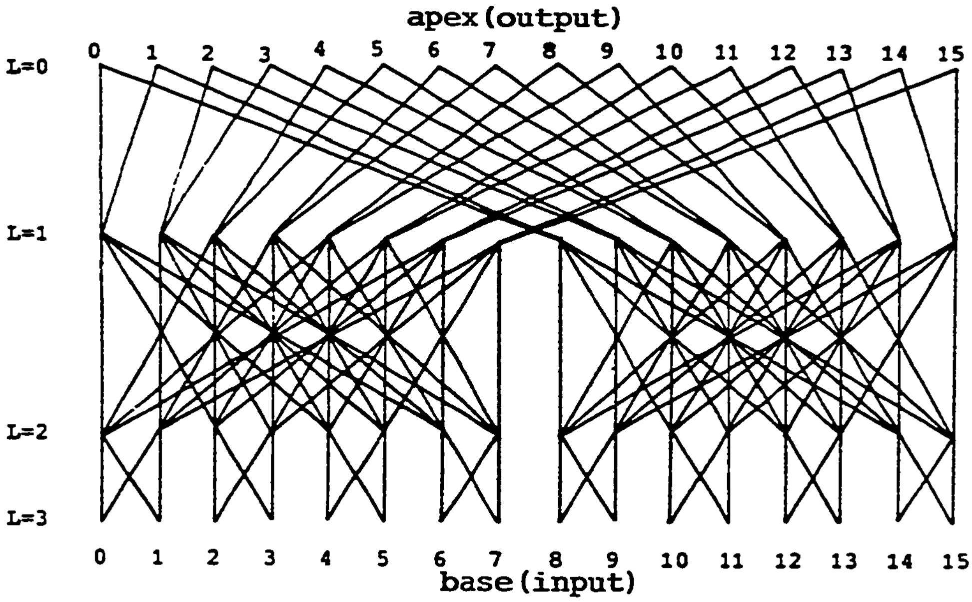


Figure 3.5(a): An example of Rectangular banyan where F=(2,4,2) and S=(2,4,2).

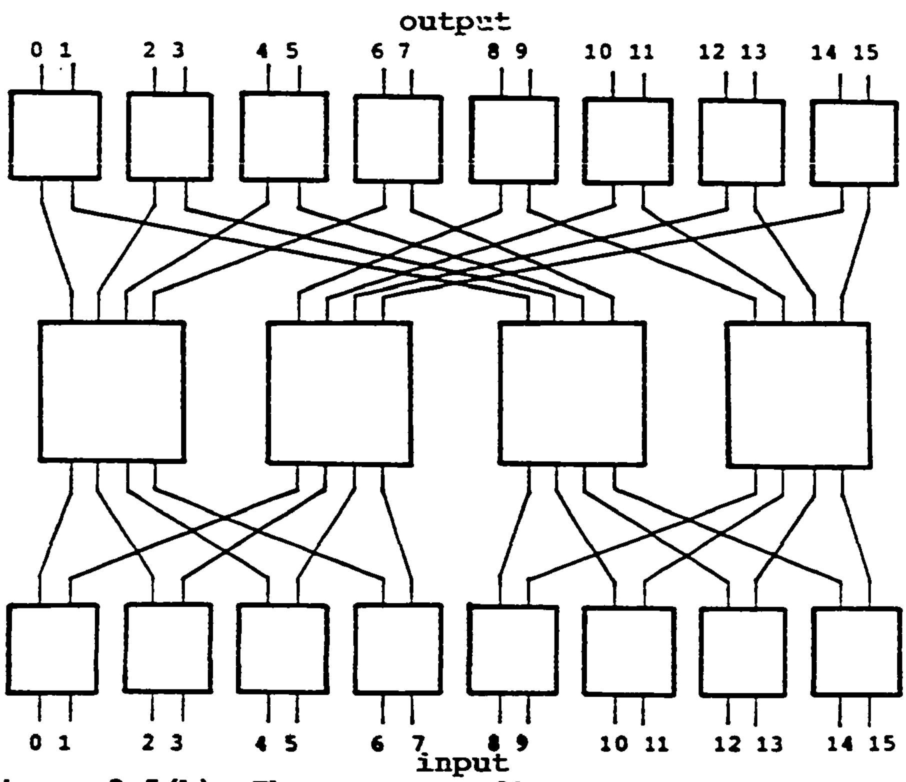


Figure 3.5(b): The corresponding network of Figure 3.5(a).

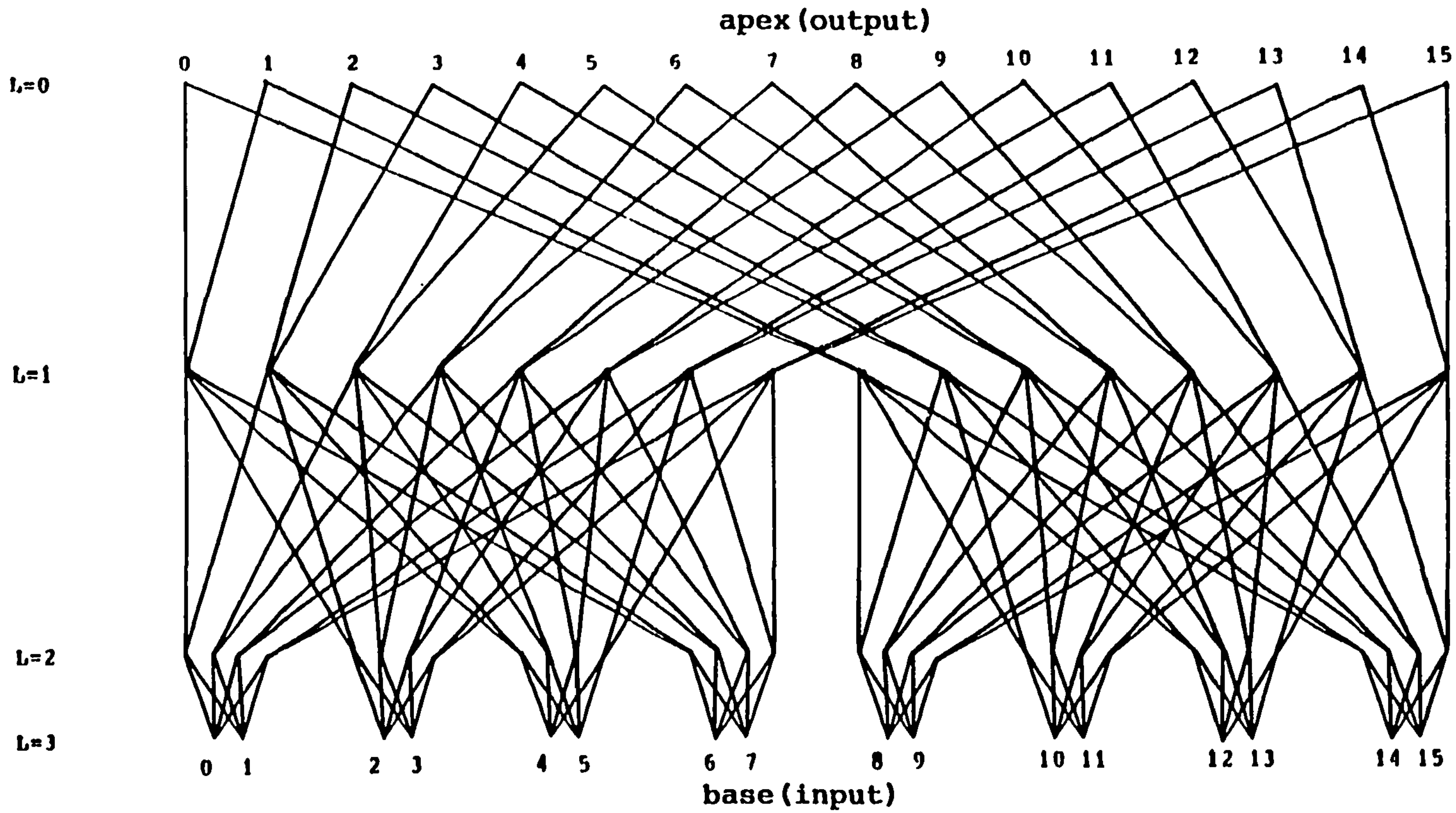


Figure 3.6(a): An example of non-Rectangular banyan,
where F=(2,2,4) and S=(2,4,2).

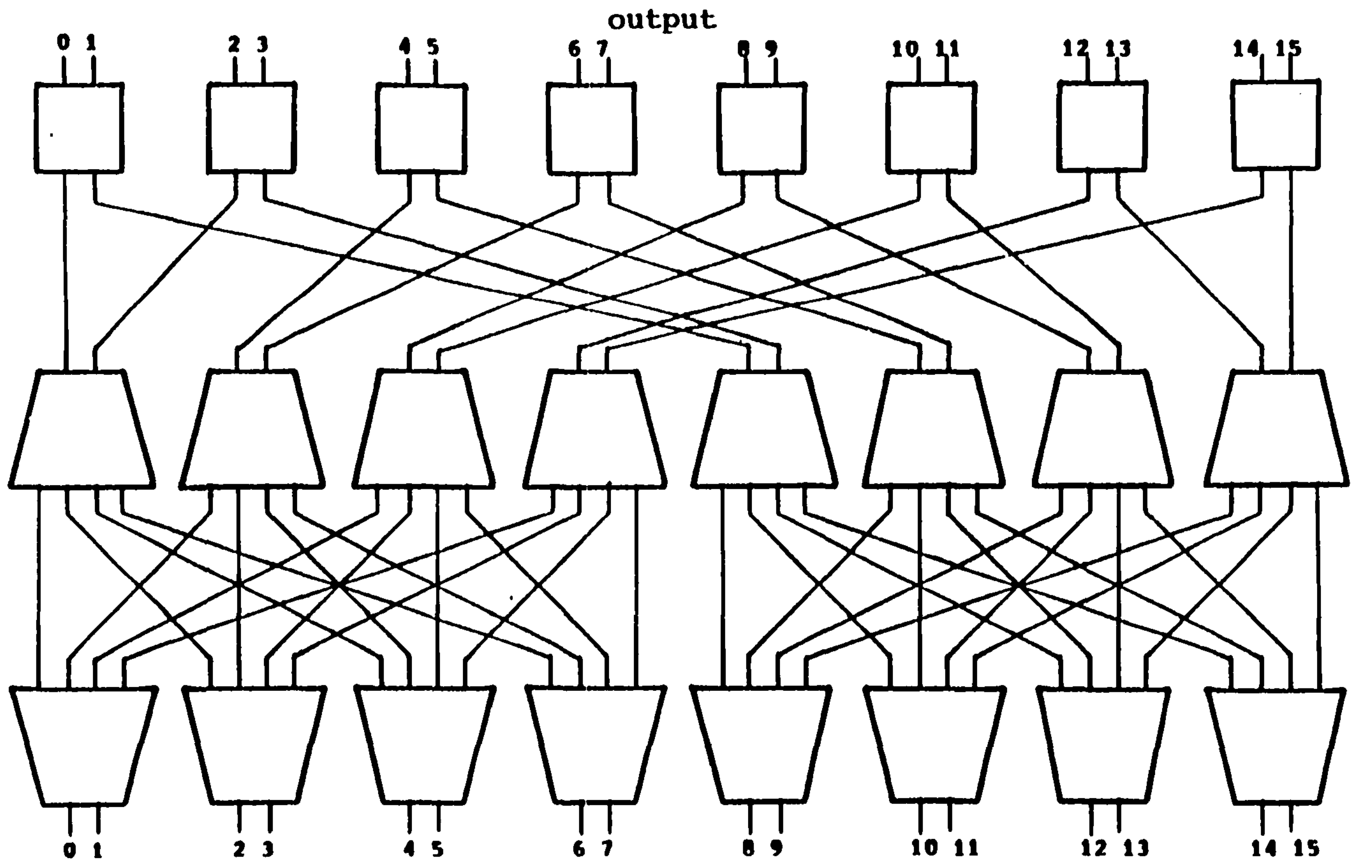


Figure 3.6(b): The corresponding network of Figure 3.6(a).

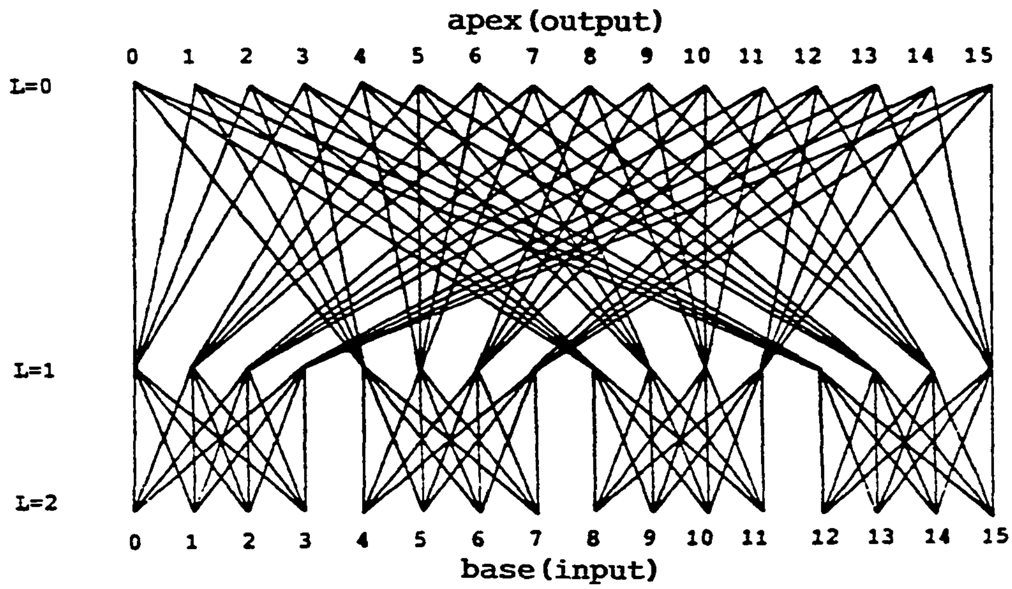


Figure 3.7(a): An example of Regular banyan, where F=(4,4) and S=(4,4).

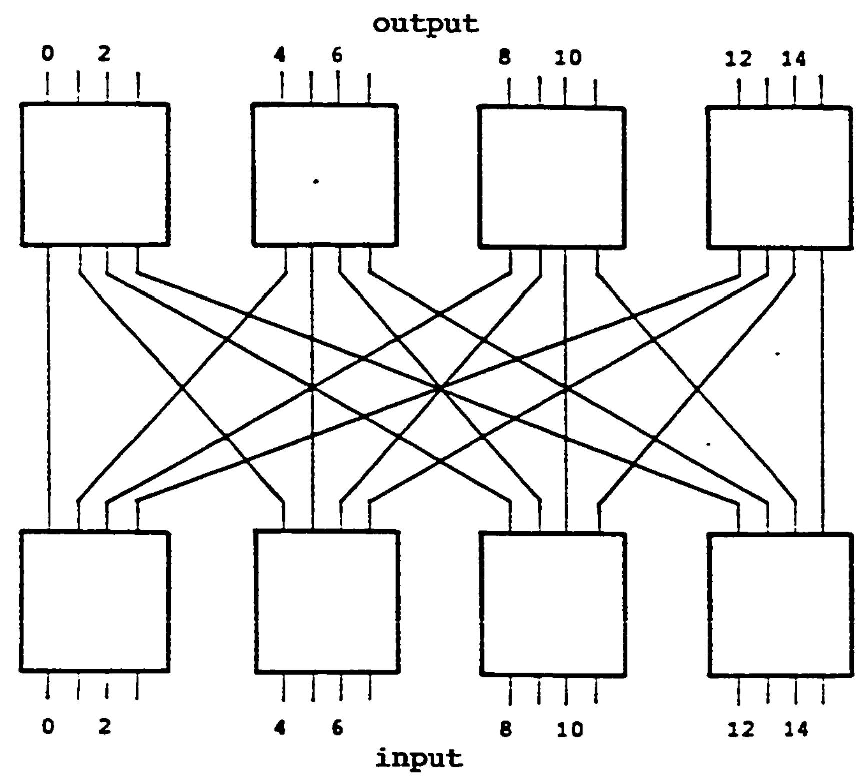


Figure 3.7(b): The corresponding network of Figure 3.7(a).

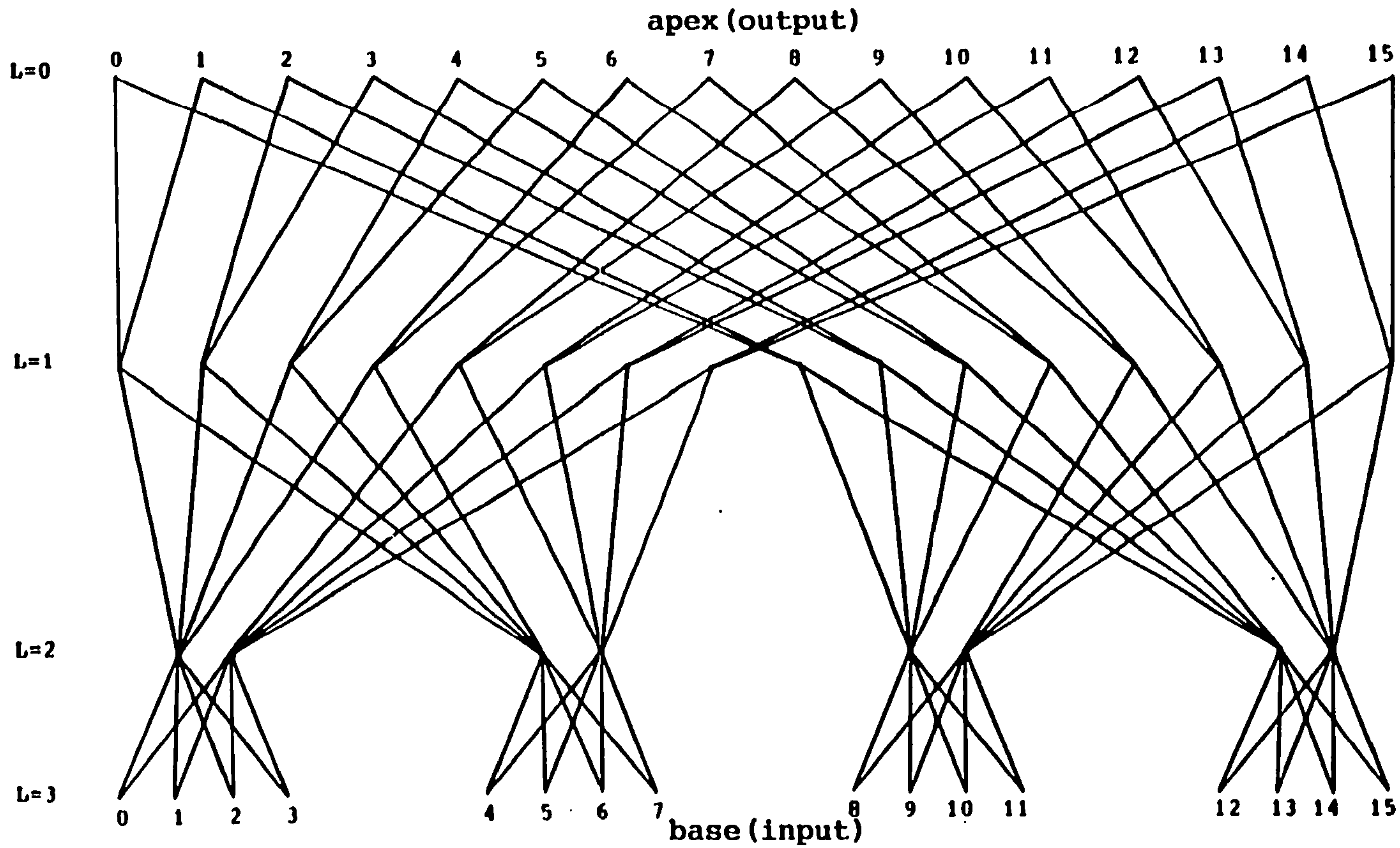


Figure 3.8(a): An example of Irregular banyan where F=(2,4,2) and S=(2,2,4).

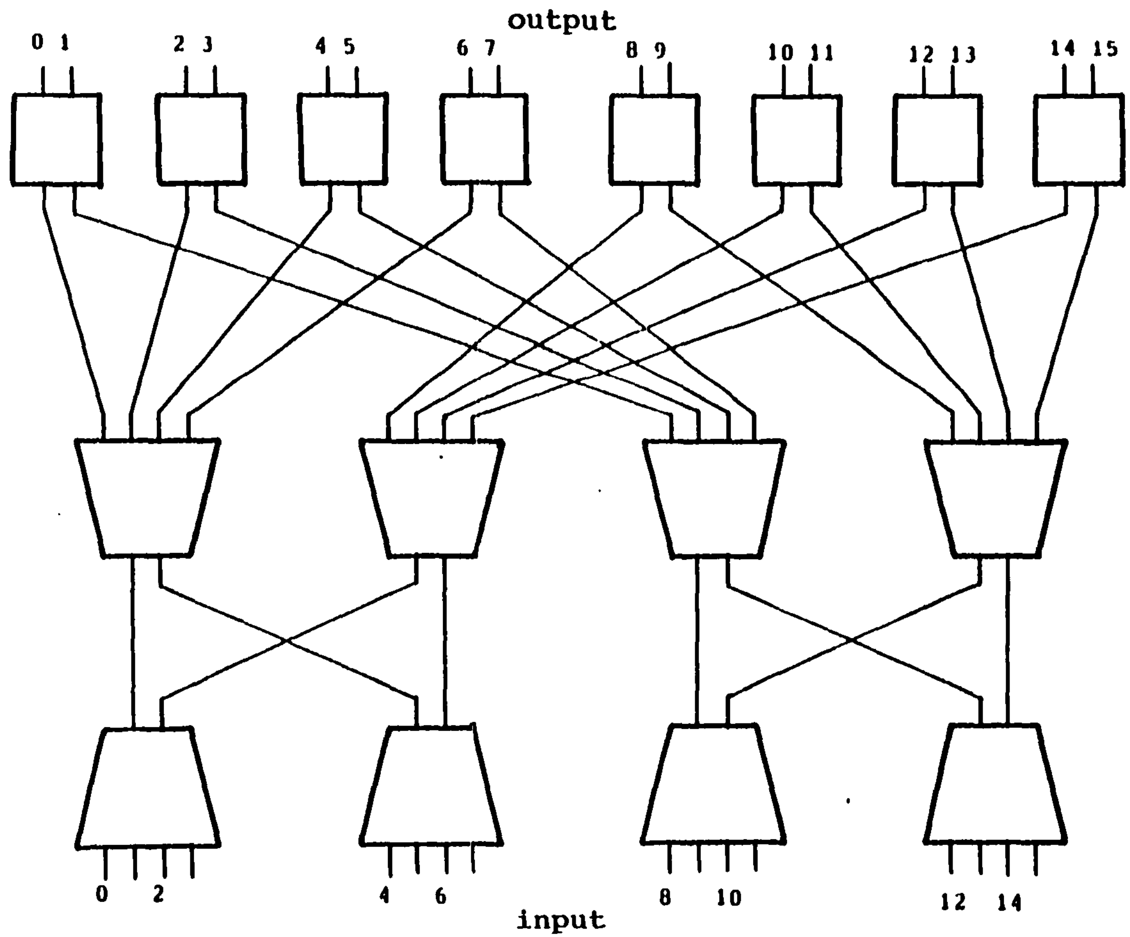


Figure 3.8(b): The corresponding network of Figure 3.8(a).

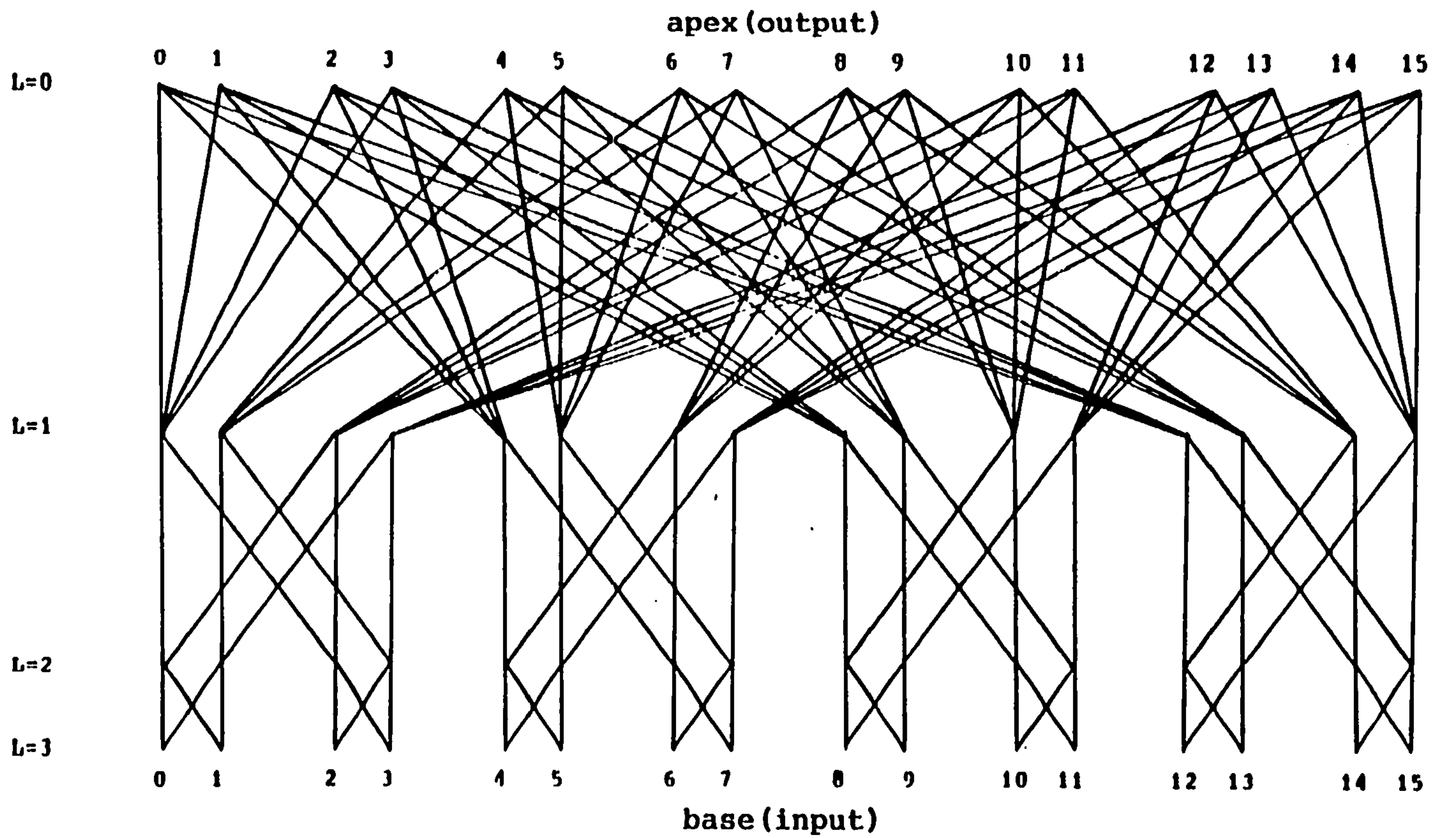


Figure 3.9(a): An example of Strongly Rectangular banyan where F=(4,2,2) and S=(4,2,2).

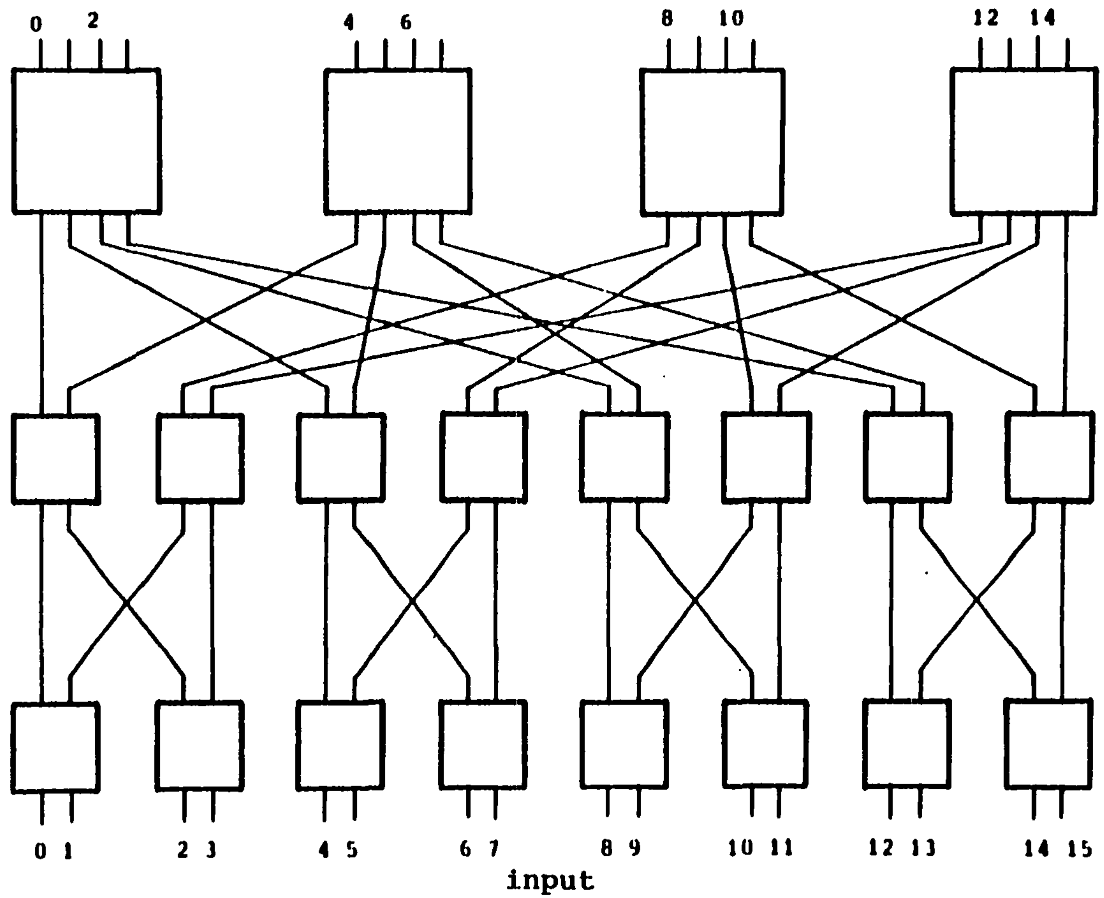


Figure 3.9(b): The corresponding netwok of Figure 3.9(a).

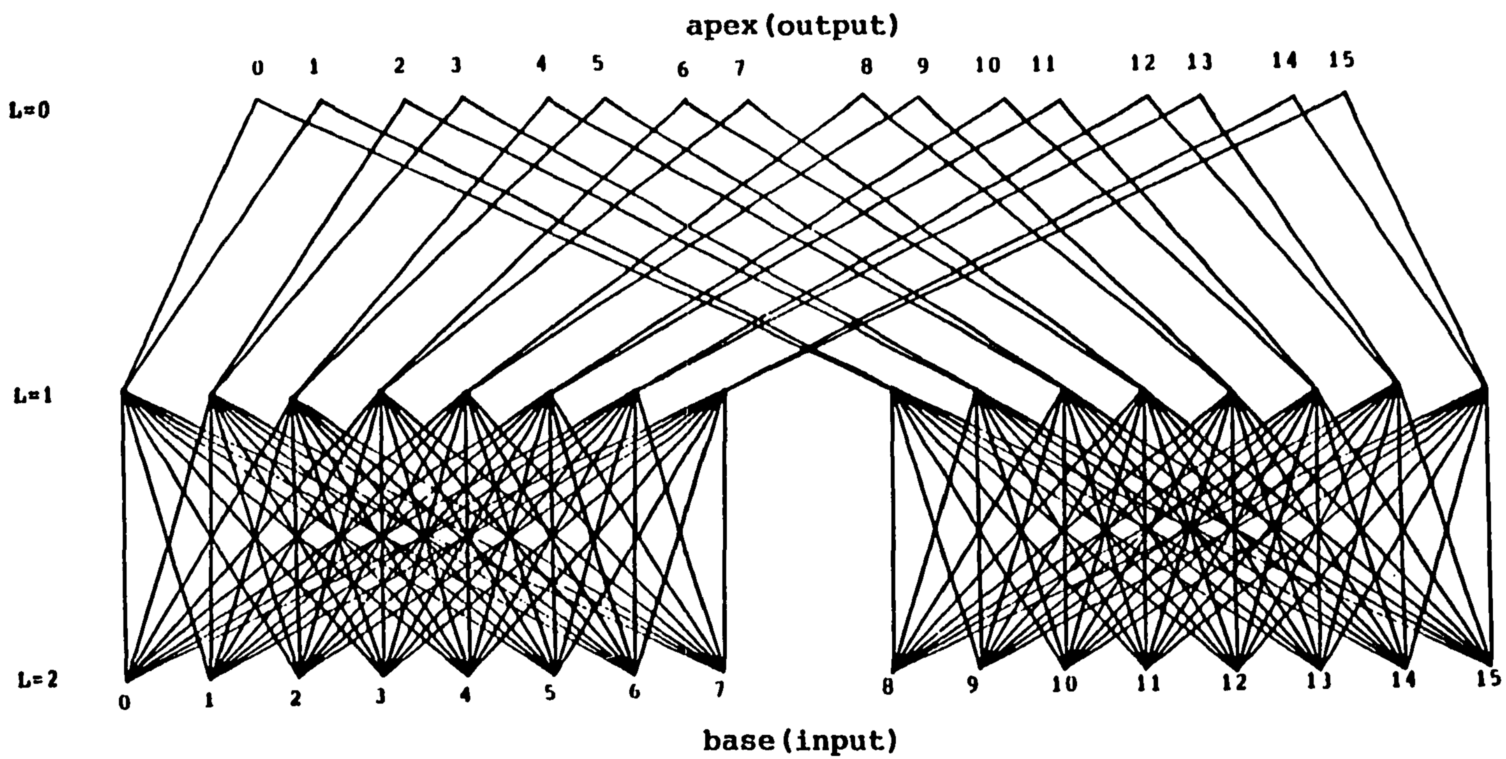


Figure 3.10(a): An example of Weakly Rectangular banyan, where F=(2,8) and S=(2,8).

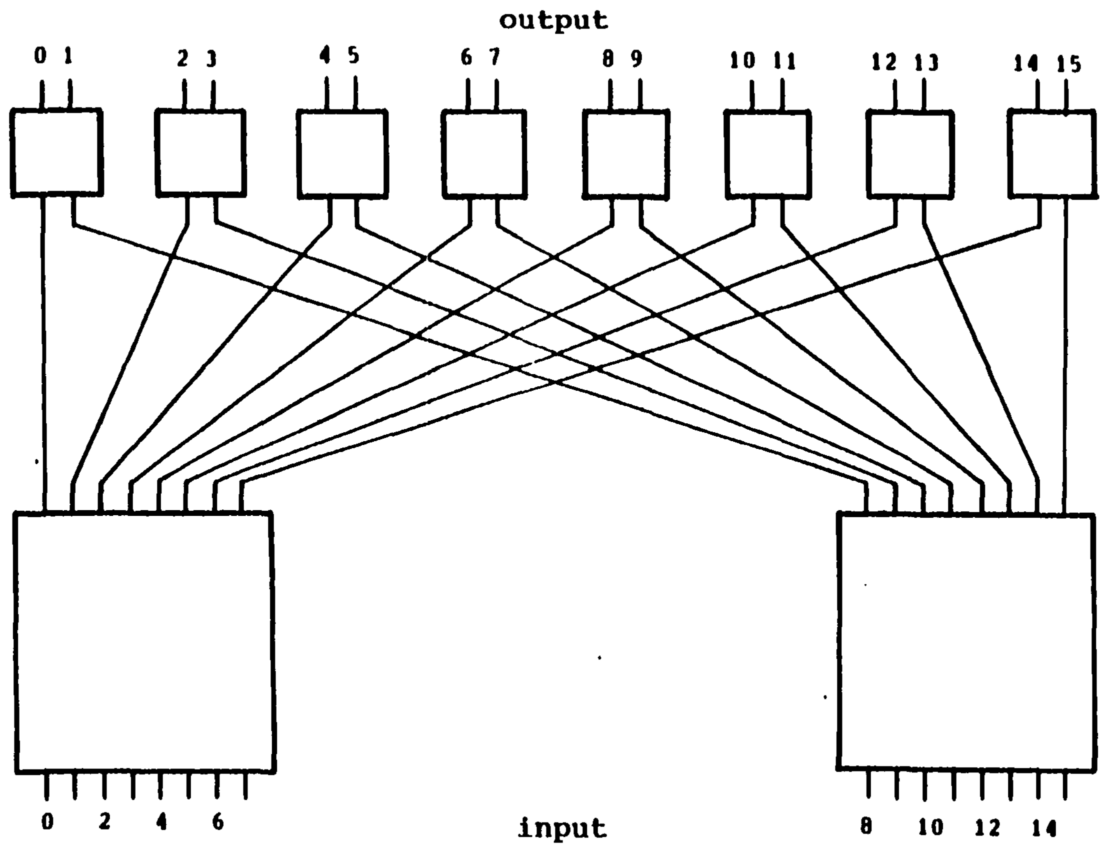


Figure 3.10(b): The corresponding network of Figure 3.10(a).

er new base-apex connection can be made if the the new connection requires a link in use by the first connection. The second connection is blocked by the first one. As a result less path blockage could imply more permutations to be realized by the network.

In this chapter, the concept of an L-stage block-structured network and the relation between this class and SW-Banyan network will be studied. Computing the number of permutations realized by an L-stage block-structured network is not trivial and often gives rise to an interesting class of enumeration problems. It will be shown that the method due to Bhuyan and Agrawal[B7] for computing the number of permutations realized is incorrect. A new method for computing the combinatorial power of these networks is introduced. We illustrate this method using a number of examples. Finally, the trade off between path blockages and combinatorial power of a network will be discussed.

3.3 SW-BANYAN AND BLOCK-STRUCTURED NETWORKS:

Let A be the set of apexes(output terminals) and B be the set of bases(input terminals). Given $b \in B$ and $a \in A$, define $R_x(a)$ to be a set of all nodes at level x ($0 \leq x$ le L) which are reachable from apex a, and $R_x(b)$ to be a set of all nodes at level x ($0 \leq x$ le L) which are reachable from base b. Clearly,

$R_0(a)=\{a\}$

$R_L(a)=B$, the set of input terminals(bases).

$R_0(b)=A$, the set of output terminals(apexes).

$R_L(b)=\{b\}$

Definition 3.6:

An L-stage uniform Banyan is called an SW-Banyan if and only if either one or both of the following conditions is true.

1) $R_x(a_i)=R_x(a_j)$ or $R_x(a_i) \cap R_x(a_j) \neq \emptyset$

2) $R_x(b_i)=R_x(b_j)$ or $R_x(b_i) \cap R_x(b_j) \neq \emptyset$

for all a_i, a_j εA, b_i, b_j εB and $0 \leq x \leq L$.

If an L-stage uniform Banyan network satisfies either 1) or 2) but not both, then it is called a <u>one-way</u> SW-Banyan network. If 1) is true then it is called an input SW-Banyan and if 2) is true then it is called an output SW-Banyan. If both 1) and 2) are true, then it is called a Two-way SW-Banyan.

From the definition of the L-stage Uniform NxM Banyan networks it follows[D1] that

$$N = \prod_{i=1}^{L} s_i \qquad\qquad M = \prod_{i=0}^{L-1} f_i.$$

Given this factorization for N and M, an L-stage input block-structured network is defined recursively as follows: The input stage (stage L) is made of N/s_L cross-bar switches, each of size s_L x f_{L-1}. This stage is followed by f_{L-1} number of (L-1) stage input block-structured networks called blocks for simplicity. Each of these blocks at its input stage contains $N/s_L s_{L-1}$ cross-bar switches of size $s_{L-1} x f_{L-2}$. The recursion stops at stage 1. Refer to Figure 3.11. An output block-structured network is likewise defined, Figure 3.12.

The output terminal of switches in stage i are connected to the input terminals of blocks in stage i+1. An important consequence of blocking is that the interconnections between stages are confined to switches within a given block. In other words, there are no inter-block connections.

We now define a general class of interconnection schemes, also called a link permutation, between stages. The link permutation between input stage L and the set of all f_L number of blocks (each of which is an (L-1) input block-structured network) is said to be <u>distributive</u> if and only if each of the f_{L-1} output terminals of a switch in stage L is connected to one of the f_L blocks. Within each block the interconnections are likwise defined.

Similarly, one can readily define output block-structured networks. An immediate consequence of the above definition is the following:

Theorem 3.1:

The interconnection graph of an NxN L-stage input(output) block-structured network with a distributive link permutation is an output(input) SW-Banyan network.

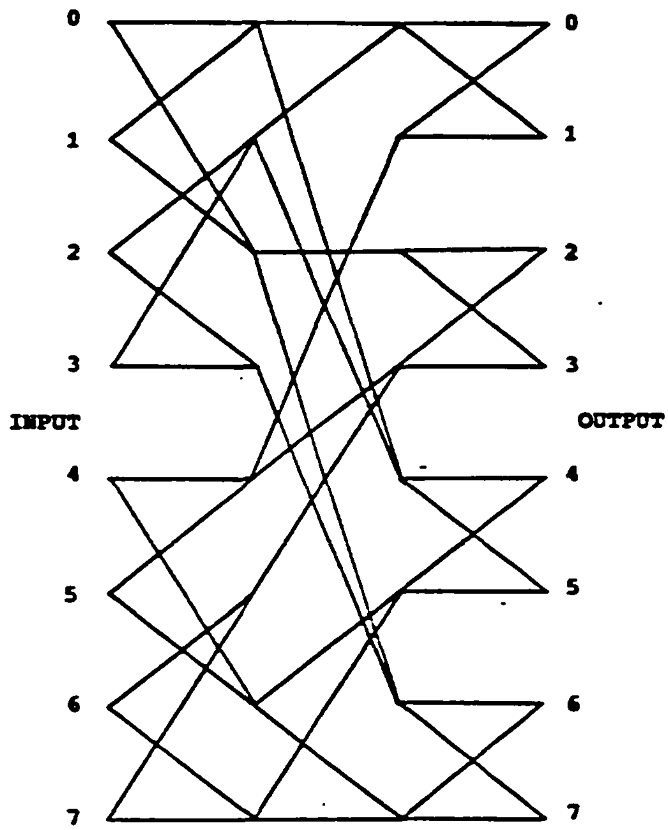


Figure 3.11(a): An example of Input block-structured banyan, where F=(2,2,2) and S=(2,2,2).

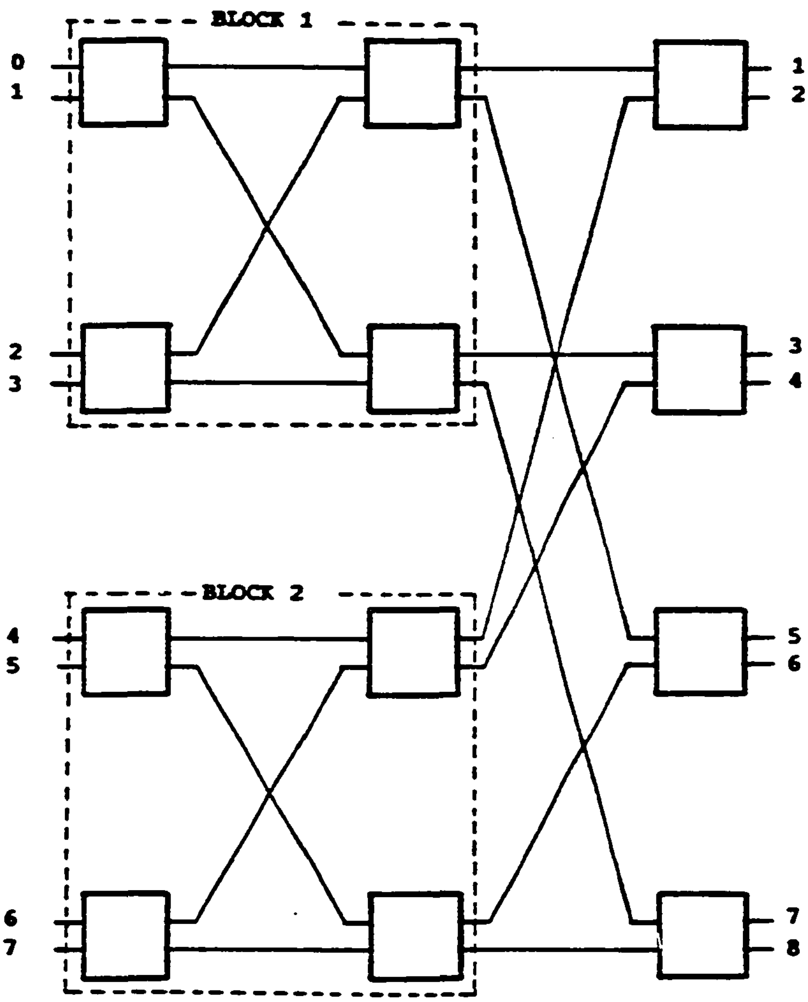


Figure 3.11(b): The corresponding network of Figure 3.11(a).

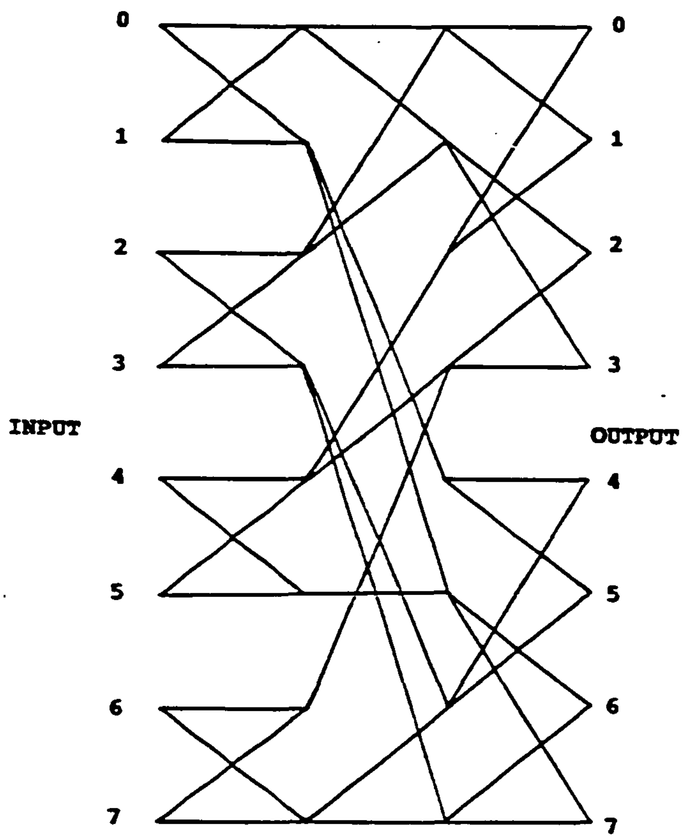


Figure 3.12(a): An example of Output block-structured banyan, where F=(2,2,2) and S=(2,2,2).

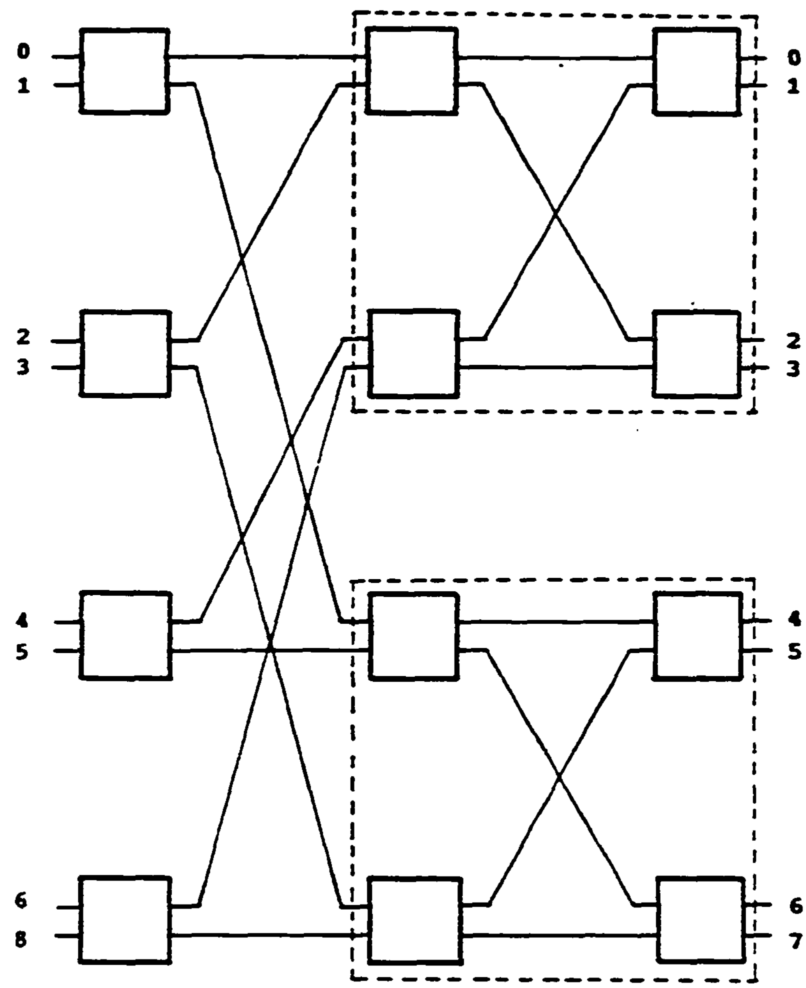


Figure 3.12(b): The corresponding netwok of Figure 3.12(a).

We now define a special class of distributive link permutations as follows.

<u>Condition C</u>:

The i^{th} output terminal of every switch stage 1 is connected to the i^{th} block and likewise within each block. Combining the above condition C and the definition given in [D3] immediately leads to the following:

Theorem 3.2:

The NxN input or output block-structured network with the link permutation satisfying the condition C is in fact a Delta network. Further, if the network is both input and output block-structured then condition C implies that it is a Δ^2 network, Figure 3.13.

Thus, every input (output) block-structured network admits a destination tag based routing algorithm. Refer to Figure 3.14 for an illustration. Thus given the factors of N the concept of block-structured networks with distributive link permutations provides a ready means for synthesising SW-Banyan networks.

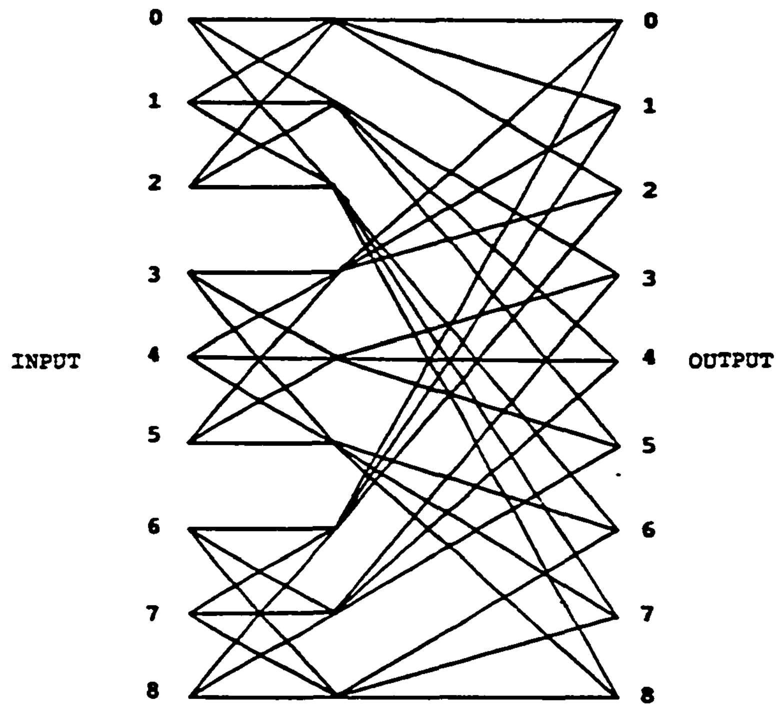


Figure 3.13(a): An example of Two way block-structured banyan, where F=(3,3) and S=(3,3).

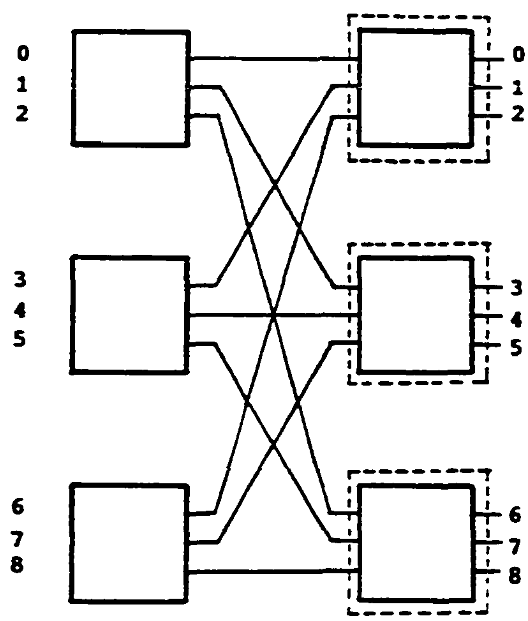


Figure 3.13(b): The corresponding network of Figure 3.13(a).

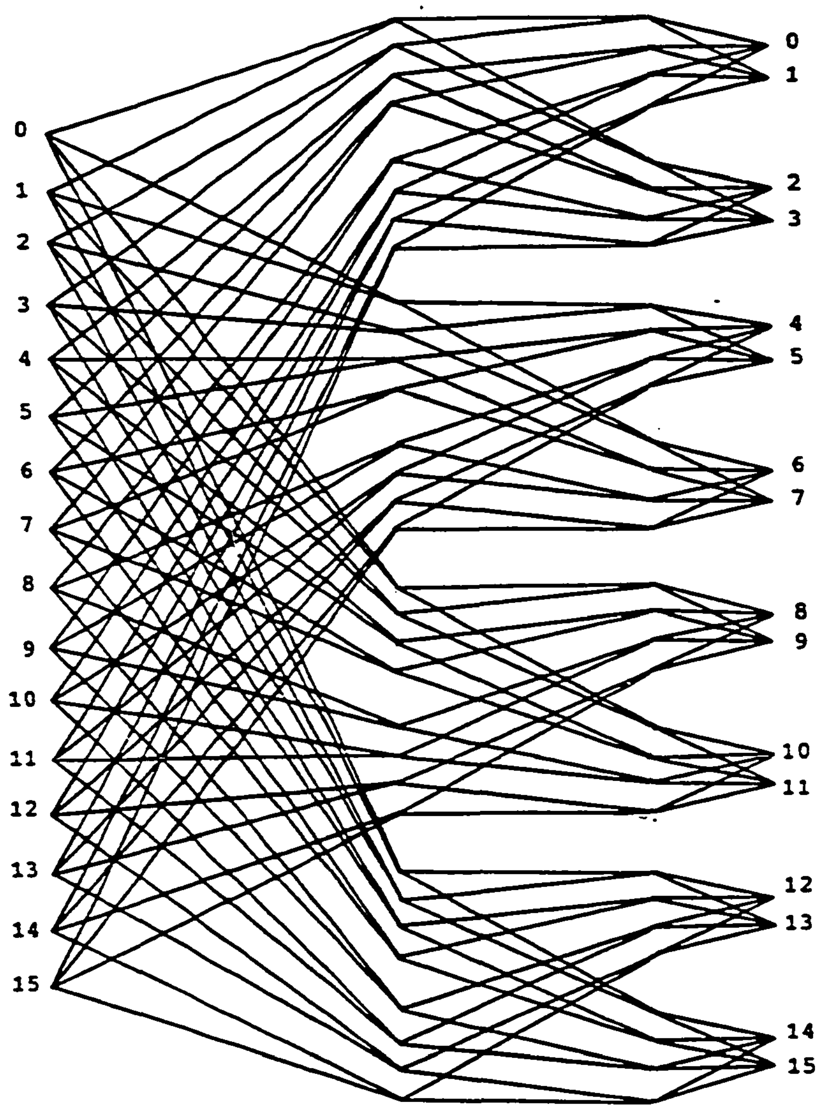


Figure 3.14(a): An example of non-regular,non-rectangular block-structure banyan where F=(2,2,4) and S=(4,2,2).

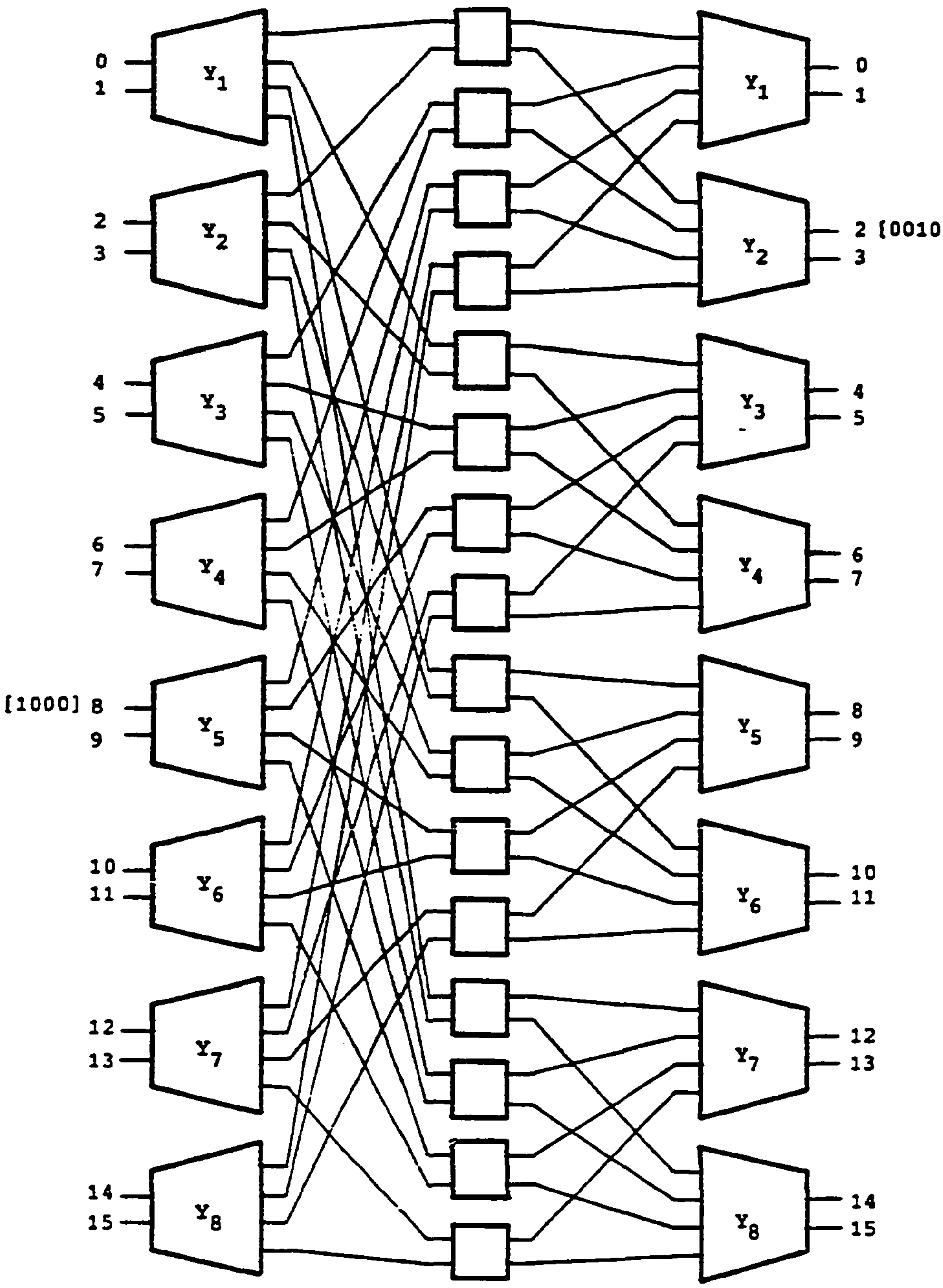


Figure 3.14(b): The corresponding network of Figure 3.14(a)

3.4 COMPUTING THE NUMBER OF PERMUTATIONS REALIZED BY A CLASS OF SW-BANYAN NETWORK:

Consider a non-prime integer $N \geq 2$. Let

$$N=n_1 \times n_2 \times \ldots \times n_L$$

be a factorization of N, where all factors are not necessarily distinct. Let $n=(n_1,n_2, \ldots ,n_L)$ be a vector of L factors. Let $F=(f_0,f_1, \ldots ,f_{L-1})$ and $S=(s_1,s_2, \ldots ,s_L)$ be two not necessarily distinct permutations of the components of n. Given S and F, consider an L-stage SW-Banyan network where the i^{th} stage is made up of complete cross-bar switches of size $s_i \times f_{i-1}$ [Gl,Dl]. Clearly, the number of switches in stage 1 is N/s_L and in stage i is given by $Ns_1s_2 \ldots s_{i-1}/f_1 f_2 \ldots f_i$ for $1 \leq i \leq L$. Let r be the number of distinct permutations of the L factors of N, then there are r^2 such networks. By exhausting the set of all factors of N, a class of all SW-Banyan networks with N inputs and N outputs can be obtained.

Example: If N=2x2x4 then there are three distinct permutations, (2,2,4), (4,2,2) and (2,4,2) and hence there are nine networks. For N=16 there are 16 different SW-Banyan networks including the 16x16 cross-bar switch which are illustrated in Table 3.1 This class of networks includes all the well known Delta networks such as Omega networks, indirect binary networks as well as the class of expanding and contracting SW-Banyan networks[Dl,K4].

TABLE 3.1

PROPERTIES OF SW-BANYAN NETWORKS WITH N=16.

Instances	f	s	Number of Switching Elements	Path Blockage	α(N), Number of Permutation Realized
1	(16)	(16)	256	0	16!
2	(4,4)	(4,4)	128	9	$2^{24}\text{x}3^{8}$
3	(2,8)	(2,8)	160	7	$(8!)^{2}\text{x}$ $(2!)^{8}=$ $2^{22}\text{x}992,25$
4	(2,8)	(8,2)	256	1	2^{16}x $f(8,\{0,1\},2),$ $=187530840$
5	(8,2)	(8,2)	160	7	$(8!)^{2}\text{x}$ $(2!)^{8}=$ $2^{22}\text{x}992,25$
6	(8,2)	(2,8)	64	49	0
7	(2,4,2)	(4,2,2)	160	9	$2^{24}\text{x}f^{2}$ $f(4,\{0,1\},2),$ $= 90$
8	(2,4,2)	(2,4,2)	128	13	$2^{16}\text{x}4!=$ $2^{28}\text{x}3^{4}$
9	(2,4,2)	(2,2,4)	96	25	0
10	(2,2,4)	(2,4,2)	160	9	$2^{24}\text{x}f^{2}$ $f(4,\{0,1\},2),$ $= 90$
11	(2,2,4)	(4,2,2)	192	5	2^{16}x $f(4,\{0,1,2\},4)$
12	(2,2,4)	(2,2,4)	128	13	$2^{28}\text{x}3^{4}$
13	(4,2,2)	(2,2,4)	80	33	0
14	(4,2,2)	(2,4,2)	96	25	0
15	(4,2,2)	(4,2,2)	128	13	$2^{28}\text{x}3^{4}$
16	(2,2,2,2)	(2,2,2,2)	128	17	2^{32}

If F=S that is $s_i=f_{i-1}$ for $1 \leq i \leq L$, then each stage is made of square cross-bar switches. If $S \neq F$, that is there exists j, $1 \leq j \leq L$ such that $s_j \neq f_j$, then there is at least one stage made of rectangular cross-bar switches. For example, consider S=(2,2,4) and F=(2,4,2), then stage one is made of 2x2 switches and the second and third stages are made of switches of size 2x4 and 4x2, respectively. Refer to Figure 3.15. If $S \neq F$, the network is called an Expanding-contracting SW-Banyan network[D1].

The path properties of Expanding-contracting networks have been analyzed by DeGroot [D1]. A parameter called the <u>blockage</u> is introduced which is defined as the number of paths in the network that are blocked when an input/output pair of terminals is connected. For any x, $0 \leq x \leq L$, define a(x), the number of apexes reachable by a node at level x and b(x), the number of bases reachable by a node at level x . Then bl(x), the number of blocked paths that pass through the busy node at level x is simply (a(x)-a(x-1))(b(x)-1), for $1 \leq x \leq L-1$. Through several examples, [D1] it is shown that for a given factorization, the Expanding-contracting SW-Banyan networks have less blockage compared to the rectangular Banyan networks with the same number of inputs and outputs. Table 1 illustrates the blockage for the set of all 16x16 SW-Banyan networks using DeGroot's formula. While there is a reason to believe that networks with less blockage may realize more number of per-

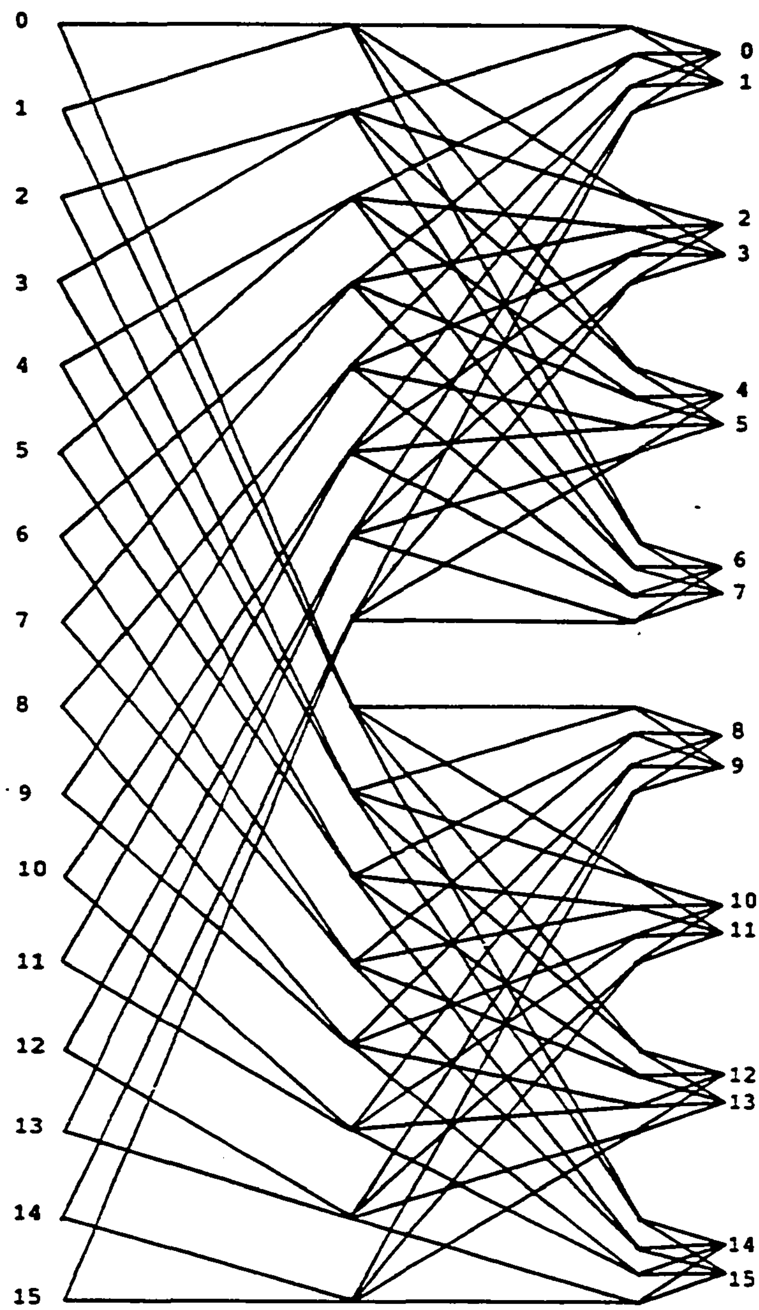


Figure 3.15(a): An example of Expanding and Contracting banyan, where F=(2,2,4) and S=(2,4,2).

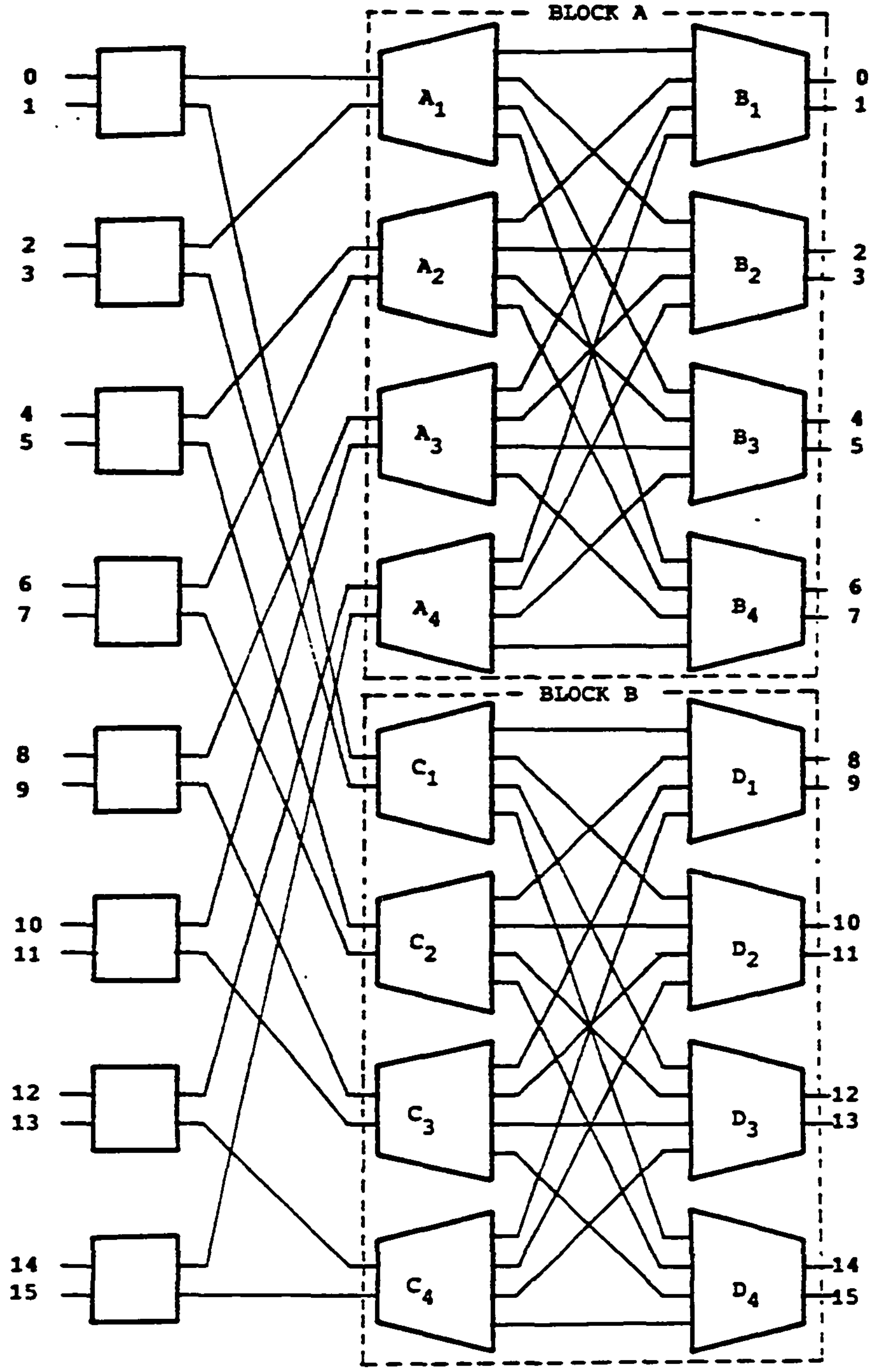


Figure 3.15(b): The corresponding network of Figure 3.15(a).

mutations, computing the latter quantity for the Expanding-contracting SW-Banyan networks is by no means trivial and in fact it is equivalent to a class of enumeration problems which are of independent interest. Two different cases will be analyzed, one for S = F and the other for S $\neq$ F.

Case A: If F = S, then all the switches of each stage have the same size, i.e. $(s_i x f_{i-1})$, $1 \leq i \leq L$, where L is the number of stages of the network. Each switch is capable of realizing $s_i!$. Then the number of permutations $\alpha(N)$ realized by an NxN SW-Banyan network is given by:

$$\alpha(N) = \prod_{i=1}^{L} (s_i!)^{m}$$

where m_i is the number of switches in stage i. The instances 1,2,3,5,8,12,15 and 16 in Table 1 correspond to this case.

Case B: If F $\neq$ S, then it is not always necessary that $\alpha(N) \geq 0$. For example if F=(8,2) and S=(2,8), then stage 1 is made of two copies of 2x8 switches and stage 2 contains two copies of 8x2 switches. Figure 3.16 illustrates the network. Since the output of stage one contains only 4 output ports, 12 of the 16 inputs are blocked by stage 1 and hence $\alpha(N)=0$. Consequently, the following theorem is immediate.

Theorem 3.3:

Let the j^{th} stage of an L-stage NxN SW-Banyan network is made of cross-bar switches of size $s_j x f_{j-1}$, $1 \leq j \leq L$.

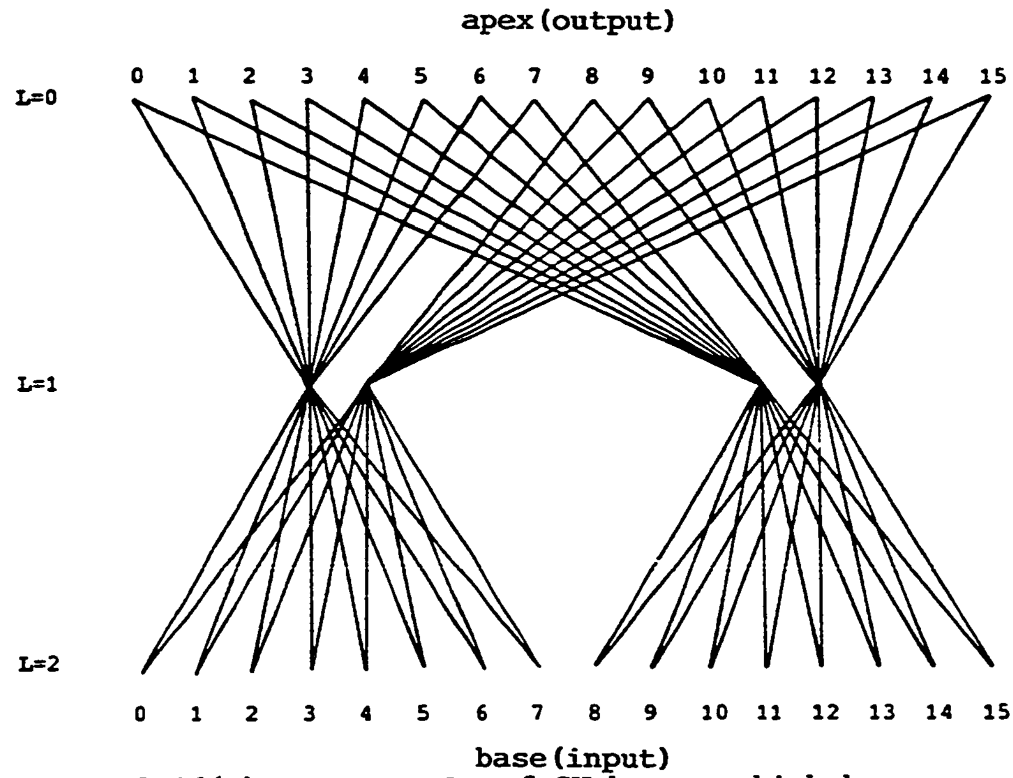


Figure 3.16(a): An example of SW-banyan which has $\alpha(N)=0$, where $S=(8,2)$ and $F=(2,8)$.

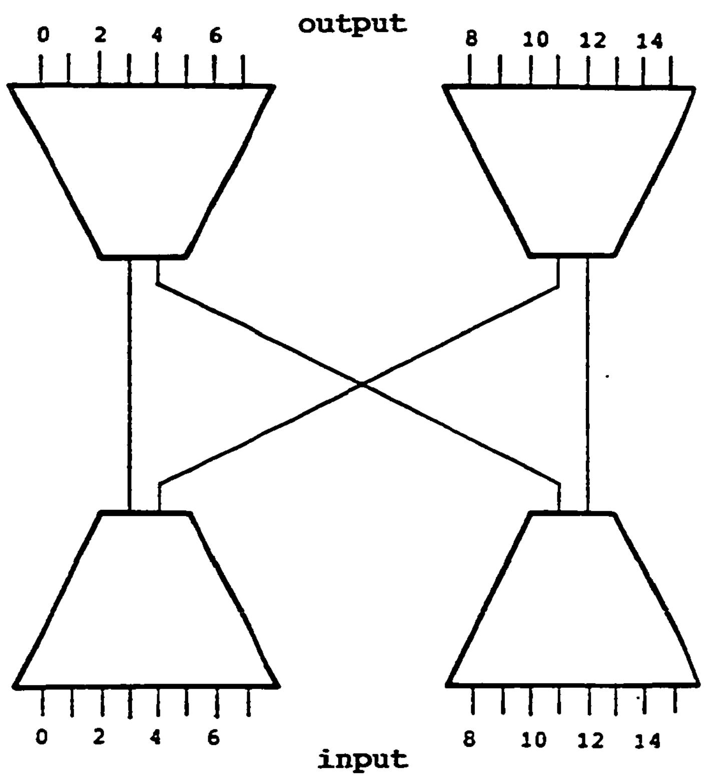


Figure 3.16(b): The corresponding network of Figure 3.16(a).

Then a necessary and sufficient condition for $\alpha(N) \geq 0$ is that $m_j x s_j \geq N$, where m_j is the number of switches in stage j.

The above condition is equivalent to the following statement:

$$s_1 s_2 \ldots s_j \geq f_1 f_2 \ldots f_j$$

It is easy to see that $\alpha(N) = 0$ for instances 6,9,13 and 14. The remaining four instances namely 4,7,10 and 11 satisfy the condition of the above theorem. The computation of $\alpha(N)$ for later instances of the SW-Banyan network gives rise to an interesting class of enumeration problems. We illustrate using one example of the instance 10, where F=(2,2,4) and S=(2,4,2). This network is given in Figure 3.15. Refering to Figure 3.15, consider the first block which consists of switches A_1 to A_4 and B_1 to B_4. The structure of the network induces the following natural constraints. Each of the switches $A_i (1 \leq i \leq 4)$ has four output ports, since there are only two input ports to A_i, only two out of four output ports can actually carry the two inputs. Likewise, each switch $B_j (1 \leq j \leq 4)$ through the link permutation can receive only four inputs, since it has only two output links. To avoid the blockage within the switches $B_j (1 \leq j \leq 4)$, it is required that B_j receives exactly two inputs from any two of the four input ports. A similar requirement holds for the switches in the second block. In other words, each switch in stage 2 and 3 receives exactly two inputs and acts as though if were a 2x2 switch. Thus, within the first block, the number of permutations realized by the network crucially depends on the number of ways in which each of the switches A_i sends its two outputs to the switches B_j under the constraint that each B_j can receive exactly two inputs. This number, as it can be seen, is the same as the number of distinct 4x4 matrices where the elements of the matrices belong to the set {0,1} and the sum of each row and the sum of each column is equal to 2.

The following matrix corresponds to the first(second) block.

$$
\begin{array}{c c}
 & \begin{array}{cccc} A_1 & A_2 & A_3 & A_4 \end{array} \\
\begin{array}{c} B_1 \\ B_1 \\ B_1 \\ B_1 \end{array} &
\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}
\end{array}
$$

where the i^{th} row corresponds to the i^{th} switch (B_i) and the j^{th} column corresponds to the j^{th} switch (B_i). The $(i,j)^{th}$ element is 1 if and only if B_j receives an input from A_i.

Let f(a,A,b) be the number of distinct matrices whose elements belong to the set A and every row and column sum is b. For the above matrix, this number is f(4,{0,1},2). It can be easily seen that this number is 90. Notice that the first block and the second block of Figure 3.15 can be controlled independently. With this constraint, each of the 4x2 and 2x4 switches acts as a 2x2 switch only. Thus each of the switches at stage 1, 2, and 3 realizes two permutations, there are 24 switches involved in the network, therefore the number of permutations realized by the network is:

$$\alpha(N) = 2^{24} f^2(4,\{0,1\},2).$$

Referring to Table 3.1, the networks corresponding to the instance 4 and 13 give rise to the computation of f(8,{0,1},2) and f(4,{0,1},4), respectively. These latter enumeration problems are of intrinsic interest.

Table 3.1 illustrates $\alpha(N)$, path blockage[D1] and the cost of each network. The cost of a network is measured in terms of the total number of switching elements and is computed as:

$$C = \sum_{i=1}^{k} m_i f_i s_i.$$

It follows that the less blockage means the more combinatorial power measured in terms of the number of permutations realized. Further, among the networks with the same cost(instances 2,8,12,15,and 16) instance 2 has a minimum number of blockage and a maximum value of $\alpha(N)$ and instance 16 has a maximum number of blockage and a minimum value for $\alpha(N)$. Another fact that emerges from this analysis is that among the networks with the same cost, those with the larger switches have less blockage and a larger value for $\alpha(N)$. Another immediate consequence of our analysis is that the formula for the number of permutations given in Theorem 3 of [B7] is incorrect, as the following example illustrates.

Example:

Consider the network of block A of Figure 3.15, it consists of two stages of 2x4 and 4x2 switches, respectively. Bhuyan and Agrawal [B7] showed that the number of permutation realized by a generalized interconnection network can be obtained by,

$$P = \prod_{i=1}^{r} S_i^{\,k_i}$$

where k_i is the number of switches at the i^{th} satge and

$$S_i = \binom{n_i}{m_i} m_i! \quad \text{for} \quad m_i \leq n_i$$

$$S_i = \binom{m_i}{n_i} n_i! \quad \text{for} \quad m_i > n_i$$

where S_i is the number of permutations achieveable by an m_i x n_i cross-bar switch at the i^{th} stage and r is the number of stages of the network.

For the network of block A of Figure 3.15, $m_1=2$, $n_1=4$, $m_2=4$, $n_2=2$, $r=2$ and $k_i=4$, for $i=1,2$ then

$$S_1 = \binom{4}{2} 2! = 12$$

$$S_2 = \binom{4}{2} 2! = 12$$

$$P = \prod_{i=1}^{r} S_i^{k_i} = (12)^4 (12)^4 = 12^8$$

The number of permutations realizeable by block A of Figure 3.15 is computed by the new method and it is 90.

3.5 CONCLUSION:

This chapter introduced the concept of block-structured networks where the link permutations satisfy a "distributive" condition. The intimate relation between block-structured networks and other well known classes of networks are shown. The main result of this chapter consists in computing the number of permutations realized by the SW-Banyan networks. It is shown that the known method for computing this number is incorrect and the correct method is given. This method gives rise to an interesting class of enumeration problems. Finally, combinatorial power, cost and path blockage are discussed.

Chapter IV

ON THE COMPLEXITY OF VERIFICATION OF REARRANGEABLE NETWORKS

The group methodology and combinatorics have been used to prove the rearrangeability of three stage Clos[Cl] networks. Clos in 1953 [Cl], for the first time, in a fundamental paper exhibited a three stage switching network which is non-blocking. Later, Benes in a series of papers analyzed a rich class of rearrangeable switching networks[Bl,B2,B3,B4]. The concepts and calculations from group theory have been used to determine the rearrangeability of a switching network made of stages of square switches[B3,B5,Ol]. It has been shown by using the Slepian-Duguid[S5] rearrangeability theorem that a cascade of two NxN networks, alternatively connected between three stages of switches, is rearrangeable if it satisfies a set of group theoretic conditions. The group methodology and mathematical notations are given in section one. In section two the above criterion will be analyzed. This new condition for rearrangeability is too sufficient and is not often satisfied by many rearrangeable networks. The complexity of the verification of rearrangeable networks based on the above condition will be analyzed in section three. Section four provides a counter example to the recent claim due to Afshar[Al] that a feed-back free network that realizes one-to-one and onto mapping of any two input to any output terminals is rearrangeable.

4.1 GROUP THEORY APPLICATIONS:

In this section, the concepts of permutation groups, symmetric groups, composition of groups and switching permutations will be discussed[H1,H3,H4].

4.1.1 Permutation Groups:

Definition: A group G is called a Permutation Group if G is a subgroup of the group of all permutations on a fixed set N.

Definition: The group S_n of permutations on the set {1,2, ... ,n} is called the symmetric group on n objects.

Let G and H be sets of group elements , the G.H is the set of products g.h such that g G and h H. The operation (.) is the composite operation on symmetric groups.

Conventions: It is convenient to use the notation P=(P(1) P(2) ... p(N)) as a permutation function, where

$$P=\begin{pmatrix} 1 & 2 & 3 & \cdots & N \\ P(1) & P(2) & P(3) & \cdots & P(N) \end{pmatrix}$$

Also, it is convenient to use exponent operation for the direct product of a group with itself some number of times.

Thus, if G is a group then G^k is the k-fold direct product of G with itself.

Example:

(321) εS_3 and (213) εS_3, then (321).(213)=(231) εS_3.

4.2 CLASSICAL THEOREM (HALL'S THEOREM):

There exists a system of representatives for a family of sets S_1 S_2 ... S_m if and only if the union of any k of these sets contains at least k-elements, $1 \leq k \leq m$.

Consider a column of r nxn switches as shown in Figure 4.1. This network realizes the subgroup that permutes the first n ports among themselves, the second n ports among themselves, etc. up to the last n ports among themselves. Such a subgroup is called imprimitive subgroup. This subgroup is isomorphic to the r-fold direct product of S_n with itself and is denoted as $(S_n)^r$.

4.2.1 Switch Permutation:

Let N=nr,($r \neq 1$) define the switch permutation function as follows:

$$SW: \{1,2, \ldots ,nr\} \text{--------}> \{1, \ldots ,r\},$$

where each switch is of size nxn, and there are r switches involved. i.e.

$$SW_i = k,$$

where $nk-n+1 \leq i \leq nk$ and $1 \leq k \leq r$.

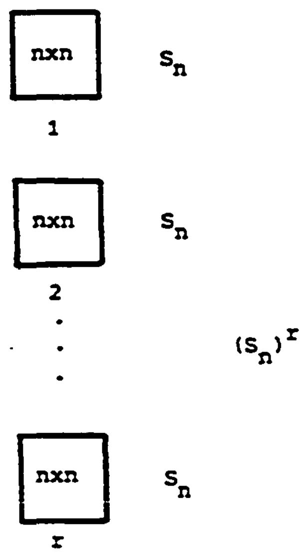


Figure 4.1: A configuration of direct product group interpretation of one stage of square switches.

4.2.2 Hall`s Decomposition:

Let π be a permutation in S_{nr}. A Hall`s decomposition [B6] of π is a partition

$$\pi = \bigcup_{s=1}^{n} P_s$$

of π into n submaps p_s such that for s=1,2, ... ,n, the set

$$q_s=\{(SW_i, SW_j):(i,j) \quad P_s\}$$

is an r-permutation. i.e. $q_s \in S_r$.

Using Hall`s theorem on distinct representatives of subsets implies that every $p \in S_{nr}$ has Hall's decompositions[B6].

Let U be a network. The switch permutations generated by the network is defined to be the set D(U) such that an element $\langle q_1, q_2 \ldots q_n \rangle \in (S_r)^n$ belongs to D(U) if and only if there exists $\alpha \in P(U)$ with Hall's decomposition, where P(U) is the set of all permutations realizable by the network U. Benes[B13] stated the following theorem.

Theorem:

If U and V are networks with nr inputs and nr outputs such that

$$(S_r)^n \subseteq D(U).D(V) \qquad (4.1)$$

then the network obtained by cascading U and V alternately between three stages of r (nxn)-switches, is rearrangeable, as shown in Figure 4.2.

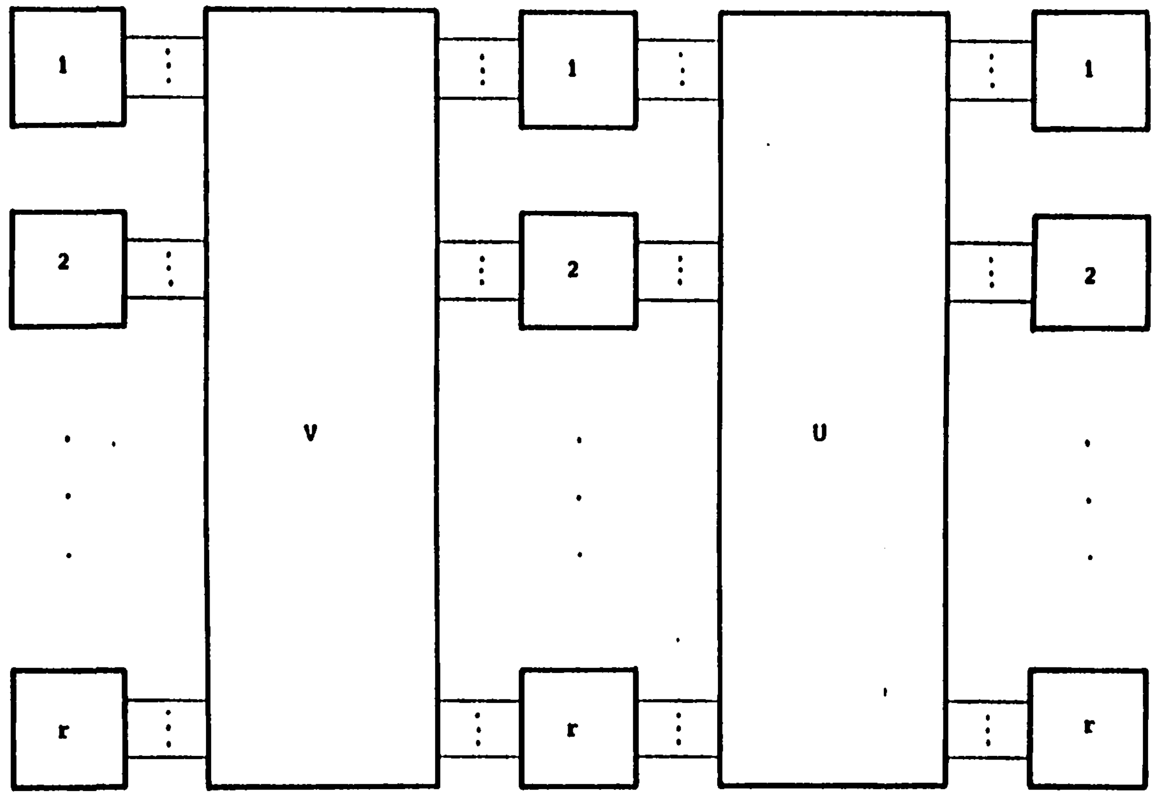


Figure 4.2: A configuration of cascading networks U and V alternately between three stages of nxn switches.

The above verification theorem involves (i) the computation of the set of all permutations realizable by the networks U and V: (ii) the set of all switch permutations (D(U) and D(V)) induced by P(U) and P(V), respectively; (iii) the computation of n-fold products of S_r and testing of the inclusion given in (4.1). The analysis and the complexity of such a verification are given in the next section.

4.3 VERIFICATION COMPLEXITY

For simplicity, let the sub-networks U and V be identical and made of switches of size (2x2), and each stage consists of r=N/2 such switches. The number of stages, k, in U and V are taken to be less than logN. The overall network obtained by cascading the networks U and V alternately between three stages of r=N/2 (2x2) switches has the configuration of Figure 4.3.

The first step in the verification of the Hall's theorem consists in computing D(U) and D(V), the set of all switch permutations realizable by U and V, respectively. D(U) and D(V) are induced by P(U) and P(V), which are the set of all permutations realized by U and V, respectively. The computations require setting of all the switches of U and V in all possible ways. The number of such settings is exponential in number of switches involved in U and V. Thus, if the network U and V each have kN/2 switches, then

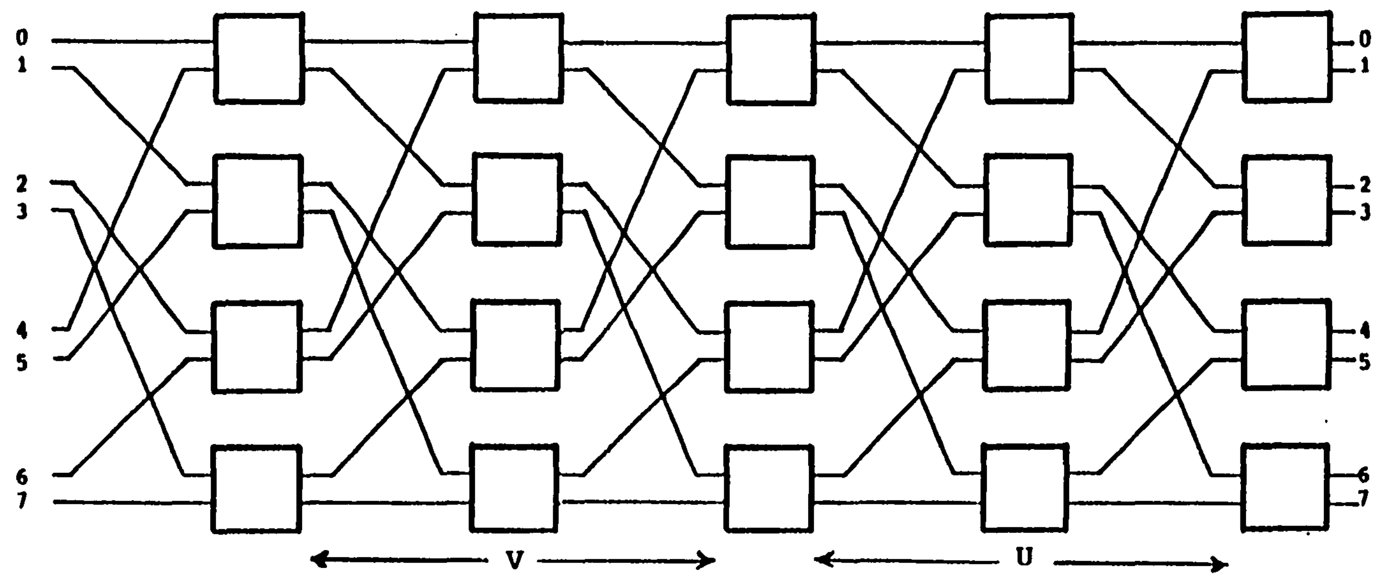


Figure 4.3: A configuration of five stages shuffle-exchange network. U and V are one column of 2x2 switches preceded and followed by shuffle permutation.

P(U) and P(V) each have $2^{kN/2}$ permutations. Thus, finding D(U) and D(V) is exponentially complex. Given a permutation, it has at least one Hall's decomposition. Marshal Hall has given an algorithm [H2] to compute one Hall's decompositon. It involves a matching problem and it is not practical for large values of N. It also gives only one set of distinct representatives while the above verification theorem needs all possible Hall's decompositions.

The second phase of the problem is computing D(U).D(V). The elements of the sets D(U) and D(V) have the form

$$\langle q_1\ q_2\ \cdots\ q_n \rangle$$

where q_i is a switch permutation of size r (N/2 in this case).

Let $|D(U)|=|D(V)|=m$, then computing the product of D(U) and D(V) is proportional to $(r!)^2$.

The third phase is testing whether $(S_r)^n \subseteq D(U).D(V)$ or not. Clearly, $|S_r|=r!$, each element of the set $(S_r)^n$ is a vector of n components and each component is an r-permutation. Computationally, it is not practical for large N to verify whether (4.1) holds or not.

In spite of the computational complexity of (4.1), this new criterion for rearrangeability is too sufficient, that is, this condition is not satisfied by many rearrangeable networks. For example, it can be easily verified that a cascade of five stages of shuffle-exchange stages is universal yet it does not satisfy the new set of Benes conditions, Appendix B. Another example of a rearrangeable network that does not satisfy (4.1) is the 4x4 Benes network, appendix A.

4.4 ROUTING ALGORITHM

Let $\Pi \in S_{nr}$ be a permutation, it has Hall's decomposition

$$\Pi = \bigcup_{s=1}^{n} p_s$$

inducing switch permutation $q_s \varepsilon (S_r)$. Thus, for each s=1,2, ... ,n, there exist a_s and b_s in S_r such that $q_s = b_s a_s$ with $a_s \varepsilon D(U)$ and $b_s \varepsilon D(V)$. For case n=2,

$$\begin{pmatrix} q_1 \\ q_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$$

Given q_1 and q_2, a_1, a_2, b_1 and b_2 have to be computed as.

$$\begin{pmatrix} 1 & 2 & \cdots & r \\ q_1(1) & q_1(2) & \cdots & q_1(r) \end{pmatrix} =$$

$$\begin{pmatrix} 1 & 2 & \cdots & r \\ b_1(1) & b_1(2) & \cdots & b_1(r) \end{pmatrix} \begin{pmatrix} 1 & 2 & \cdots & r \\ a_1(1) & a_1(2) & \cdots & a_1(r) \end{pmatrix}$$

and

$$\begin{pmatrix} 1 & 2 & \cdots & r \\ q_2(1) & q_2(2) & \cdots & q_2(r) \end{pmatrix} =$$

$$\begin{pmatrix} 1 & 2 & \cdots & r \\ b_2(1) & b_2(2) & \cdots & b_2(r) \end{pmatrix} \begin{pmatrix} 1 & 2 & \cdots & r \\ a_2(1) & a_2(2) & \cdots & a_2(r) \end{pmatrix}$$

Since for each $(i,j)\varepsilon p_s$, SW_i and SW_j are connected to the same middle stage switch[B3], we have

$$SW_j=b_s[a_s(SW_i)]$$

or

$$a_s(SW_i)=b^{-1}{}_s(SW_j)$$

then

$$\begin{pmatrix} 1 & 2 & \cdots & r \\ q_1(1) & q_1(2) & \cdots & q_1(r) \end{pmatrix} =$$

$$\begin{pmatrix} a_1(1) & a_1(2) & \cdots & a_1(r) \\ q_1(1) & q_1(2) & \cdots & q_1(r) \end{pmatrix} \begin{pmatrix} 1 & 2 & \cdots & r \\ a_1(1) & a_1(2) & \cdots & a_1(r) \end{pmatrix}$$

and

$$\begin{pmatrix} 1 & 2 & \cdots & r \\ q_2(1) & q_2(2) & \cdots & q_2(r) \end{pmatrix} =$$

$$\begin{pmatrix} a_2(1) & a_2(2) & \cdots & a_2(r) \\ q_2(1) & q_2(2) & \cdots & q_2(r) \end{pmatrix} \begin{pmatrix} 1 & 2 & \cdots & r \\ a_2(1) & a_2(2) & \cdots & a_2(r) \end{pmatrix}$$

such that

$$\begin{pmatrix} a_1(1) & a_1(2) & \cdots & a_1(r) \\ q_1(1) & q_1(2) & \cdots & q_1(r) \end{pmatrix} \text{and} \begin{pmatrix} a_2(1) & a_2(2) & \cdots & a_2(r) \\ q_2(1) & q_2(2) & \cdots & q_2(r) \end{pmatrix}$$

are induced by a permutation $\alpha \varepsilon D(V)$ and

$$\begin{pmatrix} 1 & 2 & \cdots & r \\ a_1(1) & a_1(2) & \cdots & a_1(r) \end{pmatrix} \text{and} \begin{pmatrix} 1 & 2 & \cdots & r \\ a_2(1) & a_2(2) & \cdots & a_2(r) \end{pmatrix}$$

are induced by a permutation $\beta \varepsilon D(U)$. Consider the example of Benes[B3] in Figure 4.6 and consider the permutation,

$$\Pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 5 & 2 & 3 & 6 & 1 & 4 \end{pmatrix}$$

to be realized. Π induces the switch permutation,

$$\begin{pmatrix} q_1 \\ q_2 \end{pmatrix} = \begin{pmatrix} 3 & 2 & 1 \\ 1 & 3 & 2 \end{pmatrix}$$

which can be decomposed as

$$\begin{pmatrix} 3 & 1 & 2 \\ 2 & 1 & 3 \end{pmatrix} \begin{pmatrix} 1 & 3 & 2 \\ 3 & 2 & 1 \end{pmatrix}$$

Given b_s or a_s, there is more than one way of setting the switches. For example, each of the following permutations induces such switch permutations

b_s						a_s					
1	2	3	4	5	6	1	2	3	4	5	6
6	3	1	5	4	2	1	5	6	4	3	2
6	3	2	5	4	1	1	5	6	3	4	2
6	4	1	5	3	2	1	6	5	3	4	2
5	3	1	6	4	2	1	6	5	4	3	2
5	3	2	6	4	1	2	5	6	3	4	1
5	4	1	6	3	2	2	6	5	3	4	1
5	4	2	6	3	1	2	6	5	4	3	1

We need to setup the switches of U or V in such a way that realizes one of the above permutations.

4.5 UNIVERSALITY OF FEED-BACK FREE NETWORKS:

Consider an NxN (N inputs and N outputs) permutation network made of 2x2 switches arranged in k stages (k>0) with N/2 switches in each stage. Let S_{ij} refer to the j^{th} switch at the i^{th} stage where j=1,2, ... ,N/2 and i=1,2, ... ,k. To simplify the notation we use the symbol S_{ij} to refer to both a switch as well as the "state" of the switch. Thus, S_{ij} corresponds to the direct connection (on-state) and $\bar{S}_{ij}$ refers to the crossed connection (off-state) as shown in Figure 4.5.

The number of stages clearly depends on the nature of the network. For Benes network[B4] k=2LogN -1 and for a wide variety of networks including Omega networks, Base-line networks, Indirect-binary n-cube networks etc.[L3,P4,W4], k=logN.

Let $x_1, x_2 \ldots x_N$ and $y_1, y_2 \ldots y_N$ denote the N input and N output terminals, respectively. Given an input/output pair, say (x_p, y_q), in general there could be more than one path from x_p to y_q through the network. Every such path can be uniquely represented by the sequence of states of switches. Let path(x_p, y_q) refer to a path from x_p to y_q. Then

$$\text{path}(x_p, y_q) = T_{1p_1} T_{2p_2} \cdots T_{kp_K}$$

where $T_{ij} \varepsilon \{S_{ij}, \bar{S}_{ij}\}$ and the input x_p connected to the switch S_{1p} and output y_q is connected to the switch S_{kp}. For example , in Figure 4.7, there are two paths between x_1 and y_1:

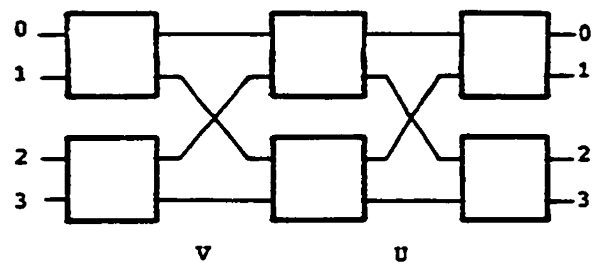


Figure 4.4: A configuration of 4x4 Benes network,
U and V have fixed permutations.

Figure 4.5: A configuration of state of switches.

a) Direct connection (state=1) and
b) Crossed connection (state=0).

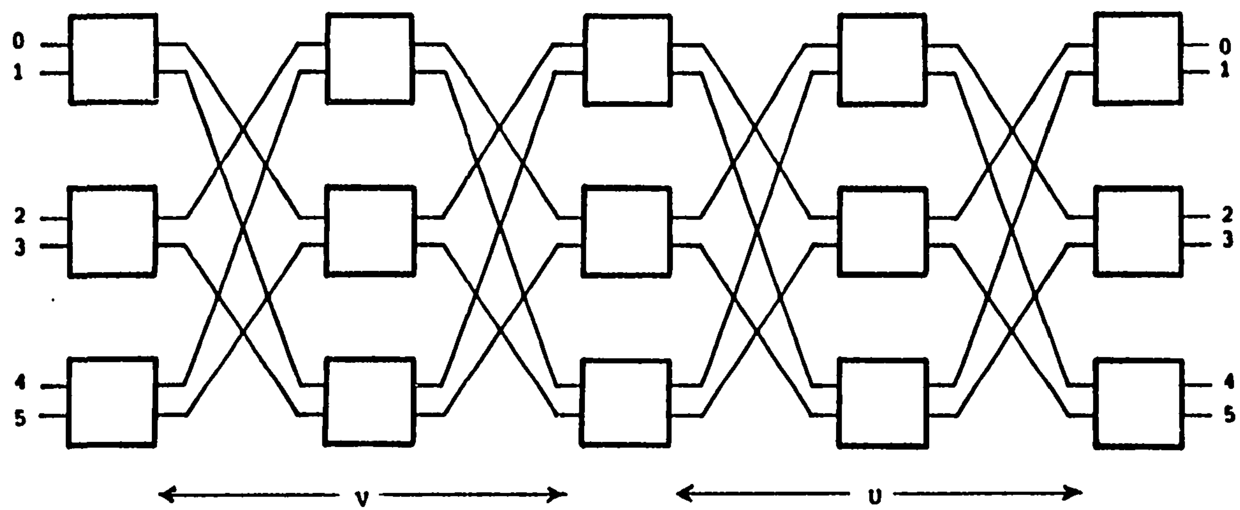


Figure 4.6: Network based on cross-connect field corresponding to the permutation (13)(25)(46).

$$\text{path}(x_1,y_1)= S_{11}S_{21}S_{31} \quad \text{or} \quad \bar{S}_{11}S_{22}\bar{S}_{31}$$

But in Figure 4.8, there is only one path between x_1 and y_1

$$\text{path}(x_1,y_1)=S_{11}S_{21}$$

An NxN permutation network is said to be feed-back free network if there is no path from an input terminal to an output terminal that passes through a 2x2 switch more than once. The networks of Figure 4.7 and Figure 4.8 are examples of feed-back free network and Figure 4.9 is an example of a feed-back networks.

Given an NxN network, if there is a unique path between every possible input/output terminal, then the network is said to possess the unique path property. These networks constitute a class called UP-Networks. The standard Omega network, the base-line network, the indirect binary n-cube network all belong to the class but the standard Benes network does not [B4]. With a view to relating the path properties of a given NxN network to the set of all permutations realized by the network, Afshar[A1] used the concept of an NxN matrix F called the transmitance matrix of the network. Thus, $F=[F_{qp}]$ where F_{qp} is the boolean expression in $\{S_{ij}\ \bar{S}_{ij} \mid i=1,2, \ldots ,k,\ j=1,2, \ldots ,N/2\}$ such that on assigning $S_{ij}=1$ and $\bar{S}_{ij}=0$ to these latter variables, F_{qp} evaluates to be 1 if and only if there exists a path from the input terminal x_p to the output terminal y_q. For the network in Figure 4.8 it can be seen that

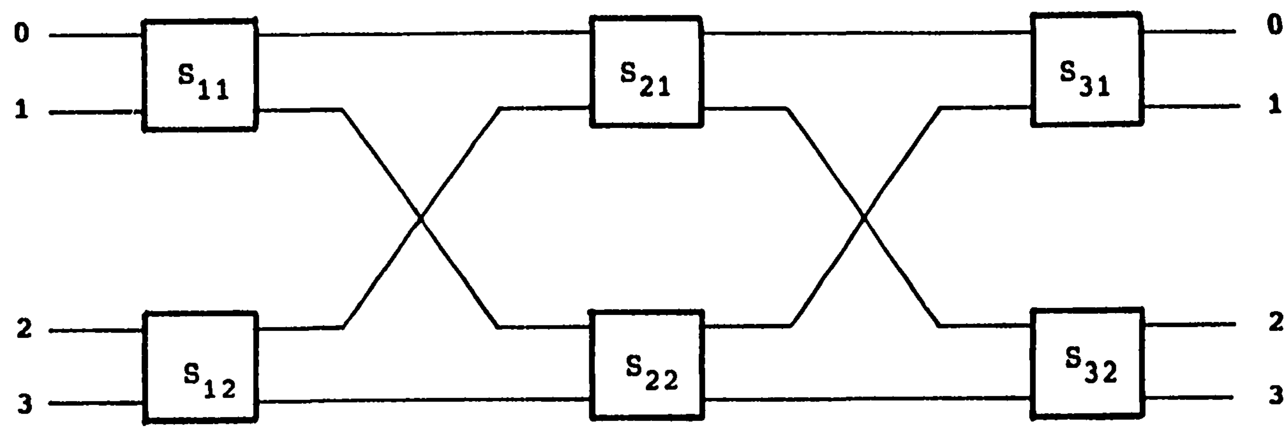


Figure 4.7: An example of4x4 Benes network.

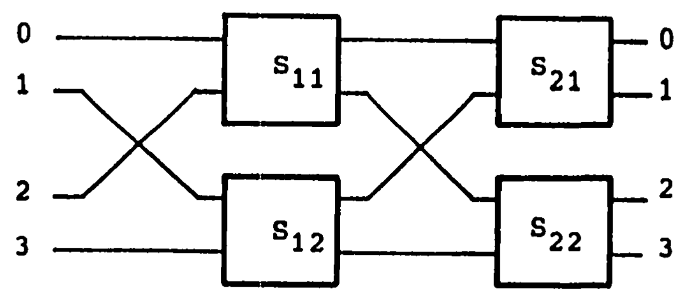


Figure 4.8: Anexample of 4x4 Omega network.

$$F = \begin{bmatrix} s_{11}s_{21} & \bar{s}_{11}s_{21} & s_{12}\bar{s}_{21} & \bar{s}_{12}\bar{s}_{21} \\ s_{11}\bar{s}_{21} & \bar{s}_{11}\bar{s}_{21} & s_{12}s_{21} & \bar{s}_{12}s_{21} \\ \bar{s}_{11}s_{22} & s_{11}s_{22} & \bar{s}_{12}\bar{s}_{22} & s_{12}\bar{s}_{22} \\ \bar{s}_{11}\bar{s}_{22} & s_{11}\bar{s}_{22} & \bar{s}_{12}s_{22} & s_{12}s_{22} \end{bmatrix}$$

Let

$$\alpha = \begin{pmatrix} 1 & 2 & \cdots & N \\ \alpha(1) & \alpha(2) & \cdots & \alpha(N) \end{pmatrix}$$

be a permutation. In [A1], Afshar proved that many standard networks such as indirect binary cube networks, regular Banyan networks with F=S=2, modified data manipulator network, flip networks, Omega networks and base line networks all satisfy the following:

A permutation α is realizeable by a network if and only if

$$F_{\alpha(i),i} * F_{\alpha(j),j} \neq 0 \qquad (4.2)$$

for all i and j, $1 \leq i,j \leq N$, where F is the transmitance matrix of the network.

Afshar proved this result for the base line network and since all the other networks listed above are topologically equivalent to the base line network[W4], the property 4.2 is true for all these networks. Afshar then conjectured that the property 4.2 is true for all feed-back free networks. In the following section, the direct proof for the necessary condition of property 4.2 will be given and the sufficiency part will be shown to be untrue.

4.6 UNIVERSALITY OF UP-NETWORKS:

Theorem 4.1: The permutation α is realizable by a UP-Network if and only if

$$F_{\alpha(i),i} * F_{\alpha(j),j} \neq 0 \qquad (4.3)$$

for all i and j, $1 \leq i,j \leq N$, where F is the transmitance matrix of the network.

Proof: Clearly, if the permutation α is realizable then 4.3 is true for any feed-back free network. To prove the converse, assume that α is not realizable. Then there exist two distinct pairs $(x_i, y_{\alpha(i)})$ and $(x_j, y_{\alpha(j)})$ of input/output terminals such that the PATH$(x_i, y_{\alpha(i)})$ and the PATH$(x_j, y_{\alpha(j)})$ pass through a common switch, say S_{ab}, where $1 \leq a \leq k$ and $1 \leq b \leq N/2$, with the condition that switch S_{ab} is to be "on" for one path and "off" for the other path simultaneously. In other words, there is a conflict in setting up the PATH$(x_i, y_{\alpha(i)})$ and the PATH$(x_j, y_{\alpha(j)})$. From this we obtain(on assigning $S_{ij}=1$ and $\bar{S}_{ij}=0$) that either $F_{\alpha(i),i}=0$ or $F_{\alpha(j),j}=0$. In order to resolve the conflict, one of these paths has to be rerouted. However, rerouting is impossible since the network is a member of the UP-Network class. Thus if α is not realizable, then there exists $i \neq j$ such that

$$F_{\alpha(i),i} * F_{\alpha(j),j} = 0.$$

and the proof is complete.

Now consider the extended Omega network[P4] given in Figure 4.10. It can be easily verified that in this network there are exactly two distinct paths between every input/output pair terminal. For example:

$$\mathrm{PATH}(x_4, y_2) = \begin{cases} S_{12} S_{24} \bar{S}_{33} S_{41} & \text{or} \\ \bar{S}_{12} S_{23} \bar{S}_{31} \bar{S}_{41} & \end{cases}$$

Similarly,

$$\mathrm{PATH}(x_3, y_1) = \begin{cases} S_{12} S_{23} \bar{S}_{31} S_{41} & \text{or} \\ \bar{S}_{12} S_{24} \bar{S}_{33} \bar{S}_{41} & \end{cases}$$

The above two paths can be realized without conflict. The entire transmitance matrix for this feed-back free network can be easily found. It is easy to verify that any two distinct input terminals can be connected to any distinct output terminals. In other words, the transmitance matrix F is such that there exist a setting of the states of the switches that satisfies the condition

$$F_{ab} * F_{cd} = 0. \tag{4.4}$$

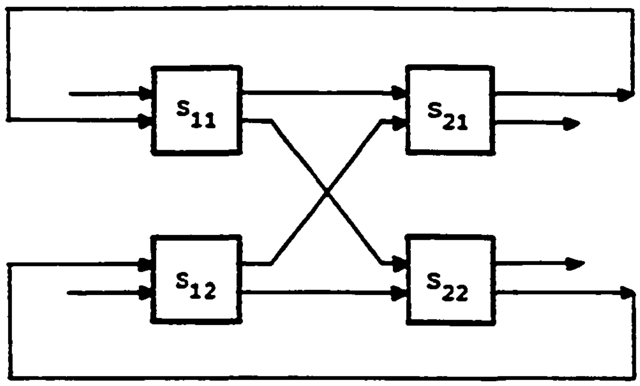


Figure 4.9: An example of feed-back network.

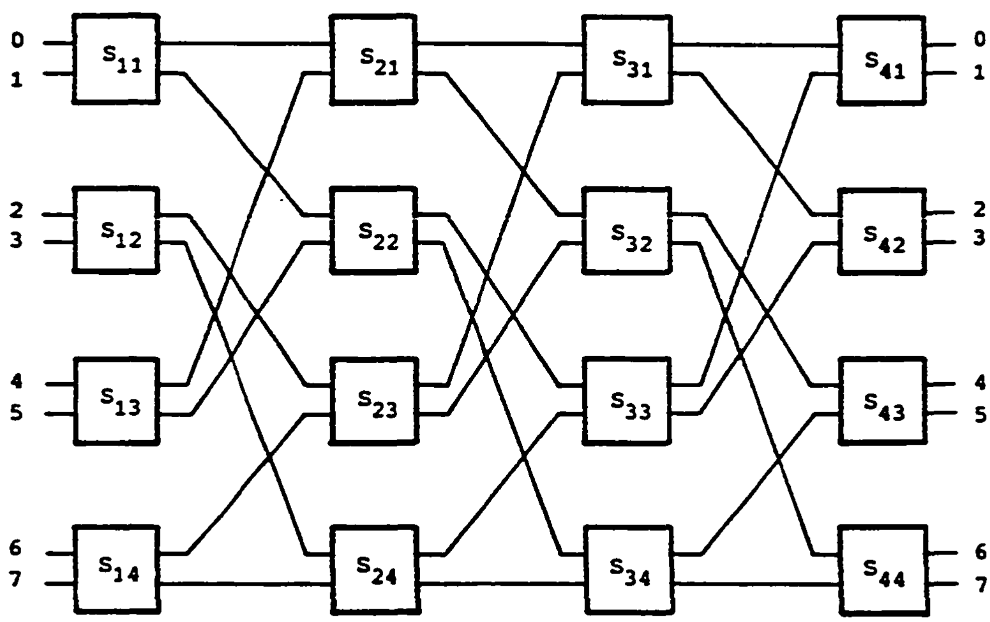


Figure 4.10: An example of extended 8x8 Omega network.

for all a,b,c and d such that $1\leq a,b,c,d\leq N$, $a\neq c$ and $b\neq d$. Notice that (4.4) is in fact the condition of the corollary to the theorem 1 in [A1]. Yet, unfortunately, the following three paths

$$\mathrm{PATH}(x_1,y_1),$$
$$\mathrm{PATH}(x_2,y_2),$$
$$\mathrm{PATH}(x_5,y_3)$$

cannot be realized simultaneously by this network. To verify this observe that:

$$\mathrm{PATH}(x_1,y_1)=\begin{cases} S_{11}S_{21}S_{31}S_{41} & \text{or} \\ \bar{S}_{11}S_{22}S_{33}\bar{S}_{41} & \end{cases}$$

$$\mathrm{PATH}(x_2,y_2)=\begin{cases} S_{11}S_{22}S_{33}S_{41} & \text{or} \\ \bar{S}_{11}S_{21}S_{31}\bar{S}_{41} & \end{cases}$$

and

$$\mathrm{PATH}(x_5,y_3)=\begin{cases} S_{13}\bar{S}_{21}\bar{S}_{31}S_{42} & \text{or} \\ \bar{S}_{13}\bar{S}_{22}\bar{S}_{33}\bar{S}_{42} & \end{cases}$$

Assume the switches S_{11} and S_{41} are on. Then in setting up the $\mathrm{PATH}(x_1,y_1)$ and the $\mathrm{PATH}(x_2,y_2)$, it is necessary and sufficient that the switches S_{21}, S_{31} ,S_{22} and S_{33} are all on. For this assignment, clearly, $\mathrm{PATH}(x_5,y_3)$ cannot be set up. The same conclusion follows if the switches S_{11} and S_{41} are off instead. In other words, any permutation θ with the subassignment $\theta(1)=1$, $\theta(2)=2$ and $\theta(5)=3$ such as,

$$\theta = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 2 & 5 & 6 & 3 & 4 & 8 & 7 \end{pmatrix}$$

cannot be realized by the network in Figure 4.10. Hence, this network is not rearrangeable[B4]. Indeed the network in Figure 4.10 is not a member of the UP-Network class but it is a feed-back free network.

4.7 CONCLUSION:

In this chapter we analyzed the Benes sufficient condition for rearrangeability of a certain classes of networks. It is shown that the verification of this criterion is exponentially complex. We discuss a routing algorithm for this network, given that it is rearrangeable. This chapter ends with a discussion of a counter example for a theorem due to Afshar on the rearrangeability of feed-back free network.

Chapter V
CONCLUSIONS

This chapter presents a summary of the results presented in this dissertation and suggests some problems for the future work in the area of interconnection networks.

1) The first result relates to the universality of shuffle-exchange networks. A well known open problem on the rearrangeability of a cascade of blocking networks has been studied. Parker[P1] proved that for any N, a cascade of three(three copies connected in series) Omega networks is rearrangeable. Wu and Feng[W4] improved the above result and showed that $3Log_2^N-1$ stages are sufficient for rearrangeability. There are two related results in this part. The first part proves that for $N=2^k$ and $k \geq 4$, a cascade of two copies of an Omega network is not transformable to a well known class of rearrangeable networks, namely Benes class networks [B4]. Consequently, the universality of a cascade of blocking networks must be settled outside of the framework of the Benes network. The second part provides a new upper bound namely that for $N=2^k$, $k \geq 4$, $3k-3$ stages (of 2x2 switches and shuffle link permutations) is rearrangeable. It should be interesting to note that the only currently available method for rearrangeability of a cascade of non-

blocking networks is to recast it, by suitable transformation, in the form of Benes a network. Except for N=4 and N=8, the question whether or not a cascade of two Omega networks is rearrangeable still remains open. Our analysis shows that this question must be settled outside the Benes framework.

2) The second result relates to the problem of computing the combinatorial power of the block-structured networks. To this end a very rich and useful subclasses of Banyan networks called L-stage Banyan networks have been introduced. One of the notable subclass of L-stage SW-Banyan networks is expanding and contracting SW-Banyan networks. Expanding and contracting SW-Banyan networks have less blockage compared to the rectangular Banyan networks[D1]. While it stands to reason to guess that networks with less blockage may realize more permutations, computing the number of permutations realized by an expanding and contracting SW-Banyan networks is by no means trivial. It is shown that the known method for computing this number is incorrect. Chapter three covers the necessary and sufficient condition to have positive combinatorial power. The number of permutations realizeable by a class of expanding and contracting SW-Banyan networks is proportional to the number of different matrices satisfying certain conditions.

3) The third result corresponds to the verification of the universality of a network. It has been shown [B3] that a

cascade of two NxN networks alternately connected between three stages of switches is rearrangeable if it satisfies a set of group theoretic conditions. Verification of this new criterion has exponential complexity. Further, this condition is too sufficient in the sense that it is not often satisfied by many rearrangeable networks. For $N=2^k$ and k=2,3 the above condition does not hold while both networks are rearrangeable. All these results indicate that the resolution of the above open problem namely universality of the cascade of blocking networks must await the development of a new technique for proving the universality of interconnection networks.

4) The final result relates to the corollary in [A1] on the rearrangeability of feed-back free networks. It has been stated that a feed-back free switching network is rearrangeable if and only if it realizes one-to-one and onto mapping of any two input terminals to any output terminals. It is shown by a counter-example that the sufficiency part does not hold for any feed-back free network.

REFERENCES

[A1]. Afshar, S. (1982). " A Block Matrix representation of Permutation Networks," Circuits, Systems and Signal Processing, vol. 1, pp. 251-266.

[A2]. Anderson, G. A. and Jensen, E. D. (1975). " Computer InterConnection Structures: Taxonomy, Characteristics and Examples," ACM Computing Surveys, vol. 7, pp. 1977-213.

[A3]. Apfelbaum, H., Case, R. P., Juliussen, J. E. Shriver, B., Stone, H. and Thrnton, J. E.(1978). " Computer System Organization Problem of the 1980's," Computer, vol. 11, pp. 20-28

[B1]. Benes`, V. (1962). "Algebraic and Topological Properties of Connecting Networks," The Bell System Technical Journal, vol. 41, pp. 1249-1274.

[B2]. Benes`, V. (1962). "On Rearrangeable Three-Stage Connecting Networks," The Bell System Technical Journal, vol. 41, pp. 1481-1492.

[B3]. Benes`, V. (1964). "Permutation Group, Complexes and Rearrangeable Connecting Networks," The Bell System Technical Journal, vol. 43, pp. 1619-1640.

[B4]. Benes`, V. (1975). "Application of Group Theory to Connecting Networks," The Bell System Technical Journal, vol. 54, pp. 409-422.

[B5]. Benes`, V. (1975). "Proving the Rearrangeablity of Connecting Networks," The Bell System Technical Journal, vol. 54, pp. 423-434.

[B6]. Benes`, V. (1965). Mathematical Theory of Connecting Networks and Telephone Traffic, Academic Press, New York.

[B7]. Bhuyan, L. N. and Agrawal D. P. (1983). "Design and Performance of Generalized Interconnection Networks," IEEE Transactions on computers, vol. 32, pp. 1081-1090.

[C1]. Clos, C. (1953). " Study of Non-blocking Switching Networks". Bell Systems Technical Journal, vol.32, pp. 406-424.

[D1]. DeGroot, D. (1983). "Expanding and Contracting SW-Banyan Networks," Proceedings of the International Conference on Parallel Processing, pp. 771-780.

[D2]. Dias, D. M. (1981). " Analysis and Simulation of Buffered Delta Networks," IEEE Transactions on Computers, vol. 30, pp. 273-282.

[D3]. Druskal, C. P. and Snir, M. (1982). " Some Results on Packet-Switching Networks for Multiprocessing," Proceedings of the 1982 Princeton Conference on Information Sciences and Systems, pp. 305-310.

[F1]. Feng, T. (1981). " A survey of Interconnection Networks," Computer, vol. 14, pp. 12-30.

[F2]. Flynn, M. J. (1972). " Some Computer Organization and Their Effectiveness," IEEE Transactions on Computers, vol. pp. 948-960.

[G1]. Goke, L. R. and Lipovski, G. J. (1973). " Banyan networks for Partitioning Multiprocessor Systems," Proceeding of the First Annual Symposium on Computer Architecture, pp. 21-28.

[G2]. Goke, L. R. (1976). " Banyan Networks for Multiprocessor Systems," Ph.D. Dissertation, University of Florida.

[G3]. Golomb, S. W. (1961). " Permutations by Cutting and Shuffling," SIAM Review, vol. 3, pp. 293-297.

[H1]. Hall, P. (1935). " On Representatives of Subsets," Journal of London Mathematics, vol. 10, pp. 26-30.

[H2]. Hall, Jr. M. (1956). "An Algorithm for Distinct Representations," Ame. Math. Monthly, vol. 63, pp. 716-717.

[H3]. Hall, Jr. M. (1959). The Theory of Groups, The McMillan Company, New york.

[H4]. Hall, Jr. P. (1967). Combinatorial Theory, Blaisdell Publishing Company.

[H5]. Hayes, J. P. (1978). Computer Architecture and Organization, McGraw Hill, New York.

[H6]. Hockney,R. W. and Jesshope, C. R. (1981). Parallel Computers, Adam Hilger Ltd, Bristol.

[K1]. Kothari, S. C. and Lakshmivarahan, S. (1983). " A Condition Known to be Sufficient for Rearrangeablity for the Class Benes Networks Made of 2x2 Switches is Also Necessary," Proceeding of the International Conference on Parallel Processing, pp. 76-78.

[K2]. Kothari, S., Lakshmivarahan, S. and Peyravi, H. (1984). "Computing the Number of Permutations Realizeable by a Class of SW-banyan Network," 1984 ACM Regional Conference, Austin, Texas.

[K3]. Kothari, S., Lakshmivarahan, S. and Peyravi, H. (1985). "A New Upper Bound on the Number of Stages Required for the Universality of Shuffle-Exchange Networks," Proceedings of the 19th Annual Conference in Information Sciences and Systems. pp. 515-519.

[K4]. Kruskel, C. P. and Snir,M. (1982)." Some Results on Multi-stage Interconnection Networks for Multiprocessors," Proceeding of the Princeton Conference on Information Sciences and Systems. pp. 305 pp.

[K5]. Kuck, D. J. (1980). The Structure of Computers and Computation. John Wiley.

[L1]. Lang, T. (1976). " Interconnections Between Processors and Memory Modules Using the Shuffle-Exchange Networks," IEEE Transactions on Computers, vol. 25, pp. 496-503.

[L2]. Lang, T. and Stone, H. S. (1976). " A Shuffle-Exchange Network with Simplified Control," IEEE Transactions on Computers,. vol. 25, pp. 55-65.

[L3]. Lawrie, D. H.(1975). " Access and Alignment of Data in an Array Processor," IEEE Transactions on Computers, vol. 24, pp. 1145-1155.

[L4]. Lakshmivarahan,S. and Peyravi, H. (1983). "Bibliography on interconnection Networks," Technical Report, School of Electrical Engineering and Computer Science, University of Oklahoma, Norman, Oklahoma, October.

[L5]. Lakshmivarahan,S. "Interconnection Networks," Class Lecture Notes, 1982-1985.

[L5]. Lenfant, J. (1978). "Parallel Permutation of Data: A Benes Network Control Algorithm for Frequently Used Permutations," IEEE Transactions on Computers, vol. 27, pp. 637-647.

[M1]. Masson, G. M., Gingher, G. C. and Nakamura, S. (1979). " A Sample of Circuit Switching Networks," IEEE Transactions on Computers, June 1979, pp. 32-48.

[O1]. Opferman, D. and Tsao-Wu. N.(1971). "On a Class of Rearrangeable Switching Networks, Part I. Control Algorithm," Bell System Technical Journal, vol. 50, pp. 1579-1600.

[P1]. Parker, Jr. D. S.(1980). " Note on Shuffle-Exchange Switching Network," IEEE Transactions on Computers, vol. 29, pp. 213-222.

[P2]. Patel, J.(1981). " Performance of Processor Memory Interconnections for Multiprocessors," IEEE Transactions on Computers, vol. 30, pp. 771-780.

[P3]. Pease. M. C.(1968). " An Adaptation of the Fast Fourier Transform for Parallel Processing," Journal of ACM, vol. 15, pp. 252-264.

[P4]. Peyravi, H., Kothari, S. C. and Lakshmivarahan, S. (1983). " An Example of an Extended Omega Network which Realizes one-to-one and onto Mappings of any Two Inputs to any Outputs but not Rearrangeable," Technical Report, School of EECS University of Oklahoma.

[P5]. Pippenger, N. (1978). " On Rearrangeable and Non-Blocking Switching Networks," Journal of Computer and System Science, vol. 17, pp. 145-162.

[S1]. Seigel, H. J., McMillen, R. J. and Muller Jr, P. T.(1979). " A Survey of Interconnection Methods for Reconfigurable Parallel Processing Systems". Proceeding of National Proceeding of the National Computer Conference,AFIPS., vol. 48, pp. 387-400, 529-542.

[S2]. Seigel, H. J. (1979)." Interconnection Networks for SIMD Computers," Computer, vol. 12, pp. 57-65.

[S3]. Seigel, H. J. (1979). " A Model of SIMD Machines and a Comparison of Various Interconnection Networks". IEEE Transaction on Computers, vol. 28, pp. 907-917.

[S4]. Seigel, H. J. (1980). " The Theory Underlying the Partitioning of Permutation Networks," IEEE Transaction on Computers, vol. 29, pp. 791-801.

[S5]. Stone, H. S. (1971). " Parallel Processing with Perfect Shuffle," IEEE Transactions on Computers, vol. 2, pp. 153-161.

[T1]. Thurber, K. J. (1974). " Interconnection Network- A Survey and Assessment," Proceedings of AFIPS National Computer Conference, pp. 909-919.

[T2]. Thurber, K. J. and Masson, G.(1979). Distributed Processor Communication Architectures. Heath and Company.

[W1]. Wallich, P. (1983). " End Seen for Shrinking of Integrated Circuit Elements". The Institute News Supplement to IEEE Spectrum, vol. 7 No. 4, April, pp. 1-4.

[W2]. Wu, C. and Feng, T. (1980). " On a Class of Multistage Interconnection Networks," IEEE Transactions on Computers, vol. 29, pp. 694-702.

[W3]. Wu, C. and Feng, T. (1980). " The Reverse Exchange Interconnection Networks," IEEE Transactions on Computers, vol. 29, pp. 801-810.

[W4]. Wu, C. and Feng, T. (1981). " The Universality of the ShuffleExchange Network," IEEE Transactions on Computers, vol. 30, pp. 324-331.

[W5]. Wu, C. (1981). (Editor). " Interconnection Networks- A Special Issue," Computer, vol. 14, pp. 8-76.

[W6]. Wulf, W. and Bell, C. G. (1972). " C.mmp-A Multi Mini Processor," Proceedings of AFIP, vol. 41, pp. 765-778

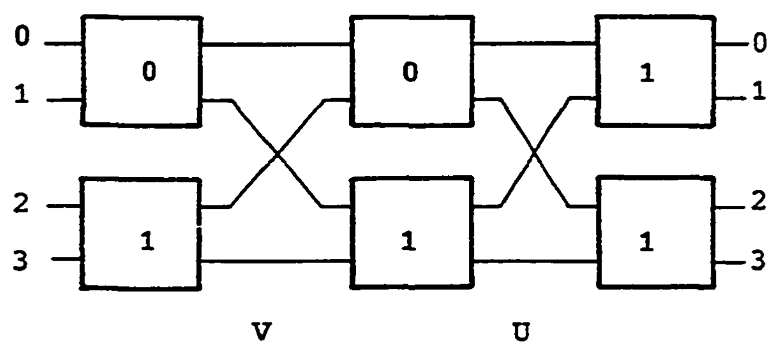
Appendix A

Note: for the simplicity of the calculations, we define a permutation

$$P=\begin{pmatrix} 0 & 1 & 2 & \cdots & N-1 \\ \\ P(0) & P(1) & P(2) & \cdots & P(N-1) \end{pmatrix}$$

as (P(0) P(1) P(2) ... P(N-1)).

Consider the following 4x4 Benes network



U and V are fixed permutation networks, i.e.

P(U)=P(V)={(0 2 1 3)}.

D(U) and D(v) are the set of all switch permutations induced by U and V respectively, i.e. D(U)=D(V)={(01 10) (10 01)}.

The direct product of S_2 with itself is $(S_2)^2$,i.e.

$(S_2)^2$={(01 01), (01 10), (10 01), (10 10)}

the direct product of D(U) and D(V) is

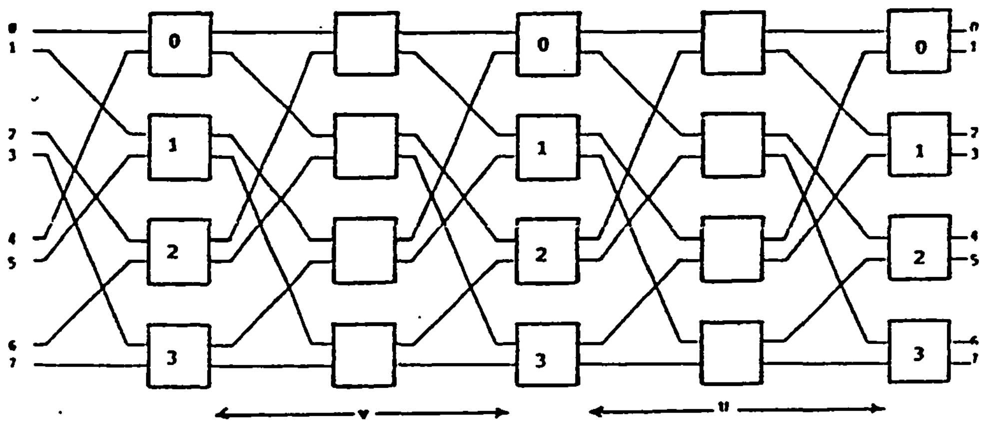
D(U)D(V)={(01 01), (10 10)}

Clearly, $(S_2)^2 \not\subseteq$ D(U)D(V). In otherwords (01 10) and (10 01) are not in D(U)D(V).

But the above network is rearrangeable.

Appendix B

Consider five stages of a shuffle-exchange network excluding the first shuffle, i.e. $(E.\sigma)^4E$.



This network is obtained by cascading U and V alternatly between three stages of four 2x2 switches. Each of the networks U and V are made of one stage of four 2x2 swithes preceded and followed by the shuffle permutation. D(U) and D(v) are the set of all switch permutations induced by U and V, respectively. Each of which has 48 pairs. The direct product of S_4 with itself has $(24)^2$ elements while the direct product of D(U)D(V) has only 312 elements. Clearly, $(S_2)^2 \not\subseteq$ D(U)D(V). In other words, the following are some pairs which are not included in D(U)D(V).

{(0123 3120), (0123 0213), (0123 0312), (0123 2130),
(0123 3021), (0123 1203), (0123 1302), (0123 2031)}.