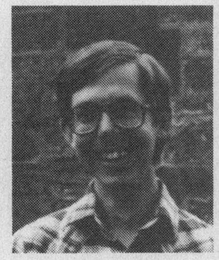
[39] A. E. Casavant, D. D. Gajski, and D. J. Kuck, "Automatic design with dependence graphs," in *Proc. 17th Des. Automat. Conf., IEEE Comput. Soc. DATC ACM SIGDA*, 1980, pp. 506–515.
[40] M. C. Mcfarland, "A formal definition of ISPS for proving properties of hardware descriptions," Des. Res. Cen., Carnegie-Mellon Univ., Tech. Rep., to be published.



Michael C. McFarland was born in Boston, MA, on October 4, 1948. He received the A.B. degree in physics from Cornell University, Ithaca, NY, in 1969, and the M.S. and Ph.D. degrees in electrical engineering from Carnegie-Mellon University, Pittsburgh, PA, in 1978 and 1981, respectively.

He is currently teaching computer science at Boston College, Chestnut Hill, MA, while studying theology at Weston School of Theology, Cambridge, MA. He has worked as a consultant in design automation and verification at Carnegie-Mellon University, the University of Southern California and Bell Labs, Murray Hill, NJ. His research interests include design automation, design verification, formal semantics and hardware description languages.

Alice C. Parker (S'78–M'75) received the M.S.E.E. degree from Stanford University, Stanford, CA, in 1971 and the Ph.D. degree from North Carolina State University, Raleigh, in 1975.

Currently, she is an Assistant Professor in the Department of Electrical Engineering-Systems at the University of Southern California. Prior to that she was an Assistant Professor of Electrical Engineering at Carnegie-Mellon University, Pittsburgh, PA. Her research interests include automatic synthesis and optimization of digital hardware, hardware-descriptive languages, and microprogramming.

Graph Theoretical Analysis and Design of Multistage Interconnection Networks

DHARMA P. AGRAWAL, SENIOR MEMBER, IEEE

***Abstract*—This paper introduces two graph theoretic models that provide a uniform procedure for analyzing 2^n-input/2^n-output Multistage Interconnection Networks (MIN's), implemented with 2-input/2-output Switching Elements (SE's) and satisfying a characteristics called the "buddy property." These models show that all such *n*-stage MIN's are topologically equivalent and hence prove that one MIN can be implemented from integrated circuits designed for another MIN. The proposed techniques also allow identical modeling and comparison of permutation capabilities of *n*-stage MIN's and other link-controlled networks like augmented data manipulator and SW Banyan Network and hence, allows comparison of their permutation. In the case of any conflict in the MIN, an upper bound for the required number of passes has been obtained.**

In a parallel system, a particular permutation may be desirable more frequently than the others. With this question in mind, two simple methodologies for designing MIN's are described which could pass a given permutation without any conflict. It is shown further that if the output stage of one MIN and the input stage of another MIN mutually satisfy the "buddy property," then these two stages could be merged together to result in a network topologically equivalent to the Benes network.

A particular case of special interest is when the second MIN is the reverse of the first MIN. Thus, any MIN can simulate a Benes network in just two passes if data are sent in the forward direction in the first pass and in backward direction in the second pass. This shows that a lowest possible bound of $(2n - 1)$ passes are required to achieve any arbitrary permutation in some specific single-stage networks.

***Index Terms*—Benes network, buddy property, conflict-free permutations, graph modeling, multistage interconnection networks, number of passes, permutability, single-stage network, topological equivalence.**

Manuscript received February 11, 1982; revised September 10, 1982. This work is based upon material partially supported by the National Science Foundation under Grant IST-7918462.

The author was with the Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI 48202. He is now with the Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC 27650.

I. Introduction

Several supersystems described in the literature [1]–[4] are shown to provide enough computational power to solve complex problems on a real-time basis. In all these systems, the computational parallelism is obtained from multiple processors, while the switches provide the reconfiguration capabilities in the system. The interconnection networks (IN's) are capable of allowing simultaneous communication between the memory blocks (MB's) and the processing elements (PE's) and hence, are considered as the heart of the parallel systems [5]. The system performance is greatly affected by the interprocessor communication [5]–[7]. A typical example of an IN is an $(m \times m)$ cross-bar switch employed in the existing C.mmp computer [8] and in proposed Burroughs scientific processor [3].

0018-9340/83/0700-0637$01.00 © 1983 IEEE

Such m^2 switches provide programmable data paths between the PE's and the MB's.

In a supercomputer environment, thousands of PE's are to be used, thereby forcing m to be considerably large, which in turn makes the use of cross-bar switches uneconomical. An alternative is to divide IN into several stages; and such a segmented network with each stage satisfying partial connection requirements yields to a Multistage IN (MIN). Several existing and proposed system designs [9]-[10] are based on the MIN's. The MIN's allow any input to be connected to any one of the outputs and possess very simple control algorithms. Hence, they are specially useful for a supercomputer with a large number of PE's and MB's. Various MIN's described in the literature utilize 2-input/2-output switching elements (SE) as a building block. Several MIN's have recently been compared by Thompson [11], Siegel [12], and Wu and Feng [13], have used topology describing rules to show the topological equivalence between baseline, modified data manipulator, flip, omega, indirect binary n-cube and regular SW Banyan ($S = F = 2$) MIN's, all implemented with 2×2 SE's. Abidi and Agrawal [14], [15] used this procedure in partitioning a permutation into two-single pass conflict-free permutations.

The existing analytical procedures for MIN are extremely complex; they are suitable for only very few MIN's with 2×2 SE's. This paper utilizes a simple graph theoretic approach for analyzing and designing the MIN's. Graph theory has been observed to be a very useful analytical tool for generalized connectors, concentrators, and connector networks [16]. Masson used the connectivity concept to obtain a graph model of cross-bar switches and developed a class of networks called binominal switching networks. Such a graph model has been illustrated in Fig. 1. In another novel work, Goke and Lipovski [17] introduced a Banyan network which utilizes a group of basic Banyan modules. The Banyans are described by asymmetric directed graphs and are basically link-controlled networks. Here we propose two different graph models of 2×2 SE shown in Fig. 2 and then extend them in obtaining graph models of any MIN. These graph models are shown to be very helpful in comparing the permutation capabilities of various MIN's. They provide an easy way of resolving the conflicts and for designing a suitable MIN.

In the first part of this paper, we are concerned with a class of N-input/N-output ($N \equiv 2^n$) MIN's, uniformly structured in n stages such that each pair of SE's from the ith stage is connected with only one pair of SE's belonging to the $(i + 1)$th stage. This is known as a "buddy" property [18] with the pair of SE's in the ith stage called "output buddies" (j_1 and j_2) of Fig. 3(a). The pair at $(i + 1)$th stage is named as "input buddies" (j_3 and j_4) of Fig. 3(a). Two input-buddy pairs also constitute two pairs of output buddies. This is illustrated in Fig. 3(b) where input buddy pairs ($j_1 - j_2$) and ($j_3 - j_4$) also form output buddies ($j_1 - j_3$) and ($j_2 - j_4$).

The strict buddy property forces the network to be modular in nature and hence facilitates expandability which makes it suitable for VLSI implementation. This, in turn, makes a network extremely useful for a supersystem. It is interesting to note that a large class of MIN's fall into the buddy type category and we do not consider other types of networks like

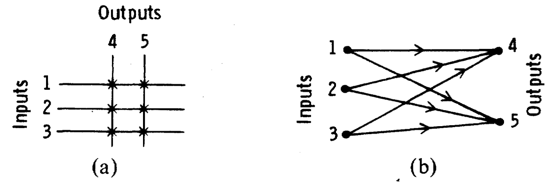


Fig. 1. (a) 3×2 cross-bar switches. (b) Directed graph model of Fig. 1(a).

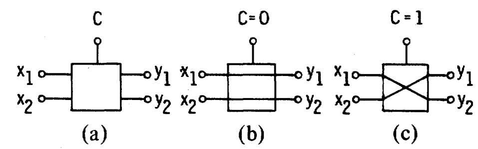


Fig. 2. (a) Switching element. (b) Parallel connection with $C = 0$. (c) Cross connection with $C = 1$.

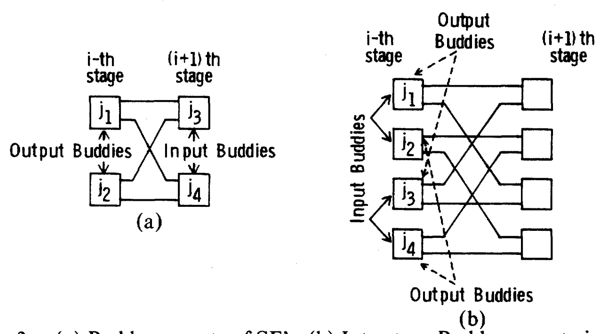


Fig. 3. (a) Buddy property of SE's. (b) Interstage Buddy property in a uniform MIN.

nonrestricted Delta networks [18]-[19] and nonuniform Banyan networks [17]. The nonbuddy type networks have been separately covered in another paper [20]. In the later part of this paper, we are concerned with the design of Benes-like [21] network from one or more buddy-type MIN's, so that any permutation could be provided in a conflict-free manner. Then, the lowest possible number of passes for achieving any arbitrary permutation in some specific single-stage network has been derived.

The two graph models of SE's are presented in Section II, and Section III provides a detailed treatment of the first model in describing topological equivalence of all MIN's with a buddy property. The usefulness of the second model is demonstrated in Section IV where the permutation capabilities of MIN's are compared with the link-based networks like the augmented data manipulator [22], [23] and the regular SW-Banyan Network with $S = F = 2$ [17]. The next section provides a technique to determine the number of passes required to achieve a desired permutation in a given network configuration. Two generalized design methodologies for designing a MIN, such that a given permutation could be achieved without any conflict while still maintaining full-connectivity, is covered in Section VI. Section VII introduces a technique to connect two MIN's so that the permutability of Benes-type network could be attained. Single stage networks with specific interconnections have been investigated to determine the number of passes required in achieving any permutation, and a lower bound has also been obtained. Concluding remarks are included in Section VIII.

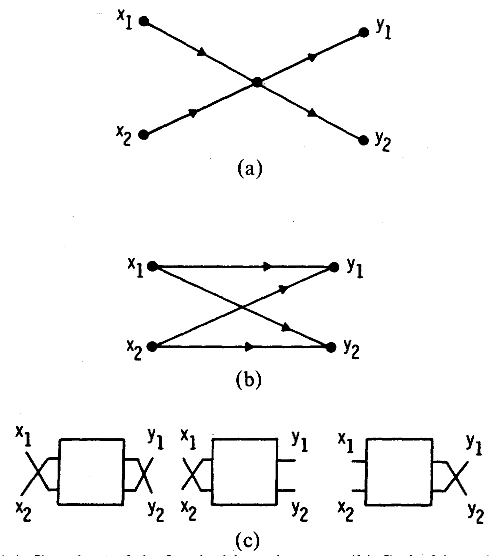


Fig. 4. (a) Graph model of switching element. (b) Switching element and its alternative graph model. (c) Alternate positioning of input/output links.

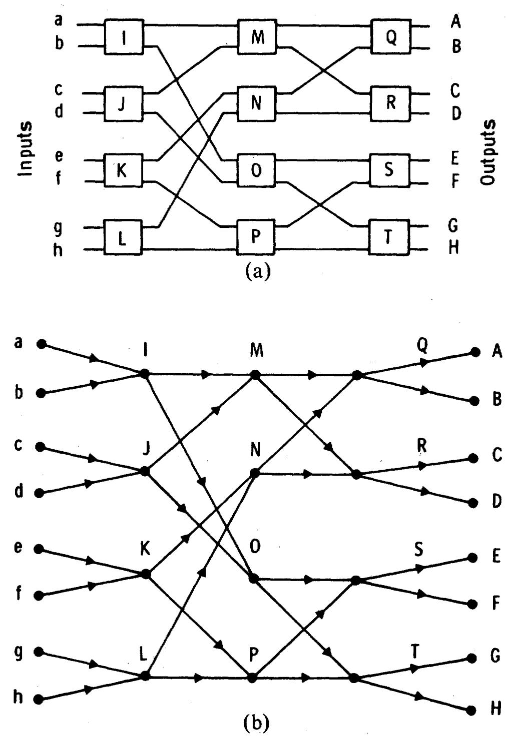


Fig. 5. (a) Baseline network. (b) Graph model of baseline network.

II. Graph Models of 2 × 2 Switching Element (SE)

First Model

A graph model is obtained by representing an SE by a node and directed edges, drawn from the inputs as the source nodes to the outputs as the sink nodes. The resulting graph of an SE without any control line is shown in Fig. 4(a); and it is similar to the physical layout of the links with each SE replaced by a node. It may also be noted that interchanging the physical positions of x_1 and x_2 and/or y_1 and y_2 (Fig. 4(c)) changes the topology describing rule [13] while the graph model of SE remains unchanged.

Second Model

An alternative approach is to assign nodes to the links rather than to the SE (Fig. 4(b)). As the control signal of the SE allows the connection of inputs $x_1 - x_2$ to $y_1 - y_2$ or $y_2 y_1$ respectively, this means that in this model, a node must be reached by only one node and along only one of the two edges when a specific value of the control signal is used. Thus, this graph model is based on the connectivity and the approach is similar to the graph model of cross-bar switches [16].

It is worth mentioning that the graph models of Fig. 4(a) and 4(b) can as well represent the upper (lower) broadcast (i.e., the input $x_1(x_2)$ is connected and transmitted to both the outputs y_1 and y_2) type of SE introduced by Lawrie for his Omega network [24].

III. Network Modeling and Their Equivalence

Initial fabrication cost of a LSI and VLSI circuit is fairly high and the production cost is relatively low. This is the main reason for avoiding custom-made chips, and it is usually advisable to concentrate on the use of off-the-shelf components. Thus, if the IC's for only one type of MIN is commercially available, then its use and versatility could be enhanced if it could be employed in implementing any desired MIN. The topological equivalence of only a few MIN's have been established in the literature [13], [23]. Here, we prove such equivalence of all possible buddy-type MIN's. This also shows that it is possible to use the IC's designed for one type of buddy MIN to obtain another type of buddy MIN suitable for a given application (see Section VI). Furthermore, it provides the equivalent mapping of various SE's so that the control algorithm for any MIN could be correctly and conveniently applied. The theorem on equivalence is also useful in proving the validity of the results obtained in Section VII.

The graph model of the SE shown in Fig. 4(a) could be easily used to obtain a graph model of a MIN, if the following are assumed.

a) A node is assigned to each SE.

b) Only the primary inputs and the final outputs are assigned to a node.

c) Nodes are not assigned to all the intermediate inputs and outputs as they do not possess any characteristics of the network topology.

d) Nodes representing SE's of two stages are connected by directed edges.

This procedure leads to a graph model of a network and such a model for 8-input and 8-output Baseline MIN [13] of Fig. 5(a) is shown in Fig. 5(b).

The strictly buddy type MI networks described in the literature look different because different shuffling strategies have been utilized for interconnecting various stages of the network. But, mathematical relations have been derived [13], [20], [23] to show that some of the strictly buddy type networks are topologically equivalent. With our graph model any buddy type MIN could be transformed into a directed graph and then the topological equivalence is mathematically translated into graph isomorphism [20].

Theorem 1: All of the 2^n-input 2^n-output nonredundant strictly buddy-type MIN's are topologically equivalent to each other provided they satisfy the following conditions.

a) Each network utilizes 2-input/2-output SE as a basic module.

b) Each has n stages.

c) There are 2^{n-1} SE's per stage.

d) For all stages of the network a strict buddy property is observed in the case of both the input and the outputs.

e) SE's are connected in such a way that any input link can be connected to any one of the output links.

Proof: The theorem is proved using a simplified representation of the graph model as shown in Fig. 6(a). The details inside the rectangular box are not shown to prove the theorem for any general value of n. It is implied that the structure of the graph provides a connecting path from any input node to any one of the output nodes. To prove the theorem, assume that it is true for Fig. 6(a) and we will show its correctness for (2^{n+1})-input MIN. Such a MIN can be obtained by having two MIN's of size 2^n and then adding one more stage either at input side or at the output side.

One such possible configuration with angmentation of a stage added at the output side is shown in Fig. 6(b). Each pair of nodes satisfying input buddy property are also marked at the nth stage of Fig. 6(b). To provide full connectivity, nodes at the nth stage of two 2^n networks are to be connected together at the $(n + 1)$th stage. Let us assume that α and γ are connected to a. Then the buddy property requires α and γ to be connected to another node; say c. Then the output buddy restrictions necessitate that while constructing the $(n + 1)$th stage, the respective input buddies of α and γ (which are β and δ) ought to be connected to two nodes of the $(n + 1)$th stage (say b and d) and there is only one unique way of doing this. This process could be continued for all nodes of nth stage, thus providing only one type of graph. As the nodes (α, β) and (γ, δ) are arbitrarily selected, it is possible to obtain other graphs which will be isomorphic to Fig. 6(b). Such a buddy type MIN is also referred to as nonredundant MIN as there is a unique path for each input-output pair. Q.E.D.

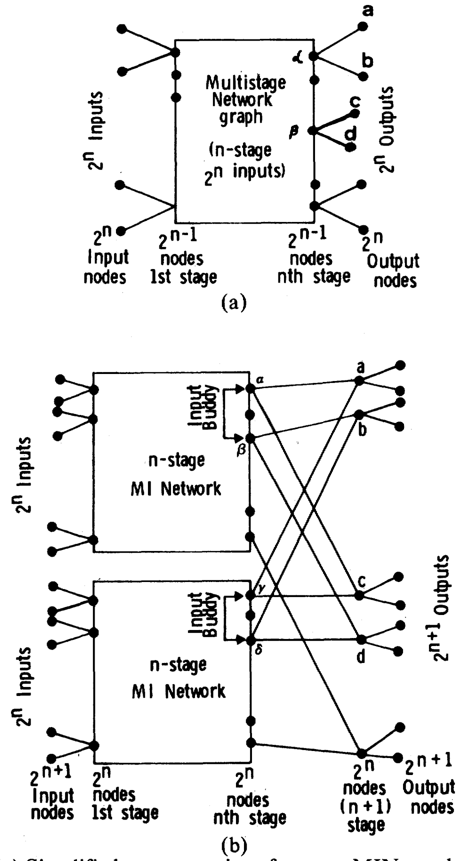


Fig. 6. (a) Simplified representation of n-stage MIN graph model. (b) Construction of (2^{n+1}) input MIN graph from two networks of size 2^n.

IV. Alternate Graph Theoretic Approach and its Usefulness

Various types of MIN's have been proposed and advocated for their use in multiprocessor systems. As a result there is an atmosphere of confusion as to which network is superior to the others. The permutation capability of a network is extremely important for efficient operation of a supersystem. A MIN which could provide a larger number of connections between its inputs and outputs is preferred provided the complexity of the control is maintained at the same level. There are networks like data manipulator [22] or augmented data manipulator (ADM) [23] and Banyan Network [17] which do not utilize 2×2 SE's as their basic modules and each link is controlled individually. It should be pointed out that Wu and Feng [13] have modified the data manipulator and the Banyan networks so that it is feasible to implement them using 2×2 SE's and they have shown topological equivalence of these modified versions. But, there does not exist any satisfactory technique which compares and quantifies permutability of ADM, Banyan and other MIN's. This section is devoted to an alternative graph theoretic technique that provides a unique model of all MIN's and hence, enables us to show the relationships between various networks.

In order to obtain similar graph models of two types of MIN's, the graph model of Fig. 4(b) is utilized resulting in another graph of baseline network, and is illustrated in Fig. 7. In ADM, each cell is controlled individually, and the data output from a cell x at stage i becomes the data input to cell x' at stage $i - 1 \cdot x'$ is defined by either of the following three possibilities.

a) Straight: $x' \equiv x$.

b) Upper broadcast: $x' \equiv x + 2^i \pmod N$.

c) Lower broadcast: $x' \equiv x - 2^i \pmod N$.

Such an ADM with individual cell control is shown in Fig. 8(a). The second graph model of Fig. 4(b) also represents the link connectivity and each cell of ADM could be considered as an input/output node of a SE. This allows the direct use of the graph model, shown in Fig. 4(b) and makes the proposed technique particularly attractive in obtaining a graph model of the ADM. This graph model shown in Fig. 8(b) contains some redundant edges to provide alternate paths in the network. For example, the input node "a" can be connected to the output node "B" using three different paths of "a-0-0-b," "a-0-2-B," and "a-4-2-B." In this way, we will have at least two additional paths. The graph of another link controlled network (Banyan) is shown in Fig. 8(c).

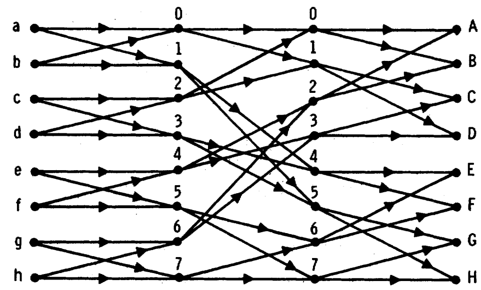


Fig. 7. An alternate graph model of the baseline MIN.

Thus, the second model is useful in comparing the permutation capabilities of a MIN like baseline network with the ADM or the Banyan. As is clear from Figs. 7, 8(b) and 8(c), these models contain the same number of nodes. Now, the relationship among the baseline, ADM, and Banyan networks could be easily examined. A subgraph of the ADM, drawn as a baseline network, is shown in Fig. 9 and this shows that the ADM is topologically equivalent to the baseline. However, the ADM still contains some unused links shown with the broken lines in Fig. 9. Thus, it is more appropriate to quantify that the baseline network is completely embedded within the ADM and the left-over redundant edges of the ADM show its versatility. This means that the ADM has more permutation capabilities compared to any other MIN as defined by the Theorem 1, thereby providing an alternate proof of an earlier reported work on ADM [23]. In terms of graph theory, the baseline, and all other buddy type MIN's defined by Theorem 1, could be viewed as subgraphs of the ADM, i.e., the ADM embeds the baseline and other MIN's. It is worth emphasizing that it is difficult to prove this result using the topology describing technique introduced in [13].

The graph model of SW Banyan [17] with $S = F = 2$, shown in Fig. 8(c) provides a very interesting result. Fig. 8(c) also shows the mapping of baseline network of Fig. 7 and thereby shows that SW Banyan with $S = F = 2$ and baseline networks are isomorphic from graph theory viewpoint and hence, are said to be topologically equivalent. Other regular SW Banyan network with larger value of spread and fan out will embed the Banyan with $S = F = 2$ and hence will have more permutation capability as compared to the Baseline or other buddy type MIN's. It may be noted that the link-controlled networks employ at least two times the number of SE's in a MIN and hence, larger number of control signals have to be supplied for the link-controlled network. In this way, the graph theoretic approach provides a unified way of comparing the permutation capabilities of all possible MIN's.

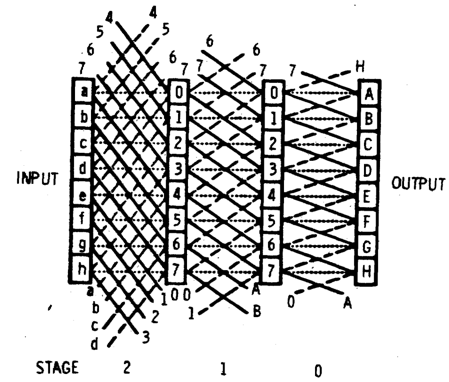


Fig. 8. (a) Augmented data manipulator. (b) Graph model for the augmented data manipulator (ADM) of Fig. 8(a). (c) SW Banyan with $S = F = 2$.

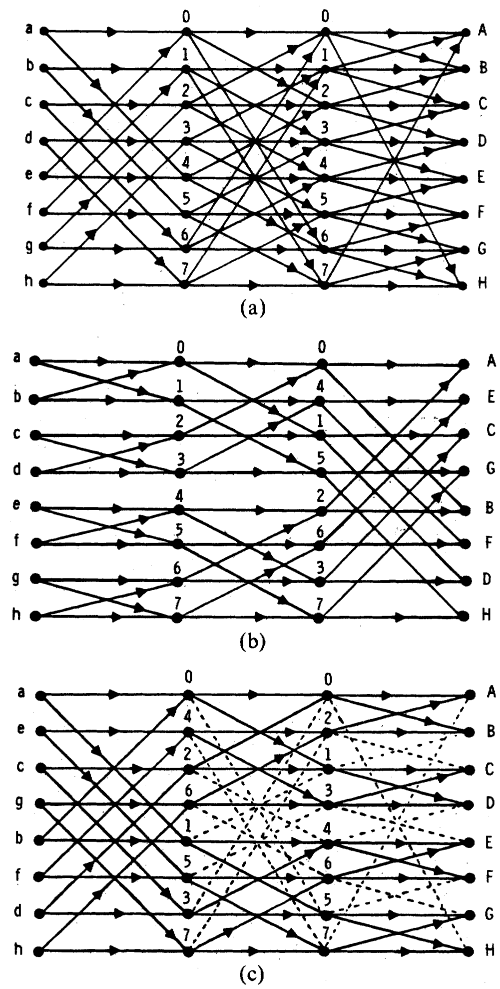


Fig. 9. Graph model of ADM with embedding of the baseline MIN (unused edges are shown with broken lines).

V. Number of Network Passes for Conflict-Free Permutations

As mentioned earlier, connectivity of a MIN is critical with respect to the overall performance of a large parallel system. The degree of parallelism that could be utilized by the PE's is determined by the parallel accessibility of the desired data. The connectivity of a network is characterized by the number of conflict-free simultaneous connections possible in a network. For a 2^n-input/2^n-output MIN, there are $2^n!$ possible input/output permutations. It has recently been shown [14] that certain types of bijections are possible without any conflict and such permutation sets are dependent on the network configuration. But, for a given network structure, not all the permutations are conflict-free [14]–[15]. In fact, it is demonstrated later in this section that in the worst case, one pass may provide connections for only $2^{\lceil n/2 \rceil}$ input–output pairs in a conflict-free manner. In a given network, it is important to investigate whether inputs-to-outputs mapping or a bijection could be achieved without any conflict or not. In case of conflict in parallel systems, it is useful to determine the minimum number of passes required to achieve such a bijection. This is performed by dividing the input–output bijections into several groups in such a way that each group of bijection could be performed in one pass. Thus, the number of passes a MIN requires for any arbitrary permutation can be considered as a reasonable measure of its effectiveness for that particular bijection.

At this point, the concept of "utilization factor" (UF) is introduced. In a given graph, if an edge is used for establishment of the necessary link between a source and a destination nodes, then the link is utilized once or the UF of that edge is assumed as 1. In this way, the UF of any link is given by a numerical value equal to the number of times that link is used.

The concept of UF is employed to compute and prove a lower bound for the minimum number of passes required to achieve a given permutation. To do this, any one of the two graph models could be utilized. In the graph model of Fig. 4(b), only two edges of the graph model are used for each state of a SE, and the other two edges are utilized only in the complementary operating mode. The graph model of Fig. 4(a) utilizes all the edges irrespective of the value of the control signal. These arguments lead to the following corollaries and are given without any proof.

Corollary 1: If a MIN is modeled using the first method, there is no conflict iff UF of any edge is not more than 1.

Corollary 2: If second model is used, then there is no conflict iff each node is reached by only one edge, i.e., the additive UF of all edges directed to a node, is not more than 1.

The corollaries 1 and 2 can be applied to evaluate the permutation capability of a MIN with respect to a desired permutation. Using models 1 or 2, the graphs for a MIN could be drawn and then the UF for a given bijection can be obtained.

Theorem 2: In a MIN, the lower bound for the number of required passes is given by k, where k is given by the largest UF (in model 1) or the cumulative largest UF (in model 2) to provide a desired permutation.

Proof: For any input–output pair conection, there is one–to–one correspondence, i.e., a unique path is required for establishing the data path. This leads to utilization of the corresponding edges in the graph model of the network. If the UF of any edge (or pair of edges in model 2) is more than one, then a conflict exists and the only way it can be resolved is to allow several passes and to use the conflicting edge only once in each pass. This requires at least as many passes as the maximum number of the UF among all the edges (or pair of directed edges in model 2). Q.E.D.

The concept of UF in the first model can be said to be similar to the entries of the conflict-resolution table [13]. But, UF is helpful in arriving at Corollary 3 and Theorem 3.

Corollary 3: If the conflict is present in only one stage, then the number of required passes is given by the k-value of that stage.

Proof: This is obvious from an example shown in Fig. 10. The conflict is present in link stage 2 (1) of Fig. 10(a), Fig. 10(b), and k-value is 2. Hence, only two passes are required to achieve the permutation. The first pass will provide

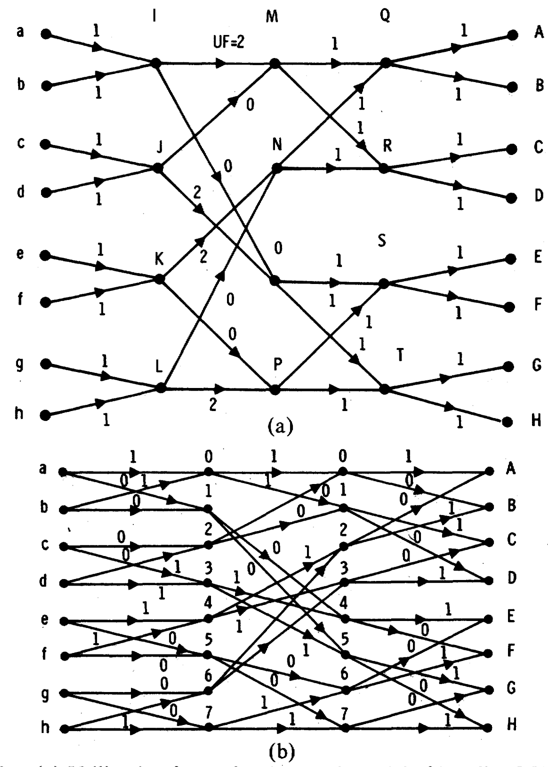


Fig. 10. (a) Utilization factor for the graph model of baseline MIN: Permutation to be achieved $\begin{pmatrix} a\,b\,c\,d\,e\,f\,g\,h \\ ACEGBDFH \end{pmatrix}$. (b) Utilization factor of the alternate graph model for the baseline MIN: Permutation to be achieved $\begin{pmatrix} a\,b\,c\,d\,e\,f\,g\,h \\ ACEGBDFH \end{pmatrix}$

$\begin{pmatrix} a & c & e & g \\ A & E & B & F \end{pmatrix}$ permutation while the rest is provided in the second pass. It is worth mentioning that certain specific bijections can be passed in 2-passes and they have been identified by Abidi *et al.* [15] and there is a need to extend this procedure for any given permutation.

Theorem 3: In a n-stage MIN, the upper bound for the number of required passes to achieve any arbitrary permutation is given by $2^{\lfloor n/2 \rfloor}$ while the minimum number of inputs-to-outputs mapping in each pass equals $2^{\lceil n/2 \rceil}$.

Proof: Before proving the theorem, let us look at an example shown in Fig. 11. The required permutation is such that the maximum value of the UF is 4 and only four edges are used in the center stage. In fact, this is the worst case and requires four passes to achieve the permutation. The path from inputs "*abcd*" to outputs "MNOP" are shown by dark lines. It is clear from this, that the paths from input nodes "*abcd*" merge together at the node "y," use a common edge "yz" and from node "z," they again start emanating outwards to "*MNOP*." Such worst case paths form two back to back connected binary trees and Fig. 12 illustrates this for various values of n. Only a part of the graph is shown for clarity of the diagram and conciseness of the text. It is clear from them that a fully balanced tree structure is obtained from the input nodes until a central link having maximum UF is reached. This is also true for the output side and hence the maximum UF of the central link can be given by $2^{\lfloor n/2 \rfloor}$. Hence, these many passes are required to achieve the worst case permutations. Also, the number of in-

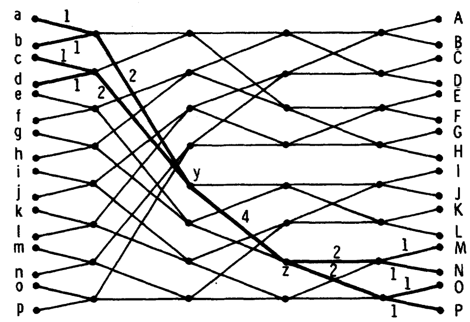


Fig. 11. Utilization factor for 16 × 16 Baseline MIN for the permutation

$$\begin{pmatrix} a\,b\,c\,d\,e\,f\,g\,h\,i\,j\,k\,l\,m\,n\,o\,p \\ MNOPABCDIJKLEFGH \end{pmatrix}.$$

Minimum number of required passes = 4.

puts-to-outputs mapping in each pass can be obtained if total number of inputs is divided by the total number of passes and can be obtained as $2^{\lceil n/2 \rceil}$. Q.E.D.

VI. Generalized Design Methodology

The family of permutations that could be passed in a conflict-free manner varies from one MIN to another MIN [14]. If it is known *a priori* that, in a supersystem, a particular permutation is utilized more frequently than others, it will be useful if the selected MIN configuration provides such a permutation without any conflict. The other bijections may or may not be achievable in a single pass. This helps in establishing a simultaneous parallel connection for a frequently used permutation and hence enhances the effectiveness of the parallel system. Our objective in this section is to define a unified design methodology so that a MIN can be designed that will allow a given bijection in a conflict-free manner while maintaining the full connectivity of the network. The full connectivity requires that the outgoing edges in the graph model are always directed to new nodes so that a link between any input to any one of the outputs, could be established. As all MIN's utilize some sort of shuffle and exchange, the problem could be envisioned as performing a proper shuffle in each stage that will resolve any conflict for a given input–output permutation.

As the network is to be designed to pass certain permutation and the first model provides a direct location of each SE, it is easier to consider the first graph model. The following exhibits the usefulness and universality of the proposed graph theoretic approach. Two different procedures of designing MIN's are presented using the first graph model. Suppose the following 8-input 8-output permutation is to be passed conflict-free.

$$\begin{pmatrix} \text{Input:} & a\ b\ c\ d\ e\ f\ g\ h \\ \text{Output:} & E\ A\ H\ C\ F\ B\ D\ G \end{pmatrix}.$$

Methodology 1: Reordering the permutation according to the physical position of the output link leads to

$$\begin{pmatrix} \text{Input:} & b\ f\ d\ g\ a\ e\ h\ c \\ \text{Output:} & A\ B\ C\ D\ E\ F\ G\ H \end{pmatrix}.$$

Once all the nodes of the graph are marked, the only question is how to connect them properly. To do this, the permutation is divided into two groups. Then each input node of first group and the corresponding node of the second group, are connected

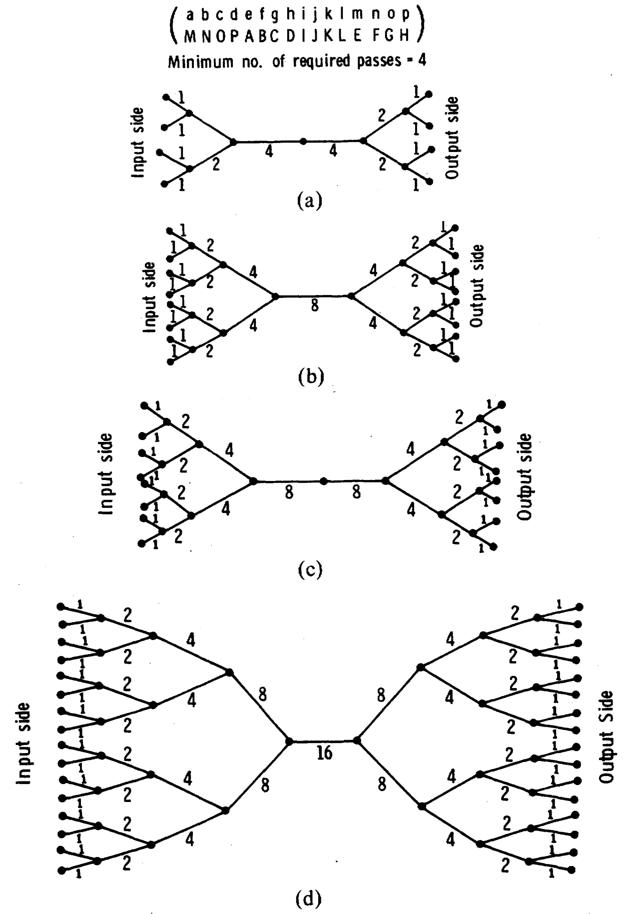


Fig. 12. Parts of the graph illustrating worst UF for various values of n. (a) Part of the graph illustrating a worst case UF in MIN with $n = 5$. (b) Worst UF path for $n = 6$. (c) Worst UF path for $n = 7$. (d) Worst UF path for $n = 8$.

to a node of the next stage and this node is marked as a combination of two previous nodes [see Fig. 13(a)]. This is repeated for all the nodes of the two groups and have been shown in Fig. 13(a). Thereafter, as illustrated in Fig. 13(b), each half is further divided into two parts and the process of the first step is continued until only one node is left in a group. This leads to a graph as shown in Fig. 14(a) and the corresponding MIN is given in Fig. 14(b). It could be easily verified that the SE's of the resulting MIN satisfy the strict-buddy property at each state of the network.

Theorem 4: The network obtained from aforementioned procedure maintains the connectivity property while the desired permutation is passed in a conflict-free manner.

Proof: In the first step, the permutation is rearranged according to the physical locations of the output nodes and then grouped such that the two input nodes are connected to the node of the second level. The corresponding output node numbers differ by $N/2$ ($N \equiv 2^n$). The next step merges two nodes whose destination nodes differ by $N/4$ and so on. This completely avoids any conflict and as the traversal from any input node to output node satisfies the strict buddy property, binary trees with roots at all the inputs (outputs) are formed. Thus full connectivity is ensured. Q.E.D.

Methodology 2: In this procedure, the graph is drawn starting from the output node backwards to the input node. The given permutation is divided into two groups and one node from each group is selected to connect to a node of the previous

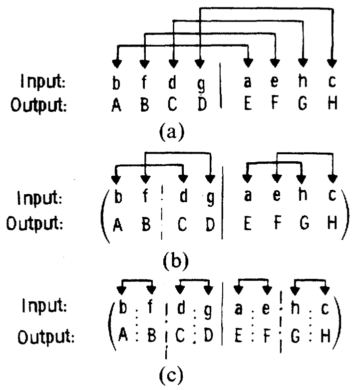


Fig. 13. Node grouping in Methodology 1. (a) Node grouping: first step. (b) Second step of node grouping. (c) Third step of node grouping.

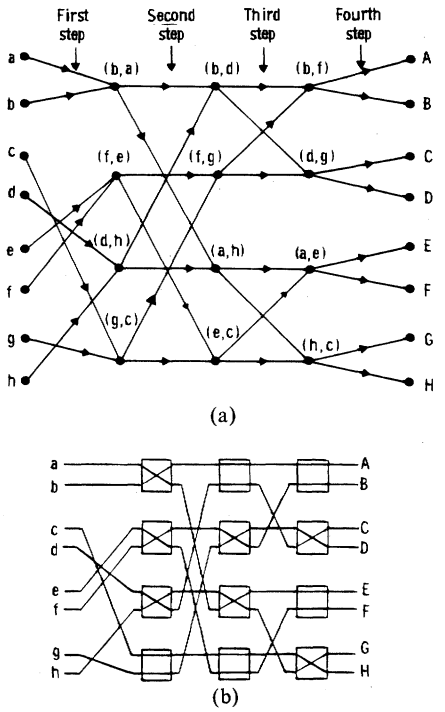


Fig. 14. (a) Methodology I graph model to implement a conflict-free permutation of $\begin{pmatrix}\text{Input} & \text{a b c d e f g h}\\ \text{Output} & \text{EAHCFBDG}\end{pmatrix}$

(b) MIN for Fig. 14(a).

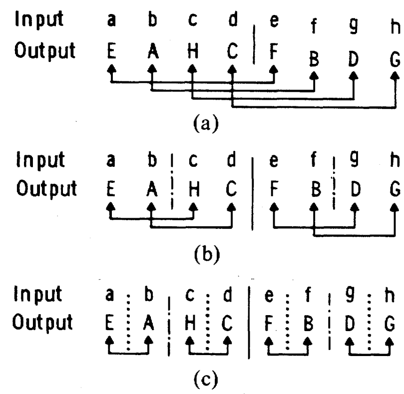


Fig. 15. Node grouping in Methodology 2. (a) Node grouping: first step. (b) Second step of node grouping. (c) Third step of node grouping.

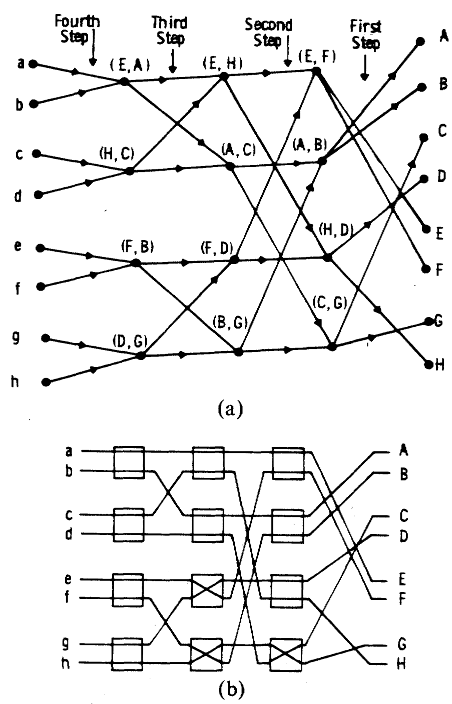


Fig. 16. (a) Methodology to 2 graph model to implement a conflict-free permutation of $\begin{pmatrix}\text{Input} & \text{a b c d e f g h}\\ \text{Output} & \text{EAHCFBDG}\end{pmatrix}$

(b) MIN of Fig. 16(a).

row. This process is continued and is illustrated in Figs. 15 and 16.

Theorem 5: The buddy type MIN obtained in Fig. 16(b) passes the given permutation without any conflict and full connectivity is also provided by the network.

Proof: The validity of this procedure can be proved following the arguments similar to Theorem 4. Q.E.D.

It is worth mentioning that the methodology 1 does not require shuffling of links at the output stage (similar to the Omega network). No shuffling is needed at the input stage when Methodology 2 is used.

Corollary 4: In the design procedure discussed above, if the correspondence between any two groups are not maintained, then the MIN obtained may be nonbuddy type [20]. But, it will pass the given permutation in a conflict-free manner.

Proof: If the order of the nodes is not maintained then the buddy property can not be assured. But, the network will still have all the input trees imbedded into it and hence will provide the permutation in a conflict-free manner. Q.E.D.

It may be noted that it would be extremely useful if a network could be implemented to allow a family of permutations and not just only one particular bijection. Much more effort is needed to identify a methodology and the proposed graph-theoretic methodology appears to be a good approach in obtaining a solution to this general problem.

VII. Benes Type and Single-Stage Networks

Two methodologies for designing MIN's to pass a given permutation without any conflict, have been described in the previous section. But, when a MIN is used for a system with a large number of processing elements and has to be utilized for several different applications, it might be impossible to ascertain the use of one particular permutation more frequently than the others. Thus, in general, different types of permutations may be desirable and the network may not be able to satisfy all required bijections. In such cases of conflicts, a simple solution is to use multiple independent passes of Section V. This is time consuming and might be prohibitive for real time applications if large number of passes are required. Another solution is to increase the number of stages to $(2n - 1)$ and have a Benes network [21] which possesses a unique property of passing any input–output bijection. A combination of baseline-reverse baseline [13] and Omega-reverse Omega [25] have been shown to be equivalent to the Benes network. Here, it is shown later that a network equivalent to Benes, can be implemented by an appropriate combination of two buddy type n-stage MIN's. This novel result is particularly attractive for a system with n-stage MIN, as one pass in the forward and another in the reverse direction can simulate a Benes network and hence can provide any arbitrary permutation only in 2-passes. Furthermore, control algorithm for the Benes network is extremely complex while the MIN's are bit controlled and the destination address itself specifies [24] the value of the control signal. Of course, it is well known that it is difficult to determine the intermediate level mapping and we are working on this problem to make the best use of MIN's bit-controlled properties.

A Benes network for $N = 8$, is shown in Fig. 17(a) and its graph using model of Fig. 4(a), is obtained in Fig. 17(b). It might be noted that the strict buddy properties are satisfied by the Benes network. The n-stage baseline [13] MIN of Fig. 5(a) also utilizes a recursive scheme similar to the Benes network. It has a unique characteristic that the baseline and the reverse baseline (i.e., the baseline network with input-output sides interchanged) are not only topologically equivalent, but are also functinally equivalent [12], [26]. This has been illustrated in Fig. 18(a), which shows the mapping of the reverse baseline MIN as a baseline of Fig. 5(a). Thus, the two networks could be interchanged without altering the functionality of the network.

As mentioned earlier, two stages of the baseline have been mathematically shown to simulate the Benes network [26]. This result could be easily proved by our graph model. Connecting the graph models of Fig. 5(b) and 18(b), it results in the graph of Fig. 19(a). It is worth noting that in the middle stage, the node pairs $A - B$ and $a' - b'$ are connected to one node in each part of the graph and elimination or overlapping of this section will not change the connectivity property of the graph and hence the permutation capabilities of the network. This leads to the graph of Fig. 19(b) which is exactly identical to the graph model of the Benes network of Fig. 17(b).

Theorem 6: A direct combination of n or $(n - 1)$ stages of the baseline and $(n - 1)$ or n stages of the reverse baseline MIN's (hereafter called composite baseline network) can pass any of the $2^n!$ possible permutations in a conflict-free manner.

Proof: As discussed earlier, the last stage of the baseline can be merged with the first stage of the reverse baseline and this results in a graph identical to the Benes network and hence the resultant network possesses the same permutation capabilities as that of the Benes network. Q.E.D.

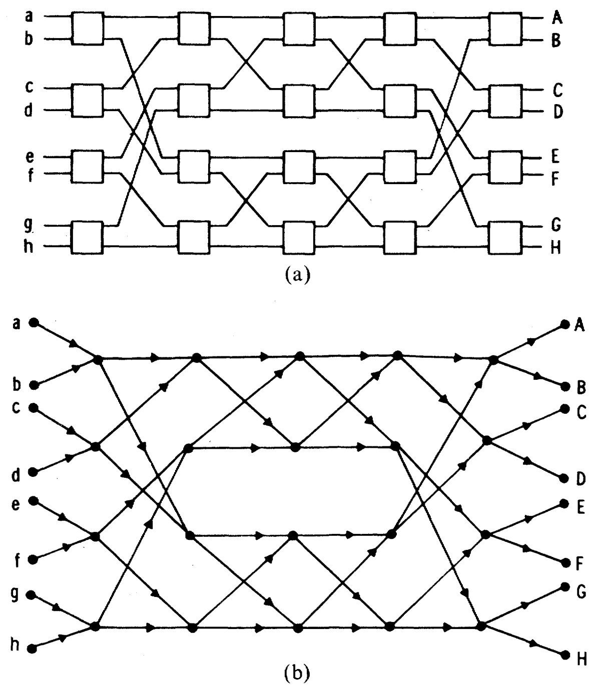


Fig. 17. (a) 8 input—8 output Benes network. (b) Graph model of the Benes network.

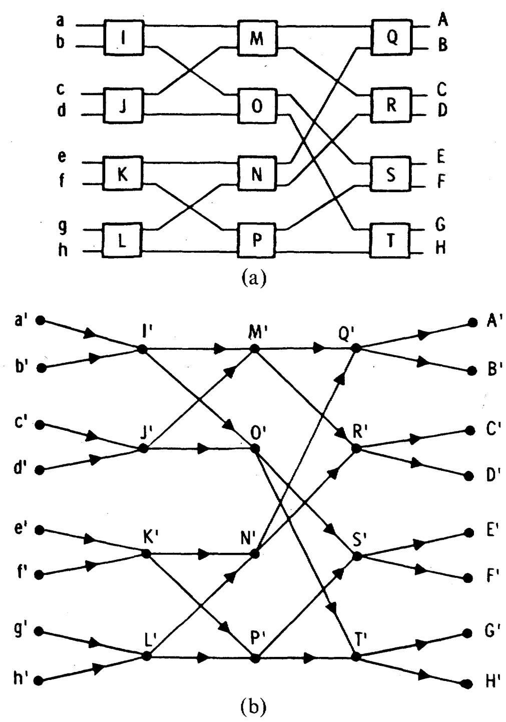


Fig. 18. (a) reverse baseline MIN mapped as a baseline MIN. (b) Graph model of reverse baseline MIN.

Corollary 5: The baseline or the reverse baseline MIN's can be used in either or in both the parts of Fig. 19.

Proof: As shown earlier, the baseline and the reverse baseline MIN's are topologically as well as functionally equivalent to each other. Hence, they can be substituted for each other. Q.E.D.

Theorem 7: Theorem 6 is valid if the baseline and the reverse baseline MIN's are replaced by any other two MIN's, provided that

a) all the conditions of Theorem 1 including the buddy

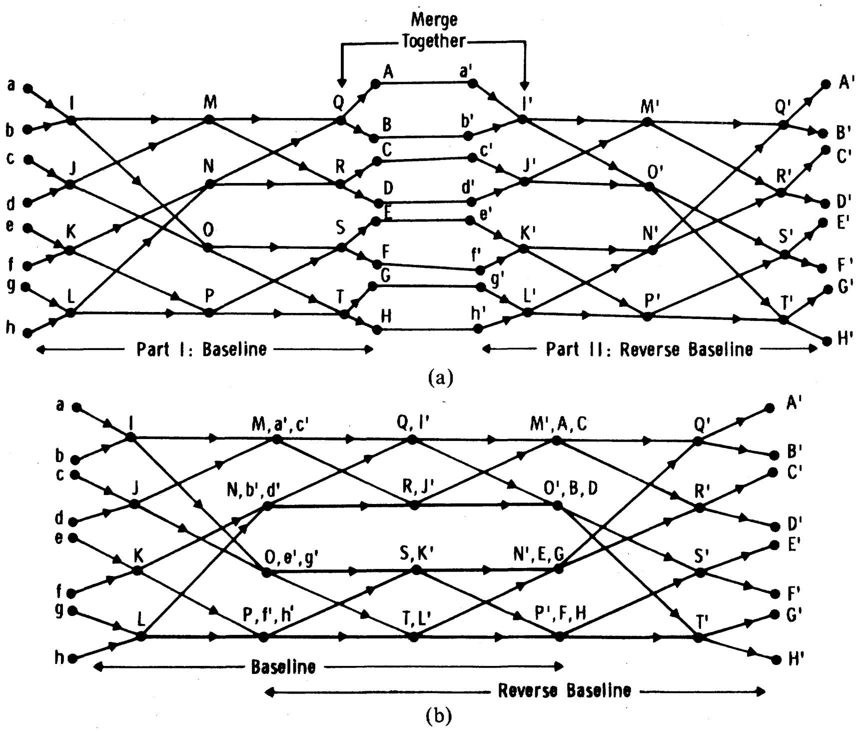


Fig. 19. (a) Composite graph of baseline and reverse baseline networks. (b) Reduced composite graph of baseline and reverse baseline networks.

property are satisfied by the MIN's,

b) buddy property between the two networks is satisfied, i.e., the input buddies of the last stage in the first MIN are merged with the output buddy pairs in the first stage of the second MIN.

Proof: Just like the Baseline MIN, two MIN's can be put together side by side as is shown in Fig. 20(a) with the details of the MIN's omitted for the generality of the networks. As per our Theorem 1, the MIN's 1 and 2 are isomorphic and topologically equivalent to the Baseline network. Q.E.D.

An example is shown in Fig. 21.

Corollary 6: Two passes, one forward and the other reverse direction in any n-stage MIN can provide any permutation without any conflict.

Proof: Any n-stage MIN and its reverse satisfy the conditions of Theorem 7 and the concatenation of a MIN and its reverse can simulate a Benes network. But, in place of putting the MIN and its reverse together, the same effect is observed if we have a MIN and move forward in one pass while going backward in the second pass. Q.E.D.

This concept is extremely useful in a supersystem as any permutation can be achieved in a n-stage MIN in just two passes. If the number of processing elements is considerably large, the implementation of n-stage MIN may be fairly complex, then it might be worth employing a single-stage network derived from specific MIN's employing similar interstage connections in all the stages. Lawrie's Omega MIN [24] is such an example. Very recently, several other single-stage networks have been reported [27]. The network derived from Omega, is shown in Fig. 22(a) Theorem 8 shows that this one stage Shuffle-Exchange-Reverse Shuffle network can provide any permutation in $(2n - 1)$ passes; which is lower than the existing bounds of $(3n - 1)$ passes for a shuffle-exchange single-stage network [28], [29]. In this network, Shuffle is used in the first n-passes by by feeding $a'b'c'd'e'f'g'h'$ to input $abcdefgh$. The Reverse Shuffle is used in the latter $(n - 1)$ passes by connecting outputs $ABCDEFGH$ to $A'B'C'D'E'F'G'H'$. An example of permutation shown in Fig. 10 requires two passes for the Baseline MIN. It can be shown that to achieve this permutation, the Omega network [24] also requires more than one pass. This permutation can be achieved by the one-stage network of Fig. 22(a) in $(2n - 1)$ passes and the state of the inputs and outputs in all the passes are shown in Fig. 22(b).

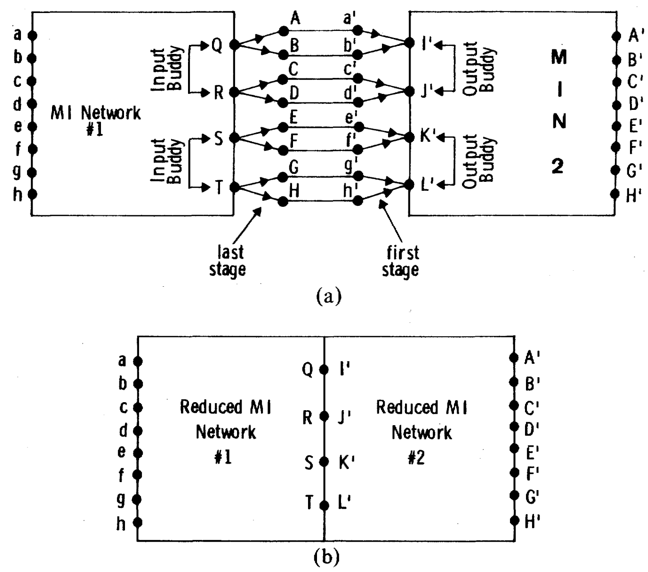


Fig. 20. (a) Composition of two MIN. (b) Reduced composition of two MIN.

Theorem 8: In a single-stage shuffle-exchange-reverse shuffle network, only $(2 \log_2 N - 1)$ passes are required for

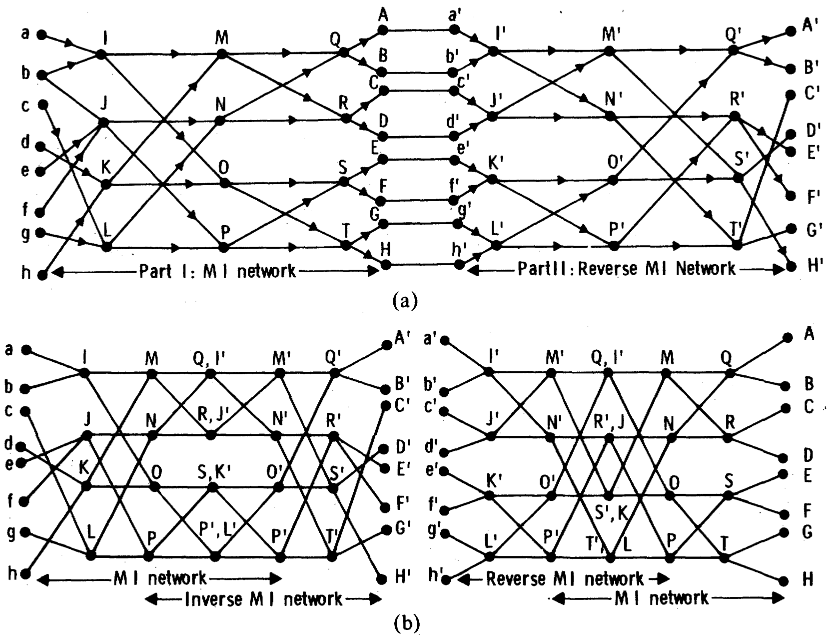


Fig. 21. (a) Composite graph of MIN and its reverse. (b) Reduced composite graph of MIN and its reverse. (c) Reduced composite graph of Reverse MIN and MIN.

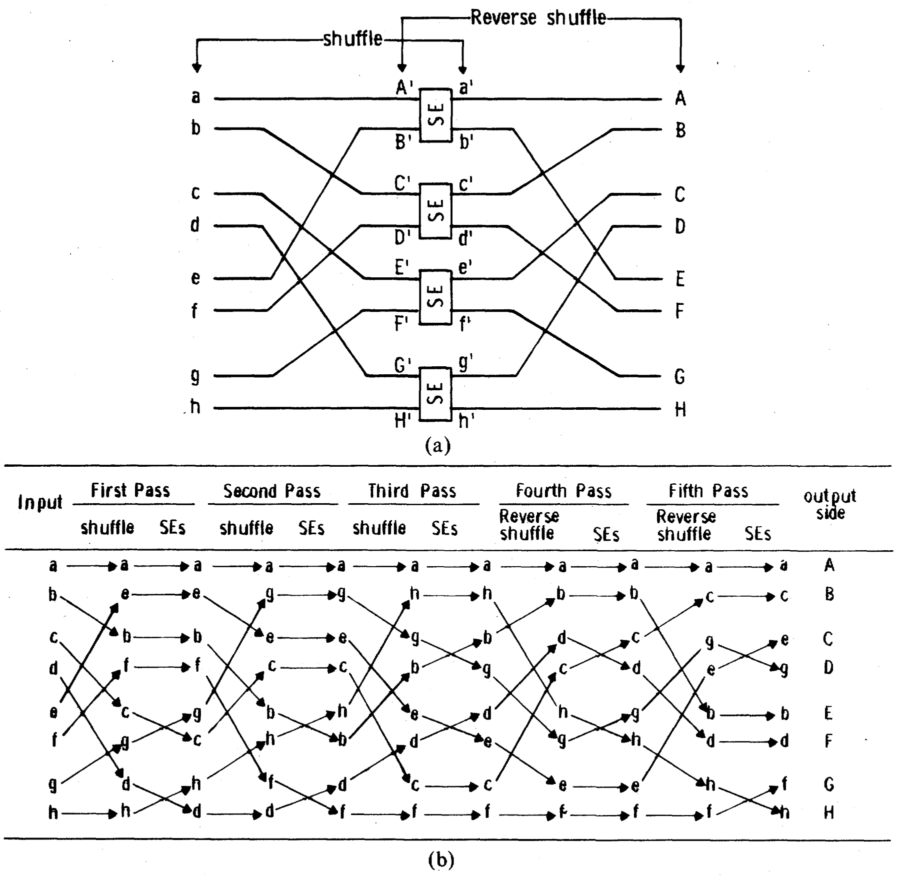


Fig. 22. (a) One stage Shuffle-Exchange-Reverse Shuffle Exchange network (n passes of shuffle and $(n-1)$ passes of reverse exchange required). (b) Successive passes in one-stage network of Fig. 22(a) for permutation $\begin{pmatrix} a\,b\,e\,d\,e\,f\,g\,h \\ ACEGBDFH \end{pmatrix}$.

any arbitrary permutation.

Proof: As per Theorem 7 and Corollary 6, a composite network obtained from overlapping Omega and Reverse Omega MIN's could simulate Benes network and a network similar to the Benes could be obtained. In this resulting structure, the first n-stages will consist of shuffle and exchange boxes and the later $(n-1)$ stages will have reverse-shuffle and exchange. In the network of Fig. 22(a), the first n-passes uses shuffle and exchange while the later $(n-1)$ passes employ reverse-shuffle and exchange. Hence, the network of Fig. 22(a) can be said to simulate a Benes network in $(2n - 1)$ passes. Q.E.D.

It may be of interest to the readers that several such single-stage networks have very recently been reported [27] which can simulate a n-stage MIN in just n-passes. This would mean that n-forward and $(n - 1)$ reverse passes through any one of these single-stage networks, could provide any arbitrary permutation.

IX. Concluding Remarks

This paper introduces novel approaches for analyzing and designing MIN's. so that they could be effectively utilized in future supersystems. The graph models are also helpful in comparing the permutability of MIN's with 2×2 SE's with the link controlled MIN's. The graph models provide a design methodology of a n-stage MIN such that a given permutation could be passed conflict-free. The versatility of the proposed graph theoretic technique is also reflected by the result showing that any buddy-type MIN can provide an arbitrary permutation in just two passes and this could be considered to be one of the major results enhancing the use of MIN's. Furthermore, for a single-stage network, the proposed technique allows the use of some specific single-stage networks in a simulating a Benes network and hence achieving conflict-free data transfers.

The proposed graph models could be easily extended for larger-sized SE's [30], [31] and redundant networks, and this would encourage the use of MIN's in supersystems. The graph model is also useful in testing of these networks, and the network partitioning could be easily done for the multiple SIMD/MIMD [32] computation mode.

References

[1] S. I. Kartashev and S. P. Kartashev, "Problems of designing supersystems with dynamic architectures," *IEEE Trans. Comput.*, vol. C-29, pp. 1114-1132, Dec. 1980.

[2] K. E. Batcher, "Bit-serial parallel processing systems," *IEEE Trans. Comput.*, vol. C-31, pp. 377-384, May 1982.

[3] D. K. Kuck and R. A. Stokes, "The Burroughs scientific processor (BSP)," *IEEE Trans. Comput.*, vol. C-31, pp. 363-376, May 1982.

[4] D. P. Agrawal and R. Jain, "A pipelined pseudoparallel system architecture for real time dynamic scene analysis," Special Issue, *IEEE Trans. Comput.*, vol. C-22, pp. 952-962, Oct. 1982.

[5] T. Y. Feng and D. P. Agrawal, "A study of communication processor systems," Rome Air Development Center, Tech. Rep. TR-29-310, Dec. 1979, 179 pp.

[6] L. Bhuyan and D. P. Agrawal, "Performance analysis of FFT algorithms on multiprocessor systems," *IEEE Trans. Software Eng.*, vol. SE-9, July 1983, to be published.

[7] B. Lint and T. Agerwala, "Communication issues in the design and analysis of parallel algorithms," *IEEE Trans. Software Eng.*, vol. SE-7, pp. 174-188, Mar. 1981.

[8] W. A. Wulf and C. G. Bell, "C.mmp—A multi- mini-processor," in *Proc. AFIPS Conf.*, vol. 41, part II, FJCC, 1972, pp. 765-777.

[9] K. E. Batcher, "The flip network in STARAN," in *Proc. 1976 Int. Conf. Parallel Process.*, Aug. 1976, pp. 65-71.

[10] J. B. Dennis, "The varieties of data flow computers," in *Proc. 1st Int. Conf. Distribut. Comput. Syst.*, Oct. 1-5, 1979, pp. 430-439.

[11] C. D. Thompson, "Generalized connection networks for parallel Processor intercommunication," *IEEE Trans. Comput.*, vol. C-27, pp. 1119-1125, Dec. 1978.

[12] H. J. Siegel, "A Model of SIMD Machines and a Comparison of Various Interconnection Networks," *IEEE Trans. Comput.*, vol. C-28, pp. 907-917, Dec. 1979.

[13] C. L. Wu and T. Y. Feng, "Routing Techniques for a class of multistage interconnection networks," *IEEE Trans. Comput.*, vol. C-29, pp. 694-702, Aug. 1980.

[14] M. A. Abidi and D. P. Agrawal, "On conflict-free permutations in multistage interconnection networks," *J. Digital Syst.*, vol. V, pp. 115-134, Summer 1980.

[15] M. A. Abidi, D. P. Agrawal, and J. J. Metzner, "Two single pass permutations in multistage interconnection networks," in *Proc. 14th Annu. Conf. Inform. Sci. Syst.*, Princeton, NJ, March 26-28, 1980, pp. 516-532.

[16] G. M. Masson, A. C. Ginher, and S. Nakamura, "A sampler of circuit switching networks," *Comput.*, pp. 32-48, June 1979.

[17] L. R. Goke and G. J. Lipovski, "Banyan networks for partitioning multiprocessor systems," in *Proc. Annu. Symp. Comput. Arch.*, Dec. 1973, pp. 21-28.

[18] D. M. Dias and J. R. Jump, "Analysis and simulation of buffered delta networks," *IEEE Trans. Comput.*, vol. C-30, pp. 273-282, Apr. 1981.

[19] J. H. Patel, "Processor-memory interconnection for multiprocessors," *IEEE Trans. Comput.*, vol. C-30, pp. 771-78, Oct. 1981.

[20] D. P. Agrawal and S. C. Kim, "On Nonequivalent multistage interconnection networks," in *Proc. 1981 Int. Conf. Parallel Process.*, Aug. 25-28, 1981, pp. 234-237.

[21] D. Opferman and N. Tsao-Wu, "On a class of rearrangeable switching networks," *Bell Syst. Tech. J.*, vol. 50, pp. 1579-1618, May-June 1971.

[22] T. Y. Feng, "Data manipulating functions in parallel processors and their implementations," *IEEE Trans. Comput.*, vol. C-23, pp. 309-318, Mar. 1974.

[23] H. J. Siegel and S. D. Smith, "Study of multistage SIMD interconnection networks," in *Proc. 5th Annu. Symp. Comput. Arch.*, Apr. 1978, pp. 223-229.

[24] D. Lawrie, "Access and alignment of data in an array processor," *IEEE Trans. Comput.*, vol. C-24, pp. 1145-1155, Dec. 1975.

[25] K. Y. Lee, "On the rearrangeability of a $(2 \log N - 1)$ stage permutation network," in *Proc. 1981 Int. Conf. Parallel Process.*, Aug. 25-28, 1981, pp. 221-228.

[26] C. L. Wu and T. Y. Feng, "The reverse exchange interconnection network," *IEEE Trans. Comput.*, vol. C-29, pp. 801-810, Sept. 1980. 1980.

[27] S. C. Kim, "Graph theoretic approach to multistage interconnection networks," Ph.D. dissertation, Wayne State Univ., ch. 6, pp. 114-127, July 1982.

[28] D. S. Parker, "Notes on shuffle/exchange-type switching networks," *IEEE Trans. Comput.*, vol. C-29, pp. 213-222, Mar. 1980.

[29] C. L. Wu and T. Y. Feng, "The universality of the shuffle-exchange networks," *IEEE Trans. Comput.*, vol. C-30, pp. 324-332, May 1981.

[30] L. Bhuyan and D. P. Agrawal, "VLSI performance of multistage interconnection networks using 4 * 4 switches," in *Proc. 3rd Int. Conf. Distribut. Comput. Syst.*, Miami, FL, Oct. 18-22, 1982, pp. 606-613.

[31] L. Bhuyan and D. P. Agrawal, "Design and performance of a general class of interconnection networks," in *Proc. 1982 Int. Conf. Parallel Process.*, Bellaire, MI, Aug. 24-27, 1982, pp. 2-9.

[32] D. P. Agrawal, "Testing and fault-tolerance of multistage interconnection networks," *Comput.*, vol. 15, pp. 41-53, Apr. 1982.

Dharma P. Agrawal (M'74-SM'79), for a biography and photograph, see p. 330 of the April 1983 issue of this TRANSACTIONS.