FUNCTIONAL AND TOPOLOGICAL RELATIONS AMONG BANYAN MULTISTAGE NETWORKS OF DIFFERING SWITCH SIZES

Abdou Youssef

Department of Elect. Eng. & Comput. Sci.
The George Washington University
Washington, DC 20052

Bruce Arden

College of Engineering and Applied Science
University of Rochester
Rochester, NY 14627

Abstract

Equivalence relations among banyan multistage networks have received much attention in recent years, mainly because two equivalent networks can simulate one another. Most of the efforts in this area have focused on comparing networks of the same building block (i.e., switch) size. This paper studies relations among banyan networks of differing switch sizes. If two $N \times N$ networks W and W' have switch sizes r and s, respectively, and if $r > s$, then W realizes a larger number of permutations than W'. Consequently, the two networks can never be equivalent. However, W may realize all the permutations of W', in which case W is said to functionally cover W' in the strict sense. More generally, W is said to functionally cover W' in the wide sense if the terminals of W can be relabeled so that W realizes all the permutations of W'. In this paper, functional covering is topologically characterized, and an optimal algorithm to decide strict functional covering is developed. The paper also shows that any $N \times N$ digit permutation network of switch size r functionally covers in the wide sense any other $N \times N$ digit permutation network of switch size s if and only if r is a perfect power of s, where a digit permutation network is a banyan multistage network such that the interconnections are permutations that permute digits in a specified manner.

§1. Introduction

Banyan multistage interconnection networks (MIN) have received much attention in recent years because of their key role in parallel processing systems [1]-[5], [9]-[11]. These networks have the unique path property, that is, there is a unique path between every source and every destination. As a result, they are efficiently controllable but do not realize all permutations [6], except in the special case when the network is a single crossbar.

As different MIN's may realize different sets of permutations, several reserach efforts have been directed towards functional equivalence relations among these networks [7] [10] [12], and algorithms to decide functional equivalence have been devised [7] [14]. However, all the research efforts have focused on symmetric equivalence relations, and often among networks of the same switch size.

In this paper, non-symmetric relations among MIN's of differing switch sizes are addressed. If two $N \times N$ MIN's have differing switch sizes, then the MIN with the larger switch size can be shown to realize a larger set of permutations. Therefore, the two networks can never be functionally equivalent. Consequently, the relation of inclusion of the smaller set in the larger set, or, stated otherwise, the relation of *functional covering* of one network by the other, is the appropriate relation to address. A network is said to *functionally cover* another network *in the strict sense* if the permutations realizable by the second network are realizable by the first network. A network is said to *functionally cover* another network *in the wide sense* if the terminals of the first network can be relabeled so that it functionally covers the second network in the strict sense.

To better see the merit of the covering problem, consider omega networks. It may seem intuitive that if $s < r$, then an $N \times N$ omega network with $r \times r$ crossbar switches as building blocks realizes all the permutations of another $N \times N$ omega network with $s \times s$ crossbar switches as building blocks. However, this is not always the case. Take for example two 64×64 omega networks, Ω_1 and Ω_2, with 8×8 and 4×4 crossbar switches as building blocks, respectively. It is clear that Ω_1 has 2 columns of 8 switches each, and realizes $(8!)^{2\times 8} = (40320)^{16}$ permutations, noting that the number of realizable permutations is the number of states of each switch raised to a power equal to the number of switches [6]. On the other hand, Ω_2 has 3 columns of 16 switches each, and realizes $(4!)^{3\times 16} = (13824)^{16}$ permutations, a much smaller set than that of Ω_1. However, not every permutation realizable by Ω_2 is realizable by Ω_1. In fact, using the characterization of non-conflicting source-destination paths in omega [6], it can be shown that the paths $0 \rightarrow 0$ and $24 \rightarrow 5$ do not conflict in Ω_2 but do conflict in Ω_1. Hence, there exists a permutation that maps 0 to 0 and 24 to 5, and that is realizable by Ω_2 but not by Ω_1. This would complicate the migration of parallel algorithms from a parallel system that is Ω_2-interconnected to another system that is Ω_1-interconnected. Functional covering is of practical interest for upward system compatibility and for network simulation. Most of the proposed MIN's use 2×2 crossbar switches as building blocks, but can be easily generalizable to use switches of larger sizes [6], [13]. It is therefore of interest to know whether, and under what

CH2908-2/90/0000/0284/$01.00 © 1990 IEEE

conditions, the permutations realizable by each of these networks remain realizable after the switches are upgraded to larger sizes, for this allows upward system compatibility. It is also of interest to know if one MIN functionally covers another MIN in the wide sense, for then the first network can simulate the second by relabeling the terminals of the first network.

This paper will investigate functional covering among MIN's of the same terminal size but of differing switch sizes. Necessary and sufficient conditions for a MIN to functionally cover another MIN will be determined. Specifically, it will be shown that an $N \times N$ MIN of $r \times r$ switches functionally covers in the strict sense another $N \times N$ MIN of $s \times s$ switches if and only if r is a power of s (i.e., $r = s^l$ for some integer l) and the topology of the second MIN can be derived from the first MIN by replacing each switch of the first MIN by some $r \times r$ MIN of $s \times s$ switches. Based on this topological characterization of functional covering, an optimal algorithm to decide functional covering in the strict sense will be given. Wide functional covering decision algorithms are harder to develop and are left for future work.

It will also be shown that most existing MIN's, such as omega [6], omega inverse, the indirect binary n-cube [8], the generalized cube network [10], and the baseline network [12] will *grow inclusively* as their switch size grows to a power. That is, any $N \times N$ such network of $r \times r$ switches functionally covers in the strict sense any $N \times N$ network of the same type and with $s \times s$ switches such that r is a power of s.

Finally, network covering among the networks of a special class of MIN's, called digit permutation networks (DPN) [13], will be studied. In a digit permutation network, the inter-column interconnections are digit permutations that permute digits in a specified manner. All existing MIN's are examples of digit permutation networks. It was shown in [13], [15] that all $N \times N$ digit permutations network of the same switch size are widely functionally equivalent, that is, every digit permutation network realizes the same permutations of every other digit permutation network after relabeling the terminals of either one of the two networks. Combining the fact that all digit permutation networks are widely functionally equivalent and the fact that omega is a digit permutation network that grows inclusively as its switch size grows to a power, it will be concluded that every $N \times N$ digit permutation network with $r \times r$ switches functionally covers in the wide sense any other $N \times N$ digit permutation network of $s \times s$ switches such that r is a power of s. This enables the first network to simulate the second. An optimal $O(N \log_s N)$ algorithm to relabel the terminals of a covering DPN in order to simulate a covered DPN will also be given.

The paper is organized as follows. The next section will review multistage networks and rigorously define the functional covering relations and related concepts. Section 2 will establish the necessary and sufficient conditions in order for a MIN to functionally cover another MIN. The algorithm to decide strict functional covering will be developed in section 3. Section 4 will show that most existing MIN's grow inclusively as their switch size grows to a power. The wide functional covering among digit permutation networks will be addressed in section 5. Section 6 will give some concluding remarks and future directions.

§1. Preliminaries and Fundamental Concepts

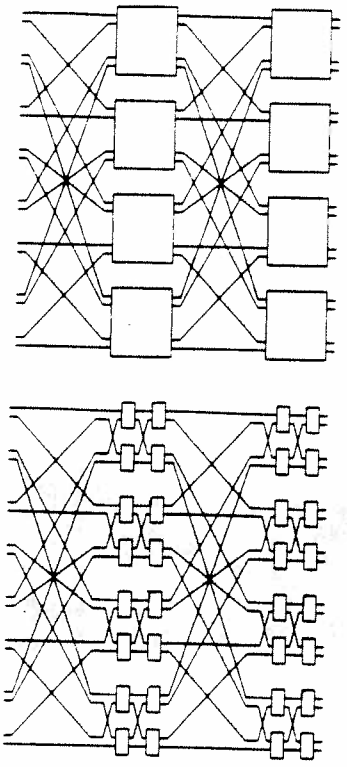
In this section banyan multistage interconnection networks are specified, functional and topological equivalence relations among them are reviewed, and functional and topological covering relations will be defined.

Banyan multistage interconnection networks have N input terminals, N output terminals, and k interconnected columns of $\frac{N}{r}$ $r \times r$ crossbar switches, where $N = r^k$ and $r \geq 2$. N is called the terminal size of the network. Each $r \times r$ crossbar switch realizes all $r!$ permutations. The interconnection between every two successive columns is a permutation of $S_N = \{0, 1, ..., N-1\}$. The connectivity of these networks is such that they have the unique path property. That is, between every input terminal i and every output terminal j there is one and only one path, which will be denoted $i \rightarrow j$. The class of these networks is denoted MIN(r, k). Figure 1 shows two MIN's.

For ease of reference, the input (i.e., left) terminals of networks in MIN(r, k) are labeled 0, 1, ... , $N-1$ from top to bottom, and so are the output (right) terminals. The columns are numbered 0, 1, ... , $k-1$ from left to right, and the switches of each column are labeled 0, 1, ... , $\frac{N}{r}-1$ from top to bottom.

If W is a network of MIN(r, k) and f a permutation of S_N, f can be viewed as an interconnection and can be appended to the right end of W, forming a network denoted Wf. Another way of viewing Wf is as W except that the output terminals of W are relabeled by f, that is, output terminal j is relabeled $f(j)$, for every $j = 0, 1, ..., N-1$. Similarly, f can be appended to the left of W forming fW. Viewed differently, fW is the same as W except that the input terminals of W are relabeled by f^{-1}, that is, every input terminal i of W is relabeled $f^{-1}(i)$.

The composition of functions is taken here from left to right, that is, $(x)fg = g(f(x))$. If $P(W)$ denotes the set of permutations realizable by W, then $P(gWf) = \{ghf \mid h \in P(W)\}$, which clearly follows from the definition of gWf



Two Banyan Multistage Networks

Figure 1

and the left-to-right view of composition.

Two networks W and W' in MIN(r,k) are *strictly functionally equivalent* if they realize the same permutations. The two networks are *widely functionally equivalent* if they can be made to realize the same permutations by relabeling the input and/or output terminals of one of the networks, that is, if there exist two permutations g and f of S_N such that gWf and W' are strictly functionally equivalent. A network W in MIN(r,k) is said to *functionally cover* another network W' in MIN(s,k') *in the strict sense* if $P(W')$ is a subset of $P(W)$. The network W is said to functionally cover W' *in the wide sense* if W can be made to functionally cover W' in the strict sense by relabeling the input and/or output terminals of W, that is, if there exist two permutations g and f of S_N such that gWf realizes all the permutations of W'.

Topological relations are defined next. To this effect, two simple operations on networks are specified. The first, called permute-links-within-switch (PL), consists of disconnecting the links connected to one side (input or output) of an $r \times r$ switch of the network and reconnecting them to different ports of the same side of the same switch. The second, called permute-switches-within-column (PS), consists of permuting the switches within a column in such a way that the links wired to a repositioned switch remain wired to it. Two networks in MIN(r,k) are *strictly topologically equivalent* if one network can be derived from the other by a sequence of PL and PS operations, and *widely topologically equivalent* if one can be derived from the other by a sequence of PL and PS operations and by relabeling the input and output terminals. It is clear that if two networks are strictly (resp. widely) topologically equivalent, then they must be strictly (resp. widely) functionally equivalent because the PS and PL operations do not fundamentally alter the structure of the network.

An $N \times N$ network W in MIN(r,k) is said to *topologically cover* another $N \times N$ network W' in MIN(s,k') *in the strict sense* if r is a power of s (i.e., $r = s^l$ for some integer l) and the switches of W can be replaced by networks in MIN(s,l) such that the resulting network is strictly topologically equivalent to W'. The network in Figure 1-(a) topologically covers the network in Figure 1-(b). The network W is said to topologically cover W' *in the wide sense* if W can be made to topologically cover W' in the strict sense by relabeling the input and/or output terminals of W. As the permutations realizable by any network in MIN(s,l) form a subset of the $r!$ permutations realizable by an $r \times r$ switch, it follows that strict topological covering implies strict functional covering. Similarly, wide topological covering implies wide functional covering.

Since the proof of the equivalence between functional covering and topological covering will proceed by induction on the number of columns of the covering network, the class of *incomplete* MIN's will be introduced. Define by IMIN(r,k,t) the class of $N \times N$ networks of t columns of $r \times r$ switches, where $N = r^k$, $t \leq k$ and each input terminal can reach exactly r^t output terminals through unique paths but cannot reach any of the other output terminals. Clearly, IMIN(r,k,k) = MIN(r,k). All the functional and topological relations can be extended to networks in IMIN's with only one addition to strict functional covering as follows. A network W in IMIN(r,k,t) is said to *functionally cover* another network W' in IMIN(s,k',t') *in the strict sense* if $P(W')$ is a subset of $P(W)$ and the set of output terminals reached from an input terminal in W' is the same set reached in W from the same input terminal. This second condition is equivalent to saying that $r^t = s^{t'}$, and is hence superfluous in the case of MIN's, that is, when $t = k$ and $t' = k'$.

§2. Equivalence between Functional Covering and Topological Covering

In this section it will be shown that a network in MIN(r,k) functionally covers another network in MIN(s,k') in the strict sense (resp. wide sense) if and only if the

first network topologically covers the second network in the strict sense (resp. wide sense). The equivalence in the strict sense is shown first, and the equivalence in the wide sense will follow as a corollary.

The proof will proceed by induction on the number of columns. For that reason, the equivalence between strict functional covering and strict topological covering will be established for the IMIN classes. A number of supporting lemmas are shown first.

2.1 Lemma. *Let W be in IMIN(r,k,l), E_i a subset of switches of some column i of W, and E_{i+1}, ... , E_j defined inductively as follows. E_t is the set of switches in column t that are linked to switches in E_{t-1}, for $t = i+1, i+2, ..., j$. Then*

(i) $|E_i| \leq |E_{i+1}| \leq ... \leq |E_j|$.

(ii) If $|E_i| \geq |E_j|$, then $|E_i| = |E_{i+1}| = ... = |E_j|$ and the switches of E_i, E_{i+1}, ... , E_j along with their interconnections form an independent $n \times n$ subnetwork of W, where $n = r|E_i|$.

Proof. (i) The number of links going out of the switches of E_i is $r|E_i|$. All these links come into the switches of E_{i+1}. As the total number of links incoming to the switches of E_{i+1} is $r|E_{i+1}|$, it follows that $r|E_i| \leq r|E_{i+1}|$, implying that $|E_i| \leq |E_{i+1}|$. The remaining inequalities can be shown similarly.

(ii) If $|E_i| \geq |E_j|$, then, using (i), we have $|E_i| = |E_{i+1}| = ... = |E_j|$. This shows that all the links coming to the switches of E_{i+1} are from the switches of E_i, those coming into the switches of E_{i+2} are from E_{i+1} and so on. As a result, the switches of E_i, E_{i+1}, ... , E_j along with their interconnections form an $n \times n$ subnetwork of W, where $n = r|E_i|$. ∎

2.2 Lemma. *Let W be in IMIN(r,k,t) and W' in IMIN(s,k',t') such that W functionally covers W' in the strict sense. Let also a_1, a_2, ... , a_r be the input terminals linked to switch x in column 0 of W, and j_0 the label of the leftmost column in W' in which there exists at least one switch u that is reachable from all a_1, a_2, ... , a_r in W'. Then $\frac{k'}{k} - 1 \leq j_0 < \frac{k'}{k}$ and hence $j_0 = \lceil \frac{k'}{k} - 1 \rceil$.*

Proof. To show that $\frac{k'}{k} - 1 \leq j_0$, note that the number of input terminals reachable from u in W' is s^{j_0+1} on the one hand and $\geq |\{a_1, a_2, ..., a_r\}| = r$ on the other hand. Thus, $s^{j_0+1} \geq r = s^{\frac{k'}{k}}$ and, therefore, $j_0 \geq \frac{k'}{k} - 1$.

To show that $j_0 < \frac{k'}{k}$, we will reason by contradiction, assuming that $j_0 \geq \frac{k'}{k}$. Let B_i be the set of output terminals of W' reachable from output port i of switch u, for $i = 1, 2, ..., s$. Let A_1, A_2, ... , A_r be the sets of output terminals of W that are reachable from output ports $1, 2, ..., r$ of switch x, respectively, as shown in Figure 2. Let also $O_u = \cup_{i=1}^{s} B_i$ be the set of all the output terminals of W' reachable from u, $I = \{j \mid O_u \cap A_j \neq \emptyset\}$, and $E = \cup_{j \in I} A_j$. We clearly have $O_u \subseteq E$. We will derive next a contrdiction in each case whether $O_u = E$ or O_u is proper subset of E. The approach is to find two paths that conflict in W but not in W', contradicting that W functionally covers W' in the strict sense.

Case 1: $O_u = E$. Then for every $j \in I$, $A_j \subseteq \cup_{i=1}^{s} B_i$. We will show that $|B_i| < |A_j|$. To see this, note that $|B_i| = s^{t'-j_0-1} \leq s^{t'-\frac{k'}{k}-1}$. Since W functionally covers W', it follows that $r^t = s^{t'}$ and $r^k = s^{k'}$. Therefore, $\frac{k'}{k} = \frac{t'}{t}$, $s = r^{\frac{k}{k'}}$ and hence

$$s^{t'-\frac{k'}{k}-1} = r^{\frac{k}{k'}(t'-\frac{k'}{t}-1)} = r^{t-1-\frac{k}{k'}} < r^{t-1}.$$

As $|A_j| = r^{t-1}$, it follows that $|B_i| < |A_j|$. Therefore, A_j cannot be contained in any single B_i. Consequently, there exist i_1 and i_2 such that $A_j \cap B_{i_1} \neq \emptyset$ and $A_j \cap B_{i_2} \neq \emptyset$. Let h be in $A_j \cap B_{i_1}$ and h' in $A_j \cap B_{i_2}$. Take any input terminal a_l such that the paths from a_1 to u and from a_l to u do not conflict in W' (such an element must clearly exist). It can now be seen that the paths $a_1 \rightarrow h$ and $a_l \rightarrow h'$ do not conflict in W' because $B_{i_1} \cap B_{i_2} = \emptyset$, while these same paths conflict in W because both h and h' are in A_j and the two paths have to go through the output port j of switch x. This contradicts the fact that W functionally covers W' in the strict sense.

Case 2: O_u is a proper subset of E. Let $O_u^c = E - O_u$ which is non-empty. As $O_u^c \subseteq E$ and $O_u^c \neq \emptyset$, there exists $i_0 \in I$ such that $O_u^c \cap A_{i_0} \neq \emptyset$. Since i_0 is in I, we have $O_u \cap A_{i_0} \neq \emptyset$. Therefore, there exist two output terminals h and h' such that $h \in O_u \cap A_{i_0}$ and $h' \in O_u^c \cap A_{i_0}$. Here too the paths $a_1 \rightarrow h$ and $a_l \rightarrow h'$ do not conflict in W' while these same paths conflict in W because both h and h' are in A_{i_0} and the two paths have to go through the switch x. Thus, we have the same contradiction as in the previous case.

Therefore, the assumption $j_0 \geq \frac{k'}{k}$ must be false. ∎

2.3 Lemma. *Let W, W', a_1, a_2, ... , a_r, x, j_0 and u be as in the previous lemma. Then the following statements hold:*

(i) Let F be the set of switches in column j_0 of W' that are reachable from every input terminal in the set $\{a_1, a_2, ..., a_r\}$, and F_i the set of switches in column j_0 that are reachable from input terminal a_i. Then $F_i = F$ for every $i = 1, 2, ..., r$.

(ii) For every two output ports c and d of the switches in F there exist two input terminals a_{j_1} and a_{j_2} such that the two paths $a_{j_1} \rightarrow c$ and $a_{j_2} \rightarrow d$ going through the first

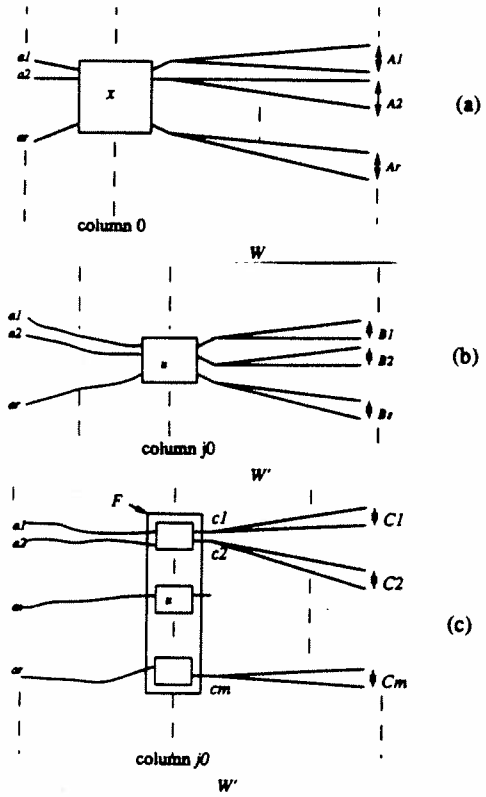


Figure 2

$j_0 + 1$ *columns of* W' *do not conflict.*

(iii) $r = s^{j_0+1}$.

(iv) The switches that can be traced from $a_1, a_2, \ldots, a_r$ *rightward to column* j_0 *in* W' *form an* $r \times r$ *independent subnetwork* S *in* $MIN(s, j_0 + 1)$.

Proof. (i) Clearly F is a subset of F_i. To show that F_i is a subset of F, let p be an arbitrary switch in F_i. Let also d be an output terminal reachable from a_i through p in W'. Since W functionally covers W' in the strict sense, it follows that the path $a_i \rightarrow d$ is realizable in W. This path must go through switch x, and thus, $a_1, a_2, \ldots, a_r$ can all reach d in W. Due to the strict functional covering, d must be reachable from $a_1, a_2, \ldots, a_r$ in W' as well. There must then exist a switch that is reachable from $a_1, a_2, \ldots, a_r$ in W' on their way to d. Let v be the leftmost such switch and j_1 the label of the column of v. Lemma 2.2 applies to v and j_1 as it does to u and j_0. Therefore, $j_1 = \lceil \frac{k'}{k} - 1 \rceil$. Hence, $j_1 = j_0$ and v is in column j_0. As the path $a_i \rightarrow d$ goes through switch p and switch v in W', and as p and v are in the same column, we must have $p = v$. Consequently, p is reachable from $a_1, a_2, \ldots, a_r$ in W', implying that p belongs to F. It follows that $F_i = F$ for every $i = 1, 2, \ldots, r$.

ii) We have two cases. The first is when both c and d are two output ports of one switch. The second is when they are output ports of two distinct switches.

Case 1: Let v be the switch of which c and d are two output terminals. Let G_i be the set of input terminals of W' that are reachable from input port i of v, for $i = 1, 2, \ldots, s$. As $v \in F$, the input terminals $a_1, a_2, \ldots, a_r$ are reachable from v in W'. Therefore,

$$\{a_1, a_2, \ldots, a_r\} \subseteq \cup_{i=1}^{s} G_i. \tag{1}$$

It is also clear that

$$|G_i| = s^{j_0} < s^{\frac{k'}{k}} = r. \tag{2}$$

It follows from (1) and (2) that the set $\{a_1, a_2, \ldots, a_r\}$ cannot be contained in a single G_i. Therefore, there exist two distinct integers i_1 and i_2, and two input terminals a_{j_1} and a_{j_2} such that $a_{j_1} \in G_{i_1}$ and $a_{j_2} \in G_{i_2}$. Consequently, the paths $a_{j_1} \rightarrow c$ and $a_{j_2} \rightarrow d$ do not conflict in W'.

Case 2: Let v and z be the two switches such that c is an output port of v and d an output port of z. Let G_{i_1}, G_{i_2}, a_{j_1} and a_{j_2} be as in the previous case, and H_i the set of input terminals of W' that are reachable from input port i of switch z. Here too the a_i's belong to the union of the H_i's. Therefore, a_{j_2} must belong to some H_{i_3}. The paths $a_{j_1} \rightarrow i_1$ and $a_{j_2} \rightarrow i_3$ do not conflict in W' because if they did, then a_{j_2} would be reachable from input port i_1 of v and hence $G_{i_1} \cap G_{i_2} \neq \emptyset$, which is impossible because the sets of input terminals reachable from two distinct input ports of a switch are disjoint. Consequently, the paths $a_{j_1} \rightarrow c$ and $a_{j_2} \rightarrow d$ do not conflict in W'.

(iii) Let $m = |F| = |F_i| = s^{j_0+1}$, and $c_1, c_2, \ldots, c_m$ be the output ports of the switches in F (also in F_i). Let also C_i be the set of output terminals reachable from c_i in W', and A_j be the set of output terminals reachable from output port j of switch x in W. We will show that for every $j = 1, 2, \ldots, r$, there exists $i = 1, 2, \ldots, m$ such that $A_j = C_i$. Afterwards, noting that $|A_j| = r^{t-1}$, $|C_i| = s^{t'-j_0-1} = \frac{s^{t'}}{s^{j_0+1}} = \frac{r^{t'}}{s^{j_0+1}}$, and $|C_i| = |A_j|$, we will conclude that $r = s^{j_0+1}$.

As all the switches in column j_0 reachable from a_i in W' form the set $F_i = F$, it follows that the output ports in column j_0 that are reachable from a_i are $c_1, c_2, \ldots, c_m$, and, consequently, the set of output terminals reachable from a_i in W' is $\cup_{i=1}^{m} C_i$. On the other hand, the set of output terminals reachable from input terminal a_i in W is $\cup_{j=1}^{r} A_j$. As W functionally covers W' in the strict sense, these two sets must be equal. Therefore, for every $j = 1, 2, \ldots, r$, A_j is a subset of $\cup_{i=1}^{m} C_i$. We claim that there must exist an i such that $A_j \subseteq C_i$, for otherwise, there would exist two distinct integers i_1 and i_2 such that $A_j \cap C_{i_1} \neq \emptyset$ and $A_j \cap C_{i_2} \neq \emptyset$. It would follow that there exist two output

terminal labels $e \in A_j \cap C_{i_1}$ and $f \in A_j \cap C_{i_2}$. Using (ii), we conclude that there exist a_{j_1} and a_{j_2} such that the paths $a_{j_1} \rightarrow c_{i_1}$ and $a_{j_2} \rightarrow c_{i_2}$ do not conflict in W'. Therefore, the paths $a_{j_1} \rightarrow e$ and $a_{j_2} \rightarrow f$, going through c_{i_1} and c_{i_2}, respectively, do not conflict in W' because $C_{i_1} \cap C_{i_2} = \emptyset$. However, these same paths conflict in W over output port j of switch x because both e and f are in A_j. Consequently, there must exist i such that $A_j \subseteq C_i$. Noting that $|C_i| = \frac{r^t}{s^{j_0+1}} \leq r^{t-1} = |A_j|$, it follows that $A_j = C_i$. Using now the equality between the cardinalities between A_j and C_i as indicated earlier, we conclude that $r = s^{j_0+1}$.

(iv) Let E_0 be the set of switches of column 0 of W' such that each switch is linked to at least one of $a_1, a_2, ..., a_r$. Let also E_i be the set of switches in column i that are linked to switches in E_{i-1}, for $i = 1, 2, ..., j_0$. After Lemma 2.1-(i) we have

$$|E_0| \leq |E_1| \leq ... \leq |E_{j_0}|. \tag{3}$$

Every switch of E_{j_0} is clearly reachable from one of the inputs $a_1, a_2, ..., a_r$. Consequently, $E_{j_0} \subseteq \cup_{i=1}^{r} F_i = F$, and hence

$$|E_{j_0}| \leq |F_i| = s^{j_0}. \tag{4}$$

As the number of all input terminals linked to the switches of E_0 is equal to $s|E_0|$ on the one hand and $\geq |\{a_1, a_2, ...a_r\}| = r$, it follows that $|E_0| \geq \frac{r}{s} = \frac{s^{j_0+1}}{s} = s^{j_0} \geq |E_{j_0}|$. Thus,

$$|E_0| \geq |E_{j_0}|. \tag{5}$$

The inequalities of (3) and (5) enable us to use Lemma 2.1-(ii) concluding that the switches of $E_0, E_1, ..., E_{j_0}$ along with their interconnections form an independent $n \times n$ network of W', where $n = s|E_0| = s^{j_0+1} = r$. As the total number of input terminals linked to the switches of E_0 is r, it follows that these input terminals are $a_1, a_2, ..., a_r$, and the lemma follows. ∎

2.4 Lemma. *Let W be in IMIN(r, k, t) and W' in IMIN(s, k', t') such that W functionally covers W' in the strict sense. Let also $l = \log_s r$. The leftmost l columns of W' can be regrouped into $r \times r$ subnetworks in MIN(s, l). Furthermore, the regrouping can be done so that the leftmost interconnections of W and W' (between the input terminals and the leftmost column) become identical, and for every j and i, the set of output terminals reachable from output port i of switch j of column 0 of W is identical to the set of output terminals reachable from output terminal i of the j-th subnetwork in W' (see Fig. 3).*

Proof. After part (iii) of the previous lemma, r is a power of s and hence l is an integer. After part (iv) of that same lemma, every $r \times r$ switch u of column 0 of W corresponds to an $r \times r$ independent subnetwork in the l leftmost columns of W' with the same input terminals as those connected to switch u. Therefore, the l leftmost columns of W' can be regrouped so that the $r \times r$ subnetwork corresponding to switch j of column 0 of W is identifiable, positioned in the j-th position (from the top), and having its inputs connected to the r input terminals in the same way as the corresponding input ports of switch j are connected to the same input terminals.

Following the same line of reasoning as in the proof of part (iii) of Lemma 2.3, we conclude that for every output port i of the switch j of column 0 of W, there exists an output terminal p_i of the corresponding network such that the set of the labels of the output terminals of W reachable from i is the same as the set of output terminals of W' reachable from p. Accordingly, for every i, the output p_i of the subnetwork is relabeled i in order for the switch-subnetwork correspondence to be perfect. ∎

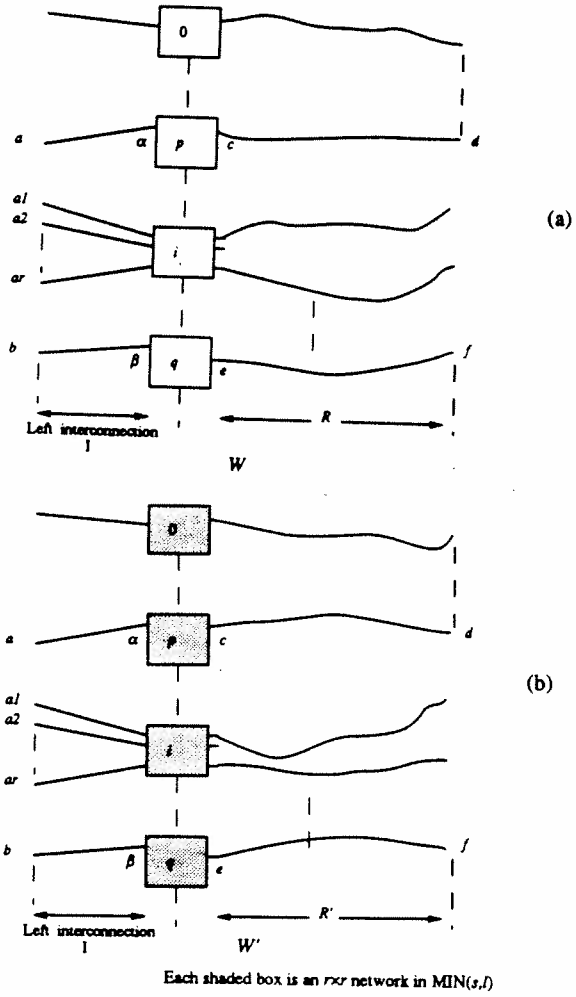


Figure 3

Now we are in a position to show by induction the equivalence between strict functional covering and strict topological covering.

2.5 Theorem. *A network W in IMIN(r, k, t) functionally covers W' in IMIN(s, k', t') in the strict sense if and only if r is a power of s and W topologically covers W' in the strict sense.*

Proof. We have seen before that topological covering implies functional covering. It remains to show the converse. Let W be a network in IMIN(r, k, t) that functionally covers another network W' in IMIN(s, k', t') in the strict sense. After Lemma 2.3-(iii), r is power of s, that is, there exists a positive integer l such that $r = s^l$. It will be shown by induction on t that W topologically covers W' in the strict sense.

Basis step: $t = 1$, that is, W has only one column of switches. As $r^t = s^{t'}$, we have $r = s^{t'}$ and thus $l = t'$. After Lemma 2.4, the l leftmost columns of W', which in this case are all the columns of W', can be regrouped into $r \times r$ subnetworks in MIN(s, l) in perfect one-to-one correspondence with the switches of column 0 of W. It can be easily seen that by replacing each subnetwork of W' by an $r \times r$ switch, we get a network identical to W.

Induction step: Assume that if a network in IMIN$(r, k, t-1)$ functionally covers another network in IMIN(s, k', t') in the strict sense, the former network then topologically covers the latter in the strict sense. After the previous lemma, the l leftmost columns of W' can be regrouped into $r \times r$ subnetworks in perfect one-to-one correspondence with the switches of the leftmost column of W such that the leftmost interconnections of both networks become identical. Do the regrouping as described in the previous lemma, then delete the leftmost interconnection and these resulting subnetworks from W', and call the resulting network R' (Fig. 3-(b)). Delete also the leftmost interconnection and the leftmost column from W, and call the resulting network R (Fig. 3-(a)). We will show that R functionally covers R' in the strict sense by showing that every two source-destination paths that do not conflict in R' do not conflict in R. Afterwards, the inductive hypothesis can be applied. Let $c \rightarrow d$ and $e \rightarrow f$ be two source-destination paths that do not conflict in R'. We need to show first that these two paths are realizable in R. The input terminals c and e can be viewed as outputs of some subnetworks p and q of the l leftmost columns of W'. They must also be two output ports of switches p and q in column 0 of W. Using the previous lemma again, the output port c reaches the same output terminals in W and W'. Therefore, c reaches the output terminal d in W because it does in W'. Similarly, the output port e reaches the output terminal f in W. It remains to show that the paths $c \rightarrow d$ and $e \rightarrow f$ do not conflict in R.

Whether $p = q$ or not, there must clearly exist two input terminals α and β of the subnetworks p and q such that the paths $\alpha \rightarrow c$ and $\beta \rightarrow e$ do not conflict in the first l columns of W'. Let a and b be the two input terminals of W' that are linked to α and β. Then the source-destination paths $a \rightarrow d$ and $b \rightarrow f$ do not conflict in W'. Consequently, these same paths do not conflict in W. Clearly these paths must go through the output ports c and d in W, respectively. Therefore, the paths $c \rightarrow d$ and $e \rightarrow f$ do not conflict in R. It follows that R functionally covers R' in the strict sense. Using the inductive hypothesis, we conclude that R topologically covers R' in the strict sense. As the first stage of W topologically covers the portion of W' that is missing from R', it follows that W topologically covers W' in the strict sense. ∎

2.6 Theorem. *A network W in IMIN(r, k, t) functionally covers a network W' in IMIN(s, k', t') in the wide sense if W topologically covers W' in the wide sense.*

Proof. By definition, W topologically covers W' in the wide sense if and only if W topologically covers W' in the strict sense after appropriate relabeling of the terminals of W', that is, by Theorem 2.5, if and only if W functionally covers W' in the strict sense after appropriate relabeling of the terminals of W'. As the latter statement is equivalent to saying that W functionally covers W' in the wide sense (by definition), the theorem follows. ∎

From the previous two theorems, the equivalence between the functionality and topology of the networks in MIN(r, k), proved in [14], follows here as a corollary:

2.7 Theorem. *Two networks W and W' in IMIN(r, k, t) are strictly (resp. widely) functionally equivalent if and only if they are strictly (resp. widely) topologically equivalent.*

Proof. Two networks in IMIN(r, k, t) are strictly (resp. widely) functionally equivalent if and only if each network functionally covers the other in the strict (resp. wide) sense, that is, by the previous two theorems, if and only if each network topologically covers the other in the strict (resp. wide) sense. As the latter statement is equivalent to saying that the two networks are strictly (reps. widely) topologically equivalent, the theorem follows. ∎

§3. Network Covering Algorithm

This section presents an algorithm that takes two $N \times N$ networks W in MIN(r, k) and W' in MIN(s, k') as input and decides if W functionally covers W' in the strict

sense. The algorithm makes use of the equivalence between functional covering and topological covering, and also of the strict functional/topological equivalence algorithm of [14], a brief description of which will be presented later in this section.

The covering algorithm checks first if r is a power of s and halts if not. Otherwise, it breaks the columns of W' into k consecutive *bands* of $l = \log_s r$ columns each, where the i-th band consists of columns $i \times l, i \times l + 1, i \times l + 2, ..., (i+1) \times l - 1$. Afterwards, it examines each band to see whether it consists of $\frac{N}{r}$ independent networks in MIN(s, l). If one of the bands does not, the overall algorithm halts answering negatively; otherwise, it replaces each subnetwork by an $r \times r$ switch. The links coming to and going from each of these $r \times r$ switches are the same links coming to the leftmost column and going from the rightmost column of the corresponding subnetwork. After this regrouping and subnetwork-to-switch change, call the resulting network W". Finally, the algorithm checks if the network W" is strictly topologically equivalent to the network W, using the network equivalence algorithm in [14]. Clearly, W is strictly topologically equivalent to W" if and only if W topologically covers W' in the strict sense, which in turn is true if and only if W functionally covers W' in the strict sense.

It remains to see how to decide if a band of l columns consists of subnetworks in MIN(s, l), and if so, how to regroup the switches into these independent subnetworks. The main idea is to start at the top leftmost switch of the band, trace rightward the tree of switches to the rightmost column of the band, and denote by F the set of the $\frac{r}{s}$ switch-leaves. Then trace backward starting from an arbitrary switch of F, up to the lefmost column of the band, denoting by $E_{i \times l}$ the set of switches reached in that column. Afterwards, march rightward from the switches of $E_{i \times l}$ computing $E_{i \times l+1}$, $E_{i \times l+2}$, ... , $E_{(i+1) \times l-1}$, where E_j is the set of switches (in column j) that are linked to switches in E_{j-1} (of column $j-1$). If $|E_j| \neq \frac{r}{s}$ for some j, the procedure halts the overall algorithm answering that W does not cover W'. If, on the other hand, all these numbers of switches in each of the columns in the band are equal to $\frac{r}{s}$, then these switches form an independent network in MIN(s, l) and are replaced by an $r \times r$ switch. The same cycle is repeated on the remaining switches of the band till the entire band is regrouped or till the regrouping fails somewhere. This procedure is called REGROUP-BAND(i), where i refers to the i-th band that is to be regrouped.

The main steps of the covering algorithm are summarized below:

```
NETWORK-COVERING(W, W')
begin
    if (r is not a power of s) then
        return (W does not cover W');
    endif
    for i = 1 to k - 2 do
        REGROUP-BAND(i);
    endfor
    Let W" be the regrouped W';
    EQUIVALENCE(W, W"); /* the algorithm of [14] */
    if W is equivalent to W" then
        return(W covers W');
    else
        return(W does not covers W');
    endif
end
```

Time Complexity: The two major steps of the covering algorithm are REGROUP-BAND and EQUIVALENCE procedures. The first procedure iterates $\frac{N}{r}$ times. To compute the time of each iteration, note that computing the set F takes $1 + s + s^2 + ... + s^{l-1} = \frac{s^l - 1}{s-1} = O(s^{l-1})$ time. Similarly, $E_{i \times l}$ takes $O(s^{l-1})$. The derivation of E_j from E_{j-1} takes $O(s|E_{j-1}|) = O(s \times \frac{r}{s}) = O(r)$. Thus, each iteration takes $O(s^{l-1}) + O(s^{l-1}) + O(l \times r) = O(l \times r)$. Consequently, REGROUP-BAND(i) takes $O(\frac{N}{r} \times l \times r) = O(l \times N)$. It follows that the overall regrouping of W' takes $O(k \times l \times N) = O(k' \times N) = O(N \log_s N)$. The EQUIVALENCE procedure was shown in [14] to take $O(N \log_s N)$ time as well. Therefore, the network covering algorithm takes $O(N \log_s N)$ time. This algorithm is then optimal up to a constant factor because W' has $\frac{N}{s} \log_s N$ switches and $N \log_s N + N$ inter-column links which have to be "looked at" at least once before W can be decided to cover W'.

A Summary of the EQUIVALENCE Algorithm:

Due to the unique path property of MIN's, each switch x of every column i in a network W in MIN(s, k') can be uniquely identified by a couple (a, b), where a is the minimum input terminal label reachable from x, and b is the minimum output terminal label reachable from x. This unique switch identification is represented by the triplet (x, a, b). Define by $T_i(W)$ the following set:
$T_i(W) = \{(x, a, b) \mid x$ is a switch in column i of W, a (resp. b) is the smallest input (resp. output) terminal reachable from $x\}$.

Observe that if two networks W and W' in MIN(s, k') are strictly functionally equivalent (and hence strictly topologically equivalent), and if (a, b) identifies a switch x in column i of W, then (a, b) must identify a switch y in column i of W'. Consequently, for all $i = 0, 1, 2..., k' - 1$, $T_i(W)$ can be derived from $T_i(W')$ by permuting the first compo-

nents of the triplets of $T_i(W')$. The following equivalence algorithm can then be easily shown to decide equivalence correctly:

EQUIVALENCE(W, W')
begin
1. **for** $i = 0$ **to** $k' - 1$ **do**
   compute $T_i(W)$ and $T_i(W')$;
2. **for** $i = 0$ **to** $k' - 1$ **do**
   determine if $T_i(W)$ can be derived from $T_i(W')$ by permuting the first components of the triplets of $T_i(W')$ by some permutation. If this can be done, permute (or equivalently, relabel) the switches of column i of W' by the same permutation; else, answer "no" and halt;
3. Let W'' be the relabeled W' after step 2. Replace the terminals and the switches of W and W'' by nodes, and let the resulting graphs be denoted by $G(W)$ and $G(W'')$, respectively. If $G(W) = G(W'')$, then W and W' are strictly functionally equivalent, else they are not; (**Comment**: Two graphs are equal if their sets of nodes are equal and so are their sets of edges)
end

The details of the implementation and time complexity of the 3 steps of the EQUIVALENCE algorithm can be found in [14].

The network covering algorithm decides strict functional covering but not wide functional covering. As mentioned earlier, wide functional covering is difficult to decide and will be left for future work. However, wide functional covering relations in the context of the interesting class of digit permutation networks turned out to yield to mathematical analysis. The next two sections will investigate covering relations among existing MIN's and among digit permutation networks.

§4. Covering Relations Among Existing Networks

An $N \times N$ omega network, denoted $\Omega(r,k)$, has k columns of $r \times r$ switches, where $N = r^k$. The leftmost interconnection as well as the inter-column interconnections of $\Omega(r,k)$ are all the shuffle interconnection S defined in the system of base r as follows: $S(x_{k-1}x_{k-2}...x_1x_0) = x_{k-2}...x_1x_0x_{k-1}$, where $x_{k-1}...x_1x_0$ is an arbitrary k-digit r-ary label.

Using Theorems 2.5 and 2.6, it can be concluded that if an $N \times N$ $\Omega(r,k)$ functionally covers in the strict or wide sense another $N \times N$ $\Omega(s,k')$, then r must be a power of s. Conversely, it will be shown that if r is a power of s, then $\Omega(r,k)$ functionally covers $\Omega(s,k')$ in the strict sense. We say then that the omega network grows inclusively as its switch size grows to a power.

In [6], Ω-realizable permutations are characterized as follows: A permutation f is realizable in $\Omega(r,k)$ if and only if

$(\forall s, s')(\forall l = 0,1,...,k-1)([s \neq s'$ and $d_{k-1}...d_{l+1} = d'_{k-1}...d'_{l+1}] \Rightarrow s_l s_{l-1}...s_0 \neq s'_l s'_{l-1}...s'_0)$

where $s = s_{k-1}...s_1s_0$ (in the system of base r), $s' = s'_{k-1}...s'_1s'_0$, $d = f(s) = d_{k-1}...d_1d_0$, and $d' = f(s') = d'_{k-1}...d'_1d'_0$.

Using this characterization, it can be proved that if r is a power of s, then every permutation realizable by $\Omega(s,k')$ is also realizable by $\Omega(r,k)$.

It follows that also omega inverse $\Omega^{-1}(r,k)$ functionally covers $\Omega^{-1}(s,k')$ in the strict sense if and only if r is a power of s.

Following the same line of reasoning, it can be shown that the generalizations of the indirect binary n-cube [8], the generalized cube network [10], and the baseline network [12] all grow inclusively as their switch size grows to a power.

§5. Covering Relations among Digit Permutation Networks

One common feature in the definitions of the existing multistage interconnection networks is that the interconnections between columns are bit permutations. The well-known shuffle interconnection is an example. Among the reasons for using these permutations as interconnections are their regularity, rich structure and ease of analysis. Therefore, the MIN's that have as inter-column interconnections bit permutations or, in the general case where the switches are $r \times r$, digit permutations that permute digits of r-ary labels, are of special interest.

Formally speaking, a permutation f of $S_N = \{0,1,...,N-1\}$, where $N = r^k$, is a ***digit permutation*** in the system of base r if there exists a permutation π of $S_k = \{0,1,...,k-1\}$ such that $f(x_{k-1}...x_1x_0) = x_{\pi(k-1)}...x_{\pi(1)}x_{\pi(0)}$, where $x_{k-1}...x_1x_0$ is an arbitrary k-digit r-ary label. An $N \times N$ digit permutation network (DPN) in MIN(r,k) is a MIN whose interconnections are digit permutations of S_N in the system of base r.

These digit permutation networks have been studied in [13] and [15]. Among other things, it was shown in [13] and [15] that all DPN in MIN(r,k) are widely functionally equivalent. It follows that if W is an $N \times N$ DPN in MIN(r,k) and W' is another $N \times N$ DPN in MIN(s,k'), then W is widely functionally equivalent to $\Omega(r,k)$ and W' is widely functionally equivalent to $\Omega(s,k')$. Therefore, W functionally covers W' in the wide sense if and only if $\Omega(r,k)$ functionally covers $\Omega(s,k')$ (in the strict or wide sense), that is, if and only if r is a power of s.

Furthermore, an optimal terminal relabeling algorithm was given in [13] and [15] that would relabel the termi-

gals of any $N \times N$ DPN N_1 in MIN(r,k) to simulate another $N \times N$ N_2 DPN in MIN(r,k). That is, the algorithm finds two permutations f and g of S_N such that fN_1g is strictly functionally equivalent to N_2. The algorithm takes $O(N \log_r N)$ time. By calling this algorithm twice, first on $\Omega(r,k)$ and W, and then on $\Omega(s,k')$ and W', we find four permutations f, g, f' and g' of S_N such that $\Omega(r,k)$ and $\Omega(s,k')$ are strictly functionally equivalent to fWg and $f'W'g'$, respectively. If r is a power of s, $\Omega(r,k)$ strictly covers $\Omega(s,k')$, and therefore, fWg strictly covers $f'W'g'$. Consequently, $f'^{-1}fWgg'^{-1}$ strictly covers W'. That is, W simulates W' by relabeling the input terminals of W by $f^{-1}f'$, and the output terminals of W by gg'^{-1}. Clearly, the time complexity to find the new labels is $O(N \log_r N) + O(N \log_s N) + O(N)$, where the last term is the time to invert f and g', and also to compose $f^{-1}f'$ and gg'^{-1}. Therefore, the overall time complexity is $O(N \log_s N)$, which is optimal for the same reason cited for the optimality of the covering algorithm.

6. Conclusions

In this paper we have studied functional covering and topological covering relations and the equivalence between these two types of relations among banyan multistage interconnection networks. An optimal algorithm to decide strict functional covering was also given. Most existing networks were shown to grow inclusively as their switch size grows to a power. Finally, digit permutation networks were shown to functionally cover in the wide sense any other digit permutation networks of the same terminal size if the switch size of the former is a perfect power of the switch size of the latter. We also described an algorithm to relabel the terminals of a covering digit permutation network in order to simulate a covered digit permutation network. Thus, parallel systems that are interconnected with arbitrary digit permutation networks are upwardly compatible.

Future work includes (1) the development of efficient wide covering algorithms and (2) the study of functional and topological relations among networks that differ in terminal size as well as in switch size. These latter relations account better for upward system compabitibity as the interconnecting networks grow in terminal size and switch size.

7. References

[1] D. P. Agrawal, "Graph Theoretical Analysis and Design of Multistage Interconnection networks," *IEEE Trans. Comput.*, C-32, pp. 636-648, July 1983.

[2] D. P. Agrawal and J. -S. Leu, "Dynamic Accessibility Testing and Path Length Optimization of Multistage Interconnection Networks," *IEEE Trans. Comput.*, C-34, pp. 255–266, Mar. 1985.

[3] D. P. Agrawal, S. -C Kim and N. K. Swain, "Analysis and Design of Nonequivalent Networks" *IEEE Trans. Comput.*, Vol. 37, pp. 233–237, Feb. 1988.

[4] R. Conterno and R. Melen, "An Analytical Model for a Class of Processor-Memory Interconnection Network," *IEEE Trans. Comput.*, Vol. C-36, pp. 1374–1378, Nov. 1987.

[5] T. Feng, "A Survey of Interconnection Networks," *Computer*, Vol. 14, pp. 12–27, Dec. 1981.

[6] D. K. Lawrie, "Access and Alignment of Data in an Arrary Processor," *IEEE Trans. Comput.*, C-24, pp. 1145–1155, Dec. 1975.

[7] A. Y. Oruc and M. Y. Oruc, "On Testing Isomorphism of Permutation Networks," *IEEE Trans. Comput.*, C-34, pp. 958–962, Oct. 1985.

[8] M. C. Pease, "The Indirect Binary n-Cube Multiprocessor Array," *IEEE Trans. Comput.*, C-26, pp. 458–473, May 1976.

[9] H. J. Siegel and S. Smith, "Study of Multistage Interconnection Networks," *Proc. Fifth Annual Symp. Comp. Arch.*, pp. 223–229, Apr. 1978.

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

[11] T. H. Szymanski and V. C. Hamacher, "On the permutation Capibility of multistage interconnection networks," *IEEE Trans. Comput.*, C-36, pp. 810–822, July 1987.

[12] C.L. Wu and T.Y. Feng, "On a Class of Multistage Interconnection Networks," *IEEE Trans. Comput.*, vol. C-29, No. 8, pp. 694–702, August 1980.

[13] A. Youssef, *Properties of Multistage Interconnection Networks,* Ph.D. dissertation, Princeton University, Feb. 1988.

[14] A. Youssef and B. Arden, "Equivalence Between Functionality and Topology of Log N-Stage Banyan Networks," *IEEE Trans. Comput.*, vol. 39, No. 6, pp. 829–832, June 1990.

[15] A. Youssef and B. Arden, "Structure of Efficiently Controllable Banyan Multistage Interconnection Networks," sumbitted to *The Second IEEE Symposium on Parallel and Distributed Processing*, Dallas, Texas, Dec. 1990.