Topo logy of Efficiently Controllable Banyan Multistage Networks

Abdou Youssef
The George Washington University
Washington, DC 20052

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

Abstract

Due to their unique path property, banyan multistage interconnection networks (MIN's) can be self-routed using control tags. This paper introduces a number of routing control classes of MIN's and studies their structure. These include the D-controllable networks where the control tags are the destination labels, the FD-controllable networks where the control tags are function of the destination labels, and the doubly D- or FD-controllable networks which are D- or FD-controllable forward and backward. The paper shows that all D- and FD-controllable networks have a recursive structure, and that all doubly D-controllable (resp., FD-controllable) networks are strictly (resp., widely) functionally equivalent to the baseline network. The subclass of MIN's where the interconnections are digit permute is also studied and shown to be doubly FD-controllable and hence equivalent to the baseline. Finally, the paper presents an efficient, parallel algorithm that relabels the terminals of any one network to simulate any other network in that subclass.

1. Introduction

Banyan multistage interconnection networks (MIN's) are increasingly important in parallel computing systems. Several networks of this type have been proposed and studied, such as omega and its inverse [4], the indirect binary n-cube [7], the baseline [10], and the generalized cube network [9].

The efficiency of MIN's is critical to overall system performance, and depends on the speed of the routing control, among other things. As these networks have the unique path property, that is, each source has a unique path to each destination, they can be self-routed via control tags. The control efficiency depends then on the speed of control tag computation. If the control tags are stored, the resulting memory cost is prohibitive for large systems. Therefore, the networks whose control tags are efficiently computable and need not be stored are of special interest. The most efficiently controllable networks are clearly those whose control tags are the destination addresses. The second most efficient are those whose control tags are simple functions of the destination tags.

Although multistage networks have received a lot of attention [1-2],[5],[6], little work has been done on control categorization of MIN's. Such categorization and the understanding of the relationship between control and structure of networks are important for the design of efficiently controllable networks. Furthermore, as will be seen later, further insight into functionality and network equivalence is gained from the study of the control-structure relationship. Particularly, one of the contributions is tying together and superseding the different approaches and results in [8], [9] and [10] related to the existing multistage interconnection networks, their underlying structure, their control and their equivalence to one another.

In this paper various control schemes or classes are introduced and the structure of the networks in each class is studied. These control classes include D-controllable networks where the control tags are simply the destination tags; FD-controllable networks where the control tags are functions of the destination tags; doubly D-controllable networks which are D-controllable from input to output and from output to input; and the doubly FD-controllable networks which are FD-controllable from input to output and from output to input. The last two control classes include all existing MIN's and are useful for two-way communication needed in shared-memory systems.

The paper focuses also on the sub-class of banyan multistage networks where the interconnections between columns are bit permutations (or digit permutations in general), which are operations that permute bits in a specified manner. These networks are called digit permutation networks. The reason for studying this subclass is twofold. First, it includes all existing banyan multistage networks. Second, it turns out that all digit permutation networks are doubly D- or FD-controllable networks and therefore share the same underlying structure as the latter networks.

The main contributions of this paper are the following. First, the topological structure of the D-controllable and FD-controllable networks is determined and shown to be recursive. Second, it is shown that all doubly D-controllable (resp., FD-controllable) networks are strictly (resp., widely) functionally equivalent to the baseline network. This allows the baseline network to simulate any doubly FD-controllable network by relabeling the input and output terminals of the baseline. Third, the paper establishes necessary and sufficient conditions for a sequence of digit permutations to construct a digit permutation network with the unique path property. An optimal algorithm is developed to decide if a sequence of digit permutations
meets the aforementioned conditions. Fourth, all digit permutation networks are shown to be doubly FD-controllable and hence widely functionally equivalent to the baseline network. In addition, the control tags are shown to be digit permutations of destination tags, thus allowing for very fast routing control. Finally, an efficient, parallel algorithm that relabels the input and output terminals of one digit permutation network in order to simulate another digit permutation network is given. Such simulation is possible when two networks are widely functionally equivalent.

The paper is organized as follows. The next section gives some preliminary definitions and fundamental concepts related to network control and equivalence. Section 3 explores the structure of D-controllable networks. FD-controllable networks are treated in section 4. The functional equivalence of doubly D-controllable and doubly FD-controllable networks is established in section 5. Section 6 studies digit permutation networks. Conclusions and future directions are given in section 7.

§2. Definitions and Fundamental Concepts

In this section banyan multistage interconnection networks are specified, functional and topological relations among them are reviewed, their routing control is discussed and various control classes are defined.

2.1 Banyan Multistage Networks

Banyan multistage interconnection networks have N input terminals, N output terminals, and k interconnected columns of $\frac{N}{r} \times r$ crossbar switches, where $N = r^k$ and $r \geq 2$. Each $r \times r$ crossbar switch realizes all $r^2$ permutations. The interconnection between every two successive columns is a permutation of $S_N = \{0, 1, ..., N - 1\}$. The interconnection from the input terminals to the leftmost column is called the left-end interconnection, and that from the rightmost column to the output is called the right-end interconnection. The connectivity of these networks is such that between every input terminal i and every output terminal j there is only one and only one path, denoted $i \rightarrow j$. The class of these networks is denoted MIN(r, k). Omega and its inverse [4], the indirect binary n-cube [7], and the baseline network [10] are examples of such networks where $r = 2$. Fig. 1 shows a baseline network in MIN(2,3).

For ease of reference, the input (i.e., left) and output (i.e., right) terminals of networks in MIN(r, k) are labeled 0, 1, ..., $N - 1$ from top to bottom. The input and output ports of each column are similarly labeled 0, 1, ..., $N - 1$. The ports are also labeled locally relative to each switch: The ports (input or output) of each switch are labeled 0, 1, ..., $r - 1$, from top to bottom. The distinction between the two labels will be clear from the context. 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. The bases of terminals and column ports are often represented in r-ary, each label having $k \times r$ digits. In this context, the local label of a switch port is represented by a single r-ary digit.

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 by $f(j)$, 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 $f^{-1}$.

It should be noted that the composition of functions is taken from left to right, that is, $(x)fg = g(f(x))$. If P(W) denotes the set of permutations realizable by W, then $P(gWf) = \{ghf | h \in P(W)\}$, which clearly follows from the definition of $gWf$ and the left-to-right view of composition.

Finally, a network W in MIN(r, k) is said to be left bare-ended if its left-end interconnection is the identity permutation. It is said to be right bare-ended if the right-end interconnection is the identity permutation. A network is bare-ended if it is both left and right bare-ended.

2.2 Network Equivalence Relations

Two networks W and W' in MIN(r, k) are strictly functionally equivalent (denoted $W \equiv W'$) if they realize the same permutations. The two networks are widely functionally equivalent if they can be made to realize the same permutations by relabelling 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 $W \equiv gWf$.

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 has been shown in [12] that two networks in $MIN(r,k)$ are strictly topologically equivalent if and only if they are strictly functionally equivalent. Similarly, they are widely topologically equivalent if and only if they are widely functionally equivalent. For that reason, we will often drop the terms topological and functional and speak merely of equivalence, strict or wide.

2.3 Control of Banyan Multistage Interconnection Networks

Due to the unique path property, the networks of $MIN(r,k)$ can be self-routed using control tags (CT). Specifically, since there is a unique path between any input terminal $i$ and any output terminal $j$ in a network $W$ in $MIN(r,k)$, there exists a unique $r$-ary tag $c_{i-1}c_{i-2}...c_0$ which can be used to establish the path $i \rightarrow j$. To show this, let $s_0, s_1, ..., s_{k-1}$ be the consecutive switches through which the path $i \rightarrow j$ goes. The unique link between switch $s_l$ and switch $s_{l+1}$ that lies on the path $i \rightarrow j$ leaves switch $s_l$ through some output port of local label $l$. Let $c_l = b_{l-1} - 1$ for all $l = 0, 1, ..., k - 1$. Now that $c_{k-1}c_{k-2}...c_0$ is defined, it can be used as a control tag as follows. Input terminal $i$ sends $c_{k-1}c_{k-2}...c_0$ through the control lines of $W$, and column $l$ uses the $r$-ary digit $c_{l-1} - 1$ to link the input port of switch $s_l$ to which the control signal comes, to output port $c_{l-1} - 1$. Clearly then, switch $s_0$ uses digit $c_{k-1}$, $s_1$ uses digit $c_{k-2}$ and so on, establishing the path to output terminal $j$.

The uniqueness of the path between $i$ and $j$ implies the uniqueness of the tag $c_{k-1}c_{k-2}...c_0$. This tag is called the control tag, denoted $CT(W,i,j)$, or just $CT(i,j)$ when no confusion arises.

It is clear that the control tag $CT(W,i,j)$ for the path $i \rightarrow j$ depends on the structure of $W$. Storing the control tags $CT(W,i,j)$'s requires $N^2k \log r$ bits which is prohibitive for large $N$. Therefore, it is preferable to have networks for which the $CT(i,j)$'s are simple to compute and need not be stored, like for instance when $CT(i,j) = j$, or when $CT(i,j) = f(j)$ for some easy-to-compute $f$.

A network $W$ in $MIN(r,k)$ is $D$-controllable (also called delta network) if $CT(W,i,j) = j$ (in $r$-ary) for all $i$ and $j$. The baseline and omega networks are $D$-controllable networks. A network $W$ in $MIN(r,k)$ is FD-controllable (also called bidental network) if there exists a permutation $f$ of $S_r$ such that $CT(W,i,j) = f(j)$ for all $i$ and $j$. In this case, $f$ is called the control function of $W$. Omega inverse network is FD-controllable with control function $\rho$ where $\rho(x_{-1}x_{-2}...x_0) = x_0x_{-2}x_{-4}...x_{-j}$ [4]. Note that in general, the control tags may be a function of both the input terminals and the output terminals.

In shared memory systems, where the input terminals represent processors and the output terminals memory modules, efficient communication is needed from left to right and from right to left. A network $W$ in $MIN(r,k)$ is doubly $D$-controllable if it is $D$-controllable from left to right, (i.e., processor to memory) and from right to left (i.e., memory to processor). In particular, to go from output terminal $j$ to input terminal $i$, the control tag needed is $i$. Similarly, a network is doubly FD-controllable if it is FD-controllable from left to right and from right to left. Double controllability can be better understood with the help of inverse networks. The inverse of network $W$, denoted $W^{-1}$, is the mirror image of $W$, where the input terminals of $W$ are the output terminals of $W^{-1}$ and vice versa. A network $W$ is doubly $D$-controllable (resp., doubly FD-controllable) if both $W$ and $W^{-1}$ are $D$-controllable (resp., FD-controllable).

§3. Structure of D-Controllable Networks

In this section a subclass of MIN's, called Generalized Recursive Networks (GRN), will be defined recursively. Then, it will be shown that every network in GRN is D-controllable and conversely. That is, the structure of D-controllable networks is the same as GRN structure.

3.1 Definition. The class of generalized recursive networks $GRN(r,k)$ is a subclass of $MIN(r,k)$, defined recursively as follows. The networks of $GRN(r,1)$ are mere $r \times r$ switches with left interconnections. For $k > 1$, $GRN(r,k)$ is the class of networks of the form $W = f(T(W_0,W_1,...,W_{r-1}))$ (as shown in Figure 2 for $r = 2$) where each $W_i$'s is a network in $GRN(r,k-1)$ such that the input terminals as well as the output terminals of $W_i$ are labeled $r^{k-1}, r^{k-1} + 1, ... , (i + 1)r^{k-1} - 1$, $T$ is a connection (i.e., permutation) that links the $i$-th output port of every switch in column $0$ of $W$ to an arbitrary input terminal of $W_i$, for $i = 0, 1, ... , r - 1$, and $f$ is the left-end interconnection.

If $W$ is as above, and if the switches are permuted within columns, the resulting network is considered to be in $GRN(r,k)$ but is said to be in a non-canonical form. The form of networks as in Figure 2 is called canonical.

Note that the control of a GRN network is the same whether the network is in canonical form or not.

The baseline network [10] is an example of a network in $GRN(2,k)$. Particularly, if we denote by $B(2,k)$ the $2^k \times 2^k$ baseline network with $2 \times 2$ switches as building blocks, then $B(2,k) = R(B(2,k - 1),B(2,k - 1))$, where $R$ is the unshuffle. The baseline can be generalized to $B(r,k)$ such that $B(r,1)$ is a mere bare-ended $r \times r$ switch, and $B(r,k) = R(B(r,k - 1),... ,B(r,k - 1))$, where $R$ is the unshuffle of $S_r$ in the system of base $r$, that is, $R(x_{i-1}x_{i-2}...x_0) = x_0x_{i-1}x_{i-2}...x_0$ for every $k$-digit $r$-ary label $x_{i-1}x_{i-2}...x_0$.

The following two theorems will show that $GRN$ is the same as the class of D-controllable networks.

3.2 Theorem. Every network in $GRN(r,k)$ is $D$-controllable.

Proof. Let $W$ be a network in $GRN(r,k)$. We need to
show that the control tag \( j_{k-1}...j_0 \) establishes the path \( i ightarrow j \) in \( W \), where \( j = j_{k-1}...j_0 \) in base \( r \). The proof is by induction on \( k \geq 1 \).

**Base:** \( k = 1 \). In this case \( W \) has only one \( r \times r \) switch. The control tag \( j_0 \) links any input \( i \) to the output \( j_0 \), which is \( j \).

**Induction:** Assume the statement is true for all networks in \( \text{GRN}(r, k-1) \). It will be proved for the network \( W \) in \( \text{GRN}(r, k) \). Let \( W = JT(W_0, W_1, ..., W_{n-1}) \) be a network in \( \text{GRN}(r, k) \), and assume without loss of generality that it is in canonical form. The digit \( j_{k-1} \), used to control column 0, will cause the input terminal \( i \) to link to the \( j_{k-1} \)-th output port of a switch in column 0. By definition of \( \text{GRN} \), port \( j_{k-1} \) is linked to \( T \) to some input \( x \) of \( W_{j_{k-1}} \). As \( j = j_{k-1} r^{k-1} + j_{k-2} j_{k-3} ... j_0 \), it follows that \( j \) is the \( j_{k-2}...j_0 \)-th output terminal of \( W_{j_{k-1}} \). Since \( W_{j_{k-1}} \) is in \( \text{GRN}(r, k-1) \), it follows from the inductive hypothesis that the control tag \( j_{k-2}...j_0 \) establishes the path \( x ightarrow j_{k-2}...j_0 \) in \( W_{j_{k-1}} \). Consequently, the path \( i ightarrow j \), which is \( i ightarrow x ightarrow j \), is established in \( W \).

To establish the converse of Theorem 3.2, the following lemmas are needed.

**3.3 Lemma.** Let \( W \) be a D-controllable network. Then the following statements are true:

(a) For every \( l \leq \frac{n}{r} - 1 \), the output terminals \( r \times l, r \times l + 1, ..., r \times l + r - 1 \) are all linked to a single switch \( s_l \) in the rightmost column. Furthermore, output terminal \( r \times l + t \) is linked to the output port \( t \) of \( s_l \) for \( t = 0, 1, ..., r - 1 \).

(b) The switches of the rightmost column of \( W \) can be permuted (within column) so that \( W \) becomes right bare ended.

**Proof.** (a) Fix \( l = l_{k-1}...l_1 \). Then \( r \times l + t = l_{k-1}...l_t \) and hence all the paths from an arbitrary source to the destinations \( r \times l + t, t = 0, 1, 2, ..., r - 1 \), are identical up to the rightmost column because the \( k - 1 \) leftmost digits of the \( (r \times l + t)'s \) are identical. Therefore, the terminals \( r \times l, r \times l + 1, ..., r \times l + r - 1 \) are all linked to the same switch (say switch \( s_l \)) in the rightmost column. In addition, as the rightmost digit \( t \) of \( r \times l + t \) controls the rightmost column connecting the incoming input port of \( s_l \) to the output port \( t \) of \( s_l \), it follows that output terminal \( r \times l + t \) is linked to the output port \( t \) of \( s_l \).

(b) Follows immediately from (a): Move switch \( s_l \) to position \( i \) for every \( l \).

In the next lemma, the switches of the rightmost \( k-1 \) columns in a D-controllable network \( W \) in \( \text{MIN}(r, k) \) will be partitioned into \( r \) groups, such that each group forms a D-controllable network in \( \text{MIN}(r, k-1) \). To this effect, let \( U_n = \{a_{n-1}...a_0 ... a_{k-1} \mid a_{k-1} = n \} \), for \( n = 0, 1, 2, ..., r - 1 \). Let also \( U_n \) be the set of switches of column \( i \) (\( i \geq 1 \)) that are reachable from some output terminal in \( U_n \).

**3.4 Lemma.** Let \( W \) be a D-controllable network. Then the following statements are true:

(a) For every \( n \neq m \) and for every \( i \), \( U_n \cap U_m = \emptyset \).

(b) For every \( i \) and \( n \), the switches in \( U_n \) are linked forward only to switches in \( U_{n+1} \).

(c) For every \( i \) and \( n \), \( |U_n| = r^{k-2} \).

**Proof.** (a) The proof is by contradiction. Let a switch be a switch in \( U_n \cap U_m \) for some \( n \neq m \). Hence, \( s \) can reach some output terminal \( nd_{k-2}...d_0 \) in \( U_n \) and another \( md_{k-2}...d_0 \) in \( U_m \). There exists some input terminal \( p \) of \( W \) that can reach through \( s \) the output terminals \( nd_{k-2}...d_0 \) and \( md_{k-2}...d_0 \) via the control tags \( nd_{k-2}...d_0 \) and \( md_{k-2}...d_0 \), respectively, because \( W \) is D-controllable. Hence, the two tags must agree in the \( i \) leftmost digits, yielding \( m = n \), and leading to a contradiction.

(b) If a switch \( s \) in \( U_n \) were linked to a switch in \( U_{n+1} \) for some \( m \neq n \), then \( s \) could reach an output terminal in \( U_m \) and another in \( U_{n+1} \). This would lead to the same contradiction as above.

(c) The proof is by backward induction on \( i \).

**Base:** \( i = k - 1 \). Follows immediately from Lemma 3.3.

**Induction:** Assume the statement is true for all values of \( i = k - 1, ..., i+1 \). It will be proved for \( i = l \). By the inductive hypothesis, we have \( |U_{l+1}| = r^{k-2} \). Using (b), it can be concluded that all the outgoing links from the switches in \( U_l \) go to switches in \( U_{l+1} \), and all the incoming links to the switches of \( U_{l+1} \) come from switches in \( U_l \). Therefore, the number of the links between the switches of \( U_n \) and the switches of \( U_{n+1} \) is equal to \( r|U_n| \) on the one hand, and to \( r|U_{n+1}| \) on the other hand. Hence, \( |U_n| = |U_{n+1}| = r^{k-2} \).

**3.5 Theorem.** Every D-controllable network in \( \text{MIN}(r, k) \) is in \( \text{GRN}(r, k) \).
Proof. Let \( W \) be a D-controllable network in \( \text{MIN}(r, k) \). It will be shown by induction on \( k \) that \( W \) is in \( \text{GRN}(r, k) \).

Base case: \( k = 1 \). It is obvious that \( W \) is a single switch that is right bare-ended (after Lemma 3.3-(b)), and hence in \( \text{GRN}(r, 1) \).

Induction: Assume the theorem is true for all values of \( i < k \). It will be proved for \( k = 1 \). By Lemma 3.4-(c), each \( U^* \) has \( r^{-2} \) switches. Permute the switches of the rightmost column so that \( W \) becomes right bare-ended. For every \( i = 1, 2, \ldots, r - 1 \), permute the switches of the \( i \)-th column so that the labels of the switches of \( U_n^* \) are \( n^{-1} - 1, n^{-1} - 2 + 1, \ldots, (n + 1)r^{-2} - 1 \). Since \( U_n^* \cap U^*_m = \emptyset \) for every \( n \neq m \) (Lemma 3.4-(a)), and since the outgoing links from the switches of \( U_n^* \) go only to switches in \( U_m^* \) (Lemma 3.4-(b)), it follows that the subnetworks \( W_n = (U^*_1, U^*_2, \ldots, U^*_r) \), for \( n = 0, 1, 2, \ldots, r - 1 \), are disjoint. Hence, \( W \) can be put in the form \( W = f_T(W_0, W_1, \ldots, W_{r - 1}) \) for some \( f \) and \( T \). It can be seen that each \( W_n \) is D-controllable. By the inductive hypothesis, each \( W_n \) is in \( \text{GRN}(r, i - 1) \). Consequently, \( W \) is in \( \text{GRN}(r, i) \).

Therefore, the topologically defined class \( \text{GRN} \) is identical to the control-characterized class of D-controllable networks. This topological characterization will be used to show that all doubly D-controllable networks are topologically and functionally equivalent. It can also be used to develop an optimal \( O(N \log N) \) algorithm to decide if a MIN is D-controllable [11].

4. Structure of FD-Controllable Networks

The class \( \text{GRN} \) is extended in this section and shown to be identical to the class of FD-controllable networks. Recall that if \( W \) is an FD-controllable network in \( \text{MIN}(r, k) \), it is a permutation \( f \), called the control function, such that the control tag \( CT(i, j) \) establishes the path \( i \rightarrow j \) is \( f(i, j) \).

4.1. Definition. The extended \( \text{GRN} \) is the class of networks of the form \( Wf \) where \( W \) is in \( \text{GRN}(r, k) \) and \( f \) is a permutation of \( S_n \).

The following lemma relates the networks of the extended \( \text{GRN} \) to the FD-controllable networks and their control functions.

4.2 Lemma. (a) If \( W \) is in \( \text{GRN}(r, k) \) and \( f \) is a permutation of \( S_n \), then \( Wf \) is D-controllable and the control tag \( CT(Wf, i, j) \) for the path \( i \rightarrow j \) is \( f^{-1}(j) \).

(b) If \( W \) is FD-controllable with control function \( g \), then \( Wg \) is D-controllable.

(c) If \( W \) is D-controllable, then there exist a network \( W \) in \( \text{GRN}(r, k) \) and a permutation \( f \) such that \( Wf = Wf \).

5. Doubly Controllable Networks

As pointed out earlier, it is of theoretical as well as practical interest to study the networks that are efficiently controllable not just from left terminals to right terminals, but also from right terminals to left terminals.

Recall that by right-to-left D-controllability it is meant that if right terminal \( j \) needs to communicate to left terminal \( i = i_{r - 1}i_{r - 2}\ldots i_0 \) in a network \( W \), then the control \( i_{r - 1}i_{r - 2}\ldots i_0 \) is used to establish the path as follows. The digit \( i_{r - 1} \) controls the switches of column \( k - 1 \) (i.e., the rightmost column), \( i_{r - 2} \) controls the switches of column \( k - 2 \), and so on. Right-to-left FD-controllability is defined similarly, where the control tag \( CT \) (denoted \( CT_{L-R}(i, j) \)) to establish the path from right terminal \( j \) to left terminal \( i \) is \( f(i) \). The permutation \( f \) is then called the right-to-left control function.

So for doubly controllable networks, we make a distinction between the left-to-right control tags \( CT_{L-R} \) (the old sense of \( CT \)) and the right-to-left control tags \( CT_{R-L} \). We also make a distinction between left-to-right control functions and right-to-left control functions, which do not have to be identical.

Note that most existing networks have been shown to be doubly D-controllable or doubly FD-controllable. In this section, this will be generalized. It will be shown that all doubly D-controllable networks are strictly functionally equivalent and all doubly FD-controllable networks are widely functionally equivalent to the baseline.

The proof of the equivalence of the doubly D-controllable networks with \( B(r, k) \) involves the following steps. First, it will be established that if \( W \) is a doubly D-controllable network, then the switches of its leftmost and rightmost columns can be repositioned so that \( W \) becomes bare-ended. Afterwards, it will be shown that the switches in column \( 1 \) can be repositioned so that \( T \) becomes identical to the corresponding interconnection \( R \) in \( B(r, k) \) and all the \( W_i \)'s become doubly D-controllable. Finally, the proof will proceed by induction on \( k \). These steps will be made precise next.

5.1 Lemma. Let \( W \) be a doubly D-controllable network in \( \text{MIN}(r, k) \). Then the following statements are true:

(a) The switches of column 0 and column \( k - 1 \) of \( W \) can be repositioned so that \( W \) becomes bare-ended.

(b) Assume that \( W \) is bare-ended, and in canonical form \( W = T(W_0, W_1, \ldots, W_{r - 1}) \). Then the switches of the leftmost column of each \( W_i \) can be repositioned so that \( T \) becomes identical to the corresponding interconnection \( R \) in the baseline \( B(r, k) \).

(c) Assume \( W \) is as in (b) and \( T = R \). Then \( W_i \) is doubly D-controllable for every \( i = 0, 1, \ldots, r - 1 \).
proof. (a) This follows by applying Lemma 3.3-(b) to \( W^{-1} \).

(b) Consider \( W_i \) for some arbitrary \( l \), and let \( i_1 \ldots i_{l-1} \) be a \((k - 2)\)-digit \( r \)-ary label. Using the double \( D \)-controllability of \( W \), it can be shown that the right ports \((i_{k-1} \ldots i_1)_{0 \leq i_k \leq r-1}\) of column 0 of \( W \) are linked to a single switch (say \( i_{k-1} \ldots i_1 \)) in column 1 of \( W \), such that the right port \((i_{k-1} \ldots i_1)_{0 \leq i_k \leq r-1}\) is linked to the \(l\)-th left port of switch \( s_{i_1} \ldots i_1 \). Now if switch \( s_{i_1} \ldots i_1 \) is moved to position \( i_{k-1} \ldots i_1 \) of column 1 of \( W \), the labels of columns 1 of \( W \) are equal to the unshuffle of the labels \((i_{k-1} \ldots i_1)_{0 \leq i_k \leq r-1}\). All the interconnections from column 0 to column 1 of \( W \) becomes identical with the unshuffle \( R \).

(c) As each \( W_i \) is \( D \)-controllable, it suffices to show that each \( W_i \) is \( D \)-controllable from right to left. Let \( i = i_1 \ldots i_{k-1} \) and \( j \) be an input terminal and an output terminal of \( W_j \), respectively, for some arbitrary \( l \). The equivalent label of \( i \) in \( W \) is \( i' = (i_1 \ldots i_{k-1})_1 \). As \( T = R \), \( i' \) is linked to the right port \( R^{-1}(i') \) of column 0 of \( W \). The path \( R^{-1}(i') - j \) from output terminal \( j \) to input terminal \( R^{-1}(i') \) of \( W \) has to go through \( i' \). As \( W \) is \( D \)-controllable from right to left, the control tag for this path is \( R^{-1}(i') = i_1 \ldots i_{k-1} \). Hence, the sub-tag \( i_1 \ldots i_{k-1} \) establishes the sub-path from \( j \) to \( i' \). Since \( i' \) is the same port as \( i \) and \( i = i_1 \ldots i_{k-1} \), it follows that the control tag for the path from \( j \) to \( i \) in \( W_i \) is \( i \). Consequently, \( W_i \) is \( D \)-controllable from right to left.

5.2 Theorem. All doubly \( D \)-controllable networks in \( \text{MIN}(r, k) \) are strictly equivalent to the baseline network \( B(r, k) \).

Proof. Let \( W \) be a doubly \( D \)-controllable network. Due to the previous lemma, we can assume that \( W = R(W_0, \ldots, W_{r-1}) \) where each \( W_i \) is doubly \( D \)-controllable. As \( B(r, k) = R(B(r, k - 1), \ldots, B(r, k - 1)) \), a simple induction on \( k \) establishes the theorem.

The next lemma will relate doubly \( FD \)-controllable networks, the baseline network and the control functions. The proof is straightforward and hence omitted.

5.3 Lemma. (a) If \( W = g(B(r, k), f) \), then \( CT_{L^{-1}}(W, i, j) = f^{-1}(j) \) and \( CT_{R^{-1}}(W, i, j) = g(i) \).

(b) If \( W \) is doubly \( FD \)-controllable, and if \( CT_{L^{-1}}(W, i, j) = g(j) \) and \( CT_{R^{-1}}(W, i, j) = f(j) \), then \( W \) is strictly topologically equivalent to \( g(B(r, k), f)^{-1} \).

5.4 Theorem. All doubly \( FD \)-controllable networks in \( \text{MIN}(r, k) \) are strictly equivalent to the baseline network \( B(r, k) \).

Proof. Follows from the previous lemma.

Lemma 5.3 has other implications, some of which have been proved elsewhere through other methods [10] about the relations among some of the existing networks. The following theorem redescribes these relations and generalizes them for arbitrary switches sizes \( r \geq 2 \).

5.5 Theorem. \( B^{-1}(r, k) \equiv B(r, k) \) and \( \Omega^{-1}(r, k) \equiv \rho \) if \( r, k \) and \( \Omega^{-1}(r, k) \equiv B(r, k) \rho \) where \( \rho \) is the digit reversal \( (\rho(x_1 \ldots x_1) = x_{r-k}x_{r-k} \ldots x_1) \).

Proof. Since \( B(r, k) \) is doubly \( D \)-controllable, it follows that its inverse is doubly \( D \)-controllable. Consequently, \( B^{-1}(r, k) \equiv B(r, k) \), after Theorem 5.2. \( \Omega^{-1}(r, k) \) is doubly \( FD \)-controllable, its left-to-right control function is the identity permutation and its right-to-left control function is \( \rho \) [4]. After Lemma 5.3-(b), we have \( \Omega^{-1}(r, k) \equiv B(r, k) \rho \). Furthermore, \( \Omega^{-1}(r, k) \equiv (\rho(B(r, k)))^{-1} \equiv B^{-1}(r, k) \rho^{-1} \equiv B(r, k) \rho \) because \( \rho^{-1} = \rho \).

The equivalence among existing \( \text{MIN} \)'s is therefore no coincidence since they are doubly \( D \)- or \( FD \)-controllable. In the remaining part of the paper, the double controllability of these networks will be shown to be a result of their inter-column interconnections, namely bit manipulation permutations, which are operations that permute the bits of binary labels in a specified manner.

§6. Digit Permutation Networks

As was just pointed out, 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. Some of the reasons for using these permutations as interconnections are their regularity, rich structure, and ease of analysis. In this section, the whole class of digit permutation networks will be studied using the concepts of \( D \)-control, \( FD \)-control and double \( FD \)-control, and taking advantage of the equivalence among all doubly \( FD \)-controllable networks.

The approach is algebraic. A relation will be derived relating \( CT(i, j) \) with \( i \) and \( j \). This relation will be used to find necessary and sufficient conditions for \( k + 1 \) digit permutations to construct a \( \text{MIN} \) that has the unique path property. Later the control tags in digit permutation networks are shown to be functions of the destination tags only. This makes them \( FD \)-controllable. Making use of the fact that the inverse of a digit permutation network is a digit permutation network, it will be concluded that digit permutation networks are doubly \( FD \)-controllable and hence widely equivalent to the baseline network.

6.1. Definition. A permutation \( f \) of \( S_N = \{0, 1, \ldots, N-1\} \), where \( N = r^k \), is a digit permutation in the system of base \( r \) if there exists a permutation \( \pi \) of \( S_k = \{0, 1, \ldots, k-1\} \) such that \( f(x_{k-1} \ldots x_1 x_0) = \pi(x_{k-1}) \ldots x_1 x_0 \), where \( x_{k-1} \ldots x_1 x_0 \) is an arbitrary \( k \)-digit \( r \)-ary label. In this case, \( f \) is denoted \( f_\pi \) and \( \pi \) is called the kernel of \( f_\pi \).

6.2. Definition. A digit permutation network, denoted \( DPN(f_0, f_1, \ldots, f_k) \), is a network in \( \text{MIN}(r, k) \) where the leftmost interconnection is \( f_0 \), the rightmost interconnection is \( f_k \), the interconnection from column \( i \) to column \( j \) is \( f_i \) for \( i = 1, \ldots, k - 1 \), and all the \( f_i \)'s are digit permutations of \( S_{r^i} \) in the system of base \( r \).

Denote by \( E_i^r \) where \( a \) is an \( r \)-ary digit and \( i = 0, 1, \ldots, k - 1 \), the following mapping from \( S_N \) to \( S_N \):

\[
E_i^r(x_{k-1} \ldots x_0) = x_{k-1} \ldots x_i x_i a x_{i+1} \ldots x_{k-1} x_0
\]

that is, \( E_i^r \) replaces the \( i \)-th digit by \( a \).
Next, the relation between an arbitrary input terminal \( s \), an arbitrary output terminal \( d \) and the control tag \( c = c_{k-1} c_{k-2} \ldots c_0 \) for the path \( s \rightarrow d \) in a digit permutation network \( \text{DPN}(f_{x_1}, f_{x_2}, \ldots, f_{x_n}) \) will be derived. Recall that the digit \( c_{k-1} \) controls column \( i \) for \( i = 0, 1, \ldots, k-1 \). Note that if the path \( s \rightarrow d \) enters column \( i \) through input port \( x_{i+1}, \ldots, x_0 \), it exits that column through the output port \( x_{i+1}, \ldots, x_{k-1} \), which is equal to \( E_{c_{k-1}}(x_{k-1}, \ldots, x_0) \). Note also that if the path exits column \( i \) through some output port \( y \), it then enters the next column, that is, column \( i \), through input port \( f_x(y) \) because the interconnection between column \( i-1 \) and column \( i \) is \( f_x \). We have thus proved the following lemma:

**6.3 Lemma.** In a digit permutation network \( \text{DPN}(f_{x_1}, f_{x_2}, \ldots, f_{x_n}) \) an arbitrary output terminal \( d \) is related to an arbitrary input terminal \( s \) and the control tag \( c = c_{k-1} c_{k-2} \ldots c_0 \) for the path \( s \rightarrow d \) by the following relation:

\[
d = (s)f_x E_{c_{k-1}}(x_{k-1}, \ldots, x_0) E_{c_{k-2}}(x_{k-2}, \ldots, x_0) \ldots E_{c_0}(x_0) f_x.
\]

The \( E \)'s will be “filtered” out to the right of the \( f \)'s in the relation above. To that effect, the following lemma is needed. The proof is straightforward and hence omitted.

**6.4 Lemma.** \( f_x f_y = f_y f_x \) and \( E_{\gamma} f_x = f_x E_{\gamma}(-1) \).

**6.5 Lemma.** Under the assumptions of Lemma 6.3 we have \( d = (s)f_x E_{c_{k-1}}(x_{k-1}, \ldots, x_0) E_{c_{k-2}}(x_{k-2}, \ldots, x_0) \ldots E_{c_0}(x_0) f_x \) where \( \beta_i = x_{i+1} \ldots x_0, \ldots, x_i \).

**Proof.** Let \( g = f_x E_{\gamma_{c_{k-1}}}(-1) E_{\gamma_{c_{k-2}}}(-1) \ldots E_{\gamma_{c_0}}(-1) \) where \( \gamma_{c_0}, \gamma_{c_1}, \ldots, \gamma_{c_k} \) are pairwise distinct.

By making repeated use of Lemma 6.4 on the expression of \( g \) (from right to left) we have

\[
g = f_x E_{\gamma_{c_{k-1}}}(-1) E_{\gamma_{c_{k-2}}}(-1) \ldots E_{\gamma_{c_0}}(-1) \ldots
\]

As \( d = g(s) \) (from Lemma 6.3), the lemma follows.

The necessary and sufficient conditions as well as the relation between the control tag and the output terminal can now be easily derived as follows.

**6.6 Theorem.** Let \( f_{x_1}, f_{x_2}, \ldots, f_{x_n} \) be \( k + 1 \) digit permutations, and \( \beta_i = x_{i+1} \ldots x_0, \ldots, x_i \)

(a) The digit permutation network \( \text{DPN}(f_{x_1}, f_{x_2}, \ldots, f_{x_n}) \) is in \( \mathbb{M} \) if and only if \( \beta_i \), \( \beta_i' \), \( \beta_j \), \( \beta_j' \) are pairwise distinct.

(b) The control tag \( c = c_{k-1} c_{k-2} \ldots c_0 \) for a path \( s \rightarrow d \) in a digit permutation network \( \text{DPN}(f_{x_1}, f_{x_2}, \ldots, f_{x_n}) \) which has the unique path property is \( c = f_x(d) \), where \( \gamma(i) = \beta_i' \).

**Proof.** (a) Let \( s \) be the input terminal, \( d \) the output terminal and \( c = c_{k-1} c_{k-2} \ldots c_0 \) the control tag that establishes the path \( s \rightarrow d \). Let \( s' = f_x(s) \). Using the previous lemma, we have

\[
d = (s') E_{\gamma_{c_{k-1}}}(-1) E_{\gamma_{c_{k-2}}}(-1) \ldots E_{\gamma_{c_0}}(-1)
\]

It can be easily seen that the effect of each \( E_{\gamma_{c_i}}(-1) \) is to replace the digit in position \( \beta_i' \) of \( s' \) by \( c_{k-1} \).

Assume first that the network has the unique path property. If \( \beta_i' \), \( \beta_i' \), \( \beta_j' \), \( \beta_j' \) are pairwise distinct, then \( \{ \beta_i' \}, \{ \beta_i' \}, \ldots, \{ \beta_j' \} \) is a proper subset of \{0, 1, ..., k - 1\}, and therefore, there exists some \( j \in \{0, 1, ..., k - 1\} \) such that \( j \neq \beta_i' \) for all \( i \). Consequently, the digits in the \( j \)-th digit position of \( s' \) and \( d \) must always agree. It follows that for a fixed \( s \), and thus \( s' \), no matter what control tag we use, we can never reach any output terminal \( d \) whose \( j \)-th digit differs from that of \( s' \). This contradicts the unique path property.

Conversely, if \( \beta_i' = 0 \), \( \beta_j' = 0 \), \ldots, \( \beta_k' = 0 \) are pairwise distinct, then

\[
\{ \beta_i' \}, \{ \beta_j' \}, \ldots, \{ \beta_k' \} = \{0, 1, ..., k - 1\},
\]

and therefore the mapping \( \gamma \) where \( \gamma(i) = \beta_i' \) is a permutation of \{0, 1, ..., k - 1\}. Furthermore,

\[
d = (s') E_{\gamma_{c_{k-1}}}(-1) E_{\gamma_{c_{k-2}}}(-1) \ldots E_{\gamma_{c_0}}(-1)
\]

implying that the digit in position \( \gamma(i) \) of \( d \) is \( c_i \), that is, \( d_{\gamma(k-1)} d_{\gamma(k-2)} \ldots d_{\gamma(0)} = c_{k-1} c_{k-2} \ldots c_0 \). Therefore, \( c = f_x(d) \) and \( d = f_x(c) \). As \( f_x \) is a permutation (a digit permutation) of \( \mathbb{N} \), it follows that for a fixed input terminal \( s \) there corresponds to every control tag \( c \) one and only one output terminal. Therefore, the network has the unique path property.

(b) The relation \( c = f_x(d) \) has just been proved in (b).

Part (a) of the previous theorem leads to a simple algorithm to determine if a sequence of \( k + 1 \) digit permutations construct a digit permutation network that has the unique path property. The algorithm takes as input a sequence of \( k + 1 \) kernels \( \pi_0, \pi_1, \ldots, \pi_k \), computes the \( \beta_i \)'s (noting that \( \beta_i = \pi_i x_{i+1} \ldots x_0 \), and hence \( \beta_i' = x_{i+1} \ldots x_0 \)), then computes the mapping \( \gamma \) such that \( \gamma(i) = \beta_i' \) as defined in the previous theorem. Finally it checks if \( \gamma \) is a permutation of \{0, 1, ..., k - 1\} by doing a bucket sort on \( \gamma(0), \gamma(1), \ldots, \gamma(k - 1) \) and checking if any of the “buckets” 0, 1, ..., \( k - 1 \) is empty. If no bucket is empty, then \( \gamma \) is a permutation, that is, the network has the unique path property; otherwise, the unique path does not have the unique path property. The time complexity can be easily seen to be \( O(k^2) = O(\log^2 N) \).

6.1 Control of Digit Permutation Networks

Part (b) of the last theorem shows that digit permutation networks are FD-controllable and that their control functions are digit permutations \( f_x \) which can be easily derived from the constituent digit permutations.

One consequence of the fact that the control function is a digit permutation \( f_x \) is the increased control efficiency. One way of controlling a DPN with control function \( f_x \) is to design the switch so that the destination labels can be used to set the switches as follows: the switches in column \( i \) use digit \( \gamma(k - i + 1) \) of the destination label as control digit, for \( i = 0, 1, \ldots, k - 1 \). Another way is to provide some additional hardware in the input terminals to permute the digits of the output terminals according to \( \gamma \) and produce the control tag. As \( k \) is relatively small in practice, the additional amount of hardware is small. A third way is to compute \( f_x(d) \) in software every time a path \( s \rightarrow d \) is to be established. This requires a small amount of memory to store \( \gamma \) (not \( f_x \)) at every input terminal. This software
6.3 Equivalence of Digit Permutation Networks

As \((f_e)^{-1} = f_{e^{-1}}\), it follows that the inverse of a DPN is a DPN. Thus all DPN’s in \(MIN(r, k)\) are doubly FD-controllable and hence widely equivalent to the baseline \(B(r, k)\), by Theorem 5.2. This result is a generalization of the equivalence among the existing networks (in [10]).

6.4 Simulation Among DPN’s

If two networks in \(MIN(r, k)\) are widely equivalent, then the terminals of one can be relabeled so that it realizes the permutations of the other. The problem is how to relabel the terminals.

We have shown in Lemma 5.3 that if a network \(W\) in \(MIN(r, k)\) is doubly FD-controllable with left-to-right control function \(f\) and right-to-left control function \(g\), then \(W\) is strictly equivalent to \(gB(r, k)f^{-1}\). Hence, the baseline network can simulate \(W\) by relabeling the input terminals of \(B(r, k)\) by \(g^{-1}\) and the output terminals by \(f^{-1}\). Generally, if \(W’\) is another doubly FD-controllable network in \(MIN(r, k)\) with left-to-right control function \(f’\) and right-to-left control function \(g’\), then \(W’\) can be strictly equivalent to \(g’B(r, k)f’^{-1}\), and consequently, \(W’\) is strictly equivalent to \(g’g^{-1}Wff^{-1}\). Therefore, \(W\) can simulate \(W’\) by relabeling the input terminals of \(W\) by \(g^{-1}\) and the output terminals of \(W\) by \(f^{-1}\). Therefore, the problem of relabeling the terminals of one network to simulate another network reduces to finding the left-to-right and right-to-left control functions of both networks. As the control function of digit permutation networks are digit permutations \(f_\gamma\)'s, it is enough to find the \(\gamma\)'s of the control functions.

The following algorithm takes as input two digit permutation networks \(W\) and \(W’\) represented by the sequences of their defining kernels \(x_0, x_4\) and \(x_0, x_4\), respectively, and relabels the terminals of \(W\) to simulate \(W’\). It computes \(\gamma\) and \(\tau\) such that \(f_\gamma\) and \(f_\tau\) are the left-to-right and right-to-left control function of \(W\), respectively. It also computes the corresponding \(\gamma’\) and \(\tau’\) of \(W’\). As the input terminals of \(W\) have to be relabeled with \(f_\gamma f_\tau^{-1}\), which is equal to \(f_\gamma f_{\tau^{-1}}\), the algorithm computes \(\gamma’\). Finally, the algorithm does the relabeling. Note that \(f_\gamma\) is the left-to-right control function of \(W’ = DPN(f_{x_0}^{-1}, f_{x_1}^{-1}, ..., f_{x_4}^{-1})\). Therefore, \(\gamma’\) is computed in the same way as \(\gamma\). The same applies to \(\tau’\).

Procedure Simulate \((W, W’]\)

1. Compute \(\gamma, \tau, \gamma’, \tau’\) and \(\gamma’^{-1} \gamma\) and \(\tau^{-1} \tau\).
2. Broadcast \(\gamma’^{-1} \gamma\) to all outputs and \(\tau^{-1} \tau\) to all inputs;
3. for \(i = 0\) to \(N - 1\) do in parallel
4. relabel input terminal \(i\) of \(W\) by \(i\) \(f_{\tau^{-1}}^{-1}\);
5. relabel output terminal \(i\) of \(W\) by \(i\) \(f_{\tau^{-1}}^{-1}\);

Time Complexity: Step 1 takes \(O(k^2)\). Steps 2, 4 and 5 take \(O(k)\) each. Thus, the procedure takes \(O(\log^2 N)\).

§7. Conclusions

This paper examined several control classes of banyan networks, namely, D-control, FD-control, double D-control and double FD-control, and showed that the first two classes have a recursive structure and the last two are equivalent to the baseline. The digit permutation networks, where the interconnections are bit (digit)-permutes, were also studied and shown to be doubly FD-controllable and hence equivalent to the baseline, thus generalizing the results about the equivalence among existing networks. An optimal algorithm to simulate any DPN by any other DPN was also presented. Future work will examine the structure and functionality of MIN’s whose control tags are easy-to-compute functions of both source and destination tags.

§8. References