Reconfiguration Algorithms for Interconnection Networks

A. YAVUZ ORUC, M. YAMAN ORUC, AND NORMAN BALABANIAN

Abstract — The correspondence examines the functional relations within a class of multistage interconnection networks. It is known that these networks are not rearrangeable. This fact has led to some research on interconnection network relations. The correspondence deals with one aspect of this research, namely, that of constructing an equivalence map between two interconnection networks. Procedures are given to test whether two such networks are equivalent. Whenever they are, these procedures also produce a map to conjugate one network onto the other.

Index Terms — Conjugation map, cycle map, functional equivalence, interconnection network, permutation map.

I. INTRODUCTION

A large set of interconnection networks has been proposed for use in parallel processing systems [2]–[7], [12], [13], [15]–[22]. It is known that most of these networks are not rearrangeable. This fact has led to some studies on interconnection network relations. Siegel and Smith [16] examined and demonstrated some functional relations among a number of networks. Siegel [17] later constructed explicit maps to provide a simulation environment within a class of interconnection networks. Wu and Feng [20], [21] further examined the relations among existing multistage interconnection networks and introduced the notion of topological and functional equivalence. Both types of equivalence have been useful in classifying interconnection networks, although the functional equivalence is a more practical notion as it amounts to just renaming the terminal nodes of a network without disturbing its internal switching structure.

The research efforts reported in [16], [17], [20], [21] all made significant contributions to the understanding of equivalence relations among interconnection networks. Despite the progress being made, however, the problem of testing and/or constructing equivalences between two networks consisting of switches other than 2×2 crossbars has so far remained open. The solution of this problem for networks whose switches are programmable for identity and cycle maps constitutes the focus of the present work.

The correspondence is organized as follows. An interconnection network model and network relations are reviewed in Sections II and III. In Section IV, the reconfiguration of single-stage networks is discussed in view of conjugation maps. In Section V, the reconfiguration of multistage networks is considered, and a procedure for constructing equivalence maps between functionally equivalent networks is developed. The correspondence is concluded with Section VI.

II. NETWORK MODEL

We begin by reviewing a network model introduced in [8] since much of the discussions in the following sections rely on this model. Throughout this section and the following ones, some elementary facts about permutation maps and groups are assumed.

An interconnection network (IN) (refer to Fig. 1) is defined in [8] as the five tuples (S, D, M, F, g) where 1) S(D) is a set of source (destination) nodes, respectively, 2) M is a set of control variables, 3) F is a set of symbols, each of which takes map values from a set

Manuscript received September 15, 1984; revised March 1, 1985.

A.Y. Oruc is with the Department of Electrical, Computer and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180.

M. Y. Oruc is with the Department of Mathematics, Syracuse University, Syracuse, NY 13210.

N. Balabanian is with the Department of Electrical and Computer Engineering, Syracuse University, Syracuse, NY 13210.





 $Q_i \subseteq S$ to another set $R_i \subseteq D$ such that $Q_i \cap Q_j = \Phi, R_i \cap R_j = \Phi$ whenever $i \neq j$ and $S = \bigcup_{i=1}^k Q_i, D = \bigcup_{i=1}^k R_i$ for some integer k, and 4) g is a surjection from F to M, called the control function.

For convenience, the elements of S(D) will be identified as integers $1, 2, \dots, S(D)$. Moreover, for the scope of this paper, it is assumed that each $f_i: Q_i \to R_i$ is a permutation map, and hence we must have $|Q_i| = |R_i|$ for all $i; 1 \le i \le k$. We shall represent permutations by a cycle notation [14]. For example, if $Q_i = \{1, 2, 3, 4, 5\} = R_i$, we write $f_i = (123)(45)$ to mean (1) $f_i = 2$, (2) $f_i = 3$, (3) $f_i = 1$, (4) $f_i = 5$, and (5) $f_i = 4$. In particular, if $f_i = (1)(2)(3)(4)(5)$, i.e., f_i is the identity map from Q_i to R_i , then we write $f_i = e$. The composition of two permutations u and v, denoted uv, is defined for each $s \in S$ as (s)uv = ((s)u)v. Thus, if u = (123), v = (134), then uv = (124)(3) = (124).

Intuitively speaking, each $f_i \in F$ identifies a switch of IN whose set of source (destination) nodes is $Q_i(R_i)$; $1 \le i \le k$. The control function g partitions F into disjoint subsets and it assigns to each subset a unique control variable $m \in M$. The control variable m is defined over a finite set of integers and for each integer value of m, the switch f_i controlled by m (i.e., $(f_i)g = m$) assumes a permutation map $p_{i,j}: Q_i \to R_i$. The role of the control function was described in detail in [8], [9]. For the scope of this paper and without loss of generality, we shall be concerned with only three of the five tuples, namely, S, D, and F. Based on this assumption, we define the following two types of network structure.

Definition 1: A single-stage network IN = (S, D, F) is a collection of k sets of permutation maps $P(f_i) = \{p_{i,j}: p_{i,j}: Q_i \to R_i, 1 \le j \le n_i\}$ where $n_i \le |Q_i|$!; $1 \le i \le k$ such that every interconnection p of IN is in the form $p = \prod_{i=1}^{k} p_{i,j}$ for some $p_{i,j} \in P(f_i)$; $1 \le i \le k$.

We shall denote the set of all interconnections of IN by P(F). Clearly, $P(F) = \prod_{i=1}^{k} P(f_i)$ where the product of two sets of permutations $P(f_1)$ and $P(f_2)$ is defined as $P(f_1) \cdot P(f_2) = \{p_{1,j} \cdot p_{2,j'}: 1 \le j \le n_1, 1 \le j' \le n_2\}$, and it has a straightforward extension to k such sets.

Definition 2: An *n*-input/*r*-stage network, IN = (S, D, F), hereafter called an (n, r) network, is a cascade of *r* single-stage networks IN_i = (S_i, D_i, F_i) ; $1 \le i \le r$ such that $|S_i| = n = |D_i|$ for all *i*; $1 \le i \le r$, (2) $S_{i+1} := D_i$; $1 \le i \le r - 1$, and (3) every interconnection *p* of IN is in the form $p = \prod_{i=1}^r p_i$ whenever $p_i \in P(F_i)$; $1 \le i \le r$.

We note that $S_{i+1} := D_i$; $1 \le i \le r - 1$ is included in the definition to emphasize the physical connections between successive stages, and it has no significance otherwise. We shall denote the set of all interconnections of IN by P(F). Clearly, $P(F) = \prod_{i=1}^r P(F_i)$.

As an example, let IN be an (8, 2) network with stages IN₁, IN₂ where $P(F_1) = \{e, (12) (345) (678)\}, P(F_2) = \{e, (13) (578)\}$. Then

0018-9340/85/0800-0773\$01.00 © 1985 IEEE

 $P(F) = \{e, (12) (345) (678), (13) (578), (123475) (68)\}.$ One of the interconnections of IN, i.e., (123475) (68) is shown in Fig. 2. The numbers in parentheses indicate how the source nodes are mapped to the destination nodes.

III. RELATIONS AMONG NETWORKS

This section reviews such concepts as covering, containment, and equivalence relations for interconnection networks. A more detailed discussion of these relations can be found in [10].

Let $p \in \text{Sym}(S)$ where Sym(S) is the set of all permutations over S and called the symmetric group over set S. The *conjugate* of p with respect to a map $h \in \text{Sym}(S)$ is defined as $h^{-1}ph$. Similarly, the *conjugate set* V of a set of permutations $U \subseteq \text{Sym}(S)$ with respect to the map h is defined as $V = h^{-1}Uh = \{h^{-1}uh : u \in U\}$. The notion of conjugation of a set of permutations is linked quite naturally to functional relations between two interconnection networks. The following two results from algebra [14] form the basis of this assertion.

Lemma 1: Let $p, h \in \text{Sym}(S)$. Then $h^{-1}ph$ has the same cycle structure as p, and it is obtained by applying h to symbols in p.

For example, if h = (256)(143) and p = (13)(247), then $h^{-1}ph = ((1)h(3)h)((2)h(4)h(7)h) = (41)(537)$. It is seen that both p and $h^{-1}ph$ are products of a 2-cycle and a 3-cycle, and thus they have the same cycle structure.

Lemma 2: p and q are conjugate in Sym(S); that is, $p = h^{-1}qh$ for some $h \in \text{Sym}(S)$ if and only if they have the same cycle structure.

Now, let IN = (S, D, F) be an (n, r) network with stages IN_i = (S_i, D_i, F_i) ; $1 \le i \le r$. Clearly, $h^{-1}P(F)h = h^{-1}(\prod_{i=1}^r P(F_i))h = \prod_{i=1}^r h^{-1}P(F_i)h$. In other words, conjugating the set of permutations of IN is equivalent to first conjugating the set of permutations of each stage and then forming the ordered composition of the conjugated sets. Moreover, by Lemma 1, the conjugation of the set of permutations of IN_i; $1 \le i \le r$ by map h is equivalent to renaming the terminal nodes of IN_i by h. Since the same map, i.e., the map h, is employed in all the conjugations, we conclude that $h^{-1}P(F)h$ amounts to renaming the terminal nodes of IN by h without altering any of the interconnections between its stages.

The foregoing discussion provides the basis for the following definitions. Let $IN_u = (S, D, F_u)$ and $IN_v = (S, D, F_v)$ be (n, r_u) and (n, r_v) networks, respectively.

Definition 3: IN_u is said to cover IN_v if for every $v \in P(F_v)$, $v = h^{-1}uh$ for some $u \in P(F_u)$ and $h \in \text{Sym}(S)$.

Definition 4: IN_u is said to contain IN_v if $P(F_v) \subseteq h^{-1}P(F_u)h$ for some $h \in \text{Sym}(S)$. If h is the identity map in Sym(S), then IN_v is said to be strictly contained in IN_u.

Definition 5: IN_u and IN_v are said to be functionally equivalent or conjugate networks if $P(F_u) = h^{-1}P(F_v)h$ for some $h \in$ Sym(S). If h is the identity map in Sym(S), then the two networks are said to be strictly functionally equivalent.

The above definitions express the degrees of freedom in interconnection network relations. As an example, consider the networks shown in Fig. 3. First, note that $(23)^{-1}P(F_u)(23) = P(F_v)$, and hence IN_u and IN_v are functionally equivalent. Moreover, it is seen that $eP(F_u)e \supseteq P(F_w)$ and $(23)P(F_v)(23) \supseteq P(F_w)$. Therefore, we conclude that both IN_u and IN_v contain IN_w where the containment relation between IN_u and IN_w is a strict one. Finally, note that for each $u \in P(F_u)$, there exists a map $w \in P(F_w)$ such that u and whave the same cycle structure, and hence, by Lemma 1, $h^{-1}uh =$ w for some $h \in Sym(\{1, 2, 3, 4\})$. Therefore, IN_w covers IN_u . It can similarly be shown that IN_w also covers IN_v .

IV. RECONFIGURATION OF SINGLE-STAGE NETWORKS

The preceding two sections have established the algebraic formalism underlying the functional relations among interconnection networks. We now turn our attention to the reconfiguration of one



Fig. 2. An (8, 2) network realizing (123475)(68).







IN



Fig. 3. Functionally related (4, 2) networks.

network as another whenever the two networks are functionally equivalent. That is, given two networks $IN_u = (S, D, F_u)$ and $IN_v = (S, D, F_v)$, we shall consider the problem of constructing a conjugation map h such that $P(F_v) = h^{-1}P(F_u)h$ whenever $IN_u =$ IN_v , i.e., the two networks are functionally equivalent. We shall design algorithms which will either lead to such a map h or otherwise indicate that $P(F_v) \neq h^{-1}P(F_u)h$ for all $h \in Sym(S)$. Covering and containment relations among interconnection networks imply somewhat weaker conditions, and they can be handled once we know how to deal with the conjugation relations.

As for the type of networks, we shall restrict our attention to those networks whose switches are programmable for cycle and identity maps over their terminal nodes. Although the networks which conform to this condition form a small subset of all interconnection networks, the main objective of the current work is to demonstrate the methods which we use to construct equivalence maps among such networks. A more general treatment of the subject will be deferred to another place. We begin with single-stage networks.

Definition 6: Let $u \in \text{Sym}(S)$. The set $M(u) = \{s : (s)u \neq s; s \in S\}$ is called the *moving set* of u in S.

Lemma 3: Let $u = (a_1, a_2, \dots, a_i)$ and $v = (b_1, b_2, \dots, b_j)$ be two cycle maps in Sym(S). If $M(u) \cap M(v) = \Phi$ then $M(h^{-1}uh) \cap$ $M(h^{-1}vh) = \Phi$ for all $h \in$ Sym(S).

Proof: By Lemma 1, $h^{-1}uh = ((a_1)h(a_2)h\cdots(a_i)h)$ and $h^{-1}vh = ((b_1)h(b_2)h\cdots(b_j)h)$. Now suppose that $M(u) \cap M(v) = \Phi$, but $M(h^{-1}uh) \cap M(h^{-1}vh) \neq \Phi$. Then for some s, t; $1 \le s \le i$ and $1 \le t \le j$, $(a_s)h = (b_t)h$, which in turn implies that $a_s = b_t$, contradicting the assumption that $M(u) \cap M(v) = \Phi$.

Theorem 1: Let $IN_u = (S, D, F_u)$ and $(IN_v = (S, D, F_v))$ be an (n, 1) network with switches $f_{u_j}(f_{v_j})$; $1 \le j \le k$. Suppose that $P(f_{u_j}) = \{e, u_j\}$ and $P(f_{v_j}) = \{e, v_j\}$; $1 \le j \le k$ where $u_j(v_j)$ is a cycle map defined over the terminal nodes of switch $f_{u_j}(f_{v_j})$; $1 \le j \le k$. Then $IN_u \approx IN_v$ if and only if $h^{-1}(\prod_{j=1}^k u_j)h = \prod_{j=1}^k v_j$ for some $h \in Sym(S)$.

Proof: Clearly, $h^{-1}(\prod_{j=1}^{k} u_j)h = \prod_{j=1}^{k} h^{-1} u_jh$. Thus, if $h^{-1}(\prod_{j=1}^{k} u_j)h = \prod_{j=1}^{k} v_j$, then $\prod_{j=1}^{k} h^{-1} u_j h = \prod_{j=1}^{k} v_j$. Now, since $M(u_j) \cap M(u_j) = \Phi$ whenever $j \neq j'$, by Lemma 3, $M(h^{-1}u_jh) \cap M(h^{-1}u_jh) = \Phi$ whenever $j \neq j'$. Hence, $\prod_{j=1}^{k} h^{-1}u_jh$ is a product of disjoint cycles where the cycles are $h^{-1}u_jh$; $1 \leq j \leq k$. On the other hand, $\prod_{j=1}^{k} v_j$ is also a product of disjoint cycles, and therefore, by Lemma 1, $h^{-1}u_jh = v_{(j)w}$; $1 \leq j \leq k$ for some $w \in \text{Sym}(\{1, 2, \cdots, k\})$. It follows that $h^{-1}P(F_u)h = h^{-1}(\prod_{j=1}^{k} \{e, u_j\})h = \prod_{j=1}^{k} \{e, n^{-1}u_jh\} = \prod_{j=1}^{k} \{e, v_{(j)w}\} = P(F_v)$. Hence, $\text{IN}_u \approx \text{IN}_v$. To prove the converse, suppose that $h^{-1}P(F_u)h = \prod_{j=1}^{s} v_j$, for some $s; 1 \leq s < k$ and, by Lemma 1, $\prod_{j=1}^{k} u_j$ and $\prod_{j=1}^{s} v_j$, must have the same cycle structure. However, this is not possible since s < k and the cycles of both products are pairwise disjoint. We conclude that $h^{-1}(\prod_{j=1}^{k} u_j)h = \prod_{j=1}^{k} v_j$, and the assertion follows.

The above theorem reduces the problem of finding a map h satisfying the set equality $h^{-1}P(Fu)h = P(Fv)$ to one of determining that map through the scalar equality $h^{-1}(\prod_{j=1}^{k} u_j)h = \prod_{j=1}^{k} v_j$. Thus, all that remains to be done is to obtain h from the latter equality.

that remains to be done is to obtain h from the latter equality. Now, since by Lemma 1 $\prod_{j=1}^{k} u_j$ and $\prod_{j=1}^{k} v_j$ have the same cycle structure, and since $M(u_j) \cap M(u_{j'}) = \Phi$ whenever $j \neq j'$, we must map u_j ; $1 \leq j \leq k$ to some $v_{j'}$; $1 \leq j' \leq k$ such that $h^{-1}u_jh = v_{j'}$. Thus, we must first test to see if there exist k pairs of maps $u_i, v_{i'}$ such that $h^{-1}u_{i}h = v_{i'}$. If not, by Theorem 1 we conclude that $IN_u \neq IN_v$. Otherwise, we construct h as follows. Let $u_j =$ $(a_1a_2\cdots a_t)$ and $v_{j'} = (b_1b_2\cdots b_t)$. Then $h^{-1}u_jh = ((a_1)h(a_2)h$ $(a_t)h = (b_1b_2\cdots b_t)$. So if we let $(a_1)h = b_1$, then we must have $(a_2)h = b_2, \cdots, (a_t)h = b_t$. By repeating this construction for all other pairs u_j , $v_{j'}$ we can determine the map h over all of the set S. We remark that the matching of $(a_1)h$ to b_1 in the above construction is arbitrary and $(a_1)h = b_2$, or $(a_1)h = b_3, \cdots, (a_1)h = b_t$ will all lead to a conjugation map. From this observation, it follows that the conjugation map between IN_u and IN_v is not unique. An exact count of such maps, as well as the complexity of the above procedure, was given in [11], and a formal algorithm for constructing the map happeared in [1]. As an example, let

$$P(f_{u_1}) = \{e, (123)\}, P(f_{u_2}) = \{e, (4567)\}, P(f_{u_3}) = \{e, (89)\},$$

$$P(f_{v_1}) = \{e, (3456)\}, P(f_{v_2}) = \{e, (178)\}, P(f_{v_3}) = \{e, (29)\}.$$

We match (123) with (178), (4567) with (3456), and (89) with (29), and obtain the map h = (2765438).

V. RECONFIGURATION OF MULTISTAGE NETWORKS

In the preceding section, we described a procedure for constructing equivalence maps between two (n, 1) networks. This procedure exploited the fact stated in Theorem 1. The following result will serve a similar purpose for multistage networks.

Theorem 2: Let $IN_u = (S, D, F_u)$ and $IN_v = (S, D, F_v)$ be (n, r)

networks with stages IN_{u_i} and IN_{v_i} ; $1 \le i \le r$. Also let $\{e, u_{i,j}\}$. $(\{e, v_{i,j}\})$; $1 \le j \le k_i$; $1 \le i \le r$ be the set of permutations of the *j*th switch of stage *i* in $IN_{\mu}(IN_{\nu})$ where k_i is the number of switches in stage *i*. Then, if $h^{-1}(\prod_{j=1}^{k_i} u_{i,j})h = \prod_{j=1}^{k_i} v_{i,j}$; $1 \le i \le r$ for some $h \in Sym(S)$, then $IN_u \simeq IN_{\nu}$.

Proof: Let $u = \prod_{i=1}^{r} u_i$ for some $u_i \in P(F_{u_i})$; $1 \le i \le r$. Since $h^{-1}(\prod_{j=1}^{k_i} u_{i,j})h = (\prod_{j=1}^{r} v_{i,j})$ by Theorem 1, $h^{-1}u_ih \in P(F_{v_i})$, and hence, $\prod_{i=i}^{r} h^{-1}u_ih = h^{-1}(\prod_{i=1}^{r} u_i)h = h^{-1}uh \in \prod_{i=1}^{r} P(F_{v_i})$ or $h^{-1}uh \in P(F_v)$. Thus, $h^{-1}P(F_u)h \subseteq P(F_v)$. It is similarly shown that $h^{-1}P(F_u)h \supseteq P(F_v)$, and hence the assertion follows. \Box

Although Theorem 2 is not as strong as Theorem 1 insofar as completely reducing the set equality $h^{-1}P(F_u)h = P(F_v)$ to the set of expressions $h^{-1}(\prod_{j=1}^{k_i} u_{i,j})h = \prod_{j=1}^{k_i} v_{i,j}; 1 \le i \le r$, it does allow one to construct functional equivalences between two (n, r) networks directly from the latter set of relations. Thus, all that remains to be done is to find a procedure to obtain h from these relations. We adopt the following approach.

Denote $\prod_{i=1}^{k_i} u_{i,j}$ by u_i and $\prod_{j=1}^{k_i} v_{i,j}$ by v_i ; $1 \le i \le r$. We need to construct h such that $h^{-1}u_ih = v_i$; $1 \le i \le r$. Now suppose $(s)u_i = s_i$; $1 \le i \le r$ and (s)h = s' where $s, s', s_i \in S$. Then it is easily verified that $(s_i)h = s'_i$ where $s'_i = (s')v_i$; $1 \le i \le r$. In other words, whenever we map $s \in S$ to $s' \in S$ by h, we must map the image of s under u_i to the image of s' under v_i ; $1 \le i \le r$. This observation forms the basis of the following conjugation map construction procedure.

Choose a pair of symbols (s, s') and form the pairs $((s)u_i, (s')v_i)$; $1 \le i \le r$. If among all the pairs, including (s, s'), there exist at least two pairs $((s)u_{i_1}, (s')v_{i_1})$ and $((s)u_{i_2}, (s')v_{i_2})$ such that either $(s)u_{i_1} = (s)u_{i_2}$ or $(s')v_{i_1} = (s')v_{i_2}$, but not both, then return to the first step as (s, s') cannot lead to a conjugation map. Else, test if every $s \in S$ appears at least once as the first entry in some pair among all the pairs constructed. If so, then stop and read the map hfrom these pairs. Otherwise, repeat the same procedure for each pair $((s)u_i, (s')v_i)$; $1 \le i \le r$; that is, form the pairs $((s)u_i \cdot u_i, (s')v_i \cdot u_i)$ v_j for all $i, j; 1 \le i \le r; 1 \le j \le r$ and test to see if these pairs, when combined with the ones formed earlier, satisfy the conditions above. This process is repeated until *either* every $s \in S$ appears at least once as the first entry in some pair and every two pairs are either disjoint or they are identical or among all the pairs constructed thus far there exist at least two pairs such that exactly one of the entries of one pair is identical to the corresponding entry of the other pair.

A formal and more complete treatment of the above procedure appears in [11]. Here, we illustrate it further with the following example. Let

$u_1 = (123) (45) (678),$	$v_1 = (154)(26)(378),$
$u_2 = (1234) (56) (78),$	$v_2 = (1254)(36)(78),$
$u_3 = (15)(26)(37)(48),$	$v_3 = (17)(28)(34)(56)$.

In order to mechanize the above procedure, we employ a tree structure introduced in [11]. Such a tree for the above pairs of permutations is shown in Fig. 4. The root of the tree corresponds to the pair (s, s'), which, in this case, is (1, 5). Accordingly, the successor nodes of (1, 5) correspond to pairs $((1)u_1, (5)v_1)$, $((1)u_2, (5)v_2), ((1)u_3, (5)v_3)$, which are (2, 4), (2, 4), and (5, 6), respectively. Note that the successor node in the middle, i.e., (2, 4) is terminated with a check mark underneath it. This clearly stems from the fact that (2, 4) appears once more at the same level and only one needs to be expanded. The entire tree is grown on a breadth-first basis by finding the successors of each node, i.e., by computing the images of 1(5) under various compositions of $u_i(v_i)$; $1 \le i \le r$. For example, the second leftmost node in fourth level, i.e., the node with pair (4, 2), is obtained by computing $((1)u_1 \cdot u_1 \cdot u_2)$, $(5)v_1 \cdot v_1 \cdot v_2$). Since all the nodes of the tree consist of pairs which are pairwise identical or disjoint, and since every $s \in \{1, 2, \dots, 8\}$ appears at least once in the tree, the networks IN_{u} and IN_{v} represented by the permutations $u_i, v_i; 1 \le i \le 3$ are conjugate, and the map h is read from the pairs in the tree as h = (1563)(24).





We remark that not each pair of symbols (s, s') chosen from $S = \{1, 2, \dots, 8\}$ will lead to a tree from which a conjugation map h can be constructed. Some pairs may result in successor nodes with conflicting entries, thereby indicating that h cannot be formed through these pairs. On the other hand, some others may not lead to any such conflicting entries, but they will not contain all of the symbols of S and hence will yield only a part of h. In such cases, several trees will have to be constructed before all of h can be determined. These and other aspects of the conjugation map construction procedure are discussed in more detail in [11].

VI. CONCLUSIONS

The correspondence has dealt with the reconfiguration of interconnection networks. The reconfigurability of an interconnection network as another one is shown to be closely linked to functional relations between the two networks. Various forms of such relations have been explored, and it has been shown that most of these derive from conjugation maps between the sets of interconnections of the networks in question. Based on these facts, two procedures have been given to reconfigure a network as another in its equivalence class. These procedures are more general than the one which appeared in [10] as they impose no restriction on the control scheme of the networks to which they apply. Their only limitation stems from the types of switches used in the networks to which they apply. Thus, extension of these procedures to networks with arbitrary switches will be a good topic for further research.

REFERENCES

- [1] J. B. Attili, "On the construction of conjugation maps for functionally equivalent interconnection networks," Masters thesis, Rensselaer Polytechnic Institute, Troy, NY, Sept. 1984. K. E. Batcher, "The flip network in STARAN," in Proc. 1976 Int. Conf.
- [2]
- Parallel Processing, Aug. 1976, pp. 65–71.
 [3] L. N. Bhuyan and D. P. Agrawal, "Design and performance of generalized interconnection networks," *IEEE Trans. Comput.*, vol. C-32, pp. 1081-1090, Dec. 1983.
- T. Feng, "A survey of interconnection networks," Comput., vol. 14, [4] No. 12, pp. 12–27, Dec. 1981. —, "Data manipulating functions in parallel processors and their im-
- [5] plementations," IEEE Trans. Comput., vol. C-23, pp. 309-318, Mar. 1074
- [6] T. Lang and H. S. Stone, "A shuffle-exchange network with simplified control," IEEE Trans. Comput., vol. C-25, pp. 55-65, Jan. 1976.
- D.K. Lawrie, "Access and alignment of data in an array processor," [7] IEEE Trans. Comput., vol. C-24, pp. 1145-1155, Dec. 1975.
- [8] Y.A. Oruc, "Interconnection networks: Group theoretic modeling," Ph.D. dissertation, Syracuse University, Syracuse, NY, Apr. 1983.
- [9] -, "A classification of cube-connected networks with a simple control scheme," *IEEE Trans. Comput.*, vol. C-33, pp. 769-772, Aug. 1984. Y. A. Oruc and M. Y. Oruc, "Equivalence relations among inter-
- [10] connection networks," J. Parallel Distrib. Comput., pp. 30-49, Feb. 1985.
- -, "An algorithm for testing and constructing equivalence maps [11] among permutation networks," in Proc. 1985 Conf. Inform. Sci. Syst., to appear.
- Y. A. Oruc and D. Prakash, "Routing algorithms for cellular interconnec-[12] tion arrays," IEEE Trans. Comput., vol. C-33, pp. 939-942, Oct. 1984.
- [13] M.C. Pease, "The indirect binary N-cube multiprocessor array," IEEE Trans. Comput., vol. C-26, pp. 458-473, May 1976.
- [14] J.J. Rotman, The Theory of Groups: An Introduction. Boston, MA: Allyn and Bacon, 1965.
- [15] H.J. Siegel et al., "A survey of interconnection methods for reconfigurable parallel processing systems," in Proc. Nat. Comput. Conf., 1979, pp. 529-542
- [16] H.J. Siegel and D.S. Smith, "Study of multistage interconnection networks," in Proc. 5th Annu. Symp. Comput. Architecture, Apr. 1978, pp. 223-229.
- [17] H.J. Siegel, "A model of SIMD machines and a comparison of various interconnection networks," IEEE Trans. Comput., vol. C-28, pp. 907-917, Dec. 1979.
- [18] H. S. Stone, "Parallel processing with the perfect shuffle," IEEE Trans. Comput., vol. C-20, pp. 153-161, Feb. 1971.
- [19] K. J. Thurber, "Interconnection networks - A survey and assessment," AFIPS Conf. Proc., vol. 43, 1974, pp. 909-919.
- [20] C. W. Wu and T. Feng, "On a class of multistage interconnection net-
- works," IEEE Trans. Comput., vol. C-29, pp. 694-702, Aug. 1980. —, "The reverse-exchange interconnection network," IEEE Trans. [21] Comput., vol. C-29, pp. 801-811, Sept. 1980.
- [22] -, "The universality of shuffle-exchange network," IEEE Trans. Comput., vol. C-30, pp. 324-332, May 1981.