Mapping algorithms for permutation networks*

Vipin Chaudhary

Parallel and Distributed Computing Laboratory
Department of Electrical and Computer Engineering
Wayne State University, Detroit, MI 48202

Abstract – The mapping problem arises when the dependency structure of a parallel algorithm differs from the interconnection of processors in the intended parallel computer (topological variation) or, when the number of processes required by the algorithm exceeds the number of processors available in the computer (cardinality variation). The problem discussed in this paper is to identify a distributed computing environment that best optimizes the objective function for the given problem. Distinct network permutations may result in equivalent process permutation depending upon the mapping of processes onto the processors. In this paper we study the permuting properties of dynamic interconnection networks taking process mapping into consideration. A uniform group theoretic representation for interconnection networks is developed. Finally, an algorithm to evaluate the number of passes required by an interconnection network to realize a given mapping is presented.

INTRODUCTION

Parallel computers have been employed in a wide variety of applications because they provide excellent speed and cost effectiveness. A fundamental task in the implementation of a parallel algorithm on a parallel computer is the allocation of processes and their dependencies in the algorithm to processors and their interconnections in the given computer [1, 2]. The above problem, however, is extremely difficult to solve and generally intractable. Instead of trying to find a mapping that optimizes certain objective function, we now look at the problem from a different viewpoint. We are given a parallel algorithm and the objective function to be optimized. The problem is to identify a distributed computing environment that best optimizes the objective function for the given problem. Although worded differently, this problem is equivalent to the mapping problem and thus intractable. The potentially dominating effect of interprocessor communication delays has motivated research in the design and analysis of interconnection networks. The permuting properties of several of these networks, and the topological and functional relationships amongst them have been established [8, 3, 10].

Interconnections networks are usually compared by the number of different interconnections they can achieve. These interconnections are greatly influenced by the mapping of processes [7]. In this paper we examine how the mapping of processes affects the permuting properties of interconnection networks. This is better illustrated with an example. Consider an interconnection network that realizes two permutations as shown in Fig. 1(a) and (b). The processes a, b, c, d, and e are mapped onto processors 1, 2, 3, 4, and 5 respectively. The processes a and b communicate in Fig. 1(a), whereas processes a and c communicate in Fig. 1(b). A remapping of processes onto the processors as shown in Fig. 1(c) can realize the permutation of Fig. 1(b) on the interconnection network of Fig. 1(a). Thus, with proper mapping we can achieve the same permutations with networks having fewer interconnection patterns. Also, given a permutation pattern, an algorithm can be executed in fewer routing steps with the optimal mapping [4]. Such interactions between mapping and interconnection networks are discussed in this paper.

*This work was supported in part by NSF Grant MIP-9309489 and Wayne State University Faculty Award

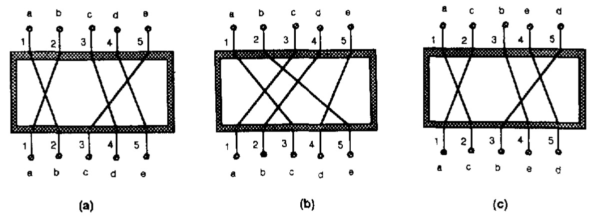

Figure 1: ***An example showing the effect of process mapping in realizing a process permutation.***

The rest of the paper is organized as follows. The following section describes the multiprocessor model where the group theoretic definitions of an interconnection network are described. This is followed with a careful look at the problem we are addressing. The next section uses the group theoretic formulation to address the relationships between interconnection networks. Following this is the section which analyzes various interconnection networks using the tools from the previous section. An algorithm to evaluate the number of passes required by a permutation networks to realize a given mapping is presented. Finally, we conclude with direction for further research for the problems addressed here.

MULTIPROCESSOR SYSTEM MODEL

A multiprocessor system is comprised of M processing elements (PEs) communicating through an interconnection network. The PEs are numbered from 1 to m. The interconnection network is composed of numerous switching elements which connect the PEs. Each input is connected to exactly one output, and each allowable switch setting simultaneously connects several sets of inputs to outputs.

From the above multiprocessor model, the interconnection network IN can be represented as a permutation group (M, G) [5]. M is the set of PEs corresponding to the input and output of the network. For simplicity of notation and without loss of generality, let $M = \{1, 2, \ldots, m\}$. $G \subseteq \mathrm{Sym}(M)$, the symmetric group on M. Let $g \in G$ be a permutation over M. Then, for $i, j \in M$, we write $g(i) = j$ to mean that the network has established an interconnection function g which transfers data from PE i to PE j. The inverses and compositions of the interconnection functions are defined similarly. This definition is extended to sets of interconnection functions. If F and G are two such sets, then $F^{-1} = \{f^{-1} : f \in F\}$ and $F.G = \{f.g : f \in F, g \in G\}$, where $f.g$ is the composition of the interconnection functions f and g.

The interconnection functions will be represented in both cycle and two row matrix form [5]. A cycle of m literals is called an m-cycle, and in particular a 2-cycle is a transposition. Let $\{M_{i,j} : 1 \leq j \leq k_i\}, 1 \leq i \leq n$, be some n partitions of M, and $G_{i,j}$ be the set to functions over $M_{i,j}, 1 \leq j \leq k_i; 1 \leq i \leq n$. Based on the above partition, we now define two types of interconnection networks [6].

Definition 1: An interconnection network $\mathrm{IN}_i = (M, G_i)$, where $G_i = \prod_{j=1}^{k_i} G_{i,j}$, is called a ***single-stage*** network, and $\mathrm{IN}_{i,j} = (M_{i,j}, G_{i,j})$, $1 \leq j \leq k_i$ are called the switches of IN_i.

Definition 2: Let $\mathrm{IN}_i = (M, G_i), 1 \leq i \leq n$ be n single-stage networks. An interconnection network $\mathrm{IN} = (M, G)$, where $G = \prod_{i=1}^{n} G_i$ is called a ***multistage*** network or an (m, n) network, and $\mathrm{IN}_i, 1 \leq i \leq n$ are called the ***stages*** of IN.

Definition 3: Let $\mathrm{IN} = (M, G)$ be a (m, n) network with stages IN_i

$= (M, G_i), 1 \leq i \leq n$. IN is said to be a ***recirculating*** or ***multipass*** network if $(g_1 . g_2 . \dots . g_n)^q$, where $g_i \in G_i, 1 \leq i \leq m$, is an interconnection function of IN for any integer $q \geq 1$.

The set of all interconnection functions of IN $= (M, G)$ in q passes is denoted by G^q. In particular, $G^0 = e$ and $G^1 = G$.

Definition 4: A single-stage network $\text{IN}_i = (M, G_i)$ is ***homogeneous*** if all its switches are of the same size, i.e., $| M_{i,j} |=| M_{i,j'} |, 1 \leq j, j' \leq k_i$. The network is ***strictly homogeneous*** if all its switches are identical, i.e., $G_{i,j} = G_{i,j'}, 1 \leq j, j' \leq k_i$.

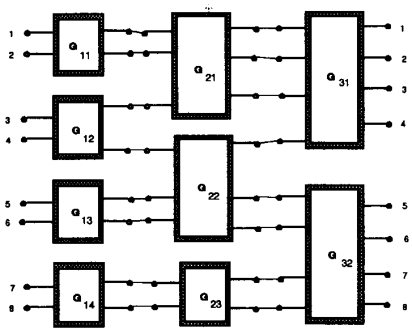

Figure 2: ***An (8,3) interconnection network.***

An (8,3) multistage network is shown in Fig. 2. It has three stages, each of which is a single-stage network. The specification of $G_{i,j}$ determines the set of interconnection functions this network can establish.

$$
\begin{array}{lll}
G_{1,1} = \{e, (12)\} & G_{1,2} = \{e, (34)\} & G_{1,3} = \{e, (56)\} \\
G_{1,4} = \{e, (78)\} & G_{2,1} = \{e, (123)\} & G_{2,2} = \{e, (456)\} \\
G_{2,3} = \{e, (78)\} & G_{3,1} = \{e, (1234)\} & G_{3,2} = \{e, (5678)\}
\end{array}
$$

$$
\begin{aligned}
G_1 = \textstyle\prod_{j=1}^{4} G_{1,j} &= \{e, (12), (34), (56), (78), (12)(56), (12)(78) \\
&\quad (34)(78), (56)(78), (12)(34)(56), (12)(34)(78) \\
&\quad (12)(56)(78), (34)(56)(78), (12)(34)(56)(78)\} \\
G_2 = \textstyle\prod_{j=1}^{3} G_{2,j} &= \{e, (123), (456), (78), (123)(456), (123)(78) \\
&\quad (456)(78), (123)(456)(78)\} \\
G_3 = \textstyle\prod_{j=1}^{2} G_{3,j} &= \{e, (1234), (5678), (1234)(5678)\}
\end{aligned}
$$

Similarly, we can compute $G = \prod_{i=1}^{3} G_i$ which is the set of interconnection functions this network (M, G) realizes.

Developing the problem

On establishing an interconnection function $g \in G$, PE i can transfer data to PE $g(i)$. All the data transfers are assumed to be completed simultaneously in one unit time. This results in ***one pass*** through the network. The set of permutations corresponding to the interconnection functions that a network can establish in one pass will be referred to as the ***fundamental set.*** These terms are applicable to both the single-stage and multistage networks. For example, the fundamental set of the (8, 3) network in Fig. 2 is evaluated as $G = \prod_{i=1}^{3} G_i$.

The communication requirements between processes of a parallel task can be specified by a permutation of the process names. These permutations specifying the logical interconnection pattern to be realized are known as ***process permutations.*** The equivalence between networks [6] and the complexity of simulating one network with another have been studied by establishing functional relationships between the fundamental sets of these networks. In this paper we study the ability of interconnection networks to realize process permutations.

Once the processes have been assigned to the PEs, each member of the fundamental set realizes a permutation. If any member of the fundamental set realizes the process permutation, then the process permutation is realized in one pass, or one unit time. We then say that the parallel task is ***ideally mappable*** on this multiprocessor. More interesting is the case when the process permutation is not ideally mappable. There are two distinct approaches to accomplish this. In the first approach, data is passed through the interconnection network several times till the desired process permutation is realized. During each pass a distinct interconnection function from the fundamental set can be set up in the network. The resulting permutation is the composition of the permutations realized in each pass. This permutation may not belong to the fundamental set. Thus, for a given assignment of processes, new permutations can be realized. The transitive closure of the fundamental set with respect to composition of interconnection functions will be denoted by P, the ***power set*** of the network.

The second approach is to change the assignment of processes onto the PEs. This could result in realizing a distinct permutation which happens to be the process permutation. Note that we may still need more than one pass to realize the required permutation, but would possibly be requiring fewer passes. This one-to-one assignment of processes onto PEs will be referred as ***process mapping.*** Since process mapping is a bijective function, it can be represented as a permutation of process names.

The set of all permutations realizable by the above two approaches along with the fundamental set forms the ***admissible set*** AS of permutations for that network [7]. In contrast to the power set which measures the physical ability of the network to establish interconnection functions, the admissible set is representative of the ability of the network to establish logical communication patterns between processes. Thus, two networks are equivalent if their admissible sets are identical. Establishing functional relationships between the admissible sets of nonequivalent networks will determine how a network can be simulated by another. The cardinality of the admissible set can be taken as a measure of the permutability of the network. This is usually much larger than the cardinality of the fundamental set. The tradeoff is the increased delay due to multiple passes or/and the overhead of mapping. Finally, we would like realize an arbitrary permutation on the network which is not a member of the fundamental set. The number of passes required along with the overhead of mapping to realize this permutation places an upper bound on the worst case performance of a network. Such issues are examined in this paper in the context of group theoretic representation of interconnection networks.

Relation among networks

Consider the affect of process mapping and multiple passes through the network. It is obvious that allowing arbitrarily large number of passes through the network will eventually realize all $M!$ permutations, i.e. $\text{Sym}(M)$ [5]. Consider a process mapping $f \in \text{Sym}(M)$ and an interconnection function $g \in \text{Sym}(M)$. Process $\mathcal{P}$ will be assigned to PE $f(\mathcal{P})$. g will establish a communication path from PE $f(\mathcal{P})$ to PE $g(f(\mathcal{P}))$. Let process $\mathcal{Q}$ also be assigned to PE $g(f(\mathcal{P}))$. Thus, $f(\mathcal{Q}) = g(f(\mathcal{P}))$ or $\mathcal{Q} = f^{-1}gf(\mathcal{P})$. Hence, the process permutation realized by the interconnection function g taking into account the process mapping f, is $f^{-1}gf$, which is conjugate to f.

Definition 5: Let (M, G) be a permutation group. An ***orbit*** of (M, G) is a subset T of M such that $\exists a \in M$ for which $T = aG$.

Definition 6: A permutation group (M, G) is ***transitive*** iff it has only one orbit (namely M). Otherwise (M, G) is ***intransitive.*** Thus, in a transitive interconnection network, there exists an input from which one can go to any output.

Definition 7: Let (M, G) be a transitive permutation group. A ***block*** B of G is a proper subset of M such that (i) $o(B) > 1$, and (ii) if $g \in G$, then either $B = Bg$ or $B \cap Bg = \phi$. These blocks correspond to the switching blocks in an interconnection network.

Lemma 1: If M is finite and B is a block for the transitive permutation group (M, G), then $o(B)$ divides $o(M)$. Thus, the size of the switches is a factor of the size of the network.

Definition 8: A ***primitive*** permutation group is a transitive permutation group without blocks. An ***imprimitive*** permutation group is a transitive group with blocks.

Lemma 2: A 2-transitive group is always primitive.

Lemma 3: If (M, G) is a transitive group of prime degree, then G is primitive.

Definition 9: A ***block system*** of an imprimitive group (M, G) is a set S of blocks such that $M = \cup\{B : B \in S\}$ and such that if $B \in S$ and $g \in G$, then $Bg \in S$.

The structure of imprimitive permutation groups is determined by the next lemma.

Lemma 4: If M is a set, $o(S) > 1, M = \cup\{B : B \in S\}$, and $o(B) = o(B') > 1$ for all B and B' in S, then

1. there is an imprimitive permutation group (M, G) with S as a block system, which contains every other such permutation group as subgroup;
2. $G = \{g \in \mathrm{Sym}(M)$: if $B \in S$ then $Bg \in S\}$;
3. if $o(M) = m$ and $o(B) = k$ are finite for $B \in S$, then $o(G) = (k!)^{m/k}(m/k)!$.

NETWORK ANALYSIS

In this section we analyze a few interconnection networks using the group theoretic representation as discussed in the previous section. The first part states some results for static networks [7], followed by dynamic networks.

Static networks

We will state results for the bidirectional ring.

1) ***Bidirectional ring:*** The bidirectional ring switching element allows two states, i.e., a unit shift in the forward direction and a unit shift in the backward direction. This network has IN $= (M, G)$ has its fundamental set $G = \{f, b\}$, where the interconnection functions f and b are defined as $f : i \to (i + 1) \bmod M$ and $b : i \to (i - 1) \bmod M$, respectively. Larger shifts are realized by a combination of these two unit shifts, i.e., by multiple passes through the network. It is easy to verify the following relationships between the elements of the fundamental set.

- $f^i = b^{M-i}$
- $b^i = f^{M-i}$
- $f^i b^i = e$
- $b^j f^i = f^{i-j}$ if $i > j$
- $b^j f^i = b^{j-i}$ if $i \leq j$
- $f^i b^j = f^{i-j}$ if $i > j$
- $b^j f^i = b^{j-i}$ if $i \leq j$

The power set $P = \{e, f^i; i = 1, \ldots, \lceil M/2 \rceil, b^i; i = 1, \ldots, \lfloor M/2 \rfloor\}$ with $f^M = B^M = e$. For sake of clarity we will assume that M is even, although the results easily extend for M being odd. The interconnection functions $b^i, i = 1, 2, \ldots, M/2$ are equivalent to $f^i, i = M-1, M-2, \ldots, M/2$, respectively.

Theorem 1: The cardinality of the admissible set of bidirectional ring network is given by

$$| AS | = \sum_{d=1}^{M} \frac{M!}{(M/\gcd(M,d))^{\gcd(M,d)} \gcd(M,d)!}$$

over all distinct $\gcd(M, d)$.

Theorem 2: Given an arbitrary permutation $x \in \mathrm{Sym}(M)$, it is admissible iff the cycle representation of x consists of cycles of equal length.

Lemma 5: Any permutation comprising of only one cycle can be realized on a bidirectional ring network in at most four passes, using the interconnection function $f^2 b^2$.

Theorem 3: Given an arbitrary permutation $x \in \mathrm{Sym}(M)$ with a cycle representation $(c_1 c_2 \ldots c_L)$ where the length of each cycle c_i is $l_i, i = 1, 2, \ldots, L$, the number of passes required to realize x on the bidirectional ring network is $\min\{4, \lceil M/l_{min} \rceil\}$, where $l_{min} = \min\{l_i : l_i \geq 2, i = 1, 2, \ldots, L\}$.

Dynamic networks

The fundamental set for a single stage interconnection network $\mathrm{IN}_i = (M, G_i)$, where $G_i = \prod_{j=1}^{k_i} G_{i,j}$, and whose switches are $\mathrm{IN}_{i,j} = (M_{i,j}, G_{i,j}), 1 \leq j \leq k_i$ is G_i.

The fundamental set for a (m, n) interconnection network IN $= (M, G)$, where $G = \prod_{i=1}^{n} G_i$, and whose stages are $\mathrm{IN}_i = (M, G_i), 1 \leq i \leq n$ is G.

The following four theorems can be proved easily and give us the cardinality of the admissible set of the single and multistage interconnection networks. They also give the condition for an arbitrary permutation to be a member of the fundamental set of the networks.

Theorem 4: For a process mapping $p \in \mathrm{Sym}(M)$, the admissible set of a single stage network $\mathrm{IN}_i = (M, G_i)$ is $AS = G_i p = \{gp : \forall g \in G_i\}$, and corresponds to a right coset of G_i in $\mathrm{Sym}(M)$.

Theorem 5: For a process mapping $p \in \mathrm{Sym}(M)$, the admissible set of a (m, n) network IN $= (M, G)$ is $AS = Gp = \{gp : \forall g \in G\}$, and corresponds to a right coset of G in $\mathrm{Sym}(M)$.

Theorem 6: An arbitrary permutation $x \in \mathrm{Sym}(M)$ belongs to the fundamental set of a single stage network $\mathrm{IN}_i = (M, G_i)$ with a process mapping $p \in \mathrm{Sym}(M)$, iff $xp^{-1} \in G_i$.

Theorem 7: An arbitrary permutation $x \in \mathrm{Sym}(M)$ belongs to the fundamental set of a (m, n) network IN $= (M, G)$ with a process mapping $p \in \mathrm{Sym}(M)$, iff $xp^{-1} \in G$.

Most practical networks are transitive. These networks could either be single stage or multistage. Since every stage of the network has several switching elements, these correspond to the blocks of the imprimitive permutation group. Because of lemma 1, it is imperative that the size of the switching elements be a factor of the size of the entire network. If the network is homogeneous, then lemma 4 gives us the order of the fundamental set of the network.

Theorem 8: For an imprimitive homogeneous network (M, G) with m inputs (and outputs) and switching elements of size b, the cardinality of the fundamental set is $o(G) = (k!)^{m/k}(m/k)!$.

The power set of the network is got by composing the fundamental set to itself. It should be apparent to the reader by now that if the network is not homogeneous then the evaluation of the cardinality of the network is not trivial. The evaluation of the permuting capability of homogeneous networks is itself very involved [8]. To evaluate the cardinality of the admissible set or checking to see if an arbitrary permutation is ideally mappable or belongs to the admissible set is extremely hard. We now present polynomial time algorithms [9] to compute the order of the power set of an interconnection network and devise a membership test (to check if an arbitrary permutation is ideally mappable).

We form the power set by using the fundamental set as a generator. If $G \subseteq \mathrm{Sym}(M)$, then the group $< G >$ ***generated*** by G consists of all permutations in $\mathrm{Sym}(M)$ which can be written as a finite product of the permutations in G. Thus, $< G >$ is the power set of G.

In order to arrive at the algorithm for evaluating the order of the power set of G and test for membership of an arbitrary permutation, we first give a few definitions and known results [9].

Definition 10: Let (M, G) be a permutation group, and $Y \subseteq M$. The subgroup $G_{[Y]} = \{g \in G | \forall x \in Y, g(x) = x\}$ is the ***pointwise stabilizer*** of Y in G. Note that if $Z \subseteq Y$ then $G_{[Z]} < G_{[Y]}$, i.e., $G_{[Z]}$ is a subgroup of $G_{[Y]}$.

Lemma 6: Let (M, G) be a permutation group, and let $G^{(i)} = G_{[Y]}, 1 \leq i \leq n$, be the pointwise stabilizer of Y_i in G. We set $G_{(0)} = G$. For $1 \leq i \leq n$, let U_i be a complete right traversal for $G^{(i)}$ in $G^{(i-1)}$. Then, for $1 \leq i \leq n$, $K_r = \bigcup_{i=r}^{n} U_i$ is a generating set for $G^{(i-1)}$.

Lemma 7: The cardinality of the power set of (M, G) is $\prod_{i=1}^{n} | U_i |$, and can be determined in $O(m^2)$ steps.

We now construct the sets U_i from a given set G of permutations generating $< G >$. The sets U_i are stored in representation matrices.

Definition 11: Let N be a nxn matrix of permutations with the following properties: (1) $N_{i,i}$ is the identity permutation e, $1 \leq i \leq n$. (2) For $i > j$, $N_{i,j}$ is empty. (3) For $i < j$, $N_{i,j}$ is either empty, or is a permutation g such that g pointwise fixes the set $\{1, \ldots, i-1\}$, and $g(i) = j$. Then N is called a ***representation matrix.***

Algorithm ..1 describes the construction of a representation matrix from generators. The input to the algorithm is the fundamental set G_i (for a single stage network) or G (for a multistage network). The output of the algorithm is the representation matrix for the power set $< G_i >$ or $< G >$ depending on the input. A queue Q is used for intermediate storing of pair products of entries in N. Algorithm ..2 is used as a subroutine whose input is a partially completed representation matrix N, and the permutation g which is to be represented. The output of algorithm ..2 is true if g is in T_N (T_N is precisely the set of all those permutations for which Ismember returns **true**), false otherwise. As a side effect, if g is not in T_N, then N is changed such that g becomes representable by replacing an empty entry with a permutation. Furthermore, the coordinates i and j of this new entry are returned.

Algorithm ..1 Construction of a representation matrix from generators.

```
begin
  for i := 1 to n do
  begin
    N_{i,i} := e;
    for j := i + 1 to n do N_{i,j} := empty;
  end
  Q := G;
  while ¬empty(G) do
  begin
    remove g from Q;
    invoke algorithm ..2 with input g;
    if  algorithm ..2 returns false
      then
        append to Q all permutations hg' and g'h,
        where g' is the new entry N_{i,j} made for g
        by algorithm ..2 and h is any non-empty entry
        in N except the diagonal elements N_{k,k}.
  end
end
```

Algorithm ..2 Ismember

```
begin
  i := 0;
  ismember := true;
  while ((i < n)∧ ismember) do
  begin
    i := i + 1;
    j := g(i);
    if  (N_{i,j} ≠ empty) then   g := g(M_{i,j})^{-1}
    else
      begin
        ismember := false;
        N_{i,j} := g;
      end
  end
  return(ismember);
end
```

Thus, theorems 6 and 8 can be extended to the power set membership check. In fact we now have an algorithmic technique to check for the membership of an arbitrary permutation in the power set of an interconnection network.

Theorem 9: An arbitrary permutation $x \in \mathrm{Sym}(M)$ belongs to the power set of a single stage network $\mathrm{IN}_i = (M, G_i)$ with a process mapping $p \in \mathrm{Sym}(M)$, iff $xp^{-1} \in < G_i >$.

Theorem 10: An arbitrary permutation $x \in \mathrm{Sym}(M)$ belongs to the power set of a (m, n) network $\mathrm{IN} = (M, G)$ with a process mapping $p \in \mathrm{Sym}(M)$, iff $xp^{-1} \in < G >$.

Thus, given an arbitrary permutation which correponds to the mapping desired, we can now evaluate the number of stages (or passes) of the interconnection network required to realize the permutation. The fundamental set of the permutation network is evaluated first. Then, the given mapping is verified for membership in the fundamental set. If the given mapping is a member of the fundamental set then it is realized in one stage (or pass). Otherwise, a second stage is added and the new fundmental set is evaluated. The membership check is done again on the new fundamental set. If the given mapping is a member of the new fundamental set then it requires only two stages (passes). This process of evaluating a new fundamental set is continued till the given permutation is a member of the fundamental set. The number of times a new fundamental set is evaluated gives the number of stages (passes) required to realize the given mapping.

Conclusion

This paper focused on characterizing interconnection networks based on their ability to realize process permutations. Group theoretic results were used to characterize them. This analysis can be done for other static and dynamic networks. Choosing a network based on process permutation rather than the power set of the interconnection network can reduce the number of passes required to realize certain process permutation. This approach could also be extended to provide a framework in which one may pursue to design networks for specific applications. An extension of the present group theoretic formulation would be to be able to formulate networks other than permutation networks.

References

[1] V. Chaudhary and J. K. Aggarwal, "Generalized mapping of parallel algorithms onto parallel architectures," *Proc. Int. Conf. Parallel Process.*, 1990, pp. 137-141.

[2] V. Chaudhary and J. K. Aggarwal, "A generalized scheme for mapping parallel algorithms," *IEEE Trans. on Parallel and Distributed Systems* Vol. 4, No. 3, Mar, 1993, pp. 328 – 346.

[3] G. B. Adams and H. J. Seigel, "On the number of permutations performable by the augmented data manipulator network", *IEEE Trans. Comput.*, Vol. C-31, Apr. 1982, pp. 270-277.

[4] H. T. Kung and D. Stevenson, "A software technique for reducing the routing time on a parallel computer with a fixed interconnection network", *High Speed Computer and Algorithm Organisation*, D. J. Kuck, D. H. Lawrie, and A. H. Sameh (Editors), Academic Press Inc. 1977, pp. 423-433.

[5] W. R. Scott, *Group Theory*, Dover Pulications Inc., New York 1987.

[6] A. Y. Oruc and M. Y. Oruc, "On testing isomorphism of permutation networks", *IEEE Trans. Comput.*, Vol. C-34, Oct. 1982, pp. 958-962.

[7] S. Yalamanchile and J. K. Aggarwal, "A characterization and analysis of parallel processor interconnection networks", *IEEE Trans. Comput.*, Vol. C-36, Jun. 1987, pp. 680-691.

[8] V. Cherkassky and M. Malek, "On permuting properties of regular rectangular SW-Banyans," *IEEE Trans. Comput.*, Vol. C-34, Jun. 1985, pp. 542-546.

[9] C. M. Hoffman, "Group-theoretic algorithms and graph isomorphism," *Lecture notes in computer science*, Springer-Verlag, Berlin, 1982.

[10] V. E. Benes, "Applications of group theory to connecting networks," *The Bell Sys. Tech. Journal*, Feb. 1975, pp. 407-420.