Short Notes

Testing the Dynamic Full Access Property of a Class of Multistage Interconnection Networks

Tsern-Huei Lee and Jin-Jye Chou

***Abstract*—A banyan network and its topologically equivalent ones have recently been adopted as the interconnection networks in a multiprocessor system. Often a multiprocessor system is reconfigured when the banyan network becomes faulty. It is possible to avoid a complicated reconfiguration process as long as the faulty banyan network still possesses the dynamic full access (DFA) property. In this short note, we determine a necessary and sufficient condition for a faulty banyan network to possess the DFA property and design a testing procedure based on the condition. The testing procedure can be used to decompose a faulty banyan network into subsystems possessing the DFA property. We also evaluate the probability that a banyan network loses the DFA property, given the number of faulty switching elements. It is found that as long as faults do not occur in switching elements located in the first and the last stages, this probability is very small, even when there are quite a few faulty switching elements.**

***Index Terms*—Multiprocessor system, multistage interconnection networks, fault tolerance, dynamic full access**

I. Introduction

Multistage interconnection networks (MIN's) have recently been adopted to interconnect processors in a multiprocessor system. Various MIN's, such as the omega network, the flip network, the indirect binary n-cube network, and the baseline network, had been proposed for different applications. However, these networks were proved [2] to be topologically equivalent to the regular SW banyan network [1] with spread and fan-out of 2. Therefore, we do not distinguish these terms in this short note, and simply use "banyan network" to represent this topologically equivalent class of MIN's.

In a multiprocessor system interconnected by a banyan network, there is a unique path from any processor to any other one. This property makes routing very simple. Unfortunately, it also makes a banyan network lack fault tolerance capability. Often a multiprocessor system is reconfigured when the banyan network becomes faulty. However, it is possible to avoid a complicated reconfiguration process as long as the faulty banyan network still possesses the dynamic full access (DFA) property; i.e., any processor can be connected to any other processor in a finite number of passes through the faulty banyan network [5]. Therefore, if a faulty banyan network still possesses the DFA property, then a packet can be routed through it several times, if necessary, until the packet reaches its proper destination. For convenience, we say that processor i can reach processor j in k passes if it can be connected to processor j in k passes through the faulty banyan network.

Manuscript received November 9, 1992; revised March 5, 1993, and September 23, 1993. This work was supported in part by the National Science Council, Taiwan, Republic of China, under Contract NSC 82-0404-E-009-334.

T.-H. Lee is with the Department of Communication Engineering, National Chiao Tung University, Hsinchu, Taiwan 30035, Republic of China; e-mail: thlee@banyan.cm.nctu.edu.tw.

J.-J. Chou is with the Institute of Electronics, National Chiao Tung University, Hsinchu, Taiwan, 30035 Republic of China.

IEEE Log Number 9403104.

The collection of all faults is called the fault set. A fault set is said to be critical if it destroys the DFA property of a banyan network. In other words, a fault set is critical if there exists a processor that cannot reach some other processors in a finite number of passes. There exist excellent and efficient algorithms [6]–[9] to detect and locate faulty switching elements. However, to date, whether a fault set is critical has not efficiently been determined. Agrawal and Leu [10] proposed to test the DFA property of a banyan network by examining the power of the reachability matrix R. The (i,j)th element of R is equal to 1 if processor i can reach processor j in a single pass. Otherwise, it is equal to 0. A faulty banyan network possesses the DFA property if and only if (iff) there exists a finite positive integer k such that none of the elements in R^k is 0. The complexity of matrix multiplication makes this approach infeasible for large networks. Varma and Raghavendra [11] and Kumar and Wang [12] determined some sufficient conditions for a fault set to be noncritical. In [11], a necessary and sufficient condition was derived for a banyan network with 16 inputs/outputs (I-O's). Unfortunately, that criterion is not valid for banyan networks having more than 16 I-O's.

The purpose of this short note is to present a procedure for testing the DFA property of banyan networks of any size. We assume that faults had been detected and located. The fault model adopted here is identical to that used in [11]; i.e., only switching elements can fail, and the failure brings down that component. A necessary and sufficient condition for a banyan network to possess the DFA property is determined in Section II. In Section III, a testing procedure based on the necessary and sufficient condition is first presented and then used to decompose a faulty banyan network into subsystems possessing the DFA property. Two examples are studied in Section IV. The probability of being critical, given the size of a fault set, is also evaluated in this section. Some conclusions are finally drawn in Section V.

II. A Necessary and Sufficient Condition for a Fault Set to be Critical

Consider an n-stage banyan network. The I-O's can be numbered, respectively, from the top by 0 to $2^n - 1$, and can be represented by binary sequences of length n, say, $a_{n-1}a_{n-2}\cdots a_0$. In stage $k, 0 < k < n$, there are 2^{k-1} remaining subnetworks. It can be verified that a switching element in stage k can be uniquely represented by $(a_n \cdots a_k, b_1 \cdots b_{k-1})$, where $a_n \cdots a_k$ denotes the label of the switching element numbered from the top within the subnetwork, and $b_1 \cdots b_{k-1}$ represents the label of the subnetwork numbered. Such a numbering scheme was proposed by Beckmann [4]. Fig. 1 illustrates an example of a four-stage banyan network with the numbering scheme. In this short note, we assume that output i is paired with input i for all i. Furthermore, we shall use A and B to denote, respectively, a set of inputs and a set of outputs for the rest of the short note.

It is clear that a switching element in fault disconnects an input set A from an output set B in the sense that a cannot reach b in a single pass for any $a \in A$ and $b \in B$. Let Ω denote the set of all binary sequences of length n. Also, let $(a_{n-1} \cdots a_k)_n$ denote a subset of Ω such that $x \in (a_{n-1} \cdots a_k)_n$ iff the leftmost $n-k$ bits of x are $a_{n-1} \cdots a_k$. It is not hard to see that a faulty switching element in stage k represented by $(a_{n-1} \cdots a_k, b_1 \cdots b_{k-1})$ disconnects $A =$

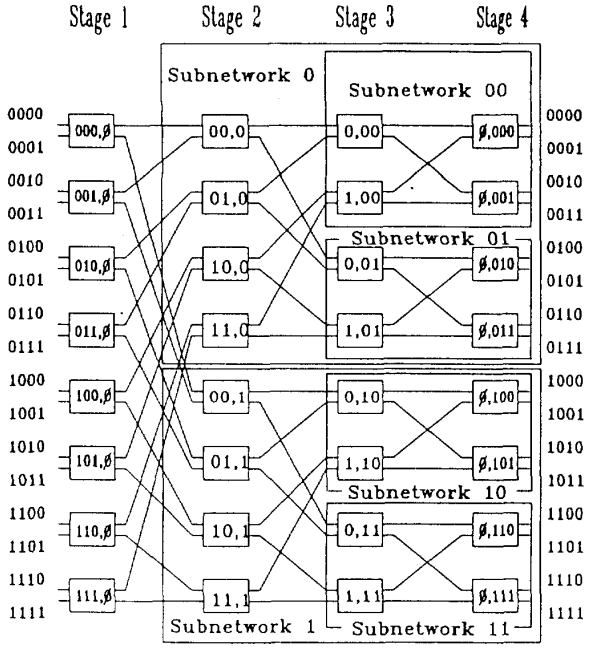

Fig. 1. Beckmann's numbering scheme for a four-stage banyan network.

$(a_{n-1}\cdots a_k)_n$ from $B = (b_1\cdots b_{k-1})_n$. Notice that in the above discussion, the two sets A and B are chosen to be as large as possible. In fact, any subset of A is disconnected from any subset of B by the faulty switching element. To generalize, an input set A is said to be disconnected from an output set B by a fault set F if a cannot reach b in a single pass for any $a \in A$ and $b \in B$. The following theorem states a necessary and sufficient condition for a fault set to be critical.

Theorem 1: A fault set F is critical iff there exist nonempty input set A and output set B, so that A and B are disconnected by F and satisfy $A \cup B = \Omega$. Here A and B are simply considered as sets of binary sequences when performing set union.

Proof: Suppose A and B are both nonempty, disconnected by F, and satisfy $A \cup B = \Omega$. If $A \subset B$, then $B = \Omega$, meaning that inputs in A are isolated. In other words, if $a \in A$, then a cannot reach any output in a finite number of passes. Therefore, the fault set F is critical. Similarly, if $B \subset A$, then $A = \Omega$, and F is critical. Consider the case where $A \not\subset B$ and $B \not\subset A$. Let $A' = A \cap B^c$, where B^c is the complement of set B. It is not hard to see that $A' \neq \emptyset$ is disconnected from B, and satisfies $A' \cup B = \Omega$.

Because output i is paired with input i for all i, any input $a \in A'$ can reach only those outputs paired with the inputs in A'. As a result, any input $a \in A'$ cannot reach any output $b \in B$ in a finite number of passes. This implies that F is critical.

Conversely, suppose the fault set F is critical. Since F is critical, there exist an input a' and an output b' such that a' cannot reach b' in a finite number of passes. Let B denote the maximal set of outputs that cannot be reached by a' in a finite number of passes. B is obviously nonempty, because $b' \in B$. If $B = \Omega$, then one can select $A = \{a'\}$ to complete the proof.

Suppose $B \neq \Omega$. Let A be the set of inputs paired with those outputs in B^c. Such a selection satisfies $A \neq \emptyset$ and $A \cup B = \Omega$. Besides, any element $a \in A$ cannot reach any element $b \in B$ in a finite number of passes. Otherwise, suppose there exist some $a \in A$ and $b \in B$ so that a can reach b in a finite number of passes. As a result, a' can reach a in a finite number of passes, and a can reach b also in a finite number of passes, implying that a' can reach b in a finite number of passes, a contradiction to the selection of B. Since a cannot reach b in a finite number of passes, it is implied that they are disconnected (in a single pass). Therefore, we conclude that if F is critical, then there exist nonempty input set A and output set B such that A and B are disconnected by F and satisfy $A \cup B = \Omega$. This completes the proof of Theorem 1.

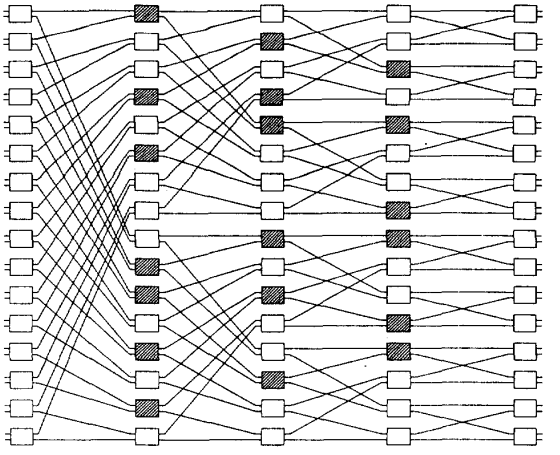

Fig. 2. The noncritical fault set for Example 1.

It follows from Theorem 1 that a fault set is critical if it contains an element in the first or the last stage. Further, the two sets A and B can be chosen to be disjoint; i.e., $A \cap B = \emptyset$.

III. A Testing Procedure

In this section, we provide a procedure based on Theorem 1 for testing the DFA property of faulty banyan networks. To avoid trivial cases, we assume faults do not occur in switching elements located in the first and last stages.

Suppose the fault set $F = \{f_1, f_2, \cdots, f_k\}$ is given, where f_j, $1 \leq j \leq k$, represents a faulty switching element. Let R_i denote the maximal input set disconnected from output set $\{i\}$. Also, let Q_i denote the maximal output set disconnected from input set $\{i\}$. According to Theorem 1, if F is critical, then either input 0 is in set A or output 0 is in set B. To expedite the testing procedure, one can partition the inputs into equivalent classes so that two inputs, i and j, are in a same class iff $Q_i = Q_j$. Similarly, the outputs can be partitioned into equivalent classes so that outputs k and l are in a same class iff $R_k = R_l$.

Since, by assumption, faults do not occur in switching elements located in the first and last stages, each equivalent class consists of at least four elements. The partition, of course, depends on the particular fault set. To make the partition useful, we require two inputs (outputs) to be in a same class iff $Q_i = Q_j$ ($R_i = R_j$) for any fault set. Under this definition, each equivalent class contains exactly four elements. For the type of networks illustrated in Fig. 1, two inputs (outputs) are in a same class iff their binary representations differ only in the rightmost two bits. For example, if $\mathrm{Cl}(i)$ represents the equivalent class containing input or output i, then $\mathrm{Cl}(0) = \mathrm{Cl}(2) = \{0, 1, 2, 3\}$. Obviously, to check the DFA property, inputs or outputs in a same equivalent class can be considered together.

Let A_{f_j} and B_{f_j} denote, respectively, the largest input set and output set disconnected by the fault set $\{f_j\}$. As a result, R_i is equal to $\cup_{f_j \in G} A_{f_j}$, where G is a subset of F, so that $f_j \in G$ iff $i \in B_{f_j}$. Similarly, we have $Q_i = \cup_{f_j \in H} B_{f_j}$, where H is a subset

TABLE I
TESTING RESULTS FOR EXAMPLE 1

iteration	content of A	content of B
1	0 1 2 3	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
2	0 1 2 3 28 29 30 31	0 1 2 3 4 5 6 7 12 13 14 15 20 21 22 23
3	0 1 2 3 28 29 30 31 8 9 10 11	0 1 2 3 4 5 6 7 20 21 22 23
4	0 1 2 3 28 29 30 31 8 9 10 11 12 13 14 15	0 1 2 3 4 5 6 7
5	0 1 2 3 28 29 30 31 8 9 10 11 12 13 14 15 16 17 18 19	$\emptyset$
6	0 1 2 3 8 9 10 11 12 13 14 15 20 21 22 23 24 25 26 27 28 29 30 31	0 1 2 3
7	0 1 2 3 8 9 10 11 12 13 14 15 20 21 22 23 24 25 26 27 28 29 30 31	0 1 2 3 4 5 6 7
8	0 1 2 3 8 9 10 11 12 13 14 15 20 21 22 23 24 25 26 27	0 1 2 3 4 5 6 7 16 17 18 19
9	8 9 10 11 12 13 14 15 24 25 26 27	0 1 2 3 4 5 6 7 16 17 18 19 28 29 30 31
10	8 9 10 11 24 25 26 27	0 1 2 3 4 5 6 7 16 17 18 19 28 29 30 31 20 21 22 23
11	24 25 26 27	0 1 2 3 4 5 6 7 16 17 18 19 28 29 30 31 20 21 22 23 12 13 14 15
12	$\emptyset$	0 1 2 3 4 5 6 7 16 17 18 19 28 29 30 31 20 21 22 23 12 13 14 15 8 9 10 11

of F, so that $f_j \in H$ iff $i \in A_{f_j}$. Therefore, R_i and Q_i can be determined as long as the fault set is given.

The following testing procedure named CHECK_DFA receives a system Ω as its input, and returns a subsystem A as its output. The system Ω possesses the DFA property iff $A = \emptyset$. In this procedure, the set Temp is used to store the binary sequences that have not been assigned to A or B.

CHECK_DFA(Ω, A)
Select the least numbered element $i \in \Omega$
  /* check if input $0 \in A$ */
Set $A = \mathrm{Cl}(i), B = Q_i \cap \Omega, \mathrm{Temp} = \Omega - A - B$
  while $(B \neq \emptyset)$ do
    {if $(A \cup B = \Omega)$, then return A
      else
      select the least numbered element $j \in$ Temp
      $A = A \cup \mathrm{Cl}(j)$
      > Temp $= (\mathrm{Temp} - \mathrm{Cl}(j)) \cup (B \cap Q_j^c)$
      $B = B \cap Q_j$}
  * check if output $0 \in B$ when input $0 \notin A$ */
  Set $A = R_i \cap \Omega, B = \mathrm{Cl}(i), \mathrm{Temp} = \Omega - A - B$
  while $(A \neq \emptyset)$ do
    {if $(A \cup B = \Omega)$, then return A
      else
      select the least numbered element $j \in$ Temp
      Set $B = B \cup \mathrm{Cl}(j)$
      Temp $= (\mathrm{Temp} - \mathrm{Cl}(j)) \cup (A \cap R_j^c)$
      $A = A \cap R_j$}
return A

Notice that before the first while-do loop, B is set to be equal to $Q_i \cap \Omega$ rather than Q_i, because the CHECK_DFA procedure will be called by another procedure presented below, and when it is called, Ω can be a subsystem of the given faulty banyan network. For the same reason, A is set to be equal to $R_i \cap \Omega$ rather than to R_i before the second while-do loop.

When a fault set destroys the DFA property of a banyan network, one may be interested in finding subsystems of the faulty banyan network that still possess the DFA property. It is possible that the DFA property is maintained if a small number of processors are excluded. Usually, the efficiency of a subsystem containing most of the processors is still much higher than that of a single-processor system. In the following, we use the CHECK_DFA procedure to decompose a faulty banyan network into subsystems possessing the DFA property. Suppose F is critical and input set A and output set B are found, so that A is disconnected from $B, A \cup B = \Omega$, and $A \cap B = \emptyset$. It should be clear that processors in A and processors in B must belong to different subsystems. Therefore, one can decompose a faulty banyan network by the following recursive procedure named DECOMPOSE(Ω).

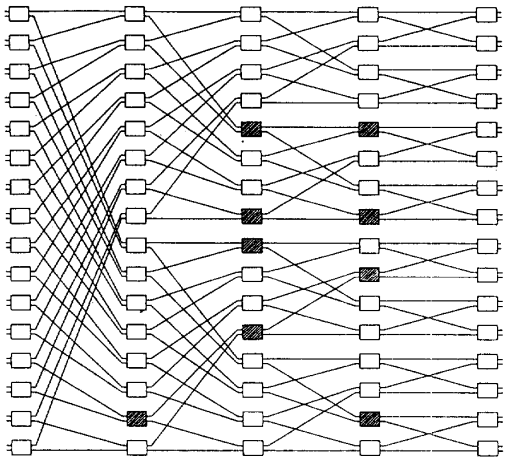

Fig. 3. The critical fault set for Example 2.

DECOMPOSE(Ω)
CHECK_DFA(Ω, A)
if $(A = \emptyset)$, then
  List Ω as a subsystem
else
  DECOMPOSE (A)
  DECOMPOSE($\Omega - A$)

TABLE II
TESTING RESULTS FOR EXAMPLE 2

First call of CHECK_DFA (Ω is used as the input)		
iteration	content of A	content of B
1	0 1 2 3	8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 28 29 30 31
2	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 28 29 30 31
3	0 1 2 3 4 5 6 7 24 25 26 27	8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 28 29 30 31

Second call of CHECK_DFA ($\{0,1,2,3,4,5,6,7,24,25,26,27\}$ is used as the input)		
iteration	content of A	content of B
1	0 1 2 3	$\emptyset$
2	$\emptyset$	0 1 2 3

Third call of CHECK_DFA ($\{8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,28,29,30,31\}$ is used as the input)		
iteration	content of A	content of B
1	8 9 10 11	8 9 10 11 16 17 18 19 20 21 22 23 28 29 30 31
2	8 9 10 11 12 13 14 15	8 9 10 11 28 29 30 31
3	8 9 10 11 12 13 14 15 16 17 18 19	$\emptyset$
4	8 9 10 11 12 13 14 15 28 29 30 31	8 9 10 11
5	28 29 30 31	8 9 10 11 16 17 18 19
6	28 29 30 31	8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23

Fourth call of CHECK_DFA ($\{28,29,30,31\}$ is used as the input)		
iteration	content of A	content of B
1	28 29 30 31	$\emptyset$
2	$\emptyset$	28 29 30 31

Fifth call of CHECK_DFA ($\{8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23\}$ is used as the input)		
iteration	content of A	content of B
1	8 9 10 11	8 9 10 11 16 17 18 19 20 21 22 23
2	8 9 10 11 12 13 14 15	8 9 10 11
3	8 9 10 11 12 13 14 15 16 17 18 19	$\emptyset$
4	8 9 10 11 12 13 14 15	8 9 10 11
5	$\emptyset$	8 9 10 11 16 17 18 19

It is clear that the DECOMPOSE procedure can be used to check the DFA property. A faulty banyan network possesses the DFA property iff there is only one subsystem listed by the DECOMPOSE procedure when it terminates.

IV. ILLUSTRATIVE EXAMPLES

In this section, we apply the DECOMPOSE procedure to examine the DFA property for some faulty banyan networks. If the DFA property is destroyed, the faulty banyan network is decomposed into subsystems possessing the DFA property. We also evaluate the probability that a fault set is critical given the number of faulty switching elements. In this section, faulty switching elements are represented by shaded squares.

Example 1: Consider the fault set shown in Fig. 2. After performing the DECOMPOSE procedure, we found that F is noncritical. The contents of A and B (inside the CHECK_DFA procedure) after each iteration are shown in Table I.

Example 2: Consider the fault set shown in Fig. 3. Again, the DECOMPOSE procedure is performed to check whether F is critical. The results are shown in Table II. Since there are three subsystems, namely, $\{0,1,2,3,4,5,6,7,24,25,26,27\}$, $\{8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23\}$, and $\{28,29,30,31\}$, that are listed by the DECOMPOSE procedure, the fault set F is critical. However, each of the listed subsystems does possess the DFA property.

We now evaluate the probability that a fault set is critical, given the number of faulty switching elements. Consider an n-stage banyan network, and denote this probability by $p_C(i)$ when there are i faulty switching elements. Again, we assume that no faults occur in switching elements located in the first and the last stages. To evaluate $p_C(i)$, we further assume that each switching element is equally likely to be faulty. The number of switching elements in stages 2 to $n-1$ is given by $M = (n-2)2^{n-1}$.

Let $N_i = \binom{M}{i}$ denote the total number of possible locations of i faulty switching elements. The probability $p_C(i)$ is defined as $C(i)/N_i$, where $C(i)$ is the number of distinct critical fault sets having i elements. The value $C(i)$ is determined by performing the DECOMPOSE procedure for all the N_i possible fault patterns. Fig. 4 illustrates the probability $p_C(i)$ against i for $n = 6, 7$, and 8. It can be seen that the probability is very small up to five faulty switching

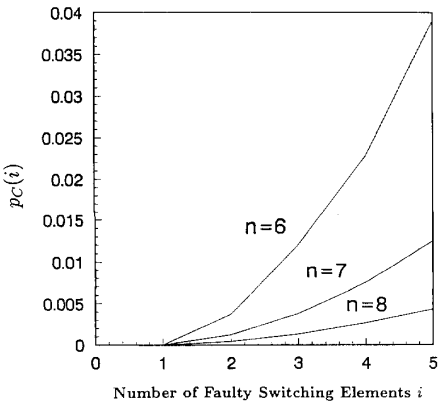

Fig. 4. $p_C(i)$ against number of faulty switching elements.

elements. For example, given that there are five faulty switching elements, this probability is roughly equal to 0.039, 0.012, or 0.004 for $n = 6$, 7, or 8, respectively. Therefore, it is very likely that a complicated reconfiguration process can be avoided, as long as the switching elements in the first and the last stages can be kept fault-free. In real implementations, the switching elements located in the first and the last stages can be well protected by adding redundant circuits.

V. Conclusion

In this short note, we determined a necessary and sufficient condition for a faulty banyan network to possess the DFA property, and designed a testing procedure based on the condition. The testing procedure was used to decompose a faulty banyan network into subnetworks possessing the DFA property. We showed that banyan networks can tolerate quite a few faulty switching elements without losing the DFA property. As a consequence, a complicated reconfiguration process can often be avoided if the switching elements in the first and the last stages can be kept fault-free. There are several interesting topics in this area that can be further studied. For example, sufficient conditions for a fault set to be noncritical, other than those determined previously, can be searched to expedite the testing. Another topic that is currently under investigation is to design and evaluate the performance of rerouting schemes.

References

[1] L. R. Goke and G. J. Lipovski, "Banyan network for partitioning multiprocessor systems," *Proc. 1st Ann. Symp. Comput. Architecture*, 1980, pp. 21–30.

[2] C. L. Wu and T. Y. Feng, "On a class of multistage interconnection networks," *IEEE Trans. Comput.*, vol. C-29, no. 8, pp. 694–702, Aug. 1980.

[3] C. P. Kruskal and M. Snir, "The performance of multistage interconnection network," *IEEE Trans. Comput.*, vol. C-32, no. 12, pp. 1091–1098, Dec. 1983.

[4] T. T. Lee, "Nonblocking copy networks for multicast packet switching," *IEEE J. Selected Areas Commun.*, vol. 6, pp. 1455–1467, Dec. 1988.

[5] J. P. Shen and J. P. Hayes, "Fault-tolerance of dynamic-full-access interconnection networks," *IEEE Trans. Comput.*, vol. C-33, pp. 443–454, Mar. 1984.

[6] T. Y. Feng and I. P. Kao, "On fault-diagnosis of some multistage networks," *Proc. 1982 Int. Conf. Parallel Processing*, 1982, pp. 99–101.

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

[8] S. Thanawastien and V. P. Nelson, "Optimal fault detection test sequences for shuffle/exchange networks," in *Proc. 13th Ann. Int. Symp. Fault-Tolerant Comput.* , 1983, pp. 442–445.

[9] C. L. Wu and T. Y. Feng, "Fault diagnosis for a class of multistage interconnection networks," *IEEE Trans. Comput.*, vol. C-30, pp. 743–758, Oct. 1981.

[10] D. P. Agrawal and J. S. Leu, "Dynamic accessibility testing and path length optimization of multistage interconnection networks," *IEEE Trans. Comput.*, vol. C-34, no. 3, pp. 255–266, Mar. 1985.

[11] A. Varma and C. S. Raghavendra, "Fault-tolerant routing in multistage interconnection networks," *IEEE Trans. Comput.*, vol. 38, pp. 385–393, Mar. 1989.

[12] V. P. Kumar and S. J. Wang, "Reliability enhancement by time and space redundancy in multistage interconnection networks," *IEEE Trans. Reliability*, vol. 40, pp. 461–473, Oct. 1991.

RH: A Versatile Family of Reduced Hypercube Interconnection Networks

Sotirios G. Ziavras

***Abstract*—The binary hypercube has been one of the most frequently chosen interconnection networks for parallel computers because it provides low diameter and is so robust that it can very efficiently emulate a wide variety of other frequently used networks. However, the major drawback of the hypercube is the increase in the number of communication channels for each processor with an increase in the total number of processors in the system. This drawback has a direct effect on the very large scale integration complexity of the hypercube network. This short note proposes a new topology that is produced from the hypercube by a uniform reduction in the number of edges for each node. This edge reduction technique produces networks with lower complexity than hypercubes while maintaining, to a high extent, the powerful hypercube properties. An extensive comparison of the proposed reduced hypercube (RH) topology with the conventional hypercube is included. It is also shown that several copies of the popular cube-connected cycles network can be emulated simultaneously by an RH with dilation 1.**

***Index Terms*—Emulation, hypercube, hypercube-like systems, interconnection networks, cube-connected cycles, parallel processing**

I. Introduction

A wide variety of interconnection networks have been proposed for parallel computing systems, including rectangular meshes, trees, shuffle exchange networks, Omega networks, indirect binary n-cubes, and direct binary n-cubes. The family of direct binary n-cubes is a special case of the family of direct k-ary n-cubes with n dimensions and k nodes in each dimension [24]. In the rest of this short note, the term *n-dimensional hypercube or n-cube* is used to denote the direct binary n-cube. The n-dimensional hypercube is composed of 2^n nodes and has n edges per node. If unique n-bit binary addresses

Manuscript received January 4, 1993; revised August 23, 1993, and November 24, 1993. This work was supported in part by the National Science Foundation under Grants CCR-9109084 and CDA-9121475. and in part by the New Jersey Institute of Technology under Grant SBR-421240.

The author is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, University Heights, Newark, NJ 07102 USA; e-mail: ziavras@hertz.njit.edu.

IEEE Log Number 9403105.