# REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188



AD-A264 591

1. AGE...

3. REPORT TYPE AND DATES COVERED

4. TITLE

"Quantitative Reliability Analysis of Redundant Multistage Interconnection Networks"

5. FUNDING NUMBERS

DAAL03-91-G-0031

6. AUTHOR(S)

Nita M. Kini
Dharma P. Agrawal
Anup Kumar



7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Electrical & Computer Engineering, Box 7911
North Carolina State University
Raleigh, North Carolina 27695-7911
Attn: Prof. Dharma P. Agrawal

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U. S. Army Research Office
P. O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

ARO 28192.2-MA

11. SUPPLEMENTARY NOTES

The view, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)

Many Redundant Multistage Interconnection Networks (RMINs) have been proposed to provide increased bandwidth and enhanced reliability by introducing alternate paths between each source-destination pair of the network. But, there exists no generalized algorithm which could provide a quantitative reliability measure of such MINs. This paper introduces a systematic technique for computing the exact reliability expression for both RMINs and NonRMINs considered in this paper include Extra Stage Cube, F, IADM, INDRA, Multipath Omega and Chained Baseline (Uniform and NonUniform). The proposed algorithm is a one-step method, which requires updating of only two vectors and completely avoids matrix multiplications. Thus the proposed GEARMIN (Generalized Evaluation Algorithm for Reliability of RMINs) is substantially less complex than existing techniques, while it is general enough to enable evaluation of all other existing reliability parameters as special cases.

14. SUBJECT TERMS

15. NUMBER OF PAGES

16. PRICE CODE

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| UNCLASSIFIED | UNCLASSIFIED | UNCLASSIFIED | UL |

DIMACS Series in Discrete Mathematics
and Theoretical Computer Science
Volume 5, 1991

# Quantitative Reliability Analysis of Redundant Multistage Interconnection Networks

NITA M. KINI, ANUP KUMAR, AND DHARMA P. AGRAWAL

ABSTRACT. Many Redundant Multistage Interconnection Networks (RMINs) have been proposed to provide increased bandwidth and enhanced reliability by introducing alternate paths between each source-destination pair of the network. But, there exists no generalized algorithm which could provide a quantitative reliability measure of such MINs. This paper introduces a systematic technique for computing the exact reliability expression for both RMINs and NonRMINs. The RMINs considered in this paper include Extra Stage Cube, F, IADM, INDRA, Multipath Omega and Chained Baseline (Uniform and NonUniform). The proposed algorithm is a one-step method, which requires updating of only two vectors and completely avoids matrix multiplications. Thus the proposed GEARMIN (Generalized Evaluation Algorithm for Reliability of RMINs) is substantially less complex than existing techniques, while it is general enough to enable evaluation of all other existing reliability parameters as special cases.

## 1. Introduction

Multistage Interconnection Networks (MINs) provide programmable data paths between processors and/or processor-to-memory modules in a multiprocessor environment. A majority of $N$ input MINs ($N = 2^n$) are designed with $n$ stages, with each stage consisting of $N/2$ 2-input 2-output switches. Redundancy is introduced in these MINs so as to provide fault tolerance and/or increased throughput between a given source-destination pair. The redundant networks have multiple paths from a given source to an arbitrary destination. Redundancy could be incorporated in MINs by either adding an extra stage, by increasing switch complexity, by duplicating the network, or by chaining the switches of individual stages. Such redundant MINs (RMINs) differ in cost and reliability, but there does not currently exist any algorithm or technique that could provide a quantitative reliability measure of these RMINs.

The use of RMINs has become increasingly important because failure of

1980 *Mathematics Subject Classification* (1985 *Revision*). Primary 68, 60, 94.

one or more elements in a system should cause a degradation in performance, rather than a complete breakdown. Such graceful degradation is easily supported by RMINs, for faults in various system modules. In this paper, we introduce an accurate algorithm for computing the reliabilities of these networks and consider the cost-performance tradeoff. Our algorithm is applicable to both redundant and nonredundant networks of any size. It may be noted that most $N$ input ($N = 2^n$) regular nonredundant networks using $2 \times 2$ switches are topologically equivalent to each other [2] and hence will have the same reliability figures. Also, a single link failure will prevent one or more source-destination linkings. Hence, in this paper we concentrate on the reliability evaluation of RMINs.

Blake et al. [3, 4] represent the Augmented Shuffle Exchange Network as a CTMC (Continuous Time Markov Chain) to obtain the upper and the lower bounds for the network reliability, and do not provide a closed form expression for networks larger than $16 \times 16$. The problem with this approach is the exponential growth of the state space, as the network size increases.

Botting et al. [5] generated a reliability expression for MINs, but the method is compute-intensive since it involves matrix multiplications. The method for making the reliability expressions of the different paths disjoint is time consuming since all paths are initially derived and then the reliability expressions are obtained by observing all other paths while computing the reliability expression of each path.

Das [7] used simulation techniques to compute the availability (rather than reliability) of the system, using certain Mean Time to Bus/Link Failure for processors, Memory Modules and Switching Elements of the MIN, computing the number of faulty units of each type at a given time, creating those many faults and finally finding where there exist paths between $N/2$ processors and $N/2$ Memory Modules. They are also concerned with reachability of $N/2$ PEs to $N/2$ MMs, just examining if there exists one path, and not considering alternate paths or disjointness of the paths (hence their approach may not be accurate). Their conclusions are mainly based on the simulation results, while they have not given any analytical approach.

Macaluso et al. [9] compute the reliability of networks by computing the reachability matrix and assuming failure rate of PEs, MMs, and SEs.

Varma and Raghavendra [14] obtained the graph model of INDRA and augmented $C$ network (ACN) and then performed a serial-parallel transformation of redundant graphs. They used a recursive relation to obtain the reliability of the networks.

In this paper, we introduce an analytical framework for computing the exact terminal reliability of MINs (both Redundant and NonRedundant). We compute the reliability expression associated with the linking from any source to any single destination. Our algorithm computes the exact expression for a network of any size and needs to update only two vectors as it moves along the graph. The updating process is accomplished by simple bit operations.

## II. Network topology

In this paper, we concentrate on the most popular RMINs, including Extra Stage Cube, $F$, IADM, INDRA, Multipath Omega, Uniform and Nonuniform Chained Baseline networks. For completeness of the text, we briefly describe the topologies of these networks.

**Extra stage cube network.** The Extra Stage Cube (ESC) network [1] is formed by adding an extra stage to the input side of the generalized cube MIN. The generalized cube is an $N \times N$ MIN, $(N = 2^n)$, with $n = \log_2 N$ stages, and each stage consists of $N/2$ switches. At a stage $i$, the upper and lower input labels of each switch differ only in the $i$th bit position. The ESC is formed by adding an extra stage to the input side of the network, along with multiplexers and demultiplexers at the input and output stages respectively. A fault in a switch of stage $n$ requires no change in the network configuration, with stage $n$ remaining disabled. If a fault occurs in stage 0, then stage $n$ is enabled and stage 0 is disabled. For a fault in a switch of any stage between 1 to $n-1$, both stages $n$ and 0 are enabled. The ESC can be controlled by a simple extension of the routing tags used for the generalized cube. The ESC for $N = 8$ is illustrated in Figure 1.[1]

**F network.** The $N$-input $(N = 2^n)$ $F$ network [6] consists of $n$ stages of $N$ nodes at each stage. The stages are numbered from 0 to $(n-1)$. The node address at the input (source) and the output (destination) stages are represented by $n$-bit binary numbers.

$$S = \sum_{k=0}^{n-1} s_k 2^k \qquad s_k \in \{0, 1\}$$

$$D = \sum_{k=0}^{n-1} d_k 2^k \qquad d_k \in \{0, 1\}$$

At stage $j$, a node $P$ $(= \sum_{k=0}^{n-1} p_k 2^k, p_k \in \{0, 1\})$ is connected to nodes $P$, $Q$, $R$ and $V$ of the next stage with addresses given as follows:

$$P = (p_{n-1}, \ldots, p_{j+1}, p_j, p_{j-1} \cdots p_0)$$
$$Q = (p_{n-1}, \ldots, p_{j+1}, \bar{p}_j, p_{j-1} \cdots p_0)$$
$$R = (\bar{p}_{n-1}, \ldots, \bar{p}_{j+1}, p_j, p_{j-1} \cdots p_0)$$
$$V = (p_{n-1}, \ldots, p_{j+1}, p_j, p_{j-1} \cdots p_0).$$

The relation $Q$ defines node $P$ with the $j$th bit being complemented and is referred to as the $j$th bit complement of $P$. The relation $R$ defines the complementary dual of node $P$; whereas the relation $V$ defines the $j$th bit complementary dual of node $P$ (refer to Figure 2).

[1] All figures and tables are at the end of this paper.

**IADM network.** The $N$-input IADM network [10] consists of $(n+1)$ stages, with $N$ nodes in each stage. The stages are numbered from 0 to $n$. The node address at the input (source) and the output (destination) stages are represented by $n$-bit binary numbers:

$$S = \sum_{k=0}^{n-1} s_k 2^k \qquad s_k \in \{0, 1\}$$

$$D = \sum_{k=0}^{n-1} d_k 2^k \qquad d_k \in \{0, 1\}.$$

At stage $j$, a node with address $P$ can be connected to any one of the three nodes given by the relation $I$, $A$ and $M$ where

$$I = P$$

$$A = (P + 2^j) \quad \text{Mod } N$$

$$M = (P - 2^j) \quad \text{Mod } N$$

The IADM network can tolerate some faults because of multiple paths between a source $S$, and destination $D$, if $S \neq D$. Figure 3 shows a 4-stage IADM network.

**INDRA network.** INDRA networks [12] can be conceived as the union of $R$ parallel subnetworks each with $\log_R N$ stages of $R \times R$ switches, with an initial distribution stage at the input. Typically the Omega network is used as a subnetwork. Other link interconnection patterns including cube interconnections are possible. The INDRA network is $(R-1)$ fault tolerant since there exist $R$ disjoint paths between any source-destination pair.

**Multipath omega network.** The Multipath Omega Network [11] is derived from the Omega MIN. A $B^m \times B^m$ Omega network consists of $m$ stages of $B \times B$ crossbar switches, linked by a generalized shuffle. The paths in a multipath omega network need not be distinct, since a single fault may affect more than one path. Routing in this network is controlled by the routing tags, which resembles the procedure used for the Omega network and the generalized shuffle network. A Multipath Omega network for $N = 8$ is illustrated in Figure 5.

**Chained baseline network.** An $N$-input Baseline network [13] consisting of $\log_2 N$ stages of $N/2$ switching elements, has a unique path between a given source and destination, and hence is very susceptible to failures of any of the switching elements (SEs). The chained baseline network is formed by chaining some/all switching elements of a stage in a Baseline network. To allow SEs to be chained together, a chain-in link and a chain-out line (in addition to existing links) are introduced in each SE to form an augmented switching element (refer to Figure 6A). Figure 6B shows six possible 'non-

conflict' states of an augmented SE. The Chaining may be uniform (refer to Figure 7) or nonuniform (Figure 8).

## III. Modelling the MIN

The MIN could be easily modelled as a directed graph [2] with inputs/outputs and intermediate nodes of a MIN constituting nodes of the resulting graph. Since switches are unreliable and are liable to fail, a $R \times R$ crossbar switch is modelled as a directed graph with $R^2$ links. All links in the graph are unidirectional. Figure 9b shows the graph model of a $4 \times 4$ ESC network of Figure 9A.

For simplicity of analysis, we could assume that each link of the directed graph has a probability $p$ of being operational and a probability $(1-p)$ of being faculty. We also assume that failures are independent of each other. In other words, the failure of any link does not provide any information about the operation/failure of any other link in the network.

We describe our algorithm to compute the reliability of each MIN, considering all possible paths in RMINs.

## IV. Reliability algorithm

Reliability evaluation algorithm for Distributed Computer Systems has been introduced in [8]. Here, we modify it further to compute the terminal reliability of both RMINs and non-RMINs. The algorithm described in this section, GEARMIN (Generalized Evaluation Algorithm for Reliability of MINs), computes the reliability expression of any MIN.

**GEARMIN algorithm.** The GEARMIN algorithm computes the exact reliability expression for paths for any source-destination pair. It takes into account the connectivity of the MIN and generates the reliability expression for each of the different paths possible for the source-destination pair and ensures that the expressions of the different paths are disjoint.

**Assumptions for the model.** The system is modelled as a directed graph $G$, wherein nodes represent the processing elements/intermediate nodes in the network and edges represent the links of the switching elements. The following assumptions are made:

i. The directed graph $G$ does not have any self loops.
ii. Failure of a link is independent of other links.
iii. A link has only two states: operational or faulty.

**Algorithm parameters.** In this algorithm, two vectors, namely, loop vector and reliability vector are used to compute the performance indices. Both these vectors are updated at each traversed node of the graph.

*Loop vector.* The Loop Vector (LV) keeps the information about those nodes that are traversed up to a given level in the tree. LV has $N$ bits.

where $N$ represents the number of nodes in the MIN, and thus has one bit reserved for each node. This vector is used to avoid traversing a node more than once in a subnetwork, thereby avoiding a loop or a cycle and for observing the termination condition. At the starting node, the LV has the bit corresponding to the starting node set to 1.

If a link goes from node $i$ to node $j$, then the updated vector can be written as $LV_j$ = (set $j$th bit in $LV_i$ to 1).

*Reliability vector.* A Reliability Vector (RV) maintains the information about the reliability expression. It is composed of $L$ tuples, where $L$ represents the number of links in the MIN and thus has one tuple for each link. RV is initialized with $L$ zeros at the root of the tree. A tuple in the RV can have one of the three values $\{0, 1, d\}$. A '$d$' in a tuple implies that the communication link represented by this tuple is not in the reliability expression and a '1' ('0') indicates that the link is a part of the reliability expression and is operational (faulty).

There are two cases while performing reliability vector updating:

Case 1: If the current node is the first son of its parent, then $(RV)_{old} = (RV)_{parent}$.

Case 2: If the current node has a brother, then $(RV)_{old} = (RV)_{brother}$.

The updating of the RV in both cases is done as follows:

$(RV)_{new}$ = Set tuple (i) corresponding to link xi to '1' in $(RV)_{old}$.

$(RV)_{brother}$ = Set tuple (i) corresponding to link xi to '0' in $(RV)_{old}$.

This updating process generates disjoint reliability terms represented by RV at any level of the tree.

**Procedures of the algorithm.** The algorithm has five main procedures:

*Initial step.* {It does all the initialization.}

*Adjacent.* {It finds all the adjacent links from current node $k$ to all nodes indicated in TAL (Temporary Adjacency List described in procedure 'Generate'). If all links from the current node lead to nodes which have no adjacencies and none of which is the destination node, we discard the path to the current node and call procedure 'modify.'}

*Generate.* {It finds all forward and parallel paths from current node}

1a. It sets TAL to the list of adjacent nodes with respect to current node for computing forward paths, and calls the procedure 'adjacent.'

1b. It sets TAL to the list of possible previous nodes, whose adjacent links could result in parallel paths, in this subnetwork, and calls the procedure 'adjacent.'

*Expansion.* {It expands the reliability and loop vectors described earlier and observes for termination condition.}

*Modify.* {It observes the reliability expression for the path being discarded, and modifies the reliability expressions of other paths, if necessary. This implies that if any link $x$, is set to '1' *only* in the reliability expression of the path being discarded, then the $x$th bit set to '0' in the reliability expression, for any other path should be changed to a '*d*.'}

**Algorithm description.** The algorithm starts at a given node and travels along the graph. Using the procedure 'adjacent,' the procedure 'generate' obtains the nodes which can be reached from any given node. The procedure 'expansion' updates both vectors described above. Thus, the reliability expression for each path is generated and updated as we move along the graph. The algorithm, in parallel, explores all possible paths from a source to a given destination, and by the very nature of the updating process of the reliability vector, the reliability expressions of different paths are kept disjoint. Figure 10 illustrates the operation of the algorithm for the network shown in Figure 9A.

**Algorithm termination.** The expansion of the tree at any node terminates if one of the following rules is satisfied at the current node:

> Rule 1: The loop vector has the bit corresponding to the destination set to '1.'
>
> Rule 2: Rule 1 is not met, but the current node has no 'adjacencies,' i.e., no further expansion is possible from the current node.

Only those nodes terminated by Rule 1 are considered for reliability expression evaluation. If, at any node, Rule 2 is applied, the reliability expressions of the other paths are modified, if necessary, using the procedure 'modify.'

The algorithm terminates when expansion of the tree stops at all possible nodes.

## V. Using GEARMIN for computing the reliability of MINs

The MIN is modelled as a directed graph (di-graph) with processors/intermediate nodes of the MIN as nodes in the di-graph, and the links of the switches are represented as directed links in the graph. To find the reliability term for all possible paths from a given source to a destination, we choose an arbitrary starting node as the source. We observe the loop vector to ensure that the required destination is reached. When the bit (corresponding to the required destination) is set to '1' in the loop vector, it can be concluded that the destination has been reached. Thus the modelling of the MIN generates all possible paths in the MIN for a given source-destination pair and simultaneously generates an expression for the reliability of the paths. We assume that all nodes of the graph are perfectly reliable and that links, being a part of the switches, are liable to fail.

In Figure 10, we show the complete tree for computing the reliability ex-

pression from node 0 to node 12 (Figure 9B). The expansion process of each node along the adjacent links and updating of the two vectors is evident from the tree structure of Figure 10. We note that at node 14, we terminate by Rule 2 and hence procedure 'modify' is called. At node 14, bit 20 is set to '1' only in this path's RV, but bit 20 is not set to '0' in any other RV. Hence no other RV has to be modified. No further expansion is possible at node 18, since the procedure 'adjacent' observes that node 18 leads to nodes 14 and 15, both of which have no adjacencies and neither is a destination node. Procedure 'modify' is invoked again, but no modification to any other RV is needed, because of the nature of the RV at node 18.

## VI. Test data and results

The technique described above has been tested for the RMINs described in Section II. For the INDRA network, the generalized cube has been chosen as the subnetwork. The reliability of 16 Input-16 Output and 32 Input-32 Output Networks has been computed and the reliability for a network with larger number of input/outputs could be easily computed in a similar way. Tables I and II show the results obtained and the relative costs of the networks in terms of number of switches used and the switch sizes. We have assumed a probability of 0.9 for an operational link and 0.1 for a faulty link.

## VII. Interpretation of results

Various redundant MINs considered earlier differ in the cost and the performance. From Tables 1A and 1B, we see that the $F$ network has the highest reliability and fewer number of switches but is of size $4 \times 4$. The nonuniform Chained Baseline approaches close to the $F$ network reliability, while using smaller size crossbar switches ($3 \times 3$). The Multipath Omega Network rates better in its 32 Input-32 Output network because of the network configuration selected. The 16 Input-16 Output Multipath Omega Network is a 4-stage network in the (4, 2, 2, 4) shuffle configuration. Hence it has 2 stages of $4 \times 4$ switches and 2 stages of $2 \times 2$ switches. On the other hand, the 32 Input-32 Output Multipath Omega Network is only a 3-stage network using $4 \times 4$ switches only.

The algorithm GEARMIN described in this paper can compute an exact reliability for any MIN (redundant and nonredundant), of any size. Our algorithm needs to update only two vectors at each node of the tree and at any particular node, no information regarding parents/ancestors is needed. We store only leaves of the tree after each expansion and expand along these leaves. Hence GEARMIN can work for large networks as well. This overcomes the drawbacks of work done described by Blake [3, 4] and Botting [5]. The availability of the system described by Das [7] can easily be computed by merely substituting the probability of the faulty links as '0' in the reliability expression obtained by GEARMIN.

## VIII. Conclusions

The algorithm GEARMIN, described in this paper, is an analytical method to compute the exact reliability for any MIN of any size. It employs simple bit operations to derive the reliability expression of each source-destination pair and hence is not compute-intensive. Previous work in the reliability of MINs gives only lower and upper bounds for large networks and no analytical approach leading to exact value of the reliability has been previously described.

TABLE 1A. 16 Input-16 output networks

| Networks | Nodes | Links | Switches | Switch size | Reliability |
|---|---|---|---|---|---|
| Extra Stage Cube | 128 | 224 | 72 | 2X2 | 0.645704 |
| F | 64 | 192 | 48 | 4X4 | 0.882090 |
| IADM | 80 | 224 | 64 | 3X3 | 0.7617724 |
| INDRA | 288 | 768 | 16<br>128 | 4X4<br>2X2 | 0.656100 |
| Multipath Omega | 80 | 192 | 8<br>16 | 4X4<br>2X2 | 0.793881 |
| Uniform CB | 104 | 248 | 24<br>8 | 3X3<br>2X2 | 0.850209 |
| Nonuniform CB | 104 | 248 | 24<br>8 | 3X3<br>2X2 | 0.866780 |

TABLE 1B. 32 Input-32 output networks

| Networks | Nodes | Links | Switches | Switch size | Reliability |
|---|---|---|---|---|---|
| Extra Stage Cube | 288 | 512 | 160 | 2X2 | 0.581134 |
| F | 160 | 578 | 128 | 4X4 | 0.873269 |
| IADM | 192 | 448 | 160 | 3X3 | 0.717747 |
| INDRA | 704 | 1792 | 32<br>320 | 4X4<br>2X2 | 0.585100 |
| Multipath Omega | 128 | 384 | 24 | 4X4 | 0 801900 |
| Uniform CB | 256 | 640 | 64<br>16 | 3X3<br>2X2 | 0.839211 |
| Nonuniform CB | 256 | 640 | 64<br>16 | 3X3<br>2X2 | 0.84401 |

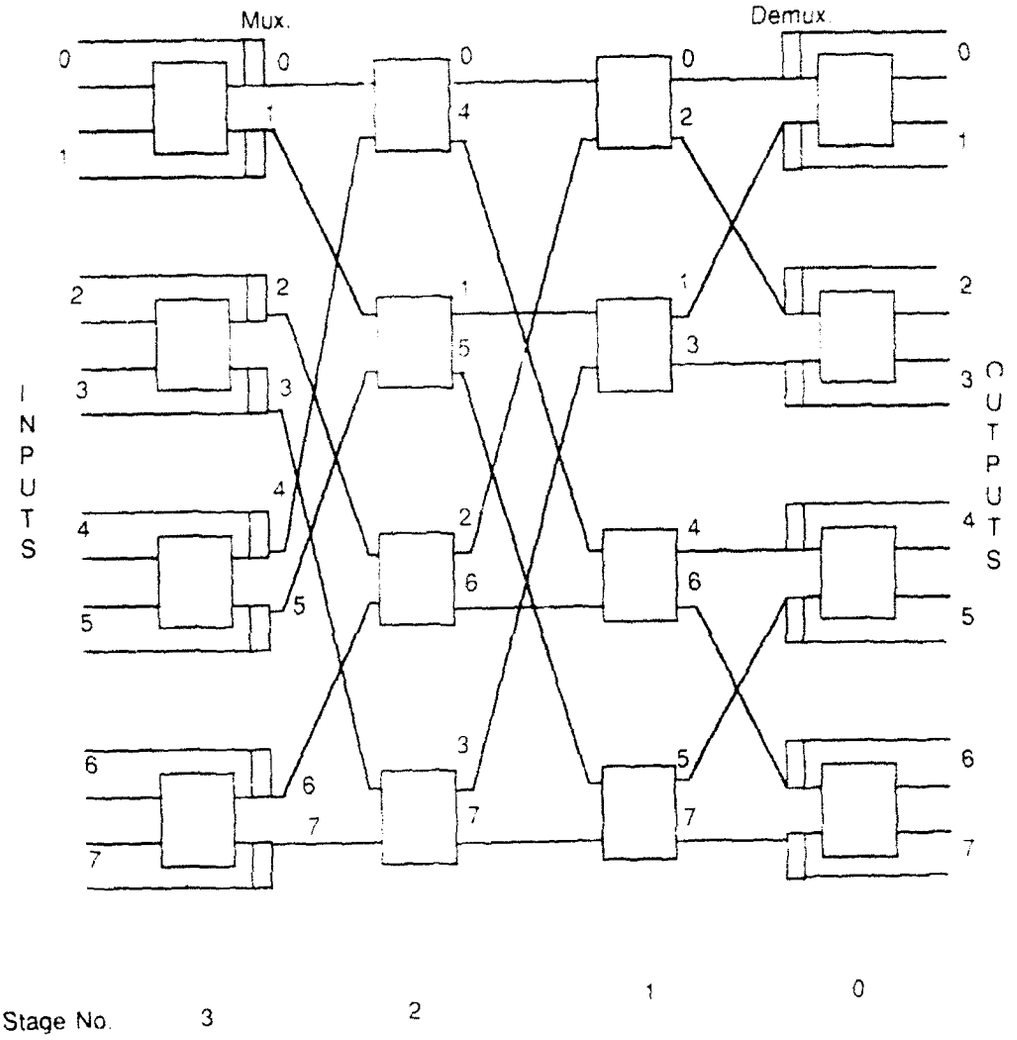


FIGURE 1.
The extra stage cube network for $N = 8$.

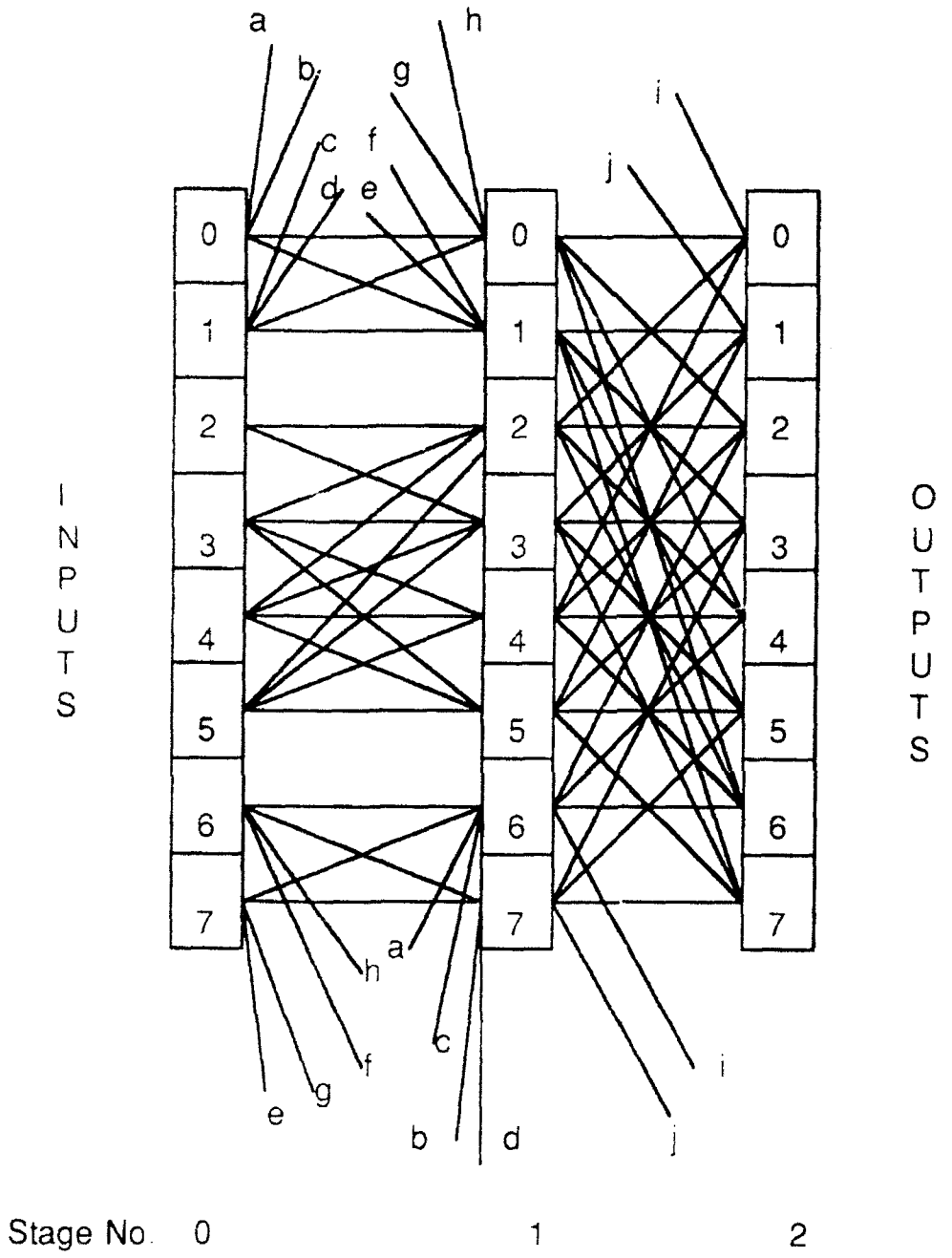


FIGURE 2. The $F$ Network for $N = 8$.

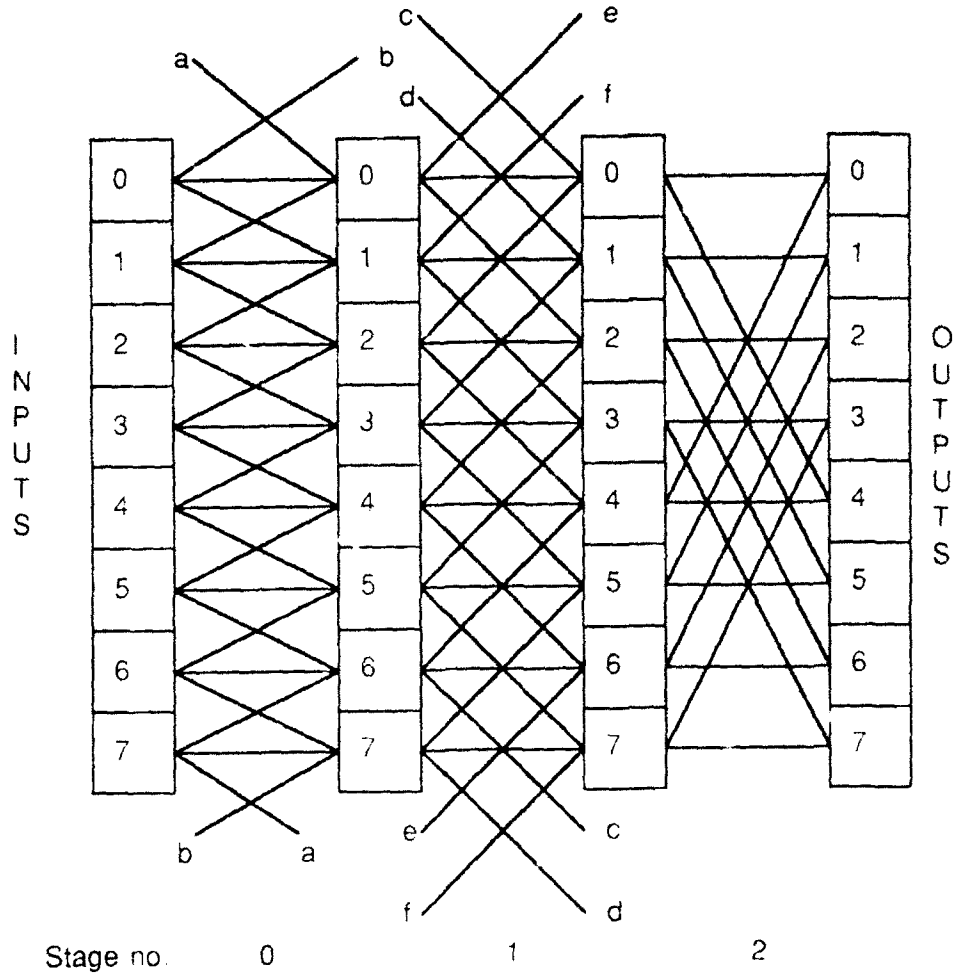


FIGURE 3. The IADM Network for $N = 8$.

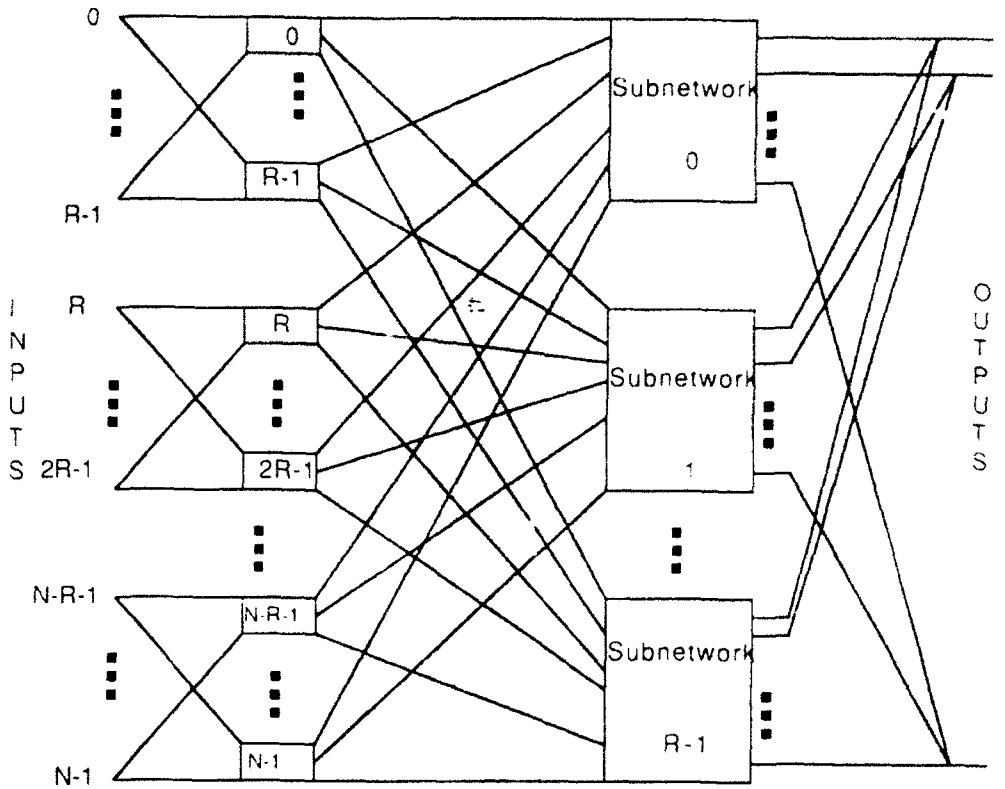


FIGURE 4. The INDRA Network Shown as the Union of $R$ Subnetworks.

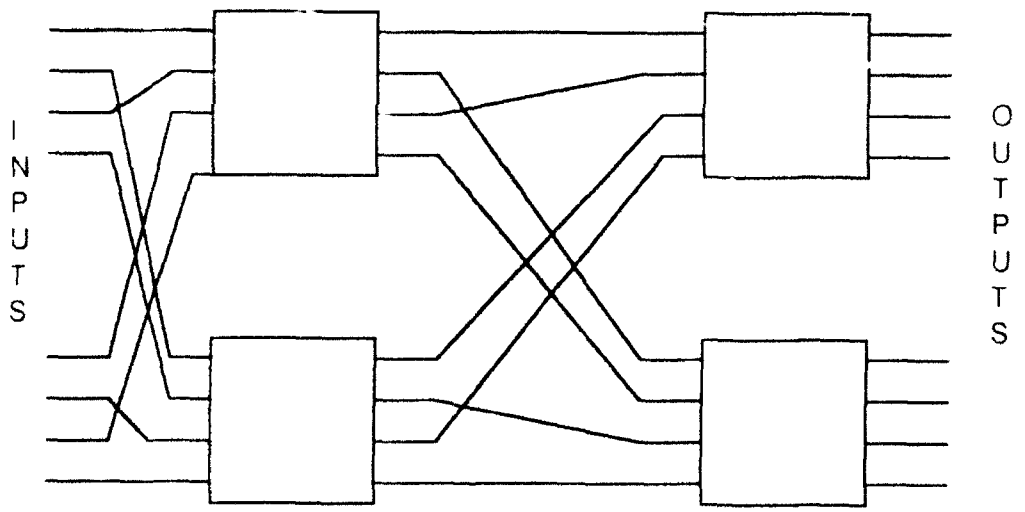


FIGURE 5. Multipath Omega Network for $N = 8$.

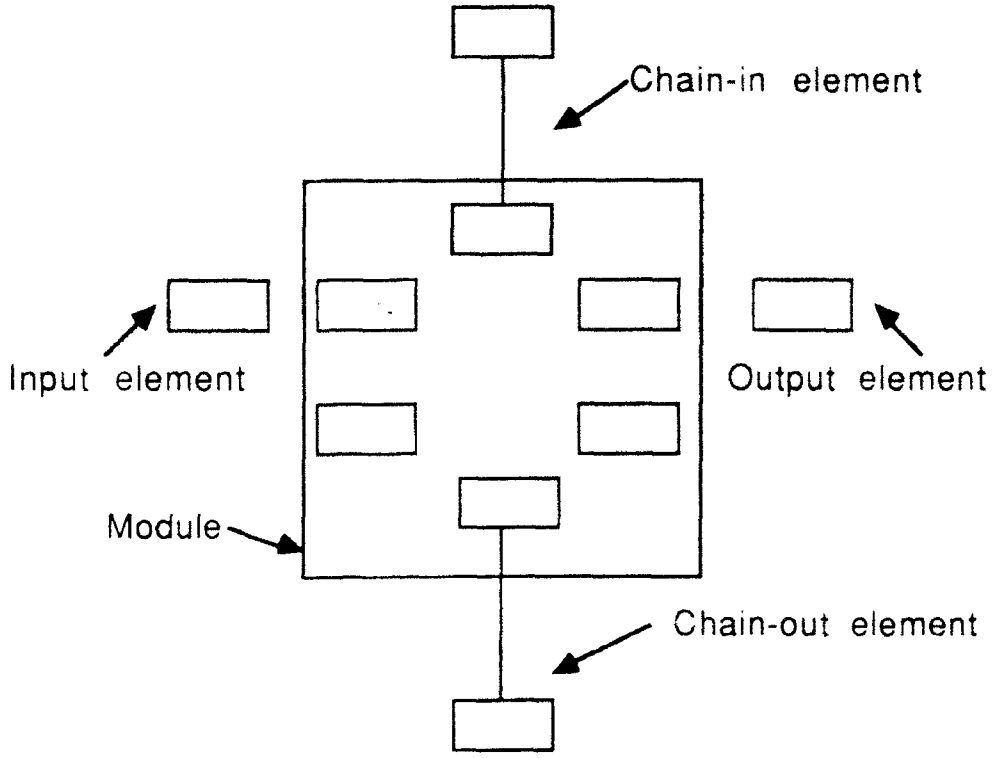


FIGURE 6A. An "Augmented" Switching Element of a Chained Baseline Network.

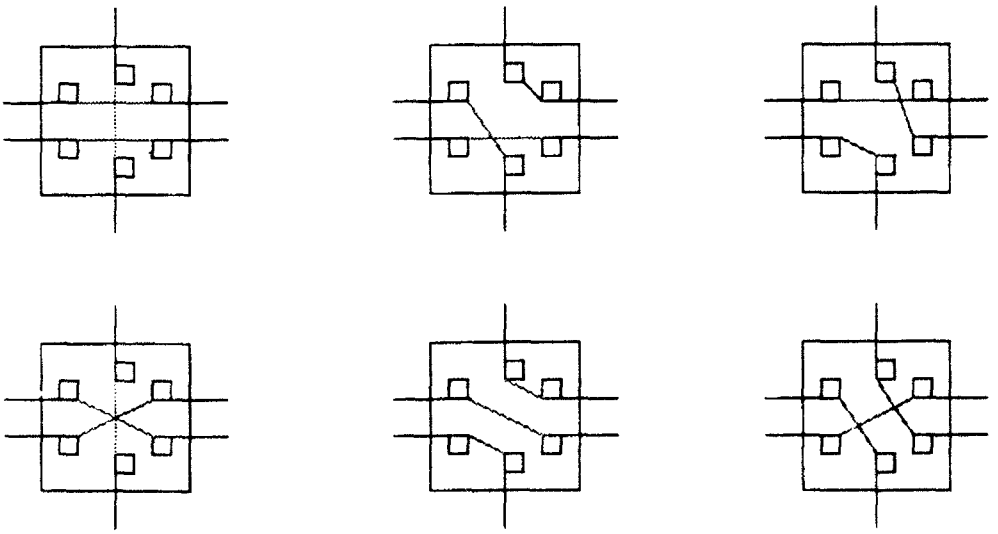

FIGURE 6B. Six Possible 'nonconflict' states in an augmented SE.

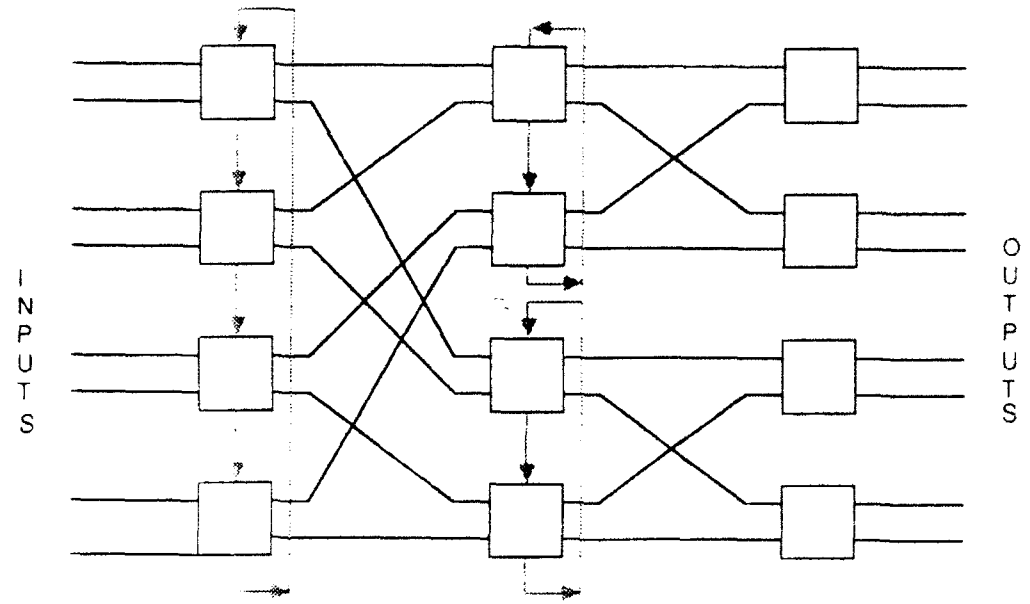


FIGURE 7. Uniform Chained Baseline Network for $N = 8$.

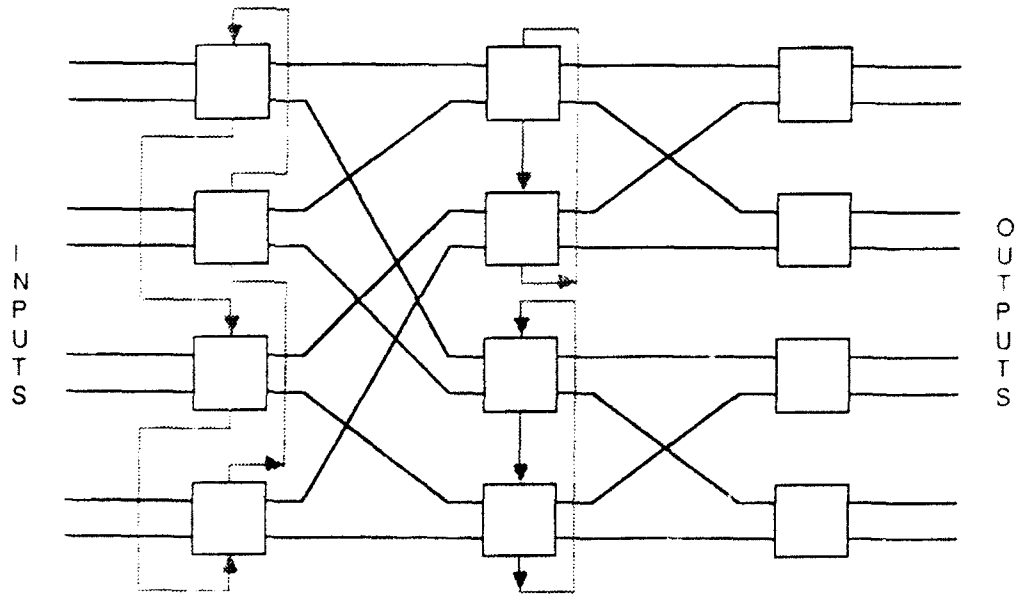


FIGURE 8. NonUniform Chained Baseline Network for $N = 8$.

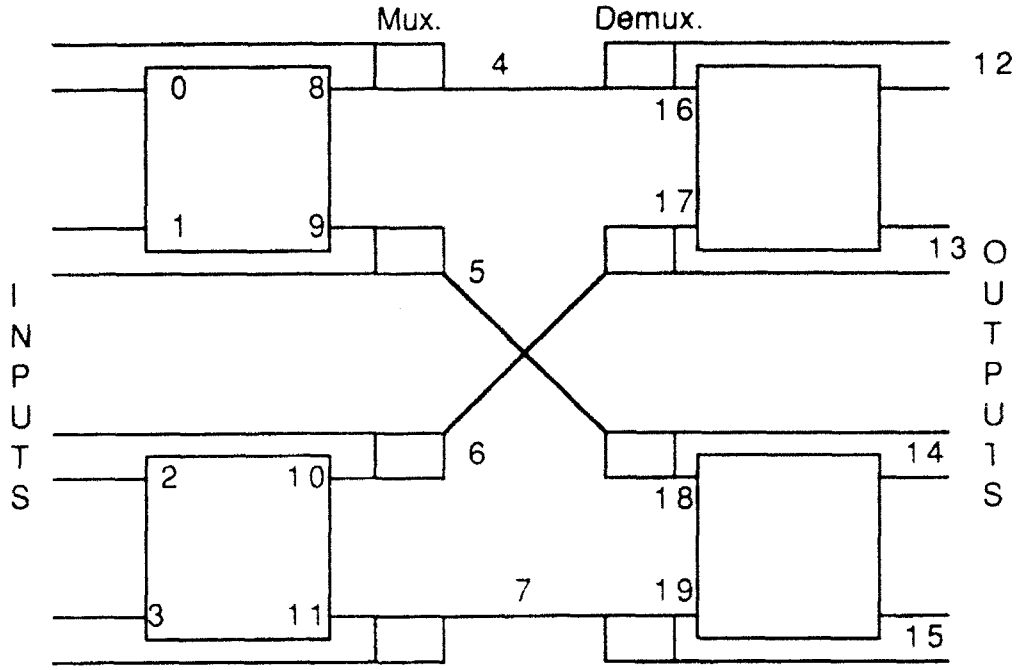


FIGURE 9A. Extra Stage Cube for $N = 4$.

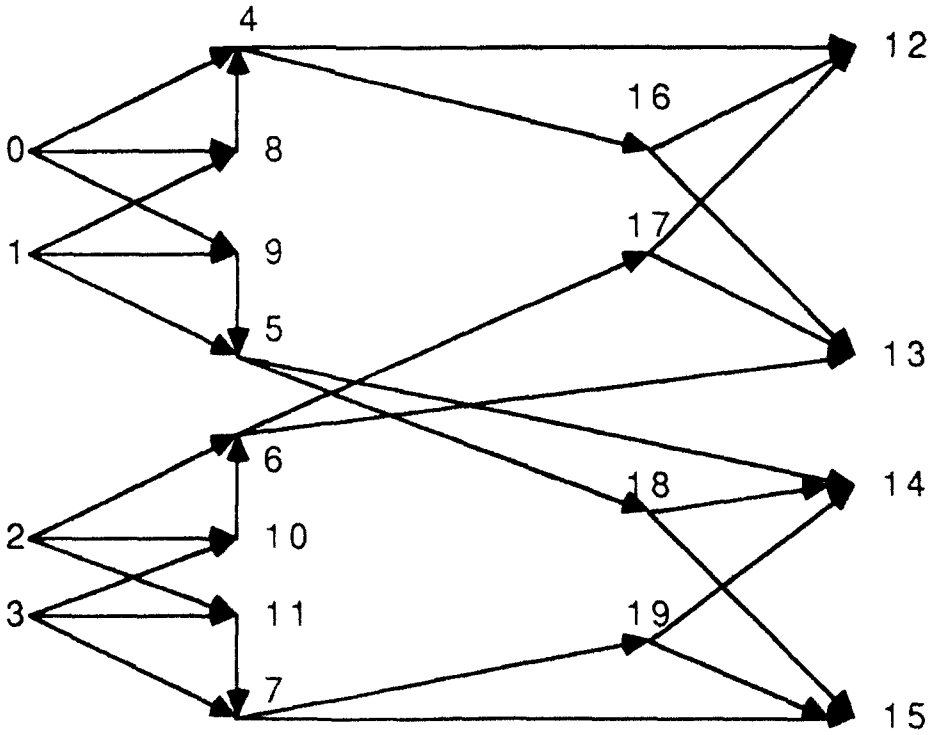


FIGURE 9B. Graph Model for Figure 9A.

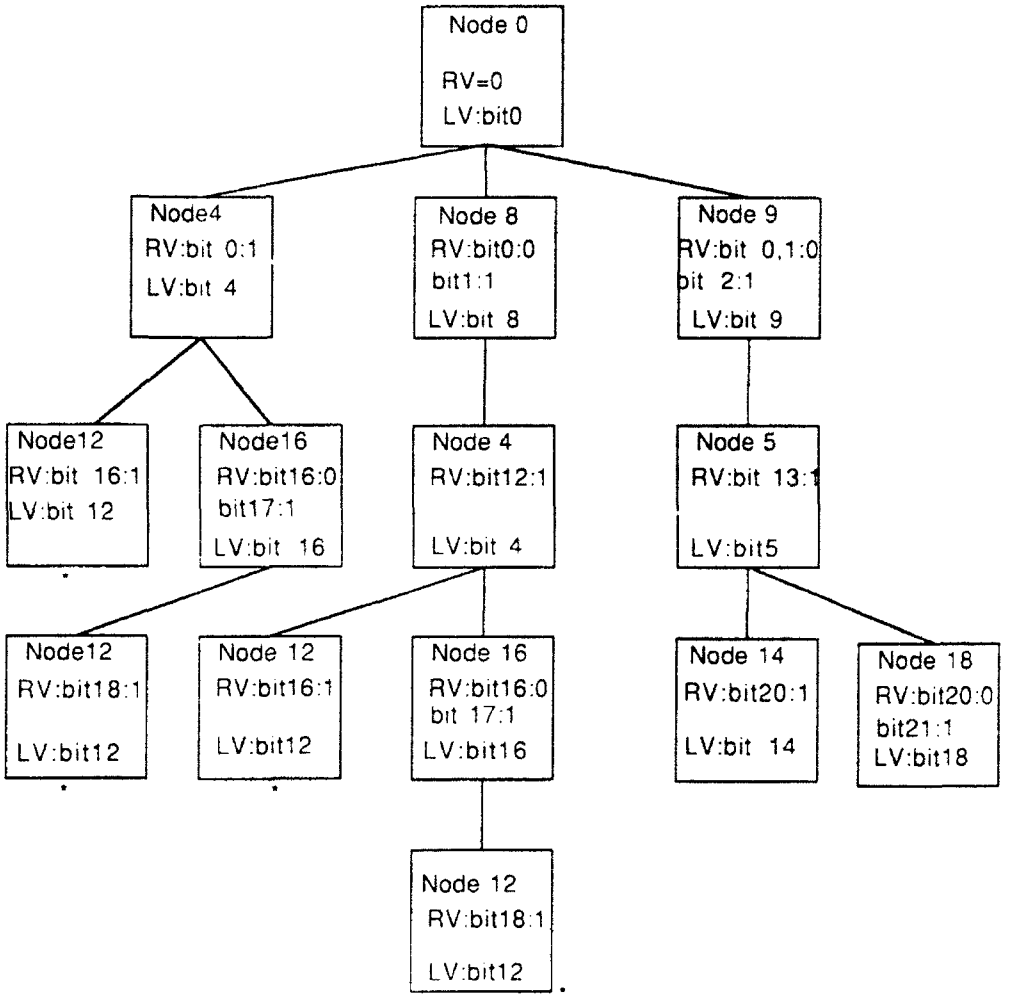


FIGURE 10. Complete tree for GEARMIN for graph of Figure 9B
Only updated bits shown for each child

* Leaves used for reliability expression calculation

| | |
|---|---|
| Link 0: Node 0 to Node 4 | Link 16: Node 4 to Node 12 |
| Link 1: Node 0 to Node 8 | Link 17: Node 4 to Node 16 |
| Link 2: Node 0 to Node 9 | Link 18: Node 16 to Node 12 |
| Link 12: Node 8 to Node 4 | Link 20: Node 5 to Node 14 |
| Link 13: Node 9 to Node 5 | Link 21: Node 5 to Node 18 |

Reliability $= p_0p_{16} + p_0q_{16}p_{17}p_{18} + q_0p_1p_{12}p_{16} + q_0p_1p_{12}q_{16}p_{17}p_{18}$ where $p_i$ = Prob. of operational link $i$ and $q_i$ = Prob. of faulty link $i$.

## REFERENCES


1. G. B. Adams III and H. J. Siegel, *The Extra Stage Cube: A Fault-Tolerant Interconnection Network for Supersystems*, IEEE Trans. Computers, Vol. C-31, No. 5, May 1982, pp. 443–454.
2. D. P. Agrawal, *Graph Theoretical Analysis and Design of Multistage Interconnection Networks*, IEEE Trans. Computers, Vol. C-32, No. 7, July 1983, pp. 637–648.
3. J. T. Blake and K. S. Trivedi, *Reliability Analysis of Interconnection Networks using Hierarchical Composition*, IEEE Trans. Reliability, Vol. 38, No. 1, April 1989, pp. 111–120.
4. J. T. Blake and K. S. Trivedi, *Multistage Interconnection Network Reliability*, IEEE Trans. Computers, Vol. C-38, No. 11, November 1989, pp. 1600–1603.
5. C. Botting, S. Rai, and D. P. Agrawal, *Reliability Computation of Multistage Interconnection Networks*, IEEE Trans. Reliability, Vol. 38, No. 1, April 1989, pp. 138–145.
6. L. Ciminiera and A. Serra, *A connecting network with Fault-Tolerance Capabilities*, IEEE Trans. Computers, Vol. C-35, No. 6, June 1986, pp. 578–580.
7. C. R. Das and L. N. Bhuyan, *Reliability Simulation of Multiprocessor Systems*, Proc. 1985 Intl. Conf. Parallel Processing, pp. 591–598.
8. A. Kumar and D. P. Agrawal, *Generalized Algorithm for Reliability Evaluation of Gracefully Degradable Systems*, submitted for publication.
9. J. J. Macaluso, C. R. Das, and W. Lin, *A Reliability Prediction for MIN-Connected Multiprocessor Systems*, Proc. 1988 Intl. Conf. Parallel Processing, pp. 392–399.
10. R. J. McMillen and H. J. Siegel, *Performance and Fault Tolerance Improvements on Inverse Augmented Data Manipulator Network*, 9th Intl. Symposium Computer Architecture, April 1982, pp. 63–72.
11. K. Padhamanabhan and D. H. Lawrie, *A Class of Redundant Multistage Interconnection Networks for Multiprocessors*, IEEE Trans. Computers, Vol. C-32, No. 12, December 1983, pp. 1099–1108.
12. C. S. Raghavendra and A. Varma, *INDRA: A class of Interconnection Networks with Redundant Paths*, 1984 Real Time Systems Symposium, Computer Soc. Press, Silver Spring, Md., 1984, pp. 153–164.
13. N-F Tzeng, P-C Yew, and C-Q Zhu, *A Fault Tolerant Scheme for MINs*, 12th Intl. Symp. Computer Architecture, June 1985, pp. 368–375.
14. A. Varma and C. S. Raghavendra, *Reliability Analysis of Redundant Path Interconnection networks*, IEEE Trans. Reliability, Vol. 38, No. 1, April 1989, pp. 130–137.



DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING, NORTH CAROLINA STATE UNIVERSITY, RALEIGH, NORTH CAROLINA 27695-7911

ENGINEERING MATHS AND COMPUTER SCIENCE DEPARTMENT, UNIVERSITY OF LOUISVILLE, LOUISVILLE, KENTUCKY 40292

E-MAIL: A0KUMA01%UKLYVX.BITNET

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING, NORTH CAROLINA STATE UNIVERSITY, RALEIGH, NORTH CAROLINA 27695-7911

E-MAIL: DPA@CSL.NCSU.EDU