2009 IEEE Symposium on Industrial Electronics and Applications (ISIEA 2009), October 4-6, 2009, Kuala Lumpur, Malaysia

Reliability of Recursive Concentrator

Sa'ed Abed
Computer Engineering Department
Hashemite University
Zarqa, Jordan
E-mail: sabed@hu.edu.jo

Sahel Alouneh
Computer Engineering Department
German-Jordanian University
Amman, Jordan
Email: sahel.alouneh@gju.edu.jo

Otmane Ait Mohamed
Electrical and Computer Engineering Department
Concordia University
Montreal, Quebec, Canada
E-mail: ait@ece.concordia.ca

***Abstract*—The inverse omega network possesses various attractive properties and its constituent node has a fixed degree independent of the system size. In this paper, a proof for a technique applied recursively to construct an N-input concentrator switching network which is similar in topology to the inverse omega network will be introduced. Also an examination of a reliable inverse omega network in the presence of any single node failure will be presented. The new configuration of the system ensures providing the same high performance results obtained before the failure. In this paper, we show how to check the passibility by the inverse omega network of any given connection set and list some of the general patterns passable by this network. We also show that the concentrate operation passable by the inverse omega network is just a special case of the more general alternate sequence operation.**

***Keywords*—Inverse omega network; recursive concentrator structure; fault tolerance; reliability index;**

I. INTRODUCTION

The Recursive concentrator structure gives an approach to demonstrate the concentration property of the inverse omega network. It shows a method that employs a stage of 2×2 switching elements and two $N/2$ input concentrator to construct an *N*-input concentrator. The method is applied recursively to yield an *ln*(N)-stage network that is topologically similar to the inverse omega network [1,2].

The inverse omega network has many attractive features like simple routing, low diameter, and good support for communication patterns generated by numerous algorithms. In addition, every node in inverse omega network requires a constant degree, independent of the system size. The property of a fixed degree is extremely advantageous, because it makes possible the use of a single type of building block for constructing a large system based on inverse omega topology [3].

There are a few research reports on fault-tolerant butterfly, binary, and inverse omega networks used in the switching context [4,5], insufficient attention has been given to the reliable design of inverse omega network. In the former situation, the fault-tolerant issue lies in assuring the existence of a path between every network input and output; whereas it requires that all the inverse omega nodes be connected in the rigid form. This paper focuses on the later fault-tolerance situation.

This paper is organized as follows. Section II briefly reviews the concentrator structure followed by a proof that inverse omega network is capable of performing the concentration function. Section III describes the inverse omega architecture, the proposed reliable inverse omega design, and the reconfiguration procedure. Section IV reviews the closest related work. Finally, Section V concludes our paper and presents the future work.

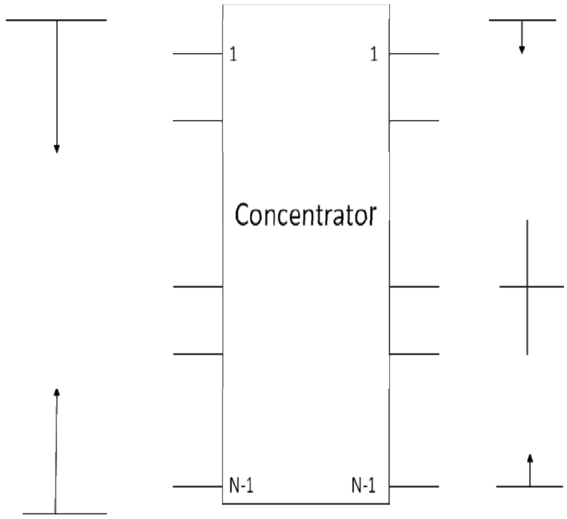

Figure 1. An $N\times N$ concentrator

I. CONCENTRATOR STRUCTURE

A concentrator network has means for separating a cell to be routed to the output port from a cell to be either re-circulated or discarded, this means including a smart routing algorithm which routes cells based on a priority setting in a routing bit or bits located in a header of the cell.

Figure 1 shows the concentrator of a $N\times N$ network, its function is to route all the active inputs to the top end of the network outputs.

The N-input concentrator can be constructed from the two $N/2$ input concentrators and a switching stage as shown in Figure 2. In this method, the states of the front-end 2×2 switches are set so that one-half of the active inputs are routed to the upper concentrator. This action forces the other half to trickle down to the lower concentrator.

The outputs from the two $N/2$ input concentrator are interleaved to obtain the final concentrated output. By decomposing each of the $N/2$ input concentrator's recursively, a network consisting of *ln*(N)-stage of 2×2 switching elements is obtained.

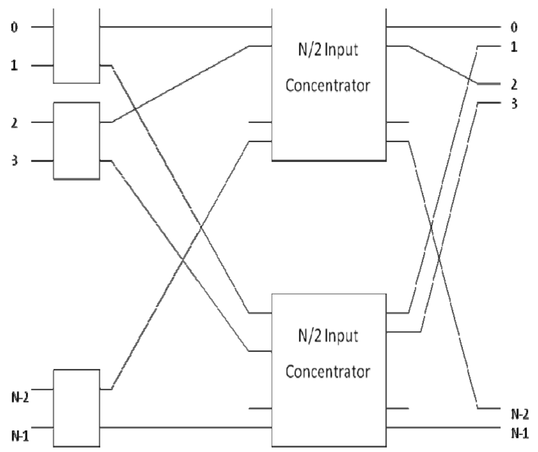

Figure 2. Recursive construction of N input concentrator

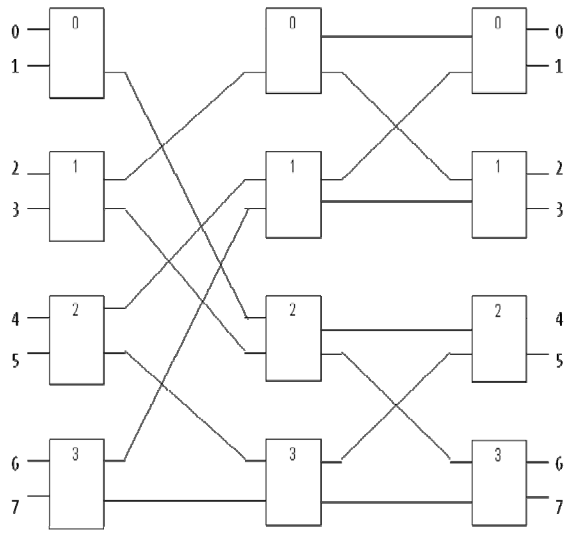

Figure 3. An 8-input concentrator

The complete decomposed concentrator for eight inputs is shown in Figure 3. The sequence of output interleaving implied by the recursion corresponds to the bit-reversal permutation between the outputs of final switching stage and the network outputs, which is topologically similar to the inverse omega network.

A. The Proof

The given proof here states that the inverse omega network is capable of performing the concentration function, and shows the similarity between their constructions.

The equivalence can be demonstrated by two ways:

1. Swapping the positions of the switching elements 1 and 2 in the last stage, so the resulting network is shown in Figure 4. It has *ln*(N)-stage, each stage consisting of an exchange permutation and an unshuffle permutation.
2. If we look to Figure 3, we notice that for an 8-input we have a concentrator consist of $n = ln(N) = 3$ stages and $N/2 = 8/2 = 4$ switching elements in each stage, so for this concentrator we have a formula given by:

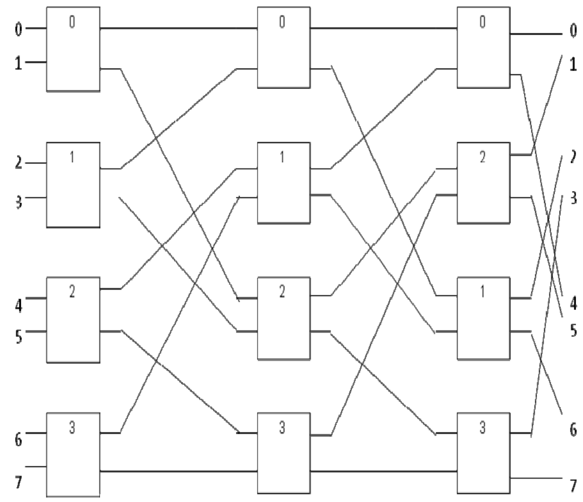

Figure 4. The 8-input concentrator of Fig. 3 is redrawn to demonstrate its topological similarity to the inverse omega network

$$E\sigma^{-1}EB_2E\rho$$

And for the inverse omega network for $n = 3$ are given by:

$$\prod_{i=1}^{n} E\sigma^{-1} = E\sigma^{-1}E\sigma^{-1}E\sigma^{-1}E$$

Let the input $X = b_3b_2b_1$ and apply X for both networks, we get:

For *Concentrator Network*

$$\sigma^{-1} = b_1b_2b_3$$
$$B_2 = b_1b_2b_3$$
$$\rho = b_3b_2b_1$$

For *Inverse Omega Network*

$$\sigma^{-1} = b_1b_2b_3$$
$$\sigma^{-1} = b_2b_1b_3$$
$$\sigma^{-1} = b_3b_2b_1$$

So by using the permutation analysis shown above, we notice that both networks yield the same output for the same input, so they are similar.

II. SYSTEM FAULT TOLERANCE

A. Proposed Reliable Concentrator Design

The second part of this paper shows a method on how to tolerate the system if one node is failed and how the system reconfigures itself in order to isolate the failed node and the system continue its work. An essential design consideration of large network system reliability is taken under consideration, i.e. when the system size grows; the probability of having all system components fault-free during an operation falls quickly and could reach unacceptably low point.

Thus, it is necessary to incorporate redundancy in the system design to ensure proper continuing operation even after some components fail, improving reliability. As soon as a failure is raised and detected, a failure-tolerant system

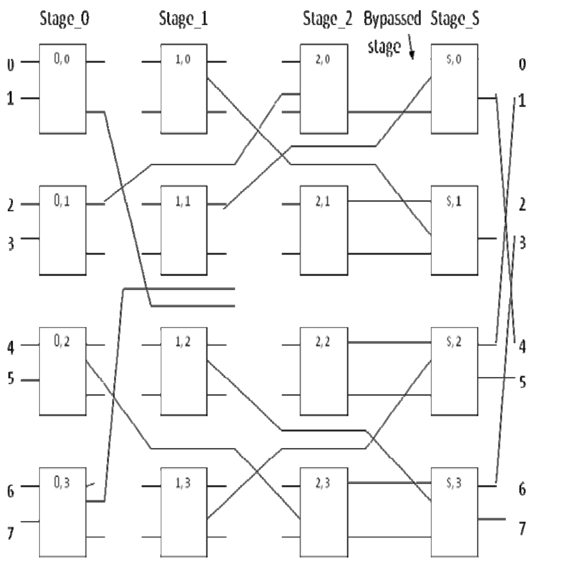

Figure 5. The fault-tolerant by adding extra links and spare nodes

reconfigures itself to isolate the failed node and so the system assured to deliver the same high performance before any failure of the nodes in the system.

In order to ensure that the machine maintains its full strict structure in spite of the fault, we may:

- Increase the connections between levels in a way that if a direct link exists between nodes (g_1, l_1) where g_1 represents the stage number 1 and l_1 represents the first level of that stage and (g_2, l_2).
- Adding extra nodes to the right end of every level as (*S*, *l*).
- Using more control switches.

If a direct link exists between nodes (g_1, l_1) and (g_2, l_2), then an extra link is placed between nodes (g_1, l_1) and (g_{2+1}, l_2) as shown if Figure 5. This addition of extra links and nodes will tolerate the system if one node is failed. Every node has degree 4, two links connected to nodes in the preceding stage and the other two links connected to the nodes in the next stage. By adding extra links, every node will have a degree of 6.

For example, a direct link is added between nodes (0, 0) and (2, 2), since there is a direct connection existing between (0, 0) and (1, 2) in the original network. Similarly, an extra links are added between node (1, 3) and (s, 1) and so on.

Node (1, 1) is connected to node (0, 2) and node (0, 3) of stage (0) which is connected to node (2, 1) of stage (2), so by adding a direct connection between node (0, 3) and (2, 1) and another one between (0, 2) and (2, 3) the system can be tolerated if node (1, 1) fails, because node (2, 1) will replace it and a connection is set between (2, 1) and (2, 0) by using a control switch set to the (V) state as shown in Figure 6.

Four switches are added to each node so that a failure or unused node can be bypassed and the replacement of the failed node will be for the node in the same level of the next stage and then will be bypassed to the other stages until the last stage. The interconnections using the control switch which can be set at either (V) or (X) state as shown

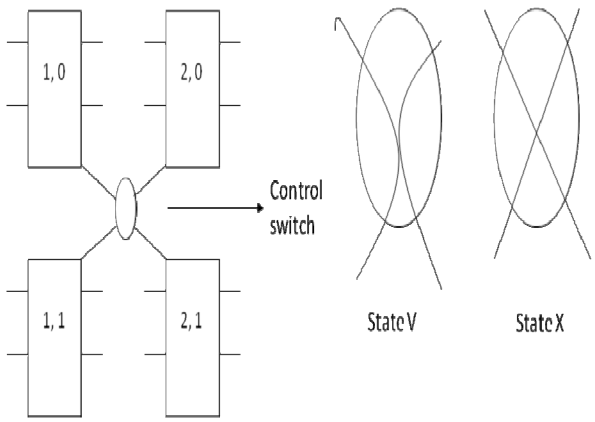

Figure 6. The control switch

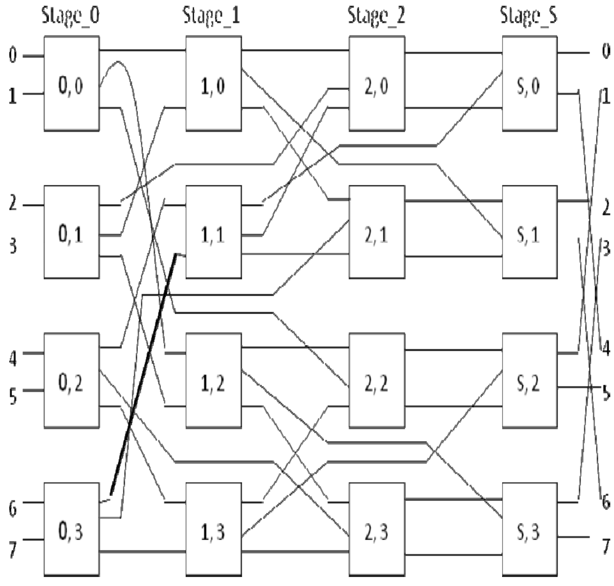

Figure 7. The system after the addition of links and a stage of nodes

above. The control switch used is normally much simpler than a node and can be made more reliable or fabricated with a node in the same module to get high reliability.

B. Tolerate Node Failure

This design only tolerates any single node failure, so when a spare node fails, it is bypassed and no reconfiguration takes place since the spare in each level is bypassed initially. If a fault happens to an active node, the failed node is bypassed and its role is replaced to its physical successor (to its right in the same level) which in turn is replaced by its successor and etc., until the spare in the level is brought into the system.

Each level can tolerate up to one node failure. After a node, say (*g*, *l*), failed and the system reconfigures a subsequent node failure may not be tolerable even when it is at a level other than 1.

As an example, if node (0, 2) try to use (1, 1), the connection between (0, 2) and (2, 0) is set and the control switch between (2, 1) and (2, 0) will be in the (V) state and so the node (2, 1) is replacing node (1, 1) and work correctly as shown in Figure 7 and Figure 8.

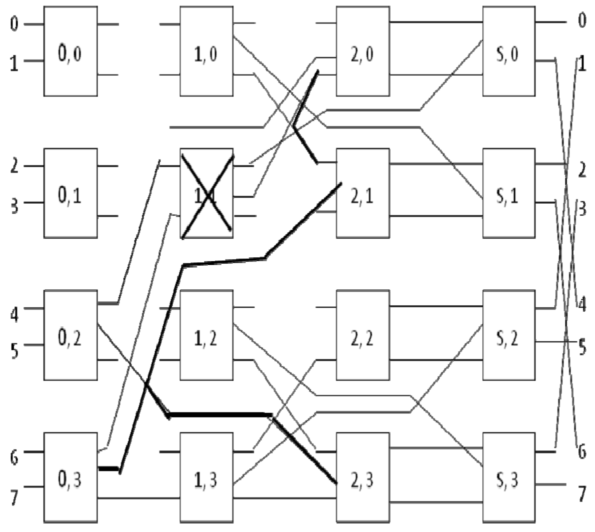

Figure 7. Tolerant node (1, 1)

I. RELATED WORK

Narasimha [1] demonstrated the construction of a multistage network that performs a superset of the functionality of a (*n, m*)-concentrator. Narasimha's network is constructed recursively. An *n* sized network is constructed from two $n/2$-input networks. The outputs of these networks are then interleaved. The recursive construction proceeds until the required network is reached, at which point a simple crossbar is used.

Research has been done on the problem of unordered networks [6,7]. A specific class of unordered network called concentrators has been defined which allow a number of inputs to connect to a smaller number of outputs. However, while some concentrator constructions are lower depth than permutation networks in all cases, they are not lower cost (area) than permutation networks for all possible configurations. This makes the choice of network architectures difficult because the network size (the number of inputs and outputs), and relative importance of cost and depth must be considered before choosing an architecture.

Theoretical research has shown that it is possible to implement an *n*-input concentrator with *O(n)* crosspoints for sufficiently large *n* [6]. In contrast, it has been shown that a rearrangeably non-blocking *n*-input permutation network must have at least *O(nlgn)* crosspoints [8]. This result highlights a fundamental difference between the two networks.

Self-routing interconnection networks with their low processing-overhead delay and decentralized routing, are an attractive option for switching fabrics in high speed networks. These interconnection networks, however, realize only a subset of all possible input-output permutations in a non-blocking fashion [9]. The non-blocking property of these networks is an extensively studied area in interconnection network theory field and efficient algorithms exist to check if any given permutation is passable by such networks without blocking. One of the most common interconnection network structures is the inverse omega network and is topologically equivalent to the reverse banyan network.

In [10], the authors demonstrated the difference in cost and depth of permutation networks and concentrator networks for a variety of network sizes. They also presented a new concentrator network construction that is lower cost and lower depth than a permutation network for all network sizes.

V. CONCLUSIONS

The concentration capability of the inverse omega network has been demonstrated starting from a recursive decomposition of the basic concentrating network and show the similarity between them.

Also a reliable inverse omega network design has been presented. The design is made to maintain the full rigid inverse omega interconnection style even in the presence of faults, so that a reconfigured system can deliver the same high performance. Reconfiguration in response to an arising fault is simple, local, and involves only a small fraction of system nodes. The proposed design tolerates any single node failure, significantly enhancing the system reliability. Tolerating the system for multiple node/link failures will be studied in future research. An extension to this reliable design with a simulation will be considered in the future work.

REFERENCES

[1] M. Narasimha, "A Recursive Concentrator Structure with Applications to Self-Routing Switching Network," *IEEE Trans in Communications*, 42(8): 896-899, August 1994.

[2] A. Huang and S. Knauer, "Starlit a Wide Band Digital Switch," In Proceedings of GLOBECOM '84, pp. 121-125, Nov. 1984.

[3] N. Tzeng, "Reliable Butterfly Distributed-Memory Multiprocessor," *IEEE Trans in Computers*, 43(9): 1004-1013, 1994.

[4] B. Grey, A. Aviziennis, and D. Rennels, "A Fault-Tolerant Architecture for Network Storage Systems," *In Proceedings 14th Int. Symp. Fault-Tolerant Computing*, pp. 232-239, Nov. 1984.

[5] G. B. Adams, IH and H. J. Siegel, "The Extra Stage Cube: A Fault-Tolerant Interconnection Network for Super Systems," *IEEE Trans in Computers*, Vol. C-31: 443-454, Jan. 1984.

[6] M. S. Pinsker, "On the complexity of a concentrator," *In Proc. 7th Int. Teletraffic Congr.* Stockholm, Sweden, pp. 318/1-318/4, 1973.

[7] F. R. K. Chung, "On Concentrators, Superconcentrators, Generalizers, and Nonblocking Networks," *Bell Sys. Tech. J.*, Vol. 58, No. 8, pp. 1765-1777, 1978.

[8] C. E. Shannon, "Memory Requirements in a Telephone Exchange," *Bell Sys. Tech J.*, pp. 343-349, 1950.

[9] S. Chung, Carlos A. Berenstein, "Omega-Harmonic Functions and Inverse Conductivity Problems on Networks," *SIAM Journal of Applied Mathematics* 65(4): 1200-1226, 2005.

[10] B.R. Quinton and S.J.E. Wilton, "Concentrator Access Networks for Programmable Logic Cores on SoCs," *Proc. of the IEEE International Symposium on Circuits and Systems*, pp. 45-48, Kobe, Japan, May 2005.