A CENTRALLY CONTROLLED SHUFFLE NETWORK FOR RECONFIGURABLE AND FAULT-TOLERANT ARCHITECTURE

Nripendra N. Biswas, S. Srinivas and Trishala Dharanendra,
Department of Electrical Communication Engineering,
Indian Institute of Science, Bangalore 560 012, **INDIA**

ABSTRACT

The paper describes a multistage shuffle interconnection network which is controlled by a central monitor. A control code broadcast by the monitor to all the basic switching elements of the network simultaneously, makes the network dynamically reconfigurable. The control code plays three vital roles. Firstly, it establishes conflict-free paths between several source-destination pairs. Thus the problem of collision, a major obstacle of a self-routing network, is completely eliminated. Secondly, the direct paths are established in one clock period irrespective of the number of stages. This makes the system faster. Thirdly, the control code also acts as a grouping code for executing a table of arbitrary data exchange requests between nodes in minimum number of passes. It is also shown that the network can be made fault-tolerant by the addition of an extra stage. Any single fault and some multiple faults in the intermediate stages can be tolerated by this scheme. Moreover, the switching from the faulty state to the fault-free state can be done in a single clock period, thus enabling fast fault-tolerant reconfiguration.

KEYWORDS

centrally controlled interconnection network, fault-tolerant architecture, interconnection networks, multiprocessor systems, reconfigurable architecture, routing techniques, shuffle network.

INTRODUCTION

The performance of a multiprocessor system depends primarily on the efficiency of the interconnection network(IN). Many multistage interconnection networks have been proposed for interconnecting multiple processors [1]. The multistage shuffle network (also known as omega network [2]) has been shown to be a very good interconnection scheme in some particular applications. In this paper, we describe a centrally controlled multistage shuffle network and study its properties. It is shown that the IN will allow fast dynamic restructuring of node-node (processor-processor or processor-memory) connections. Further, it is seen that fault-tolerance can be achieved in the IN by the addition of an extra stage.

DESCRIPTION OF THE CENTRALLY CONTROLLED SHUFFLE NETWORK

The basic switching element (BSE) of the IN considered in this paper is a 2x2 switch with an additional control line. Each BSE has two settings: straight or exchange, depending upon whether the control line is 0 or 1 respectively (see fig. 1).

Consider a multiprocessor system with N nodes (processors). We interconnect the N nodes by a shuffle network consisting of m stages ($m = \log_2 N$). Each stage consists of N/2 BSEs. The different stages are interconnected by a perfect shuffle [3]. But we do not have a shuffle in the front end (at the input of the first stage).

The control lines of all the BSEs of a particular stage are connected to a single line going to the central monitor. A 3-stage shuffle network connecting eight processors (0,1, ...,7) is shown in fig.1.

Thus, it is evident that the routing in the above scheme is obtained by an m-bit routing code broadcast by a hardware circuit in the central monitor. It is important to note that all the switching elements in any one stage of the IN receive the same control bit and hence switch to the same setting. For example, fig. 1 shows the switch settings for the 3-bit control code 011.

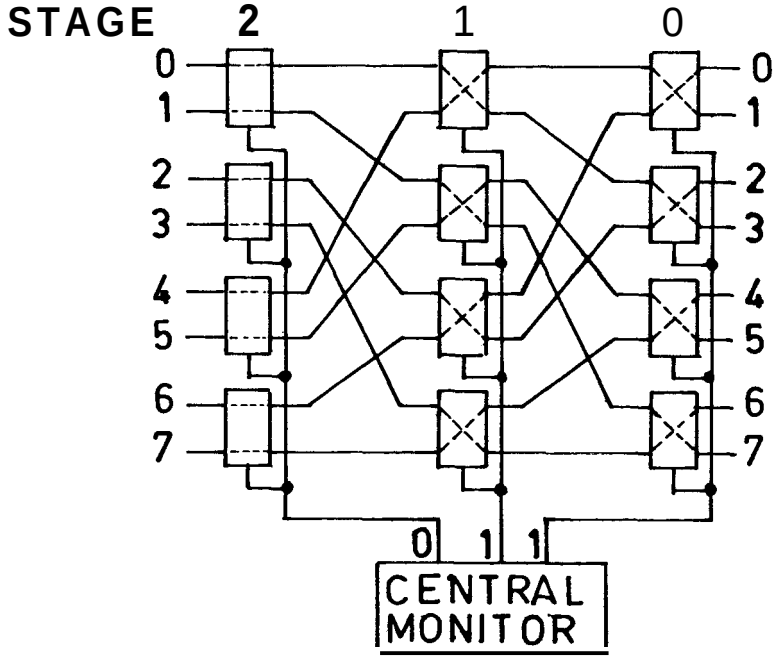

Fig. 1: A 8-node (3-stage) shuffle network with centralized control

CHARACTERISTICS OF THE CENTRALIZED CONTROL SCHEME

We have described a multistage shuffle network where the node-node interconnection is obtained by a code broadcast by a central monitor. This scheme is different from the earlier versions [2] where the routing was obtained by the destination code. The central monitor seems viable if we consider a system model in which all the processors are connected to one control unit (CU). The same CU also controls the interconnection network.

We shall now examine the advantages that the centralized control scheme will have over the self-routing shuffle network.

1. For a particular m-bit control code (CC) broadcast to all the BSEs by the

monitor, each source node NS establishes a connection with a unique destination node, ND, where ND is given by

$$ND = CRS(NS) + CC \quad (1)$$

where CRS(NS) represents a one-bit circular right shift of NS and + is the mod 2 addition (that is, exclusive or) operation. For example, in fig.1, node 1 (001) is connected to the destination given by

$$ND = CRS(001) + 011 = 100 + 011 = 111 (7).$$

Further, from (1) we get

$$CC = CRS(NS) + ND \quad (2)$$

Equation (2) shows that every source-node pair **(NS-ND)** has a control code CC. This gives the network reconfigurable capability. We can have N different configurations, that is, sets of various source-node pairs, which are directly connected, corresponding to different values of the m-bit control code. For N = **8,** the various configurations are given in table **1.**

TABLE 1

CONTROL CODE	SETS OF THE NS-ND PAIRS
000	[0-0], (1-4), (2-1), (3-5), (4-2), (5-6), (6-3), [7-7]
001	(0-1), (1-5), (2-0), (3-4), (4-3), (5-7), (6-2), (7-6)
010	(0-2), (1-6), (2-3), (3-7), (4-0), (5-4), (6-1), (7-5)
011	(0-3), (1-7), [2-2], (3-6), (4-1), [5-5], (6-0), (7-4)
100	(0-4), (1-0), (2-5), (3-1), (4-6), (5-2), (6-7), (7-3)
101	(0-5), [1-1], (2-4), (3-0), (4-7), (5-3), [6-6], (7-2)
110	(0-6), (1-2), (2-7), [3-3], [4-4], (5-0) (6-5), (7-1)
111	(0-7), (1-3), (2-6), (3-2), (4-5), (5-1), (6-4), (7-0)

Note: The NS-ND pairs shown in square brackets [ ] are identity pairs where NS =ND.

2. The hardware of the BSE will be simpler as it need not generate the control bit needed to set the switch either to straight or exchange mode.

3. The direct paths between various source-node pairs are established in one clock time irrespective of the number or stages as all the bits of the control codes are broadcast simultaneously. Thus the route is established parallely rather than serially (stage by stage). This makes the system faster.

4. The calculation of the CC for a particular NS and ND pair can be achieved by a hardware circuit inside each source node, acting as a parallel control code generator.

5. All the N paths of a configuration established by a control code are conflict-free. Therefore, the possibility of collision and the associated problem of its determination and avoidance need not be considered at all.

6. The data communication paths through the network are completely isolated from the control code broadcast path.

APPLICATION TO REAL-TIME DATA EXCHANGE

A frequently encountered problem in a multiprocessor system is the fast execution of several data exchange requests between processor-processor or processor-memory pairs. A sample data exchange table depicting such a situation in an 8-node system is shown in table 2. The data exchange table under consideration is totally arbitrary. It need not follow any particular permutation like bit reversal, bit permute or bit permute complement [4], [5].

We now apply the dynamic restructuring capability of the centrally controlled shuffle network to enable fast execution of the data exchange table.

TABLE 2 : DATA EXCHANGE TABLE

Exchange No.	1	2	3	4	5	6	7	8
Source node (NS)	0	1	2	3	4	5	6	7
Destination node (ND)	2	6	7	4	1	4	0	4
Control code (CC)	010	010	110	001	011	010	011	011

Each source node intially generates the control code corresponding to the destination node with which it has to communicate. Then the control codes generated by all the source nodes are transmitted to the central monitor. The monitor takes up the various control codes one at a time and allows the exchanges having the same CC in one pass. Thus, in the above table, the requests 1, 2 and 6 are sent in the first pass (CC=010), request 3 in second pass (CC=110), request 4 in the third pass (CC=001) and requests 5, 7 and 8 in the fourth pass (CC=011).

It is evident from the above discussion that the control code also acts as a grouping code.

AUGMENTING THE NETWORK FOR FAULT-TOLERANCE

It can be seen that the unique path property of the IN is retained even in case of centralized control, that is, there is only one path between any source-destination pair. Hence the failure of any single BSE destroys the full access property of the network, since one or more sources would be prevented from reaching certain destinations.

The network can be made fault-tolerant by providing an alternative path between any source-destination pair, by augmenting the network with an extra stage in front. The augmented 8-node network is shown in fig.2. To control such a network, the central monitor has to broadcast an (m + 1)-bit code corresponding to (m + 1) stages. Let the (m +1)-bit CC be written as follows:

$$CC\ (m+1) = C_m C_{m-1} \cdots C_1 C_0$$

where C_m, C_{m-1}, . . ., C_1, C_0 are the (m + 1) bits of the CC, corresponding to the (m + 1) stages of the network as shown in fig. 2. The mth bit C_m of the control code corresponding to the additional stage is maintained at '0' in case of no failure and its existence-can be ignored. The-remaining m-bit CC,($C_{m-1} \cdots C_1 C_0$) is given by

$$CC(m) = NS(m) + ND(m) \quad (3)$$

where CC(m), NS(m) and ND(m) are the m-bit representations of the CC, NS and ND respectively.

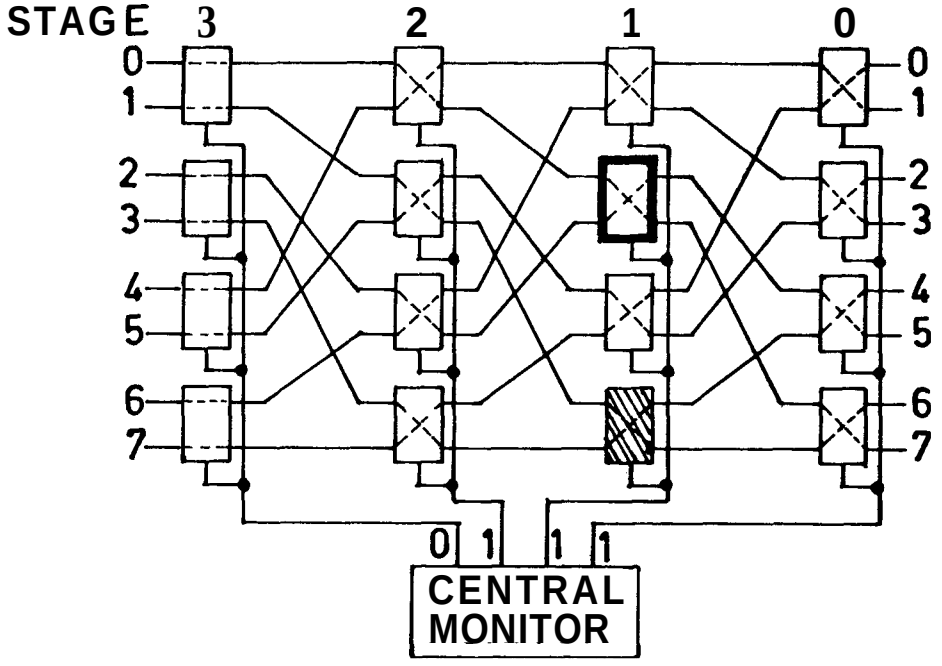

Fig. 2 : An augmented shuffle network for fault tolerance

IMPORTANT FEATURES OF THE AUGMENTED SHUFFLE NETWORK

We have observed the following interesting properties of the augmented network. These properties can be stated and proved as theorems.

1. The two paths starting from a BSE of the additional (mth) stage will always enter another BSE of the last (0th) stage.
2. These two paths do not share the same BSE in any intermediate stage.

These properties can be used to make the network fault-tolerant in a very simple way.

In case of any single fault at any intermediate stage, paths from one or more source nodes may be blocked from reaching their respective destinations. Then if the mth and 0th bits of the control code CC(m +1) are complemented, and the remaining bits are maintained unaltered, an alternative path is set up from all source nodes to their respective destinations. This is illustrated in fig. 3(a) and 3(b). Only the two BSEs of additional and last stages are shown, as control code

of other stages remain the same.

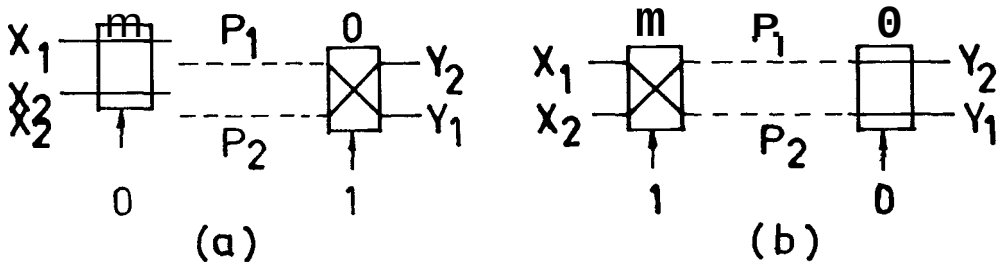

Fig. 3 : Two alternative paths for source-destination connection; (a) before, and (b) after, complementing the mth and 0th control code bits.

X_1 and X_2 reach Y_1 and Y_2 via paths P_1 and P_2 respectively in fig. 3(a). Now if the two CC bits are complemented as shown in fig.3(b), the source-destination pairs connected remain the same but the two paths P_1 and P_2 taken in the intermediate stages get exchanged.

Let us suppose that the following data exchange is to be realized by an 8 node augmented network.

NS:	0	1	2	3	4	**5**	**6**	7
ND:	7	**6**	5	4	3	2	1	0

This can be realized in a single pass in case of no faults. The control code CC(m + 1) needed for all these exchanges is 0111 in case of no faults. Now suppose that a BSE in 1st stage is faulty. Fig. 2 shows the faulty BSE (shown hatched). Now inputs from processors 1 and 3 through the network have to pass through the faulty BSE and hence may either get obstructed or may reach a wrong destination. These two exchanges can be realized in the second pass by complementing the first and the last control code bits. The new control code would be 1110, and the processors are now routed to their respective destinations via paths which avoid the faulty BSE and pass through the BSE (shown bold) in stage 1.

When realizing an arbitrary data exchange, if k passes are required when there are no faults, then atmost 2k passes are required in case of any single fault in the intermediate stages. The centrally controlled system we have described may also tolerate many multiple faults. As long as these faults are noncritical, that is, they do not block both paths emerging from a single BSE at the additional stage, they can be tolerated. Also the number of passes required remains atmost 2k.

Another interesting property of the centrally controlled scheme is that there is no necessity for fault location. Moreover, unlike a self-routing system [6], it does not require any conflict analysis to provide an alternative path. Non-acknowledgement of the proper data by a destination node is the indication to the central

monitor that one or more BSEs has gone faulty. Immediately, irrespective of which BSE is faulty, it complements the mth and the 0th control code bits providing an alternative path for only those data which could not reach their destinations in the previous pass.

CONCLUSION

The above discussion clearly brings out several distinct advantages of a centrally controlled shuffle network compared to a self-routing one. The only price that we pay for gaining these desirable features is to run an extra wire per stage from the central monitor to all the switching elements of one stage. However, this increase will be adequately compensated by decrease in the hardware of each basic switching element.

REFERENCES

1. T-y Feng, "A survey of interconnection networks", IEEE Computer, Vol. 14, December 1981, pp. 12-27.
2. D.H. Lawrie, "Access and alignment of data in an array processor", IEEE Transactions on Computers, Vol. C-24, December 1975, pp. 1145-1155.
3. H.S. Stone, "Parallel processing with the perfect shuffle", IEEE Transactions on Computers, Vol. C-20, February 1971, pp. 153-161.
4. D. Nassimi and S. Sahni, "Parallel permutation and sorting algorithms and a new generalized connection network", Journal of the ACM, Vol. 29, No. 3, July 1982, pp. 642-667.
5. J. Lenfant, "Parallel permutation of data : A Benes' network control algorithm for frequently used permutations", IEEE Transactions on Computers, Vol.C-27, July 1978, pp. 637-647.
6. C.S. Raghavendra and Anujan Varma, "Fault-tolerant multiprocessors with redundant path interconnection networks", IEEE Transactions on 'Computers, Vol. C-35, April 1986, pp. 307-316.