NASA Contractor Report 172476

ICASE REPORT NO. 84-54

ICASE

SHUFFLE-EXCHANGES ON AUGMENTED MESHES

Shahid H. Bokhari

Contract No. NAS1-17070
October 1984

INSTITUTE FOR COMPUTER APPLICATIONS IN SCIENCE AND ENGINEERING
NASA Langley Research Center, Hampton, Virginia 23665

Operated by the Universities Space Research Association

NASA
National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

SHUFFLE-EXCHANGES ON AUGMENTED MESHES

Shahid H. Bokhari
Institute for Computer Applications in Science and Engineering
and
University of Engineering and Technology, Lahore, Pakistan

Abstract

Prior research has shown how a mesh connected array of size $N=2^K$, K an integer, can be augmented by adding at most one edge per node such that it can perform a shuffle-exchange of size $\frac{N}{2}$ in constant time.

We now show how to perform a shuffle-exchange of size N on this augmented array in constant time. This is done by combining the available perfect shuffle of size $\frac{N}{2}$ with the existing nearest neighbor connections of the mesh. By carefully scheduling the different permutations that are composed in order to achieve the shuffle, the time required is reduced to 5 steps, which is shown to be optimal for this network.

This research was supported by NASA Contract NAS1-17070 while the author was in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center.

1. Introduction

It has been shown [1] that a mesh connected computer of size $N=2^K$, integer K, can be augmented by adding at most one edge per node so that it can carry out the shuffle-exchange permutation [2] of $\frac{N}{2}$ elements in constant time. On a 4-nearest neighbor mesh, for example, this requires 3 time steps.

The motivation for this work was to construct new interconnection structures that combine the capabilities of shuffle-exchange networks and nearest neighbor arrays but with cost less than the sum of the costs of the constituent networks. It is clearly trivial to superimpose two networks to get the sum of their capabilities. The augmentation described in [1] permits a mesh of size N to perform a shuffle exchange of size $\frac{N}{2}$ at the cost of at most one additional edge per node. A question that was unresolved in [1] is how efficiently the augmented network could carry out a shuffle-exchange of size N.

In the present paper we demonstrate that this augmented network can be used to perform the shuffle-exchange of N points in constant time. Furthermore, we show that, by carefully scheduling the data transfers between different parts of the network, this can be done in 5 time steps, which is optimal for this network.

2. The Augmented Mesh

A shuffle-exchange network of size 8 is shown in Fig. 1. Fig.2 shows how a 4-nearest neighbor mesh of size 16 may be augmented by adding the required shuffle connections. It is clear that data at nodes

$0,1,\ldots,7$ can be shuffled over to nodes $0',1',\ldots,7'$. The exchange operation can then be performed using the vertical mesh connections between pairs of nodes $0'$-$1'$, $2'$-$3'$ etc. and the results moved back to the original set of nodes 0, 1,..., 7 via the horizontal mesh connections. For example, data in nodes 1 and 5 would, after shuffling, end up in nodes $2'$ and $3'$, respectively. The exchange would be performed using the vertical edge joining $2'$ and $3'$. Finally data from $2'$ and $3'$ would be shipped back to 2 and 3 via horizontal edges. The entire shuffle-exchange operation of size $\frac{N}{2}$ requires 3 time steps.

The augmentation procedure is to give unprimed labels 0 to $(\frac{N}{2}-1)$ to the odd columns and primed labels $0'$ to $(\frac{N}{2}-1)'$ to the even columns. Shuffle connections are then added between these nodes. It is important to note that this augmentation can be applied to any array as long as it is a "true" rectangle (each size has at least 2 nodes along it) and the number of nodes is a power of 2. Such an array can be divided into $\frac{N}{4}$ squares of size 4 each (e.g. $0,0',1,1'$ in Fig. 2) which can carry out the exchange operation.

3. Complete Shuffles on an Augmented Mesh

We now show how a shuffle of size N can be performed on a mesh that has been augmented in the manner described in the previous section. As a running example, we use a rectangular mesh of size 32.

The shuffle-exchange operation in the previous section utilizes all the added shuffle edges plus some of the original mesh edges. In Fig. 3 we have deleted from our augmented rectangular mesh all connections unnecessary for the shuffle-exchange operation. The layout of the original mesh is not important to our analysis. It could be configured as 2x16, 4x8, 8x4 or 16x2 nodes. In all cases we would add shuffle connections as described in the previous section. Our problem now is to show how a shuffle-exchange operation can be performed on the 32 data items stored in nodes 0..31 of Fig. 3.

Recall that the shuffle operation is defined [2] as follows:

$$P(i) = 2i, \qquad 0 \leq i \leq \frac{N}{2} - 1 \qquad (1)$$

$$P(i) = 2i-N+1, \qquad \frac{N}{2} \leq i \leq N-1 \qquad (2)$$

It is easy to trace through the network of Fig. 3 and see that a path of length no greater than 4 exists between any node i and its P(i), as defined by (1) and (2) above. This does not necessarily imply, however, that the shuffle can be performed in four steps because for any nodes i and j, the paths i to P(i) and j to P(j) can have several common edges, leading to delays.

Inspection of the edges in Fig. 3 reveals that they can be divided into four classes: the horizontal and vertical edges of the original network and the shuffle connections from left to top right and from left to bottom right.

In the following, we specify routings between all i and their corresponding P(i) such that the shuffle operation can be performed in constant time, independent of the size of the array. In the next section we show how these routings can be scheduled so that the shuffle is done in optimal time.

The edges of the augmented network allow us to perform the following permutations.

1) Vertical edges:

$V(i) = i + 1 \qquad 0 \leq i \leq N-1, \quad \text{even } i$ (3)

$V^{-1}(i) = i - 1 \qquad 0 \leq i \leq N-1, \quad \text{odd } i$ (4)

2) Horizontal edges:

$H(i) = i + \frac{N}{2} \qquad 0 \leq i \leq \frac{N}{2} - 1$ (5)

$H^{-1}(i) = i - \frac{N}{2} \qquad \frac{N}{2} \leq i \leq N-1$ (6)

3) Shuffle edges from left towards top right:

$T(i) = 2i + \frac{N}{2} \qquad 0 \leq i \leq \frac{N}{4} -1$ (7)

$T^{-1}(i) = \frac{i}{2} - \frac{N}{4} \qquad \frac{N}{2} \leq i \leq N-1, \quad \text{even } i$ (8)

4) Shuffle edges from left towards bottom right:

$B(i) = 2i + 1 \qquad \frac{N}{4} \leq i \leq \frac{N}{2} -1$ (9)

$B^{-1}(i) = \frac{i - 1}{2} \qquad \frac{N}{2} \leq i \leq N-1, \quad \text{odd } i$ (10)

We will be applying the permutations (3)-(10) above on groups of nodes. For notational convenience, we replace "(i)" in (3)-(10) with the triple "[begin, end, step]". In this vector notation, $V[1 ,\frac{N}{2} - 1, 2]$ means that permutation V is applied to every second node, starting with node 1 and going up to node $\frac{N}{2} - 1$.

The following are the permutations that must be composed in order to obtain the perfect shuffle. (We use the left composition convention in this discussion.)

$$H^{-1}[\frac{N}{2}, N-2, 2] \circ T[0, \frac{N}{4} -1, 1] \qquad (11)$$

$$V^{-1}[\frac{N}{2} +1, N-1, 2] \circ B[\frac{N}{4}, \frac{N}{2} -1, 1] \qquad (12)$$

$$H^{-1}[\frac{N}{2} +3,N-1,4] \circ V[\frac{N}{2} +2,N-2,4] \circ T[1, \frac{N}{4} -1,2] \circ H^{-1}[\frac{N}{2} +1, \frac{3N}{4} -1,2] \qquad (13)$$

$$V[0, \frac{N}{2} -4, 4] \circ H^{-1}[\frac{N}{2}, N-4, 4] \circ T[0, \frac{N}{4} -2, 2] \circ H^{-1}[\frac{N}{2}, \frac{3N}{4} -2, 2] \qquad (14)$$

$$B[\frac{N}{4}, \frac{N}{2} -1, 1] \circ H^{-1}[\frac{3N}{4}, N -1, 1] \qquad (15)$$

It is easy to verify that (11) to (15) correspond to (1) and (2) over the correct ranges.

To avoid edge conflicts, we can successively apply (11) to (15) to the network. The perfect shuffle permutation can thus be achieved in 14 time steps (the sum of the times required for (11)-(15)). The exchange permutation involves nothing more than the interchange of data via the vertical mesh links, i.e. the application of V and V^{-1} to all even and odd nodes respectively. This can be done in one step, resulting in a total of 15 time steps for the shuffle-exchange.

4. Optimal Scheduling

In this section we describe how we can optimally perform the perfect shuffle. We view this as a scheduling problem. The jobs are data 0 through N. Permutations (3)-(10) are the available processors which must be applied to these jobs according to the sequences (11)-(15).

Each permutation requires one time step. At any time step a datum may have only one permutation applied to it. Each permutation (strictly speaking, each edge) may be used only once during each time step.

By inserting idle times (null permutations) very carefully, the schedule of Table I is obtained. In order to be consistent with the left composition convention, time advances from right to left in this table. It may be verified that this schedule satisfies all of the above constraints.

The longest compositions, (13) and (14), are of length 4. Thus our schedule, also being of length 4, is optimal. A further step is required for the exchange, giving a total of 5 steps.

Table II gives the instance of Table I for N=32 (corresponding to Fig. 13.) The extra leftmost column in this table shows the range of the last permutation in each row, demonstrating that all points are included.

5. Conclusions

We have shown that an augmented mesh of size N can perform the shuffle- exchange in constant time and have also shown how this can be done optimally in 5 time steps. This result indicates that the shuffle augmented mesh of [1] is a powerful interconnection structure that combines the advantages of nearest neighbor arrays and shuffle exchange networks.

6. Acknowledgement

The author is grateful to Dr. R. G. Voigt for several discussions and for a critical reading of the manuscript.

7. References

[1] S. H. Bokhari and A. D. Raza, "Augmenting Computer Networks," Proc. 1984 Int. Conf. on Parallel Proc., August 1984, pp. 338-345.

[2] H. S. Stone, "Parallel Processing with the Perfect Shuffle," IEEE Trans. Computers., vol. C-20, No. 2, pp. 153-161, February 1971.

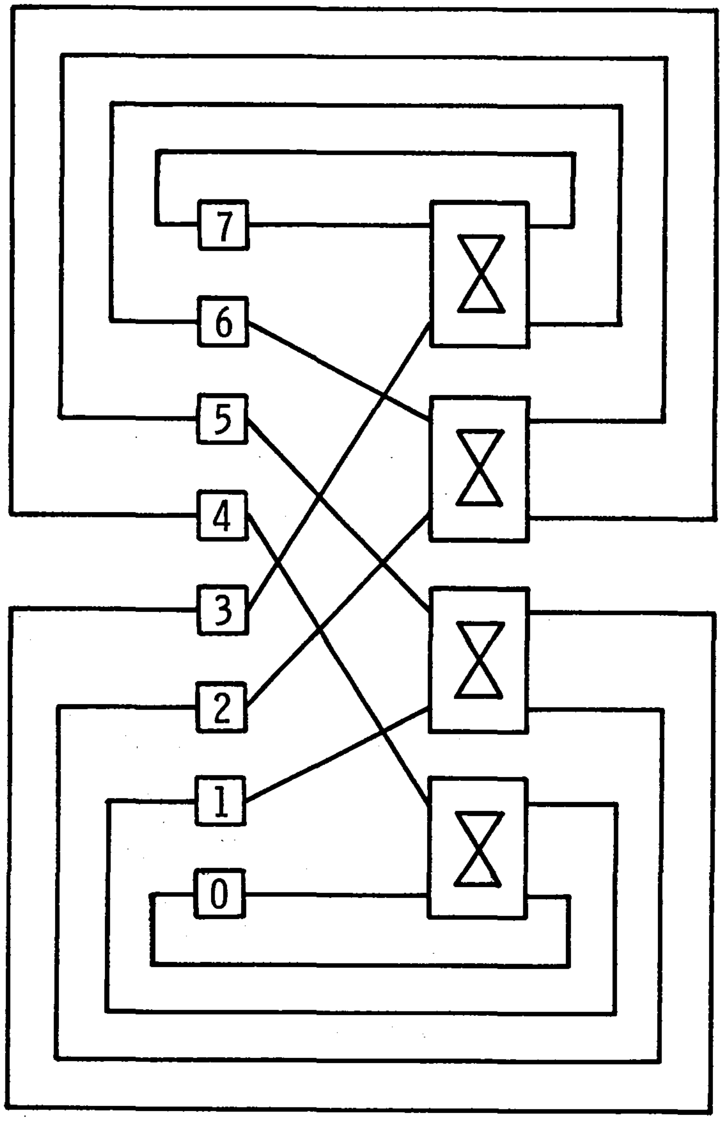

Fig. 1. Single stage recirculating shuffle-exchange network of size N=8.

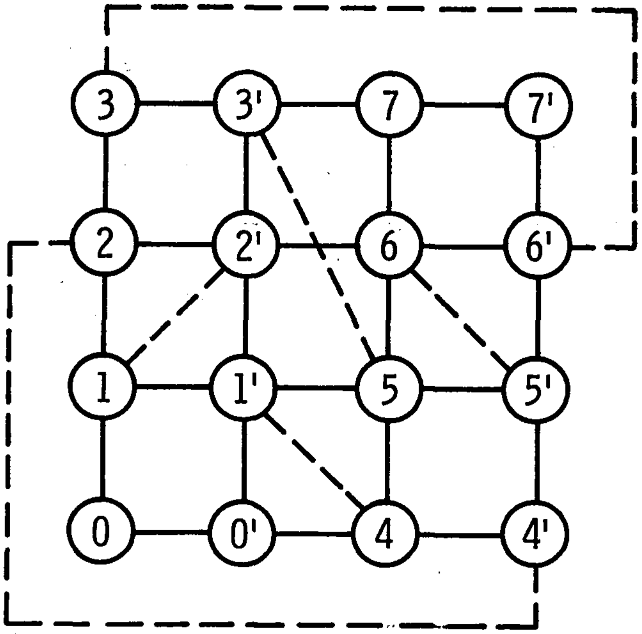

Fig. 2. Shuffle connection augmented 4x4 4-nearest neighbor array that can emulate the network of Fig. 1 in constant time.

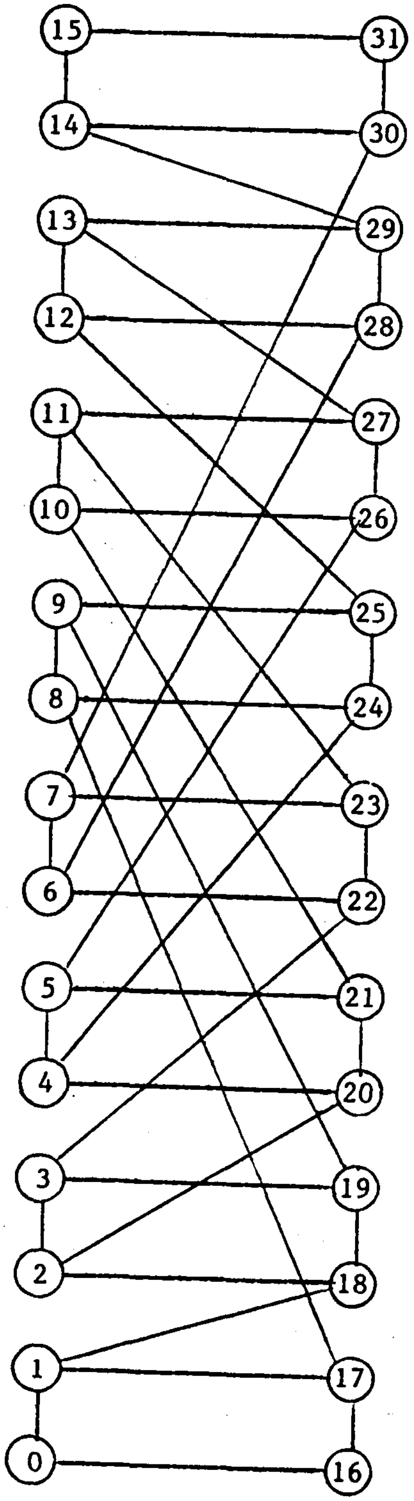

Fig. 3 Essential Features of a 32 node shuffle augmented rectangular mesh.

Table I. Optimal schedule for the perfect shuffle.

(Time advances from right to left.)

$H^{-1}[\frac{N}{2}, N-2, 2]$	IDLE	IDLE	$T[0, \frac{N}{4} -1, 1]$	11
IDLE	$V^{-1}[\frac{N}{2} +1, N-1, 2]$	IDLE	$B[\frac{N}{4}, \frac{N}{2} -1, 1]$	12
$H^{-1}[\frac{N}{2} +3, N-1, 4]$	$V[\frac{N}{2} +2, N-2, 4]$	$T[1, \frac{N}{4} -1, 2]$	$H^{-1}[\frac{N}{2} +1, \frac{3N}{4} -1, 2]$	13
$V[0, \frac{N}{2} -4, 4]$	$H^{-1}[\frac{N}{2}, N-4, 4]$	$T[0, \frac{N}{4} -2, 2]$	$H^{-1}[\frac{N}{2}, \frac{3N}{4} -2, 2]$	14
IDLE	$B[\frac{N}{4}, \frac{N}{2} -1, 1]$	$H^{-1}[\frac{3N}{4}, N -1, 1]$	IDLE	15

Table II. Optimal schedule for N=32.

The last (leftmost) column contains the range of the last permutation in each row.

[0,2,4,6,8,10,12,14]	H^{-1}[16,18,20,22,24,26,28,30]	IDLE	IDLE	T[0,1,2,3,4,5,6,7]	11
[16,18,20,22,24,26,28,30]	IDLE	V^{-1}[17,19,21,23,25,27,29,31]	IDLE	B[8,9,10,11,12,13,14,15]	12
[3,7,11,15]	H^{-1}[19,23,27,31]	V[18,22,26,30]	T[1,3,5,7]	H^{-1}[17,19,21,23]	13
[1,5,9,13]	V[0,4,8,12]	H^{-1}[16,20,24,28]	T[0,2,4,6]	H^{-1}[16,18,20,22]	14
[17,19,21,23,25,27,29,31]	IDLE	B[8,9,10,11,12,13,14,15]	H^{-1}[24,25,26,27,28,29,30,31]	IDLE	15

1. Report No. NASA CR-172476 ICASE Report No. 84-54	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle

SHUFFLE-EXCHANGES ON AUGMENTED MESHES

5. Report Date
October 1984

6. Performing Organization Code

7. Author(s)

Shahid H. Bokhari

8. Performing Organization Report No.

84-53

9. Performing Organization Name and Address

Institute for Computer Applications in Science
and Engineering
Mail Stop 132C, NASA Langley Research Center
Hampton, VA 23665

10. Work Unit No.

11. Contract or Grant No.

NAS1-17070

12. Sponsoring Agency Name and Address

National Aeronautics and Space Administration
Washington, D.C. 20546

13. Type of Report and Period Covered

Contractor Report

14. Sponsoring Agency Code

505-31-83-01

15. Supplementary Notes

Langley Technical Monitor: J. C. South, Jr.
Final Report

16. Abstract

Prior research has shown how a mesh connected array of size $N = 2^K$, K an integer, can be augmented by adding at most one edge per node such that it can perform a shuffle-exchange of size N/2 in constant time.

We now show how to perform a shuffle-exchange of size N on this augmented array in constant time. This is done by combining the available perfect shuffle of size N/2 with the existing nearest neighbor connections of the mesh. By carefully scheduling the different permutations that are composed in order to achieve the shuffle, the time required is reduced to 5 steps, which is shown to be optimal for this network.

17. Key Words (Suggested by Author(s))

array processors, augmented networks,
meshes, nearest neighbor arrays networks
perfect shuffle, shuffle-exchange

18. Distribution Statement

62 - Computer Systems
66 - Systems Analysis

Unclassified - Unlimited

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 13	22. Price A02

For sale by the National Technical Information Service, Springfield, Virginia 22161