



Journal of Applied Sciences

ISSN 1812-5654

science
alert

ANSI*net*
an open access publisher
<http://ansinet.com>

Grouping Technique for Re-Routing a Rearrangeable MIN

M. Reza Salehnamadi

Islamic Azad University of Iran, Tehran-South Branch, P.O. Box 16535-334, Tehran, Iran

Abstract: A new method for routing in a rearrangeable three-stage CLOS MIN is presented. The contribution of study is grouping all inputs according to a proper specification of MIN and then executing the selected algorithm within each group. The grouping title is selected because the process has two groupings: The first-one is the inputs grouping and the second is the out_pins of first_stage boxes grouping. This idea has the following advantages: (1) the rerouting time is short, (2) the routing of the other boxes are not suspended or interrupted and (3) the controller is very simple because it is limited to only one box. To analyze of nonblocking condition of selected grouping, a helpful mathematical model is presented. After designing an algorithm according to grouping idea, the minimum extra boxes is computed. Some modifications have been taken to reduce the additional hardware. And finally the general rule, same as the additional boxes which could be neglected, is extracted. All of these steps are presented and analyzed in the study and by using the extracted rule, additional boxes can be neglected for large amount of inputs. The time complexity depend on the routing algorithm. The presented controller with a simple serial method, has the time complexity $O(\sqrt{N})$ where N is the input number.

Key words: MIN, CLOS, rearrangeable, nonblocking, routing

INTRODUCTION

Interconnection network has a great effect on performance of multiprocessors. The Multistage Interconnection Network (MIN) has less complexity to that of the crossbar interconnection network regarding of the number of switching elements (Chen, 2007). Generally, there are two types of MINs: blocking and nonblocking. Many studies are being carried out to increase the performance of MINs. In all of them, permutation capability can isolated by multicast and broadcast capabilities (Tian *et al.*, 2006).

The nonblocking type has more applications and is more complicated. Various configurations for this type have been reported in the study. The goal of each design is to reduce the hardware and/or time complexity (Lai, 2000). In fact, there are two independent but correlated factors in the network topology and routing. Some study on topology and gain the high performance with proposing a special topology. Others, with a general topology, like as the 3-stage CLOS, present routing algorithms with the lower complexity. All of these designs are divided in two groups: one_pass and multi_pass mechanisms. In multi_pass routing, by using a standard topology, packets reach to destination after multi pass through the network.

Alleyne and Scherson (2000) introduces a dynamic two-stage delta network and analyzes it for permutation routing. Messages are reached from input ports to output ports in $2N$ passes where N is the input number.

In this way, the number of boxes and crosspoints are reduced, but the propagation delay is increased. For example; if $N = 4096$, it takes 16 passes with two stages in each pass. So it actually has a 32-stages delay time. Similarly, Lai (2000) presents an algorithm for performing permutation of message in MIN in two passes. He uses a generalized CUBE interconnection topology and a routing algorithm in $O(N)$ time. Regarding of propagation delay time, there are $2\log N$ stages in front of each message to reach out.

The CLOS interconnection network has gained attention due to its potential users in data networks and computing systems (Yang and Wang, 2004). The general scheme of $V(m,n,r)$ CLOS network is shown in Fig. 1. The efficiency of MIN is critical to overall system performance and depends on the speed of the routing control among other parameters. A parallel algorithm for the routing assignment in the CLOS network is studied (Lee and Liew, 1996), in which, $m = n$ and the time complexity of the algorithm is $O(\log^2 N)$. H. Lee presents a decomposition algorithm to route a rearrangeable three stage CLOS network with serial routing with $O(nN)$ time complexity (Hwang *et al.*, 1996).

Nonblocking condition and blocking probability in (Datta and Kobayashi, 2004) stochastic model for hierarchical structure of nonblocking R-CLOS (Morimura *et al.*, 2006) and new nonblocking requirement at strictly nonblocking CLOS have studied in (Hung-Lin and Hwang, 2005).

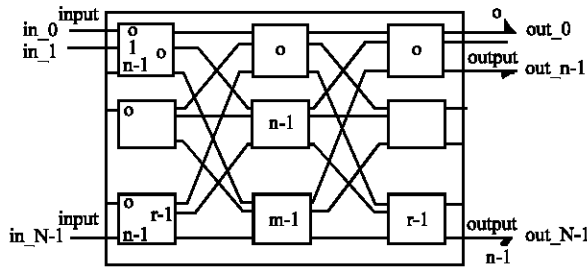


Fig. 1: Three-stage CLOS interconnection network

In this study, a new method is presented for routing in a rearrangeable 3-stage CLOS multistage interconnection network. Inputs are divided in some groups and rearranging is done only in one group of the first stage routing in the second and third stages is done in a delta distributed fashion. In the first stage, switching of each box is determined by a centralized controller.

THE CONTRIBUTION

The main contribution of this study is to limit the rearrangement process to the group that one of its inputs requires a path to get through the MIN. Usually, rearrangement is done in two ways: serial and parallel. In a fast serial method, the rearrangement routing for N inputs are processed pin-to-pin until all inputs are routed. This method is slow, i.e., $O(N \log N)$. The second method is parallel (Lee, 1996). All inputs are routed simultaneously using graphs, matrices or other tools. The second method is faster, i.e., $O(\log^2 N)$, but requires a heavy hardware. In both methods, all message transmissions are suspended while the rearrangement is being done.

The main idea is to make some groups from all inputs based on a rule and then limit the rearrangement routing process to the group in which the destination of one of its inputs has been changed. This simple process is enhanced by using the self-routing method in the second and third stages of MIN. This mechanism has the following advantages:

- **Controller simplicity:** In contrast to parallel rearrangeable algorithm, This method has a simple controller: Manipulating g inputs versus N inputs, where, N is the number of the all inputs and g is the number of the inputs in one group.
- **High speed:** In contrast to serial rearrangement algorithms, this method is very quick.: $O(g)$ versus $O(N \log N)$.
- **Less suspending:** In this method, no message transmission is suspended except the messages in the box being rerouted.

- **Hardware reduction:** The number of extra boxes added to the middle stage boxes are not comparable with the first stage boxes, i.e. the added boxes approaches zero for large amount of N.

The interconnection network is based on a 3-stage CLOS network. To limit the rearrangement process to a grouping re-routing requires some boxes added to the middle stage. A mathematical model is presented to determine the required additional boxes in the second stage. According to this model, a solution to reduce the additional boxes is shown. It is shown that for large N, required additional boxes approach zero and it is a great advantage for rearrangeable MINs.

To switch the input pins of a box to its output pins, the following three steps should be done:

- Check the out-pins and determine which one is free
- For free pins, check which one has no assignment for the corresponding pin-number of other boxes for the same destination box-number
- After switching all input pins to output pins in a box, all assignments are recorded for the next box routing

DEFINITIONS

To describe the routing algorithm, some definitions are presented. The network is a 3-stage CLOS network $(v(m,n,r))$ with the size of $N \times N$ (Fig. 1). The first Stage has n boxes with n in-pins and the third stage has n box with n out-pins. In $v(m,n,r)$ network if $n = r$, the number of crosspoints are minimum. So, in this study without losing of generality, it is assumed that $n = r$. grouping rule is all inputs of a box are in same group. Also, to clear the points, an example is presented whenever it is necessary with $N = 16$ and $n = 4$. Now some terms are defined as follows:

- OUT_i : The destination out-pin for the input i
- O_{bi} : The ith output box
- I_{bi} : The ith input box
- O_{pi} : The ith out-pins of the first stage boxes
- O_{pij} : The ith out-pin of jth box of the first stage
- I_{pij} : The ith input-pin of jth box of the first stage

A permutation is shown as follows:

$$P = \begin{bmatrix} 0 & 1 & 2 & \dots & N-2 & N-1 \\ \text{OUT}_0 & \text{OUT}_1 & \text{OUT}_2 & \dots & \text{OUT}_{N-2} & \text{OUT}_{N-1} \end{bmatrix}$$

The first row includes the in-pin numbers and the second one consists of includes the destination numbers.

Note 1: The routing process is to find a path between input and output boxes. So, the description of the second row of P in terms of Box-numbers can be useful.

Note 2: Since we are interested in the single-box rerouting algorithm, only output-box numbers assigned to an input-box are processed.

Definition 1: To switching of IBk, the IPOBk matrix is defined to establish the relationship between the input-pins of the IBk to the output-boxes as follows:

$$IPOBk = [Q_{ij}]$$

where, Q_{ij} is an element of IPOBk matrix and $0 \leq i, j, k \leq n-1$. Each row is for a pin of IBk and each column is for an output-box.

$$Q_{ij} = \begin{cases} 1 & : Ip_{ik} \text{ has a message to } OB_j \\ 0 & : \text{Otherwise} \end{cases}$$

For the presented example:

$$IPOB_0 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \quad IPOB_1 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

$$IPOB_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \quad IPOB_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

For example, in IPOB0 the value of 1 in row 0 and column 2 specify that Ip00 must be connected to OB2.

Definition 2: OPik can be routed to any output box (OBj) when none of the corresponding Opi's is routed to the same OBj. Free or occupied routing status of Opi's to an OBj is determined by a flag bit. The set of these flags is named W matrix. All elements of this matrix is initially set to 1 as follows:

$$W = \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \end{pmatrix}$$

W is a matrix with m rows and n columns where m is the number of middle-stage boxes (or the number of out-pins of the first stage boxes) and n is the number of output box. Note that m is not specified yet and one the purposes of this mechanism is to reduce the size of middle stage boxes as possible.

IPOBk and W matching: Using the above definitions, routing of the IBk is achieved by slipping IPOBk on W. IPOBk and W have information about the required routing. IPOBk determines the requested output boxes and W determines the occupied pins. Matching these two matrix, by slipping down IPOBk on W, extracts the correct switching. In fact, this matching extraction focuses on nonblocking conditions.

$IPOBk(q, j) = 1$ implies that Ipqk requires a path to Obj. By slipping each rows of IOBk on W, if $w(i, j) = 1$, $0 \leq i \leq m-1$, $0 \leq j \leq n-1$ and if $w(i, j)$ is selected, it is set to 0. This means that OPik is occupied. In each slipping, only one element can be selected in each row; otherwise, it means that OPik is connected to more than one box simultaneously which is not practical.

When all rows of IPOBk are slipped down to W, the switching information for IBk is completed. Now, W matrix is changed to W'. The elements of $W'(i, j) = w'_{ij}$ are defined as follows:

$$W'_{ij} = \begin{pmatrix} 1 & Op_i \text{ can be routed to } OB_j \\ 0 & \text{ther is no route from } Op_i \text{ to } OB_j \end{pmatrix}$$

For the described example, W' is as follows:

$$W' = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

In general form, W' has m rows, but in above example, it shows an special case with $m = n = 4$. With this example, "only reminder 1" case is visible which will be used, in column_2 in the next section.

THE BLOCKING CONDITION AND ITS REMOVAL

To remove the blocking condition, first two lemmas are defined. Then the necessary conditions are mentioned for avoiding blocking and finally some simple mechanisms for grouping rerouting are presented.

Lemma 1: In W', if the only remainder 1 of any two columns, have the same row, blocking may happen in routing of the next boxes.

Proof: If two columns i and j have this condition and a new IBk permutation requires output boxes i and j , by slipping down IPOBk on W' , both i and j in one row are required. This means that one output pin of the input box is going to connect to two output boxes. it is not practical and a conflict happens.

Note 1: Only remainder 1 of two columns, can produce only one common row condition.

Note 2: If however common row condition doesn't create or disappeared, conflict in the other boxes can't be happened.

Lemma 2: For a box with n inputs, there are up to $n-2$ states in which lemma_1 maybe happen.

Proof: lemma 1 condition requires at least a couple of columns to occur.

- The first column needs another column to be coupled, so it can not cause any conflict individually.
- After slipping the last column there is not any column to perform a couple. This means that the last column, can't cause any conflict.

Therefore, two columns can't participate in lemma-1 condition and so up to $n-2$ case is remained λ .

For example by $n = 8$, columns 0 and 1 can form a couple with only one common row. Also columns 0 and 2 can form a couple with only one common row, or briefly: (0, 1), (0, 2), (0, 3), (0, 4), (0, 5), (0, 6). The effect of couples like (3, 4) is covered by (0, 3) and (0, 4) couples.

Blocking avoidance: To avoid blocking, from lemma 1, it is sufficient to add some new rows to W Matrix, so that the state of common row vanishes. In other words, when we add some rows, the situation of only 1 reminder vanishes and as a result the situation of common row vanishes. From the hardware point of view, sufficient boxes must be added to the middle stage to avoid the blocking condition. Now, the important question is the number of added rows.

The required number of rows: As mentioned, each common row occurrence needs one row to be non-effective. According to lemma 2, there are $n-2$ common row occurrences and so, $n-2$ additional rows are required. This result has no advantage, since the strictly nonblocking MIN needs to $n-1$ additional boxes. The advantage of the presented model is the ability of reducing additional row.

To reduce the required additional rows, the routing mechanism must distribute 0 setting in various columns. Therefore, the probability of occurrence the common row becomes minimum. According to this probability, the number of additional rows are determined. In the following, two algorithms based on the extracted model, are described.

Mechanism_1: Top-down column: If even columns are routed from top to the bottom and odd columns from bottom to top in w , probability of occurrence of lemma 1 condition will be half because of scattering of routes. For example:

$$W' = \begin{pmatrix} 0 & 0 & \dots \\ 0 & 0 & \dots \\ 0 & 0 & \dots \\ 1 & 1 & \dots \end{pmatrix} \quad W'' = \begin{pmatrix} 0 & 1 & \dots \\ 0 & 0 & \dots \\ 0 & 0 & \dots \\ 1 & 0 & \dots \end{pmatrix}$$

in W' both columns 0 and 1, are routed from top to the bottom and have a common row but W'' is routed by described top-down method and the common row is removed.

Lemma 3: Using the top-down mechanism and $n/2-1$ additional rows, blocking is avoided.

Proof: W is a matrix with m rows, let $m = n+d$ (d is an integer number). The one 1 remaining condition occurs after manipulating $n-1$ elements in each column. The even columns are routed from top to the bottom. The worst case is, when all even columns are in the common row condition, In a such situation, there are $m-(n-1)$ free rows in the second half of W . For elimination of $n/2-1$ common row of even columns $n/2-1$ additional rows must be added. Therefore:

$$m-(n-1) \geq n/2-1$$

substituting m with $n+1$, results in

$$r \geq n/2-2$$

therefore $r = n/2-1$ condition ensures the nonblocking condition.

Likely, odd columns are routed from bottom to top. There are $m-(n-1)$ free rows in first half of W and we need $n/2-1$ additional rows to eliminate $n/2-1$ common row.

When even columns are manipulated, the added $n/2-1$ rows appear in the bottom side of W as additional rows and when odd columns are manipulated, the added

$n/2-1$ rows, appear in the upper side of W as additional rows. So $n/2-1$ additional rows are sufficient to avoid blocking of odd and even boxes.

Mechanism 2: Octal routing: All boxes are divided into 8 groups. In the following description, k is in the range from 0 to $n/8$.

Boxes with numbers equal to $8k+q$, are routed from output pin No. q , where $0 \leq q \leq 7$.

In this mechanism, boxes in each group have their own rows, so there is no probability for occurrence of the common row condition. Therefore, the probability of common row occurrence in N input reaches to $n/8$. According to lemma 2, $n/8-2$ additional box can guarantee the non-blocking condition. In other side, it is needed to 7 rows for supporting the octal condition inside the group. And hence:

Additional box number = $5+n/8$.

The general mechanism: The general mechanism is stated according to grouping architecture. This mechanism is explained as the following lemma.

Lemma 4: If we can divide all n columns of W to p groups so that each group has no conflict with other groups, $n/p + p-3$ additional rows are needed.

Proof: The n columns of W are divided in p groups. So, there are n/p columns in each group. According to lemma 2, $n/p-2$ additional rows are needed to avoid the conflict inside a group. On the other hand, these groups are not actually similar to W , that is, their last columns don't mean actually the last column. In fact it can perform conflict probability with first column of the next group. Two groups need one additional row to remove this type of conflict and all $p-1$ additional rows need p groups. Therefore, $n/p+p-3$ additional rows are needed.

This result is in conformance with the above mechanism. In the top-down mechanism, $p = 2$ then $n/2+2-3 = n/2-1$. In the octal, $p = 4$ then $n/4+4-3 = n/4+1$.

To calculate the Optimum value for p , it suffices to differentiates $n/p+p-3$ by p , with $p = \sqrt{n}$, the minimum amount for m is $n+2\sqrt{n}-3$.

THE GROUPING SWITCHING CONTROLLER

Here, a grouping switching controller is explained. This controller is based on top-down mechanism and operates in serial fashion, pin-to-pin inside the box. The status bits of out-pins of the box must be checked out to

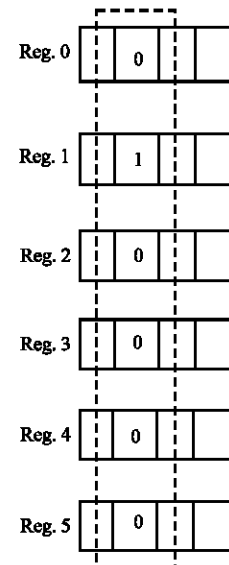


Fig. 2: Status registers representing the alternative path

determine the switching of a box and one of the free pins is selected. Each Opk_i has an n -bit register that if j th bit of status register of Opk_i is 0, it means that i th pin is occupied for OB_j . If IPk_i requires a path to OB_j , the j th bit of all status registers determine the state of free pin to OB_j . For example in Fig. 2 there is only one alternative to select a pin (second one).

The basic idea about the controller is that in a box, the pins with less selection choice is routed first. For example, if $Opjk$ has only one alternative to select and $OPik$ has two alternatives, first j th and then i th routing is determined. When a pin is routed, it is rejected from the other alternative sets. To ensure that this mechanism makes no conflict, the following lemma is proved.

Lemma 5: Starting with implementation of routing priority with low alternative pins in the box, conflict doesn't take place. Inputs of the box are divided in following subgroups:

- SUB1 : Inputs that have one alternative to select
- SUB2 : Inputs that have two alternatives to select
- SUBn : Inputs that have n alternatives to select

By adding sufficient rows to W , the network is nonblocking and because of applying the lemma 2, the columns with one alternative choice are not in the common row condition. So all of one alternative pins are in distinct rows and routing is done surely. Hence, the SUB1 groups routing is guaranteed. By routing of these pins, other two or more alternative pins may be done in two states:

- Their alternatives do not change however, it is routed from one of its alternatives
- Their alternative numbers decrease because of supplying additional row. It has at least one alternative and it carries on counting in SUB1, so it is certainly routed

The routing algorithm

- For each input pin, the number of alternative path of output pins is determined
- Inputs with low alternative choice has upper priority in routing. One of its alternatives is selected and that out-pin is removed from the alternative sets of other pins. The criterion of selection the alternative is as even column from top-to-bottom and odd column from down-to-top
- Are there any non-routed pins?
Yes: Go to step-1
No: End the switching process and record the selected switching

The time complexity analysis: The steps of algorithm have the following time complexity:

- A: Output pins are selected according to the priority mechanism, in n steps
- B: Selection recording is done in n steps. So the overall time complexity is $O(\sqrt{N})$

HARDWARE USED IN IMPLEMENTATION OF GROUPING

The MIN is a CLOS type with grouping rerouting mode. The first stage, is controlled by a central controller as shown in Fig. 3. The second stage works as a delta network. It is routed in a self-routing fashion. All of used boxes have to support the MINs permutation capability.

The box hardware: Since routing is done by the central controller, there is no need to ACK-NACK circuits inside the boxes. So, we select Multiplex based on structure inside the boxes in order to connect inputs of a box to its outputs.

Controller hardware: Figure 3 shows the controller block diagram.

- The destination code is a $\log(N)$ -bits code and so destination-code-block uses a memory with $N \cdot \log(N)$ capacity.

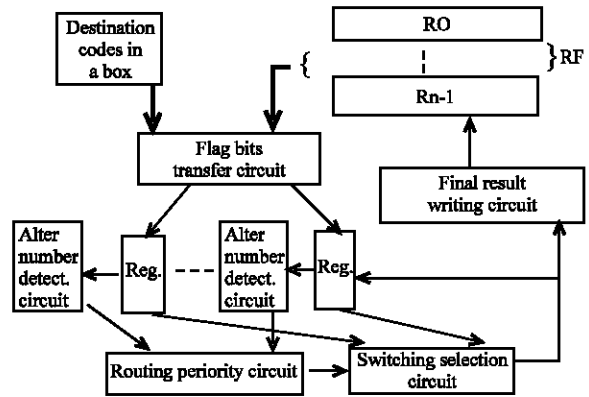


Fig. 3: Blockdiagram of the switching circuit in each box

- RF is flag-bits set that determines free or occupied pins for routing. It is in fact W^T . Each register is assigned to an output box pin, so R.F. is a $n \times m$ register.
- Flag bits transfer circuit: In routing of IB_i , according to required destination code of IB_i , content of related registers are extracted from RF by this block.
- Alternate number detecting circuit: it counts the amount of free pins for a pin.
- Priority box detecting block is to select the minimum alternative pin.
- Switching selecting circuit, selects one of valid alternative for routing pin. The selection criterion is implied mechanism, as described top-down mechanism.
- Final result registration: this block converts the 1 to 0 in RF for selected pins.

EVALUATION OF IDEA

To evaluate the advantage and the usefulness of the designed mechanism, it is compared with the other designs from two aspects: (1) The MIN architecture complexity and (2) the routing time complexity. The MIN architecture is also considered from two points of view: topology hardware and controller hardware.

The hardware comparison: Here, the hardware of the presented design, is compared with other designs regarding topology and controller hardware.

Topology hardware: The hardware specification of the presented design is compared to two other designs as shown in Table 1. First, Rana strictly nonblocking with $m = 2n-1$ and the central controller (Rana, 1992). Second, the rearrangeable CLOS network of Lee with $m = n$ and the parallel routing algorithm, under the central controller

Table 1: The hardware comparison of the grouping rerouting architecture with other CLOS MINs

Design	Second stage box No.	Total box No.	Total cross-point No.	CP increase (base is m = n)
Strictly nonblocking Rana (1992)	m = 2n-1	4n-1	3N(2n-1)	100%
Parallel algorithm. Lee and Liew (1996)	m = n*	3n	3Nn	0
top-down serial-algorithm grouping	m = 1.5n-1	3.5n-1	3N(1.5n-1)	50%
Octal serial-algorithm grouping	m = 1+ 1.125n	1+3.125n	3N(1+1.125n)	12.5%
General form serial-algorithm grouping	m = n + 2 $\sqrt{n} - 3$	3n + 2 $\sqrt{n} - 3$	3N(n + 2) $\sqrt{n} - 3$	100(2 $\sqrt{n} - 3$)/n%

*m = n is based for the comparison

strategy (Lee and Liew, 1996). Since in nonblocking CLOS MIN, minimum topology hardware is gained with m = n, so in column crosspoint (CP) increase percent of Table 1 computation, m = n is selected as reference. Each boxes in MIN with x inputs and y outputs, has xy crosspoint. This is used in total crosspoint no. computation in Table 1.

Additional boxes reduction: The second stage additional boxes are reduced according to N. As it is shown in Table 1, the hardware difference between general form of the grouping and the minimum state of v (m, n, r) i.e., m = n, is $(2\sqrt{N} - 3)/n$. Clearly when n increases, the ratio of hardware difference between the general form of grouping rerouting architecture in comparison with the minimum state of v (m, n, r), i.e., m = n, approaches zero (Table 2, Fig. 4).

Hardware of controller: Strictly non-blocking MINs have minimum controller hardware. Comparing the Rana controller and the grouping controller in Fig. 3, shows the equality of hardware complexity. Note that in RANA, is m = 2n-1 and in the grouping, m < 1.5n.

The time complexity of routing: The time complexity of grouping rerouting algorithm, is compared with two fast serial and parallel algorithms for rearrangeable 3-stage CLOS networks in Table 3.

The rearrangeable MIN with m = n, uses the serial routing algorithm with time c complexity $O(N\log(N))$ which is longer than $O(n)$ of the grouping time complexity. Also, the parallel routing algorithm, as a fast algorithm (Lee and Liew, 1996), has $O(\log^2 N)$ time complexity and a complex hardware required to solve N equations and yet the speed is less than $O(n)$ of grouping rerouting algorithm (up to N = 64 k). If we apply a parallel algorithm inside the box, the time complexity can be reduce significantly.

Table 2: The ratio hardware difference between general form of grouping and m = n

N	16	256	4 K	64 k	1024 K	16M
N	4	16	64	256	1024	4096
100(2 $\sqrt{n} - 3$)/n%	25	31	20	11	6	3 0

Table 3: The comparison of time complexity of other MINs with the grouping

Design	Time complexity	Time-complexity with N = 16 K	Time-complexity with N = 64 K	Time-complexity with N = 256 K
Walksman serial algorithm	$O(N \log N)$	224 K	1024 K	4608K
Lee and Liew (1996) Parallel algorithm	$O(\log^2 N)$	196	256	324
Serial-algorithm grouping	$O(n)$	128	256	512

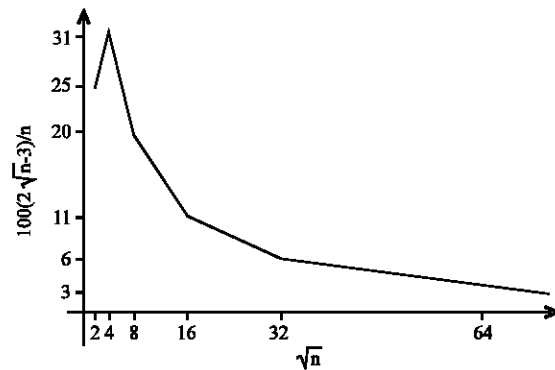


Fig. 4: Hardware difference between general form of grouping and m = n (Table 2)

Another aspect of the time parameter is the propagation delay time. Alleyne design (Alleyne Scherson, 2000), has a multi-pass process within the delta MIN, which results in a high propagation delay time, but the grouping rerouting algorithm, similar to other 3-stage CLOS MINs, only has a three-stage delay.

CONCLUSION

A new method for rearranging of a non-blocking 3-stages CLOS network is presented. This method limits the rearranging process to the box in which one of its inputs requires a path to output boxes. Non-blocking condition is analyzed and its removing is determined. It is shown that hardware redundancy is low. With a simple top-down algorithm, the number of additional intermediate boxes reduces to half.

With a general scheme for the grouping rerouting algorithm the number of boxes approaches zero for large input numbers. Since the routing algorithm is limited to a group, only low amount of inputs/outputs are required. This results in various algorithms for implementing the routing process with lower parameter.

Also, the presented idea provides the capability of time reducing and hardware complexity according to algorithm. Finally, if we apply a parallel algorithm inside the box, the time complexity can be reduced significantly.

REFERENCES

- Alleyne, B.D. and I.D. Scherson, 2000. On evil twin networks and value limited randomized routing. *IEEE Trans. Parallel Distributed Syst.*, 11: 910-925.
- Chen, X., 2007. A survey of multistage interconnection network in fast packet. *Int. J. Digital Analog Commun. Syst.*, 4: 33-59.
- Datta, S. and H. Kobayashi, 2004. Blocking probability of bicast connections in a rearrangeable CLOS network. *Advances in Wired and Wireless Communication, IEEE/Samof Symposium.* 26-27 April, IEEE., pp: 107-110.
- Hung-Lin, F.U. and F.K. Hwang, 2005. On 3-stage CLOS networks with different nonblocking requirements on two types of calls. *J. Combinatorial Optimization*, 9: 263-266.
- Hwang, F.K., H. Lee and J.D. Carpinelli, 1996. A new decomposition algorithm for rearrangement clos interconnection networks. *IEEE Trans. Comput.*, 44: 32-38.
- Lai, W.K., 2000. Performance permutation on interconnection networks by regularly changing switch states. *IEEE Trans. Parallel Distributed Syst.*, 11: 829-837.
- Lee, T.T. and S. Y. Liew, 1996. Parallel routing algorithm in Benes-Clos networks. 15th Annual Joint Conference of the IEEE Computer Societies 24-28 March, IEEE Proceedings of Infocom'96, San Francisco CA, USA., pp: 279-286.
- Morimura, T., K. Iwai and H. Amano, 2006. Hierarchical multistage interconnection network R-CLOS. *Elect. Commun. Jap.*, 89: 30-39.
- Rana, D., 1992. A control algorithm for three-stage non-blocking networks. *Proceedings of IEEE GLOBECOM'92*, 6-9 Dec, Communication for Global Users. pp: 1477-1481.
- Tian, H., K. Ajay and Y. Pan, 2006. A novel multistage network architecture with multicast and broadcast capability. *J. Supercomput.*, 35: 277-300.
- Yang, Y. and J. Wang, 2004. A fault-tolerant rearrangeable permutation network. *IEEE Trans. Comput.*, 53: 414-426.