



Crosstalk Free Routing in Optical Multistage Interconnection Networks using Fast Window Method

Abd El-Naser A. Mohamed^a, Ahmed Nabih Zaki Rashed^a,
and Ahmed Mohamed El-Eraki^b

^a Electronics and Electrical Communication Engineering Department Faculty of Electronic Engineering, Menouf 32951, Menoufia University, EGYPT

^b Faculty Engineering, Egyptian Russian University, EGYPT, Email: a.eleraki@hotmail.om, +20-100-070-0166

ARTICLE INFO

Article history:

Received 13 February 2018
Received in revised form
04 April 2018
Accepted 15 June 2018
Available online 28 October
2018

Keywords:

Optical interconnection
networks
Fast window method
Crosstalk free routing
algorithms
Parallel computing.

ABSTRACT

A major problem in parallel computing systems which are based on optical interconnection networks is optical crosstalk that degrades the performance of network, and limits the network size. This paper discusses the implementation of fast scheduling methods (fast window methods) to the traditional zero routing algorithm and in term of running time and the ability of enlarging the network. The proposed routing algorithm reduces the running time and gives the ability to increase the network size.

1. Introduction

In parallel computing systems, interconnection network work as a key element [1]. Multistage interconnection networks (MINs) organize a reliable communication between the sources and destinations in parallel computing systems [2]. The electronic MINs and the optical MINs have the same architecture, with fundamental differences such as crosstalk problem and optical-loss in the optical switches [1, 2].

Optical communication networks which is characterized by high channel bandwidth, parallel processing, and low latency that face the growing instance of high-performance communication applications and computing systems. [3].

Optical multistage interconnection networks (OMINs) are a solution for many complicated problem such as error probability, high bandwidth

requirements, and large transition capacity [4]. Available optical MINs were built mainly on banyan or its equivalent baseline and omega networks. They are fast in switch setting (self-routing) and also have a small number of switches between an input-output pair [2].

Omega network connects N input to N output nodes using n stages, where $n = \log_2 N$ with each stage containing 2^{n-1} switching elements (SEs) [5].

OMINs introduce optical crosstalk, the problem of crosstalk occurs when two signals interact with each other within at the same switching element. Optical crosstalk degrades the performance of OMINs in terms of reduced signal-to-noise ratio and limits the size of the network [6].

Reducing the effect of optical crosstalk has been a challenging issue considering trade-offs between aspects i.e. performance, hardware and software complexity. The three common approaches by which the effect of crosstalk can be reduced are through

network dilation in either the space, time or wavelength domain [7, 8].

The main target of time domain approach is permitting only one input and output to be active at a switching element at the same time slot to solve the crosstalk problem. In another description the time domain approach build a set of permutation connection which are allocated into several groups and each group is a crosstalk free group [9].

Figure 1 show the framework of the time domain approach which begins with the permutation decomposition. Permutation is randomly generated on the source and destination addresses. The result of the permutation is a one to one mapping in the optical omega network OON each source node to a corresponding destination node, which helps to build the combination matrix. Conflict checking is achieved by applying window method (WM) to the Combination matrix to schedule the messages into crosstalk free groups [10]. Window method is a way used to find which messages cannot be sent in the same group to avoid crosstalk in the network and results the conflict matrix or a conflict map [11].

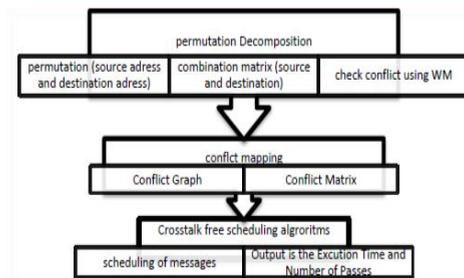


Fig. 1. Time domain approach frame work.

The conflict matrix is defined as a square matrix, M with matrix size of $N \times N$ where N is the network size. Similar to the conflict graph, the conflict matrix can be generated based on the results from the WM [9].

Given a conflict matrix M with size, generated by window method, the algorithm consists of some steps to rout the message form sours to destination without conflict; there are many ways of selecting a message for scheduling [12].

2. Data Analysis

2.1. Window methods

Window method is a scheduling technique and is used to find out which messages should not be sent in

the same group. For network size $N \times N$, with N source and N destination address, a window of size $(M-1)$ where $M = \log_2 N$ is applied to the combination matrix from the left hand side to the right hand side with the elimination of first and last column of the matrix. When two messages in the same window have the same bit pattern, they will cause a crosstalk conflict in the network. Therefore, they must be routed in different time slots, in other words, they should be routed in different groups [11]. Improved window method (IWM) was proposed as it does not check for conflicts in the first window, because the resultant conflicts are rebated in the next windows [13]. Compared to the standard WM, the execution time is reduced approximately by $1/S$ where S is the number of stages [14]. In the bitwise window (BWM) method each binary bit optical window of the standard WM is transformed into its equivalent decimal figuration using bitwise functionality [1].

The last update to the window method is Fast window method, which minimize the running time of the several WM's types by arranging each window before checking the conflict and generating the conflict matrix. This fast search method is applied to the WM, improved window method (IWM) and bitwise window method (BWM) to produce fast WM, fast IWM, and fast BWM [15].

2.2. Routing algorithms

Routing algorithms aims to be rout the messages in different independent subsets in order to avoid conflicts in the network [16].

The heuristic algorithms are the sequential increasing, degree increasing, sequential decreasing, and degree descending algorithms .Based on this method there are four strategies for selecting the message. To select a message sequentially in increasing order of the message source address or select a message sequentially in a decreasing order of the message source. Address can also can be selected based on the order of increasing degrees in the conflict graph, or on the order of decreasing degrees in the conflict graph, the degree of each message in the conflict graph is the number of conflicts it has to other messages in the conflict graph[17].

Zero algorithms Based strategy seek to reach zero matrix from the conflict matrix through summing etch column in the conflict matrix and put the results in row $N+1$ of this matrix, then select all source addresses to cross ponding zero values on that row putting them in a group, following by making cells of the conflict matrix cross ponding to elements of the

group equal to zero. Summation is applied for the new entries of the matrix with repeating the same steps until the matrix becomes a Zero matrix. Now all the entries of matrix are separate in individual groups. ZeroX, ZeroY and ZeroXY are types of that algorithm working through summing the columns, rows or applying the summation to both row and columns [18].

There are other routing algorithms that solve the crosstalk problem in optical multistage interconnection network with other strategies like Genetic algorithm, Ant colony optimization algorithm [19] Simulated Annealing algorithm [20].

Based on the comparative analysis in [16] it was concluded that all routing algorithms in consume a longer time to compute a solution compared with zero algorithm in terms of the running time, and uses minimum number of passes in the network to rout the message from source to destination.

3. Proposed Routing Algorithm

Based on comparative analysis performed in [13], it was concluded that the time spent for identifying conflicts is very high compared to routing the messages, working on this conclusion we modified the zero algorithm to reduce the running time by replacing the traditional scheduling method (window method) with fast window method.

The pseudo code of the proposed fast scheduling method is shown Fig. 2.

```

Com_Matrix =[s_add d_add]
//combination matrix
Conflict_Matrix =zeros(N,N);
//initialization of conflict
matrix NxN
For i=2 to M
//M is number of stages

For k=i+1 to i+M-1
Window[k] =Com_Matrix [i][k];

Sort (Window[k]);
For j=1:2: N-1
//N is network size
Conflict_Matrix [j][j+1]=1;
//constructing the conflict matrix
End for;
End for;
End for;
    
```

Fig. 2. Pseudo code of the fast scheduling method.

The following flow chart Fig. 3 represents the zero algorithm steps after implementing the fast scheduling method.

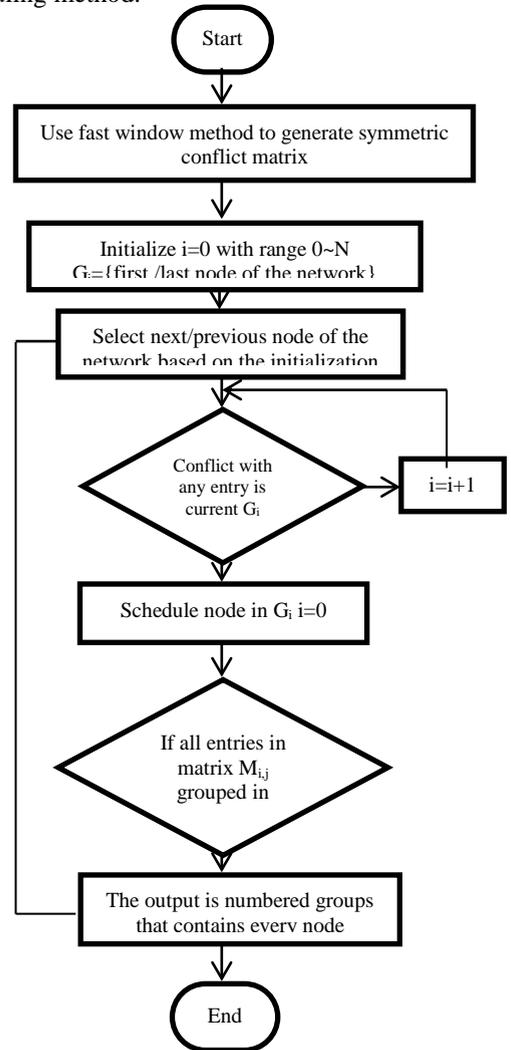


Fig 3. Flow chart of the proposed fast zero algorithm

4. Simulation Results and Comparative Analysis

This section discusses the implementation of the three fast scheduling methods such as fast WM, fast IWM and fast BWM to the zero routing algorithms. As mentioned that the time spent for identifying conflicts is very high compared to routing the messages The fast window method, fast improved window method and the fast bitwise window method take minimum time for searching the simulates in the same window, by dispending time taken in the for loops and sort the window. Hence, totally the execution time is decreased, which gives the ability to increase the network size. Based on the

programming analysis and list of operating parameters, the obtained results including the running time for the traditional network sizes and the large network size in the series of Figs. (4-9) shows comparison the fast bitwise zero, fast improved Zero, and fast Zero algorithms after implementation of the fast scheduling methods to the original zero algorithms.

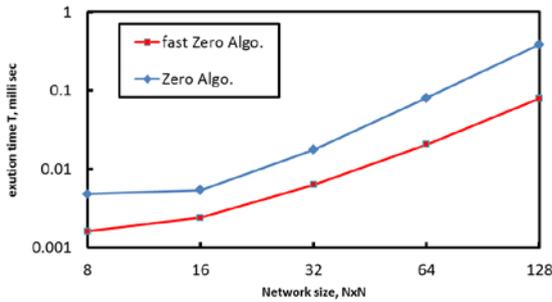


Fig. 4. Comparison between zero algorithm and fast zero algorithm in terms of running time for small network size

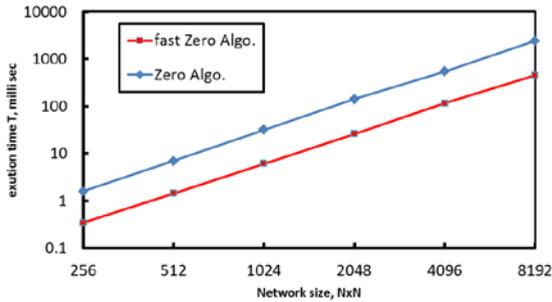


Fig. 5. Comparison between zero and the proposed fast zero in terms of time for large network size.

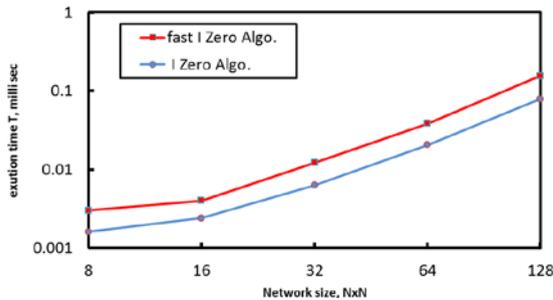


Fig. 6. Comparison between improved zero algorithm and fast improved zero algorithm for small network size

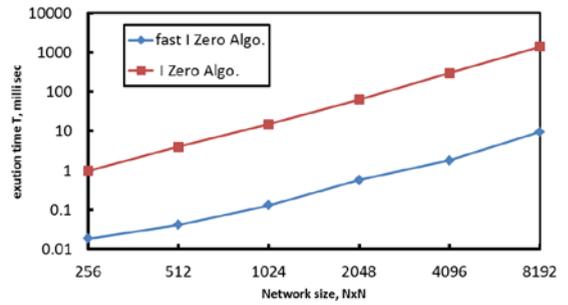


Fig. 7. Comparison between improved Zero algorithms, fast improved zero algorithm for small network size

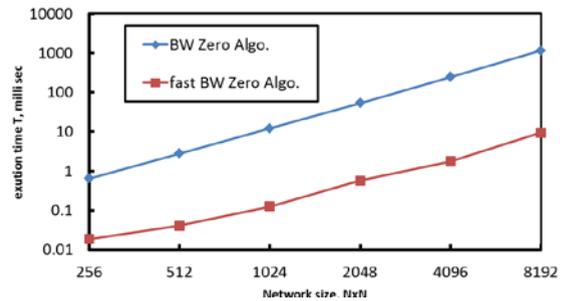


Fig. 8. Comparison between bitwise Zero algorithms before and after applying, fast bitwise window method, in terms of time for large network

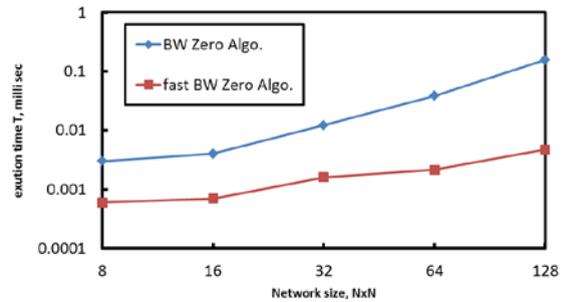


Fig. 9. Comparison between bitwise Zero algorithms before and after applying, fast bitwise window method, in terms of time for large network size

By comparing the three proposed methods fast zero, fast I zero, and fast BW zero algorithms, Fig. 10 shows the variations of the new scheduling technique in terms of running time for different network sizes

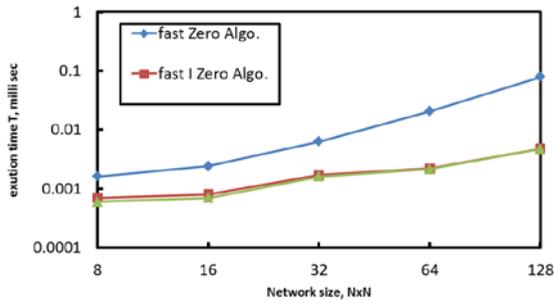


Fig. 10. Comparison between fast BW zero, fast I zero and fast zero in term of time for small network size

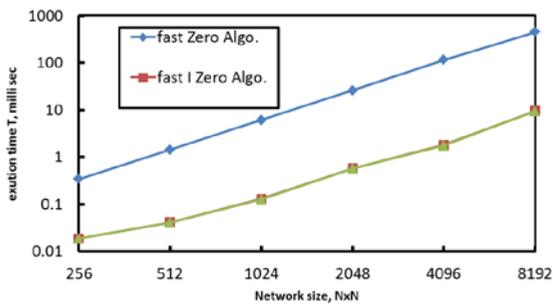


Fig. 11. comparisson between the fast bitwise zero, fast improved zero, and fast zero algorithms for large network size

5. Conclusion

After examining comparisons Figs. (4-9) which shows that the proposed fast searching function that provide a lot of time in scheduling the messages and crating the conflict matrix. Scheduling messages represent the major time in routing the messages in the (MIN), which illustrates the advantage of applying the new fast WM, fast IWM, and fast BWM on the Zero routing algorithm. Which Reduce the time taken in routing the message, at the same it gives the ability to enlarge the network size to 2^{13} computing processing unit. Because the routing time dose not excised 2 second for this large network size,

Figures. (10, 11) shows that the fast zero achieves the worst, on the other hand the fast improved zero and fast bitwise zero achieves the better time in routing the message from source to destination.

References

- [1] S. Kaur, R. Vohra and S. Kaur, "Analysis of Various Crosstalk Avoidance Techniques in Optical Multistage Interconnection Network", International Journal of P2P Network Trends and Technology. www.internationaljournalssrg.org Vol.1, No.2, pp.1-5, 2011.
- [2] S. Kaur, and N. shulka "Reducing Crosstalk in an Optical Multistage Interconnection Network" .International Journal of Computer Application, Vol.46, No.14, pp.8-13, May 2012.
- [3] M. Othman, and T. D. Shahida, "The Development of Crosstalk-Free Scheduling Algorithms for Routing in Optical Multistage Interconnection Network, Trends in Telecommunication Technologies Christos J Bouras (Ed.)," www.intechopen.com pp.643-663, March 2010
- [4] T. D. Shahaida, M. Othman and M. K. Abdullah, "Fast ZeroX Algorithm for Routing in Optical Multistage Interconnection Networks." IIUM Engineering Journal, Vol. 11, No.1, pp. 28-39 2010.
- [5] O. L. Ladouceur, "Breakthroughs in Photonics 2014: Optical Interconnection Networks" IEEE photonic journal, Vol. 7, No. 3, p.p 751-756 June 2015.
- [6] M. A. Al-Shabi and M. Othman, "A new Algorithm for Routing and Scheduling in Optical Omega Network, International Journal of The Computer", the Internet and Management, Vol 16, No.1, pp. 26-31, 2008
- [7] V. P. Bhardwaj, Nitin and V. Tyagi, "Minimization the Switch and Link Conflicts in an Optical Multistage Interconnection Networks." IJCSI International Journal of Computer Science Issues, Vol 8, No 1, pp. 1-9 2011
- [8] M. Abdullah, M. Othman and R. Johari, "Efficient Approach for Messages Routing in Optical Omega Network, International Journal of the Computer", the Internet and Mangment, Vol. 14, No.1, pp. 50-60, 2006
- [9] S. Kumar and A. Kumar, "Multistage Interconnection Networks: A Transition to Optical", IJRET International Journal of Research in Engineering and Technology, Vol. 1, No. 3, Nov 2012.
- [10] Y. Yang, J. Wang, and Y. Pan "Permutation Capability of Optical Multistage Interconnection Networks," Journal of Parallel and Distributed Computing (JPDC), Vol. 60, No. 1, Jan. pp. 72-91, 2000.
- [11] F. Abed and M.Othman "Fast Method To Find Conflict In Optical Multistage Interconnection Networks", International Journal of The Computer, The Internet and Management Vol. 16, No.1 , pp. 18-25, January-April, 2008
- [12] M. Abdullah, "Efficient Sequential and Parallel Routing Algorithms in Optical Multistage Interconnection Networks", Master Thesis, Universiti Putra Malaysia (2005).
- [13] M. Abdullah, M. Othman, and R. Johari, "Efficient Parallel Routing Algorithms in Optical Multistage Interconnection Network", IEEE International Conference on Communication Networks, 13th IEEE International Conference pp. 505-509, 16-18 Nov. 2005.
- [14] M. A. Al-Shabi, and M. Othman "A New Algorithm for Routing and Scheduling in Optical Omega Network." International Journal of The Computer, Vol. 16, No. 1, pp. 26-31, 2008
- [15] .A. K. Katangur, S. Akkaladevi and Y. Pan, "Analyzing the performance of optical multistage interconnection networks with limited Crosstalk", Cluster Computing, Vol.10, No 2, pp. 241-250, June 2007.

- [16] H. Sharshar, A. El-Eraki "Scheduling Methods Comparison Based Optical Multistage Interconnection Networks with Using Proposed Fast Window Methods" International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE) Volume 4, Issue 6, pp. 1489-1496, June 2015
- [17] A. K. Katangur, A. Somashekar, Y. Pan, M. D. Fraser. "Applying Ant Colony Optimization to Routing Optical Multistage Interconnection Networks with Limited Crosstalk", NIDISC Vol 04, April 26 2004.
- [18] A. Yunus, M. Othman "Reliability Evaluation and Routing Integration in Shuffle Exchange Omega Network", Journal of Networks, Vol. 9, No. 7, pp. 1732-1737, Jul. 2014.
- [19] A. K. Katangur, Y. Pan, M. D. Fraser, . "Message Routing and Scheduling in Optical Multistage Networks Using Simulated Annealing". IPDPS 2002.
- [20] Ramniwas, L. Lodha "Reduce the Cross Talk in Omega Network by Using Windowing Techniques", International Journal on Recent and Innovation Trends in Computing and Communication, Vol. 4, No. 6, pp. 54-57, Jun. 2016.