Optimal Path Diagnosis by Genetic Algorithm for NoCs

Setareh Shafaghi[1], Reza Sabbaghi-Nadooshan[2]

[1]Electrical Engineering Department, Central Tehran Branch, Islamic Azad University, Email: strshafaghi@yahoo.com
[2]Electrical Engineering Department, Central Tehran Branch, Islamic Azad University, Email: r_sabbaghi@iauctb.ac.ir

Abstract

Nowadays Network-on-Chips is used instead of System-on-Chips for better performance. This paper presents a new algorithm to find a shorter path, and shows that genetic algorithm is a potential technique for solving routing problem for mesh topology in on-chip-network.

Keywords: Genetic Algorithms, routing, on chip network

1. Introduction

By rapid development of the industry of semiconductors, the number of implementation units has increased in System-on-Chips (SoCs). This growth has allowed us to perform parallel processing on a chip. Network-on-Chip (NoC) is introduced for large number of processors. Due to the lack of proper performance in interconnection bus in the SoCs in these years, implementation and addressing them is important for researchers.

Selecting the appropriate routing algorithms and selecting the proper connections will result in reducing power consumption and improving performance. An on-chip network simply consists of links and nodes. Each node consists of a process element and a router. The graph that consists of connected nodes in a network design specifies network’s topology. The nodes can be connected through the most available topologies such as mesh, Ring and Torus arbitrarily. Among them, due to simple structure and easy implementation, mesh topology is used more than the others [1, 7].

Fig.1 shows an on chip network, which core of the network is connected to a switch. The cores communicate with each other via sending data packets and thus multiple paths may exist.

Delivery time of a packet to the destination directly and largely is effective on the better performance of on-chip network. This also depends on the routing technique used in on-chip network. A routing technique selects a path for sending a packet to the destination. In general, a routing technique is divided into two parts of the selecting output and input. When a packet enters the switch from one input channel, to continue its path it may have several choices. In order to select an output among the existing output channels, one suitable output is selected for directing packets to the destination. The input-selection part determines that, among several input channels, which of inputs have priority access to the output channel [6].

Thus, routing is one of the important issues in on-chip networks that has important effect on the speed of the network performance and the power. A routing algorithm is effective on the buffer consumption, the number of steps that a packet takes to reach the goal, the number of links that clocks on the router, creating a deadlock and blocked packets, the power, delay and bandwidth and area required for network implementation. Therefore selecting an appropriate routing algorithm is very important in the network. In the deterministic algorithm between a specific source and destination one path is always selected.

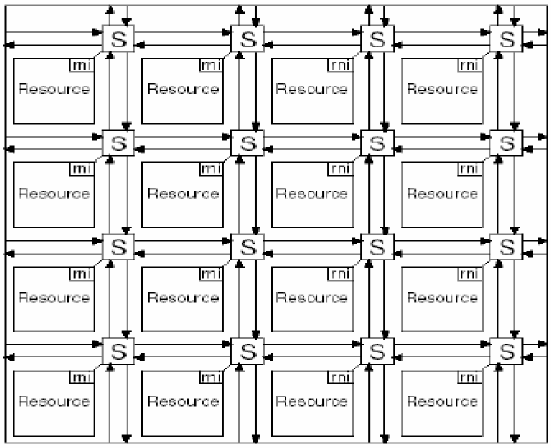


Fig.1. a sample structure for NoC

But in fully adaptive algorithms between a specific source and destination, it can choose different paths and usually will increase the performance. Of course, it is more complex the logic of these algorithms and will cause power consumption increase. As the logic is simple routing algorithm; the implementation would be more appropriate and has less delay [4]. The main advantages of the deterministic algorithms is low latency when network traffic is low and in contrast to the main advantages of the adaptive algorithms has low latency when network traffic is high [6]. Popona et al. compared four fully adaptive algorithms with a partially adaptive algorithm and a XY deterministic algorithms using wormhole switching in the interconnection networks. This study shows that a partially adaptive algorithm (North-Last algorithm) has higher latency rather than XY deterministic algorithms. The North-Last algorithm can create unbalanced traffic in the network. The result of this study was that adaptive algorithms will not necessarily reduce the delay. Vieira et al. in 2004 have examined several routing algorithms in order to perform comparison in the mesh topology. They evaluated west-first North-Last, Negative-First, XY algorithms, and this study shows that XY algorithms works in most different test cases faster than the others [4].Nowadays, it seems appropriate to use methods based on computational intelligence and methods inspired from nature to solve complex problems. The natural algorithms' advantages can be mentioned widely applying them in solving various problems, high flexibility and easy implementation. Although they can't achieve a global optimum, but in most cases they give a proper approximate solution. Among these methods neural networks, genetic algorithms, Ant colony systems, Tabu search and SA algorithms can be referred to [5].

The rest of the paper is organized as follows: in section 2 the genetic algorithm background is described. Section 3 presents the proposed methods, and section 4 presents the simulation results. Finally in section 5 discussion and conclusion are given.

2. The Background

According to existing algorithms and their advantages and disadvantages, to find the optimal routing, we must select new algorithm so that the shortest path can be chosen between source and destination. One of the best ways is genetic algorithm that is an effective search method for large spaces. It leads to orientation toward finding the response in the shorter time and it is appropriate for solving NP-Complete problems (the class of problems that are not solved in the usual methods).

2.1. The history of genetic algorithm

Genetic algorithms came about according to Darwin's theory of evolution. Then the theory of evolutionary computations was introduced by Richenberg in 1960.This theory was developed by other researchers and led to the emergence of genetic algorithms by Holand and his colleagues. J.Hu et al. offered a finite and Sub-algorithm for minimum consumption of energy in two dimensional mesh structure on the basis of XY routing. Because this method is limited and constrained, when the chip on Network is large it does not guarantee optimal solutions.

Lee and et al. proposed genetic algorithm for this purpose. Therefore, genetic algorithms are used in this field [3, 8].

2.2. Principles of genetic algorithm

These algorithms work with a series of code variables. Therefore their advantage is converting the continuance space to discrete space. One main difference with other traditional methods of optimization is that in GA we work with a population or a set of points at a certain moment, while in the old methods, we operated just for a certain point, meaning that many projects can be processed at one time by GA.

Also, it can give IP heat intelligent placement based on genetic algorithm. It has been shown that the exact IP placement can reduce the temperature of the hot spot and can give the design of the balanced heat. Another important point is that GA principles are based on random processes. The genetic algorithm is in a group of optimization methods based on imitation of evolutionary natural selection. Fig. 2 depicts the flow of genetic algorithm optimization so that the solution in the binary string is coded that is called chromosome. Instead of working with a single solution, the research was started by a random group of chromosomes called initial population. Each chromosome has a fitness rating directly associated with the optimization objective function. This approach will lead to searches with very little time [8, 2].

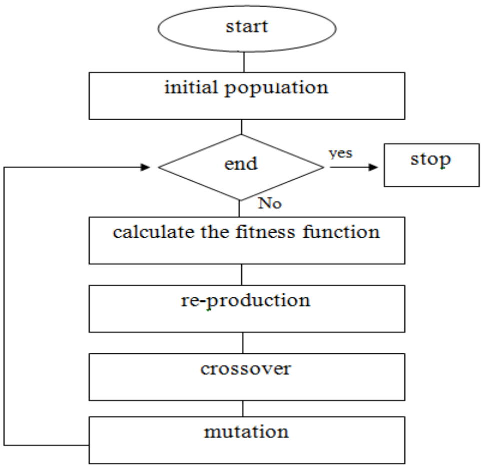


Fig.2. The flowchart of genetic algorithms

Primarily, GA should be used for the three following concepts:
1- Define the objective function or the cost function
2- Definition and implementation of genetic space
3- Definition and implementation of GA operators.
If these components define properly, GA will perform well. Finally using many changes in the systems, performance increases.

Now, briefly, we describe each genetic algorithm step as follows:

1- Encoding

This is perhaps the most difficult step of solving problems with genetic algorithm.

This step is the concept that we must continue by offering a good representation for all possible solutions of the next steps, which is important because it depends on how we proceed to this step. In fact, in this step we may build the chains of chromosome or the same bit string for possible answers. Perhaps, we can simulate with a good algorithm for answers in a reasonably good time to take before. After putting together the structure of each possible answer, the initial population of these structures can be built.

2- Evaluation

This step plays also very important role in solving problems with genetic algorithm. In this step, we realized by introducing a criterion and size population. This criterion can be expressed by assigning a number to each possible answer with being better or not of each possible answer from initial population.

3- Cross over

In genetic algorithm to create a new generation or a set of new chromosomes from answers, we applied a series of changes on selected answers or chromosomes. These series of changes we call applying operators. Our operators in these changes are two operators, crossover and mutation. We see first how to perform the crossover operator. Crossover point on the candidate answers, is randomly selected on the chromosome string. Then the left or right of that area's point is moved in the chromosome string.

4- Mutations

Mutation is also another operator that created another possible answer. This is how it functions so that it selects randomly a point on a selected chromosome and will change its form.

5- Decoding

In this step, after algorithm presented the best answer for problems, it is necessary to apply inverse encoding action on answers or the same decoding until we clearly have a real answer [8].

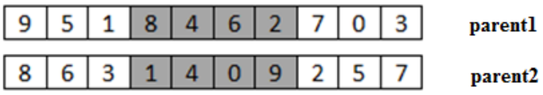
3. The proposed algorithm

The aim of this algorithm is routing and finding the shortest and lowest traffic path thus resulting in accessing more bandwidth and lower latency.

In this article in order to increase the speed of genetic algorithms compared with other genetic algorithms, a new method was proposed in the part of Initial population and Crossover operator following as below:

1. At initial population section at this method we directed the algorithm to the optimal paths, and we did not select Initial Population randomly. It means that at each step, each current node, lateral node selected from among four available options may be conducted in such a way that distance to targets will be short and make less traffic.

2. At crossover section we used a new method. In this method, two break points were determined randomly. As Fig. 3 the first part of the first parent in the offspring and the third part of the second parent is copied, and intermediate part is done on the basis of the proposed method in producing initial population.



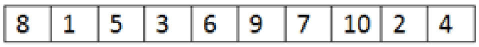
9	5	1	8	4	6	2	7	0	3	parent1
8	6	3	1	4	0	9	2	5	7	parent2

Fig.3. The selected parents for producing the next new generation

3.1. Encoding

In genetic algorithm, there are various methods for coding that is determined depending on the nature of the problem. In this project, due to the essence of the problem and interdependence between the genes, we used permutation encoding in the encoding.

Each gene can have a positive integer, thus each chromosome represents a sequence of integers that is shown in Fig. 4. It represents a path in which initial node and final node are 8 and 4 respectively.



8	1	5	3	6	9	7	10	2	4

Fig.4. The permutation encoding used in this paper

3.2. Initial population

Genetic algorithm is an iterative process and each of repeating steps are called generation (population). Population size in each iteration has introduced the number of existing chromosomes. In each iteration genetic algorithm was started with a set of solutions represented by a set of chromosomes. This set is called Initial Population that is usually created randomly.

3.3. Objective function

In genetic algorithm for each chromosome, a fitness value is assigned that its suitability of each chromosome and ability to survive and chromosome producing determines an offspring. Here, we considered the objective function in such a way to minimize the distance (minimizing the number of intermediate nodes) and to minimize the traffic path found by each chromosome with this objective function. As the route of a chromosome has passed through less node and less traffic, there will be less coefficient of error (cost).

3.4. Selection

As mentioned before, genetic algorithm starts with initial population and after evaluating the fitness of each chromosome with its introduced objective function, some of the fittest of chromosomes as the parent population are selected to create the next generation of offspring. Here, chromosomes that have fewer errors are selected as the parents of the next generation.

3.5. Crossover

The most important operator in Genetic algorithm is the crossover operator. Crossover is a process in which the old generation chromosomes are combined together, with this purpose that new generation improves in comparison with the old generation. In other words, chromosomes that are considered in the selection step as a parent, in this section exchanged their genes and created new members—chromosomes obtained from the combination are called offspring that has genes from both parents.

Different methods have been introduced as the crossover operator usually not useable in such problem.

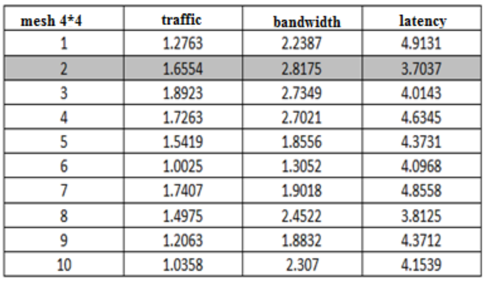
3.6. Reproduction of population

At each step of the genetic algorithm, after crossover and mutation the offspring with the best parents of the previous generation, will be replaced in new generation to be used in the next iterations as a new generation. This operation is called reproduction. Here, the parents of each iteration are transmitted directly to the next generation and others are created from combination of parents.

4. Simulation results

The evaluations of the Algorithms are presented in this section. Simulations have been done with MATLAB software version 2010. Simulations have been done on 4 × 4, 8× 8 and 16×16 two-dimensional mesh. It should be mentioned that the delay parameter is considered as the required time to find a shorter and low traffic path that each packet goes through each desired source node to each desired destination node.
Fig. 5 illustrates the best routing (2nd row at Table.1) that packet negotiates from each desired source node (1, 2) to each desired destination node (3, 3). Here the far node is considered. Each desired node also can be selected.

Table.1
10 of the best results obtained by proposed genetic algorithm in various iterations for mesh 4× 4



mesh 4*4	traffic	bandwidth	latency
1	1.2763	2.2387	4.9131
2	1.6554	2.8175	3.7037
3	1.8923	2.7349	4.0143
4	1.7263	2.7021	4.6345
5	1.5419	1.8556	4.3731
6	1.0025	1.3052	4.0968
7	1.7407	1.9018	4.8558
8	1.4975	2.4522	3.8125
9	1.2063	1.8832	4.3712
10	1.0358	2.307	4.1539

Fig. 6 illustrates the best routing (8th row at Table.2) that packet negotiates from each desired source node (1, 2) to each desired destination node (7, 7). Here again the far node is considered.

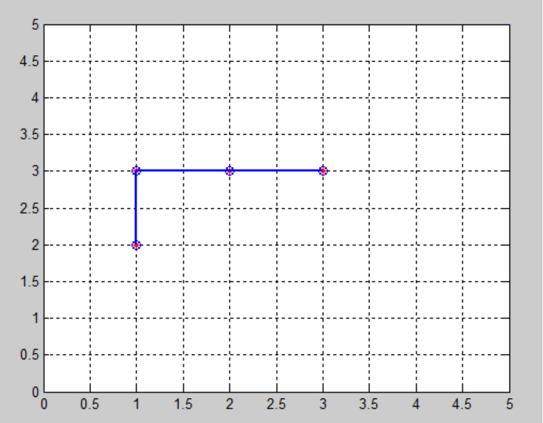


Fig.5. The best routing between desired nodes

Table.2
10 of the best results obtained by proposed genetic algorithm in various iterations for mesh 8× 8

mesh8*8	traffic	bandwidth	latency
1	5.4549	5.0728	8.4552
2	3.9888	5.8331	8.8327
3	4.4534	8.2118	8.9023
4	3.9101	5.9216	9.1772
5	4.7850	6.0766	9.6531
6	5.7097	6.9461	8.0111
7	4.5337	6.9408	8.9561
8	4.8556	7.0184	8.4022
9	4.5126	6.3082	8.5396
10	5.0831	4.7642	8.9430

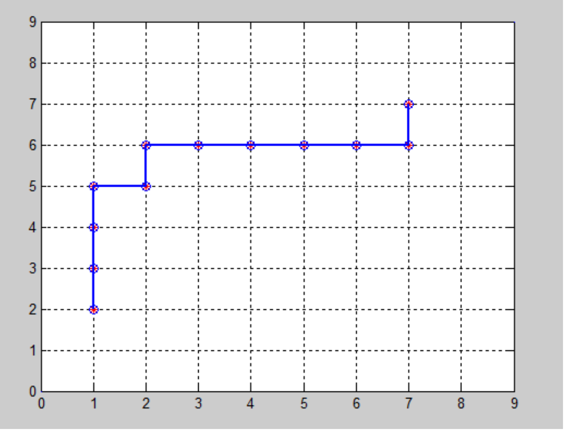


Fig.6. the best routing for mesh 8× 8

Fig.7 illustrates the best routing (1st row at Table.3) that packet negotiates from each desired source node (1, 2) to each desired destination node (15, 15). Fig. 8 compares the delay of 10 repetitions.

Table.3.
10 of the best results obtained by proposed genetic algorithm in various iterations for mesh 16× 16

mesh16*16	trafiic	bandwidth	latency
1	14.2641	16.7245	10.9021
2	13.3614	12.4393	11.1738
3	18.0184	16.1217	10.7814
4	18.2175	18.2939	11.0100
5	13.0477	16.9796	11.1614
6	15.0093	14.0755	11.1023
7	13.1067	15.4195	11.1933
8	18.3610	18.2280	10.9052
9	13.9020	18.9231	11.1357
10	15.1914	15.6052	11.2832

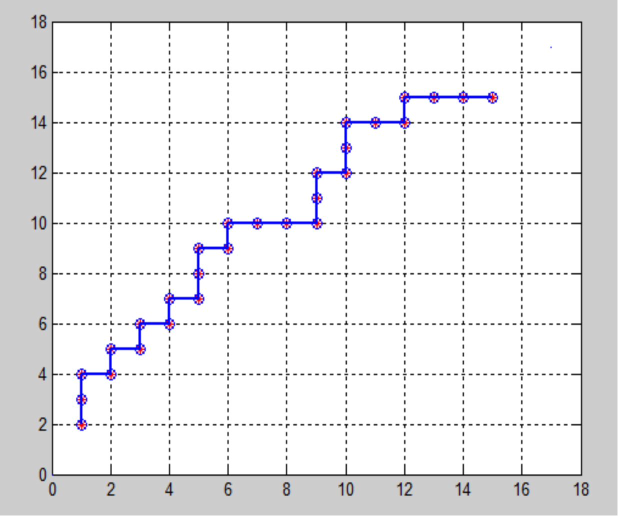


Fig.7. the best routing for mesh 16× 16

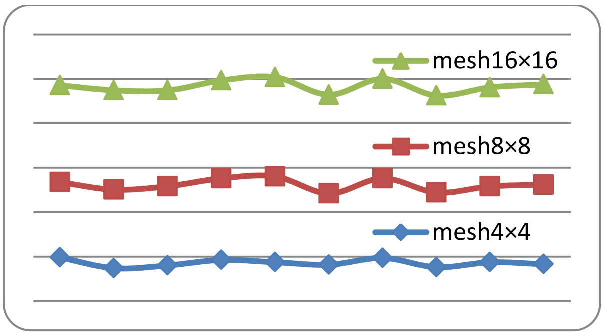


Fig.8. comparing the delay of 10 repetitions

Fig. 9 depicts the best results for each of the three previous cases. As can be seen, the larger the size of the mesh is, the better the genetic algorithm works. Among the less traffic, shorter path, less delay and bandwidth trade-off has been done. It is for this reason that the genetic algorithm gives better results in large search spaces.

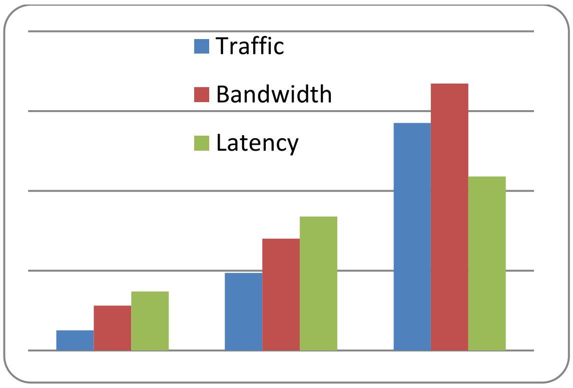


Fig.9. The best result of the cases

Fig. 10 shows the best results of consuming power obtained by proposed genetic algorithm in various iterations. From Fig. 10 it is obvious that mesh with large size consumes more power than mesh with small size. But whether the path is shorter or not, the power will reduce, and GA can find the shortest path.

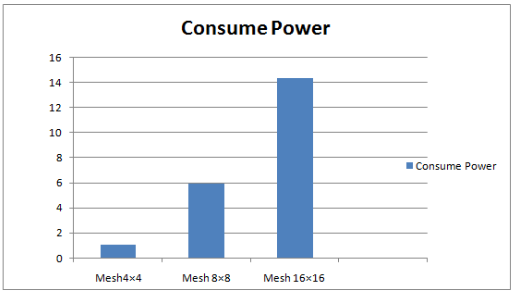


Fig.10. The best power results

5. Conclusion

In this paper, a method for routing on NoC is presented. The proposed algorithm at each step of routing is getting to target minimally. This algorithm would act better than other algorithms indifferent traffic conditions because this is customizable, problem independent and has a global search. By using this new genetic algorithm, the shortest and lowest-traffic path through each desired source and destination node have been specified in mesh topology. Therefore shorter path leads to minimum hop, delay, energy and maximum bandwidth. The performance of this algorithm is dependent on encoding and crossover. In network on chip optimized routing largely affects its performance. In proposed algorithm decision is based on the lowest traffic path. Packets move through the shortest path to the destination node. As the results of the simulations, the larger the size of the mesh is, the better the genetic algorithm works; as a matter of fact, the proposed algorithm would act better in large size of the mesh.

References

[1] Z. Emrani and K. Mohammadi, "A Technique for NoC Routing Based on Extended Compact Genetic Optimization Algorithm", Electrical Engineering (ICEE), 19th Iranian conference, pp.1-4, 2011.

[2] W. Hung, C. Addo-Quaye, T. Theocharides, Y. Xie, N. Vijaykrishnan, and M. J. Irwin, "Thermal-Aware IP Virtualization and Placement for Networks-on-Chip Architecture", Proceedings of the IEEE International Conference on Computer Design (ICCD'04), pp. 430-437, 2004.

[3] GE Fen and WU Ning, "Genetic Algorithm Based Mapping and Routing Approach for Network on Chip Architectures", Chinese Journal of Electronics, Vol.19, No.1, 2010.

[4] M. Sanjani, M. Hossein ghadiri, and M. manzoori shalmani, "Comparison Between Performance and Power in NoC Routing Algorithms", Iran's 15th conference on Electronic Engineering, 2007.

[5] A. Toroghi haghighat, M. Kord afshari, and M.Haji esmaeilian, " A multicast routing algorithm based on genetic algorithm considereing quality of services criteria", Iran's 15th conference on Electronic Engineering , 2007.

[6] M. Behroozian nejad, E. Behroozian nejad, and A. Mehran zade, "A Survey of Performance on Deterministic and Adaptive Routing Algorithm for NoC", 2nd information technology congress, present, future", 2011.

[7] R. Rejaei, F. Ahmadi kakhaki, and B. golvard zadeh, "Evaluation of Routing Algorithms and Introduce a New Adaptive Algorithm for Network on Chip (NoC)", 13th Iranian Student Conference on Electrical Engineering, 2010.

[8] P. khan teimuri, "Genetic Algorithms and TSP Problem Solving", ICT conference, Iran Science and Technology University.