International Journal of Electronics Communication and Computer Engineering
Volume 5, Issue 6, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

Performance Improvement of Application Specific Network on Chip Design using Genetic Algorithm

Mahnaz Rafie
Department of Computer Engineering, Ramhormoz Branch,
Islamic Azad University, P.O. Box 151-63825, Tohid Square, Ramhormoz, Iran
Emails: m.rafie@srbiau.ac.ir, mahnaz.rafie@gmail.com

***Abstract* – This paper presents a technique which finds a mapping of the vertices of a task graph to the tiles of a mesh based network on chip (NoC) architecture. The proposed algorithm is basically a genetic algorithm. The algorithm helps us achieve optimal or near optimal solutions in large size applications with reasonable time. In this process different types of Genetic algorithms are applied in the basic framework for solving the mapping problem on two real core graphs Video Objective Plan Decoder and mp3encmp3dec. The experimental results show the comparisons of these different meta heuristic algorithms with each other. It show that the proposed algorithm performs as well as the most previously proposed mapping algorithms considering the communication cost parameter. It is a common metric in evaluation of different mapping algorithms which have direct impact on power consumption and performance of mapped NoC.**

***Keywords* – Communication Cost, Core Graph, Genetic Algorithm, Mapping, Network on Chip.**

I. INTRODUCTION

Network on chip (NoC) is a developing and promising on chip communication paradigm that improves scalability and performance of system on chips. NoC design flow contains many problems from different areas, for example networking, embedded design and computer architecture [1]. Application mapping is one of the most important dimensions in NoC research. It maps the cores of the application to the routers of the NoC topology, affecting the overall performance and power requirement of the system [2]. It should be noted that many mapping algorithms have been recently proposed to improve several parameters used in the NoC design. One of the most important parameters is the communication cost. Using small hop counts between related cores will significantly drop the communication cost. Moreover, small hop counts will reduce the energy consumption and other performance metrics like latency. So, in this paper, the modified genetic algorithm is used to achieve the best solution of this parameter.

II. RELATED WORKS

The current researches mostly focus on mapping techniques for NoC platforms with two dimension mesh topology. The NMAP method that runs mapping with regard to bandwidth constraints and minimizing communication delay is selected as the criterion in most projects, and is highly efficient as far as communication cost is considered [3]. In [4] a binomial mapping method is introduced. The latter comes along with an optimal algorithm aiming at minimizing total traffic on network, the number of hops, and hardware costs. The Branch-and Bound algorithm, presented in [5] has been able to map IP cores on tile-based NoC architecture, and it has tried to minimize total communication energy. The performance constraint was handled via bandwidth reservation. Also, the mapping of clusters onto the physical topology of processors has been studied in the field of parallel processing. In [6], PMAP, a two-phase mapping algorithm for placing clusters onto processors is presented. In this field, the mappings produced by the PMAP algorithm are shown to have lower communication costs than mappings with previous algorithms.

In [7] genetic algorithm is used to map an application on a mesh-based NoC architecture so as to minimize the execution time. A multi-objective mapping to mesh-based NoC architectures is proposed in [8]. The approach was reformulated using genetic algorithms [9]. A multi-objective genetic algorithm based application mapping for NoC has been presented in [10], which targets mapping with Network Assignment (NA) for heterogeneous distributed embedded systems to improve the performance and reduce the power consumption and area. This technique first allocates tasks to cores, and then maps the cores to different tiles of NoC satisfying communication requirements. The mapping of IP cores onto NoC tiles, together with routing path allocation has been referred as network assignment (NA). The network assignment is usually performed after task mapping to reduce on-chip inter-communication distance. The Genetic Algorithm based optimization technique MGAP proposed in [11] minimizes the power consumption by reducing the number of switches in the communication path between cores and also maximizes the throughput. A multi-objective Genetic Algorithm (MOGA) based application mapping technique has been proposed in [12], where one–one as well as many–many mapping between switches and tiles have been taken into consideration to minimize energy consumption and required link bandwidth. It is used to find optimal solution from the pareto optimal solutions as in [11]. In [13, 14], CGMAP, a Genetic Algorithm based application mapping technique has been proposed that uses the chaotic mapping operator instead of the random processes in GA. Here the concept of chaotic sequences has been combined with genetic algorithm for an optimal mapping solution. The same authors in [15] presented a different one-dimensional chaotic mapping technique onto NoC. Here authors have combined different chaotic operators with GA to arrive at a better solution. GBMAP, an evolutionary approach for mapping cores onto NoC

architecture has been proposed in [16], which reduces energy consumption and total bandwidth requirement of NoC. GAMR [17], a genetic algorithm based mapping and routing approach addresses a two phase mapping of IP cores onto NoC architecture and generates a deterministic dead-lock free minimal routing path for each communication to minimize the total communication energy and maximize link bandwidth utilization of the NoC architecture. In [18], authors have proposed Architecture-Aware Analytic Mapping algorithm (A3MAP) for NoC with homogeneous and heterogeneous cores on regular and irregular mesh or custom architecture. The task mapping problem is solved by two effective heuristics, a successive relaxation algorithm as a fast algorithm and a genetic algorithm to find better mapping solutions. In [19], a genetic algorithm based mapping technique has been proposed for customized NoC architecture to reduce the communication energy. The same authors in [20] proposed a GA based congestion aware mapping technique for irregular customized NoC architecture to reduce the communication energy. A Multi-objective Adaptive Immune Algorithm (MAIA), based on evolutionary approach has been proposed in [21], which maps the application tasks onto NoC to reduce the power consumption and overall network latency. The adaptive immune algorithms integrate a wide set of features that improve local search while preventing the premature convergence by preserving the diversity of solutions in the population. The same authors in [22] have proposed an improved version of MAIA to solve the multi-application NoC problem. It produces a set of mapping alternatives by exploring the mapping space. The main drawback of such genetic approach is the slow rate of convergence. It often requires the GA to evolve a large number of generations to converge to a solution. The best solution at the end is taken to be the solution of the mapping problem. To accelerate the rate of convergence, the mutation rate can be increased. However, it mostly converges to local best solutions, rather than finding the global best.

III. Problem and Communication Cost

Following two concepts are introduced to formulate a mapping problem [23]:

Definition 1: The core graph is a directional graph, G (V,E), whose each vertex, $v_i \in V$ shows a core, and a directional edge, $e_{i,j} \in E$ illustrates connection between v_i and v_j. The weight of $e_{i,j}$ that is shown as $comm_{i,j}$, represents the bandwidth requirement of the communication from v_i to v_j. An IP core is showed along with a router connected to it by Resource Network Interface as a tile.

Definition 2: The NoC architecture graph is a directional graph, A(T, L), whose each vertex, $t_i \in T$, represents a tile in the NoC architecture, and its directional edge which is shown by $l_{i,j} \in L$ shows a physical link from t_i to t_j. The definitions are presented in fig 1. In core graph each edge is treated as a flow of single commodity, represented as c^k and its value which indicates required bandwidth for each edge is displayed with $vl(c^k)$. The set of all commodities represented as C is achieved as follows:

$$C = \begin{Bmatrix} c^k : vl(c^k) = comm_{i,j}, k = 1,2,\dots |E|, \forall e_{i,j} \in E, \\ \text{with } source(c^k) = map(v_i), dest(c^k) = map(v_j) \end{Bmatrix} \quad (1)$$

The core graph mapping G(V, E) on NoC architecture graph A(T, L) is defined by a one to one mapping function (Eq. (2)).

$$map : V \to T, s.t.\, map(v_i) = t_j, \forall v_i \in V, \exists t_j \in T, |V| \le |T| \quad (2)$$

Communication cost is calculated according to the Eq. (3):

$$commcost = \sum_{k=1}^{|E|} vl(c^k) \times hop_{count}\left(src(c^k), dst(c^k)\right) \quad (3)$$

Where $src(c^k)$ is source and $dst(c^k)$ is destination of c^k.

IV. Genetic Algorithm

Genetic algorithms are inspired by Darwin's theory about evolution. Solution to a problem solved by genetic algorithms is evolved. The Algorithm is started with a set of solutions (represented by chromosomes) called population. Solutions from one population are taken and used to form a new population. This is motivated by a hope, that the new population will be better than the old one. Solutions which are selected to form new solutions (offspring) are selected according to their fitness – the more suitable they are the more chances they have to reproduce. This is repeated until some condition (for example number of populations or improvement of the best solution) is satisfied. The basic operations of the genetic algorithm are simple and straight-forward [10]:

Reproduction: the act of making a copy of a potential solution:

Crossover: The act of swapping gene values between two potential solutions, simulating the "mating" of the two solutions.

Mutation: the act of randomly altering the value of a gene in a potential solution.

There are several methods for crossover operator. Some of them are as follows:

Single point: chooses a random integer *n* between 1 and number of variables, and selects the vector entries numbered less than or equal to *n* from the first parent, selects genes numbered greater than *n* from the second parent, and concatenates these entries to form the child.

Two points: selects two random integers *m* and *n* between 1 and number of variables. The algorithm selects genes numbered less than or equal to m from the first parent, selects genes numbered from *m*+1 to *n* from the second parent, and selects genes numbered greater than *n* from the first parent. The algorithm then concatenates these genes to form a single gene.

Intermediate: creates children by a random weighted average of the parents. It was shown as follows:

Child1=Parent1+rand×Ratio×(Parent2-Parent1) (4)

If Ratio is in the range [0,1], the children produced are within the hypercube defined by the parents locations at opposite vertices.

International Journal of Electronics Communication and Computer Engineering
Volume 5, Issue 6, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

Scatter: creates a random binary vector. It then selects the genes where the vector is a 1 from the first parent, and the genes where the vector is a 0 from the second parent, and combines the genes to form the child.

The different methods for mutation operator are like:

Uniform: it is a two-step process. First, the algorithm selects a fraction of the vector entries of an individual for mutation, where each entry has the same probability as the mutation rate of being mutated. In the second step, the algorithm replaces each selected entry by a random number selected uniformly from the range for that entry.

Gaussian: adds a random number to each vector entry of an individual. This random number is taken from a Gaussian distribution centered on zero. The standard deviation of this distribution can be controlled with two parameters. The Scale parameter determines the standard deviation at the first generation. The Shrink parameter controls how standard deviation shrinks as generations go by. If the Shrink parameter is 0, the standard deviation is constant. If the Shrink parameter is 1, the standard deviation shrinks to 0 linearly as the last generation is reached.

The selection function chooses parents for the next generation based on their scaled values from the fitness scaling function. During the past two decades, many selection methods have been proposed, examined and compared [10]. The common types are as follows:

Uniform: selects parents at random from a uniform distribution using the expectations and number of parents. This results in an undirected search. Uniform selection is not a useful search strategy, but it can be used to test the genetic algorithm.

Tournament: selects each parent by choosing individuals at random, the number of which you can specify by Tournament size, and then choosing the best individual out of that set to be a parent.

Roulette wheel: simulates a roulette wheel with the area of each segment proportional to its expectation. The algorithm then uses a random number to select one of the sections with a probability equal to its area.

Migration in genetic algorithms is the movement of individuals between subpopulations. The best individuals from one subpopulation replace the worst individuals in another subpopulation. Direction specifies the direction in which migration can take place.

- ✓ If you set Direction to Forward, migration takes place toward the last subpopulation. That is the *n*th subpopulation migrates into the (*n*+1)th subpopulation.
- ✓ If you set Direction to Both, the *n*th subpopulation migrates into both the (*n*–1)th and the (*n*+1)th subpopulation.

In this paper several kinds of genetic algorithms are carried out in this study and are shown in Table I.

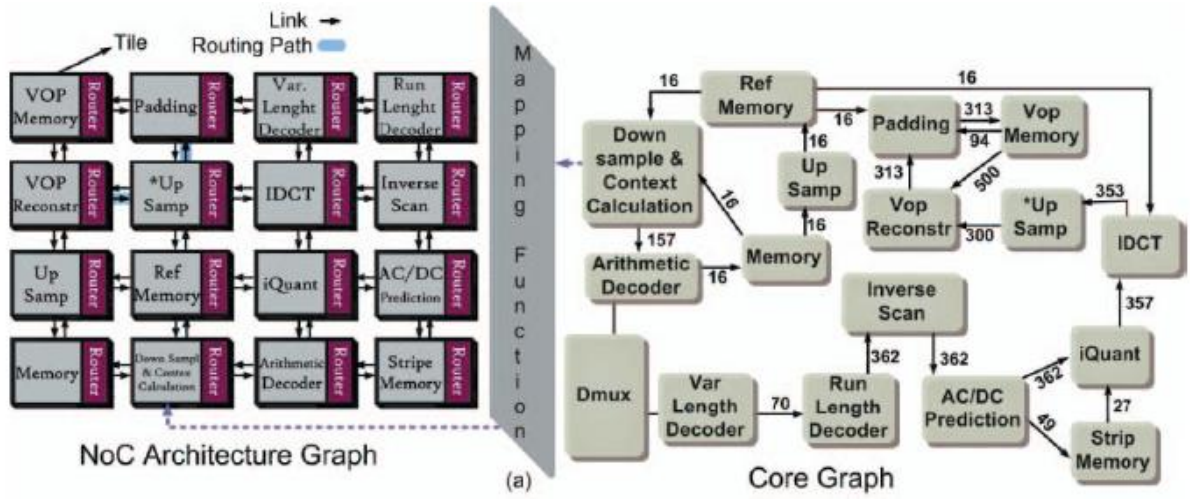


Fig.1. Mapping of VOPD core graph on NoC Architecture graph

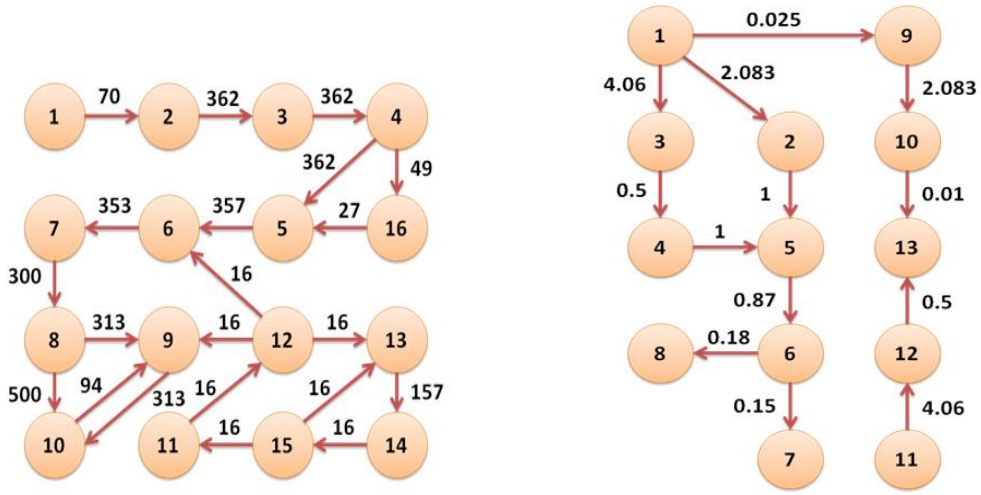
Table I: Different operators of genetic algorithms

	Selection	Mutation	Crossover	Migration
GA1	Roulette	Gaussian	Intermediate	Both
GA2	Roulette	Gaussian	Single-point	Both
GA3	Roulette	Gaussian	Two-point	Both
GA4	Roulette	Gaussian	Scatterd	Both
GA5	Roulette	Uniform	Intermediate	Both
GA6	Roulette	Uniform	Single-point	Both
GA7	Roulette	Uniform	Two-point	Both
GA8	Roulette	Uniform	Scatterd	Both
GA9	Tournament	Gaussian	Intermediate	Both
GA10	Tournament	Gaussian	Single-point	Both
GA11	Tournament	Gaussian	two-point	Both

GA12	Tournament	Gaussian	Scatterd	Both
GA13	Tournament	Uniform	Intermediate	Both
GA14	Tournament	Uniform	Single-point	Both
GA15	Tournament	Uniform	two-point	Both
GA16	Tournament	Uniform	Scatterd	Both
GA17	Uniform	Gaussian	Intermediate	Both
GA18	Uniform	Gaussian	Single-point	Both
GA19	Uniform	Gaussian	two-point	Both
GA20	Uniform	Gaussian	Scatterd	Both
GA21	Uniform	Uniform	Intermediate	Both
GA22	Uniform	Uniform	Single-point	Both
GA23	Uniform	Uniform	two-point	Both
GA24	Uniform	Uniform	Scatterd	Both

V. EXPERIMENTAL RESULTS

In this section, we present the results of the execution of the genetic algorithm with different operators on two benchmark applications, Video Object Plane Decoder (VOPD) with 16 IP-cores and 20 links and Mp3encmp3dec with 13 nodes and 13 links, which the core graphs are shown in fig 2 and the results are illustrated in table II. After wards we compare the best results of the genetic algorithms with those of previous mapping algorithms such as NMAP [24], BMAP [25], PBB [26], etc. using the same routing and scheduling characteristics. The results are presented in table III.



(a) VOPD (b) Mp3encmp3dec

Fig 2: Application core graphs with communication bandwidth (MB/s)

Table II: the results of different genetic algorithms for (a) VOPD and (b) mp3encmp3dec

(a)

VOPD	Best	Mean
GA1	4125	4189.455
GA2	4141	4238.455
GA3	4151	4465.555
GA4	4183	4257.92
GA5	4215	4409.135
GA6	4205	4351.12
GA7	4221	4310.8
GA8	4323	4543.79
GA9	4173	4295.13
GA10	4307	4307
GA11	4403	4403
GA12	4167	4167
GA13	4167	4224.625
GA14	4325	4343.265
GA15	4301	4357.655
GA16	4299	4364.98

(b)

mp3encmp3dec	Best	Mean
GA1	17.246	17.246
GA2	17.196	20.598
GA3	17.396	17.6595
GA4	17.066	18.9453
GA5	17.546	18.6036
GA6	17.416	18.0634
GA7	17.726	18.2118
GA8	17.066	17.5837
GA9	17.076	18.8842
GA10	17.531	17.962
GA11	17.346	17.3722
GA12	17.046	17.179
GA13	17.531	17.9506
GA14	17.756	18.0034
GA15	17.091	17.5976
GA16	17.726	17.9934

GA17	4613	7142.785
GA18	4247	8911.145
GA19	4285	9215.98
GA20	4269	7653.36
GA21	4339	7469.005
GA22	4337	7225.255
GA23	4281	7721.94
GA24	4437	8609.8

GA17	17.601	30.2494
GA18	17.346	32.9243
GA19	17.691	36.6797
GA20	17.436	38.87
GA21	17.761	33.4358
GA22	17.086	36.6493
GA23	17.396	33.745
GA24	17.081	33.1403

Table III: the results of different mapping algorithms for (a) VOPD and (b) mp3encmp3dec

(a)

	VOPD
GA12	4167
Cluster + ILP	4205
GMAP	5553
A3MAP-GA	4141
PBB	4317
Elixir	4249
CGMAP	4300
GBMAP	4217
PMAP	7054
BMAP	4351
CHMAP	4249
CMAP	4281
CasNet	4135
A3MAP-SR	4265
NMAP	4265
SA	4231
CSA	4169
LMAP	4189
PSMAP	4119
ILP	4119

(b)

	Mp3encmp3dec
GA12	17.046
NMAP	18.171
LMAP	17.856
PSMAP	17.021
ILP	17.021

VI. Conclusion

In this paper to validate our approaches, some experiments are done on real core graphs VOPD and Mp3encmp3dec. All of the evolutionary algorithms were carried out for 10 times with 200 initial population and 200 generations. Table II indicates the best and mean results of all proposed algorithms on two real core graphs. Based on the results, it can be concluded that the GA12 is relatively better than the rest of the algorithms in terms of mean communication cost. Also, Table III shows that, our proposed algorithm performs as well as the most algorithms considering their communication costs. In fact, GA12 achieves lower communication cost than the most algorithms. The obtained results have been modeled and simulated by MATLAB, and communication cost parameter has been calculated for the evolutionary algorithms.

Acknowledgment

This paper is based on the result of the research project named "Improving Communication Cost in Mapping of Application Specific Network-on-Chips Using Machine Learning" which is implemented with financial support of the Islamic Azad University Ramhormoz branch.

References

[1] C. Celik, and C. F. Bazlamacci, "Effect of application mapping on network on chip performance", *in Proceedings of the 20th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, pp. 465-472, 2012.

[2] P. K. Sahu, and S. Chattopadhyay, "A survey on application mapping strategies for Network-on-Chip design", *in Proceedings of the Journal of Systems Architecture*, Vol. 59, No. 1, pp. 60-76, 2013.

[3] S. Murali and G. De Micheli, "Bandwidth Constrained Mapping of Cores onto NoC Architectures", *in Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE)*, Vol. 2, pp. 896-901, Feb. 2004.

[4] T. Shen, C. H. Chao, Y.K. Lien, and A. Y. Wu, "A New Binomial Mapping and Optimization Algorithm for Reduced-Complexity Mesh-based on-chip Network", *in Proceedings of the First International Symposium Network-on-Chip (NOCS'07)*, pp. 317-322, May 2007.

[5] J. Hu, and R. Marculescu, "Energy-aware mapping for tile-based NoC architectures under performance constraints", *Asia and South Pacific Proceedings of the ASP-DAC on Design Automation Conference*, pp. 233-239, 2003.

[6] N. Koziris et al., "An Efficient Algorithm for the Physical Mapping of Clustered Task Graphs onto Multiprocessors Architectures", *in Proceedings of the 8th Euromicro Workshop on Parallel and Distributed Processing*, pp.406-413, 2000.

[7] T. Lei, and S. Kumar, "A Two-Step Genetic Algorithm for Mapping Task Graphs to a NoC Architecture", *in Proceedings of the Euromicro Symposium on Digital System Design*, pp. 180-187, 2003.

[8] G. Ascia, V. Catania, and M. Palesi, "Multi-Objective Mapping for Mesh-Based NoC Architectures", *in Proceedings of the ICHSC/ICSS*, 2004.
[9] G. Ascia, V. Catania, and M. Palesi, "A Multi-Objective Genetic Approach to Mapping Problem on Network-on-Chip", *in Proceedings of the JUCS*, Vol. 22, No. 4, 2006.
[10] A. H. Benyamina, and P. Boulet, "Multi-objective mapping for NoC architecture", *in Proceedings of the Journal of Digital Information Management*, Vol. 5, pp. 378–384, 2007.
[11] R.K. Jena, and G.K. Sharma, "A multi-objective evolutionary algorithm based optimization model for Network-on-Chip synthesis", *in Proceedings of the IEEE International Conference on Information Technology (ITNG)*, pp. 977–982 , 2007.
[12] K. Bhardwaj, and R.K. Jena, "Energy and bandwidth aware mapping of IPs onto regular NoC architectures using multi-objective genetic algorithms", *in Proceedings of the International Symposium on System-on-Chip (SOC)*, pp. 27–31, 2009.
[13] F. M. Darbari, A. Khademzadeh, and G. G. Fard, "Evaluating the performance of a chaos genetic algorithm for solving the network on chip mapping problem, *in Proceedings of the International Conference on Computational Science and Engineering*, pp 366–373, 2009.
[14] F. M. Darbari, A. Khademzadeh, and G. G. Fard, "CGMAP: a new approach to Network-on-Chip mapping problem", *in Proceedings of the IEICE Electronics Express*, Vol. 6, No. 1, pp. 27–34, 2009.
[15] G. G. Fard, A. Khademzadeh, and F. M. Darbari, "Evaluating the performance of one dimensional chaotic maps in Network-on-Chip mapping problem", *in Proceedings of the IEICE Electronics Express*, Vol. 6, No. 12, pp. 811–817, 2009.
[16] M. Tavanpour, A. Khademzadeh, S. Pourkiani, and M. Yaghobi, "GBMAP: an evolutionary approach to mapping cores onto a mesh-based NoC architecture", *in Proceedings of the Journal of Communication and Computer*, Vol. 7, No. 3, pp. 1–7, 2010.
[17] G. Fen, W. Ning, "Genetic algorithm based mapping and routing approach for network on chip architectures", *in Proceedings of the Chinese Journal of Electronics*, Vol. 19, No.1, pp. 91–96, 2010.
[18] W. Jang, and D.Z. Pan, "A3MAP: Architecture-aware analytic mapping for Network-on-Chip", *in Proceedings of the Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 523–528, 2010.
[19] N. Choudhary, M.S. Gaur, V. Laxmi, and V. Singh, "Energy aware design methodologies for application specific NoC", *in Proceedings of the NORCHIP*, pp. 1–4, 2010.
[20] N. Choudhary, M.S. Gaur, V. Laxmi, V. Singh, "GA based congestion aware topology generation for application specific NoC", *in Proceedings of the IEEE International Symposium on Electronics Design, Test, and Application*, pp. 93–98, 2011.
[21] M. J. Sepulveda, M. Strum, and W. J. Chau, "A multi-objective adaptive immune algorithm for NoC mapping", *in Proceedings of the International Conference on Very Large Scale Integration (VLSI-SOC)*, pp. 193–196, 2009.
[22] M. J. Sepulveda, M. Strum, W. J. Chau, and G. Gogniat, "A multi-objective approach for multi-application NoC mapping", *in Proceedings of the IEEE Latin American Symposium on Circuits and Systems (LASCAS)*, pp. 1–4, 2011.
[23] M. Janidarmian, A. Khademzadeh, and M. Tavanpour, "Onyx: A new heuristic bandwidth-constrained mapping of cores onto tile based Network on Chip", *in Proceedings of the IEICE Electron. Express*, Vol. 6, No. 1, pp.1-7, January 2009.
[24] S. Murali and G.D.Micheli, "Bandwidth-Constrained Mapping of Cores onto NoC Architectures", *in Proceedings of the DATE'04*, Vol. 2, pp. 896-901, Feb. 2004.
[25] W. T. Shen, C. H. Chao, Y. K. Lien, and A. Y. Wu, "A new binomial mapping and optimization algorithm for reduced-complexity mesh-based on-chip-network", *in Proceedings of the 1st International Symposium on Network-on-Chip (NOCS'07)*, June 2007.
[26] J. Hu and R. Marculescu, "Energy-Aware Mapping for Tile-based NoC Architectures Under Performance Constraints", *in Proceedings of the ASP-DAC*, pp. 233-239, Jan. 2003.

AUTHOR'S PROFILE

Mahnaz Rafie
received her B.Sc. in Computer Engineering from Allameh Mohaddese Noori Institute of Higher Education, Iran, in 2006, MSc in Computer Architecture from Azad University of Arak in 2011. She is currently a PhD candidate in Computer Architecture at Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran. Also, she is a member of young researchers club since 2010 till now. Indeed, she is with the Department of Computer Engineering, Islamic Azad University, Ramhormoz Branch. Her current research interests include network on chips, sensor networks, machine learning, and computer architectures.