IJICTR International Journal of Information & Communication Technology Research

Volume 6- Number 1- winter 2014 (33-42)

Improved Mapping Algorithms Performance in NoCs Design Based on Cellular Learning Automata

Mohammad Bagher Keley
Department of Computer Science and Research Branch
Islamic Azad University
Tehran, Iran
mbkeley@gmail.com.ir

Ahmad Khademzadeh
ICT Research Institute
ITRC
Tehran, Iran
zadeh@itrc.ac.ir

Mehdi Hosseinzadeh
Department of Computer Science and Research Branch
Islamic Azad University
Tehran, Iran
hosseinzadeh@srbiau.ac.ir

Received: September 22, 2012- Accepted: July 3, 2013

***Abstract*—NOC technology is a solution to cover the communication challenges of complex systems. The important note in the matter of application mapping for those of NoCs who are based on mesh architecture is their NP-hard problem. Also some methods have been proposed trying to overcome the mentioned problem. A low complexity mapping algorithm cannot present the optimal mapping for all applications. Then, adding an optimization phase to mapping algorithms can have an impact on their performance. This study presents an optimization phase based on Cellular Learning Automata to achieve this goal. To evaluate the proposed algorithm, we compare mapping algorithm of Nmap, CastNet, and Onyx before and after optimization. Mathematical analysis and simulation of mapping algorithms for five real applications shows that using the proposed algorithm optimizes efficiency in mapping algorithms.**

Keywords: Network on Chip; Mapping algorithm; Cellular Learning Automata; optimization algorithm

I. INTRODUCTION

A system on chip (SoC) is an integrated circuit that integrates all components of electronic system into a single chip. It may contain digital, analog, mixed-signal, and often radio-frequency functions on a single chip substrate. A typical application is in the area of embedded systems [1]. Scalability and complex communications are considered as challenging issues in the SoC. Network-on-Chip (NoC) is an effective approach to overcome scalability and complex communications problems in designing the SoC [2]. The main characteristic of NoC is the separation between computation and communication, which provides advantages such as scalability, reusability and high performance [3]. In a NoC system, modules such as processor cores, memories and specialized IP blocks exchange data using a network as a "public transportation" sub-system for the information traffic. Although there are various topologies exist for NoC architectures, the basic and well accepted topology is the mesh topology. Teraflops Research Chip of Intel [4] used mesh based topology which is known as a multi-core architecture with commercial usages. Actually 80 connected processing cores in a two dimensional mesh network strategy (2D mesh network) can be viewed in the so-called chip. Mapping the applications in an optimal way is one of the most important challenging points which exists for mesh based NoC architectures. As noted in [5] this would be

NP-hard problem when the applications are mapped on the mesh topologies.

Also group of strategies have been considered and proposed [6],[7],[8] while paying attention to energy saving as a crucial factor. The mapping idea proposed in [2] entitled PMAP considers single minimum path routing in parallel with traffic routing while the authors of [8] focused on a high speed branch-and-bound method which is capable to exploit flexible routing and also can enhance quality of made solution, as well. The authors of [9] present MOCA that utilize tree based task mapping and produces some routes (paths) on mapping result. Among the ideas proposed for solving the mentioned problems ONYX and CastNet that are revealed by [7] and [10] respectively, are assumed as two heuristic methods who utilize the mesh's symmetric feature as a starting point. CGMAP is the presented idea of [11] who uses chaos-genetic-based method which is able to achieving at close results of other meta-heuristic methods.

Integer Linear Programming (ILP) [12] based methods determine the optimum mappings in very long execution times. Cluster-based mapping method [13] proposes a clustering based relaxation for ILP formulations in lower execution times than ILP, but it does not have optimal mapping.

In [14] authors present a detailed survey of the work done in last one decade in the domain of application mapping. Apart from classifying the reported techniques, it also performs a quantitative comparison among them. Comparison has been carried out for larger sized test applications also, by implementing some of the prospective techniques. It classifies the reported techniques into groups like dynamic and static mapping approaches. Static mapping techniques have further been categorized as exact methods, branch and-bound, transformative, and constructive approaches. We have also presented a performance comparison between the static mapping techniques. Apart from the existing benchmarks, we have generated some test cases having 64 and 128 cores. Communication cost and mapping times of some of the algorithms have been compared. Thus, it provides a fair understanding of the effort needed and quality of solution obtained in different mapping approaches.

Studies show that a mapping algorithm with optimal solution for all applications is a very complicated one and because of its long time implementations is not effective enough. On the other hand, a complicated algorithm with algorithmic complexity and little time implementation cannot guarantee best results for all applications.

So, to optimize efficiency of mapping algorithm with little time implementation, using a supplementary level can be effective. In this study, an algorithm based on Cellular Learning Automata has been proposed. The algorithm can be used by all mapping algorithms as a supplementary optimization phase.

Cellular automata of (CA), [15], [16] was proposed by Van Neumann in 1940 and thereafter was proposed by a mathematician called Ulam as a model to study the behavior of complex systems. Cellular automata are, in fact, discrete dynamical systems whose behavior is based completely on local communication.

In order to evaluate the performance of our method, we have compared result of many mapping algorithm such as Nmap, Castnet and ONYX. For this purpose, some parameters such as communication cost, Maximum Bandwidth Requirement, Throughput, total energy and Delay have been used.

II. MATHEMATICAL FORMULATION OF THE MAPPING

Obviously, each application is modeled by a task graph (TG), including some nodes equivalent to the tasks and edges which indicate the relationships among the tasks. The following definitions are used to formulate mapping algorithms[6].

Definition 1: An Application Characterization Graph *(APCG)* is a directional graph, $G(V,E)$, in which each vertex denoted by $V_i \in V$ refers to one task and each directional edge, $e_{i,j} \in E$, indicates the communication between V_i *and* V_j. The weight of the edge $e_{i,j}$ represents the bandwidth of the communication between V_i and V_j [6].

The communication volume, namely the ratio of transmitted data from V_i to V_j , denoted by *CommVolume* $e_{i,j}$, is measured in Mbit/sec and is computed as shown in Eq. (1).

$$\begin{aligned} &\forall e_{i,j} \in E, \\ &\exists CommVolume_{i,j} = volume(e_{i,j}) \end{aligned} \tag{1}$$

The NoC topology is usually modeled by a directional graph. Figure 1 shows a 4×4 NoC topology.

Definition 2: The Architecture Characteristics Graph (ARCG) is a directional graph represented by g'=G (T, P), where each vertex, $t_i \in T$, refers to one tile in NoC topology and each directional edge, $p_{i,j} \in P$, refers to a path from t_i to t_j. A Task graph, g=G (Vertex, Edge), is mapped to the topology graph, g'=G (tile, path), by a one-to-one mapping function, i.e. *map( )* function, as follows[6]:

$$Size(APCG) \le Size(ARCG) \tag{2}$$

$$\forall v_i \in V, map(v_i) \in T \tag{3}$$

$$\forall v_i \ne v_j \in V, map(v_i) \ne map(v_j) \tag{4}$$

$$d\begin{cases} d^k : vl(d^k) = comm_{i,j}, k = 1,2,..|E| \ \forall e_{i,j} \in E \\ source(d^k) = map(v_i),\ dest(d^k) = map(v_j) \end{cases} \tag{5}$$

Communication Cost parameter: The communication cost is one of the evaluation criteria for mapping algorithm onto NoC, which is calculated as follows [6].

$$commcost = \sum_{k=1}^{|E|} vl(d^k) dist(source(d^k), dest(d^k)), \quad (6)$$

where $vl(d^k)$ is data transfer rate from source to destination and $|E|$ is the Number of edges.

Communication cost criteria is used to make a preliminary evaluation for mapping algorithms. To establish a complete evaluation, other parameters should be taken into account.

Bandwidth Requirement parameter: The communication cost is calculated based on total traffic of links and minimum distance among task tiles in NoC. So, the communication cost is independent of routing algorithms. However, to determine the actual Bandwidth Requirement on links, the routing path between source and destination nodes should be considered [17].

Bandwidth Requirement on links equals the amount of transferred data during implementation of the application

$$\forall link\ l_{i,j} \in L\ B(l_{i,j}) = \sum_{k=1}^{|E|} f(l_{i,j}, path(source(d^k), dest(d^k))) \quad (7)$$

$$f(l_{i,j}, path(S,D)) = \begin{cases} vl(d^k)\ if\ l_{i,j} \in path(S,D) \\ 0 \quad otherwise \end{cases} \quad (8)$$

where $l_{i,j}$ is Bandwidth requirement on link, and $f(l_{i,j}, path(S,D))$is data transfer rate from source to destination through $l_{i,j}$.

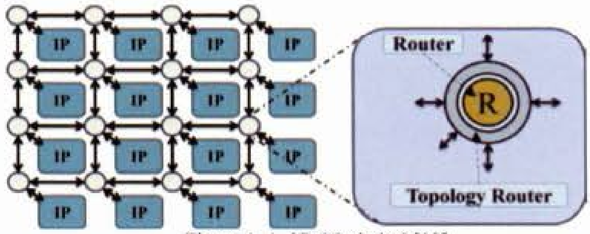

Figure 1. 1. 2D-Mesh 4×4 [18]

An efficient mapping algorithm should minimize the Bandwidth Requirement on NoC links, during the implementation process of application. Note that the mapping algorithm, dependent on the routing algorithm, must operate so that the Bandwidth requirements are less than the physical bandwidth.

$$\forall l_k;\ max(B(l_k)) = B_{noc} \leq B_{max} \quad (9)$$

where B_{max} is the maximum allowed bandwidth of the system and B_{NoC} is the minimum required bandwidth of our design.

Power Consumption parameter: A model for energy consumption in routers has been proposed in [8]. The amount of consumed energy for transferring one bit from a router to its neighbor is calculated by Eq. (10):

$$E_{bit} = E_{Sbit} + E_{Bbit} + E_{Wbit} + E_{Lbit} \quad (10)$$

where E_{Sbit}, E_{Bbit} ,E_{Wbit}, and E_{Lbit} represent the consumed energy in switch, buffer, interconnection wires inside the fabric and communication lines, respectively. The consumed energy in buffers and interior lines is trivial, compared to the E_{Lbit} $(E_{Bbit} + E_{Wbit} < E_{Lbit})$. Therefore, Eq. (10) can be represented as follows:

$$E_{bit} = E_{Sbit} + E_{Lbit} \quad (11)$$

The average consumed energy in sending one bit from tile t_i to tile t_j is calculated by Eq. (12) and it is as follows:

$$E_{bit}^{t_i,t_j} = \sum_{k=1}^{nhop} E_{Sbit} + \sum_{k=1}^{nhop-1} E_{Lbit} \quad (12)$$

and since a bit may be routed through 3×3, 4×4, and 5×5 routers, the equation Eq. (13) can be used to calculate the above parameter, as follows:

$$\begin{aligned} E_{bit}^{t_i,t_j} &= hops_{(3x3)} \times E_{Sbit(3x3)} + \\ &hops_{(4x4)} \times E_{Sbit(4x4)} + \\ &hops_{(5x5)} \times E_{Sbit(5x5)} + \\ &\sum_{k=1}^{nhop-1} E_{Lbit} \end{aligned} \quad (13)$$

where $hops_{(3\times3)}$ and $E_{sbit\ (3\times3)}$ are the number of routers with 3×3 crossbar from t_i to t_j and power consumption for 3×3 crossbar when it routes a bit, respectively. The total energy consumed for data transferring is determined by multiplying the obtained energy for one bit, by the data amount, as formulated in Eq. (14):

$$E_{total}^{t_i,t_j} = DataSize_{t_i,t_j} \times E_{bit}^{t_i,t_j} \quad (14)$$

III. CELLULAR LEARNING AUTOMATA

In this section, we briefly describe Learning Automata, Cellular Automata and Cellular Learning Automata.

Cellular Automata

A cellular automaton consists of a regular grid of cells, each in one of a finite number of states, such as on and off (in contrast to a coupled map lattice). The grid can be in any finite number of dimensions. For each cell, a set of cells called its neighborhood (usually including the cell itself) is defined relative to the specified cell. Time proceeds discretely and its rules are as general through which in each cell, considering its neighbors, it obtains a new state in each stage. The rules of cellular automata determine the way a cell is influenced by neighboring cells. We call a cell a neighboring cell when it can influence its neighbor in a stage based on the dominant rule. To obtain the current cell state, one can take into account the previous state of cell in addition to the previous state of neighbor's cell.

Definition-1: The **d**-dimensional cellular automata is a multiple $CA = (Z^d, \phi, N, F)$ so that:

- Z^d is a d-dimensional lattice of cells. This network can be a finite, semi-finite, or infinite network.
- $\varphi = \{1,...,m\}$ is a finite set of states.
- $N = \{\bar{x}_1,...,\bar{x}_{\bar{m}}\}$, $\bar{x}_i \in Z^d$, is a subset of finite Z^d set which is denoted as neighboring vector. Neighboring vector, determines the relative position of its neighbors for each u cell in cellular network as follows:

$$N(u) = \{u + \bar{x}_i \mid i = 1,...,\bar{m}\} \tag{15}$$

- $F : \phi^{\bar{m}} \to \phi$ is CA local rule.

In modeling physical and biological systems, it is sometimes necessary to take rules as probabilities. The probable behavior can be interpreted as noise in the system. A problem in cellular automata is determining definite forms of needed rules for special application. There are different solutions to this problem. One such solution is to take rules as probabilities. So, we take all rules as probable rules and assume probabilities for their activation. Yet, identifying probabilities in unknown systems remains unsolved. Then, we should be heading toward a direction in which the tool itself can extract proper rules as time passes.

Learning Automata

A learning automata is a machine which can perform finite number of action. Each selected action is evaluated by a probable environment and the evaluation result is either a positive or negative response. The automata, then, is influenced by this response in its next action. The final goal for the automata is to learn to choose the best action from between its actions. The best action is the action which maximizes the probability of getting a reward. Performance learning automata in interacting with the environment is shown in Figure 2 below.

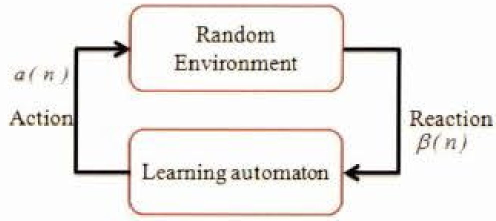

Figure 2. Probable Environment Response β(n)[19]

Environment can be shown as triple $E \equiv \{\alpha, \beta, c\}$ in which $\alpha \equiv \{\alpha_1, \alpha_2, ..., \alpha_r\}$ are input sets, $\beta \equiv \{\beta_1, \beta_2, ..., \beta_m\}$ are output sets, and $c \equiv \{c_1, c_2, ..., c_r\}$ is probable penalty set. If β is a double-member set, the environment is of $\boldsymbol{p}$ type. In such an environment, $\beta_1=1$ considered as penalty and $\beta_2=0$ as reward. In a Q-type environment β *(n)* can be a discrete value in [0, 1] interval and in s-type environment, can be the random variable in [0, 1] interval. c_i is a probability showing that α_i performance was unfavorable. In a stationary environment c_i amount remains unchanged, while in a stationary -static environment, these amounts change as time passes.

Learning automata with fixed structure is shown by a quintuple $\{\alpha, \beta, F, G, \varphi\}$ in which $\alpha \equiv \{\alpha_1, \alpha_2, ..., \alpha_r\}$ is the automata's performance set, $\beta \equiv \{\beta_1, \beta_2, ..., \beta_m\}$ are automata's input set, $\varphi \equiv \{\varphi_1, \varphi_2, ..., \varphi_s\}$ are automata's internal states, $F: \varphi \times \beta \rightarrow \varphi$ is transition function of automata and $G: \varphi \rightarrow \alpha$ is output function which maps the current state of automata to the next output.

Learning automata with variable structure can be shown by a quadruple set of $\{\alpha, \beta, p, T\}$ in which $\alpha \equiv \{\alpha_1, ..., \alpha_r\}$ is automata's performance set, $\beta \equiv \{\beta_1, ..., \beta_m\}$ is automata's input set, $p \equiv \{p_1, ..., p_r\}$ is probable vector of selecting each action and $p(n+1)=T[\alpha(n), \beta(n), p(n)]$ is learning algorithm. The algorithm below is a type of linear learning algorithm. Suppose α_i action in *n*th step is selected:

-The favorable response

$$p_i(n+1) = p_i(n) + a[1 - p_i(n)] \tag{16}$$

$$p_j(n+1) = (1-a)p_j(n) \qquad \forall j \;\; j \neq i \tag{17}$$

-The unfavorable response

$$p_i(n+1) = (1-b)p_i(n) \tag{18}$$

$$p_j(n+1) = (b/r-1) + (1-b)p_j(n) \quad \forall j \;\; j \neq i \tag{19}$$

In the relations above, P_i refers to Probability of choosing a_i and P_j refers to Probability of choosing the other actions. Also, a is the reward parameter and b is the penalty parameter. Considering a and b values, we can consider the three following states. When a and b are equal, we call the algorithm L_{RP} When b is smaller than a, we call it $L_{R\epsilon P}$ and when b equals 0, we call it L_{RI}.

C. Cellular Learning Automata

Many problems cannot be solved by learning automata alone. The main power of learning automata appears when they are used collectively. Regarding this issue, a new model was developed by combining learning automata and cellular automata [19]. The following formal definition has been given for cellular learning automata [20].

Definition 2- cellular learning automata: A d dimension cellular learning automata is a multiple $CLA = (Z^d, \phi, A, N, F)$ so that:

- Z^d ,is a d-dimensional lattice of cells. This network can be a finite, semi finite or infinite network.
- ϕ ,is a finite set of states.
- A, is a set of learning automata each of which is assigned to a cell of cellular automata.
- $N = \{\bar{x}_1, ..., \bar{x}_{\bar{m}}\}$, is a finite subset of Z^d which is denoted as neighbor vector.
- $F : \phi^{\bar{m}} \rightarrow \underline{\beta}$, is the local rule of cellular learning automata so that $\underline{\beta}$ is the total values which can be accepted as reinforcement signal.

The performance of cellular learning automata is described as the following. The learning automata select an action from between its own actions in any Iteration. This action can be selected randomly or based on previous observations. The selected action is penalized or rewarded based on selected actions of the neighboring cells and the dominating rule of the learning automata.

Then, based on whether the selected action is rewarded or penalized, the automata modify its behavior and the internal structure of the automata is updated. Updating for all automata are normally done simultaneously. After updating, each automata in cellular learning automata selects an action from its actions set again and performs it. The process of selection, rewarding and penalizing continues until the system reaches a stable state or previously defined criteria is established. The updating process of existing automata structure in cellular learning automata is carried out by a learning algorithm.

Cellular learning automata is called uniform if neighboring function, local rule and learning automata for all cell are similar, otherwise it is called non-uniform. The other kind of cellular learning automata is the open cellular learning automata (OCLA)[21]. In OCLA, in addition to local environment a global environment is also considered [21]. In OCLA, penalizing or rewarding the selected action by a cell depends not only on the selected action of its neighbors but also on the response of the global environment. It has been proved in [21] that this model like the closed CLA [20][19] can be convergent to the local optimized points for mobility rules. If the connections among learning automata are not regular (like one dimensional or two dimensional array), the cellular learning automata is called irregular [22]. If cells updating is carried out synchronously, it is called a synchronous automata otherwise it is called asynchronous [23][24].

IV. THE PROPOSED ALGORITHM

In this section, based on cellular learning automata, an approximate algorithm is proposed to optimize the efficiency of mapping algorithms. In this algorithm, first, a regular synchronous uniform cellular learning automata, which is based NOC topology. For example in Figure 3, for an NOC topology with 3×3 dimensions, systematic cellular learning automata considered with the same dimensions. The assumed neighbor type for this automata is of Van Neumann type. In other words, in this automata, each cell is neighboring the four cells which are above, below, right to, and left to it. For the bordering cells, neighborhood is defined as wrap-round.

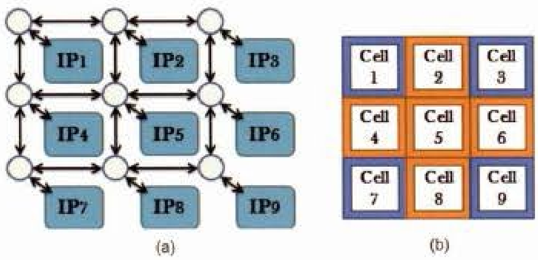

Figure 3. a) NoC with 3×3 mesh topology[16] , b) Cellular Learning Automata with 3×3 cells.

Each of the learning automata in the cellular learning automat has five actions of 'Exchange with the above neighbor', 'Exchange with the below neighbor', 'Exchange with the right neighbor', 'Exchange with the left neighbor' and 'no exchange'.

$$\alpha \equiv \{\alpha 1, \alpha 2, \alpha 2, \alpha 2, \alpha 2\} = \{\text{'Ex\_above '}, \text{'Ex\_ below'}, \text{'Ex\_ right'}, \text{'Ex\_ left '}, \text{'No exchange'}\} \quad (20)$$

The number of each cell in cellular learning automata is the number of the tile in NOC topology of the problem (from 1 to $n \times m$).

Learning automata in each cell of L_{RP} type has the rewarding and penalizing rate of 0.01. The possibility of selecting actions of every learning automata equals 0.2 (1/5) in the beginning. The proposed algorithm called CLA-mapping-optimization is as the following: Each cell selects simultaneously an action from the existing actions. Then the $comm\_cost_x$ for the influenced cells by the selected action is calculated. If the calculated $comm\_cost_x$ is less or equal than $comm\_cost_x$ in before action selection, action gives reward, else action gives penalty. Procedure continues to the time when the selected action by cells in a stage is 'not changing'. Finally, the obtained arrangement is presented and tested as the optimized NOC topology.

The $comm\_cost_x$ parameter calculated in term of Eq.(21).

$$comm\_cost_x = \sum_{k=1}^{deg(v_x)} volume(e_{x,k}) * (dist(map(v_x), map(v_k))) \tag{21}$$

The pseudo- code of this algorithm is shown in Figure 4.

```
Algorithm CLA-mapping-optimization
Input: Graph G(V,E), mapped task graph
Output: Optimized mapped task graph
{Construct a regular
        CLA for mapped task graph
Repeat
        For all cells do in parallel
        Old_Comm ← Comm_cost_x
            Select an action
        Compute cell's Comm_cost_x
            If (the cell's Comm_cost_x ≤
                        Old_Comm )
                  Reward the cell's action
            else
                  Penalize the cell's action
            End If
        End For
Until all cell's action are no change action
Return the mapped task graph
}
```

Figure 4. The proposed algorithm

V. Results and Discussions

To evaluate the performance of optimization algorithm, we compared mapping algorithms, such as Nmap [6], CastNet [10] and Onyx [7] from mathematical analysis and simulation viewpoints, before and after optimization phase. For this purpose we selected five video applications; namely, Video Object Plane Decoder (VOPD) [6], MPEG-4 decoder [25], 263EncMp3Dec [26], 263DecMp3Dec [26], and Mp3EncMp3Dec [26].

A. Mathematical analysis

In the mathematical analysis, some parameters such as communication cost, maximum bandwidth requirement and power consumption were used.

In mathematical analysis, the communication cost of VOPD for mentioned mapping algorithms, before and after optimization phase, were calculated and presented in Table 1.

Optimization phase behaves in a way so that communication cost is reduced as far as possible and communication cost Parameter can affect maximum bandwidth requirement and power consumption Parameter. Then, the improved ratio for these parameters is calculated in term of Eq.(22)

For Improved Ratio parameter, a positive sign shows the success of optimization phase. If it has the N.C (No Change), it means that the optimization phase has had no effect, and if the sign is negative, it means the optimization phase has had a negative effect.

$$Im\ proved\ \ Ratio = \frac{Value\ \ of\ \ parameter_{Before\ \ Im\ proved} - Value\ \ of\ \ parameter_{After\ \ Im\ proved}}{Value\ \ of\ \ parameter_{Before\ \ Im\ proved}}$$

Table 1. Improved Ratio of Communication Cost for Mapping algorithms

Task Graph	Before Improved			After Improved			Improved Ratio		
	Nmap[6]	CastNet[10]	Onyx[7]	Nmap	CastNet	Onyx	Nmap	CastNet	Onyx
VOPD	4297	4135	4119	4265	4135	4119	0.74%	N.C (NO.CHANGE)	N.C
MPEG-4	3672	3852.5	3817.5	3672	3772.5	3772.5	N.C	2.08%	1.18%
263EncMp3Dec	230.432	230.417	230.417	230.417	230.417	230.417	0.01%	N.C	N.C
263DecMp3Dec	17.948	17.948	18.361	17.948	17.948	18.148	N.C	N.C	1.16%
Mp3EncMp3Dec	17.521	17.046	17.046	17.021	17.046	17.046	2.85%	N.C	N.C

In Table 1, columns two to four give the communication cost for the mapping algorithms generated by Nmap, CastNet and Onyx, respectively. The next three columns show the communication cost for mapping methods after optimization phase. The three last columns display the improved ratio of communication cost for application task graph.

The result shows that the optimization phase could reduce the communication cost for some applications in different mapping algorithms. We should notice that compared to other mapping algorithms for mapping task graph of different applications, the mapping algorithms of NMAP, CastNet, and Onyx have less communication cost. The optimization phase for the algorithm of the increased mapping has less time and can optimize communication cost to 0.74%. The optimization could reduce communication cost parameter %1.34 on average for the mapping algorithms of Nmap, CastNet and Onyx for the mentioned benchmarks. Mapping algorithms of Nmap, CastNet and Onyx cannot guarantee the best results for all applications and each of them is useful for some applications. Then, if a mapping algorithm gives the best result for a application, the optimization phase is not useful. For example, Onyx mapping for VOPD could give the best mapping with communication cost of 4119 and no other mapping algorithm could give a communication cost less than 4119.

Therefore, the optimization phase has not been effective; however, the same algorithm could not give the best result for MPEG-4 and 263DecMp3De. Then the optimization phase has been very effective and could give the best result for the output of Onyx mapping algorithm for MPEG-4 and 263DecMp3Dec applications. Other similar cases have been generated for the other mapping algorithms. The important thing is that the optimization algorithm could give positive effect for all mapping algorithms.

Table 2 shows the maximum bandwidth requirement and comparisons among mapping algorithms, regarding the XY routing method. As the improved ratios in columns seven to nine show optimization phase improved maximum bandwidth requirement in mapping algorithms for some applications such as: VOPD for Nmap, MPEG-4 for CastNet, MPEG-4 and 263DecMp3Dec for Onyx. Our algorithm has optimized the limitation of maximum bandwidth requirement to 3.40%.

We use the power/energy model presented in [8] to estimate the energy consumption of each mapping algorithm, in term of Eqs.(13),(14). In [18], the results of applying energy model over routers, topology switches and links are shown inTable 3 It is mentioned that a packet contains a 96 bits of data. The result of energy consumption in Table 4 shows that our algorithm could reduce power consumption.

TABLE 2. IMPROVED RATIO OF MAXIMUM BANDWIDTH REQUIREMENT FOR MAPPING ALGORITHMS

Task Graph	Before Improved			After Improved			Improved Ratio		
	Nmap	CastNet	Onyx	Nmap	CastNet	Onyx	Nmap	CastNet	Onyx
VOPD	829	813	613	813	813	613	1.93%	N.C	N.C
MPEG-4	926.25	942	926	926.25	910	910	N.C	3.40%	1.73%
263EncMp3Dec	46.733	46.733	46.733	46.733	46.733	46.733	N.C	N.C	N.C
263DecMp3Dec	4.06	4.06	4.172	4.06	4.06	4.06	N.C	N.C	2.68%
Mp3EncMp3Dec	4.06	4.06	4.06	4.06	4.06	4.06	N.C	N.C	N.C

TABLE 3. CHARACTERIZATION OF THE ROUTERS, TOPOLOGY SWITCHES AND LINK [18]

Module	Energy per packet(pJ)
Link,1 mm	21
5×5 Router (Topology switch)	32(0.6/0.8)
4×4 Router (Topology switch)	31(0.6/1.1)
3×3 Router (Topology switch)	30(0.6/1.3)

TABLE 4. IMPROVED RATIO OF POWER CONSUMPTION FOR MAPPING ALGORITHMS.

Task Graph	Before Improved(pJ)			After Improved(pJ)			Improved Ratio		
	Nmap	CastNet	Onyx	Nmap	CastNet	Onyx	Nmap	CastNet	Onyx
VOPD	3621.5083	3516.6313	3520.7536	3540.2563	3516.6313	3520.7536	2.24%	N.C	N.C
MPEG-4	3193.6082	3282.2266	3262.7813	3193.6082	3238.7096	3238.7096	N.C	1.33%	0.74%
263EncMp3Dec	203.8280	203.8199	203.8199	203.8199	203.8199	203.8199	0	N.C	N.C
263DecMp3Dec	15.7475	15.7425	16.0631	15.7475	15.7425	15.9396	N.C	N.C	0.77%
Mp3EncMp3Dec	15.0866	14.8910	14.8910	14.8025	14.8910	14.8910	1.88%	N.C	N.C

B. Simulation results

We have performed the simulations by NOXIM tools [27] and the results have been reported in term of some parameters, such as total received packets, max delay , throughput and total energy. The simulation has been performed during 12000 cycles, and the evaluation has begun from cycle 1000. In this simulation, the XY routing algorithm is used and the buffer size is 4 flits. In evaluation of optimization phase, optimal solution is a case in which max delay and total energy parameters reduces and total received packets and throughput increases. Then the improved ratio for max delay and total energy parameters is calculated based on the relation of Eq.(22) and total received packets and throughput are calculated based on the relation of Eq.(23) The total received packets shows the amount of sending and receiving Packets in the network. Total received packets in Table 5 shows that using optimization algorithms in mapping methods generates optimization 1.22% in sending and receiving packets.

$$Im\ proved\ \ Ratio = \frac{Value\ \ of\ \ parameter_{After\ \ Im\ proved} - Value\ \ of\ \ parameter_{Before\ \ Im\ proved}}{Value\ \ of\ \ parameter_{Before\ \ Im\ proved}} \tag{22}$$

TABLE 5. IMPROVED RATIO OF TOTAL RECEIVED PACKETS FOR MAPPING ALGORITHMS

Task Graph	Before Improved			After Improved			Improved Ratio		
	Nmap	CastNet	Onyx	Nmap	CastNet	Onyx	Nmap	CastNet	Onyx
VOPD	14272	14220	14350	14389	14220	14350	0.82%	N.C	N.C
MPEG-4	6396	6391	6451	6396	6450	6456	N.C	0.92%	0.08%
263EncMp3Dec	6888	6870	6820	6920	6870	6820	0.46%	N.C	N.C
263DecMp3Dec	8718	8657	8688	8718	8657	8729	N.C	N.C	0.47%
Mp3EncMp3Dec	8804	8969	8854	8911	8969	8854	1.22%	N.C	N.C

TABLE 6. IMPROVED RATIO OF MAX DELAY FOR MAPPING ALGORITHMS

Task Graph	Before Improved			After Improved			Improved Ratio		
	Nmap	CastNet	Onyx	Nmap	CastNet	Onyx	Nmap	CastNet	Onyx
VOPD	15928	18027	13775	15683	18027	13775	1.54%	N.C	N.C
MPEG-4	18162	18513	16831	18162	18639	17054	N.C	-0.68%	-1.32%
263EncMp3Dec	15824	15933	18641	15774	15933	18641	0.32%	N.C	N.C
263DecMp3Dec	13529	13181	13668	13529	13181	13771	N.C	N.C	-0.75%
Mp3EncMp3Dec	15451	15597	15498	15728	15597	15498	-1.79%	N.C	N.C

TABLE 7. IMPROVED RATIO OF THROUGHPUT FOR MAPPING ALGORITHMS

Task Graph	Before Improved			After Improved			Improved Ratio		
	Nmap	CastNet	Onyx	Nmap	CastNet	Onyx	Nmap	CastNet	Onyx
VOPD	0.3003	0.2991	0.3032	0.3013	0.2991	0.3032	0.33%	N.C	N.C
MPEG-4	0.2029	0.1845	0.2024	0.2029	0.1866	0.2029	N.C	1.14%	0.25%
263EncMp3Dec	0.2719	0.2729	0.3110	0.2721	0.2729	0.3110	0.07%	N.C	N.C
263DecMp3Dec	0.2273	0.2275	0.2280	0.2273	0.2275	0.2277	N.C	N.C	-0.13%
Mp3EncMp3Dec	0.2525	0.2572	0.2540	0.2556	0.2572	0.2540	1.23%	N.C	N.C

Studying the maximum delay parameter in Table 6 shows that our algorithm has decreased the max delay in some cases and increased it in some other cases. The proposed algorithm based on the reduction communication cost makes the optimization operations possible and may cause traffic on some points and therefore increases time delay. You should notice that by changing optimization parameter, optimization algorithm can be changed so that it can be useful for maximum delay parameter. Table 7 shows that throughput has also been optimized. Using optimization algorithm in Onyx mapping algorithm for the 263DecMp3Dec has decreased Throughput, because of the increase in time delay. Using the optimization phase could be 0.19% useful on average in mapping algorithms for all applications.

TABLE 8. IMPROVED RATIO OF TOTAL ENERGY FOR MAPPING ALGORITHMS

Task Graph	Before Improved			After Improved			Improved Ratio		
	Nmap	CastNet	Onyx	Nmap	CastNet	Onyx	Nmap	CastNet	Onyx
VOPD	8.750E-05	8.577E-05	8.639E-05	8.745E-05	8.577E-05	8.639E-05	0.06%	N.C	N.C
MPEG-4	5.412E-05	5.526E-05	5.528E-05	5.412E-05	5.512E-05	5.417E-05	N.C	0.25%	2.01%
263EncMp3Dec	4.101E-05	4.119E-05	4.087E-05	4.108E-05	4.119E-05	4.087E-05	-0.17%	N.C	N.C
263DecMp3Dec	5.026E-05	5.022E-05	4.987E-05	4.951E-05	5.022E-05	4.987E-05	1.49%	N.C	N.C
Mp3EncMp3Dec	5.211E-05	5.170E-05	5.107E-05	5.154E-05	5.170E-05	5.107E-05	1.09%	N.C	N.C

Table 8 shows that optimization algorithm could reduce total energy consumption 2.01%. Of course in Nmap mapping algorithm for the 263EncMp3Dec energy consumption has increased -0.17%. Regarding total energy consumption, using the optimization algorithm could be 0.23% useful for mapping algorithms in all applications.

VI. CONCLUSION

A low complexity mapping algorithm cannot guarantee optimal mapping for different applications in designing NOC. Then using a complementary phase for optimization in mapping operation can prove useful. In this study, we have used Cellular Learning Automata for optimization of the efficiency of mapping algorithms in designing NOC. In this method, a new algorithm using a set of five actions is used which uses the aspects of learning CLA and tries to select better states for each cell based on reduction cost of communication in mapping algorithms and the increase of NoCs performance. The obtained results of the mathematical analysis and simulation have shown that this algorithm is efficient for mapping operations.

Using optimization algorithm for mapping algorithms for all applications in mathematical analysis has improved to 0.53% for communication cost, 0.65% for maximum bandwidth requirement and 46% for Power Consumption. In simulation result, it has optimization for total received packets %26, for throughput %0.19, for total energy %0.32 and it has a negative effect of %0.18 for max delay.

The proposed algorithm based on the reduction communication cost makes the optimization operations possible and may cause traffic on some points and therefore increases time delay.

You should notice that by changing optimization parameter, optimization algorithm can be changed so that it can be useful for maximum delay parameter.

VII. REFERENCES

[1]. S. Pasricha, and N. Dutt, On-Chip Communication Architectures: System on Chip Interconnect, first ed., Morgan Kaufmann, Burlington (MA), 2008, chapter 12.

[2]. L. Benini , and G.D. Micheli, "Networks on Chips: A New SoC Paradigm", IEEE. Computer, 35(1) (2002) 70–78.

[3]. http://techresearch.intel.com/articles/Tera-Scale/1421.html.

[4]. R. Lemaire, F.Clermidy, Y. Durand, D. Lattard, and A. Jerraya, "Performance Evaluation of a NoC-Based Design for MC-CDMA Telecommunications using NS-2", in RSP'05 Intl Conference, 2005.

[5]. M.R. Garey, and D.S. Johnson, Computers and Intractability: A guide to the theory of NP-completeness, San Fransisco (CA), Freeman and Co, 1979.

[6]. S. Murali, and G. De Micheli, "Bandwidth-Constrained Mapping of Cores onto NoC Architectures", in: Proceedings on Design, Automation and Test in Europe, 2004.

[7]. M. Janidarmian, A. Khademzadeh, and M. Tvanpour, "Onyx: A new heuristic bandwidth constrained mapping of cores onto tile-based Network-on-Chip", IEICE Electron. Express, 6(1), January 2009, 1-7.

[8]. J. Hu, and R. Marculescu, "Exploiting the routing flexibility for energy/performance aware mapping of regular NoC architectures", in: Proceedings of the Conference on Design, Automation and Test in Europe ,2003.

[9]. K. Srinivasan, and K.S. Chatha. "A technique for low energy mapping and routing in Network-on-Chip architectures". In: Proc ISLPED'05, San Diego, California, 2005, p.387–92.

[10]. S. Tosun, "New heuristic algorithms for energy aware application mapping and routing on mesh-based NoCs", J. System Architecture, 57(1) ,January 2011, 69-78.

[11]. F. Moein-darbari, A. Khademzade, and G. Gharooni-fard. "CGMAP: a new approach to Network-on-Chip mapping problem". IEICE Electron Express, 2009, 6(1), 27–34.

[12]. S. Tosun, O. Ozturk, and M. Ozen. "An ILP formulation for application mapping onto Network-on-Chip". In: 3rd International conference on application of information and communication technologies, AICT2009, Azerbaijan, Baku, 2009.

[13]. S. Tosun, "Cluster-based application mapping method for Network-on-Chip", J. Advances in Engineering Software, July 2011, 868–874.

[14]. P. Kumar Sahu, and S. Chattopadhyay, "A Survey on Application Mapping Strategies for Network-on-Chip Design", J. Advances in Engineering Software, October 2012.

[15]. S. Wolfram, "Cellular Automata", Los Alamos Science, 1983, Vol. 9, pp. 2-21.

[16]. S. Wolfram, "Universality and Complexity in Cellular Automata", Physica D., 1984, No. 10, pp. 1-35.

[17]. M. Janidarmian, M. Tinati, A. Khademzadeh, M. Ghavibazou, and A. Roshan Fekr, "Special issue on a Fault Tolerant Network on Chip Architecture", AIP, 2010

[18]. M. Stensgaard et al. "ReNoC: A Network-on-Chip architecture with reconfigurable topology". In NOCS, 2008, 55–64.

[19]. M.R. Meybodi, H. Beigy, and M.Taherkhani, "Cellular Learning Automata and Its Applications", Journal of Science and Technology, University of Sharif, Autumn/Winter 2003-2004, No. 25, pp.54-77.

[20]. H.Beigy, and M.R.Meybodi, "A Mathematical Framework for Cellular Learning Automata", Advanced in Complex Systems, 2004, Vol. 7, No. 3&4, pp. 294-319.

[21]. H. Beigy, and M.R.Meybodi, "Open Synchronous Cellular Learning Automata", Proceedings of the 8th world Multi-conference on Systemic, Cybernetics and Informatics(SCI2004), pp. 9-15, 2004.

[22]. M.Asnaashari, and M.R.Meybodi, "Irregular Cellular Learning Automata and Its Application to Clustering in Sensor Networks", Proceedings of 15th Conference on Electrical Engineering (15th ICEE), Volume on Communication, Telecommunication Research Center, Tehran, Iran, 2007.

[23]. H. Beigy, and M.R.Meybodi,, "Asynchronous Cellular Learning Automata" Automatica, Journal of International Federation of Automatic Control, 2007.

[24]. H.Beigy, and M.R., Meybodi, "Asynchronous Cellular Learning Automata", Proceedings of 10th Annual CSI Computer Conference Iran, Telecommunication Research Center, Tehran, Iran, 2005, pp. 271-280.

[25]. T. Shan, 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", Proceedings of the First International Symposium on Networks-on-Chip (NoCS'07), IEEE (2007).

[26]. K. Srinivasan, K.S. Chatha, and G. Konjevod, "Linear-programming-based techniques for synthesis of network-on-chip architectures", IEEE Trans. Very Large Scale Integration, 2006, Syst. 14 (4) 407–420.

[27]. http://noxim.sourceforge.net/.

Mohammad Bagher Keley was born in Dezful. He received hir B.Sc. degree in Computer Engineering from Islamic Azad University, Dezfoul branch, Iran in 2003, and is M.Sc. degree in Computer Architecture Engineering from Science and Research Islamic Azad University, Tehran, Iran, in 2006. His research interests include design and analysis of reliable, fault tolerant and mapping in Network-on-Chips and 3-Dimentional Network-on-Chip interconnections.

Ahmad Khademzadeh was born in Mashhad, Iran, in 1943. He received the B.Sc. degree in applied physics from Ferdowsi University, Mashhad, Iran, in 1969 and the M.Sc., Ph.D. degrees respectively in Digital Communication and Information Theory & Error Control Coding from the University of Kent, Canterbury, UK. He is currently the Head of Education & National Scientific and International Scientific Cooperation Department at ICT Research Institute (ITRC). He was the head of Test Engineering Group and the director of Computer and Communication Department at ITRC. He is also a lecturer at Tehran Universities & he is a committee member of the Iranian Electrical Engineering Conference Permanent Committee. Dr. Khadem-Zadeh has been received four distinguished national and international awards including Kharazmi International Award, and has been selected as the National outstanding researcher of the Iran Ministry of Information and Communication Technology.

Mehdi Hosseinzadeh Was born in Dezful in 1981. He received his B.Sc. degree in Computer Hardware Engineering from Islamic Azad University, Dezful branch, Iran in 2003. He also received his M.Sc. and Ph.D. degrees in Computer System Architecture from the Science and Research Branch, Islamic Azad University, Tehran, Iran in 2005 and 2008, respectively. He is currently an assistant professor in Department of Computer Engineering of Science and Research Branch of Islamic Azad University, Tehran, Iran. His research interests are computer arithmetic with emphasis on residue number system, cryptography, network security and e-commerce.