Modified Genetic Algorithm Based Method on Low-Power Mapping in Network-on-chip

Dai Qi-hua[1,a], Liu Qin-rang[1,b], Shen Jian-liang[1,c], Sun Miao[1,d]

[1]National Digital Switching System Engineering & Research Center, Zhengzhou, Henan 450002, China.

[a]278591289@qq.com,[b]qinrangliu@sina.com,[c]jlshen@163.com,[d]smilesun3@126.com

Keywords: Network-on-Chip; low-power; mapping; modified genetic algorithm; structural feature; better initial solution

Abstract: Focusing on the power decreasing of large-scale applications in network-on-chip, this paper proposed amodified genetic algorithm based method on low-power mapping. With communication weights of task nodes and structural features of mapping platform, this method acquired better initial mapping solution set with the consideration of task node priority and its connection. Moreover, we introduced the roulette wheel selection, best-neighbor selection and reverse evolution, and selected the initial solution with a certain probability at each iteration to prevent the algorithm stagnation.Experimental results show that, when maintaining the same task model and mapping platform, compared with the genetic algorithm and random mapping algorithm, our proposed algorithm greatly decreases the energy consumption.

1 Introduction

With the stable development of deep submicron IC technology, the number of components integrated on a chip becomesmuch more. According to ITRS(International Technology Roadmap for Semiconductors), the physical dimension of semiconductor devices is expected to be reduced to 8nm by 2020[1], and the complexity of chips will have a further improvement. Traditional SoC(System-on-Chip)[2] technology (such as Bus structure , the effective-address space) seriously restricted the rapid development of the chip. Thus, researchers put forward a new kind of integrated circuit structure in 2001, named NoC (Network-on-Chip)[3]. From the SOC to the NOC, energy consumption has become the primary limiting factor in chip designing. The designof integrated circuit focuses more on energy consumption demand than chips' functional requirements.

In this paper,the optimal solution is obtained based on MGA (Modified Genetic Algorithm). GA (Genetic Algorithm)has a strong global search capability, but its local search capability is poor, and is easy to fall into the trap of premature. It will take a long time to find the optimal solution, so the GA needs to be improved. This paper prioritized the nodes according to the communication weight of task nodes, and nodes with highest communication volume were mapped first, which makes genetic manipulations starting from an optimal initial solution. In each iteration, MGA made use of the initial solution in a certain probability, to ensure the diversity of population and to avoid the stagnation problem; meanwhile, roulette turn bet, the best neighbors and evolutionary reversal strategy were used to adjust the direction of evolution and improve the convergence speed.

2 Related work

It has been a hotspot in the NoC technology field to study how to successfully reduce the total energy consumption through mapping a given set of task nodes onto a regular NoC architecture effectively under the given constraints.

The author of [4] adopted the ILP(integer linear programming) based on clustering. This algorithm used the clustered-based scheme to reduce the ILP space and the search time, and then obtained the best mapping. However, the standard of cluster was questionable, which made the idea difficult to promote. The work in [5] mixed the genetic algorithm and colony algorithm, using the

rapid search capability of genetic algorithm to obtain a better initial solution. The assignment of the ant colony route was given according to the optimal sequence of these optimal solutions, and was used to initialize the pheromone concentration distribution. The positive feedback mechanism of ant colony algorithm was fully used to search for the better solution of mapping. In [6], AGA(Adaptive Genetic Algorithm) was presented. This algorithm regulated the crossover and mutation ratein each iteration based on the best fitness and the average fitness, thus best-fit mapping could be gained much quicker. In literature [7], particle swarm optimization algorithm mapping was used.The population initialization module initializedthe position of particles; the optimal particle update module updatedwhen the new personal best particle and particle position limits changed; non-inferior solution set update module selected optimum solution based on the communication relationship of new particles; iterationcontinuedtill the approximate optimal solution mapping is obtained.

In general, these algorithms can reduce the NoC communication energy consumption, but there is still plenty of room for continuing optimizing the convergence speed and optimization effect.

3 NoC Mapping

The NoC mapping refers to a process that the task nodes are mapped to the appropriate resource nodes in ARCG (Architecture Characteristic Graph) under the design constraints of a given task, topology and bandwidth. It will make a smooth and efficient completion of various applications, and the energy consumption is minimized at the same time [8]. As we can see in Fig 2, once task graph (TG) has been processed by mapping algorithm, each task node will be distributed to 2D Mesh resource. Thus, NoC mapping problem is abstracted to the mapping problem between TG and ARCG. NoC mapping problem belongs to the NP (Non-deterministic Polynomial) problem. When it comes to a large-scale problem, accurate solution is almost impossible and it takes a long time to find the solution, which makes the feasibility quite low. Therefore, there is a need to get the help of swarm intelligence algorithm for its fast search capabilities, and then find the better mapping result smoothly and efficiently.

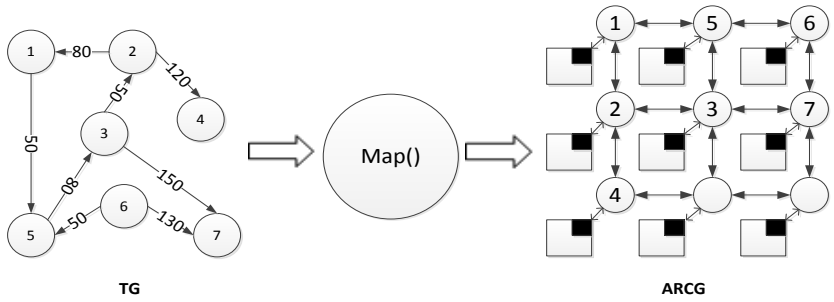


Fig 1Mapping flowchart

In order to make the study more representative, the regular NoC 2D Mesh topology is used in this paper. This Mesh has a regular structure, and its symmetry and extensibility is also good. What is more, 2D Mesh supports multiple routing protocols, flow control technology and exchange algorithm. Its two-dimensional structure can easily be mapped to the silicon surface, which means it's easy to implement. Therefore, it is currently one of the most widely used topologies [9].

3.1 Definition

Before elaborating the details of the mapping algorithm, we firstly give the necessary definitions.

Definition1. A TG is a directed acyclic graph, as is shown in figure 1, Vertex $t_j \in$ T represents a task node, and directed arc v_{t_i,t_j} represents the relationship between two task nodes t_i and t_j, shown in formula 1. D_{t_i} represents the number of nodes t_i connected with in the application except itself. The value of the arc between t_i and t_j on behalf of the amount of data transfer between two nodes in Mbits/s.

$$d(v_{t_i,t_j}) = \begin{cases} 1 & \text{nodes i,j connected} \\ 0 & \text{nodes i,j not connected} \end{cases} \quad (1)$$

$$D_{t_i} = \sum_{j=1}^{M} d(v_{t_i,t_j}) \quad (2)$$

Definition 2. A ARCG is also a directed acyclic graph， shown in figure 1. Vertex $p_{i \in} P$ denotes a resource node in regular 2D Mesh. Each edge $l_{i,j}$ denotes a physical link between $p_i$and p_j. $h_{i,j}$represents the Manhattan distance between $p_i$and p_j.

It is a one-to-one correspondence relationship between the task nodes and resource nodes. Therefore, in order to ensure the correct implementation of the mapping results, the number of resource nodes in ARCG must be greater than equal to the number of task nodes in TG, namely:

$$Size(TG) <= Size(ARCG) \quad (3)$$

Mapping algorithm *map()* distributes the task nodes of TG to the resource nods of ARCG, namely:

$$\forall t_i \in T, map(t_i) \to p_m \in P \quad (4)$$

$$\forall t_j \in T, map(t_j) \to p_n \in P \quad (5)$$

$$t_i \neq t_j \quad (6)$$

$$p_m \neq p_n \quad (7)$$

Formula (4), (5), (6)and (7) indicate that the task node and the resource node havea one-to-one relationship. Each task is completed by a resource node, while each resource node executes only one task. When the mapping is finished, hop count between task nodes are also finalized.

3.2 Energy model

In this paper, we use awell-acceptedenergy model given in formula (8). In this energy model, E_{bit} is the estimate energy consumption of a single bit transferring from a source node to a destination node on the NoC Mesh.

$$E_{bit} = E_{s_{bit}} + E_{B_{bit}} + E_{W_{bit}} + E_{L_{bit}} \quad (8)$$

In formula (8), $E_{s_{bit}}$, $E_{B_{bit}}$ and $E_{W_{bit}}$ denote the energy consumption of a bit on the routers, buffer and internal circuit. $E_{L_{bit}}$ denotes the energy consumption of a bit transferring between two adjacent resource nodes on physical links.

With the current technique, the size of resource nodes on 2D NoC Mesh has reached nanometer level. The energy consumption of a single bit transferring on the router and internal circuit is much less than that on physical links [10].

$$E_{B_{bit}} + E_{W_{bit}} \ll E_{L_{bit}} \quad (9)$$

So formula (8)can be simplified as this：

$$E_{bit} = E_{s_{bit}} + E_{L_{bit}} \quad (10)$$

The energy consumption of a signal bit transferring between resource nodest$_i$,t$_j$ is:

$$E_{bit}^{t_i,t_j} = (h_{t_i,t_j} + 1) * E_{S_{bit}} + h_{t_i,t_j} * E_{L_{bit}} \quad (11)$$

h_{t_i,t_j} denotes the hop count between resource nodest$_i$, t$_j$, usually expressed in Manhattan distance. Assuming that regular NoC 2D Mesh is described in a XY axis, then h_{t_i,t_j} represents as $\left|X_{t_i} - X_{t_j}\right| + \left|Y_{t_i} - Y_{t_j}\right|$.

The size of regular NoC 2D Mesh is N*N. And the total amount of energy consumption of the network can be calculated using the following formula:

$$E_{NoC} = \sum_{i=1}^{N} \sum_{j=1}^{N} W_{t_i,t_j} * E_{bit}^{t_i,t_j} \quad (12)$$

After the power model is simplified, the most direct way to reduce energy consumption is to

reduce the sum of Manhattan distance. More precisely, the Manhattan distance between the task nodes which have larger communication weights should be reduced.Calculating and comparing the energy consumption under various mapping results, then find the minimum energy consumption mapping result, which is also shown in the following formula:

$$Min\left\{E_{NoC}=\sum_{i=1}^{N}\sum_{j=1}^{N}W_{t_i,t_j}*E_{bit}^{t_i,t_j}\right\} \tag{13}$$

4 Modified Genetic Algorithm Based Low-Power Mapping Method

GA is an evolutionary algorithm. Its basic principle is to emulate the evolutionary law in biosphere, "natural selection, survival of the fittest". In the genetic algorithm, the problem parameters are encoded as chromosomes, and then an iterative approach is used for crossover and mutation. By doing this, the coding information in chromosome can be exchanged, the most dominant chromosome can be closer, and the chromosome which is satisfied with the optimization goals can be generated. GA has broad search surface and coverage, and it can reach approximate optimal solution area at a fast speed. It is an ideal heuristic algorithm to solve NoC mapping problems. Traditional GA has strong global search capability but poor local search ability and easy to premature convergence. In view of these weaknesses, some improvements have been made in this paper.

4.1 Algorithm Pseudocode

Algorithm: MGA

Input：Maxgen，//* evolution generations *//
Sizepop，//* population size *//
Pcross，//* crossover probability *//
Pmutation，//* mutation probability *//
MeshsizeNoC，//* NoC Mesh size *//
Ini，//* initial solution use probability *//
Com()，//*communication matrix*//

Output: Chrom，//* best fitness chromosome*//

```
initialize the parameters
for n=1:Sizepop  begin
get optimum initial solution set P
end
calculate the average fitness of the P
for i=1:Maxgen  begin
    select operation of the P
    //*roulette turn bet, the best neighbors and evolutionary reversal strategy were used to adjust the direction of evolution and improve the convergence speed *//
    crossover operation of the P
reverse evolution operation of the P
      for j=1:Sizepop  begin
        recalculate average fitness of the P
        find the individual chromosome in the P which has the best fitness
      end
end
```

1） Set algorithm parameters

The definition and the set of the basic parameters are shown in Table 1. Parameter setting has great impact on MGA. If the mutation gets easily to happen, the average and the best fitness of each generation may not be able to fit well, and it will be unable to reach a stable value, which makes it easier to have experimental errors. According to the reference [11], combining the task features as well as the structural features, we set and adjust the parameters.

Table 1 Parameter definition

Parameter	Set up
Maxgen	150
Sizepop	100
Pcross	0.7
Pmutation	0.01
Meshsize	N
Ini	0.05

2） Initialize code, generate task-node communication matrix

Get the number of task nodes M in task graph TG (*t, v*), and then decimal coding task nodes, giving 1-M values that cannot be repeated to each task node. Extract the communication volume of each task node with other task nodes, to form a M * M communication matrix.

3） Construct optimum initial solution set:

Known from the above analysis, ARCG(*p,h*) is N*N regular NoC 2D Mesh. At the beginning of GA operation, all initial mapping is random distribution. M task nodes will lead to $N^2!/(N^2-M)!$ mapping results in Mesh Architecture. It is impossible to have all of the random distributions substituted into the calculation as the initial population solution when the number of task nodes is large. This is because if so, the size of the population is too large, and it will be an inefficient evolution. In order to decrease the number of times of iterations, and reach the optimal solution area as soon as possible, we build an optimum initial solution set, reducing the time in generating the optimal solution.

Steps of constructing optimum initial solution set

Step 1: Calculate the communication volume of each task node $Com_{t_i} = \sum_{j=1}^{M} W_{t_i,t_j}$, the higher the communication volume, the higher priority the task node.

Step 2: Take the task node with highest priority for the first mapping node. Assuming the communication volume of multiple tasks nodes are identical, take the task node with lager average communication volume for the first mapping node. As is shown in formula 14, the denominator means the task node *i* is connected with $\sum_{j=1}^{M} d(v_{i,j})$ task nodes. The lager is theA_i, the higher is the priority.If the total amount of communication volume as well as the A_i of more than two task nodes are equal, then choose one randomly. Table 2 is the priority of nodes calculated by the above rules.

$$A_i = \sum_{j=1}^{M} W_{t_i,t_j} \Big/ \sum_{j=1}^{M} d(v_{i,j}) \tag{14}$$

Table 2 Priority assignment for TG in Fig 1

Task *i*	Com_{t_i}	A_i	Priority
1	130	65	6
2	250	83	3
3	280	93	2
4	120	120	7
5	180	56	5
6	180	90	4
7	280	140	1

Step 3: select the mapping node in ARCG (*p*, *h*).

Regular NoC 2D Mesh structure is symmetrical. In order to minimize the whole hop count between the highest priority task nodes, we first place the task node with the highest communication weight onto the symmetric resource node. Regular NoC 2D Mesh has multiplesymmetric resource nodes. And the number of the free one hop distance neighbors of the symmetric resource nodes is 2,

3 or 4. After mapping the first task node to the symmetricresource node, we are trying to map the remaining task nodes. If the number of task nodes connected with the mapped task node is less than or equal to 4, we choose the resource node whose one distance hop neighbors are as same as the number; otherwise, we choose the resource node whose one distance hop neighbors will maximize the difference between the number of task nodes connected with the mapped task node and the number of one hop distance resource nodes. If there are more than one candidate resource nodes, randomly choose one. Next, we select the highest priority task node connected with the mapped task node and map it onto the resource node connected with the mapped resource node, which results in the decreasing of the whole hop count. As is shown in Fig 2, we first map 2 or 3 task nodes which have highest communication weight, and then randomly map the remaining task nodes one by one onto the Mesh. When all task nodes are mapped, a better initial population is done.

4) GA manipulation. On the basis of optimum initial solution, producing new chromosomes by making use of the operation of choice, crossover and mutation in every iteration, and then the dominant populations is constructed. Reach the optimal value throughrepeated evolution.

Step1: selection. Regard the energy consumption of mapping result as fitness, the greater the power, the smaller the fitness, and otherwise.We use roulette bet turn, the optimal neighbor strategy to select advantage chromosome. The dial is evenly divided into M areas, so that each chromosome is drawn with equal probability. Contrast the fitness of chromosome selected by roulette bet turn and its neighbors, choose chromosomes with large fitness and make them into a new population for crossover and mutation. Due to the rapid breeding of few predominant individual, the diversity of the population is undermined. It makes it easy for genetic manipulation to fall into the trap of premature convergence. Therefore, in each iteration, the initial solution chromosomes replace the chromosomes chosen by GA operation with a certain probability, ensuring the diversity of the population and preventing the premature convergence.

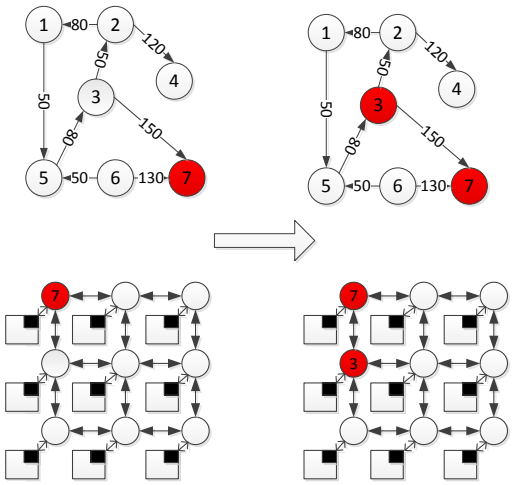


Fig 2Task node with larger communication volume map first

Step 2: Crossover. Crossover is an important operation to generate new solutions. Select two chromosomes as well as the chromosomes intersections randomly. The intersections of two chromosomes are identical, ensuring the exchange parts of chromosome have the same length. Gene at the same position (from the beginning of the chromosomes to the mutative parts of them) of two chromosomes is replaced by each other. Gene duplication on chromosomes may occur after the replacement, which is in need of gene repair, so that the reliability of the chromosome can be ensured. As is shown in Fig 3, taking the task graph in Fig 1 as an example, every chromosomeconsists of 7genes.

Step 3: Mutation. Mutation is to select a chromosome randomly, and choose one gene on that chromosome arbitrarily to mutate. Gene mutation may cause gene duplication, which is unable to meet the conditions of the algorithm. Thus, gene repair is required to ensure that there is no presence of the same genes on the same chromosome.As is shown in Fig 4.

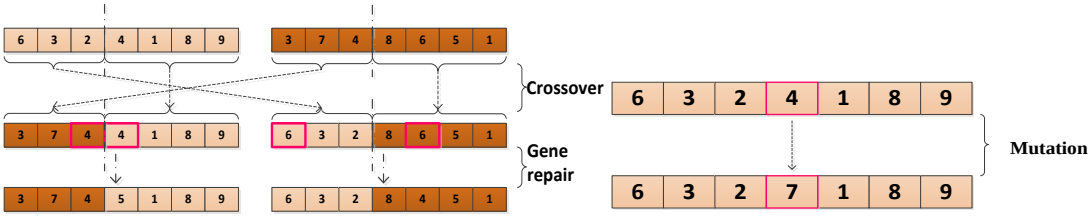


Fig 3 CrossoverFig 4 Mutation

5 Experiment and analysis

5.1 Mapping algorithm

In order to prove that the algorithm can effectively reduce the energy consumption of NoC communication, this paper conducted a series of tests. Mapping algorithm source code is written in C ++, compiled and simulated in the Linux environment. In order to reflect that MGA algorithm is effective to a variety of applications, this paper used the actual communication task graph and random communication task graph to test.Attributes are shown in Table 3. The task graph MPEG-4, VOPD, 263 Enc, 263 Dec [12] are typical practical application diagrams, G1, G2 are application diagrams of different sizes randomly generated by software TGFF [13]. 6 kinds of application diagrams are different in the number of task nodes, the number of edges of communication and total communication volume. They can reflect the merits of the mapping algorithm more comprehensively. The ARCG size was 4 * 4,5 * 5,6 * 6, corresponding to the application diagrams. BothGAand MGA areintelligent heuristic algorithm, and there either is a certain randomness. To ensure the accuracy of the test results and improve the representation of the experimental data, this paper calculated each task graph in different Mesh size for 20 times under different mapping algorithms. The result ofenergy consumption was recorded for each calculation, and took the average of 20 calculate results.

Table 4 compared the energy consumption value calculated by MGA, AGA, GA and random mapping.Figure 3 is a histogram showing the average of energy consumption value, which is generated by 6 kinds of task graph using different mapping algorithm on 3 different scale Meshes. Assuming the random mapping energy consumption value is 100%, and the vertical axis is the ratio of the energy consumption value calculatedby MGA, AGA, GA mapping algorithm to the energy consumption valuefrom random mapping。

Table 3 Graph characteristics

Graph	Nodes	Edges	Com_{t_i} （MB/S）
MPEG-4	12	13	3466
VOPD	16	21	3471
263 Enc	12	12	230.214
263 Dec	14	15	19.627
G1	7	8	710
G2	17	20	790

Table 4 Energy consumption comparisons of MGA, GA, and Random

App	Node Num	Energy consumption comparison								
		4*4			5*5			6*6		
		>AGA	>GA	>Ran	>AGA	>GA	>Ran	>AGA	>GA	>Ran
MPEG-4	12	0.86%	1.40%	37.10%	1.46%	3.09%	43.54%	2.75%	4.74%	49.10%
VOPD	16	1.53%	2.02%	37.50%	1.66%	2.23%	47.30%	2.64%	4.61%	56.45%
263 Enc	12	0.74%	1.02%	33.26%	1.78%	3.88%	45.58%	3.12%	5.63%	53.08%
263 Dec	14	1.58%	2.72%	34.17%	2.01%	3.65%	45.18%	3.23%	5.68%	51.20%
G1	7	1.64%	2.74%	34.91%	6.15%	14.64%	45.47%	8.58%	15.04%	43.04%
G2	17	0	0	0	4.68%	8.03%	48.94%	4.23%	7.72%	50.61%

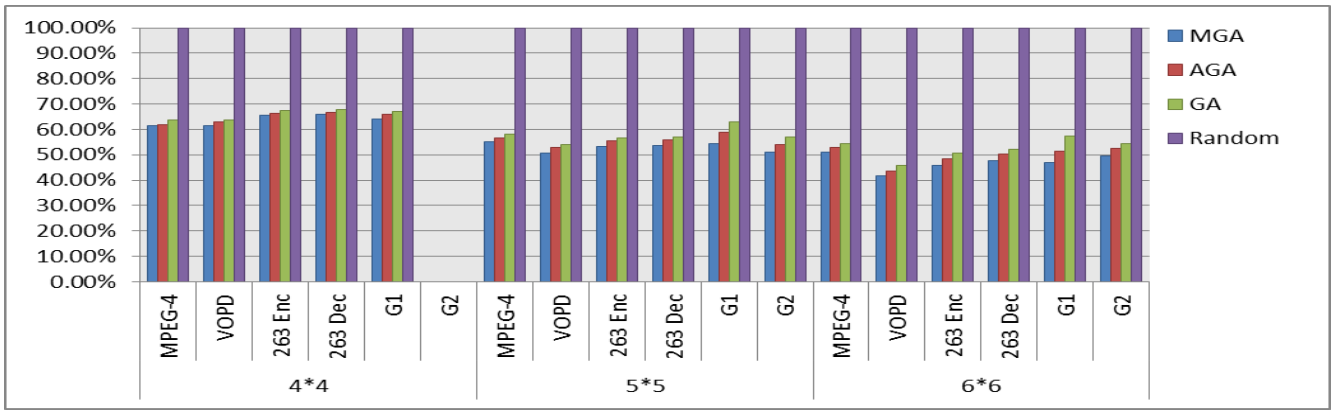
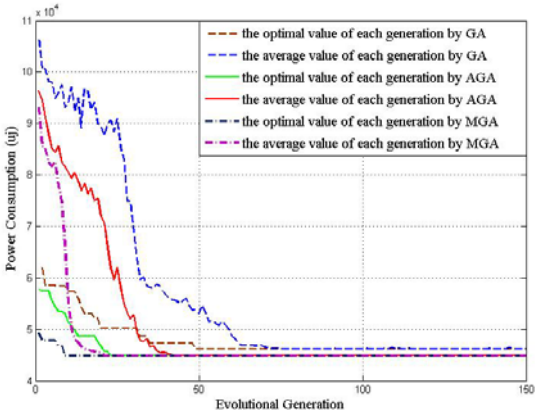


Fig 5 Energy consumption comparison

As is shown in Figure 5, energy consumption value calculated by MGA, AGA, GA is much smaller than energy consumption value from random mapping, which proves that the mapping algorithm can greatly reduce the required energy of communication on NoC. Energy consumption produced by MGA is less than that by AGA and GA, which proved MGA being more effective. The decreasing of energy consumption is more obvious as the Mesh scale increases. The number of task nodes in G2 is 17, more than the resource nodes of 4 * 4 Mesh. Due to the one to one mapping relationship, effective mapping results are not able to be produced.

In Table 4, compared to the AGA, GA and random mapping, energy consumption of MGA under 4 * 4 NoC Mesh platform decreases by 1.27%, 1.98% and 35.39%. When the Mesh platform scale increases to 5 * 5 and 6 * 6, the value decrease by 2.96%, 6.09%, 46.00% and 4.09%, 7.24%, 50.58%. Thus, the mapping results of MGA are generally better than that of AGA and GA, and the generated energy consumption is far below that of random mapping. With the expanding of Mesh scale, MGA performance is more evident. MGA starts mapping from optimum initial solution, placing task nodes with large communication volume on the symmetrical Mesh resource nodes, greatly reducing the probability of task nodes with large communication volume being on unfavorable resource nodes, thus reduces the hop count of task graph, and finally reduces the energy consumption of the task graph.



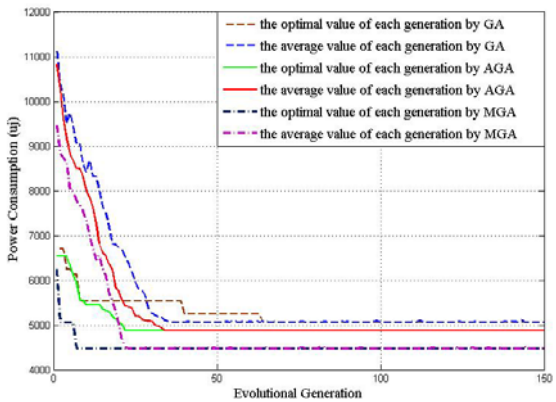


Fig 6 MPEG-4 ValueFig 7 G1 Value

Take the MPEG-4, G1 application diagram as an example, as is shown in Fig 6 and Fig 7， when the evolution algebra is 0, the average energy consumption of MGA mapping result is less than that of AGA and GA, which demonstrates the feasibility of the foregoing optimum initial solution. Also it can be clearly seen that when the evolution algebra is the same, the average energy consumption of GA and AGA are higher than the results of the MGA mapping, and the convergence of MGA is faster than that of AGA and GA. When the best fitness and the averages fitness of each generation are consistent, and the increase of the evolution algebra will not have significant fluctuations in them, the algorithm has reached an optimum stable value. In Fig 6, MGA evolves about 20 generations to reach a stable value, AGA, GA iterate about 40 generations and 70 generations to

reach a stable value; in Fig 7, MGA evolves about 22 generations to reach a stable value, AGA, GA evolves about 35 generations and 65 generations for that. The result shows that MGA optimizes faster. The averages fitness of AGA and GA prone to fluctuate during the process of convergence, and the final result are higher than MGA results. At this moment, GA was trapped by premature, optimizing calculation stagnates.

The mapping results of MPEG-4 and G1 is recorded under different mapping algorithms, and then get the running time by testing the mapping results on the simulation platform—NoCmap[14]. As is shown in Fig 8, it is obviously to see that the MGA has better performance than AGA,GA.

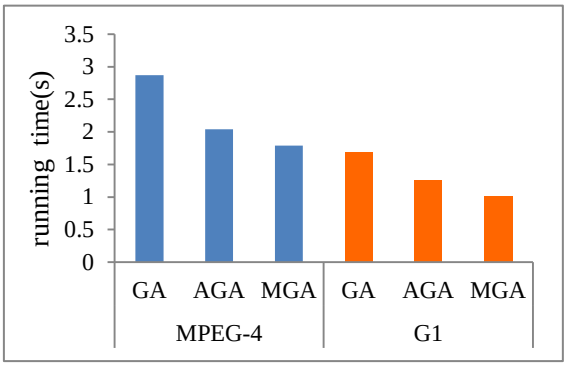


Fig8 The running time of TG under different mapping algorithms

In each iteration, MGA uses certain probability to make the initial solution be involved in genetic manipulation, ensures the diversity of the population, and avoids the advantages of individual breeding too fast to find the optimal solution. This algorithm adopts evolutionary reversal, the best neighbors and other strategies to adjust the evolutionary direction,speeding up the convergence and avoiding the excessive volatility in the process of convergence.

6 Conclusion

In this paper, MGA is used to solve the NoC mapping problems. This algorithm uses the initial solution with a certain probability to avoid stagnation; and introduces turn roulette bet, the best neighbors, evolutionary reversal strategy into the genetic manipulation to speed up the convergence rate. According to the experimental results, MGA mapping result is obviously better than AGA and GA. For the next step in the research process, the time complexity of the algorithm should be further considered for building a better objective function.

Acknowledgment

This paper (Title:Modified Genetic Algorithm Based Method on Low-Power Mappingin Network-on-chip; Authors:DaiQihua(278591289@qq.com), LiuQinrang, ShenJianliang, Sun Miao) supported in part by National 973 Program（No.2012CB315904）of China.

References

[1] Yang Pengfei, Wang Quan. Effective Task Scheduling and IP Mapping Algorithm for Heterogeneous NoC-Based MPSoC[J]. Mathematical Problems in Engineering, 2014.

[2] Benini L, De Micheli G. Networks on chips: a new SoCparadigm[J]. Computer, 2002, 35(1): 70-78.

[3] Wang Jian, Li Yu-bai, and Jiang Yong-nan. Communication performance analytical model and buffer allocation optimizing algorithm for Network-on-Chip[J]. Journal of Electronics & Information Technology, 2009, 31(5): 1059-1062.

[4] S.Tosun, O.Ozturk, M.Ozen. An ILP for mulation for application mapping onto network-on-chips, in:3rd International Conference on Application of Information and Communication Technologies, AICT2009, Baku, 2009.

[5] Ning Wu, Yifeng Mu, Fang Zhou, et al. GA-MMAS: an Energy-aware Mapping Algorithm for 2D Network-on-Chip. Proceeding of the World Congress on Engineering and Computer Science 2011 Vol Ⅱ. WCECS 2011, October 19-21, 2011.
[6] Wang F, Zhang Z. An Adaptive Genetic Algorithm for Mesh-Based NoC Application Mapping[J]. TELKOMNIKA Indonesian Journal of Electrical Engineering, 2014, 12(11): 7869-7875.
[7] Sahu P K, Shah T, Manna K, et al. Application mapping onto mesh-based network-on-chip using discrete particle swarm optimization[J]. Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 2014, 22(2): 300-312.
[8] Yang Sheng-guang, Li Li, Gao Ming-lun. An Energy and Delay-aware Mapping Method of NoC. Acta Electronica sinica, 2008,5(36):937-942.
[9] Jingcao Hu, RaduMarculescu. Energy-and Performance-Aware Mapping forRegularNoC Architectures. IEEE Transacitons on Computer-Aided Design of Integrated Circuits and Systems, VOL.24, NO.4, April 2005.
[10] Terry TaoYe, Giovanni De Micheli, Luca Benini, et al. Analysis of power consumption on switch fabrics in network routers, Proceedings of Design Automation Conference, 524-529, 2002.
[11] Fen Ge, Ning Wu, Xiaolin Qin, et al. Clustering-Based Topology Generation Approach for Application-Specific Network on chip. Proceeding of the World Congress on Engineering and Computer Science 2011 Vol Ⅱ. WCECS 2011, October 19-21, 2011.
[12] Pradip Kumar Sahu, Santanu Chattopadhyay. A Survey on Application Mapping Strategies For Network-on-Chip Design. Journal of Systems Architecture 59 (2013) 60–76.
[13] Dick R P, Phodes D L, Wolf W. TGFF: Task Graphs for Free[C]//Proceeding of the Sixth International Workshop on Hardware/Software Codesign. Seattle: IEEE, 1998;97-101.
[14] http://www.ece.cmu.edu/~sld/software/nocmap.php