Journal of Software Engineering

ISSN 1819-4311

Academic
Journals Inc.

www.academicjournals.com

OPEN ACCESS Journal of Software Engineering

ISSN 1819-4311
DOI: 10.3923/jse.2017.194.201

Research Article
Hybrid Quantum Genetic Algorithm Used for Low-power Mapping in Network-on-chip

Tianhua Liu, Shoulin Yin, Jie Liu and Lin Teng

Software College, Shenyang Normal University, No. 253, HuangHe Bei Street, HuangGu District, Shenyang, P.C. 110034, China

Abstract

Background: Traditional genetic algorithm has low robustness and easily falling into local solution with a long convergence time. **Materials and Methods:** In order to improve the work efficiency and decrease large scale applications power in Network-on-chip (NoC), it puts forward a new Network-on-chip low-power mapping method based on hybrid quantum genetic algorithm in this study. This new method adopts communication weight of task node and structure characteristics of mapping platform to make priority compartment for task node. According to the priority and its connections, it gets better initial mapping solutions set. In solving process, combining adaptive rotating angle adjustment strategy, quantum bit crossover mutation operation and group catastrophe idea, it proposes a new hybrid quantum genetic algorithm. It adds roulette, optimal neighborhood selection and evolution reversal operation into genetic algorithm. What's more, it chooses a initial solution with a certain probability in each iteration to prevent stagnation of algorithm. **Results:** Finally, the experiments show that hybrid quantum genetic algorithm greatly reduces the power consumption than traditional genetic algorithm and random mapping methods when used for Network-on-chip in the same task model and mapping platform. **Conclusion:** The hybrid quantum genetic algorithm is an effective method for Network-on-chip low-power mapping, which can effectively improve the accuracy of Network-on-chip.

Key words: Network-on-chip, low-power mapping, hybrid quantum genetic algorithm, roulette, optimal neighborhood selection, evolution reversal operation

Received: October 25, 2016 Accepted: December 10, 2016 Published: March 15, 2017

Citation: Tianhua Liu, Shoulin Yin, Jie Liu and Lin Teng, 2017. Hybrid quantum genetic algorithm used for low-power mapping in Network-on-chip. J. Software Eng., 11: 194-201.

Corresponding Author: Shoulin Yin, Software College, Shenyang Normal University, No. 253, HuangHe Bei Street, HuangGu District, Shenyang, P.C. 110034, China

Competing Interest: The authors have declared that no competing interest exists.

Data Availability: All relevant data are within the paper and its supporting information files.

INTRODUCTION

Traditional system on chip[1] (SoC) technology (e.g., bus structure, the effective address space) seriously restricts the development of the chip. Therefore, researchers put forward a new integrated circuit structure Network-on-chip (NoC)[2] in 2005. From SoC to NoC, energy consumption has gradually become the primary limiting factor for chip design, the key point of integrated circuit design has also been transformed from the functional requirements of the chip to power requirements.

Quantum Genetic Algorithm (QGA) was a kind of probability optimization method based on the principle of quantum computing, which was proposed by Han and Kim[3]. The QGA has obtained the good effect for solving combinatorial optimization problems. The QGA introduces quantum state vector into the genetic encoding, it makes full use of the quantum state superposition, parallelism and quantum entanglement character. The QGA is superior to general genetic algorithm in terms of convergence speed, global and stability. In recent years, QGA has been widely used in power system reactive power optimization[4,5], economic operation[6,7], the grid planning, distribution network reconfiguration, etc[8-10].

In this study, it obtains the optimal solution based on hybrid quantum genetic algorithm for Network-on-chip low-power mapping. Traditional genetic algorithm has strong global search ability but its local search ability is poor. It easily falls into premature phenomenon and takes a long time to achieve the optimal solution. Therefore, it proposes a hybrid quantum genetic algorithm[11,12] to offset the above disadvantages. It uses communication weight of task node to divide priorities and first maps task node with big communication traffic. Genetic algorithm will start with a better initial solution. It combines adaptive rotating angle adjustment strategy, quantum bit crossover mutation operation and group catastrophe idea to improve genetic algorithm. In each iteration calculation, it uses initial solution with a certain probability to guarantee the population diversity and avoid the optimization stagnate. At the same time, it adopts roulette, optimal neighborhood selection and evolution reversal operation to adjust the evolution direction and improve convergence speed of algorithm.

MATERIALS AND METHODS

Network-on-chip mapping: Net-on-chip indicates that NoC allocates task nodes or logic IP cores into appropriate resources nodes in Architecture Characteristic Graph[13] (ACG) under the condition of given task, topology structure and bandwidth, which makes various applications efficiently and smoothly completed and consumes the least energy. Figure 1 shows that, Task Graph[14] (TG) allocates task nodes into resource nodes after mapping algorithm processing. So, NoC mapping can be abstracted as the mapping between TG and NoC structure characteristics graph. The NoC mapping question belongs to NP problem. When the size of problem is larger, it is almost impossible to get accurate solution and time consuming is longer, feasible degree is very low. Therefore, it needs to adopt intelligence algorithm to find the best mapping result with its fast searching ability.

In order to make the study more representative, it adopts NoC 2D mesh topological structure. The 2D mesh structure rule has better symmetry and extensibility, supports multiple routing protocols, flow control technology, exchanging algorithm and its two-dimensional structure is easily mapped into the silicon surface, which is easy to implement. Currently, it is one of the used widely topology structures[15].

Problems definition

Definition 1: Task graph TG(t,v) is an acyclic directed graph. In Fig. 1, vertex $t_j \in T$ denotes a task node. Directed arc v_{t_i,t_j} denotes the communication relationship between task node t_i and t_j as $d(v_{t_i,t_j})$. The D_{t_i} denotes the number of

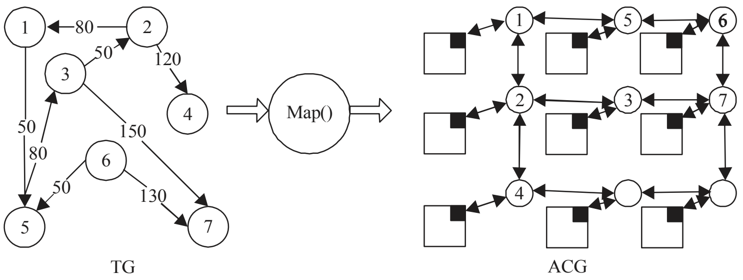

Fig. 1: Mapping flow chart

connecting with other task nodes except itself. The value in the arc between task node t_i and t_j is the communication traffic, written as W_{t_i,t_j}:

$$d(v_{t_i,t_j}) = \begin{cases} 1, \text{ i and j are interconnection} \\ 0, \text{ i and j are disconnection} \end{cases} \tag{1}$$

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

Definition 2: Architecture characteristic graph ACG(p, h) is a directed graph. In Fig. 1, vertex $p_i \in P$ denotes a resource node in 2D Mesh. Directed arc $h_{i,j}$ denotes the physical link between p_i and p_j. Its value is the Manhattan distance.

The task nodes and resource nodes are one-to-one relationship, in order to ensure the proper execution of the mapping results, resource nodes number of ACG must be greater than or equal to the task node in the TG. Namely:

$$\text{Size(TG)} \le \text{Size(ACG)} \tag{3}$$

Mapping function map() allocates task nodes in TG(t, v) into resource nodes of ACG(p, h):

$$\forall t_i \in T, \text{ map}(t_i) \rightarrow p_m \in P \tag{4}$$

$$\forall t_j \in T, \text{ map}(t_j) \rightarrow p_n \in P \tag{5}$$

$$t_i \neq t_j \tag{6}$$

$$P_m \neq p_n \tag{7}$$

The Eq. 4-7 shows that each task is completed by one resource node. Each resource node executes one task. After mapping, jump number will be determined among each task node.

Power model: Unit bits data transmission between adjacent node resources consumes energy E_{bit}:

$$E_{bit} = E_{S_{bit}} + E_{B_{bit}} + E_{W_{bit}} + E_{L_{bit}} \tag{8}$$

where, $E_{S_{bit}}$, $E_{B_{bit}}$ and $E_{W_{bit}}$ are consumed energy in exchange structure, buffering and internal wiring respectively. The $E_{L_{bit}}$ is consumed energy in physical link. Under the existing technological level, size of resource nodes in the NoC has reached nanometer level, energy consumption in exchange structure, buffering and internal wiring is far less than that in the physical link.

$$E_{bit} = E_{W_{bit}} \prec E_{L_{bit}} \tag{9}$$

So Eq. 8 can be written as:

$$E_{bit} = E_{S_{bit}} + E_{L_{bit}} \tag{10}$$

Therefore, energy consumption of unit bits data transforming from t_i to t_j is:

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

where, h_{t_i,t_j} is switches from resource node t_i to t_j denoted by Manhattan distance. Assuming that NoC 2D Mesh is in a x-y coordinate axis, so $h_{t_i,t_j} = |X_{t_i} - X_{t_j}| + |Y_{t_i} - Y_{t_j}|$. Size of NoC 2D mesh is N·N. Total energy consumption of information interaction in NoC is:

$$E_{NoC} = \sum_{i=1}^{N} \sum_{j=1}^{N} W_{t_i,t_j} \bullet E_{bit}^{t_i,t_j} \tag{12}$$

where, W_{t_i,t_j} is communication traffic from resource node t_i to t_j. It can see that when the power consumption model is simplified, the direct optimization way for reducing the energy loss is to reduce the sum of the Manhattan distance. More specifically, it reduces the the Manhattan distance with bigger communication weight between mapping resource nodes. After computing and comparing the power consumption under each mapping result, it can find the minimum energy consumption. As Eq. 13:

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

Low-power mapping based on hybrid quantum genetic algorithm: In this study, it proposes a hybrid quantum genetic algorithm (HQGA) based on quantum genetic algorithm and the general genetic algorithm theory, its basic idea is that it uses adaptive rotation angle adjustment strategy to adaptively adjust the algorithm search speed; through quantum crossover and mutation operators, it realizes the combination between common genetic algorithm and quantum genetic algorithm which is beneficial to keep relatively good gene fragments and improve the local search ability, it introduces catastrophe strategy, which is conducive to jumping out of local optimal algorithm and improving the global search ability. Then it codes the question parameters into chromosomes and reuses iterative manner to make crossover and mutation operation to exchange the coded information in the chromosome, which is gradually close to the best chromosomes, finally it produces chromosomes meeting the optimization goal. Genetic algorithm[16-18] has a big search range, can approximate optimal solution area with a faster speed. It is an ideal heuristic algorithm to solve the NoC mapping. Genetic algorithm has strong global search ability and poor local search ability, it uses hybrid quantum genetic algorithm to improve the disadvantages of traditional genetic algorithm.

Process of hybrid quantum genetic algorithm:

- **Setting parameters of algorithm:** Parameters are as in Table 1. Parameters setting has a great influence on the genetic algorithm, for example the mutation probability is bigger that can lead to the mean and optimal value of each generation fitting not well. And it cannot achieve the stable value
- Initializing code, generating task node communication matrix. It obtains task node number M in task graph makes decimal coding for each task node. It gives no-repeat 1-M values for task node, extracts communication traffic and generates M×M communication matrix
- **Constructing better initial solution:** In ACG(p,h) with the rule of NoC mesh structure, initial mapping of genetic algorithm is random allocation. The M task nodes have $N^2!(N^2-M)!$ mapping results. When the task node number is larger, it is not possible to put all random map as the initial population generation into the calculation. Population size is too big, so evolution efficiency is very low. In order to reduce the number of iterations and achieve the optimum solution area, it builds a better initial solution set and reduces the time-consuming of generating optimal solution. The following is the process to calculate better initial solution
  - Calculating each task node communication traffic $\mathrm{Com}_{t_i} = \sum_{j=1}^{M} W_{t_i,t_j}$. The high communication traffic is, the high task node priority is
  - It selects the task node with highest priority as first node. Assuming that multiple task nodes communication traffic is consistent, it takes the larger average traffic task node as the first node as in Eq. 14. The A_i is bigger, then the priority is high. If more than two communication traffics are same, then the A_i is unchanged. Table 2 is the each node priority order according to the rule:

$$Ai = \sum_{j=1}^{M} W_{t_i,t_j} / \sum_{j=1}^{M} d(v_{t_i,t_j}) \qquad (14)$$

  - The selection of initial mapping resource node has a great influence on following nodes mapping. So, it first puts node with high priority on the symmetry resource node of NoC 2D mesh to reduce the jump number of highest priority task node. Because NoC 2D mesh is symmetrical structure with several symmetric resource nodes. Transition number is 1, its neighborhood node are 2, 3 and 4. Because it is one-to-one correspondence relation, connecting task node of the highest priority task node is less than or equal to 4, it chooses the neighborhood resource nodes equal to connected task nodes as the mapping nodes. If connected task node number is greater than 4, then it selects the neighborhood resources node which has a big difference from the connected task nodes, it can save the node and link resources. When there are multiple optimal resource nodes, it randomly selects a node as a mapping point. Second it places the highest priority task node connecting mapping task nodes on the connected resource nodes, thereby it can reduce transition number between big communication traffic task node as Fig. 2. First it places 2-3 nodes with big communication weight, then the rest of task nodes randomly map to construct better initial solution
- Adaptive rotation angle adjustment strategy. Quantum revolving door adjustment operation is:

Table 1: Definition of parameters

Parameters	Definition	Value
Maxgen	Population evolution number	150
Sizepop	Population size	100
Pcross	Crossover probability	0 .7
Pmutation	Mutation probability	0.01
Meshsize	Mesh size in NoC	N
Ini	Initial solution using probability	0.05

Table 2: Priority computing

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

$$\begin{bmatrix}\alpha_i^{'}\\ \beta_i^{'}\end{bmatrix}=\begin{bmatrix}\cos\theta_i & -\sin\theta_i\\ \sin\theta_i & \cos\theta_i\end{bmatrix}\begin{bmatrix}\alpha_i\\ \beta_i\end{bmatrix} \tag{15}$$

$$\theta_i = k\cdot s(\alpha_i,\beta_i) \tag{16}$$

where, $(\alpha_i,\ \beta_i)^T$ and $(\alpha_i^{'},\beta_i^{'})^T$ are the probability amplitude of i-th quantum bit revolving door before and after updating in chromosome. θ_i is revolving angle. The k is coefficient related to convergence speed. The k is bigger, algorithm will be precocity. Otherwise, speed of algorithm is very slow. Function $s(\alpha_i,\ \beta_i)$ will leads algorithm search towards to optimal solution. Its value is as Table 3. The +1 denotes that rotation angle moves along counterclockwise direction. The -1 denotes that rotation angle moves along clockwise direction

- **Genetic operation:** Based on the better initial solution, it gradually produces new chromosome through selection, crossover and mutation operation to construct new population group and reach at the optimal solution

Table 3: Value of function $s(\alpha_i,\ \beta_i)$

			$s(\alpha_i,\ \beta_i)$			
χ_i	$Best_i$	$f(\chi)<f(best)$	$\alpha_i\ \beta_i>0$	$\alpha_i\ \beta_i<0$	$\alpha_i = 0$	$\beta_i = 0$
0	0	True	0	0	0	0
0	0	False	0	0	0	0
1	1	True	0	0	0	0
1	1	False	0	0	0	0
0	1	True	-1	+1	±1	0
0	1	False	+1	-1	0	±1
1	0	True	+1	-1	0	±1
1	0	False	-1	+1	±1	0

◦ It selects communication power consumption of mapping results as fitness. Communication power consumption is low, the greater fitness is. It uses roulette, the optimal neighbor strategy to make chromosome selection. It divides wheel into M parts, which ensures that the probability of each chromosome selected is same. Then it chooses the chromosome with big fitness to make up a new group and execute crossover and mutation operation. In that some individuals breed next generation faster, which destroys the population diversity and makes the genetic operations fall into premature convergence trap. So, initial solution chromosome randomly replace that in a selected group with a certain probability of random in each iteration to ensure the population diversity and avoid premature convergence phenomenon

◦ **Quantum crossover operation:** In quantum genetic algorithm, crossover operation object is a quantum bit rather than gene. It introduces crossover operation to ensure group integrity and greatly improve local search ability of algorithm. Quantum crossover method in this study uses single point crossover operation, it randomly selects two chromosomes and generates a crossover location and then it keeps the individual pieces before the crossover position, only exchanges quantum bits after crossover position. Crossover process is as shown in Fig. 3

◦ **Quantum mutation operation:** Quantum mutation operation is to prevent algorithm premature and improve the local searching ability. It selects a chromosome with a certain probability, then it determines the mutation

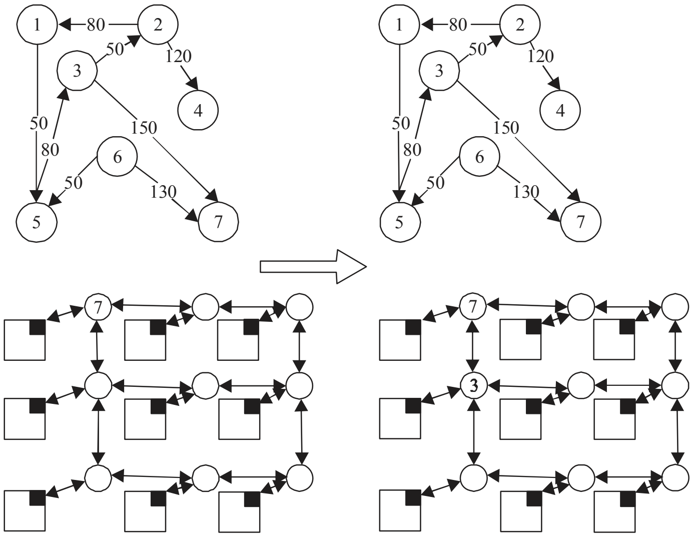

Fig. 2: Node priority mapping

$$\begin{gathered}\left\langle \begin{array}{c|cc} \alpha_{11} & \alpha_{12} & \alpha_{13} \\ \beta_{11} & \beta_{12} & \beta_{13} \end{array} \right\rangle \left\langle \begin{array}{c|cc} \alpha_{21} & \alpha_{22} & \alpha_{23} \\ \beta_{21} & \beta_{22} & \beta_{23} \end{array} \right\rangle \\ \Downarrow \\ \left\langle \begin{array}{c|cc} \alpha_{11} & \alpha_{22} & \alpha_{23} \\ \beta_{11} & \beta_{22} & \beta_{23} \end{array} \right\rangle \left\langle \begin{array}{c|cc} \alpha_{21} & \alpha_{12} & \alpha_{13} \\ \beta_{21} & \beta_{12} & \beta_{13} \end{array} \right\rangle \end{gathered}$$

Fig. 3: Crossover operation process

$$\left\langle \begin{array}{c|cc} \alpha_{11} & \alpha_{12} & \alpha_{13} \\ \beta_{11} & \beta_{12} & \beta_{13} \end{array} \right\rangle \Rightarrow \left\langle \begin{array}{c|cc} \alpha_{11} & \beta_{12} & \alpha_{13} \\ \beta_{11} & \alpha_{12} & \beta_{13} \end{array} \right\rangle$$

Fig. 4: Mutation operation process

Table 4: Characteristic pattern attribute

Application graph	Task node	Communication side	Total communication
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

quantum bits in chromosome. Finally, it executes CNOT operation for the selected quantum bits as Fig. 4

- **Quantum catastrophe idea:** It indicates that it puts a great disturbance on population in evolution process, which makes it jump out of local optimal and begins to research. It can effectively prevent the algorithm falling into local optimum. If the best value of individual after successive generations does not change, it suggests that algorithm may fall into local optimum. Then it should adopt quantum catastrophe strategy

RESULTS AND DISCUSSION

In order to verify the effectiveness of our hybrid quantum genetic algorithm (HQGA) used for low-power mapping in Network-on-chip, it makes some experiments. To reflect that HQGA is effective to all kinds of applications, this study uses the actual task communication diagram (including MPEG-4, VOPD, 263Enc and 263Dec) and random task communication diagram (G1, G2 generated by TGFF software) for testing as Table 4. The task node number, communication side and total communication are different in this six application graphs, which can fully reflect the merits and demerits of mapping algorithm. In order to guarantee the accuracy of experimental results and improve the representation of the experimental data, it makes 20 calculation with different mapping algorithms under the different mesh size, record every power consumption value. Finally, it takes average results of the 20 times calculations. Mesh size is 4×4, 5×5, 6×6.

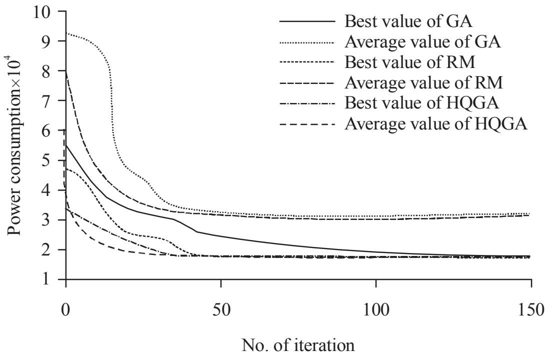

Fig. 5: MPEG-4 function value

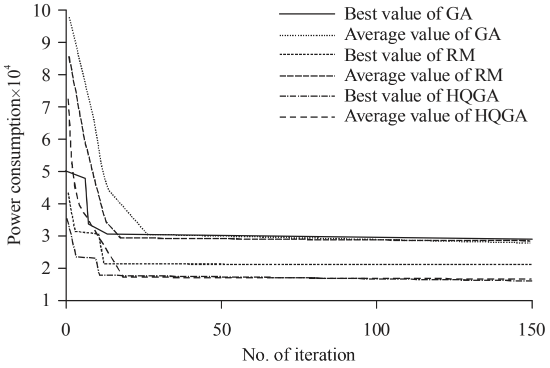

Fig. 6: G1 function value

Table 5 compares the power consumption with HQGA, Genetic Algorithm (GA), Random Mapping (RM). From Table 5, it can know that power consumption with HQGA decreases by 1.27, 1.98 and 35.39%, respectively in 4×4 NoC mesh condition compared to GA and RM. When NoC mesh is 5×5 and 6×6, it falls by 2.96, 6.09 and 46.00% and 4.09, 7.24 and 50.58%, respectively. So, mapping result of HQGA is generally superior to GA and RM and far less than power consumption of random mapping. With the increase of mesh size, performance of HQGA is more obvious. The HQGA places the task node with large communication traffic in the mesh symmetric resource nodes from the optimal initial solution mapping, greatly reduces the probability of task node in adverse resource nodes, thus it reduces the communication power consumption of task graph.

In this study, it selects MPEG-4 and G1 to illustrate the effectiveness as Fig. 5 and 6. When the iteration of evolution is 0, average power consumption value of HQGA mapping results is less than GA and RM, which demonstrates the feasibility of above optimal initial solution. In addition, average power consumption of GA and RM is higher than HQGA in the same evolution iteration, which shows that

Table 5: Power consumption decrease proportion

		Power consumption decrease proportion (%)								
		4×4			5×5			6×6		
Application	Task node	HQGA	GA	RM	HQGA	GA	RM	HQGA	GA	RM
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	0.72	1.02	33.26	1.78	3.88	45.58	3.12	5.63	53.08
263 Dec	14	1.57	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	14.70	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

convergence rate of HQGA is faster than GA and RM. When the best value and average value of each generation are consistent and increase with the evolution iteration, there will be no obvious fluctuation, the algorithm optimization has reached optimal stable value. In Fig. 5, HQGA reaches a steady value at 20 generations, however, GA and RM are at 40 and 70, respectively. It is similar to Fig. 6. The HQGA has the minimum power consumption and the fewest evolution iteration, followed by GA, then is RM in Fig. 6. Average power consumption of GA and RM easily occur fluctuation. Its optimization direction is different easily falling into local solution. Initial solution in HQGA can execute genetic operation with a certain probability to ensure the diversity of the population. At the same time, HQGA uses reverse evolution, the optimal neighbor strategy to adjust evolutionary direction, accelerate convergence time, avoid excessive volatility in the process of algorithm.

At present, how does it effectively map task node into each resource node in NoC structure feature diagram under system constraints to reduce total communication power consumption, which is becoming a research hot issue in the field of NoC technology. Tosun *et al*.[19] adopted integer linear programming strategy based on clustering to reduce integer linear programming space and decrease search time and then could obtain the best mapping. However, the clustering standard was uncertain. Wu *et al*.[20] combined genetic and ant colony algorithm. The quick search ability of genetic algorithm was used to obtain the optimal initial solution and assign pheromone concentration distribution for ant colony path according to optimal order of optimization solutions, which could make full use of the positive feedback characteristics of ant colony algorithm to search the better solution of mapping. Zang and You[21] adopted adaptive genetic algorithm and used the change of best fitness and the average fitness to adjust probability of crossover and mutation in each iteration, thus it accelerated the best mapping acquirement. Sahu *et al*.[22] presented a discrete Particle Swarm Optimization (PSO) based strategy to map applications on both 2-D and 3-D mesh-connected Networks-on-chip. The basic PSO formulation had been augmented by (1) Running multiple PSOs and (2) Deterministically generating a part of the initial population for PSO. In general, the algorithms have reduced the NoC communication power consumption but the convergence speed and the optimization effect still can be optimized. Hence, this study puts forward a new Network-on-chip low-power mapping method based on hybrid quantum genetic algorithm. Finally, it conducts comparison experiments with GA and RM from best value and average value, the simulation results and the theoretical analysis show that this new method can effectively improve the accuracy of Network-on-chip mapping.

CONCLUSION

In this study, it proposes a hybrid quantum genetic algorithm to solve low-power mapping problem in Network-on-chip. The new method adopts communication weight of task node and structure characteristics of mapping platform to make priority compartment for task node. And according to the priority and its connections, it gets better initial mapping solutions set. Meanwhile, it combines adaptive rotating angle adjustment strategy, quantum bit crossover mutation operation and group catastrophe idea. It also adds roulette, optimal neighborhood selection and evolution reversal operation into genetic algorithm. What's more, it chooses a initial solution with a certain probability in each iteration to prevent stagnation of algorithm. The experimental results show that the HQGA is obviously better than GA and RM. In the future, it need further to consider time complexity of algorithm and build up a more representative objective function.

SIGNIFICANCE STATEMENTS

- This study puts forward a new Network-on-chip low-power mapping method based on hybrid quantum genetic algorithm
- This new method adopts communication weight of task node and structure characteristics of mapping platform to make priority compartment for task node

- According to the priority and its connections, it gets better initial mapping solutions set. In solving process, combining adaptive rotating angle adjustment strategy, quantum bit crossover mutation operation and group catastrophe idea, it proposes a new hybrid quantum genetic algorithm
- It adds roulette, optimal neighborhood selection and evolution reversal operation into genetic algorithm. What's more, it chooses a initial solution with a certain probability in each iteration to prevent stagnation of algorithm

REFERENCES

1. El Shobaki, M. and L. Lindh, 2015. A hardware and software monitor for high-level system-on-chip verification. Proceedings of the 2001 International Symposium on Quality Electronic Design, March 28-28, 2001, San Jose, CA., pp: 56-56.
2. Pande, P.P., C. Grecu, M. Jones, A. Ivanov and R. Saleh, 2005. Performance evaluation and design trade-offs for network-on-chip interconnect architectures. IEEE Trans. Comput., 54: 1025-1040.
3. Han, K.H. and J.H. Kim, 2000. Genetic quantum algorithm and its application to combinatorial optimization problem. Proceedings of the 2000 Congress on Evolutionary Computation, July 16-19, 2000, California, pp: 1354-1360.
4. Chen, Y.F., H.Q. Wang, J.S. He and X.Y. Bi, 2007. Reactive power optimization based on improved genetic algorithm. J. Northwest Hydroelectr. Power, 23: 536-537.
5. Eremia, M. and A.B.L. Tawfeeq, 2014. Reactive power optimization based on genetic algorithm with new technique of mutation. Proceedings of the 2014 International Symposium on Fundamentals of Electrical Engineering (ISFEE), November 28-29, 2014, Bucharest, Romania.
6. Deng, G., M. Wei, Q. Su and M. Zhao, 2015. An effective co-evolutionary quantum genetic algorithm for the no-wait flow shop scheduling problem. Adv. Mech. Eng., Vol. 7. 10.1177/1687814015622900.
7. Gu, J., M. Gu and X. Gu, 2015. A mutualism quantum genetic algorithm to optimize the flow shop scheduling with pickup and delivery considerations. Math. Problems Eng. 10.1155/2015/387082.
8. Chidanandappa, R., T. Ananthapadmanabha and H.C. Ranjith, 2015. Genetic algorithm based network reconfiguration in distribution systems with multiple DGs for time varying loads. Proc. Technol., 21: 460-467.
9. Zhang, G., H. Rong, J. Cheng and Y. Qin, 2014. A population-membrane-system-inspired evolutionary algorithm for distribution network reconfiguration. Chin. J. Electron., 23: 437-441.
10. Shamsudin, N.H., N.F. Omar, A.R. Abdullah, M.F. Sulaima, N.A. Abidullah and H.I. Jaafar, 2014. An improved genetic algorithm for power losses minimization using distribution network reconfiguration based on re-rank approach. Res. J. Applied Sci. Eng. Technol., 8: 1029-1035.
11. Yin, S., J. Liu and L. Teng, 2016. An Improved Artificial Bee Colony Algorithm for Staged Search. TELKOMNIKA Telecomm. Comput. Electron. Control, 14: 1099-1104.
12. Zhang, Q., Z. Chen and L.T. Yang, 2015. A nodes scheduling model based on Markov chain prediction for big streaming data analysis. Int. J. Commun. Syst., 28: 1610-1619.
13. Sun, J., N. Guan, Y. Wang, Q. Deng, P. Zeng and W. Yi, 2016. Feasibility of fork-join real-time task graph models: Hardness and algorithms. ACM Trans. Embedded Comput. Syst., Vol. 15. 10.1145/2809780.
14. Hu, J. and R. Marculescu, 2005. Energy-and performance-aware mapping for regular NoC architectures. IEEE Trans. Comput. Aided Design Integrated Circ. Syst., 24: 551-562.
15. Yao, K., 2014. Microvascular Architecture Characteristic of Early Gastric Cancer as Visualized by Magnifying Endoscopy (ME) and Clinical Applications. In: Zoom Gastroscopy: Magnifying Endoscopy in the Stomach, Yao, K. (Ed.). Chapter 6, Springer, Japan, ISBN: 978-4-431-54206-3, pp: 27-48.
16. Meng, L., S. Yin and X. Hu, 2016. An improved Mamdani Fuzzy Neural Networks Based on PSO Algorithm and New Parameter Optimization. Indonesian J. Electr. Eng. Comput. Sci., 1: 201-206.
17. Doerr, B., C. Doerr and F. Ebel, 2015. From black-box complexity to designing new genetic algorithms. Theor. Comput. Sci., 567: 87-104.
18. Oliveto, P.S. and C. Witt, 2015. Improved time complexity analysis of the simple genetic algorithm. Theor. Comput. Sci., 605: 21-41.
19. Tosun, S., O. Ozturk and M. Ozen, 2009. An ILP formulation for application mapping onto network-on-chips. Proceedings of the International Conference on Application of Information and Communication Technologies, October 14-16, 2009, Baku, pp: 1-5.
20. Wu, N., Y. Mu and F. Ge, 2012. GA-MMAS: An energy-and latency-aware mapping algorithm for 2D network-on-chip. IAENG Int. J. Comput. Sci., Vol. 39.
21. Zang, M. and H. You, 2012. Low power NOC process element mapping using genetic algorithm. J. Inform. Comput. Sci., 9: 557-563.
22. Sahu, P.K., T. Shah, K. Manna and S. Chattopadhyay, 2014. Application mapping onto mesh-based network-on-chip using discrete particle swarm optimization. IEEE Trans. Very Large Scale Integration (VLSI) Syst., 22: 300-312.