

PAPER • OPEN ACCESS

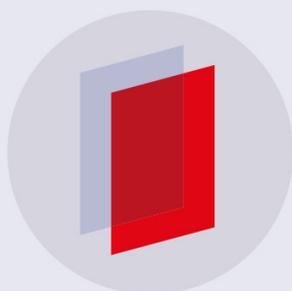
DI_GA: A Heuristic Mapping Algorithm for Heterogeneous Network-on-Chip

To cite this article: Juan Fang *et al* 2019 *IOP Conf. Ser.: Mater. Sci. Eng.* **490** 042021

View the [article online](#) for updates and enhancements.

Recent citations

- [Intelligent Mapping Method for Power Consumption and Delay Optimization Based on Heterogeneous NoC Platform](#)
Juan Fang *et al*



IOP | ebooks™

Bringing you innovative digital publishing with leading voices to create your essential collection of books in STEM research.

Start exploring the collection - download the first chapter of every title for free.

DI_GA: A Heuristic Mapping Algorithm for Heterogeneous Network-on-Chip

Juan Fang^{1,2,*}, Huan Zong¹ and Haoyan Zhao¹

¹Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China

²Beijing Institute of Smart City, Beijing University of Technology, Beijing 100124, China

*Corresponding author e-mail: fangjuan@bjut.edu.cn

Abstract. Heterogeneous multi-core processors have become the forefront of processor development due to their advantages in system throughput and execution efficiency, but they also bring many new challenges to system design. The mapping of heterogeneous Network-on-Chip (NoC) is challenging. To solve this problem, this paper proposes a heterogeneous multi-core processor task mapping algorithm based on an improved genetic algorithm. By constructing a good initial population method to improve the initial population quality, a dual population genetic mechanism is used in the iteration process. The algorithm can make tasks more reasonably distributed to various network nodes, and has high efficiency for optimizing network power consumption on heterogeneous multi-cores.

1. Introduction

With the continuous development of information technology, applications are showing more and more complex trends. The traditional System on Chip (SoC) communication generally adopts the bus structure. The communication efficiency of the bus structure is low, and the communication task between multiple processors cannot be handled well. Network on Chip (NoC) applies the traditional network communication to the internal communication design of the chip, and solves the growing communication demand between the cores on the multi core chip, and is a new scheme [1, 2] to deal with the many problems caused by the dimensions and communication requirements in the design of the ultra large scale integrated circuits.

Mesh-based IP core mapping problem for NoC is NP-hard [3], and the result will directly affect the chip's power consumption and other attributes. Therefore, how to establish an efficient algorithm to effectively obtain a low power mapping solution is a difficult problem for NoC designers [4].

Current research work has proposed some NoC mapping algorithms with power consumption or delay as the optimization goal. Khalili F et al. [5] proposed a fault-tolerant multi-application mapping technique, which maps the application core map to the free and error-free kernel by using a heuristic algorithm, and then adjusts the placement of other free cores on other free and error-free cores. Sahu P K et al. [6] proposed a discrete particle swarm optimization algorithm mapping algorithm that can be applied to two-dimensional Mesh and three-dimensional Mesh IP core mapping to obtain lower power consumption and more reliable performance. Tosun et al. [7] proposed their own ILP formula for mapping applications to a grid-based NoC in order to reduce energy consumption of different



benchmarks. Macron et al. [8] compared some NoC mapping algorithms used to obtain low power consumption communication weighted models. The NoC mapping results obtained by these algorithms have good performance, but all of these papers discuss the mapping of the task to the homogeneous processing element, and do not pay attention to the mapping of the heterogeneous processing elements. First of all, this paper gives two steps for the application mapping of heterogeneous multi-core NoC platform, and then gives the mathematical model of NoC processing element power consumption and communication power consumption, and finally uses algorithm proposed in this paper as a search tool to effectively obtain the best mapping solution or the best approximation solution.

Two Steps of NoC Mapping:

The NoC mapping problem in this paper is divided into two steps: the first step is shown in Figure 1 (a). The tasks in the application graph are assigned to the corresponding IP cores in a certain strategy. Multiple frequent interaction tasks in complex applications can be assigned to the same IP core in order to reduce communication costs and achieve faster resource sharing goals. The second step is shown in Figure 1 (b). The IP cores that have been bound to the previous task are assigned to the NoC resource nodes by some mapping algorithm, and the various applications are successfully and efficiently completed under the constraint conditions, and consume the least energy.

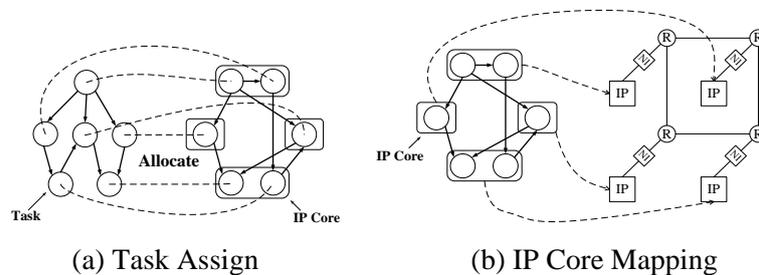


Figure 1. Two Steps of NoC Mapping.

2. Energy Model

Our research aims to reduce overall communication costs and system power consumption. In this section, the energy model used is defined.

Energy consumed in the operation of the entire NoC system consists of energy E^{core} consumed by the IP core in the task processing and energy E^{net} consumed in the data transmission network, that is:

$$Energy = E^{core} + E^{net} \quad (1)$$

For the same amount of data processing, the energy consumption on the core is only related to the type of the core (the conditions of core voltage, frequency, etc. are determined). The processing energy consumption E_{Δ}^{ip} of the Δ bit data at a particular core is represented as:

$$E_{\Delta}^{ip} = \rho_{ce} \times \Delta \quad (2)$$

In the formula, ρ_{ce} is a constant related to the process element type.

So, the total energy E^{core} consumed on the IP core is:

$$E^{core} = \sum_{ip} E_{\Delta}^{ip} \quad (3)$$

In this paper, the communication power consumption model proposed by [9] is used. The energy consumed by the single bit data from the resource node T_i to T_j is:

$$E_{bit}^{i,j} = (h_{i,j} + 1) \times E_{Sbit} + h_{i,j} \times E_{Lbit} \quad (4)$$

Where E_{Sbit} is the energy consumed by the unit bit data in the switch fabric: E_{Lbit} is the energy consumed by the unit bit data transmitted on the physical link of the adjacent resource node, $h_{i,j}$ is the number of transitions between the resource nodes T_i to T_j , Usually expressed in Manhattan distance.

The communication power consumed by NoC is:

$$E^{net} = \sum_{i=1}^N \sum_{j=1}^N V_{i,j} \times E_{bit}^{i,j} \tag{5}$$

So the total power consumption of the NoC system can be expressed as:

$$Energy = E^{core} + E^{net} = \sum_{ip} E_{\Delta}^{ip} + \sum_{i=1}^N \sum_{j=1}^N V_{i,j} \times E_{bit}^{i,j} \tag{6}$$

3. Improved Genetic Algorithm

Since the mapping algorithm is an NP-hard problem, recent studies have found that swarm intelligence algorithms such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and so on. The quadratic assignment problem (QAP), using the swarm intelligence algorithm can calculate a solution closer to the optimal solution. As a relatively common genetic algorithm, it mainly simulates the evolutionary process and evolutionary principle of the survival of the fittest, and the survival of the fittest in the biological world, which makes the genetic algorithm have high value in practical applications, especially in the NoC mapping problem.

However, genetic algorithms also have certain problems in the process of solving:

1. The initial population is randomly generated, which makes the initial population have an uneven distribution of individuals, and there is a problem that the initial population quality is low. In order to solve this problem, we try to use the greedy algorithm to generate the initial population. The purpose is to improve the initial population quality and accelerate the convergence speed by constructing a good initial population.

2. Genetic algorithm is easy to fall into local optimal solution in the later step. This phenomenon is due to the reduction of population diversity in the late iteration of the algorithm. For this problem, we adopt a dual-population genetic mechanism. The dual-population genetic algorithm is a parallel genetic algorithm that uses multiple groups to simultaneously evolve and exchange the genetic information carried by the excellent individuals between the populations to break the equilibrium state within the population to reach a higher equilibrium state and jump out of the local optimum.

So far, a dual-population genetic algorithm (DI_GA) with an optimized initial population has been established.

3.1. Steps of mapping algorithm from task to IP core:

1. **Coding:** The number of tasks and processing elements is often different. Let the current application have a total of N tasks. There are M available processing elements. These processing elements are heterogeneous and classified as m types, according to the relationship between the number of subtasks and the number of processing elements, the coding is divided into the following three cases.

a) The number of processing elements is equal to the number of subtasks. The length of the chromosome is equal to the number of subtasks. Assuming $M = 10, N = 10$, the chromosomes (5, 4, 2, 1, 6, 8, 10, 9, 7, 3) are a feasible scheduling scheme, see Table 1.

Table 1. Example Of Encoding.($M = N$)

Task ID	1	2	3	4	5	6	7	8	9	10
Processing Element ID	5	4	2	1	6	8	10	9	7	3

b) The number of processing elements is larger than the number of subtasks. The length of the chromosome is equal to the number of processing elements, assuming $M = 10, N = 7$. See Table 2 for examples of coding. The IP cores with sequence numbers 9, 7, and 3 are not assigned tasks.

Table 2. Example Of Encoding.($M > N$)

Task ID	1	2	3	4	5	6	7	×	×	×
Processing Element ID	5	4	2	1	6	8	10	9	7	3

c) The number of processing elements is smaller than the number of subtasks, assuming $M = 8, N = 10$, and the coding example is shown in Table 3. Among them, task 1 and task 8 are assigned to the IP core with sequence number 5, and tasks 7 and 10 are assigned to the IP core with sequence number 3.

Table 3. Example Of Encoding.($M < N$)

Task ID	1	2	3	4	5	6	7	8	9	10
Processing Element ID	5	4	2	1	6	8	3	5	7	3

2. The algorithm proposed in this paper uses a dual-population genetic strategy, so we construct two initial populations A , B , and the crossover and mutation probabilities of population A are P_{c1} , P_{m1} , respectively. The crossover and mutation probability of population B are P_{c2} , P_{m2} , respectively. The main processes for constructing the initial population are as follows:

a) Generate a random number $0 \leq r < n$, where n is the number of IP cores of the CTG, and place this random number in the first position of the individual X .

b) The available set $P = \{1, 2, 3, \dots, n\}$ is initialized, and r is deleted from the available set P .

c) Traverse the number x in the available set P , put x into all available positions of the individual X , calculate the fitness value of X after placing these positions, find the largest fitness value Fit from these fitness values, and recording x in the position j where the fitness value of X is maximum, and storing $\langle x, j, Fit \rangle$ into the collection F .

d) Traverse the set F , find the four elements with the largest Fit value according to Fit , randomly select one of the four elements $\langle x, j, Fit \rangle$, put x to the j th position of the individual X , and put x from the available set. Deleted in P .

e) Repeat steps c) and d) until the available set P is empty. When P is empty, it means that a new individual X is generated and X is added to the population pop .

3. Genetic manipulation, on the basis of a better initial solution, successively iteratively generate new chromosomes through selection, crossover, mutation and other operations to construct a dominant population. Through multiple evolutions, the optimal value is achieved.

a) The reciprocal of all IP core's total power consumption is selected as the fitness function. The larger the total power consumption of the IP core is, the smaller the fitness is. The smaller the total power consumption of the IP core is, the greater the fitness is. Use roulette to convert the dominant chromosomes.

b) Crossover is an important operation to generate new solutions, randomly selecting two chromosomes and randomly selecting chromosome intersections. The intersections of the two chromosomes are in the same position, ensuring the length of the chromosome exchange part is consistent. The genes at the same position (from the beginning of the chromosome to the chromosomal variation) of the two chromosomes are replaced with each other. The replaced chromosome may have a phenomenon of gene duplication. As shown in Figure 2, gene repair is required to ensure the reliability of the chromosome.

c) Mutation operation refers to randomly selecting a chromosome and selecting any gene of the chromosome for mutation. As shown in Figure 3, the mutated gene may be duplicated with an existing gene and cannot satisfy subsequent algorithmic conditions. Gene repair is needed to ensure that no identical genes are present on the same chromosome.

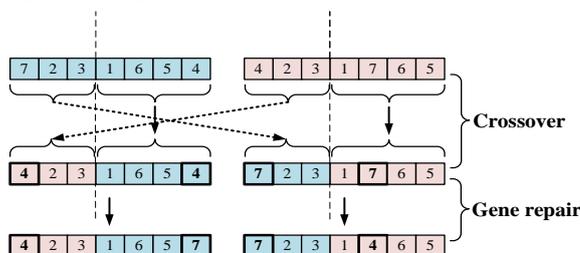


Figure 2. Crossover.

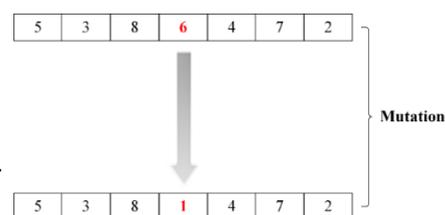


Figure 3. Mutation.

4. Dual-population immigration: When each generation is finished, a random number num is generated, and the optimal chromosome and num chromosomes are selected from A and B for

hybridization to increase the diversity of the population and improve the global search ability of the algorithm.

3.2. Steps of mapping algorithm from IP core to NoC network platform:

The second step of the NoC mapping algorithm execution completes the location mapping of the IP core to the NoC platform with the goal of minimizing power consumption. The initial input of the second step algorithm is the first step feasible solution set. The search process at this step still uses the DI_GA algorithm, but the specific algorithm flow is slightly different from the previous step.

In the encoding process of the second step mapping: each individual of the initial population represents a chromosome, representing a mapping solution. Each chromosome is produced by a string of genes, each of which represents a node of the NoC structure, and its value is the processing elements' number in the set of processing elements that are selected to map to that node. Moreover, in the second step mapping process, since the mapping scheme of the IP core to the NoC platform aiming at minimizing the power consumption is obtained, the NoC's communication power consumption reciprocal is selected as the fitness function, and the communication power consumption of the NoC is larger. The smaller the fitness is, the smaller the communication power of the NoC and the larger the fitness is.

4. Experiment Evaluation

In this section, we will explain how to design an experiment and evaluate the performance of the algorithm. Our experiments are taken on a 2-D 4×4 mesh network platform, and the routing algorithm uses the XY routing algorithm. In order to evaluate the proposed algorithm, the algorithm is applied here to the NoC mapping scheme search of four instances, the examples are MPEG-4, MWD, VOPD, and the mapping is completed by the mapping scheme based on DI_GA. The experimental parameter definition and corresponding values are shown in Table 4. We use the basic genetic algorithm and the random mapping algorithm to compare the effects with DI_GA. To get more accurate experimental simulation data, we used the BookSim simulation simulator. BookSim [10] is a widely used cycle-precision interconnect network simulator invented by the Stanford University nocs group. After some modifications, it can read our custom traffic, simulate NoC operations and get power data.

Table 4. Experimental Parameter Definition

Parameter	Definition	Value	Parameter	Definition	Value
MaxGen	Maximum generation	150	Pmut_Pop_A	Mutation probability of population A	0.03
SizePop	Population size	100	Pcro_Pop_B	Crossover probability of population B	0.2
Pcro_Pop_A	Crossover probability of population A	0.6	Pmut_Pop_B	Mutation probability of population B	0.003

As can be seen from Figure 4. In the MPEG-4 application diagram, the power consumption is reduced by 41.1% compared to the random mapping algorithm. Compared with the basic genetic algorithm, the power consumption is reduced by 6.2%. Compared with the random mapping algorithm in the MWD application diagram, the power consumption is reduced by 47.2%, compared with the basic genetic algorithm, the power consumption decreased by 22.4%. In the VOPD application diagram compared to the random mapping algorithm, the power consumption decreased by 44.5%, basic genetic algorithm is compared with DI_GA, the power consumption is reduced by 10.1%.

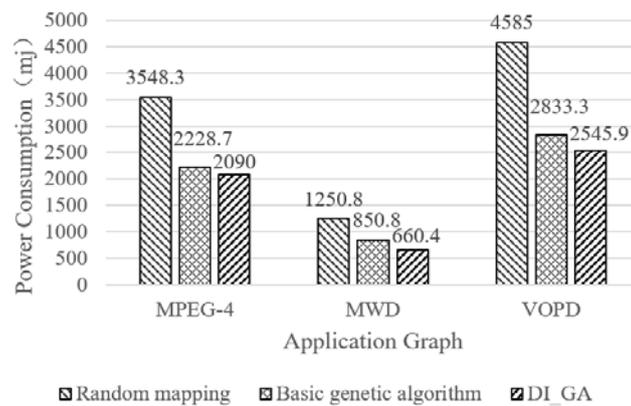


Figure 4. Comparison of three algorithms for MPEG-4, MWD, VOPD power consumption results.

5. Conclusion

In this paper, the DI_GA optimization algorithm is designed. Firstly, the energy consumption calculation model is proposed for the two steps of NoC mapping. The heterogeneous multi-core cooperative system is solved by the staged NoC mapping with energy consumption as the optimization target. In the experiments of simulating complex multimedia systems, the proposed mapping algorithm shows lower power consumption than random mapping and basic genetic algorithms.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. 61202076), along with other government sponsors. The authors would like to thank the reviewers for their efforts and for providing helpful suggestions that have led to several important improvements in our work. We would also like to thank all teachers and students in our laboratory for helpful discussions.

References

- [1] Flich J, Bertozzi D. Designing Network On-Chip Architectures in the Nanoscale Era[M]. Chapman & Hall/CRC, 2010.
- [2] Furhad M H, Kim J M. A shortly connected mesh topology for high performance and energy efficient network-on-chip architectures[J]. Journal of Supercomputing, 2014, 69(2):766-792.
- [3] Sahu P K, Chattopadhyay S. A survey on application mapping strategies for Network-on-Chip design[J]. Journal of Systems Architecture, 2013, 59(1):60-76.
- [4] Zang M, You H. Low power NOC process element mapping using genetic algorithm[J]. Journal of Information & Computational Science, 2012, 9(3):557-563.
- [5] Khalili F, Zarandi H R. A Fault-Tolerant Low-Energy Multi-Application Mapping onto NoC-based Multiprocessors[C]// IEEE, International Conference on Computational Science and Engineering. IEEE, 2013:421-428.
- [6] Sahu P K, Shah T, Manna K, et al. Application Mapping Onto Mesh-Based Network-on-Chip Using Discrete Particle Swarm Optimization[J]. IEEE Transactions on Very Large Scale Integration Systems, 2014, 22(2):300-312.
- [7] Tosun S, Ozturk O, Ozen M. An ILP formulation for application mapping onto Network-on-Chips[C]// International Conference on Application of Information and Communication Technologies. IEEE, 2009:1-5.
- [8] Marcon C A M, Moreno E I, Calazans N L V, et al. Comparison of network-on-chip mapping algorithms targeting low energy consumption[J]. Iet Computers & Digital Techniques, 2008, 2(6):471-482.
- [9] Hu J, Marculescu R. Energy- and performance-aware mapping for regular NoC architectures[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2005, 24(4):551-562.

- [10] Nan Jiang, Daniel U. Becker, George Michelogiannakis, James Balfour, Brian Towles, John Kim and William J. Dally. A Detailed and Flexible Cycle-Accurate Network-on-Chip Simulator. In Proceedings of the 2013 IEEE International Symposium on Performance Analysis of Systems and Software, 2013.