PAPER

Performance-Aware Hybrid Algorithm for Mapping IPs onto Mesh-Based Network on Chip*

Guang SUN[†], Shijun LIN[†], *Nonmembers*, Depeng JIN^{†a)}, Yong LI[†], Li SU[†], *Members*, Yuanyuan ZHANG[†],
and Lieguang ZENG[†], *Nonmembers*

SUMMARY Network on Chip (NoC) is proposed as a new intra-chip communication infrastructure. In current NoC design, one related problem is mapping IP cores onto NoC architectures. In this paper, we propose a performance-aware hybrid algorithm (PHA) for mesh-based NoC to optimize performance indexes such as latency, energy consumption and maximal link bandwidth. The PHA is a hybrid algorithm, which integrates the advantages of Greedy Algorithm, Genetic Algorithm and Simulated Annealing Algorithm. In the PHA, there are three features. First, it generates a fine initial population efficiently in a greedy swap way. Second, effective global parallel search is implemented by genetic operations such as crossover and mutation, which are implemented with adaptive probabilities according to the diversity of population. Third, probabilistic acceptance of a worse solution using simulated annealing method greatly improves the performance of local search. Compared with several previous mapping algorithms such as MOGA and TGA, simulation results show that our algorithm enhances the performance by 30.7%, 23.1% and 25.2% in energy consumption, latency and maximal link bandwidth respectively. Moreover, simulation results demonstrate that our PHA approach has the highest convergence speed among the three algorithms. These results show that our proposed mapping algorithm is more effective and efficient.

key words: mapping algorithm, network on chip, energy consumption, latency, bandwidth

1. Introduction

With the development of the semiconductor technology, large quantities of transistors are available on a single chip, which allows designers to integrate numerous processors together with large amounts of embedded memory [1], [2]. In order to mitigate the complex communication issues which occur as the number of on-chip components increases, Network on Chip (NoC) architecture has been recently proposed as a promising communication paradigm to replace global interconnects [3]–[6]. NoC addresses the on-chip communication problem with a networking approach and provides notable improvements in terms of area, performance, scalability, reliability and flexibility over the traditional bus-based structures [7]–[9].

Regular topology, especially the mesh topology, becomes a kind of popular architecture for NoC design, which

is very simple from a layout perspective [10]. Moreover, the local interconnections between resources and switches are independent of the size of the network in mesh topology [11]. So we focus on mesh topology architecture and assume all PEs have the same size in this paper. The design flow for NoC architecture involves the following three steps [11]. First, analyze the traffic characteristics of an application to obtain a task graph and then convert it into a set of concurrent communicating tasks. Second, according to the property of tasks and the IPs available, select different IPs, assign the tasks and schedule the tasks. Third, map IPs onto the tile-based mesh architecture in such a way as to optimize the metrics of interest. The first two steps have been well addressed in the area of hardware/software co-design and IP-reuse. The last step, mapping, is new to the CAD community and has a strong impact on latency, energy consumption, maximal link bandwidth etc [12], [13]. The mapping problem for NoC is to decide how to topologically place the chosen IP cores onto the tiles of the NoC so as to minimize some given cost function. Generally speaking, this problem allows $p!$ possible solutions if the number of tiles is p . Thus, the solution space of exhaustive search (ES) is clearly unfeasible in the affordable execution time as the size of NoC increases. Moreover, the mapping problem is proved to be a NP-hard one [14]. So it becomes necessary to develop an efficient and effective heuristic mapping algorithm to get an optimal approximation of the desired performance indexes.

In this paper, we discuss the mapping problem. We focus on latency, energy consumption and maximal link bandwidth and propose a performance-aware hybrid algorithm (PHA) for mapping problem so as to search an optimal approximation of the focused performance indexes in a cost-efficient approach. In the PHA, a chromosome represents one topological mapping solution and a gene, which encodes the identifier of the IP core, represents one tile in the NoC architecture. Figure 1 shows an example of mapping IP cores onto a 3×3 mesh-based NoC architecture. The number of IP cores is 9, which equals to the number of the NoC tiles. Each mapping solution corresponds to one permutation from 1 to 9. And the length of chromosome is also 9. Each gene in the chromosome can only take one real value between 1 and 9, and meanwhile, all genes can not be repeated. So a chromosome, which makes up of these genes, also corresponds to one permutation from 1 to 9. Therefore, a chromosome represents one mapping solution. The PHA

Manuscript received January 25, 2010.

Manuscript revised October 25, 2010.

[†]The authors are with State Key Laboratory on Microwave and Digital Communications, Tsinghua National Laboratory for Information Science and Technology, Department of Electronic Engineering, Tsinghua University, Beijing 100084, China.

*This work is supported by the National Natural Science Fund (NNSF-90607009) and the National High Technology Research and Development Program (No. 2008AA01Z107).

a) E-mail: jindp@mail.tsinghua.edu.cn

DOI: 10.1587/transinf.E94.D.1000

A chromosome:

Gene 1	Gene 2	Gene 3	Gene 4	Gene 5	Gene 6	Gene 7	Gene 8	Gene 9
9	3	7	8	5	2	6	4	1

A mapping solution:

Tile 1	Tile 2	Tile 3	Tile 4	Tile 5	Tile 6	Tile 7	Tile 8	Tile 9
9	3	7	8	5	2	6	4	1

Fig. 1 An example of mapping IP cores onto a 3×3 mesh-based NoC architecture.

takes the advantages of Greedy Algorithm, Genetic Algorithm and Simulated Annealing Algorithm. There are three aspects that make PHA a very effective and efficient mapping algorithm.

1. First, greedy swap generates fine initial population efficiently.
2. Second, genetic operations such as crossover and mutation, which are operated with adaptive probabilities according to the diversity of population, make PHA an effective algorithm of strong global parallel search ability.
3. Third, probabilistic acceptance of a worse solution using simulated annealing approach greatly improves the performance of local search.

This algorithm has been implemented and evaluated on three randomly generated benchmarks of a varied number of IP cores [15] and a well known application, namely Video Object Plane Decoder (VOPD) with 16 IP cores [11], [16]. And then, we compare the simulation results among PHA and some previous mapping algorithms such as MOGA [15] and TGA [17]. The optimization objectives of MOGA are energy consumption and maximal link bandwidth, and the optimization objective of TGA is latency. Therefore, we compare MOGA and our PHA in view of energy consumption and maximal link bandwidth respectively, and compare TGA and our PHA in view of latency. Compared with MOGA and TGA, the simulation results show that our algorithm enhances the performance by 30.7%, 23.1% and 25.2% in energy consumption, latency and maximal link bandwidth respectively. Moreover, simulation results demonstrate that our PHA approach has the highest convergence speed among the three algorithms. These results reveal the effect and efficiency of our PHA approach.

The rest of the paper is organized as follows. Section 2 discusses the previous work related to the mapping algorithm. Section 3 formulates the mapping problem mathematically and Sect. 4 depicts the performance models which comprises energy model, latency model and maximal link bandwidth model. In Sect. 5 we propose the PHA for mapping problem and describe its implementation details. Section 6 gives the simulation results compared with MOGA and TGA. Finally, in Sect. 7 we draw conclusions.

2. Related Work

The problem of mapping IPs onto NoC architecture has been attempted by a lot of researchers. Hu et al. [18] propose a branch and bound algorithm to map a given set of IP cores onto a regular tile-based NoC architecture such that the total energy consumption is minimized. Macron et al. [19] compare several NoC mapping algorithms targeting low energy consumption using a communication-weighted model (CWM).

At the same time, several heuristic algorithms for mapping problem have been proposed in view of different performance indexes. Kshitij and Rabindra [15] propose a Multi-objective Genetic Algorithm (MOGA) to map IPs onto regular tile-based NoC in view of minimizing energy and link bandwidth requirements. The MOGA based approach, which is carried out with crossover probability 0.98 and mutation probability 0.01, has been implemented for solving core-tile mapping problem. Tang Lei and Shashi Kumar [17] develop a NoC architecture specific communication delay model to estimate the execution time and present Two-step Genetic Algorithm (TGA) to solve mapping problem in view of minimizing latency requirement.

It is well-known that latency, energy consumption and maximal link bandwidth are very important performance indexes in NoC design. However, the aforementioned mapping algorithms only take into account one or two of these performance indexes. Therefore, these algorithms may lead to a solution that is not globally optimized, for example, low latency but high energy consumption or vice versa. In this paper, our proposed algorithm, namely PHA, takes into consideration all these performance indexes. Moreover, the aforementioned heuristic mapping algorithms do not consider the efficiency of the algorithms. In this paper, we take convergence speed as a performance parameter for mapping algorithm. Performance indexes such as latency, energy consumption and maximal link bandwidth reveal the effect of the mapping algorithms, and convergence speed reveals the efficiency of the mapping algorithms. For heuristic algorithms, the efficiency of the algorithms is a key consideration.

3. Problem Formulation

In order to formulate the mapping problem mathematically, we need the following two definitions.

The communication of the IP cores of the application

is represented by the Application Characterization Graph, which is defined in *definition 1*.

The connective property of the NoC topology architecture is represented by the NoC Topology Architecture Graph, which is defined in *definition 2*.

Definition 1 (ACG): The Application Characterization Graph is a directed graph, $ACG=G(C,W)$, where $C = \{c_1, c_2 \dots c_n\}$ represents the set of IP cores, corresponding to the set of vertices of ACG, and the directed edges which represent the communication dependency between cores are associated with two variables, namely W and Bw . Weight $W = \{w_{ij} : \forall(c_i, c_j) \in C\}$ denotes the communication amount, in bits, from c_i to c_j and $Bw = \{bw_{ij} : \forall(c_i, c_j) \in C\}$ denotes the link bandwidth that satisfies the required communication transmitted from c_i to c_j . N denotes the number of IP cores.

Definition 2 (TAG): The NoC Topology Architecture Graph is a directed graph, $TAG=G(T,L)$, where $T = \{t_1, t_2 \dots t_m\}$ represents the set of tiles of NoC, corresponding to the set of vertices of TAG, and $L = \{l_{ij} : \forall(t_i, t_j) \in T\}$ denotes the direct connective link from t_i to t_j , corresponding to the set of edges. M represents the number of NoC tiles.

Problem formulation:

Using the aforementioned definitions, we can formulate the mapping problem mathematically as follow:

Given an ACG and a TAG that satisfy

$$N \leq M, \quad (1)$$

Find a mapping function from ACG to TAG and satisfy

$$\forall c_i \in C, \text{map}(c_i) = t_j \in T, \quad (2)$$

$$\forall c_i \neq c_j, \text{map}(c_i) \neq \text{map}(c_j), \quad (3)$$

Such that the objective functions, which comprises energy consumption, latency and maximal link bandwidth, are optimized.

4. NoC Performance Models

In this section, we develop NoC performance models including energy model, latency model and maximal link bandwidth model. According to these models, we can evaluate the performance of the mapping algorithms.

4.1 Energy Model

Energy consumption of an application primarily comes from routers and communication links. The dynamic energy consumed by one bit of data transmitting from core c_m to core c_n can be calculated as follow [15], [19]:

$$E_{Bit-mn} = \alpha \times E_{Rbit} + (\alpha - 1) \times E_{Lbit}, \quad (4)$$

where E_{Rbit} and E_{Lbit} represent the energy consumed by router and the links between tiles when transmitting one bit of data, and α represents the number of routers that the bit

of data passes through. E_{Rbit} and E_{Lbit} are 5.445 PJ and 0.43 PJ in [15].

Then, the total amount of NoC energy consumption (E_{NoC}) for all inter-core communications is calculated as [19]:

$$E_{NoC} = \sum_{m=1}^N \sum_{n=1}^N w_{mn} \times E_{Bit-mn}, \forall(c_m, c_n) \in C, \quad (5)$$

where w_{mn} represents the communication amount transmitted from core c_m to core c_n and N denotes the number of IP cores.

4.2 Latency Model

In this paper, we use the latency model proposed in [17], which assumes that data always transmits with minimum-path routing in the network. When the data transmits from tile (x_i, y_i) to tile (x_j, y_j) , the distance is evaluated as:

$$D = |x_i - x_j| + |y_i - y_j|. \quad (6)$$

The transmission delay from tile (x_i, y_i) to tile (x_j, y_j) can be evaluated as:

$$T_{ij} = k \times w_i \times D, \quad (7)$$

where k is a constant decided by the properties of links and switches and w_i represents the data-width in NoC.

4.3 Maximal Link Bandwidth Model

For single minimum-path routing, like XY routing, the bandwidth of every link can be computed as long as the core-tile mapping is determined. Let t_i and t_j represent the tiles to which core c_m and core c_n are, respectively, mapped. Therefore, $t_i = \text{map}(c_m)$ and $t_j = \text{map}(c_n)$. Let BW_{ab}^{mn} represent the link bandwidth from tile t_i to tile t_j , namely the bandwidth of l_{ab} , which is occupied by the communication weight transmitting from cores c_m to c_n . BW_{ab}^{mn} is computed as follow:

$$BW_{ab}^{mn} = \begin{cases} bw_{mn}, & l_{ab} \in \text{Path}(\text{map}(c_m), \text{map}(c_n)) \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

The total link bandwidth of l_{ab} , which is denoted by BW_{ab} , can be obtained by the following equation:

$$BW_{ab} = \sum_{m=1}^N \sum_{n=1}^N BW_{ab}^{mn}. \quad (9)$$

The maximal link bandwidth of the NoC can be represented as follow:

$$BW_{\max} = \max(BW_{ab}), \forall l_{ab} \in L. \quad (10)$$

5. Performance-Aware Hybrid Algorithm (PHA)

In this section, we describe the PHA and its implementation details. This technology is a hybrid of greedy algorithm, genetic algorithm and simulated annealing algorithm. Figure 2 shows the main process of PHA.

In PHA, a chromosome corresponds to one topological mapping solution and a gene, which encodes the identifier of the IP core, corresponds to one tile in the NoC architecture. Therefore, for the mapping problem of $X \times Y$ mesh-based NoC architecture, if the identifier of IP core, which the k th gene in chromosome encodes, is i , core c_i is mapped onto the k th tile which is in the $\lceil k/X \rceil$ th row and in the $(k \bmod X)$ th column in mesh-based NoC architecture. And the fitness value of a chromosome corresponds to the performance of a mapping solution such as energy consumption, latency and maximal link bandwidth.

To describe the PHA, the following definitions of variables are needed. Table 1 describes some commonly used variables in the PHA.

Fig. 2 The process of PHA for mapping problem.

In the following, we describe the precess of PHA. The implementation details of PHA consist of six steps.

Step 1: Obtain the input parameter values and set the parameter values for iterations.

Step 1.1: Input the ACG and TAG to get communication extraction and target architecture model.

Step 1.2: Set the parameter values for iterations, which comprise P_c , P_m , T , G_{max} , POP , SSC and SSC_{max} . The annealing temperature T is $10 \ln N$ [19], where N is the number of IP cores.

Step 2: Create the initial population through greedy swap and calculate the fitness values

Step 2.1: Initialize the POP population members randomly

Step 2.2: For every individual, obtain better solution through greedy swap [19], which is implemented by two nested loops.

At the beginning of the external loop, we fix an NoC position as the pivot for evaluation. The fixed pivot changes each time the external loop is excused. The first fixed position is the top-left corner of the mesh-based NoC and the last is the bottom-right corner. In the internal loop, we swap the chosen pivot with each IP core in the remaining NoC positions which include only those not yet chosen by external loop as fixed positions. At the end of internal loop, the mapping which leads to the best solution in the current internal loop will be updated as the input for next internal loop.

Step 2.3: Calculate the fitness value for every individual.

Step 3: Selection, crossover and mutation for the evolution of new population.

Step 3.1: Roulette wheel selection [20] is operated to preserve the high quality chromosomes to the next generation.

We select two chromosomes from current generation and for every chromosome, the probability to be selected is proportion to the fitness value. The greater the fitness value, the higher probability the chromosome is selected with.

Step 3.2: Implement crossover operation with the probability of P_c .

Because the identifiers of the IP cores can not be duplicated, we need special crossover operation that satisfies the constraint. Specially, we use permutation crossover ap-

Table 1 List of commonly used variables in the PHA.

Variable	Description
P_c	the probability of crossover
P_m	the probability of mutation
T	the annealing temperature
G_{max}	the maximal number of generation evolved.
POP	the number of individuals in every generation.
SSC	the search stagnation counter, which will add one if the best solution of the next generation is not better than the one of the current generation, otherwise reset as zero.
SSC_{max}	the maximal number of SSC
F_{max}	the maximal fitness value in the population
F_{ave}	the average fitness value in the population

Fig. 3 Crossover operation.

proach between two chromosomes selected in step 3.1. Figure 3 demonstrates an example of crossover operation. The crossover process consists of two steps. In the first step, a crossover point is chosen randomly. In the second step, the genes of *Child_A* behind the crossover point inherit the genes of *Parent_A* behind the crossover point. The genes of *Child_A* before the crossover point inherit the genes of *Parent_B* in order and the genes which are the same as the ones of *Child_A* behind the crossover point are deleted. The rule of crossover operation for *Child_B* is the same as the one for *Child_A*.

Step 3.3: Carry out mutation operation with the probability of P_m .

The mutation operation is implemented by swapping two positions of genes three times, which are chosen randomly.

Step 4: Reproduce new population for the next generation

Step 4.1: Repeat step 3 until we obtain new *POP* population members and calculate the fitness values of the new population.

Step 4.2: Sort the $2 \times POP$ population members, which consist of the current generation and the new *POP* population members, in a non-increasing order according to the fitness values of chromosomes. Select the best *POP* individuals as the next generation.

Step 5: Local search for every chromosome of the next generation with the simulated annealing approach to get better solution [11].

Step 5.1: Local search for better solution with the simulated annealing approach.

For every individual of the next generation, set the current mapping as the local best mapping, and swap two randomly chosen genes of chromosome to carry out local search. After this, calculate the fitness value of the new mapping and compare with the one of the local best mapping. If the former is better than the latter, it is stored as new local best mapping and becomes the new current mapping. Otherwise, accept the worse mapping with the probability of P depicted as follow:

$$P = \exp\left(\frac{-\Delta fitness}{k \times T}\right), \quad (11)$$

where $\Delta fitness$ represents the difference of fitness values of

the two mappings, k is a constant and T denotes the annealing temperature. T is used to control a stochastic acceptance procedure. The higher the temperature, the greater probability the worse mapping solution is accepted with.

Step 5.2: Let p be the number of local search and repeat step 5.1 p times to search better solution locally.

Step 5.3: Repeat step 5.1 and 5.2 until all chromosomes of the next generation have implemented the local search to get better solution.

Step 6: Update the parameter values for iterations and check whether it satisfies iteration stopping criterion.

Step 6.1: Calculate the fitness values of next generation and update the following parameters for iterations:

$$G = G + 1, \quad (12)$$

$$T = T \times 90\%, \quad (13)$$

$$P_c = 0.5 + 0.5 \times \exp\left(\frac{F_{max} - F_{ave}}{F_{max}} \times (-10)\right), \quad (14)$$

$$P_m = 0.15 \times \exp\left(-\frac{F_{max} - F_{ave}}{F_{max}}\right). \quad (15)$$

Find the best solution in the next generation and compare with the one in the current generation. *SSC* will add one if the best solution in the next generation is not better than the one in the current generation, otherwise reset as zero.

Step 6.2: Check whether it satisfies iteration stopping criterion. The iteration stopping criterion is

$$G \geq G_{max}, \quad (16)$$

or

$$SSC \geq SSC_{max}. \quad (17)$$

If the iteration stopping criterion is satisfied, finish the search and output the best solution. Otherwise, go to step 3 for the next iteration.

6. Simulation Results

In this section, we implement the simulation on three randomly generated benchmarks of a varied number of IP cores [15] and a well known application, namely Video Object Plane Decoder (VOPD) with 16 IP cores [11], [16]. The evaluated chip architecture is mesh-based. Target chips in four benchmarks consist of '16', '25', '36' and '16' IP cores, namely 4×4 , 5×5 , 6×6 and 4×4 , respectively. After that, we compare the simulation results among PHA and some previous mapping algorithms such as MOGA [15] and TGA [17], in order to evaluate the performance of these algorithms.

All algorithms are implemented in MATLAB and executed on PC with DUO CPU E8400 with 3.0GHZ operation frequency and 2 GB of main memory. In every benchmark, three mapping algorithms use the same routing, performance model and have the same input parameters such as ACG and TAG. In order to compare the search efficiency and effect, both the execution time and the improvement of

Fig. 4 Energy saving regarding to same random mappings.

performance indexes are taken into account.

The optimization objectives of MOGA are energy consumption and maximal link bandwidth, and the optimization objective of TGA is latency. Therefore, we compare MOGA and our PHA in view of energy consumption and maximal link bandwidth respectively, and compare TGA and our PHA in view of latency.

Figure 4 and Fig. 5 depict the comparison of MOGA and PHA in energy saving and maximal link bandwidth

Fig. 5 Maximal link bandwidth reduction regarding to same random mappings.

reduction respectively regarding to the same random mappings. These show that our PHA based approach saves 30.7% of energy consumption in average and reduces 25.2% of maximal link bandwidth in average as compared to the MOGA based approach.

Figure 6 illustrates the comparison of TGA and PHA in latency reduction regarding to the same random mappings. It shows that our PHA based approach can reduce 23.1% of latency in average as compared to the TGA based approach.

Moreover, Fig. 4, Fig. 5 and Fig. 6 all demonstrate that our PHA approach has the highest convergence speed among the three algorithms. These performance curves rep-

Fig. 6 Latency reduction regarding to same random mappings.

resent the performance improvements at different simulation time. Each plot in the curves corresponds to a performance improvement after some iterations. So the first plot in the performance curve represents the performance improvement after the first iteration. And the first iteration costs more time because of some initial operations. Our algorithm costs more time than MOGA and TGA in every iteration. However, our algorithm needs less iteration than MOGA and TGA to get good results. Therefore, our algorithm is faster and more efficient.

7. Conclusions and Future Work

In this paper, a performance-aware hybrid algorithm (PHA) for mapping IP cores onto mesh-based NoC is proposed. In the PHA, we take into consideration not only the effect of algorithm but also the efficiency of algorithm. Performance indexes such as latency, energy consumption and maximal link bandwidth reveal the effect of the mapping algorithms, and convergence speed reveals the efficiency of the mapping algorithms. The PHA takes the advantages of Greedy Algorithm, Genetic Algorithm and Simulated Annealing Algorithm. There are three reasons that make PHA a very effective and efficient mapping algorithm. First, instead of generating initial population randomly, it creates a fine initial population efficiently through greedy swap. Second, effective global parallel search is implemented by genetic operations such as crossover and mutation. Moreover, according to the diversity of population, the operations of crossover and mutation are implemented with adaptive probabilities instead of invariant probabilities. So the convergence of algorithm can speed up. Third, in order to improve the performance of local search for the PHA, probabilistic acceptance of a worse solution using simulated annealing method is carried out for every chromosome of the next generation. Compared with MOGA and TGA, simulation results show that our algorithm improve the performance by 30.7%, 23.1% and 25.2% in energy consumption, latency and maximal link bandwidth respectively. Moreover, simulation results show that our PHA approach has the highest convergence speed among the three algorithms. These results reveal the effect and efficiency of our PHA approach.

The performance models in our algorithm is coarse-grained. So we will develop fined-grained performance models in our future work and consider some important low-power techniques, such as DVFS and power gating.

References

- [1] J. Hu and R. Marculescu, "Energy-aware mapping for tile-based NoC architectures under performance constraints," *Proc. ASP-DAC* 2003, pp.233–239, 2003.
- [2] L. Benini and G.D. Micheli, "Networks on chips: A new SoC paradigm," *Computer*, vol.35, no.1, pp.70–78, 2002.
- [3] R. Marculescu, U. Ogras, L.S. Peh, N. Jerger, and Y. Hoskote, "Outstanding research problems in NoC design: System, microarchitecture, and circuit perspectives," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol.28, no.1, pp.3–21, 2009.
- [4] C. Marcon and A. Borin, "Time and energy efficient mapping of embedded applications onto NoCs," *ASP-DAC05*, pp.33–38, 2005.
- [5] M. Kreutzand, C. Marcon, L. Carroand, N. Calazans, and A. Susin, "An energy and latency evaluation of NoC topologies," *ISCAS05*, vol.6, pp.5866–5869, 2005.
- [6] S. Murali, M. Coenen, A. Radulescu, K. Goossens, and G.D. Micheli, "A methodology for mapping multiple use-cases onto networks on chips," *Proc. Des. Autom. Test Eur. Conf.*, pp.118–123, 2006.
- [7] H.G. Lee, N. Chang, U.Y. Ogras, and R. Marculescu, "On-chip communication architecture exploration: A quantitative evaluation of point-to-point, bus, and network-on-chip approaches," *ACM Trans.*

- Des. Autom. Electron. Syst., vol.12, no.3, pp.20–40, 2007.
- [8] W. Dally and B. Towles, "Route packets, not wires: On-chip interconnection networks," Proc. Des. Autom. Conf., pp.684–689, 2001.
 - [9] M. Horowitz, R. Ho, and K. Mai, "The future of wires," Proc. IEEE, vol.89, no.4, pp.490–504, 2001.
 - [10] P. Gratz, C. Kim, R. McDonald, S.W. Keckler, and D.C. Burger, "Implementation and evaluation of on-chip network architectures," Proc. Int. Conf. Comput. Des., pp.477–484, 2006.
 - [11] F. Moein-darbari, A. Khademzadeh, and G. Gharooni-fard, "Evaluating the performance of a chaos genetic algorithm for solving the network on chip mapping problem," CSE'09, vol.2, pp.366–373, 2009.
 - [12] L. Hua, Z. Liang, et al., "A fast hierarchical multi-objective mapping approach for mesh based networks on-Chip," Acta Scientiarum Naturalium University Pekinensis, vol.44, no.5, pp.711–716, 2008.
 - [13] A. Hansson, K. Goossens, and A. Radulescu, "A unified approach to mapping and routing on a network-on-chip for both best-effort and guaranteed service traffic," Hindawi VLSI Design, 2007.
 - [14] M. Garey and D.S. Johnson, Computers and intractability: A guide to the theory of NP-completeness, W H Freeman and Co, San Francisco, 1979.
 - [15] K. Bhardwaj and R.K. Jena, "Energy and bandwidth aware mapping of IPs onto regular NoC architectures using multi-objective genetic algorithms," SOC 2009, pp.027–031, 2009.
 - [16] S. Murali and G.D. Micheli, "Bandwidth-constrained mapping of cores onto NoC architectures," Proc. Des., Autom. Test Eur. Conf, pp.896–901, 2004.
 - [17] T. Lei and S. Kumar, "A two-step genetic algorithm for mapping task graphs to a network on chip architecture," Digital System Design, pp.180–187, 2003.
 - [18] J. Hu and R. Marculescu, "Energy-aware mapping for tile-based NOC architectures under performance constraints," Proc. Asia South Pacific Des. Autom. Conf., pp.233–239, 2003.
 - [19] C. Marcon, E. Moreno, et al., "Comparison of network-on-chip mapping algorithms targeting low energy consumption," IET Computers & Digital Techniques 2008, vol.2, no.6, pp.471–482, 2008.
 - [20] D. Zhang, Research on multi-mode SSR and optimal design of SEDE, Master thesis. Tsinghua University, Beijing, 2007.

Guang Sun received the B.S. from Xiamen University, Xiamen, China in 2008 in the department of electronics engineering. Now he is pursuing the Ph.D. degree in Tsinghua University. His research interests include on-chip network, message-passing multiprocessor and NoC architecture.

Shijun Lin received the B.S. degree from Xiamen University, Xiamen, China in 2005 and Ph.D. degree from Tsinghua University, Beijing, China in 2010. Now he is an assistant professor at Xiamen University. His research interests include On-Chip interconnect, high-speed networks and ASIC design.

Depeng Jin received the B.S. and Ph.D. degrees from Tsinghua University, Beijing, China, in 1995 and 1999 respectively both in electronics engineering. Now he is an associate professor at Tsinghua University and vice chair of Department of Electronic Engineering. Dr. Jin was awarded National Scientific and Technological Innovation Prize (Second Class) in 2002. His research fields include telecommunications, high-speed networks, ASIC design and future internet architecture.

Yong Li received the B.S. from Huazhong University of Science and Technology, Wuhan, China in 2007 in the department of electronics and information engineering. Now he is pursuing the Ph.D. degree in Tsinghua University. His research interests include wireless networking, mobility management and future internet architecture.

Li Su received the B.S. degree from Nankai University, Tianjin, China, in 1999 and Ph.D. degree from Tsinghua University, Beijing, China in 2007 respectively both in electronics engineering. Now he is a research associate with Department of Electronic Engineering, Tsinghua University. His research interests include telecommunications, future internet architecture and on-chip network.

Yuanyuan Zhang received the B.S. from Shandong University, Jinan, China in 2007 in the department of electronics engineering. Now she is pursuing the Ph.D. degree in Tsinghua University. Her research interests include Network-on-Chip network, Multiprocessor System-on-Chip and Chip Multiprocessor.

Lieguang Zeng received the B.S. from Tsinghua University, Beijing, China, in 1970. Since 1970, he has been with the Department of Electronic Engineering, Tsinghua University, Beijing, China, where he is currently a Professor. Prof. Zeng has been awarded National Scientific and Technological Innovation Prize (Second Class) three times, in 1987, 1991 and 2002 respectively. He has also been awarded provincial and ministerial level science and technology progress prize eight times. Prof. Zeng has published more than 150 research papers, and obtained 4 patents. His research interests include telecommunications, high-speed networks, ASIC design, and future internet architecture.

published more than 150 research papers, and obtained 4 patents. His research interests include telecommunications, high-speed networks, ASIC design, and future internet architecture.