Journal of Theoretical and Applied Information Technology
31st March 2013. Vol. 49 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

LOW ENERGY MAPPING TECHNIQUE FOR HIERARCHICAL NETWORK-ON-CHIP

[1]XIAONA XIE, [2]QINGXIN ZHU, [3]ZHENGWEI CHANG, [3]YU FANG

[1]School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, Sichuan, China; Physical Education College of Zhengzhou University, Zhengzhou 450044, Henan, China

[2]School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, Sichuan, China

[3] Sichuan Electric Power Research Institute, Chengdu 610072, Sichuan, China

E-mail: [1]xnxieok@163.com, [2]qxzhu@uestc.edu.cn

ABSTRACT

Intellectual property (IP) core-to-node mapping is an important but intractable optimization problem in Network-on-Chip (NoC) application design. In this paper, we present an approach to map cores onto hierarchical NoC architecture which consists of two levels. The top-level interconnection network is realized by a 2D mesh of communicating routers, and a tree based topology is used at the second level of interconnection hierarchy. We formulate the problem of low energy mapping, and introduce a three-phase optimization strategy to solve it. The cores are clustered firstly, and a tabu search based mapping technique is proposed to map the core clusters to the top-level interconnection network. Then the cores within each cluster are mapped to the nodes of every sub-network. Experimental results show that the proposed method is very fast and effective in terms of energy optimization.

Keywords: *Hierarchical Network-on-Chip, Low Energy, Mapping, Cluster, Tabu Search*

1. INTRODUCTION

With the advance of integrated circuits and the semiconductor technology, billions of transistors and hundreds of computation resources or intellectual property (IP) cores are allowed to be put on a single chip. System-on-chips (SoCs) with these capabilities require efficient communication architecture to offer scalable bandwidth and parallelism. Network-on-chip (NoC) [1, 2] has been proposed to overcome the complex on-chip communication problem of SoC design. For these complex NoCs, power consumption becomes the most pressing design problem.

An important phase in the design flow of NoCs is to map a set of IPs onto a set of resource nodes which connected to local ports of routers to optimize a cost function (e.g. energy consumption). Generally, consider a system composed of n IPs, this mapping problem allows $n!$ possible solutions. Even for a small problem size, the entire search space of this problem is huge. In [3], a heuristic branch-and-bound algorithm is developed to tackle mapping for tile-based NoC. In addition, a lot of works with different objectives have been conducted in NoC mapping [4-6], and a survey is given in [7]. Most of the existing approaches assume a mesh NoC topology, since it is straightforward for chip surface layout.

On the other way, application specific NoC is an emerging paradigm handle the communication problem between the set of computational resources in modern embedded system applications. This type of NoC optimizes design of the on-chip network to comply with the application specific SoC requirements. Hierarchical NoC architecture [8] can achieve the trade-off of regular and custom network topology, and is more suitable for larger size or application specific systems. Moreover, it can reduce the complexity of hierarchical NoC design using platform-based design techniques or divide-and-conquer methods. In this work, we assume hierarchical NoC architecture which consists of two levels. The top-level interconnection network is realized by a 2D mesh of communicating routers, and a tree based topology is used at the

Journal of Theoretical and Applied Information Technology
31st March 2013. Vol. 49 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

second level of interconnection hierarchy. Trees are natural for exploiting locality of traffic. So the IPs within a sub-network could communicate efficiently [9].

In this paper, we address energy-aware mapping of IPs onto hierarchical NoC architecture. We first describe the NoC model and the associated energy model, and then formulate the low energy mapping problem. A three-phase clustering and tabu search based mapping approach is proposed to map the IPs to the resource nodes of a given NoC, such that the total communication energy is minimized.

The rest of this paper is organized as follows. Section 2 describes the problem definition. Section 3 presents the low energy mapping algorithm of hierarchical NoC architecture. Section 4 shows experimental results. Finally, Section 5 gives the conclusions.

2. PROBLEM DEFINITION

In this section, we describe the two-level hierarchical NoC architectures and its associated energy model. Then the low energy mapping problem formulation is given.

2.1 NoC Model

As shown in Fig. 1, the on-chip communication architecture under consideration in this study is composed of two levels. At the top level, the main network is interconnected by a 2D mesh of communicating routers. We call these routers as global routers. Each global router is also connected to the four neighboring routers via links. A local sub-network is connected to a global router. A global router may connect to other global routers only.

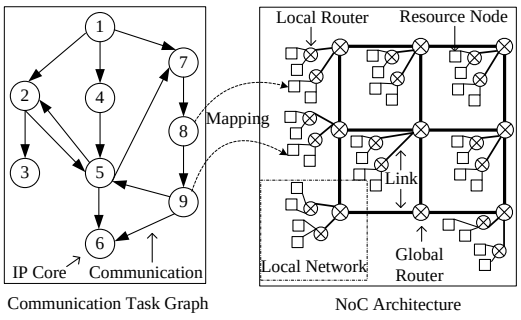

Fig.1: The Mapping Problem Models

At the second level, all the leaves and vertices are interconnected by a binary tree as shown in Fig. 1. Besides binary tree based topology, the *k*-ary trees ($k > 2$) or fat trees such as SPIN [10] may be used in the sub-networks. In our model, the leaf nodes do all computation, and the rest of the nodes (routers) are used only for communication. We call these non-leaf routers in the sub-networks as local routers. In other words, the resources are placed at the leaves and the local routers placed at the vertices.

The routing of the local network is quite easy as the best path is defined by the topology itself. The messages are generated by the computation resources. When a packet comes from a downlink, the local router has to see if it belongs to its sub-tree. If it isn't the router must address it to the father router in the high level. Static XY routing is assumed for the global communication. It first routes packets along the X-axis. Once it reaches the column under which the destination tile is located, the packet is then routed along the Y-axis.

2.2 Energy Model

When packets travel on the interconnection network, both the routers and the inter-routers links will result in energy dissipation. We are concerned with the dynamic energy dissipation caused by the communication tasks. Consider the communications between two resource nodes of n hops, the energy consumption of sending one bit of data is modeled as:

$$E_{bit}(n) = n \times E_{r_{bit}} + (n-1) \times E_{l_{bit}} \quad (1)$$

where n is the number of routers the data passes on its way along a path, and $E_{r_{bit}}$ and $E_{l_{bit}}$ represent the energy consumed by a router and a link per bit, respectively.

2.3 Problem Formulation

As Fig.1 shown, the objective of the mapping problems is to select IPs from a communication task graph and assign them to different resource nodes of the NoC architecture, such that the total communication energy consumption is minimized. More formally, this problem can be formulated as follows. Given:

- a directed communication task graph *CTG(C,A)*, where each $c_i \in C$ denotes a IP core, and the directed edge $a_{ij} \in A$ denotes the communication from c_i to c_j. For every a_{ij}

Journal of Theoretical and Applied Information Technology
31st March 2013. Vol. 49 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$\in A$, $v(a_{ij})$ denotes the communication volume in bits from c_i to c_j.

- a hierarchical NoC architecture defined as a tuple $NA = (CN, GR, SR, GP, CP)$, where each $cn_i \in CN$ denotes a resource node, each $gr_i \in GR$ denotes a global router, each $sr_{ij} \in SR$ denotes a local router of the sub-network connected gr_i directly, each $gp_{ij} \in GP$ denotes the routing path from a global router gr_i to another global router gr_j, and each $cp_{ij} \in CP$ denotes the routing path from resource node cn_i to cn_j. For every $gp_{ij} \in GP$, $e(gp_{ij})$ denotes the average energy consumption of sending one bit of data from global router gr_i to gr_j. For every $cp_{ij} \in CP$, $e(cp_{ij})$ denotes the average energy consumption of sending one bit of data from resource node cn_i to cn_j.

When $|C| \leq |CN|$, the problem of minimizing the total communication energy is to obtain a one to one mapping function $\phi : C \to CN$ which:

$$\min\left\{E(\phi) = \sum_{\forall a_{ij} \in A}\left(v(a_{ij}) \times e(cp_{\phi(c_i),\phi(c_j)})\right)\right\} \quad (2)$$

such that:

$$\forall c_i \in C, \exists cn_i = \phi(c_i) \in CN \quad (3)$$

$$\forall c_i \neq c_j \in C, \phi(cn_i) \neq \phi(cn_j) \quad (4)$$

Therefore, the communication energy consumption can be minimized by minimizing the cumulative traffic flowing through each routing paths. Moreover, the traffic flowing along GP has more impacts on the total energy consumptions. So the IPs communicate frequently should be mapped on to a common sub-network.

3. THE MAPPING ALGORITHM

Finding an optimal mapping for NoC that consumes the least energy is known to be NP-hard. For the large-size hierarchical NoC system, the mapping problem is even harder.

In this work, we combine two techniques, namely, cores clustering and clusters mapping to address the IP mapping problem of hierarchical NoC. Clustering can greatly simplify the mapping problem because the mapping can be conducted cluster wise instead of node wise. As shown in Fig.2, we propose a three-phase mapping approach.

In the first phase, the IP cores that communicate with each other frequently are clustered into a group, such that the inter-cluster communication volumes are less.

In the second phase, a tabu search algorithm is used to mapping the clusters on to the global routers.

Finally, in the third phase, the IP cores within each cluster are mapped to the resource nodes of the tree based sub-networks.

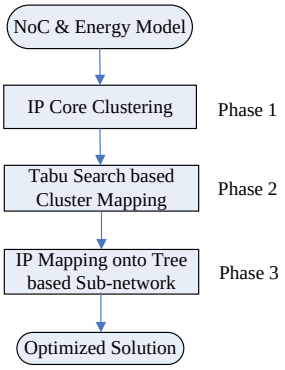

Fig.2: Three-phase Mapping Algorithm

3.1 IP Core Clustering

Clustering is a general technique to partition nodes (cores) into groups according to "*distance*" property. Here, the clustering exploits the knowledge about the communication demands of the tasks to achieve communication locality. The distance of a core depends on the number of connections and the communication volume for each connection [11].

The first phase is to partition IP cores into clusters, matching the number of global routers $|GR|$. And the size of each cluster equals the number of resource nodes of a sub-network. The implementation of IP cores clustering is omitted here.

Formally, as the output of the first phase, we define the cluster communication task graph $CLTG(CL, AC)$, where $CL = \{cl_i \mid 1 \leq i \leq |GR|\}$ is a set of IP clusters, and each $cl_i \in CL$ is a set of IP cores. And the directed edge $ac_{ij} \in AC$ denotes the communication from c_i to c_j. So in the second phase, we should find a cores cluster to global router mapping $\phi_{CL} : CL \to GR$, such that the

Journal of Theoretical and Applied Information Technology
31st March 2013. Vol. 49 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

communication energy consumption of the top level main network $E(\phi_{CL})$ is minimized.

3.2 Tabu Search based Cluster Mapping

The second phase is to make a one-to-one mapping ϕ_{CL} using tabu search. Tabu search is a heuristic introduce by Glover [12]. This method has been successfully applied to a variety of problems. Several researchers have compared tabu search with other optimization heuristics, such as simulated annealing and genetic algorithms, and it is observed that tabu search is superior with regard to optimization time and quality of results [13, 14].

The main ideas of tabu search may be sketched as follows. Tabu search starts from an initial solution *S*, and moves repeatedly from a solution to a neighboring one. At each step of the procedure, a set $N(S)$ of the neighboring solutions of the current solution *S* is considered and the move that improves most the objective function value is chosen. If there are no improving moves, tabu search chooses one that least degrades the objective function. In order to avoid returning to the local optimal just visited, the reverse move must be forbidden. This is done by storing the move or a characterization of the move in a memory called a tabu list. The tabu list keeps information on the last *h* moves during the search process. The parameter *h* is called the tabu list size. However, the tabu list may forbid certain relevant or interesting moves. Consequently, an aspiration criterion is introduced to allow tabu moves to be chosen. As a result, the tabu search approach accepts uphill moves and simulates convergence toward a global optimum by taking advantage of the search history at selection of the next move.

The key elements of tabu search are object function, neighborhood, tabu list, tabu list size, and aspiration function. In addition, in this work diversification is introduced to search more efficiently.

(1) Object function. Let ϕ_{CL}^{now} represent the current configuration of IP cluster mapping, and its object function is given by $E(\phi_{CL}^{now})$.

(2) Neighborhood. A move from one solution to another solution consists of exchanging two IP clusters so that each occupies the global router formerly occupied by the other. Let $N(\phi_{CL}^{now})$ denote the neighborhood of ϕ_{CL}^{now} in the solution space, which is the set of all the neighbors of ϕ_{CL}^{now}. Starting from a placement ϕ_{CL}^{now}, a neighboring placement $\phi_{CL}^{next} \in N(\phi_{CL}^{now})$ is obtained by permuting IP cl_i and cl_j:

$$\phi_{CL}^{next}(cl_i) = \phi_{CL}^{now}(cl_j) \quad (5)$$

$$\phi_{CL}^{next}(cl_j) = \phi_{CL}^{now}(cl_i) \quad (6)$$

$$\forall cl_k \in CL \wedge cl_k \notin \{cl_i, cl_j\}, \phi_{CL}^{next}(cl_k) = \phi_{CL}^{now}(cl_k) \quad (7)$$

(3) Tabu list. The tabu list is constituted of pairs (cl_i, cl_j) of IP clusters that cannot be exchanged. The prohibition of the list is implemented with a 2-dimential array *TL* of integers where the element *TL*[*i*, *j*] identifies the value of the future iterations at which the two IPs may again be exchanged with each other. By this means, testing whether a move is tabu or not require only one comparison.

(4) Tabu list size. The tabu list size *h* varies randomly between $0.4\times|CL|$ and $0.6\times|CL|$.

(5) Aspiration function. We ignore the tabu status of a move if the solution produced is better than the best found so far.

(6) Diversification. If the best solution has not been updated for $0.2\times|CL|$ iterations, tabu search may trap into a local optimum. In this case, a diversification phase follows in which a rarely used move is performed in order to force the heuristic to escape the current region and explore different region in the design space. A diversification of the current solution ϕ_{CL}^{now} is achieved by performing a rare move in the neighborhood $N(\phi_{CL}^{now})$.

The steps of the tabu search algorithm for cluster mapping can be summarized as follows:

(1) Initialization: Generate initial cluster mapping configuration ϕ_{CL}^{now} randomly, and calculate the object function $E(\phi_{CL}^{now})$. Initialize the best-so-far solution, set $\phi_{CL}^{best} = \phi_{CL}^{now}$. Initialize the tabu list, set *TL*[*i*, *j*] = 0, for $1 \le i \le |CL|$ and $1 \le j \le |CL|$.

(2) Calculate the object function $E(\phi_{CL}^{next})$.

(3) For each ϕ_{CL}^{next} in increasing order of $E(\phi_{CL}^{next})$, represent the move as (cl_m, cl_n), and repeat the following steps

Journal of Theoretical and Applied Information Technology
31st March 2013. Vol. 49 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

(3.1) If $E(\phi_{CL}^{next}) < E(\phi_{CL}^{best})$, then set $\phi^{now} = \phi^{next}$, $\phi_{CL}^{best} = \phi_{CL}^{next}$, and go to step (4)

(3.2) If ϕ_{CL}^{next} is not tabu, i.e. $TL[m, n] < 0$, then set $\phi_{CL}^{now} = \phi_{CL}^{next}$, and go to step (4)

(4) Update the tabu list: Set $TL[i, j] = TL[i, j] - 1$, for $1 \le i \le |CL|$ and $1 \le j \le |CL|$; Generate h randomly, set $TL[m, n] = h$.

(5) If iterations since previous best solution > $0.2 \times |CL|$, then perform diversification: let $TL[m, n] = \min\{ TL[i, j] \mid 1 \le i \le |CL| \text{ and } 1 \le j \le |CL| \}$, generate initial cluster mapping ϕ_{CL}^{now} by permuting cl_m and cl_n, and set $TL[i, j] = 0$, for $1 \le i \le |CL|$ and $1 \le j \le |CL|$.

(6) If restarts < *max_restart_times*, then go to step (2).

(7) Return ϕ_{CL}^{best}.

Obviously, the above tabu search algorithm can be applied to IP mapping problem of 2D mesh or other type of NoC topology well.

3.3 IP Mapping in the Sub-networks

In the third phase, after every cluster has been mapped to a global router, i.e. a sub-network, then a recursive bi-partitioning algorithm is used to map the IPs within a cluster to the resource nodes of the corresponding sub-network [9].

As a result, the total energy consumption is:

$$E(\phi) = E(\phi_{CL}) + \sum_{i=1}^{|GR|} E(\phi_i) \quad (8)$$

where $E(\phi_i)$ is the communication consumption of the sub-network connected gr_i, and ϕ_i is the optimized mapping function of the IPs mapped onto this sub-network.

4. EXPERIMENTAL RESULTS

The purpose of our experiments is to validate the advantages of our method in runtime and quality of solutions. In this section, we present the results obtained by the execution of our technique on the benchmark applications with different size and characteristics. We have also implemented the tabu search based IP mapping without clustering as the base line. All algorithms were implemented in C++, running on a notebook with a 2.5 GHz processor, and 2 GB main memory.

10 applications of 36, 64, 100, 144, 196 IPs were generated by TGFF [15], mapped onto NoC of 3×3, 4×4, 5×5, 6×6, and 7×7 mesh main network, respectively. Each sub-network is based on binary tree topology, including 4 resource nodes and 3 local routers.

We compare the quality of the solution of two algorithms:

(1) **Cluster & Mapping.** This is our proposed three-phase algorithm combined clustering and tabu search based clusters mapping.

(2) **IP Mapping.** This is the tabu search based "IP" mapping algorithm as illustrated in section 3.2. The input of this implementation is $CTG(C, A)$, and the output of IP mapping ϕ is $E(\phi)$.

Fig. 3 shows how our algorithm performs as the system size scales up. For each problem, IP Mapping algorithm has been set to run same time as Cluster & Mapping. In Fig.3, the energy of Cluster & Mapping is normalized against IP Mapping. As we can see, our proposed method produces better solutions, especially for large size problem. In the best case, it saves 27% energy over IP mapping.

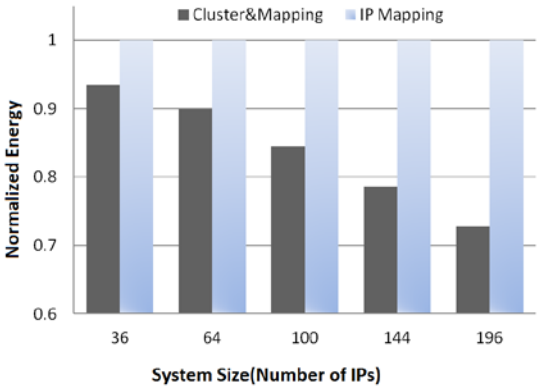

Fig.3: Comparison of Energy Optimization

Table 1 shows the computational time ratio of the two algorithms for obtaining the almost same energy optimization results. As Table 1shown, our method takes less time than IP Mapping. When the system size is 196, the runtime ratio of IP Mapping to Cluster & Mapping is almost 50.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 1: Comparison of Computation Time Ratio

System Size (Number of IPs)	Runtime of IP Mapping / Runtime of Cluster & Mapping
36	1.7
64	3.9
100	9.1
144	21.3
196	49.6

5. CONCLUSIONS

In this paper, a computationally efficient three-phase algorithm for energy-aware mapping IPs onto hierarchical two-level NoC architecture has been presented. The algorithm combined IP cores clustering and tabu search based clusters mapping. The effectiveness of this algorithm is evaluated by comparing with pure tabu search based "IP" mapping algorithm. Experimental results have shown, our method obtains high quality of solutions, but requires significantly less computational time.

ACKNOWLEDGMENT

This work was supported by: the Natural Science Foundation of China (No. 61003032, 61100118), and Artificial Intelligence Key Laboratory of Sichuan Province of China (No. 2010RY010, 2011RYJ05).

REFRENCES:

[1] T. Bjerregaard and S. Mahadevan, "A survey of research and practices of Network-on-chip," *ACM Computing Surveys*, vol. 38, No.2, 2006, pp. 1-51.

[2] U. Y. Ogras, J. Hu, and R. Marculescu, "Key research problems in NoC design: a holistic perspective," *Proceedings of the 3rd IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis*, 2005, pp. 69-74.

[3] J. Hu and R. Marculescu, "Energy-aware mapping for tile-based NoC architectures under performance constraints," *Proceedings of the 2003 conference on Asia South Pacific design automation*, 2003, pp. 233-239.

[4] S. Murali and G. De Micheli, "Bandwidth-Constrained Mapping of Cores onto NoC Architectures," *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition*, 2004, pp. 896 - 901.

[5] G. Ascia, V. Catania, and M. Palesi, "Multi-objective mapping for mesh-based NoC architectures," *Proceedings of the 2nd IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis*, 2004, pp. 182-187.

[6] R. Chae-Eun, J. Han-You, and H. Soonhoi, "Many-to-many core-switch mapping in 2-D mesh NoC architectures," *Proceedings of IEEE International Conference on Computer Design*, 2004, pp. 438-443.

[7] P. K. Sahu and S. Chattopadhyay, "A Survey on Application Mapping Strategies for Network-on-Chip Design," *Journal of Systems Architecture*, vol. 59, No.1, 2013, pp. 60-76.

[8] T. Hollstein and M. Glesner, "Advanced hardware/software co-design on reconfigurable network-on-chip based hyper-platforms," *Computers & Electrical Engineering*, vol. 33, No.4, 2007, pp. 310-319,.

[9] Z. Chang, G. Xiong, and N. Sang, "Energy-aware Mapping for Tree-based NoC Architectures by Recursive Bipartitioning," *Proceedings of International Conference on Embedded Software and Systems*, 2008, pp. 105-109.

[10] P. Guerrier and A. Greiner, "A generic architecture for on-chip packet-switched interconnections," *Proceedings of the conference on Design, automation and test in Europe*, 2000, pp. 250-256.

[11] Z. Lu, L. Xia, and A. Jantsch, "Cluster-based simulated annealing for mapping cores onto 2D mesh networks on chip," *Proceedings of 11th IEEE Workshop on Design and Diagnostics of Electronic Circuits and Systems*, 2008, pp. 1-6.

[12] F. Glover and M. Laguna, *Tabu search*: Springer, 1997.

[13] P. Eles, Z. Peng, K. Kuchcinski, and A. Doboli, "System Level Hardware/Software Partitioning Based on Simulated Annealing and Tabu Search," *Design Automation for Embedded Systems*, vol. 2, pp. 5-32, 1997.

[14] O. Hajji, S. Brisset, and P. Brochet, "Comparing stochastic optimization methods used in electrical engineering," *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, 2002.

[15] R. P. Dick, D. L. Rhodes, and W. Wolf, "TGFF: task graphs for free," *Proceedings of the 6th international workshop on Hardware/software codesign*, 1998, pp. 97-101.