Evaluation of Application Mapping for Network-on-Chips

M. Jalal, Z. Shirmohammadi, A. Patooghy, S. G. Miremadi
Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
mjalal@ce.sharif.edu, shirmohammadi@kish.sharif.edu, patooghy@ce.sharif.edu, miremadi@sharif.edu

Abstract– **Mapping of tasks on the cores of a Network-on-Chip (NoC) has direct impact on the efficiency of the network. This paper provides a comprehensive study regarding application mapping for NoCs to clarify their pros and cons. The study considers different aspects including performance, power consumption, and reliability of mappings. Four mappings named NMAP, RMAP, Random, and Adhoc are used in this study by the means of a cycle accurate NoC simulator. Our study shows that the RMAP provides the maximum reliability for NoC with a low performance overhead. On the other hand, Random mapping requires the lowest time to complete mapping of the task on the chip. The power estimation patch of Orion is used in the simulation to explore the power consumption of a typical NoC using any of the mentioned mappings. Simulations are done for three different benchmarks i.e., MPEG, VOPD, and OPD application graphs.**

I. INTRODUCTION

Network-On-Chip (NoC) paradigm was proposed to solve the problems of flexibility, scalability and reusability of traditional buses in System-on-chips (SoCs) [1]. NoC nodes which can be specific-proposed processors, DSPs, memories, or I/O blocks are arranged beside each other according to the specific topology. The embedded router in each node is responsible to establish the packet-based communication with the adjacent nodes. Since NoCs are fabricated in nano-scale VLSI technologies, reliability and power consumption are among the main concerns [2][3].

One important issue that affects the main concerns of NoC design is application mapping [4][5][6]. Using the application mapping, tasks of an application which should be run on the chip, are placed on appropriate nodes of NoC. In this regard the target application is defined by a task graph. A task graph is a directive acyclic graph $(V, E \subset V \times V)$ where each vertex V_i corresponds to a task $\tau_i$and each edge $(V_i ; V_j) \subset E$ models a data dependency between tasks $\tau_i$andτ_j. Application mapping can be done with respect to different considerations like power consumption, reliability, performance, bandwidth constraint of channels, and etc [5][7]. Static mappings do not consider the runtime behavior and tasks are scheduled before the run time [8]. In contrast, in dynamic mapping different configurations can be applied according to the variations of the system during the working condition [8]. Based on the nature of mapping problem, as much as the task graph grows, the search space increases. Consequently, application mappings do their best to effectively use the NoC resources and meet the NoC requirements [4][9][10].

A multi-objective mapping for the mash-based NoCs based on evolutionary computing is proposed to optimize the power and performance of the NoC [5]. This mapping searches all the possible solutions and the optimal answer is selected by the use of trace-based simulator. Genetic algorithms are also used in application mapping [7] to minimize the execution time of the application. Solving the problem of the mapping for regular NoCs has been proposed in [11] by a communication dependency and computation model. The model simultaneously considers both the execution time and the energy consumption in the NoC. In [4] a mapping based on a branch and bound algorithm for mesh based NoC architecture is proposed. This mapping provides a way to reduce the data transmission power while satisfying the performance constraints. The communication delay is minimized in [12] with a heuristic approach. This is done by exploiting the possibility of splitting the traffic among different paths in network on chips. NMAP algorithm proposed in [6] considers the mapping problem as a shortest path problem while the BMAP algorithm uses greedy binomial and optimization method for reducing the time and needed hardware. In [6], transient fault prevention during the mapping is provided. This is done by adding redundant communications that increase the probability of correct packet delivery in the network. Although authors of [10] tried to improve the reliability in mapping process, additional channels cause the congestion in the network channels. Also some kinds of error exist that can occur in other parts on the NoCs like single and multiple event upsets that should be considered for reliability improvements in the network. Authors in [9] have tried to solve the mentioned drawbacks. In this work a reliability-aware application mapping called RMAP is proposed that considers transient faults occurring both in channels and switches of NoCs with respect to the traffic distribution over the network channel.

In spite of several application mapping algorithms proposed in the literature, to best of our knowledge there is no work that study the efficiency of these algorithms. Such a study helps NoC designers to adopt an appropriate application mapping for their designs. This paper provides a comprehensive study regarding application mapping for NoCs to investigate their merits and demerits. The study considers different aspects including performance, power consumption, and reliability of mappings. Four mappings of NMAP, RMAP, Random, and Adhoc are used in the study by the means of a cycle accurate NoC simulator. Mappings are evaluated with three different benchmarks i.e., MPEG, VOPD, and OPD application graphs.

The rest of this paper is organized as follow. An overview on Network-on-Chips is presented in Section II. Section III discusses the most important application mapping for NoCs. Evaluations of these mappings are presented in IV and finally conclusion remarks are given in Section V.

II. BACKGROUND

This section elaborates required backgrounds for NoCs including the implementation details of a typical NoC router, switching and routing algorithm. These are important since most of NoC reliability improvement methods (including reliability improvement mapping) use the hardware features of NoC routers to reach their aims. In addition this section discusses the reliability improvement methods for NoCs.

NoC Overview

Each node in an NoC is composed of a switching element and a processing element. Processing element performs the processing while the switching element provides a reliable, high bandwidth connection between the processing element and other nodes of the network. Nodes of an NoC are arranged and communicate with each other according to a predefined structure called topology such as mesh topology or torus topology. Several researches have been focused on mesh topology for NoCs due to its simplicity in fabrication process and good electrical properties [13].

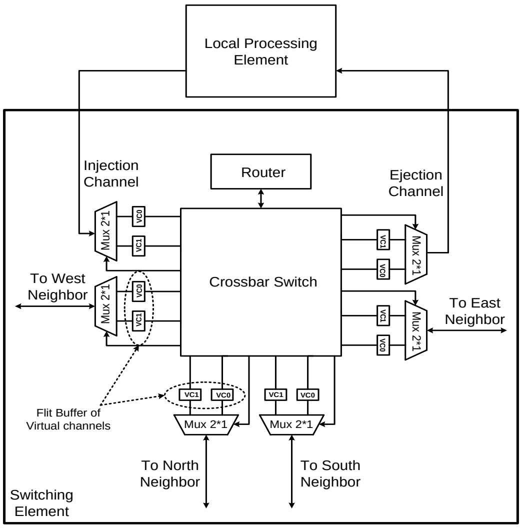


Figure 1. A mesh based NoC node architecture.

Switching method determines how a packet of data should be transmitted through the network. As an example in wormhole switching which is the most prevalent switching method for NoCs [11], the packets are divided into a sequence of fixed-size units, called flits. Flits of a packet follow the header flit of the packet in pipeline manner. In this way, flits of a packet form a worm in the network. Routing algorithm determines the channels which a packet should pass to reach its destination. Dimension order (so called XY routing) is one of the widely used routing algorithms for NoCs. XY routing has two phases, in the first phase packets are routed until the packet reaches the column of its destination. The second phase routes the packet in the Y dimension to reach the destination node [14].

The switching element of a typical NoC router is depicted in figure 1. This router consists of 5 physical channels to communicate with its north, south, east and west neighbors, and the processing element. This router can be used in 2 dimensional mesh NoCs. As shown in the figure, physical channel of the router is timely multiplexed between some buffers, namely virtual channels. Virtual channels improve the performance of the network.

Reliability in NoCs

Reliability improvement, have been the subject of many research groups up to now. Work done in this regard can be classified into error control coding methods, flood-based routing algorithms, and fault tolerant routing algorithm.

The main idea behind the error control methods is adding information redundancy to the packets/flits [15]. The added information redundancy helps to protect packets/flits during the transmission in NoC. Error control methods can be applied in end-to-end or switch-to-switch types. In the switch-to-switch methods the redundancy is added to each flit and is checked in each switch. The main advantage of this type of error control is its low error detection latency. However, frequent flit checking imposes high power overhead in switch-to-switch methods. The end-to-end methods apply the redundancy in the packet level and the integrity checking is done only by the destination node. End-to-end methods impose rather low power overhead while the detection latency is high. A common disadvantage in both types of error control methods is their variable power consumption and extra buffer overhead.

In the flood-based routing algorithms, as its name shows, huge numbers of the redundant packets are sent toward destination node [2]. Since redundant packets pass different paths, they have different error probabilities. In this way, the probability of having a correct packet at the destination node increases. In flood-based algorithms whenever a node receives a new packet, it sends the packet to all of its adjacent nodes. In directed-flooding algorithm, adjacent nodes are selected with a predefined probability i.e., the packet is sent to each node with the probability p and is not sent with the probability of $1-p$[16]. Main drawback of flood-based algorithms is high power consumption and performance overheads due to the huge amount of redundant packets. In addition extra hardware is needed for deadlock handling and producing random number generator.

In addition to the mentioned methods, reliability improvement can be achieved by the aim of application mapping. As an example, application mapping adds redundant tasks to the application graph, or redundant communication between tasks to improve the reliability of NoC.

In [10], the transient errors fault prevention during the mapping is provided. This is done by adding redundant communications that increase the probability of correct packet delivery in the network. Although authors in [10] tried to improve the reliability in mapping process, additional channels cause the congestion in the network, because the distribution of traffic is not considered in this work. Authors in [9] have proposed a reliability-aware application mapping for mesh-based NoCs. In the next section we describe RMAP and NMAP in details which are used in our experimental evaluations.

III. APPLICATION MAPPING FOR NOCs

As mentioned application mapping directly influences performance, power consumption and area requirement of NoCs [4][5][6]. Using the application mapping, tasks of an application which should be run on the chip, are placed on appropriate nodes of NoC. In this regard the target application is defined by a task graph. A task graph is a directive acyclic graph $(\mathrm{V}, \mathrm{E} \subset \mathrm{V} \times \mathrm{V})$ where each vertex V_i corresponds to a task $\tau_\mathrm{i}$and each edge $(\mathrm{V}_\mathrm{i} ; \mathrm{V}_\mathrm{j}) \subset \mathrm{E}$ models a data dependency between tasks $\tau_\mathrm{i}$andτ_j. Figure 2.A shows the application graph of the VOPD application. As it can be seen in this figure, the VOPD application has 16 tasks which communicate with each other according to what represented in this figure. Application mapping can be done with respect to different considerations like power consumption, performance, bandwidth constraint of channels [5][7]. To best of our knowledge, a few of previously proposed application mapping consider the reliability NoCs. RMAP [9] is one of those mappings, which is described in the following sub-section.

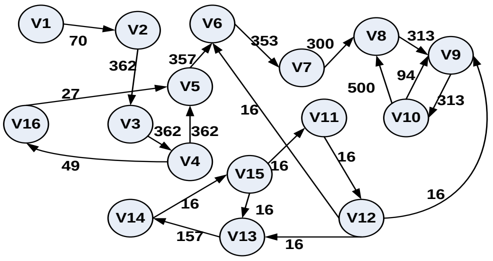


Figure 2.A sample application graph, nodes represent tasks and links represent communication rate between the tasks.

RMAP

RMAP [9] is a reliable application mapping for mesh based NoC architecture. This mapping considers transient faults occurring in both channels and switches of network on chip. This algorithm divides the application graph into two sub-graphs to minimize the communication traffic between the sub-graphs and to minimize the communication traffic in each sub-graph. Then, one of the sub-graphs is mapped onto the upper triangular nodes of the NoC and the other sub-graph is mapped onto the lower triangular nodes of the NoC. This results in lower traffic loads in north/south channels which are efficiently used to route packets of redundant communications. Performance and power consumption overheads of the redundant communications are minimized by minimizing the total number of packets × hops in the proposed mapping.

To more precisely study the RMAP mapping algorithm, we need the following definitions. *Topology graph* models the topology of the network, in TG =G (N, L) where N is the set of networks node and L is set of channels. The channel$\mathrm{l}_{\mathrm{i,j}} \in \mathrm{L}$ connects $\mathrm{n}_\mathrm{i} \in \mathrm{N}$ to$\mathrm{n}_\mathrm{j} \in \mathrm{N}$. Also the application graph TG =G (T, R) is directed-weight graph where T is set of application graph and R is set of application between nodes. When task r_i wants to send data to r_j it can be shown as r_{ij} while the rate of communication is $\mathrm{weight}(\mathrm{r}_{\mathrm{i,j}})$. The application graph is equipped with additional $\mathrm{r'}_{\mathrm{i,j}}$ communication for each$\mathrm{r}_{\mathrm{ij}}$to improve the reliability in the network. The weight of these two communications is considered the same. Transient faults in the network are tolerated by adding one parity bit to the packets in the source node. When the packet is received in the destination node, the parity is produced again and compared with the source node parity. If they are not the same, the corresponding flits in the redundant channel can be used. RMAP uses the XY routing algorithm to route main packets and YX routing algorithm to route redundant packets. This causes the traffic to distribute uniformly in the networks channels. Mapping in this paper can be considered as finding the one-to-one function RMAP() with a given topology and application graphs such that:

$$RMAP: T \longrightarrow N, RMAP(T_i) = n_j , t \in T, n \in N$$

Authors defined RMAP for $\|\mathrm{T}\| = \|\mathrm{N}\| = m^2$ tasks such that the following cost function is minimized:

$$COST = \sum_{for\ all\ r_{i,j} \in R} weight(r_{i,j}) \times distance_{XY}[RMAP(t_i), RMAP(t_i)] + \sum_{for\ all\ r'_{i,j} \in R} weight(r'_{i,}\ ) \times distance_{XY}[RMAP(t_i), RMAP(t_i)]$$

where $\mathrm{distance}_{\mathrm{XY}}(\mathrm{n}_\mathrm{i}, \mathrm{n}_\mathrm{j})$ shows the needed hops to deliver the packets from node n_i to n_j with XY algorithm.

NMAP

Mulari et al. have introduced a fast NoC mapping algorithm which splits down the traffic and maps the cores onto NoC. NMAP presented in[12] is a bandwidth constrained mapping which minimizes the average delay by iterations. NMAP solve the mapping problem as a shortest-path optimization problem. The cost of NMAP is beyond $O(N^4 \times \log N)$ where N is the number of IPs. NMAP algorithm consists of two phase: First phase is finding an initial answer and the second phase performs iterations to optimize the answer.

In the first phase an initial mapping is computed and the most communicating core is mapped to a node in NoC with most number of neighbours and the mapping is iteratively refined in the second phase. Recent mappings are mostly an extended version of NMAP with improving the mapping algorithm or combining some other cost factor to make the mapping problem a multi-objective one.

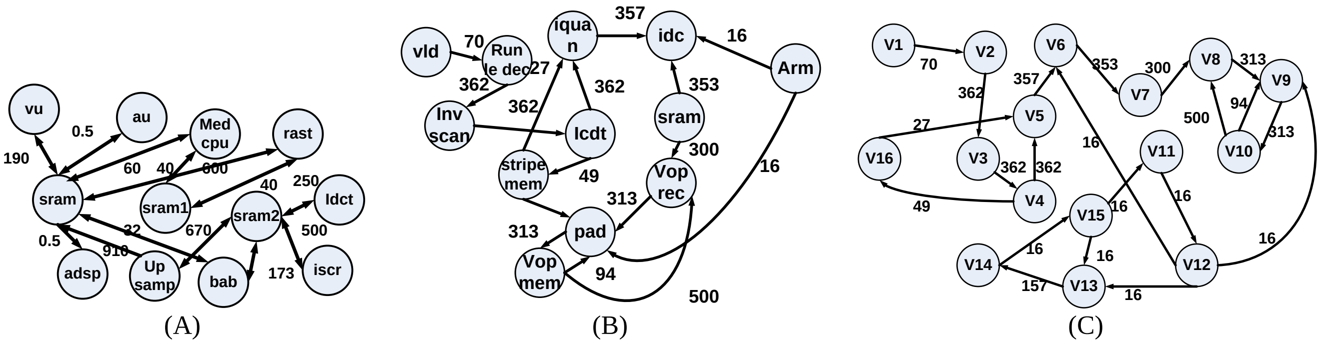


Figure 3. Application graphs of VODP (A), MPEG Decoder (B) and VOD (C) mapped on the NoC by RMAP, NMAP, Random, and Adhoc mappings.

IV. EVALUATION OF APPLICATION MAPPING FOR NOCs

Simulation Platform

In order to evaluate the performance, power consumption and the reliability characteristics of RMAP and NMAP we used an event-based NoC simulator, called Xmulator [17][18]. Xmulator is a listener-based integrated simulation platform for NoC. This simulator is a fully-parameterized flit-level simulator providing detailed results after each simulation run. In addition, the Orion power library [19] is imported to Xmulator for calculating power consumption. Simulations are performed for 4×3 and 4×4 networks respecting their task graphs with the flit width of 128-bit and packet length of 8 flits.

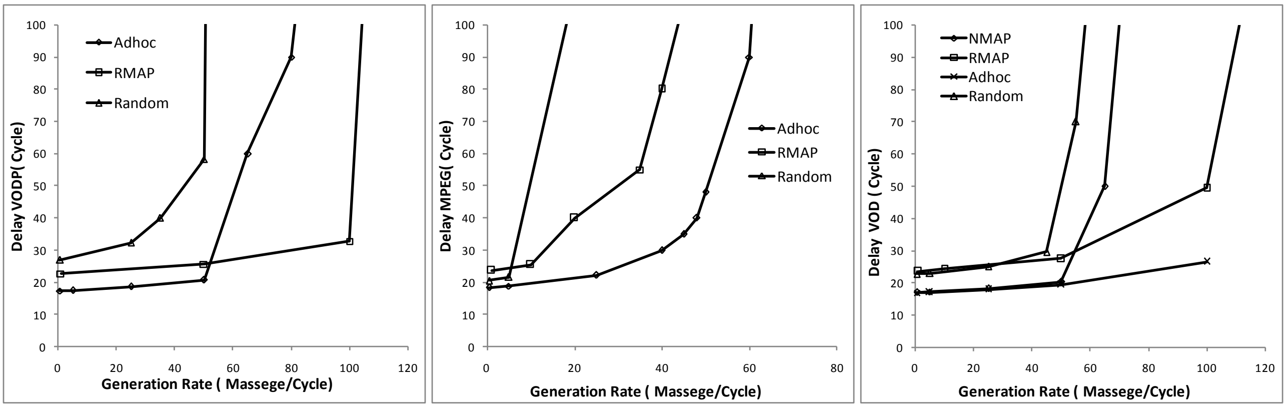
In addition to RMAP and NMAP which are described above, we used a Random mapping as well as an Adhoc mapping in our simulations. Adhoc and Random mappings are used because of their time overhead and their efficiency. In Random mapping, tasks are mapped onto cores through use of two random number generators. Using the first random number generator, one of the remained tasks is chosen, while the second random number generator determines the core onto which the task should be mapped. Obviously Random mapping requires the lowest time to place tasks of an application graph onto an NoC. Note that the Random mapping does not consider the performance of the NoC in the mapping process i.e., a high performance penalty which is imposed on NoC by Random mapping. In contrast, Adhoc mapping is heuristically done by NoC designers to minimize the performance overhead of application mapping. This way requires an NoC designer and the knowledge about the used routing and switching algorithms in the NoC. The designer places the task in a way minimizing the average packet transmission time. Evidently this way imposes a high design cost and high design time. Since Adhoc and Random mappings have completely opposite characteristics i.e., these methods are extremes of performance overhead and design cost, we used them in our comparisons. Such a comparison clarifies the pros and cons of RMAP and NMAP methods.

Simulation Results

For performance evaluation of each mapping, three task graphs named MPEG-4, VOPD and OPD which are shown in Figure 3 have been used. The first two task graphs were mapped onto a 4×3 NoC while the last one was mapped onto a 4×4 NoC. SEUs (single bit flips) are the greatest concern threatening NoC buffers, hence we injected SEUs into our network so as to evaluate different mapping of different task graphs. Error injection rate was variable between 10% and 70%. We extracted ratio of correctly delivered packets to whole packets in order to estimate reliability of each mapping. Each of the task graphs were mapped using the entire mentioned mapping algorithm first with XY routing and then with XYX routing.

In order to see the performance behaviour of the different mapping, we used the traffic rates of each graph multiplied by a constant of λ. Using several values for λ we extract the performance and power behaviour of the mapping in variable traffic conditions. As it is depicted in Figure 4 increasing the generation rate increases performance until reaching the saturation point in which no further increase in generation rate will lead to performance improvement. Due to use of low-overhead channels, networks using RMAP have a greater saturation point hence have better performance regarding other mapping algorithms used. As it can be observed from Figure 4 the performance overhead of RMAP algorithm is really negligible. Although it should be noted that the power overhead of RMAP, Figure 5, is somewhat considerable compared to other mapping algorithms owing to its redundant packets sent throughout the network.

Illustrated in Figure 6 ratio of detected flits to whole flits traversed the network. As is can be seen in this figure, RMAP is higher than other mapping owing to the way cores are mapped in the network and duration of their existence in the network. Also it is shown that increasing the generation rate has an insignificant effect on the detection ratio of each network because both correctly delivered packets and whole packets increase correspondingly. Simulations depicted in Figure 6 are done under the error injection rate of 40% i.e., 40% of all flits traversing the NoC are injected by single bit flip error.



(A) (B) (C)

Figure 4. Average packet delivery time for 4×4 and 3×4 NoC when application graphs of VODP (A), MPEG Decoder (B) and VOD (C) are mapped on the NoC.

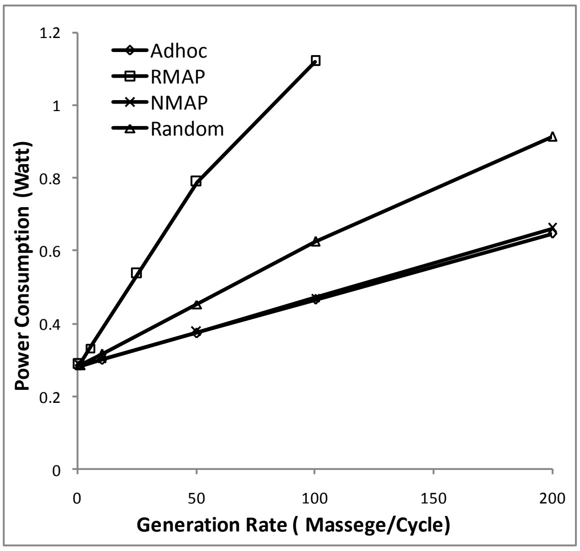


Figure 5. Average power consumption for evaluated mappings in different traffic rates.

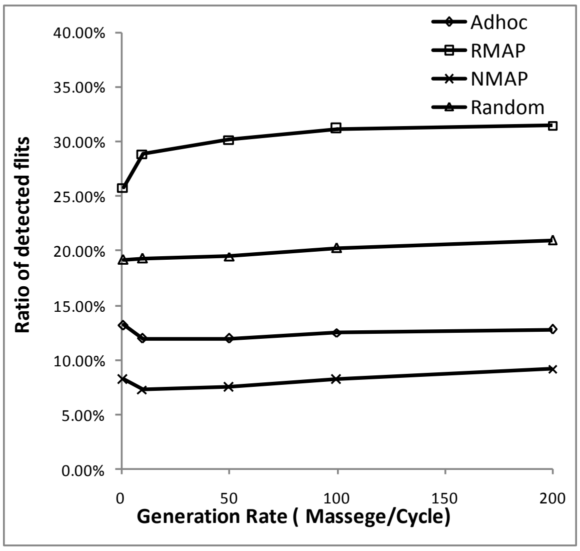


Figure 6. Average ratio of detected faults for evaluated mappings under the error injection rate of 40%.

V. CONCLUSIONS

This paper compared the efficiency of widely used application mapping for NoCs. The comparison is done in terms of reliability, power consumption, and performance of the mesh-based NoC as well as the required time for the application mapping. Adhoc and Random mappings are also considered in the simulations. Comparisons show that the RMAP improves the reliability of NoC while its power consumption and performance overheads are low. This is because the RMAP uses low-traffic channels to send redundant packet. In this way, channels which are overloaded by original traffic are hardly used by the redundant traffic and vice versa. Adhoc mapping provides the best in terms of network performance and needs the high in terms of mapping time.

REFRENCES

[1] M. Ali, M. Welzl, and M. Zwicknagl, “Networks on chips: scalable interconnects for future systems on chips.”

[2] D. Park, C. Nicopoulos, J. Kim, N. Vijaykrishnan, and C.R. Das, “Exploring fault-tolerant network-on-chip architectures,” International Conference on Dependable Systems and Networks, 2006. DSN 2006, 2006, pp. 93–104.

[3] N. Banerjee, P. Vellanki, and K.S. Chatha, “A power and performance model for network-on-chip architectures,” Design, Automation and Test in Europe Conference and Exhibition, 2004. Proceedings, 2004.

[4] J. Hu and R. Marculescu, “Energy-aware mapping for tile-based NoC architectures under performance constraints,” Design Automation Conference, 2003. Proceedings of the ASP-DAC 2003. Asia and South Pacific, 2003, pp. 233–239.

[5] G. Ascia, V. Catania, and M. Palesi, “Multi-objective mapping for mesh-based NoC architectures,” International Conference on Hardware/Software Codesign and System Synthesis, 2004. CODES+ ISSS 2004, 2004, pp. 182–187.

[6] W.T. Shen, C.H. Chao, Y.K. Lien, and A.Y. Wu, “A New Binomial Mapping and Optimization Algorithm for Reduced-Complexity Mesh-Based On-Chip Network,” 2007.

[7] T. Lei and S. Kumar, “A two-step genetic algorithm for mapping task graphs to a network on chip architecture,” 2003.

[8] S. Kumar, A. Jantsch, J.P. Soininen, M. Forsell, M. Millberg, J. Oberg, K. Tiensyrja, and A. Hemani, “A network on chip architecture and design methodology,” IEEE Symposium on VLSI, 2002, pp. 117–124.

[9] A. Patooghy, H. Tabkhi, and S.G. Miremadi, “RMAP: A Reliability-Aware Application Mapping for Network-on-Chips.”

[10] S. Manolache, P. Eles, and Z. Peng, “Fault and energy-aware communication mapping with guaranteed latency for applications implemented on NoC,” Proceedings of the 42nd annual Design Automation Conference, 2005, p. 269.

[11] J. Duato, S. Yalamanchili, and L.M. Ni, Interconnection networks: An engineering approach, Morgan Kaufmann, 2003.

[12] S. Murali and G. De Micheli, “Bandwidth-constrained mapping of cores onto NoC architectures,” 2004.

[13] W.J. Dally and B. Towles, “Route packets, not wires: on-chip interconnection networks,” Design Automation Conference, 2001. Proceedings, 2001, pp. 684–689.

[14] T. Bjerregaard and S. Mahadevan, “A survey of research and practices of network-on-chip,” ACM Computing Surveys (CSUR), vol. 38, 2006, p. 1.

[15] D. Bertozzi, L. Benini, and G. De Micheli, “Low power error resilient encoding for on-chip data buses,” Proceedings of the conference on Design, automation and test in Europe, 2002, p. 102.

[16] M. Pirretti, G.M. Link, R.R. Brooks, N. Vijaykrishnan, M. Kandemir, and M.J. Irwin, “Fault tolerant algorithms for network-on-chip interconnect,” 2004.

[17] A. Nayebi, S. Meraji, A. Shamaei, and H. Sarbazi-Azad, “Xmulator: A listener-based integrated simulation platform for interconnection networks,” First Asia International Conference on Modelling & Simulation, 2007. AMS'07, 2007, pp. 128–132.

[18] A. Nayebi, S. Meraji, A. Shamaei, and H. Sarbazi-Azad, “XMulator: an object oriented XML-Based Simulator,” Proceedings of the First Asia International Conference on Modelling & Simulation, 2007, pp. 128–132.

[19] H.S. Wang, X. Zhu, L.S. Peh, and S. Malik, “Orion: a power-performance simulator for interconnection networks,” 35th Annual IEEE/ACM International Symposium on Microarchitecture, 2002.(MICRO-35). Proceedings, 2002, pp. 294–305.