

A Topology Design Customization Approach for STNoC

Gianluca Palermo^{1,2} Giovanni Mariani² Cristina Silvano¹ Riccardo Locatelli³ Marcello Coppola³

¹Politecnico di Milano - Dipartimento di Elettronica e Informazione

{gpalermo,silvano}@elet.polimi.it

²ALaRI - Faculty of Informatics - University of Lugano

giovanni.mariani@lu.unisi.ch

³STMicroelectronics - Advanced System Technology

{riccardo.locatelli,marcello.coppola}@st.com

ABSTRACT

To support high bandwidth SoCs, a communication design flow is necessary for the design space exploration respecting tight design requirements. In order to exploit the benefits introduced by the NoC approach for the on-chip communication, the paper presents a design flow for the core mapping and customization of the network topology applied to STNoC, the Network on-Chip developed by STMicroelectronics. Starting from ring topology, the proposed application-specific flow tries to find a set of customized topologies, optimized in terms of performance and area/energy overhead, by adding links. The generated STNoC custom topologies provide a reduced cost with respect to the spidergon topology.

1. INTRODUCTION

The scalability and the success of switch-based networks and packet-based communication in parallel computing have inspired the researchers to propose the Network-on-Chip (NoC) architecture [1] as a viable solution to complex on-chip communication problems. Although sharing some similarities (e.g. topology, packetized routing) with traditional networks in multicomputers [2] (sometimes called *macro-networks*), the NoC communication paradigm has specific properties which differentiate it from macro-networks.

Design-time specialization is an important factor for the NoC paradigm. In fact, most NoC architectures are specifically developed either for one embedded application or as a platform for a small class of applications and consequently, the traffic characteristics cover an important role for the network customization.

Designing a NoC architecture, the first step to be done is the topology selection. Due to the regularity characteristics and the success in macro-networks, standard topologies (such as 2D mesh and torus) have been selected as basis for the on-chip network infrastructure. However, in the on-chip domain, the connectivity theoretically offered by standard topologies cannot be fully exploited due to the nature of communication traffic in real embedded applications. In other word, we can pay in area and power overhead more than the performance benefits that the topology can offer. In fact, in certain application scenarios, simple topologies such as ring can show a better area/energy/performance trade-off with re-

spect to more complex standard topologies. To further optimize area/energy/performance, an approach based on the customization of simple standard topologies can be much more effective.

In this direction, the proposed approach starts from a simple topology (such as ring) for the application-specific topology customization process to boost the network performance without requiring the area and energy overhead of more complex networks. In particular, this paper presents an application-specific design flow for the topology customization of STNoC [3, 4], the Network on-Chip developed by STMicroelectronics. Starting from ring topology, the design technique tries to find the optimal topology in the STNoC family of topologies (ranging from ring to spidergon) by adding a set of custom links to optimize the network performance on the target application.

The paper is organized as follows. Section 2 reviews some related works and presents the context where the work takes place. While in Section 3 and 4 are respectively presented the theory on communication metrics to evaluate NoCs and the STNoC architecture, Section 5 describes the proposed flow. The experimental results and some considerations on the STNoC topology optimization are shown in Section 6. Finally, Section 7 concludes the paper summarizing some future trends of our research.

2. PREVIOUS WORK

Designing application specific Network on-Chip has recently become a must for many academics and industrial research groups.

Although many works have been presented in the past as general evaluations of NoC architectures using random stimuli [5–8], recently the design of application specific NoCs is the focus of the research in the on-chip communication field [9–16].

In [9–13] the problem of an application-specific mapping of cores onto the Network tiles is presented. This problem has been widely explored and different approaches have been proposed. In [9] and [10] two heuristic approaches derived from *branch and bound* and *greedy* techniques have been presented, while in [11, 12] the mapping problem has been modeled by using a genetic approach. In [13], the authors extend the mapping problem to multi-use-cases application on set-top boxes and TV-SoC design.

More structural approaches for the application specific customization of the on-chip communication have been presented in [14–16]. In [14] the authors present a methodology for the customization of the STBus crossbar. In [15, 16] the target of the customization is the topology of the network, using a decomposition approach for a fully customization [16] and the insertion of long links to the 2D mesh standard topology [15].

In this paper, with respect to the previous research, we present a methodology for mapping and topology customization for NoC applied to an industrial case study, STNoC by STMicroelectronics. In particular, the proposed approach tries to find the best applica-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

tion specific topology for the STNoC family of topologies ranging from ring to spidergon.

3. THEORY ON COMMUNICATION METRICS FOR NOC

Before starting the description of the design flow used for the STNoC topology customization, let us introduce some theoretical concepts. First of all, we define two main graphs that are the object of our discussions: the core graph, used to describe the target application, and the topology graph.

The *core graph* (also called application graph or communication graph) is a directed graph, $G(V, E)$ with each vertex $v_i \in V$ representing a core and the directed edge (v_i, v_j) , denoted as $e_{i,j} \in E$, representing the communication between the cores v_i and v_j . The weight of the edge $e_{i,j}$ represents the bandwidth of the communication from v_i and v_j .

The NoC *topology graph* is a directed graph $P(U, F)$ with each vertex $u_i \in U$ representing a node in the topology and the directed edge (u_i, u_j) , denoted as $f_{i,j} \in F$, representing a direct communication between the vertices u_i and u_j . The weight of the edge $f_{i,j}$, denoted by $bw_{i,j}$, represents the bandwidth available across the edge $f_{i,j}$.

Given a NoC topology graph $P(U, F)$, we define as *NoC Aggregate Bandwidth* a cost function AB that is based on the overall communication bandwidth allocated in the selected NoC topology. More precisely: $AB = \sum_{k=1}^{|F|} bw(link_k)$ where $bw(link_k)$ returns the bandwidth allocated to the link k and $|F|$ is the number of all the available connections $f_{i,j}$ described before. The values of the AB cost function reflects the dynamic behavior of the system. In fact, small values of the cost function mean that, on average, nodes in the *core graph* with high communication edges are placed close to each other (reducing the topological delay of the communication) and it is reduced the probability of contentions in the access of a shared resource (in this case, a link of the network).

In this direction, two metrics representing the contention in the utilization of a shared link are:

- *Average link utilization* (LU_{av}) that gives a global view of the network traffic: $LU_{av} = \frac{AB}{|F|}$
- *Maximum link utilization* (LU_{max}) that gives information on the worst case: $LU_{max} = Max_k\{bw(link_k)\}$

The introduced metrics have been used in the proposed design flow to evaluate the effectiveness of the solutions.

4. THE STNOC ARCHITECTURE

STNoC [3] is a flexible and scalable packet-based on-chip micro-network designed according to a layered methodology for the communication design. The most important feature of the spidergon [4] architecture is that the conceptual simplicity of the topology also translates into the most cost-effective silicon implementation of the key components: routers and network interfaces. In the spidergon topology all the IP blocks are arranged in a ring where each IP block is connected to its clockwise and its counter-clockwise neighbor as in a simple ring topology. In addition, each IP block is also connected directly to its diagonal counterpart in the network, which allows the routing algorithm to minimize the number of nodes that a data packet has to traverse before reaching its destination. One of the particular added value of spidergon with respect to previous topologies is the ability to provide the proper cost/performance trade-off in the NoC domain. For example, topologies such as 2D-mesh (that theoretically provide high communication speeds) are expensive to implement in silicon because of their large number of router ports and connections. The ability to downsize the architecture by removing unused components and links depending on

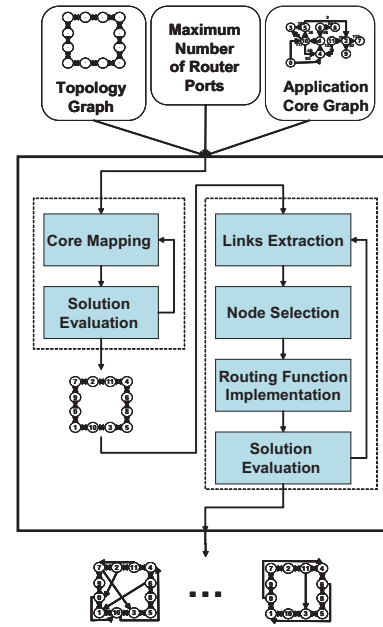


Figure 1: Mapping and Topology Customization Design Flow.

the application traffic allows STNoC to actually support a range of topologies, from tree, to simple ring and up to spidergon topology.

5. PROPOSED DESIGN FLOW

Although spidergon offers good characteristics in terms of network diameter, connectivity and regularity, a customized topology could better satisfy the requirements imposed by the target embedded application. Since the complexity of on-chip interconnect could not be increased too much and a fully customized topology would not be feasible in a reasonable design time, we start from the simple ring topology by adding customized unidirectional links up to the maximum given by the spidergon (including one more input and one more output port for each node of the ring topology). In fact, STNoC includes a flexible router architecture with up to 3 input/output ports for intra-router communication. This feature enables to optimize STNoC IP-blocks performance, improve their maintainability and reduce their verification effort, while keeping efficient application customization in a reasonable design time.

Figure 1 shows the proposed design flow for the mapping and topology customization phases. The proposed design flow starts receiving as input the application core graph, the initial topology graph (the ring in our case) and the maximum number of ports for the routers in the topology. The design flow is composed of two optimization loops: the first one representing the core mapping loop while the second one representing the topology customization loop. Since both problems cannot be solved using an exhaustive approach in a reasonable design time, two heuristic approaches have been developed in the design flow.

The core mapping loop is iterated a predefined number of times, and it is in turn composed of two simple steps:

- **Core Mapping.** The first step performs the mapping of the application core graph into the basic topology based on a genetic algorithm [12].
- **Solution Evaluation.** Once the mapping has been generated, the solution is evaluated in terms of maximum link utilization and NoC aggregate bandwidth. If the evaluated solution does not respect the maximum link utilization constrain required

by the target NoC architecture, the solution is not inserted in the list of feasible solutions.

In the proposed design flow, from the list of feasible solutions, we select the best mapping configuration in terms of aggregate bandwidth.

Starting from the mapping solution, the second loop performs the topology customization phase by adding custom links without modifying the mapping. This phase is composed of a predefined number of iterations of a body loop in turn composed of four passes:

- *Links Extraction.* This phase extracts the number of unidirectional links to be added at the base topology. The number is extracted randomly in the range from one to N , where N is the number of network nodes (that is also the difference in terms of link number between spidergon and ring).
- *Nodes Selection.* This phase consists of the selection of the nodes to be connected by the additional links. The selection of both the source and destination nodes for each link is based on a roulette wheel algorithm. In the selection phase, a node is not considered if it has already reached the maximum number of connection ports to be added.
- *Routing Function Implementation.* This phase selects the minimal path between each source and destination nodes on the core graph, efficiently utilizing the custom links. The routing function is designed to avoid deadlocks by using only the 2 virtual channels in the external ring. The family of routing functions is static and it is based on tables. To route a packet, one of the main characteristics is that each router needs only to know the packet destination node and not its source node. This means that all the packets that arrive to a router with the same destination are routed to the same output port. This technique reduces the hardware complexity of the router.
- *Solution Evaluation.* Once the customized topology is generated, the aggregate bandwidth, the maximum value of link utilization and the average link utilization metrics are evaluated.

Since the optimization is multiobjective, at the end of the second loop, all the evaluated solutions are filtered in order to select the Pareto solutions in terms of aggregate bandwidth, maximum value of link utilization, average link utilization, and number of added unidirectional links.

Although in this paper the presented design flow has been applied to the STNoC family of topologies, it could be easily applied to the customization of other standard topologies.

6. EXPERIMENTAL RESULTS

In this section, we will show the results obtained by applying the proposed design flow to a set of case studies belonging to two categories:

- *Random Generated:* Two application task graphs have been generated by using Task Graph For Free (TGFF) [17], a general purpose random task graph generator. The two generated graphs are called TGFF12, composed of 12 nodes, and TGFF14, composed of 14 nodes, that are presented in Figures 2(a) and 2(b) respectively.
- *Real applications:* Two real multimedia applications have been extracted from literature [18] and reported in Figures 2(c) and 2(d): VOPD (Video Object Plane Decoder) and MPEG4.

Figure 2 shows the communication core graphs of the target applications where the nodes represent the cores while the edges the communication among the cores (the arrows points to the target core). The weight on each edge represents the communication bandwidth between the two cores.

Figure 3 shows the results obtained by applying the presented design flow to the four target core graphs. Each sub-figure compares a set of twenty customized topologies (labeled as CTn on X-Axis) generated from the ring, used as base topology for the customization, to spidergon. The comparison is in terms of aggregate bandwidth (primary Y-Axis) defined before. The value of the aggregate bandwidth is represented as a bar for each customized topology while two horizontal dashed lines are used to represent the aggregate bandwidth for ring and spidergon (the highest and the lowest line respectively). For both ring and spidergon, the mapping solutions with the lowest value of aggregate bandwidth have been considered. In Figure 3, the polygonal line represents the number of unidirectional links added to the initial ring topology (secondary Y-Axis) for each custom topology.

As shown in Figure 3, the aggregate bandwidth decreases by increasing the number of added links for all the four applications. Except for MPEG4 (Figure 3(d)), a set of customized topologies offer reduced aggregate bandwidth with respect to spidergon (requiring 12 added links for TGFF12, VODP and MPEG4 while 14 added links for TGFF14). In particular, for TGFF12 (see Figure 3(a)) the CT9 configuration, with only 4 additional links, is under the spidergon line showing better performance in terms of average bandwidth. After the CT14 configuration (6 added links) all the custom topologies reduce the aggregate bandwidth values of the network. In Figure 3(b) (TGFF14) the custom topologies with 6 or more added links overtake the spidergon performance (except for CT10). The same behavior for the same number of added links is shown also for VODP in Figure 3(c). For MPEG4 (figure 3(d)) although the average bandwidth decreases with the increment of the added links, the spidergon performance remains better than the performance obtained by the custom topologies. This phenomenon can be motivated by the presence on the core graph of nodes with high connectivity (up to 7 edges in Figure 2(d)). In our opinion, graphs with similar characteristics can be better managed by using topologies with more complex routers (such as spidergon).

Figure 4 shows the STNoC topology comparison among ring, spidergon and two selected custom topologies generated for the TGFF12(a) core graph. The performance of the two custom topologies, labeled as CT7 and CT15, compared with the two standard topologies are shown in Table 1 in terms of area, power consumption, network latency, aggregate bandwidth and maximum and average link utilization.

To derive area, power and network latency results, the generated configurations are given as input to a NoC compiler. Starting from the target configuration, the NoC compiler builds the target networks considering all the routers as three-stages pipeline, with a 4-word buffers depth and 32-bit data width. The NoC compiler generates the synthesizable Verilog RTL model and the simulatable cycle-accurate SystemC model of the network. The generated RTL model of the networks have been synthesized to gate level using the STMicroelectronics HCMOS9 Low Leakage 0.12 μ m technology libraries and power/area values have been estimated by using Synopsys design tools. To compare network performance also with latency values, statistical simulations have been done by using the SystemC model of the networks. The packet injection process has been done by using a Bernoulli process, while the packet size is assumed to be constant with a value of 4 flits. Using this memoryless process, a packet is injected from a node i in the network at each cycle with a probability equal to ρ_i . The values of ρ_i for each node and the destination of the packet are generated according to the bandwidth values of the target application core graph. The la-

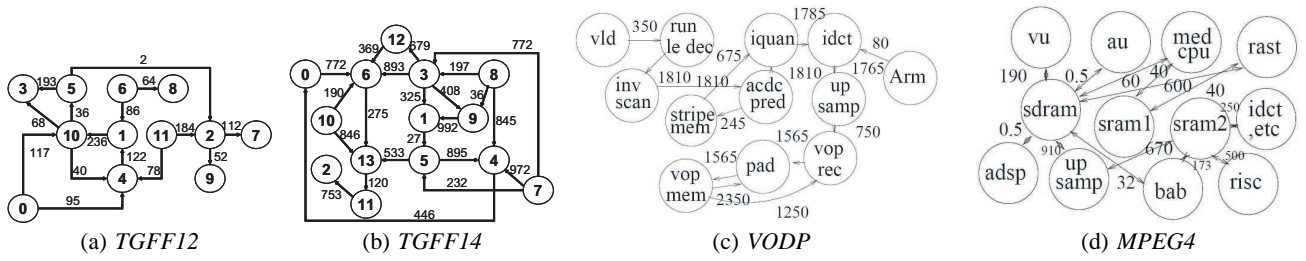


Figure 2: Communication core graphs of two random generated applications obtained by using TGFF [20] and two multimedia applications derived from [21].

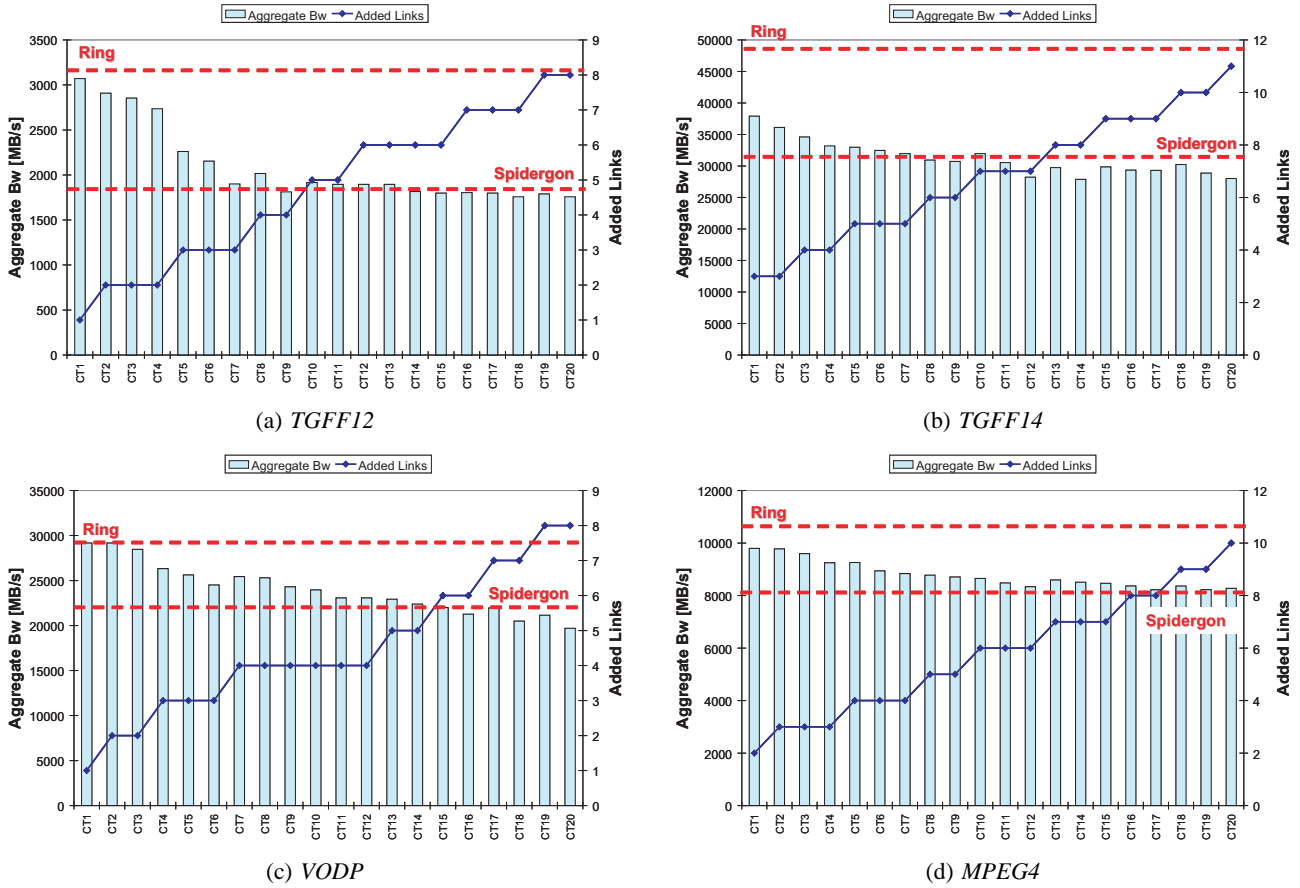


Figure 3: Aggregate network bandwidth comparison for the different Customized Topologies (CTn), the ring and the spidergon topologies, showing also the number of added links.

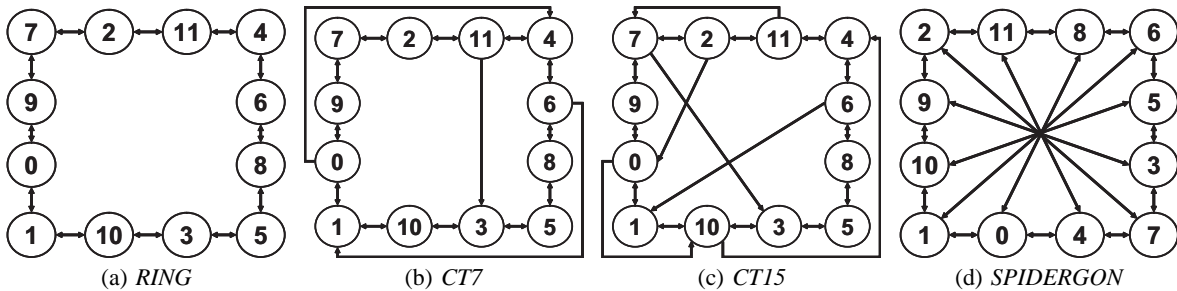


Figure 4: STNoC topologies comparison: ring, spidergon and two examples of customized topologies (CT7 and CT15) for the TGFF12 case study.

Table 1: Metrics comparison among the STNoC topologies in Figure 4 for TGFF12.

	Ring	CT7	CT15	Spidergon
Area [%]	-	+8.2	+19.2	+41.4
Power [%]	-	-38.6	-40.4	-24.4
Latency [%]	-	-29.72	-32.3	-30.7
AB [MB/s]	3192	1900	1798	1876
LU_{avr} [MB/s]	133.0	70.4	59.9	52.1
LU_{max} [MB/s]	353	353	236	353

tency values are calculated considering the time spent on average in the network by each flit.

While for area, power consumption and network latency the results are presented in Table 1 in percentage with respect to the ring values, for the bandwidth results (aggregate bandwidth and maximum and average link utilization) the reported values are absolute. As expected, increasing the number of added links with respect to ring, from 3 (CT7) to 12 (spidergon) passing through 6 (CT15), the area values for the network also increase rapidly, from 8.2% for CT7 to 41.4% for spidergon. This increment in area complexity has only a limited effect on the average power consumption of the network which is more related to the dynamic communication behavior of the system. In fact, since the aggregate bandwidth reduces rapidly with the increment of the number of added custom links, the average power consumption decreases despite of the area increment. Passing from CT15 to spidergon, the power consumption increases since the greater complexity of the spidergon topology is not counterbalanced by a corresponding reduction in aggregate bandwidth.

Similar behavior is also shown for the network latency which benefits by the reduced network distances due to the custom links and from low values of bandwidth, that means low contention probability. CT15 outperforms spidergon performance also in terms of network latency. In fact, the greater number of links of the spidergon topology are not fully exploited by the target application since they are not customizable but placed in fixed positions.

As expected, the average link utilization reduces its value with the increment of the number of links, also reducing the probability of link contention on the whole network. The last line of the Table 1 shows the maximum link utilization: the worst case link utilization corresponds to the highest probability of contention. The value is the same for ring, CT7 and spidergon, while it is reduced for CT15, since one of the added links in CT15 is used to split the flow of the most used link in more than one link. The maximum link utilization value for CT15 corresponds also to the highest value of the edge on the core graph, meaning that this is the best value that can be obtained for this metric.

7. CONCLUSIONS AND FUTURE WORKS

In this paper, we presented the application-specific topology customization of STNoC, an industrial network on-chip architecture developed by STMicroelectronics. Starting from ring topology, the basis of the STNoC family of topologies, the presented customization flow tries to add links to reduce the network distance among the mapped nodes (increasing the performance) without exceeding the maximum number of predefined STNoC router ports, thus keeping limited the area and power overhead.

8. REFERENCES

- [1] L. Benini and G. De Micheli. Networks on Chip: A New SoC Paradigm. *IEEE Computer*, 35(1):70–78, January 2002.
- [2] J. Duato, S. Yalamanchili, and L. Ni. *Interconnection Networks An Engineering Approach*. Morgan Kaufmann, 2002.
- [3] STMicroelectronics. Stnoc: Building a new system-on-chip paradigm. *White Paper*, 2005.
- [4] M. Coppola, R. Locatelli, G. Maruccia, L. Perialisi, and A. Scandurra. Spidergon: A Novel On-Chip Communication Network. In *SOC 2004: International Symposium on System-on-Chip*, November 2004.
- [5] L. Bononi and N. Concer. Simulation and analysis of network on chip architectures: Ring, spidergon and 2d mesh. In *Design, Automation and Test in Europe, 2006. DATE '06. Proceedings*, volume 2, pages 1–6, 06–10 March 2006.
- [6] Leonel Tedesco, Aline Mello, Diego Garibotti, Ney Calazans, and Fernando Moraes. Traffic generation and performance evaluation for mesh-based nocs. In *SBCCI '05: Proceedings of the 18th annual symposium on Integrated circuits and system design*, pages 184–189, New York, NY, USA, 2005. ACM Press.
- [7] P. P. Pande, C. Grecu, M. Jones, A. Ivanov, and R. Saleh. Performance Evaluation and Design Trade-Offs for Network-on-Chip Interconnect Architectures. *IEEE Transaction on Computers*, 54(8):1025–1040, August 2005.
- [8] G. Palermo and C. Silvano. PIRATE: A Framework for Power/Performance Exploration of Network-On-Chip Architectures. In *PATMOS-04: Proc. of International Workshop on Power and Timing Modeling, Optimization and Simulation*, 2004.
- [9] J. Hu and R. Marculescu. Energy-Aware Mapping for Tile-based NoC Architectures Under Performance Constraints. *ASP-DAC 2003*, Jan 2003.
- [10] S. Murali and G. De Micheli. Bandwidth-Constrained Mapping of Cores onto NoC Architectures. *IEEE DATE-04: Design, Automation, and Test in Europe*, pages 896–901, Feb. 16–20, 2004.
- [11] G. Ascia, V. Catania, and M. Palesi. Multi-Objective Mapping for Mesh-based NoC Architectures. *CODES-ISSS: Second International Conference on Hardware/Software Codesign and System Synthesis*, pages 182–187, Sep, 2004.
- [12] T. Lei and S. Kumar. A Two-Step Genetic Algorithm for Mapping Task Graphs to a Network on Chip Architecture. *Euromicro Symposium on Digital Systems Design*, Sep, 2003.
- [13] S. Murali, M. Coenen, A. Radulescu, K. Goossens, and G. De Micheli. A methodology for mapping multiple use-cases onto networks on chips. In *Proceedings of DATE '06. Design, Automation and Test in Europe.*, 2006.
- [14] S. Murali and G. De Micheli. An application-specific design methodology for stbus crossbar generation. In *Proceedings of Design, Automation and Test in Europe.*, 2005.
- [15] U.Y. Ogras and R. Marculescu. "it's a small world after all": Noc performance optimization via long-range link insertion. *IEEE Transactions on*, 14(7):693–706, July 2006.
- [16] U.Y. Ogras and R. Marculescu. Energy- and performance-driven noc communication architecture synthesis using a decomposition approach. In *Design, Automation and Test in Europe, 2005. Proceedings*, pages 352–357Vol.1, 2005.
- [17] R.P. Dick, D.L. Rhodes, and W. Wolf. TGFF: task graphs for free. In *(CODES/CASHE '98) Proceedings of the Sixth International Workshop on Hardware/Software Codesign.*, pages 97–101, 15–18 March 1998.
- [18] D. Bertozzi, A. Jalabert, Srinivasan Murali, R. Tamhankar, S. Stergiou, L. Benini, and G. De Micheli. Noc synthesis flow for customized domain specific multiprocessor systems-on-chip. *Parallel and Distributed Systems, IEEE Transactions on*, 16(2):113–129, Feb 2005.