An ILP Formulation for Application Mapping onto Network-on-Chips

Suleyman Tosun
Computer Engineering Department,
Ankara University
06500, Besevler, Ankara, Turkey
tosun@eng.ankara.edu.tr

Ozcan Ozturk
Computer Engineering Department,
Bilkent University
06800, Bilkent, Ankara, Turkey
ozturk@cs.bilkent.edu.tr

Meltem Ozen
Computer Engineering Department,
Ankara University
06500, Besevler, Ankara, Turkey
meltemmozen@gmail.com

***Abstract*-Ever shrinking technologies in VLSI era made it possible to place several modules onto a single die. However, the need for the new communication methods has also increased dramatically since traditional bus-based systems suffer from signal propagation delays, signal integrity, and scalability. Network-on-Chip (NoC) is the biggest step towards the communication bottleneck of System-on-Chip (SoC) architectures. In this paper, we present an Integer Linear Programming (ILP) formulation for application mapping onto mesh based Network-on-Chips to minimize the energy consumption of the system. The proposed method obtains optimal or close to optimal results within the given computation time limit. We also experimentally investigate the impact of the size of the mesh architecture on the application mapping and total communication.**

I. INTRODUCTION

As International Technological Roadmap for Semiconductors (ITRS) reports [1], integrated circuits will be implemented in less than 11 nm technology in 2022 allowing to place several computational and storage cores as elements of System-on-Chip. This technological development will also bring communication problems among several cores since the signal propagation will span multiple clock cycles. Network-on-Chip [2][3] has been proposed in the beginning of this century as a new communication infrastructure to overcome the stated problems. NoC architectures mimic the traditional interconnection network concepts on a single chip and several design techniques are adopted from it.

NoC architectures can be constructed by using either regular topologies or irregular (custom) topologies. Both topologies have advantages and disadvantages one another: Most of the cores in an application are heterogeneous in size and functionality and also demand different communication bandwidth. Irregular topologies suits well to the optimization of these different requirements such as link size, number of routers to be used etc. However, they are fixed to the designed application that cannot be used for new designs. Regular topologies can be reused and are easy to design. Most of the multi-core architectures employ regular topologies, especially mesh topology. An example is Intel's Teraflops Research Chip [4] that has 80 cores connected in a 2D mesh network.

Application mapping onto mesh topologies has been a well known NP-hard problem [5]. There have been several techniques [6][7][8][9] proposed for the mapping problem, mainly having the energy minimization as an objective criteria. In [6], authors propose a mapping algorithm called PMAP that supports single-minimum-path routing and split-traffic routing. MOCA [7] uses slicing tree based core mapping and generates routes on the mapping result. ONYX [8] is also a heuristic method that maps the cores based on the lozenge-shaped path order. CGMAP [9] employs chaos-genetic-based algorithm that obtains close results compared to other algorithms.

In this work, we present a 0-1 Integer Linear Programming formulation that obtains optimum results in a tolerable time as our experiments demonstrate. We test the impact of our ILP based framework on several real benchmarks under a given CPU time limit. Our experiments show that under the given time limit our tool obtains optimum results most of the time. However, when the number of tasks in the application increases, in some cases, it may not find the optimum results in the given time. We observe that, in such cases, our tool obtains very close results to the optimum one. In our experiments, we also investigated the effects of the mesh size on the final application mapping.

We organized the rest of the paper as follows: In the next section, we present the problem definition and energy model of the proposed system. We explain our formulations in Section III. We demonstrate the experimental data in Section IV. Finally, in Section V, we conclude this paper with future directions.

II. SYSTEM DEFINITIONS

In this section, we first define the mapping problem with the models we used to represent the application and the target architecture. We then present the energy model used to estimate the overall energy of the final system.

A. Problem Definition

We use weighted communication task graph (WCTG) and topology graph (TG) to represent the input application and target architecture, respectively, as we define them as follows:

Definition 1: A WCTG is a graph $G(V, E)$, where each vertex $v_i \in V$ represents a task in the application and each edge $e_{i,j} \in E$ represents a dependency between v_i and v_j. The amount of data transfer between two tasks, v_i and v_j, is represented by the weight $w_{i,j}$ for all $e_{i,j}$ in bits per second.

Definition 2: A TG is a graph $M(T, L)$, where each node $t_i \in T$ denotes the router of a tile in the topology and each edge denotes a physical link $l_{i,j} \in L$ between t_i and t_j. $c_{i,j}$ represents the capacity (i.e. the maximum allowed data transfer in bits per second) of a link $l_{i,j}$.

978-1-4244-4740-4/09/$25.00 ©2009 IEEE

The NoC mapping problem is to determine the one to one mapping function that maps each vertex in the WCTG onto tiles of TG. Formally:

$F: V \to T, s.t.\ f(v_i) = t_i, \forall v_i \in V, \exists t_i \in t, |V| \leq |T|$

and the total communication energy is minimized.

B. Energy Model

In this work, we use a well-known and -accepted energy model [8] given in (1). In this energy model, E_{bit} is the estimate energy consumption of a single bit from source tile to the destination tile on the network. In (1), E_{Rbit} and E_{Lbit} denote the energy consumption of the bit on the routers and physical links, respectively. In this model, we also accept the assumptions of [2] that the length of the physical links between tiles is 3 mm and the energy consumptions of each router for a single bit is equal to E_{Rbit}.

$$E_{bit} = E_{Rbit} + E_{Lbit}. \quad (1)$$

If there is a communication trace between $t_i$and t_j through α routers, the total dynamic energy consumed by a single bit of this communication can be computed using (2).

$$E_{bit}^{t_i,t_j} = \propto \times E_{Rbit} + (\propto -1) \times E_{Lbit}. \quad (2)$$

Let v_a and v_b are vertices mapped onto the tiles t_i and t_j, respectively. The communication between two tiles can be computed by (3).

$$E_{total}^{t_i,t_j} = \mathrm{w_{a,b}} \times E_{bit}^{v_a,v_b}. \quad (3)$$

Finally, the total amount of energy consumption, E_{NoC}, of the network can be calculated using the following formula:

$$E_{NoC} = \sum_{\mathrm{e_{i,j}} \in \mathrm{E}} E_{total}^{t_i,t_j}. \quad (4)$$

III. ILP Formulation

In an ILP problem, problems are formulated and optimized using a linear objective function and using linear functions as constraints, whereas the solution variables are restricted to be integers. The 0-1 ILP is a smaller subset of the general ILP problem in which each (solution) variable is restricted to be either 0 or 1.

In this paper, we used *Xpress-MP* [10], a commercial tool, to formulate and solve our ILP problem, though its choice is orthogonal to the focus of this paper. In our ILP formulation, we view the chip area as a 2D grid and assign tasks to tiles within this grid. Table I gives the constant terms and variables used in our formulations.

The structure of our ILP based solver and optimizer is presented in Fig.1. As shown in this figure, our tool takes the WCTG and TG as inputs and outputs the mapping of the tasks on the architecture. Additionally, it computes the total communication weight and energy which is the optimum result. In the mesh architecture, each router has a 2D coordinates, *x* and *y*. In Fig.1, we indicate these dimensions on the router in the parenthesis such as (2,0), meaning that $x = 2$ and $y = 0$ for that router.

TABLE I
CONSTANTS AND VARIABLES USED IN THE FORMULATIONS.

Constants & Variables	Definitions
n	The number of tasks in input WCTG
$w_{i,j}$	The communication weight between tasks *i* and *j* in WCTG
$Xdim$	The size of the mesh architecture in *x* dimension.
$Ydim$	The size of the mesh architecture in *y* dimension.
$\propto_{i,x,y}$	Binary variable. $\propto_{i,x,y} = 1$ if task *i* is mapped to the router in the coordinates (*x*,*y*) . $\propto_{i,x,y} = 0$ otherwise.
$X_{i,j,a}$	Binary variable. $X_{i,j,a} = 1$ if the distance in *x* dimension between task *i* and *j* is equal to *a*. Otherwise, $X_{i,j,a} = 0$.
$Y_{i,j,b}$	Binary variable. $Y_{i,j,b} = 1$ if the distance in *y* dimension between task *i* and *j* is equal to *b*. Otherwise, $Y_{i,j,b} = 0$.
$Xcost$	The total communication cost of the network in *x* dimension.
$Ycost$	The total communication cost of the network in *y* dimension.

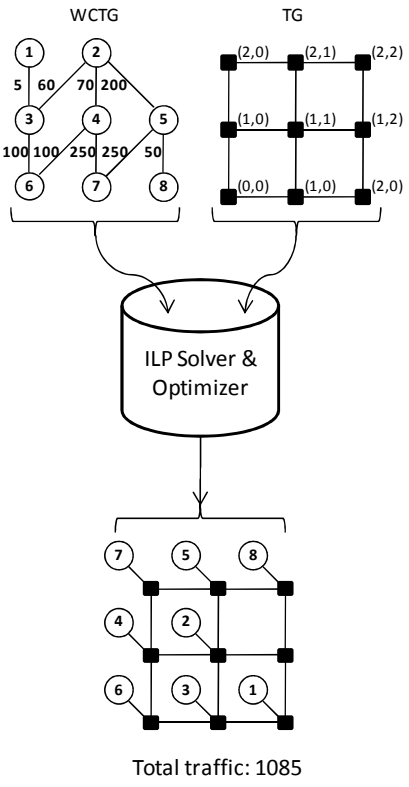

Fig. 1. The structure of presented ILP model. Our ILP solver and optimizer takes WCTG and TG as inputs and outputs the optimal mapping with total traffic and energy consumption.

In our formulation, n represents the number of tasks in WCTG. The dimension bounds of the mesh architecture are represented by $Xdim$ and $Ydim$ and they must hold the following inequalities:

$$Xdim \leq Ydim \leq Xdim + 1. \quad (5)$$

$$|V| = n \leq Xdim \times Ydim. \quad (6)$$

Inequality (5) forces the mesh to be in a square-like shape as much as possible. This gives more routing options between each tile in the mesh. However, this may not be the best mesh architecture for each specific application since each application's communication trace may vary significantly. For example, assume we have 14 nodes in the application. We may have 4×4, 3×5, 2×7, 1×14 configurations for the target mesh architecture. Which of these configurations fits best to the given application depends on the communication structure of the application. We show the effects of the mesh dimensions on the final mapping in Section IV. Equation (6) indicates that the number of routers must be greater than the number of nodes in the application graph. After selecting the dimensions of the architecture, we can input it to the ILP solver together with the application graph.

In our formulation, we define a binary variable $\propto_{i,x,y}$ which indicates that task i is mapped to a router in the coordinates (x,y) if $\propto_{i,x,y} = 1$, otherwise $\propto_{i,x,y} = 0$.

The ILP formulation starts with (7) indicating that every task i must be mapped to a router with the coordinate (x,y) and only one task can be mapped to a single router. The number of tasks in the WCTG may be less than the number available routers. In this case, some routers will not have any task mapped on it. Equation (8) captures this constraint.

$$\sum_{x=0}^{Xdim} \sum_{y=0}^{Xdim} \propto_{i,x,y} = 1, \quad \forall i. \tag{7}$$

$$\sum_{i=1}^{n} \propto_{i,x,y} \leq 1, \quad \forall x, y. \tag{8}$$

The total communication of the nodes helps find the total energy consumption of the final design as explained in the energy model in Section II. Thus, we have to calculate the number of hops between two tasks mapped on the mesh which means that we have to find the Manhattan distance (city block distance) between two mapped communicating tasks. For this calculation, we define two binary variables $X_{i,j,a}$ and $Y_{i,j,b}$ representing the distance in x and y dimensions, respectively, between tasks i and j where $e_{i,j} \in E$. Equations. (9) and (10) are used to determine the distance a (for x dimension) and b (for y dimension), respectively.

$$X_{i,j,a} \geq \propto_{i,x_1,y_1} + \propto_{i,x_2,y_2} - 1, \quad \forall i,j.\, e_{i,j} \in E$$

$$0 \leq x_1, x_2 \leq Xdim, \; 0 \leq y_1, y_2 \leq Ydim$$

$$s.t. \quad a = |x_1 - x_2| \tag{9}$$

$$Y_{i,j,b} \geq \propto_{i,x_1,y_1} + \propto_{i,x_2,y_2} - 1, \quad \forall i,j.\, e_{i,j} \in E$$

$$0 \leq x_1, x_2 \leq Xdim, \; 0 \leq y_1, y_2 \leq Ydim$$

$$s.t. \quad b = |y_1 - y_2| \tag{10}$$

We then calculate the cost in x and y dimensions using (11) and (12). In these formulas, a and b represent the number of hops in x and y dimensions, respectively. Multiplying these values with the communication weight, $w_{i,j}$, of two communicating tasks, i and j, gives us the total communication cost of these two nodes on the architecture.

$$Xcost = \sum_{e_{i,j} \in E} \sum_{a=0}^{Xdim} w_{i,j} \times a \times X_{i,j,a} \tag{11}$$

$$Ycost = \sum_{e_{i,j} \in E} \sum_{b=0}^{Ydim} w_{i,j} \times b \times X_{i,j,b} \tag{12}$$

Consequently, our objective function can be expressed as:

$$minimize : Xcost + Ycost. \tag{13}$$

Minimizing the total communication cost results in minimized energy as we showed in our energy model that these two metrics are directly proportional.

IV. Experimental Results

In our experiments, we use several real multimedia benchmarks, namely; VODP [8], MPEG4 [8], MWD [12], 263-dec mp3 dec [11], 263-enc mp3 dec [11], and mp3-enc mp3 dec [11]. We used *Xpress-MP* tool for our ILP model and our machine that we run our experiments is a PC with Intel E8400 CPU runs at 3.00 GHz with 4 GB RAM.

In our first experiment, we run our tool to map our benchmarks to given mesh architecture. We limit the running time of our tool to one hour (3600sec). In Table II, we present the results of this experiment. In this table, first four columns give some information about the benchmarks; their names, the number of vertices, the number of edges, and the total communication between tasks in Mbit per second. We mapped the given applications to 4×4 mesh architecture based on the number of vertices. In the next three columns (columns 5-7), we present the total communication of our mappings, optimum mapping, and the ratio of these two results, respectively. Our total communication results are obtained under one hour. As the column seven illustrates, most of the time, we obtain the optimum results within this time limit. There is only one benchmark, VODP, that we obtain very close result to the optimum. In fact, the total communication we obtained for this benchmark is very close that there is only 1% difference from the optimum one. Column eight in Table II gives the average hop count of our mapping. As one can observe, optimum average hop count is very close to 1. However, there are cases that some of the nodes are apart from each other more than one hop.

In the next three columns (columns 9-11) of Table II, we present the total power consumption of our mapping, the total power consumption of optimal mapping, and the difference between these two results, respectively. Note that, these power consumptions are only for the network. That is, we only count for the power consumption of the routers and the links between the tiles. We exclude the power consumption of the tiles (i.e. the processing elements). We use the power consumption values in 100-nm technology as they are given in [11]. In [11], the power consumption of the input port of the router is estimated as 328 nW/Mb/s and the power consumption of the output port is given as 65.5 nW/Mb/s. The power consumption of the physical link is

estimated as 79.6 nW/Mb/s/mm. In our calculations, we assume the link length between the tiles as 3 mm as suggested in [2]. As the power consumption difference is given in Table I, we obtain the optimum power consumption values most of the time.

Last three columns of Table II show the running time of the benchmarks. If the benchmark runs more than one hour, we use the solution found within this time limit. As one can observe from these results, we only obtain the optimum result for MPEG4 under this time limit. However, even the benchmark runs more than this limit; the result is optimum since the optimizer cannot find any better result afterwards.

The biggest problem of ILP models is the running time. When the number of variables increases, the running time increases tremendously. For example, our benchmarks VODP and 263 dec mp3 dec have the highest vertex and edge numbers. As a result, the running times of these two are greater than other small sized benchmarks. To overcome this bottleneck, the solution space can be relaxed. In [11], authors propose a clustering based ILP method for irregular topologies. However, this technique makes a concession from the total power consumption in some cases. That is, the result is a little worse than original method. However, it obtains very close results to optimum solution most of the cases. A similar approach can be used for mesh based architecture: The tasks in the application graph can be clustered. Based on the number of clusters, the mesh architecture can be divided into smaller mesh structures. Then, our ILP based technique can be applied to map the clusters onto corresponding sub meshes. We think that the results would be obtained in a very short time based on the number of the tasks in each cluster. However, the final result may not be optimum. Our future work will be the relaxation techniques of our ILP-based method.

In our experiment, we also study the impact of the mesh dimensions on the final mapping. For this experiment, we choose MPEG4 multimedia benchmark as an input application. We selected 4×4, 4×3, 6×2, and 12×1 mesh sizes for the target architecture.

Fig.2.(a) presents the WCTG of the MPEG4 application. The application mapping on the mesh sizes of 4×4, 4×3, 6×2, and 12×1 are given in Fig.2.(b),(c),(d), and (e), respectively. As we observe from this experiment, when the dimensions of the mesh are close to each other, we have more routing options. As a result, the mapping results in a better energy/communication cost. However, when the difference between two dimensions increase (i.e. The final size is a bus topology as seen in Fig.2.(e).), the routing freedom decreases. We also run two of our other benchmarks; MWD and 263 Enc mp3 Dec for the same mesh dimensions. In Fig.(3), we present the total communication and the CPU running time for these benchmarks. In Fig.3.(a), we give the total communications of these three benchmarks under different mesh sizes. While the total communication changes for every mesh dimension for MPEG4, it is not the case for other two benchmarks. When we investigate the application graphs of these three benchmarks, we see that MPEG4 is more strongly connected than the other two. It has 13 edges while the other two has 12. Thus, the degree of connectedness of the graph decides mesh size for the optimum results.

V. CONCLUSION

In this work, we presented a new Integer Linear Programming based application mapping tool for mesh-based Network-on-chip architectures. Our tool finds optimal or close to optimal results under given time limit. We show the results for six multimedia benchmarks. We also demonstrated the effects of the mesh sizes on the final mapping.

As a future work, we plan to study the relaxation techniques of our ILP formulation since it takes very big run times when the number of tasks in application graph increases. We also included the investigation of the relation of mesh sizes and the application graph in our agenda.

ACKNOWLEDGMENT

This work is supported by Scientific and Technological Research Council of Turkey (TUBITAK) under the project ID 108E233.

TABLE II
ILP SOLUTIONS ON DIFFERENT MULTIMEDIA BENCHMARKS.

Application	Vertex #	Edge #	Total communicat. (MBit/sec)	Total comm. (MBit/sec)		Ratio (O/I)	Average hop count	Power Consumpt. (μW)		Difference (P-G) (μW)	Time (sec)		Ratio (T/Z)
				Ours (I)	Opt (O)			Ours (P)	Opt (G)		Ours (T)	Opt (Z)	
VODP	16	20	3731	4119	4087	0.99	1.1040	4466.29	4413.27	53.02	3600	28840	0.19
MPEG4	12	13	3466	3567	3567	1	1.0291	3719.27	3719.27	0	380	380	1
MWD	12	12	1120	1184	1184	1	1.0571	1253.79	1253.79	0	3600	7750	0.46
263 Dec	14	15	19.636	19.823	19.823	1	1.0095	20.43	20.43	0	3600	23514	0.15
263 Enc	12	12	230.214	230.407	230.407	1	1.0008	236.24	236.24	0	3600	5825	0.62
Mp3 Enc	13	13	16.521	17.021	17.021	1	1.0303	17.75	17.75	0	3600	8476	0.42

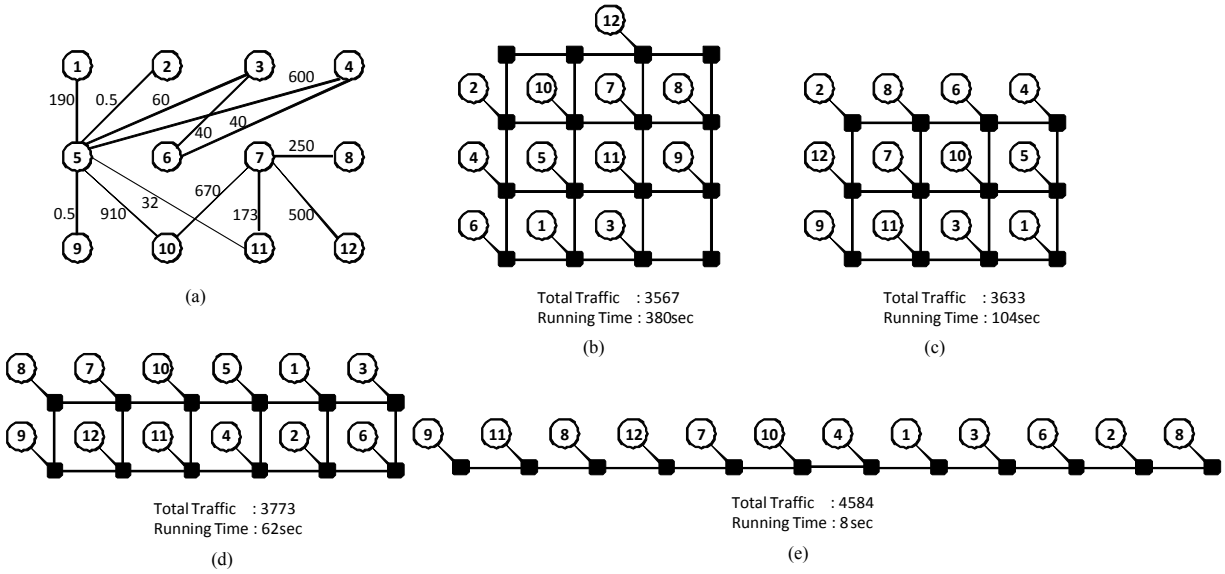

Fig. 2. The mappings of MPEG4 onto different mesh sizes. (a) WCTG of MPEG4 Decoder, Mapping of MPEG4 onto (b) 4x4 mesh, (c) 4x3 mesh, (d) 6x2 mesh, and (e) 12x1 mesh (bus topology).

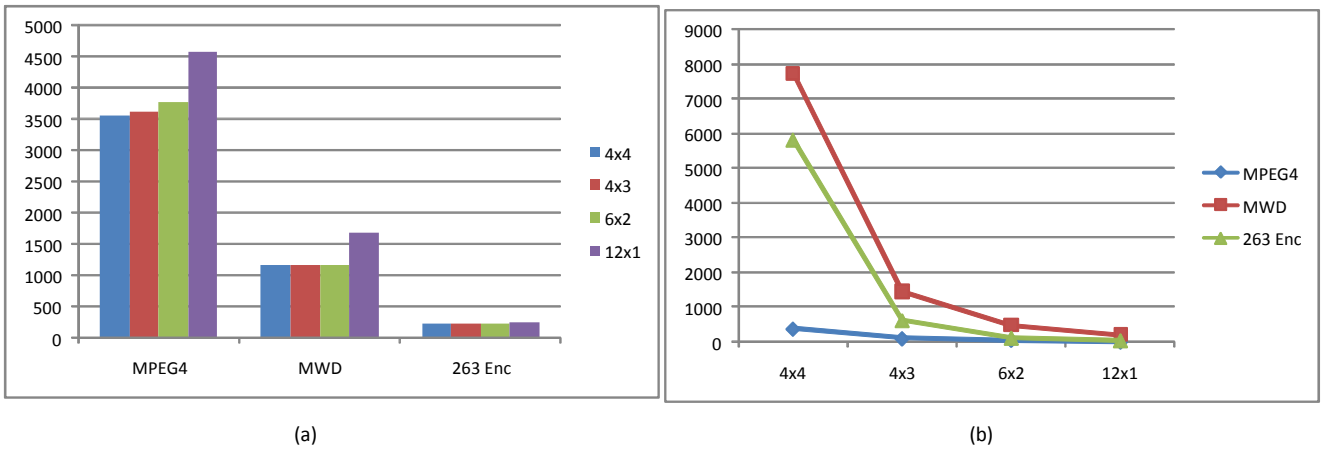

Fig. 3. The effect of mesh sizes on total communication and CPU running time. (a) The total communication (in Mbit/sec) of three benchmarks on different mesh sizes. (b) The effect of mesh sizes on CPU running time (in sec).

REFERENCES

[1] Visit http://www.itrs.net for ITRS 2007 edition.

[2] W. J. Dally and B. Towles, "Route Packets, Not Wires: On-Chip Interconnection Networks," *Proc. Design Automation Conference*, Las Vegas, Nevada, USA, pp. 684-689, 2001.

[3] L. Benini and G. De Micheli, "Networks on Chips: A New SoC Paradigm," *IEEE Computer,* vol. 35, no. 1, pp. 70-78, Jan. 2002.

[4] Visit http://techresearch.intel.com/articles/Tera-Scale/1421.html for Tera-Scale Computing Research Program.

[5] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman and Co., San Fransisco, CA, 1979.

[6] S. Murali and G. De Micheli, "Bandwidth-Constrained Mapping of Cores onto NoC Architectures," *Proc. DATE'04*, vol.2, pp. 896-304, Feb. 2004, Paris, France.

[7] K. Srinivasan and K. S. Chatha, "A technique for low energy mapping and routing in network-on-chip architectures," *Proc. ISLPED'05*, pp. 387-392, Aug. 2005, San Diego, California.

[8] M. Janidarmian, A. Khademzadeh, and M. Tavanpour, "Onyx: A new heuristic bandwidth-constrained mapping of cores onto tile-based Network on Chip," *IEICE Electron. Express,* vol. 6, no. 1, pp. 1-7, Jan., 2009.

[9] F. Moein-darbari, A. Khademzade, and G. Gharooni-fard, "CGMAP: a new approach to Network-on-Chip mapping problem," *IEICE Electron. Express*, vol. 6, no. 1, pp. 27-34, Jan., 2009.

[10] Visit http://www.dashoptimization.com for Xpress-MP.

[11] Srinivasan, K., Chatha, K. S., and Konjevod, G. 2006. Linear-programming-based techniques for synthesis of network-on-chip architectures. *IEEE Trans. Very Large Scale Integr. Syst.* 14, 4 (Apr. 2006), 407-420.

[12] K.-C. Chang and T.-F. Chen, Low-power algorithm for automatic topology generation for application-specific networks on chips, *IET Comput. Digit. Tech.*, 2008, Vol. 2, No. 3, pp. 239–249.