STG-NoC: A Tool for Generating Energy Optimized Custom Built NoC Topology

Surbhi Jain
Department of CSE
College of Technology and Engineering, MPUAT
Udaipur, India

Naveen Choudhary
Department of CSE
College of Technology and Engineering, MPUAT
Udaipur, India

Dharm Singh
Department of CSE
College of Technology and Engineering, MPUAT
Udaipur, India

ABSTRACT
Network on Chip (NoC) has emerged as a viable solution to the complex communication requirements of constantly evolving System on Chip (SoC). The communication centric architecture of NoC can be optimized across a variety of parameters as per the design requirements. With the development of customized application the inclination has shifted from regular architectures to irregular topology which leaves researchers with larger spectrum of optimization parameters. Many heuristic methods have been explored as the optimization problems encountered are NP-hard. This paper presents a customized topology generator STG-NoC which implements a heuristic technique based on simulated annealing for achieving the objective of energy optimization.

General Terms
Network-on-Chip (NoC) Simulated Annealing (SA).

Keywords
Customized Network-on-Chip (NoC), Energy optimization, Simulated Annealing (SA), STG-NoC (Simulated Annealing based Topology Generator for Network on Chip).

1. INTRODUCTION
Evolving standards and customized applications have resulted in a rapid increase in the number of cores on SoCs [1].

A viable solution to address the communication problems of such SoCs is presented by NoCs [2-4] where cores are interconnected using on chip micro networks. This feature of NoC helps in better modularity and design predictability in comparison to traditional systems based on bus architecture.

The design challenges [5] that need to be addressed for NoC includes application specific communication patterns, high predictability, energy efficiency constraints, real time requirements and VLSI issues viz. structure and wiring complexity.

Standard topologies like meshes, tori, fat tree or k-ary n-cubes etc which are characterized by their simplicity were favored by early researchers [6-7]. However the regular structures are best suited for general purpose applications. The heterogeneous architecture of most SoCs renders these regular structures inadequate in terms of wiring complexity and energy overhead. Customized NoCs are apt for these complex and irregular architecture. The efficiency of the selected routing function is highly dependent on underlying topology. Therefore a topology generated according the requirement of application specific communication pattern is more effective [8-9]. More over the optimization in terms of area, energy and bandwidth can be achieved using application specific communication structure.

Various phases involved in NoC synthesis [5] are topology task identification, core to tile mapping, floorplanning and optimization. Different optimization approaches follow different order of the given process and can be broadly classified [10] as area optimized NoC synthesis and interconnection optimized NoC synthesis.

In this paper a customized topology generator tool STG-NoC is proposed. The energy optimized topology generation is based on a heuristic technique adapted from Simulated Annealing process. The methodology used by the tool falls under the area optimized NoC synthesis [10] category.

Simulated annealing [11-12] is a meta-heuristic optimization technique derived from annealing process where a metal at a high temperature cools to attain a minimum crystalline structure. The cooling process is marked by movement in neighborhood allowing significant and random changes during high temperatures and as temperature lowers the movement is slow accepting only those changes which minimize objective function.

The paper is organized as follows. The next section, Section II provides a brief overview of communication model required for customized topology generation. Section III explains in detail the methodology incorporated by STG-NoC. Experimental results are given in Section IV and the paper concludes with Section V.

2. COMMUNICATION MODEL
This section describes the communication model, NoC energy model and architecture and the applicable routing function for the customized NoC.

2.1 Communication Model
For an abstract level model of the behavior of multi-core SoC applications which are inherently complex use of Task Graphs [7], [13] is favorable. The model assumes a point to point connection of cores and the communication is unidirectional. Each task T_i is mapped to a set of cores represented by v_j. Figure 1 shows the generic communication model.

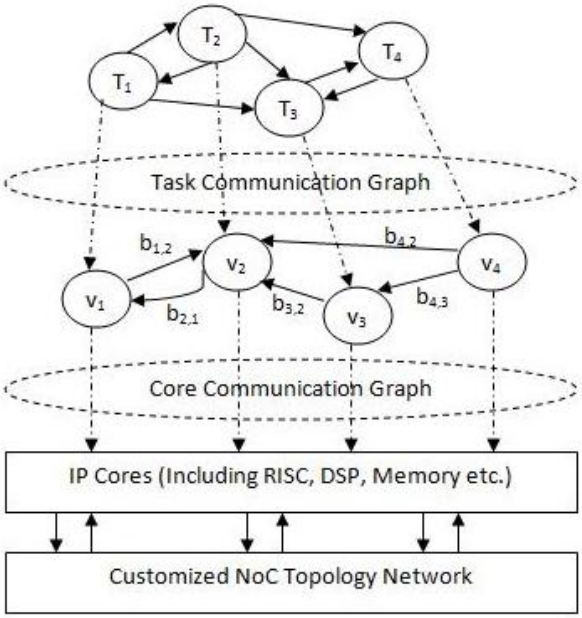

Figure 1: Application specific communication model in NoC

Definition 1: Core Graph is a directed communication graph, $G\ (V, E)$ where a vertex $v_i \boldsymbol{\epsilon}\, V$ represents an IP core. A directed edge $e_{i,j} \boldsymbol{\epsilon}\, E$ is a representation of the communication channel between the IP cores v_i and v_j and its weight $b_{i,j}$, represent the required average bandwidth of the communication from v_i tov_j .

Definition 2: In NoC topology graph $N\ (U, F)$ which is a directed graph, each vertex $u_i \boldsymbol{\epsilon}\, U$ represents a node/tile of the topology. A weighted-directed edge $f_{i,j} \boldsymbol{\epsilon}\, F$ is an abstraction of the direct communication channel from vertex u_i to u_j . Weight of the edge $f_{i,j}$ denoted by $Ab_{i,j}$ denotes the link/channel bandwidth available across the edge $f_{i,j}$.

2.2 NoC Energy Model and Layout

Floorplanning can be done using non-slicing based floorplannners such as B*-Trees [14]. The energy model [7] considered for the customized Network-on-Chip is described as follows:

$$E_{bit}\ (t_i, t_j) = n_{hops} \times Er_{bit} + (n_{hops} - 1) \times El_{bit} \quad \textbf{(1)}$$

Where $E_{bit}\ (t_i, t_j)$ represents the average dynamic energy consumption for sending one bit of data across tiles t_i and t_j, n_{hops} is the number of hops (router to router) the bit requires to traverse from tile t_i to tile t_j, Er_{bit} is the energy consumed by the router for transport of one bit of data and El_{bit} is the energy required by a unit of link/channel for transportation of one data bit.

2.3 Routing function

The routing functions considered by STG-NoC for customized NoC in this paper are up*/down* and Left-Right routing functions [15-16]. These functions are distributed in nature. The implementation is achieved through look-up table and the routing tables are derived using breadth-first search (BFS) on the topology graph for customized NoC. These routing function are based on turn restriction and thus achieve deadlock avoidance.

3. TOPOLOGY GENERATION USING STG-NoC

This section describes the process involved in topology generation using STG-NoC. The description is followed by figure 2 which gives a brief overview of the process.

3.1 STG-NoC design methodology

The method adopted by the tool to generate energy efficient customized topology takes application communication characteristic, defined by Task Graph and Core Graph, as input. The floorplans to be evaluated are obtained using B* -Trees [14] which is a non-slicing floor planner.

Link length is calculated according to Manhattan Distance and a physical constraint of maximum permitted channel length (e_{max}) is maintained. In addition nodes of the generated topology cannot exceed maximum permitted node degree (n_{max}). This prevents slowing down of routers due to delays caused by heavy traffic.

Topology generation is initiated by determining the shortest path using Dijkstra's algorithm followed by construction of the minimum spanning tree using Prim's algorithm. Due to the constraint of e_{max} and n_{max} the order in which traffic patterns (as observed from Core Graph) are considered determine the communication energy requirement of the topology. To identify this appropriate order a heuristic technique based on simulated annealing described in the next sub-section is utilized by STG-NoC.

The routing tables of the nodes/routers are updated with identified shortest paths during the process. Routing decision based on the selected routing function (up-down, Left-Right etc.) gives highest priority to shortest path. A modified Dijkstra's algorithm is used to find routes from each node in the shortest path to the corresponding destination according to the up*/down* (Left Right) rule. Thus a customized topology which has optimized energy requirements is generated along with required routing tables.

3.2 Annealing Schedule

A heuristic technique based on simulated annealing is used to identify the most appropriate order of the traffic pattern which will lead to optimized energy requirement of the generated customized topology.

The optimization schedule commences with a high initial temperature. The initial solution ($currentSolution$) is generated with a random order of traffic patterns obtained from Core graph. The objective function is calculated in terms of energy as given by equation **(1)**. The variable $bestSolution$ tracks the best solutions obtained during the schedule.

A search for an optimum energy solution is made in neighborhood ($newSolution$) of the initial solution. This is achieved by rearrangement of traffic patterns of the $currentSolution$. The extent of rearrangement is determined by the corresponding temperature and the size of the topology being considered. This allows for significant changes during the initial phase of the schedule. The acceptance of the solution is determined by a probability function $exp\ (diff/T)$ which is temperature dependent. This ensures that the search does not get trapped in local minima as the solution is accepted even if it drifts away from optimization target during early phases of the process.

For each temperature the process is repeated iteratively and the iteration count depends on the factor '$asize$' which

represents the number of corresponding traffic patterns. In addition a stopping criterion determines that the process runs for specific iterations and marks the end of schedule if no further improvement is observed in the results even if the final temperature is not reached. This improves time efficiency of algorithm.

Requirement
Input: Task graph, Core graph
Output: Customized topology

$\boldsymbol{E}$**:** Energy (objective) function
$\boldsymbol{T_0}$**:** Initial temperature
$\boldsymbol{crate}$**:** Temperature cooling rate

Adaptive simulated annealing schedule
1. Generate an initial solution $currentSolution$
2. Set temperature T to an initial value $T_0 > 0$
3. Set the final temperature T_f
4. Set $bestSolution$ to $currentSolution$
5. Set the iteration count $icnt$ corresponding to $asize$
6. **while** $T > T_f$ **or** $stopping\ criteria$ not true **do**
7. **for** $i = 1$ to $icnt$ do
    a. Generate a neighbour solution $newSolution$
    b. $diff = E(newSolution) - E(currentSolution)$
    c. **if** $diff < 0$ then
    d. $currentSolution = newSolution$
    e. else
    f. Generate a random number x uniformly in the range (0,1)
    g. **if** $x < exp\,(diff/T)$ then
    h. $currentSolution = newSolution$
    i. end if
    j. end if
    k. **if** $E(newSolution) < E(bestSolution)$
    l. $bestSolution = newSolution$
    m. end if
8. **end for**
9. $T = T * crate$
10. $currentSolution = newSolution$
11. check for $stopping\ criteria$
12. **end while**
13. **return** $bestSolution$

Figure 2: Overview of the heuristic used by STG-NoC

Figure 3 represent the optimization achieved with respect to communication energy using the heuristic technique as described above for the topology generation process of STG-NoC. The values were observed for $bestSolution$ variable. The observations are from a sample with 81 tile size and 89 communication patterns. Similar optimization trends have been observed for other tile sizes and configurations as well. As can be seen the initially drastic changes are observed which stabilize as the schedule progresses.

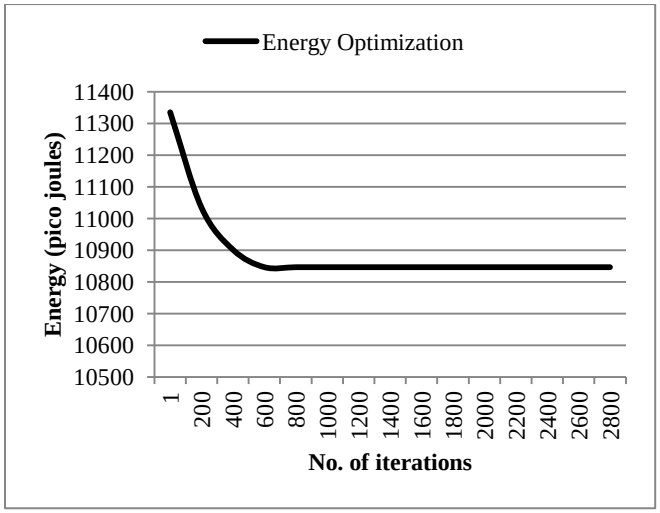

Figure 3: Energy optimization achieved using STG-NoC

4. EXPERIMENTAL RESULTS

The core graphs samples were generated using TGFF [13]. The experiments were conducted for about 100 samples evenly distributed across topologies ranging from size 16(4*4) to 81(9*9). Various communication patterns with diverse bandwidth requirement were considered for the evaluation.

For customized NoC generated using STG-NoC the maximum permitted channel length (e_{max}) was assumed 1.5 times the core length. The maximum number of ports (n_{max}) was restricted to 4. El_{bit} calculated for 0.18 micro meter technology was .000007 pico-joules per mili-meter length of wire. Er_{bit} was estimated to 0.52 pico-joules for 2-virtual channel router and 32 bit flit with help of Orion power simulator [17].

The simulator IrNIRGAM [18] which is an extended version of NIRGAM [19] was utilized for experimental evaluation. It is a cycle accurate, SystemC based simulator which supports wormhole switching. It has provision for architecture based on virtual channel and source and table based routing. IrNIRGAM supports irregular and customized topologies as well as escape path routing for deadlock avoidance applied to the customized topologies obtained from the tool STG-NoC.

The simulator also allows evaluation of regular topology configuration like mesh, tori etc and routing function viz. XY and OE. It was run for 10000 clock cycles uniformly.

For comparative evaluation the tile size and task to core/tile mapping of each sample under test was applied to regular 2D-mesh structure with deterministic XY routing function and adaptive OE routing function and comparison was done with respect to average dynamic energy consumption and latency.

The subsections below give the details of comparative results.

4.1 Energy consumption

Figure 4 shows the comparison of average of dynamic communication energy in pico-joules observed across various tile configurations for customized topologies generated using STG-NoC and the regular structured topologies. As can be observed from the graph the energy required by the topology generated by STG-NoC is much less than the regular configuration topologies. When compared with deterministic routing function XY the difference estimates to 30%, while a

difference of around 50% is observed in comparison with adaptive routing function OE. The difference becomes more significant with increase in tile size.

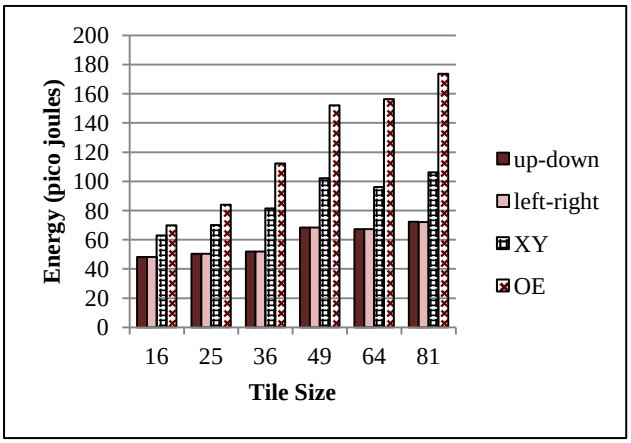

Figure 4: Avg. communication energy requirement comparison.

4.2 Latency Comparison

As can be observed from figure 5 the latency comparison shows patterns similar to that of energy requirement. The average flit latency observed in terms of clock cycles shows reduction when STG-NoC topology with Left-Right routing is compared with regular 2D-Mesh topologies with same tile size.

Customized topologies achieve a saving of around 21 clocks and 50 clocks against the regular topologies with XY and OE routing function respectively. The reduction is observed maximum for 81 tile size.

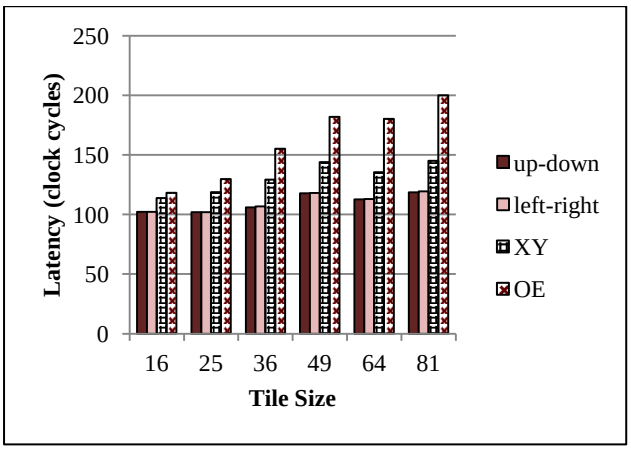

Figure 5: Latency comparison for various tile sizes.

4.3 Comparison for random benchmarks

Performance analysis of STG-NoC with regular 2D-mesh topology for a generic Multi Media System (MMS) application with 25 tile size and 30 communication tasks is in confirmation with the results presented in this section. The MMS application includes an h263 video encoder, an h263 video decoder, an mp3 audio encoder and an mp3 audio decoder.

A reduction of 4 and 27 clocks is observed when STG-NoC result (83.5 clock cycles for Left-Right routing) is compared with XY (87.6 clock cycles) and OE (106.3 clock cycles) routing function of mesh topology.

Similarly the average communication energy consumption of STG-NoC generated topologies (42.5 pico-joules) shows an improvement of 4% and 27% respectively over regular topology routing functions XY(46.7 pico-joules) and OE(69.6 pico-joules).

5. CONCLUSION

This tool STG-NoC presented in this paper generates customized topology while attempting to optimize communication energy requirements of the generated topology. When the results obtained from the tool are compared with those of 2D mesh topology with respective routing functions the tool is observed to succeed in its optimization objective. The improvements are obtained in terms of energy and latency as well. Similar success is observed with the bench mark application also. The tool can be upgraded to incorporate enhanced architecture like 3D NoC and other optimization objectives like bandwidth requirement can be explored as well.

6. REFERENCES

[1] International technology roadmap for semiconductors. Semiconductor Industry Association, 2003

[2] Dally, W. J., and Towles, B. 2001. Route Packet Not Wires: On-Chip Interconnection Networks. In IEEE Proceedings of the 38th Design Automation Conference (DAC), pp. 684–689.

[3] Bjerregaard, T., and Mahadevan, S. 2006. A Survey of Research and Practices of Network-on-Chip. In ACM Computing Surveys, Vol. 38, March 2006, Article 1.

[4] Duato, J., Yalamanchili, S., and Ni, L. 2003. Interconnection Networks: An Engineering Approach, Elsevier.

[5] Benini, L. 2006 Application Specific NoC Design. DEIS Universit´a di Bologna, IEEE website.

[6] J. Hu, R. Marculescu, "Energy- and performance-aware mapping for regular NoC architectures". In IEEE Trans. on CAD of Integrated Circuits and Systems, 24(4), April 2005.

[7] Hu, J., and Marculescu, R. 2003. Energy-aware mapping for tile-based NoC architectures under performance constraints. ASP-DAC 2003.

[8] Choudhary, N., Gaur, M.S., Laxmi, V., and Singh, V. 2011. GA Based Congestion Aware Topology Generation for Application Specific NoC. In: Electronic Design, Test and Application (DELTA), 2011 Sixth IEEE International Symposium.

[9] Choudhary, N., Gaur, M.S., Laxmi, V., and Singh, V. 2010. Energy aware design methodologies for application specific NoC. In NORCHIP, 2010.

[10] Choudhary, N., Singh, D., and Jain, S. 2011. Analyzing Methodologies of Irregular NoC Topology Synthesis. In IJCA Special Issue on Communication and Networks comnetcn (1):35-39, December 2011.

[11] Busetti, F. 2003. Simulated annealing overview. citeseerx.ist.psu.edu

[12] Bertsimas, D., and Tsitsiklis, J.1993. Simulated Annealing. Statistical Science. 1993, Vol. 8, No. 1, 10—15.

[13] R. P. Dick, D. L. Rhodes, W. Wolf, "TGFF: task graphs for free". In Proceeding of the International Workshop on Hardware/Software Codesign, March 1998.

[14] Chang, Y. C., Chang, Y. W., Wu, G. M., Wu, S. W. 2000. B*-Trees: A New Representation for Non-Slicing Floorplans. In Proceeding of 37th Design Automation Conference, pp. 458-463, 2000.

[15] Glass, C., and Ni, L. 1992. The Turn Model for Adaptive Routing. In Proceeding of 19th International Symposium on Computer Architecture. pp. 278–287, May 1992.

[16] Jouraku, A., Funahashi, A., Amano, H., and Koibuchi, M. 2001. L-turn routing: An Adaptive Routing in Irregular Networks. In Proceeding of the International Conference on Parallel Processing, pp. 374-383, Sep.2001.

[17] Wang, H-S., et al. 2002. Orion: A Power-Performance Simulator for Interconnection Network. In Proc. International Symposium on Microarchitecture, Nov 2002.

[18] Choudhary, N., Gaur, M. S., and Laxmi, V. 2011. Irregular NoC Simulation Framework: IrNIRGAM. In IEEE International conference on Emerging Trends in Network and Computer Communications (ETNCC), Udaipur, India.

[19] Jain, L., Al-Hashimi, B.M., Gaur, M.S, Laxmi, V. and Narayanan, A. 2007. NIRGAM: A Simulator for NoC Interconnect Routing and Application Modelling. Proc. DATE 2007.