Network-on-Chip Design for High Performance Demanding Multimedia Application

Naveen Choudhary
College of technology and Engineering
MPUAT
Udaipur, India

M. S. Gaur
Malaviya National Institute of Technology
Jaipur, India

V. Laxmi
Malaviya National Institute of Technology
Jaipur, India

ABSTRACT
Systems-on-Chip architecture integrates several heterogeneous components on a single chip. A key challenge is to design the communication between the different entities of a SoC in order to minimize the communication overhead. Network-on-chip (NoC) is a new approach for communication infrastructure of Systems-on-Chip (SoC) design, which provides network based solution for on-chip communication. Networks on Chip can be designed in different ways, according to the network architecture and protocol choice. It is important to balance the communication needs across the different links to avoid congestion and hot spots especially for high performance multimedia application. Application specific irregular topology based Network on Chip design can cater to the need of High performance demanding multimedia application which are congestion and energy aware. To achieve the mentioned objectives, this paper proposed two NoC design methodologies. Statistical experimental results show significant savings in communication bandwidth and communication energy for the high performance demanding multimedia applications when compared to existing standard 2D-Mesh NoCs.

General Terms
Network-on-Chip, System-on-Chip, Interconnection Networks, Embedded Systems, VLSI

Keywords
Genetic Algorithms, Core Graph, On-Chip Networks, Network-on-Chip, Optimization

1. INTRODUCTION
Systems on Chip (SoC) consists of numerous cores which are pre-existing components such as processors, DSPs, memory blocks. The use of standard hardwired busses to interconnect these cores is not scalable. But, as the systems grow and the design cycle time requirements decrease, the need for more generalized solutions becomes pressing. To overcome this problem, *Networks on Chips* (NoCs) [1], [2] have been proposed for interconnecting the cores and replacing traditional routing methods. In contrast to traditional SoC communication methods, such a distributed communication media scales well with chip size and complexity. The use of on-chip interconnection network has several advantages, including better structure, performance and modularity. Additional advantages include increased aggregated performance by exploiting parallel operation.

There has been a discussion in the NoC research domain about how to find an adequate Network-on-Chip (NoC) architecture that efficiently accommodates the applications communication needs. As noted in [3] an on-chip network is defined mainly by its topology and the protocol implemented by it. Topology concerns the layout and connectivity of the nodes and links on the chip, thus has a great impact on performance and implementation costs. Protocol dictates how these nodes and links are used. A vast range of NoC architectures has been proposed in the Noc research domain based on regular building patterns [1], [2] such as meshes (Figure 1) and torus, for the implementation of on-chip networks to overcome traditional bus-based designs with all their known limitations [1].

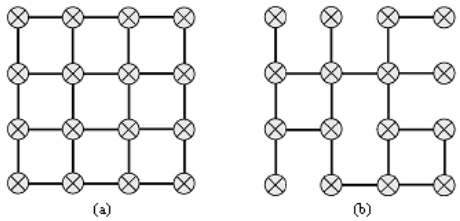


Fig 1: (a) Regular NoC Topology, (b) Irregular NoC Topology

Most SoC platforms are special-purpose tailored to the domain-specific requirements of their application. They are usually built from a large diversity of heterogeneous components with each core having different size, functionality and communication requirements, which communicate in a very specific, mostly irregular way. Thus, regular topologies can have a structure that poorly matches application traffic. This leads to large wiring complexity after floorplanning, as well as significant power and area overhead. In such systems the size and nature of cores may vary quite widely and therefore irregular topology proves better in performance in comparison to regular topology. Routing is the process of selecting paths in a network along which to send network traffic. The routing function is tightly coupled to the underlying topology, defining the set of allowed paths on which packets may travel from a source to the destination node. In the design of network architecture the proper selection of the adequate topology and routing function plays a key role.

The methodologies presented by the author in [4] and [5] are extended and critically compared in this paper for high

performance demanding multimedia applications. The work presented in this paper address the issue of customized topology/network design according to performance metric such as energy and congestion along with deadlock free communication for application specific homogenous or heterogeneous NoC according to communication requirements. In Section 2 we discuss standard NoC architecture. The Customized NoC architecture, communication model, energy model and routing function are defined in Section 3. Section 4 outlines the proposed energy aware and congestion aware customized NoC design for high performance demanding multimedia application. Section 5 presents some experimental results of the proposed design methodologies for high communication demanding multimedia application followed by a brief conclusion of the work in Section 6.

2. NOC ARCHITECTURE

An NoC is constructed from multiple data links interconnected by routers (Figure 2), such that messages can be relayed from any source module to any destination module over several links, by making routing decisions at the switches.

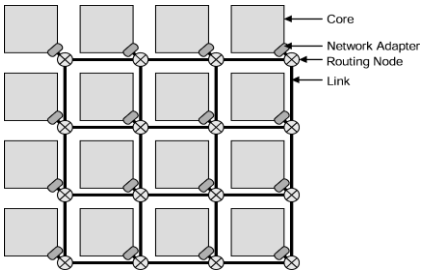


Fig 2: Topological aspects of NoC

If Switching is mere transport of data, then routing is the intelligence behind it, that is, it determines the path of the data transport. NoC uses packet switching as the fundamental transportation mode. In packet switching, traffic is forwarded on a per-hop basis. Packet switched communication may be either connection-oriented or connectionless. There are mainly three kinds of packet switching schemes: store-and-forward, virtual cut-through and wormhole switching. The proposed methodology has assumed wormhole switching. According to [3] *wormhole switching* combines packet switching with the data streaming quality of circuit switching to attain minimal packet latency [3]. The node looks at the header of the packet to determine its next hop and immediately forwards it. The subsequent flits are forwarded as they arrive. This causes the packet to *worm* its way through the network, possibly spanning a number of nodes, hence the name. The latency within the router is not that of the whole packet. A stalling packet, however, has the unpleasantly expensive side effect of occupying all the links that the worm spans. However *virtual channels (VCs)* [3] can relieve this side effect. VCs are the sharing of a physical channel by several logically separate channels with individual and independent buffer queues. In NoC generally 2 to 16 VCs per physical channel are recommended. Their implementation results in an area, power and latency overhead due to the cost of control and buffer implementation but there are also a number of advantages of *Virtual channels* like avoiding deadlocks, optimizing wire utilization, and improving performance.

3. CUSTOM NOC ARCHITECTURE AND COMMUNICATION MODEL

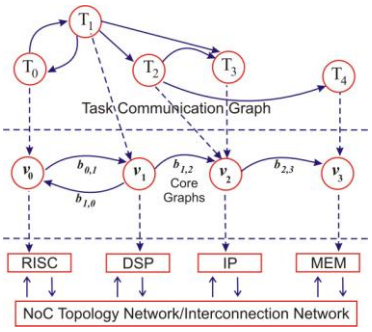


Fig 3: Custom NoC communication model

The generic communication model including Task graphs [6], [7], Core Graph, and NoC topology/interconnection network is shown in Figure 3 Where *Core Graph* is a directed graph with vertices representing the IP cores and the directed edges between vertices represents the communication bandwidth requirement of the application. Similarly *NoC topology graph* is a directed graph with each vertex representing a core/node in the topology and a directed edge among vertices represents the direct available communication channel along with the actual physically available bandwidth in the topology. For Performance evaluation, a discrete event, cycle accurate simulator *IrNIRGAM* can be used for Irregular NoC. *IrNIRGAM* is an extension of *NIRGAM* [8]. Ir*NIRGAM* is a cycle-accurate SystemC based performance simulator which supports irregular topology framework with table based routing.

A regular mesh based chip layout can be assumed for homogenous cores otherwise for NoC with heterogenous core, floorplanning according to desired metric such as area can be done as a pre-processing step using non-slicing based floorplannning tools such as B*-Trees [9] assuming over the cell routing [10] and Manhattan distance as channel length. The Chip layout information and Core Graph exhibiting traffic characteristics are taken as inputs for the methodologies presented in Section 4. Moreover in the proposed methodologies the link length is not allowed to exceed the maximum permitted channel length (e_{max}) due to constraint of physical signaling delay. Similarly a constraint on maximum permitted node-degree (nd_{max}) prevents the algorithm from instantiating slow routers with large number of I/O-channels which would otherwise decrease the achievable clock frequency due to internal routing and scheduling delay of the router.

The energy model [6] for the Network-on-Chip is defined as follows:

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

Where $E_{bit}(t_i, t_j)$ is the average dynamic energy consumption for sending one bit of data from tile t_i to tile t_j, n_{hops} is the number of routers the bit traverses from tile t_i to tile t_j, Er_{bit} is the energy consumed by router for transporting one bit of data and El_{bit} is the energy consumed by link/channel for transporting one bit of data. For the NoC networks with unequal link length, the 2nd term of the summation in the above mentioned equation can be replaced as the summation of bit energy consumed by each link/channel in the route, the bit follows from communication source core to the destination core. The above mentioned equation can be rewritten as follows for the NoC with unequal link lengths.

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

The methodologies presented in this paper uses generic routing functions for irregular NoCs such as up*/down* routing [11]. For the energy efficient methodology presented in Section 4, a variation of [11] as proposed in [12] is used where routing function is modified in such a way that packets arriving on a router are allowed to use virtual channels associated with minimal (shortest) paths as first priority and deadlock free *escape path* virtual channels for following deadlock free up*/down* path are allocated to packets only if the minimal path channels are unavailable as second priority. However, once a packet acquires a deadlock free channel following up*/down* path, it is not allowed to do transition to channels associated with minimal path anymore to avoid deadlock situation. However for the congestion aware methodology presented in Section 4, the deterministic routing function as proposed in up*/down* routing [11] is used.

4. NOC DESIGN METHODOLOGIES FOR HIGH PERFORMANCE MULTIMEDIA APPLICATION

4.1 Communication Energy Conscious Design for Multimedia Application

Since the Topology generation is the prime objective of the complete design flow (Figure 4) that spans multiple abstraction levels, down to placement and routing. A complete flow for generating the proposed congestion optimized topology for NoC is shown in Figure 4. In the first phase, the application traffic characteristics, and positioning ((x, y) coordinates) of cores are obtained for the given multimedia application. Floorplanning can done with the help of B*-tree [9] Floorplanner which is based on the B*-tree floorplan representation and is a multistage simulated annealing methodology. The cores are considered as rectangular tiles. The *Manhattan distance* (d) between two cores with lower right coordinates (x_1, y_1) and (x_2, y_2) can be calculated as $(d= |x_2 - x_1| + |y_2 - y_1|)$.

To avoid clock skew due to long link lengths, the links longer than permitted value are prohibited (e_{max}). Further, to prevent any switch from being the hot spot of congestion, each switch is allowed to directly communicate with a maximum of four switches only (nd_{max}).

In the second phase, minimum spanning tree is constructed using Prim's Algorithm. The objective of constructing the spanning tree is to ensure that all cores are connected. The tree is used to obtain the information regarding link types (i.e. up link or down link) which is required for up*/down* routing [11].

The third phase uses the *Genetic Algorithm (GA)* [13] to generate the topology for NoC which is optimum with respect to traffic load distribution across the channels of the NoC and is also communication energy sensitive. Genetic algorithms are a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover (also called recombination). Over many generations, natural populations evolve according to the principles of natural selection and "survival of the fittest". By mimicking this process, genetic algorithms are able to "evolve" solutions to real world problems. Genetic algorithms are implemented in a computer simulation in which a population of abstract representations (called chromosomes or the genotype of the genome) of candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem evolves toward better solutions. A typical genetic algorithm requires a genetic representation of the solution domain, and a fitness function to evaluate the solution domain. The fitness function is defined over the genetic representation and is the measures of the quality of the represented solution. The major disadvantage of GA is it's high time complexity, however it always provides an answer and the answer gets better with time. The different steps in this phase are presented in Figure 4. The population size is taken large enough to ensure a diverse set of population to avoid being stuck in a local minimum. A chromosome represents a topology of NoC architecture and each gene within a chromosome represents a particular traffic characteristic and the paths which carry the load of this traffic characteristic.

Dijkastra algorithm is used to determine the shortest basic tree path for each gene. The use of up*/down* routing ensures free paths. In the next step, mutation is performed on the population set in two ways: one in which lightly loaded links are removed thus resulting in the reduction of topology, and in the other the links are added to relieve heavily loaded links thus resulting in expansion of topology. Here the fitness value of a chromosome is with respect to communication load. Thus mutation results in a better distribution of traffic

by splitting it among multiple paths. In the next step two chromosomes undergo crossover to produce two new individuals which replace the parent chromosomes if they have a higher fitness value with respect to both communication load and energy consumption. In the last step mutation is performed to optimize energy consumption. The best chromosome obtained after performing the Genetic Algorithm for a large number of generations represents the desired topology. The output of this phase is a congestion conscious topology of the NoC for high performance demanding multimedia application.

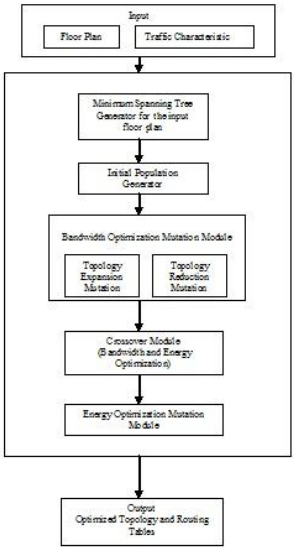


Fig 4: Control Flow Diagram of congestion conscious custom NoC generation for multimedia application

4.2 Energy Conscious Design for Multimedia Application

This methodology is similar to the methodology proposed in section 4.1 with the exception that here shortest energy path based topology is designed according to the Manhattan distance and in place of multiple deadlock free deterministic path, a shortest energy path is used by default and in case of congestion or deadlock a deadlock free escape path is utilized.

The Control flow diagram for the energy conscious NoC design methodology for multimedia application is exhibited in Figure 5. In the input module of Figure 5, the traffic characteristic and floor plan comprising of coordinates for the placement of cores is obtained from the floorplanner for the desired high performance demanding multimedia application. The floorplan information from the input module is utilized to generate energy conscious topology of the NoC for high performance demanding application. The length of links (e_{max}) between two tiles is restricted to certain value so as to avoid clock skew; links falling out of range are discarded. Further, to prevent any core from being the hot spot of energy consumption, each core is restricted with a maximum of 4 ports (nd_{max}) ie is each core can directly be connected to at-most 4 neighbours. By implementing dijkstras algorithm, shortest energy path from a specified source to destination core is evaluated. Component generator module searches for the components of the networks. A component is a collection of core and channel which are reachable from one another. The Merge module joins the various components of the topology/network by using a possible minimum distance channel. Therefore with the help of merge module a fully connected network/topology is obtained. Further to ensure that all cores are connected and a deadlock free up*/down* are available for each source-destination pair, a minimum spanning tree is constructed using Prim's Algorithm.

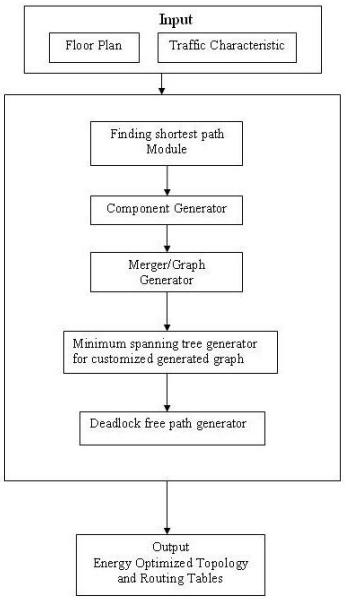


Fig 5: Control Flow Diagram of energy conscious custom NoC generation for multimedia application for each Chromosome

The Minimum spanning tree is used to mark all the channels as up or down. Based on the up*/down* rule of the up*/down* routing the shortest deadlock free energy path for all the souce-destination pairs of the network cores are discovered using dijkastra's algorithm. This path basically acts as escape paths for the shortest paths to avoid deadlocks and congestion on the shortest paths. The proposed methodology not only reduces the energy requirement by following the shortest energy paths for high performance demanding multimedia applications but also avoids deadlocks by providing deadlock free escape paths from each core. A genetic algorithm based design is used for the proposed methodology. The initial population is a large collection of chromosomes of the shortest energy paths. The chromosomes of initial population are mutated and crossover to find the best order of the genes (traffic characteristic) for achieving the topology with best communication energy requirements. The constraints nd_{max} and e_{max} are firmly kept throughout the design process.

5. EXPERIMENTAL SETUP AND RESULTS

In order to obtain Congestion Conscious Customized Network-on-Chip design methodology (referred as ***CCC-NoC***) and Energy Conscious Customized Network-on-Chip design methodology (referred as *ECC-NoC*) genetic

algorithm was run for 1000 generation with population size of 500. In *CCC-NoC* (*ECC-NoC*) mutations are done on 45% (50%) of the population and crossover on 35% (40%) of the population in each generation. NoC performance simulator ***IrNIRGAM*** [14] was run for 10000 clock cycles with applied packet injection interval to evaluate the NoC performance for multimedia application with desired traffic load. The dynamic communication energy consumption by router in transmitting a bit is evaluated using the power simulator orion [15] for 0.18µm technology. Moreover the dynamic bit energy consumption for inter-node links (El_{bit}) can be calculated using the following equation.

$$El_{bit} = (1/2) \times \alpha \times C_{phy} \times V_{DD}^2$$

Where α is the average probability of a 1 to 0 or 0 to 1 transition between two successive samples in the stream for a specific bit. The value of α can be taken as 0.5 assuming data stream to be purely random. C_{phy} is the physical capacitance of inter-node wire under consideration for the given technology and V_{DD} is the supply voltage. The proposed methodologies were evaluated for multimedia application on NoC regular topology based NoC as well as irregular custom designed NoCs. For NoC, the permitted channel length (e_{max}) and permitted node/core degree (nd_{max}) are assumed as 2 times the length of the core/node and 4 respectively. For *CCC-NoC* deterministic deadlock free *up*/down** routing for communication is assumed whereas for *ECC-NoC* escape path based *up*/down** routing was used. However for regular 2D-Mesh based NoCs, XY (deterministic) and OE (odd-even: adaptive) routing function were assumed.

5.1 Performance of *CCC-NoC* and *2D-Mesh* for MMS

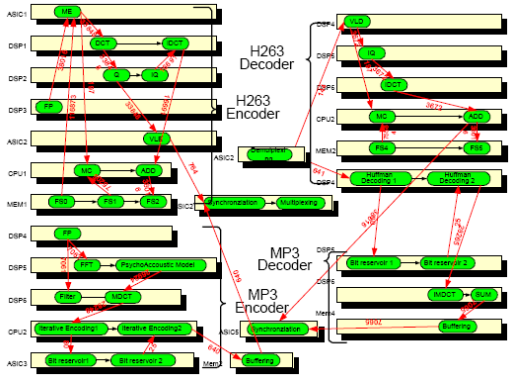


Fig 6: MMS Communication Trace Graph

The proposed *CCC-NoC* was compared with regular 2D-Mesh NoC for a realistic multimedia application (referred as *MMS*) *MMS* is an integrated video/audio system which includes an h263 video encoder, an h263 video decoder, an mp3 audio encoder and an mp3 audio decoder. The application was partioned into 40 distinct tasks and then these tasks were assigned and scheduled onto 25 selected IPs. These IPs range from DSPs, generic processors, embedded DRAMs to customized ASICs. The communication trace graphs as shown in Figure 6 for the same were obtained from the work presented by Hu et al. [6].

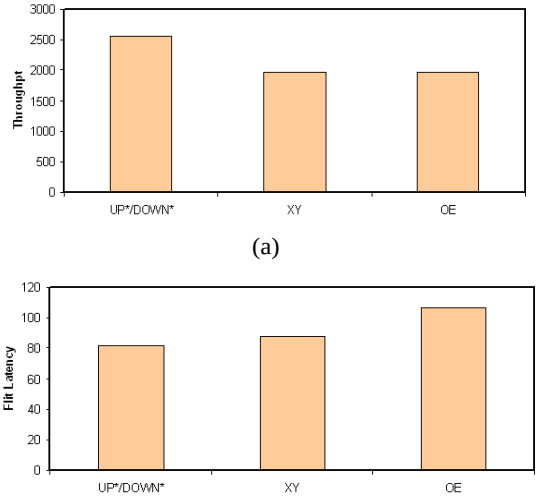


Fig 7: Performance results of *CCC-NoC* (*up*/down) and regular *2D-Mesh* on realistic *MMS benchmark* for (a) throughput (in flits) and (b) flit latency (in clocks)**

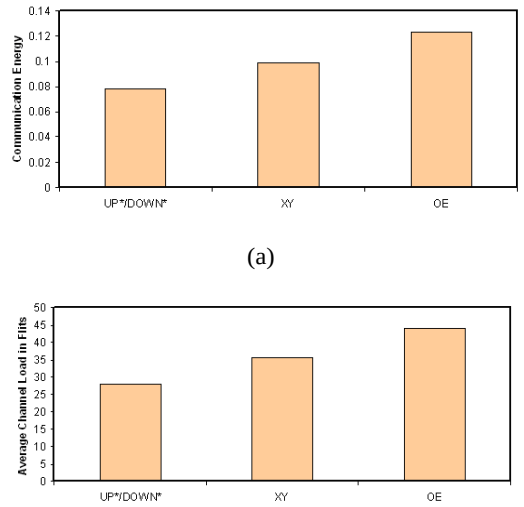


Fig 8: Comparison of traffic distribution across the channels of *CCC-NoC* (*up*/down) and regular *2D-mesh* on realistic MMS benchmark (a) average per channel communication energy consumption (in pico joules) (b) average per channel communication traffic load (in flits)**

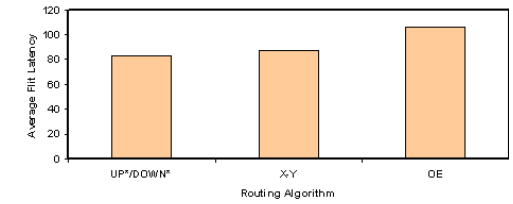
Figure 7 shows that CCC-NoC with up*/down* routing function exhibit an increase in throughput of 30.5% and reduction in average flit latency of 6% and 22.8% in comparison to 2D-Mesh with XY and OE routing respectively for MMS application. Moreover the CCC-NoC exhibits better distribution of application specific traffic across the channels of the generated topology as is evident from Figure 8(a, b).

Figure 8(a, b) exhibits the effect of traffic load and energy distribution across the channels of the network for 1000 injected flits into the NoC according to the application

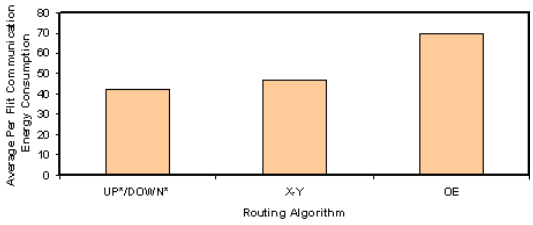
requirement. CCC-NoC with up*/down* routing function shows reduction in average traffic load per channel of 22% and 36.8% and reduction in average per unit length communication energy consumption per channel of 21.2% and 36.6% in comparison to 2D-Mesh with XY and OE routing respectively for the realistic MMS application.

5.2 Performance of *ECC-NOC* and *2D-Mesh* for MMS

The performance of *ECC-NoC* was analyzed in comparison to the regular *2D-mesh NoC* for a high performance demanding multimedia application (MMS). The communication traffic characteristics of the MMS was assumed according to the Figure 6. Figure 9 shows reduction in latency on average of 4.2 clocks and 22.5 clocks and reduction in average per flit communication energy of 9% and 39% in comparison to 2D-Mesh with XY and OE routing respectively in support of *ECC-NoC* methodology.



(a)



(b)

Fig 9: *ECC-NoC* and *2D-Mesh* performance comparison for *MMS* (a) Average flit latency (in clock cycles) and (b) Average communication energy consumption per flit (in pico joules)

6. CONCLUSION

In the presented work, the proposed custom NoC were compared with regular NoC for high performance multimedia application. The proposed methodologies show significant improvement in performance for multimedia application. In future we plan to extend the proposed methodologies for several of high performance demanding applications.

7. REFERENCES

[1] Dally. W. J. and Towles. B. 2001. Route packets, not wires: on-chip interconnection networks, In IEEE Proceedings of 38th Design Automation Conference (DAC), 684–689.

[2] Benini. L. and De Micheli. G. 2002. Networks on chips: A new SoC paradigm. Journal of IEEE Computer, vol. 35, no. 1, 70–78.

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

[4] Choudhary. N, Gaur. M. S, Laxmi V. and Singh. V. Fast Energy Aware Application Specific Network-on-Chip Topology Generator, Proceedings of IEEE International Conference on Advance Computing, Patiala, India, pp. 250-255, Feburary 19-20, 2010.

[5] Choudhary. N, Gaur. M. S, Laxmi V. and Singh. V. Genetic Algorithm Based Topology Generation for Application Specific Network-on-Chip, Proceeding of IEEE International Symposium on Circuits and Systems (ISCAS) 2010, Paris, France, pp. 3156-3159, May, 30 - June, 2, 2010.

[6] Hu. J. and Marculescu. R. 2003. Energy-aware mapping for tile-based NOC architectures under performance constraints. In proceedings of ASP-DAC.

[7] Dick. R. P., Rhodes. D. L. and Wolf. W. 1998. TGFF: Task graphs for free. In proceedings of international workshop on Hardware/Software Codesign.

[8] Jain. L., Al-Hashimi. B. M., Gaur M. S., Laxmi. V. and Narayanan. A. 2007. In proceedings of NIRGAM: A simulator for NoC interconnect routing and application modeling. In proceedings of DATE.

[9] Chang. Y. C., Chang Y. W., Wu G. M. and Wu. S. W. 2000. B*-Trees: A new representation for non-slicing floorplans. In Proceedings of 37th Design Automation Conference, 458-463.

[10] Srinivasan. K. and Chatha. K. S. 2006. Layout aware design of mesh based NoC architectures. In proceedings of 4th international conference on Hardware Software Codesign and System Synthesis, 136-141.

[11] Schroeder. M.D. *et al.* 1991. Autonet: A high-speed self-configuring local area network using point-to-point links. Journal of Selected Areas in Communications, vol. 9.

[12] Silla. F. and Duato. J. High-performance routing in networks of workstations with irregular topology, *IEEE Transactions on Parallel and Distributed Systems*, vol. 11, pp. 699-719, July 2000.

[13] Eiben. A. E. and Smith. J. E. 2003. Introduction to Evolutionary Computing, Springer-Verlag, Berlin, Heidelberg.

[14] Choudhary. N, Gaur. M. S. and Laxmi V. "*Irregular NoC Simulation Framework: IrNIRGAM*", IEEE International conference on Emerging Trends in Networks and Computer Communications (ETNCC 2011), April 22-24, 2011, Udaipur, India

[15] Wang. H-S. *et al.* 2002. Orion: A power-performance simulator for interconnection network. In proceedings of international symposium on Microarchitecture.