

Communication Bandwidth Adaptable Network Design of Complex Application Specific SoC

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

The communication interconnects among the cores of the futuristic SoC is a vital challenge. NoC is being proposed as the appropriate solution for addressing these communication challenges of complex SoCs. To address design complexity and reuse, NoC systems are typically desired to be built from pre-designed and pre-verified homogenous or heterogeneous building blocks such as programmable RISC cores, DSPs, memory blocks. However most application specific SoC are special-purpose and are tailored to the domain-specific requirements of the desired application, which communicate in a very specific, mostly irregular way. In this work, a methodology for the design of communication centric customized irregular network infrastructure of SoC is proposed. The proposed methodology exploits a priori knowledge of the application's communication attributes to generate an optimized network and associated routing tables to enable sufficient number of deadlock free paths for enhanced communication traffic and energy distribution across the network infrastructure of the SoC. In the proposed methodology the network is generated according to the requisite deadlock free paths having appropriate distribution of communication traffic.

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

SoC (System-on-Chip) architecture consists of numerous cores which may be processors, DSPs, memory block etc. The use of standard hardwired busses to interconnect these cores is not scalable. As the systems grow and design cycle time requirements decreases, the need for more generalized solution becomes pressing. To overcome this problem, Network-on-Chip (NoC) [1]-[3] is being proposed as the interconnect solution for the futuristic SoC of nanoscale regime. The NoC as an interconnection network for nanoscale SoC has several advantages such as better structure, performance and modularity to name a few. Several early works [2], [4] advocated the use of standard topologies such as meshes, tori, or fat trees under the hypothesis that the wires can be well structured in such networks. These networks are adequate for generic SoCs where the communication traffic attributes of the application cannot be predicted statically. Regular NoCs are also famous for

supporting design reuse. However even in custom NoC, the switch architecture itself can be kept regular and so can be easily parameterized (on number of ports, width of physical links, etc). Moreover, for most SoCs the system is designed with static or semi-static mapping of tasks to hardware resources/cores and hence the communication traffic characteristics of such SoC can be well characterized at design time. Therefore it is expected that networks with application specific irregular structure/topology tailored to the application's communication requirements and supporting deadlock free communication to have an edge over the NoC with regular network as their communication infrastructure.

The admired turn prohibition [5] based deadlock free topology independent routing algorithms are up*/down* [6], Left-Right [7], L-turn [7], and down/up [8]. In the proposed, design a *genetic algorithm* based methodology is developed to facilitate the generation of customized communication centric irregular network/topology for the NoC along with routing tables for enhanced performance, deadlock free communication and improved traffic load and energy distribution across the NoC.. A brief account of related previous work is presented in Section 2. Application specific communication model and architecture for SoC is presented in Section 3. The proposed methodology based on *Genetic Algorithm* for generation of optimized communication bandwidth adaptable NoC is presented in Section 4. Section 5 summarizes the experimental results followed by a brief conclusion in section 6.

2. PREVIOUS WORK

Several recent surveys [2], [4], [9], on NoCs provide pointers to recent research and development. Methods to collect and analyze traffic information that can be fed as input to the bus and NoC design processes have been presented in [10], [11]. Mappings of cores onto standard NoC topologies have been explored in [12]-[14]. In [12], [14] a floorplanner is used during the mapping process to get area and wire-length estimates. These works only select from a library of standard topologies, and cannot generate a fully customized topology. In [13], a unified approach to mapping, routing and resource reservation has been presented. However, the work does not explore the topology design process. Important research in macro networks has considered the topology generation problem [15]. As the traffic patterns on these networks are difficult to predict most approaches are tree-based (like spanning or Steiner trees) and only ensure connectivity with node degree constraints [15]. These techniques cannot be directly extended to address the NoC synthesis problem.

Application-specific custom topology design has been explored in [16]-[18], [19], [20]. The works from [16]-[17], do not consider the floorplanning information during the topology design process. In [20], a floorplanner is used during topology design to reduce power consumption on wires. It does not consider the area and power consumption of switches in the design. Also, the number and size of network partitions are manually fed. In [18], a slicing tree based floorplanner is used during the topology design process. This work assumes that the switches are located at the corners of the cores and it does not consider the network components (switches, network interfaces) during the floorplanning process. Moreover the actual sizes of the cores in [18], [19] are considered only after generating their relative positions. The resulting floorplan can be extremely area inefficient when compared to the standard floorplanning process. A range of issues in the design methods and tools for efficient synthesis of application specific Network-on-Chip interconnect for 3D SoC were addressed in [21]- [22]. At the research frontier, the focus is now on bridging the gap between high level analysis of communication requirements and synthesis of customized NoC topology. The methodology proposed in this work address the issue of topology/network design with deadlock free communication for application specific homogenous or heterogeneous NoC according to communication requirements. The proposed methodology accepts application's communication characteristics and floorplanning information as input. Therefore this methodology is especially suitable for applications where optimized placement of cores in chip layout during floorplanning based on metric such as area is done in advance with highest priority.

3. APPLICATION SPECIFIC COMMUNICATION MODEL AND ARCHITECTURE FOR SoC

3.1 Application Specific SoC Communication Model

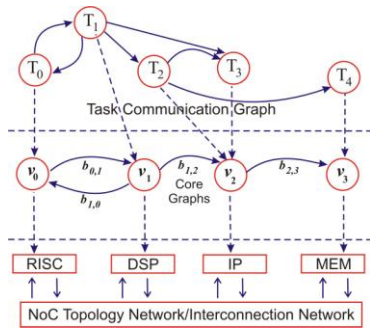


Fig 1: Application specific communication model in NoC

The generic communication model including *Task graphs* [23]-[24], *Core Graph*, and *NoC topology/interconnection network* is shown in Figure 1.

Definition 1: *Core Graph* is a directed graph, $G(V, E)$ with each vertex $v_i \in V$ representing an IP core and a directed edge $e_{i,j} \in E$, representing the communication between the

cores v_i and v_j . The weight of the edge $e_{i,j}$ denoted by $b_{i,j}$, represent the desired average bandwidth requirement of the communication from v_i and v_j .

Definition 2: *NoC topology graph* is a directed graph $N(U, F)$ with each vertex $v_i \in U$ representing a node/tile in the topology and a directed edge $f_{i,j} \in F$ represents direct communication channel between vertices v_i and v_j . Weight of the edge $f_{i,j}$ denoted by $Ab_{i,j}$ represents the available link/channel bandwidth across the edge $f_{i,j}$.

3.2 Irregular NoC Simulation Architecture

A discrete event, cycle accurate simulator IrNIRGAM is used for Irregular NoC. IrNIRGAM is an extension of NIRGAM [25]-[26]. NIRGAM is a cycle-accurate SystemC based simulator for regular NoC. IrNIRGAM supports irregular topology framework with table based routing. In IrNIRGAM, input buffered routers can have multiple virtual channels (VCs) and uses wormhole switching for flow control. A round-robin scheme for switch arbitration is used in the router nodes.

3.3 Chip Layout, NoC Energy Model and 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 floorplanning tools such as B*-Trees [27].

The energy model [23] for the Network-on-Chip is defined as follows:

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

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 (1) 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*. Equation (1) can be rewritten as

$$E_{bit}(t_i, t_j) = n_{hops} \times Er_{bit} + \sum_{k=1}^{n_{hops}-1} El_{bit}^k \quad \text{-----(2)}$$

In this paper deterministic version of *up*/down** [6] and *Left-Right* [7] routing are used for providing deadlock free communication.

4. COMMUNICATION BANDWIDTH ADAPTABLE NETWORK DESIGN FOR SoC

Based on the information from chip layout and traffic characteristics of the application, the global and detailed physical routes for the customized NoC are generated using the proposed methodology assuming over the cell routing [28]. Irregular topology construction is initiated by creating a minimum spanning tree (MST) based on *Manhattan distance* among the IP cores with root as the node/core having

maximum communication requirement. Moreover the *permitted node degree* ($nd_{tree,max}$), i.e., number of allowed ports per IP core in initial stage of the methodology is kept less than the actual *permitted node degree* (nd_{max}) to allow better search space for valid shortcuts. The *MST* helps in classifying all the channels of the topology as “up” (Left) or “down” (Right) in addition to making the initial topology strongly connected, providing a path between every pair of nodes. In the next phase of the methodology a genetic algorithm [29] based heuristic is used for the extended design of customized NoC/topology. Genetic algorithm [29] is a search technique used in determining exact or approximate solutions to optimization and search problems.

The generated customized NoC topology is expected to exhibit reduced congestion and average flit latency leading to increased throughput for the application specific injected traffic. In the proposed methodology the link/channel length is not allowed to exceed the *maximum permitted channel length* (e_{max}) due to constraint of physical signaling delay. The nodes of the generated topology are not allowed to exceed a given *maximum permitted node-degree* (nd_{max}). This constraint prevents the algorithm from instantiating slow routers with a large number of I/O-channels which would decrease the achievable clock frequency due to internal routing and scheduling delay of the router. Figure 2 briefly illustrates the proposed methodology. The *Genetic algorithm* formulation is explained as follows.

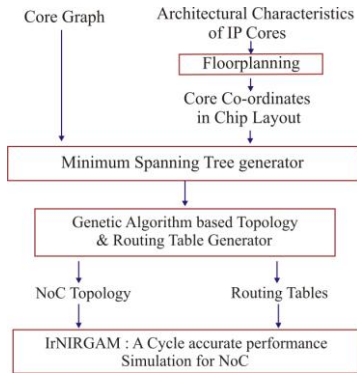


Fig 2: Communication centric NoC generation flow using genetic algorithm

4.1 Initial Population Generation

Modified dijkstra's shortest path algorithm [30] is used to find energy shortest deadlock free path in accordance to the up/down (Left-Right) rule in the NoC topology graph (MST in the initial topology). Routing table entries for the routers of the NoC is generated for each traffic characteristic (edge) in the Core Graph. At this stage the traffic load to these tree paths is assigned according to the bandwidth requirement of the traffic characteristics. Moreover to bring variety as well as to include the possible shortest deadlock free paths in the topologies of the initial population, a large number of genes of initial population are mutated by laying energy shortest deadlock free path in accordance to the up*/down* (Left- Right) rule with constraints nd_{max} and e_{max} firmly kept.

4.2 Solution Representation

In the proposed formulation, each chromosome is represented by an array of genes. Maximum size of the gene array to be equal to the number of traffic characteristics (i.e. edges) in the Core Graph, in other words a chromosome represents an instance of NoC topology and each gene represents a collection of deadlock free paths with upper limit of n (configurable parameter) for a traffic characteristic in the Core Graph along with necessary information for these paths. In each gene at least one path is the shortest energy path through the channels exclusively pertaining to MST, guarantying the connectivity between the source and destination pair of the gene (traffic characteristics).

4.3 Mutation

Three mutation operations called *Topology Extension*, *Topology Reduction*, and *Energy Reduction* with equal probability are applied in each generation of the genetic algorithm.

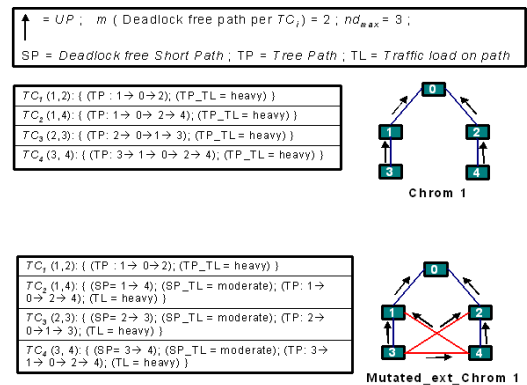


Fig 3: Example topology extension mutation operation in BA-TGM assuming $nd_{max} = 3$

4.3.1 Topology Extension Mutation

A random number of genes are picked from the selected chromosome and their paths along with assigned traffic load are analyzed. If a heavily loaded path (i.e. path having assigned bandwidth load greater than the preferred bandwidth load) is discovered, then a suitable shortcut channel is inserted in the topology for laying a new deadlock free path using this shortcut, according to the chosen routing function. However if the discovered path happens to be longer than the tree path then the shortcut is rejected and a new shortcut is tried. Moreover the excess traffic load of the selected path is transferred to the channels of the new path if it does not lead to overloading of the new path's channels otherwise the shortcut is rejected. Fig. 3 shows an example topology extension mutation.

4.3.2 Topology Reduction Mutation

A random number of genes are picked from the selected chromosome and their paths are analyzed. This mutation tries to remove the paths having very lightly loaded channels from the topology. Load of the path to be removed is transferred to an existing path of the gene having minimum load on its channels with the constraint that the average traffic load on the channels of the target path remains within permissible limits. Moreover with low probability a path of the selected gene is randomly

removed and its load is transferred to the existing path accordingly. However channels belonging to MST are not allowed to be removed. Fig. 4 shows an example topology reduction mutation.

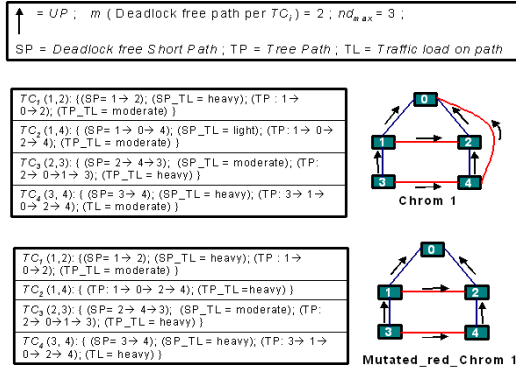


Fig 4: Example topology reduction mutation operation in BA-TGM assuming $nd_{max} = 3$

4.3.3 Energy Reduction Mutation

This mutation is done on randomly selected chromosome with bias towards the Best Class of the population in each generation. In this mutation a replacement shorter energy path for each path of the gene of the chromosome is attempted to be discovered with help of inclusion of appropriate shortcuts in the topology. Fig. 5 shows an example energy reduction mutation.

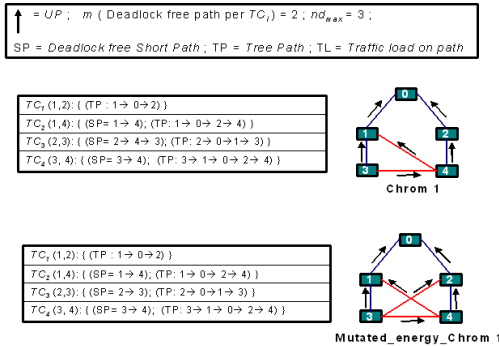


Fig. 5. Example energy reduction mutation operation in BA-TGM assuming $nd_{max} = 3$

4.4 Crossover

Crossover is done on a large size of the population with the bias towards the Best Class of the chromosome population. For achieving crossover of two chromosomes, a random crossover point is selected and then genes of these chromosomes are mixed over the crossover point to produce two new chromosomes. The new chromosome is accepted only if its corresponding topology satisfy constrain of nd_{max} . Moreover the new chromosome should have valid channels available to satisfy all the paths/routes of the chromosome. Fig. 6 shows an example crossover operation.

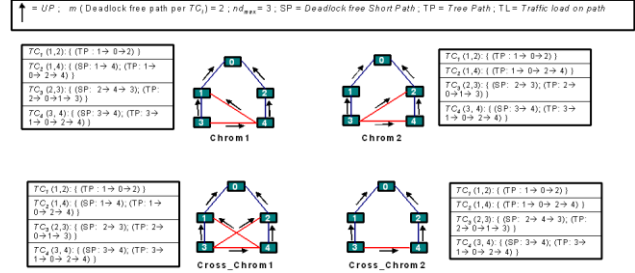


Fig 6: Example crossover operation in BA-TGM assuming $nd_{max} = 3$

4.5 Measure of Fitness

The fitness measure essentially has two components - (1) *average bandwidth requirement overflow* in comparison to preferred bandwidth load, (2) *dynamic communication energy requirement* of the traffic for the customized NoC. The fitness (cost) function can be formulated as under.

$$Cost_i = \alpha \times (Ec_i / X_1) + (1 - \alpha) \times (Bc_i / X_2)$$

Where X_1 be maximum chromosome dynamic communication energy requirement, X_2 be maximum possible bandwidth requirement of a channel, Ec_i is the energy requirement, Bc_i is the average bandwidth requirement overflow per channel for chromosome c_i and α is an empirical constants.

Through exhaustive experimentation, the optimum value of α was determined as 0.4. Fitness of chromosome is regarded as high if its cost approaches zero. It may be noted that, the best 20% chromosomes (referred as *Best Class*) at any generation are directly transferred to the next generation, so as not to degrade the solution between the generations. After genetic algorithm methodology is made to run for a requisite number of generations, the chromosome with the *best Cost* is selected as the output chromosome. The NoC topology along with routing tables corresponding to the best chromosome is accepted as the customized application specific customized irregular NoC (*IrNoC*). Similarly the traffic load mapping (*directly proportional to the packet injection interval*) for the paths of the output chromosome is accepted as traffic load to path mapping for the NoC performance simulator.

5. EXPERIMENTAL RESULTS

The generated customized application specific topology was evaluated with respect to the performance metric such as throughput, latency, energy on the *IrNIRGAM* simulation framework. The communication tends to be highly irregular in such platforms because of the diversity of hardware components. In order to obtain a broad range of different irregular traffic scenarios, multiple *Core Graphs* were randomly generated using TGFF [24] with diverse bandwidth requirement of the IP cores. For generating application specific NoC topology the proposed genetic algorithm based methodology (referred as *BA-TGM*) was run for 1000 generation with population size of 500. The mutations are done on 45% of the population and crossover on 35% of the population in each generation. For performance comparison, the NoC simulator *IrNIRGAM* was run for 10000 clock cycles and network throughput in flits, average flit latency,

traffic load and energy distributions per channel were used as parameters for comparison. Network throughput is the number of flits received by various cores of the NoC during the simulation run. The flit latency determines the number of clock cycles it takes from entering the network until the reception at the destination node. All data queues in the network routers can buffer eight flits per channel. Further traffic load per channel and energy per channel exhibits the traffic load in flits per channel and communication energy consumed per channel respectively for the simulation run. The dynamic communication energy consumption by router in transmitting a bit is evaluated using the power simulator orion [31], [32] for 0.18 μ m 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 *BA-TGM* was compared on various experiment sets including a realistic multimedia application as described in Sub-section *Experimental Results* with *permitted channel length* (e_{max}) taken as 2 times the length of the core/node and *permitted node/core degree* (nd_{max}) of 4 and/or 6. The experimental results presented in Sub-section *Experimental results* summarizes the comparative performance results averaged over 50 generated irregular NoCs/topologies (*IrNoC-BA-TGM*) for number of cores varying between 16 to 81. For *IrNoC*, table based *up*/down** and/or *Left-Right* routing supporting deterministic deadlock free routing function were used. Moreover in the comparative experiments results the tile sizes and task to core/tile mapping are kept same for the compared methodologies so as to keep the comparison fair.

5.1 BA-TGM and 2D-Mesh for Random Benchmarks

Fig. 7 – Fig. 10 summarizes the averaged (over 50 irregular NoCs) comparative performance results of irregular customized topologies (*IrNoC-BA-TGM*) with permitted node/core degree (nd_{max}) of 4 and 2D-Mesh with XY [33] and OE (odd-even) [33] routing. Fig. 7 shows the comparative results regarding throughput and average flit latency of *IrNoC-BA-TGM* with *up*/down** routing and Left-Right routing in comparison to 2D-Mesh with XY and OE routing. *IrNoC-BA-TGM* with *up*/down** (Left-Right) routing function shows on average an increase in throughput of 32.3% (30.2%) and reduction in average flit latency of 11.3% (9%) and 28% (26%) in comparison to 2D-Mesh with XY and OE routing respectively.

In the proposed *BA-TGM* the first priority is for better traffic load distribution among the channels of the generated topology and in doing so it tries to keep the traffic paths as short as possible as second priority. Therefore it is possible that the irregular topology (*IrNoC*) generated with the proposed methodology (*BA-TGM*) may not constitute the best shortest energy paths and so the average communication

energy consumed by the flits reaching their destination may not be minimal as is evident from the Figure 8.

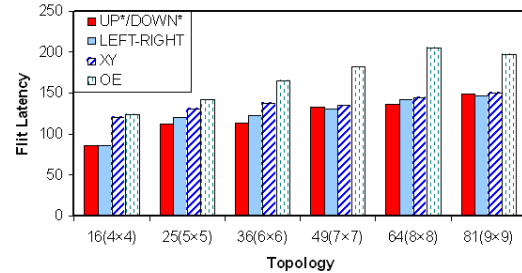
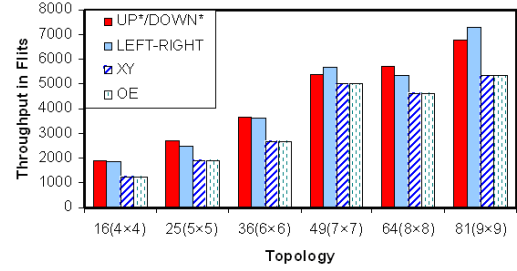


Fig 7: Performance results of *IrNoC* and 2D Mesh for throughput (in flits) and average flit latency (in clocks)

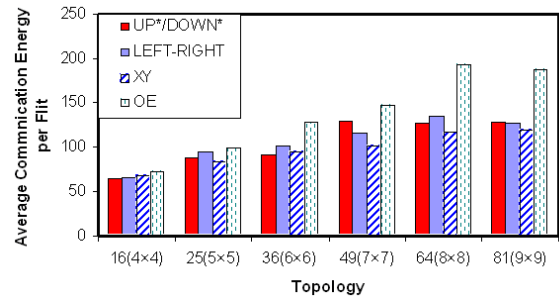


Fig 8: Comparison of average communication energy consumed by flits in reaching their destination for *IrNoC-BA-TGM* and 2D-Mesh

However the *BA-TGM* scores in performance by better distribution of application specific traffic across the channels of the generated topology as is evident from Fig. 9 and Fig. 10. Fig. 9 and Fig. 10 exhibit the effect of energy and traffic load distribution across the channels of the topology for 1000 injected flits into the NoC according to the application requirement.

The *IrNoC-BA-TGM* consistently shows reduced average traffic load (in flits) per channel and reduced average per unit length communication energy consumption per channel in comparison to *2D-Mesh*. *IrNoC-BA-TGM* with *up*/down** (*Left-Right*) routing function shows average reduction in traffic load per channel of 30.7% (28%) and 46.7% (44.5%) and reduction in average per unit length communication energy consumption per channel of 31.7% (29%) and 46.7% (44.5%) in comparison to *2D-Mesh* with XY and OE routing respectively.

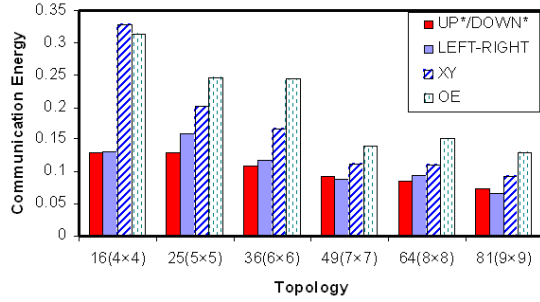


Fig 9: *IrNoC* and *2D-Mesh* comparison of average per channel communication energy consumption (in pico joules)

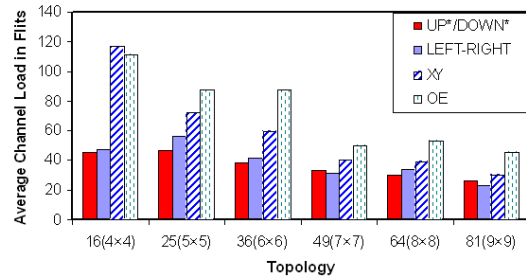


Fig 10: *IrNoC* and *2D-Mesh* comparison of average per channel communication traffic load (in flits)

5.2 BA-TGM and 2D-Mesh with Intelligent Task to Core Mapping

In [23], a methodology to intelligently map the IPs/Cores onto the *2D-Mesh* regular NoC with the objective to minimize total communication energy was proposed. Fig. 11 – Fig. 13 summarizes the averaged (over 50 irregular NoCs) comparative performance results of irregular customized topologies (*IrNoC-BA-TGM*) with *permitted node/core degree* (nd_{max}) of 4 and *2D-Mesh* with intelligent task to core (tile) mapping as proposed in [23]. Fig. 11 shows the comparative results regarding throughput and average flit latency of *IrNoC-BA-TGM* with up*/down* routing in comparison to *2D-Mesh* with XY and OE routing. *IrNoC-BA-TGM* with up*/down* routing function shows on average an increase in throughput of 22.3% and reduction in average flit latency of 11.2% and 19% in comparison to *2D-Mesh* with XY and OE routing respectively.

In this experiment set *IrNoC-BA-TGM* showed more consistent and uniform improvement in latency as well as better traffic load and energy distribution over the channels of the generated topology across various sized (16 to 81 node) topologies because in such scenario: 1) The application's communicating cores tends to be near to each other in the chip layout leading to discovery of better short energy paths by the proposed methodology (*IrNoC-BA-TGM*), 2) *IrNoC-BA-TGM* will be able to generate topologies with better traffic load and energy distribution as communication requirement of the application are expected to be localized in the various regions of the generated topology/NoC due to intelligent task to core mapping.

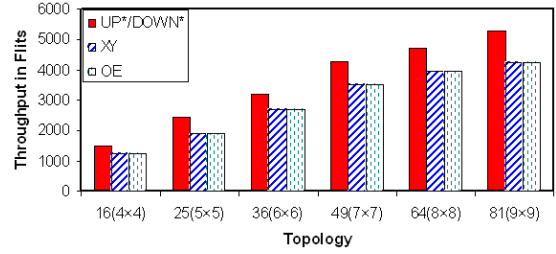


Fig 11: Performance results of *IrNoC* and *2D Mesh* with mapping as proposed in [23] for throughput (in flits) and flit latency (in clocks)

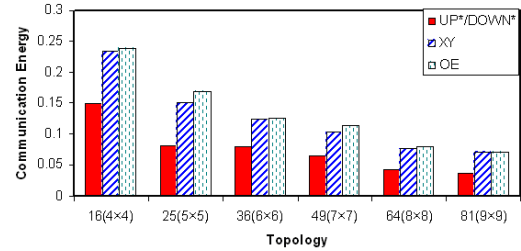
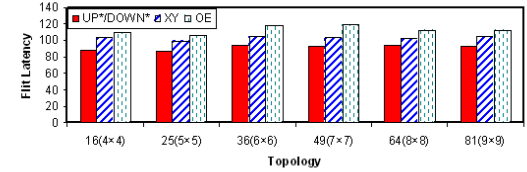


Fig 12: Comparison of average per channel communication energy consumption (in pico joules) for *IrNoC* and *2D-Mesh* with mapping as proposed in [23]

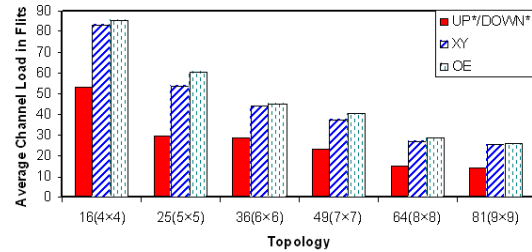


Fig 13: Comparison of average communication traffic load (in flits) for *IrNoC* and *2D-Mesh* with mapping as proposed in [23]

As in previous cases the average communication energy consumed by the flits reaching their destination tends to be more in *BA-TGM* in comparison to *2D-Mesh*. However the *BA-TGM* scores in performance by better distribution of application specific traffic across the channels of the generated topology as is evident from Fig. 12 and Fig. 13. Fig. 12 and Fig. 13 exhibit the effect of traffic load and energy distribution across the channels of the topology for 1000 injected flits into the NoC according to the application requirement. The *IrNoC-BA-TGM* consistently shows a

relatively (as compared to experiments with random benchmarks) uniform reduction in average per unit length communication energy consumption per channel and a uniform reduction in average traffic load (in flits) per channel in comparison to *2D-Mesh*. *IrNoC-BA-TGM* with *up*/down** routing function shows average reduction in traffic load per channel of 40.6% and 43.6% and reduction in average per unit length communication energy consumption per channel of 41% and 44% in comparison to *2D-Mesh* with XY and OE routing respectively.

5.3 BA-TGM and 2D-Mesh for Random Benchmarks with Increased Available Ports

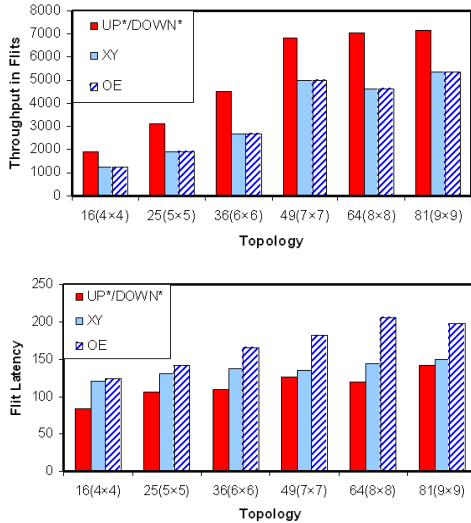


Fig. 14: Performance results of *IrNoC* and *2D Mesh* with $nd_{max} = 6$ for throughput (in flits) and flit latency (in clocks)

Fig. 14 – Fig. 16 summarizes the averaged (over 50 irregular NoCs) comparative performance results of irregular customized topologies (*IrNoC-BA-TGM*) with permitted node/core degree (nd_{max}) of 6 and *2D-Mesh* with XY and OE (odd-even) routing. The nd_{max} of 6 is assumed in anticipation that the increase in permitted node/core degree will not only help the BA-TGM to find better short energy paths for the given application but the increased availability of valid channels will also help in generating the topology/NoC with better traffic load and energy distribution across the channels of the generated topology leading to improved performance. Fig. 14 shows the comparative results regarding throughput and average flit latency. *IrNoC-BA-TGM* with *up*/down** routing function shows on average an increase in throughput of 51.5% and reduction in average flit latency of 16.5% and 32% in comparison to *2D-Mesh* with XY and OE routing respectively.

As explained in previous experiment sets the average communication energy consumed by the flits reaching their destination may not be minimal in *IrNoC-BA-TGM* but the BA-TGM scores in performance by better distribution of application specific traffic across the channels of the generated topology as is evident from Fig. 15 and Fig. 16.

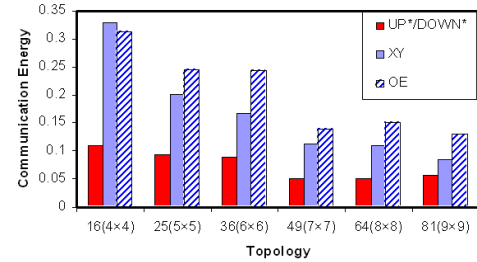


Fig 15: *IrNoC* with $nd_{max} = 6$ and *2D-Mesh* comparison of average per channel communication energy consumption (in pico joules)

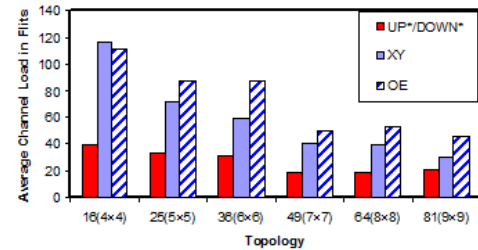


Fig 16: *IrNoC* with $nd_{max} = 6$ and *2D-Mesh* comparison of average per channel communication traffic load (in flits)

Fig. 15 and Fig. 16 exhibit the effect of energy and traffic load distribution across the channels of the topology for 1000 injected flits into the NoC according to the application requirement. The *IrNoC-BA-TGM* consistently shows reduced average traffic load (in flits) per channel and reduced average per unit length communication energy consumption per channel in comparison to *2D-Mesh*. *IrNoC-BA-TGM* with *up*/down** routing function shows average reduction in traffic load per channel of 51.1% and 62.3% and reduction in average per unit length communication energy consumption per channel of 51.5% and 62.7% in comparison to *2D-Mesh* with XY and OE routing respectively. The distribution of traffic load and communication energy across the channels of the *IrNoC* proved to be better in comparison to previous example set.

6. CONCLUSION

In the presented work, a genetic algorithm based methodology was implemented to tailor the congestion aware network topology in NoC according to the communication requirements of the application captured in the *Core Graph* according to the chosen routing algorithm. All the routes generated through the presented methodology are kept deadlock free. Additionally the presented methodology is adaptable according to any routing function where generic routing rules can be enforced for the routing function. It is believed that the combined treatment of the routing algorithm and topology generation as done in the presented methodology offers a huge potential of performance optimized future application-specific NoC architectures.

7. REFERENCES

- [1] W.J. Dally and B.Towles, "Route Packets, Not Wires: On-Chip Interconnection Networks," IEEE Proceedings of 38th Design Automation Conference (DAC), 2001, pp. 684–689.
- [2] L. Benini and G. De Micheli., "Networks on Chips: A New SoC Paradigm," journal of IEEE Computer, vol. 35, no. 1, January 2002, pp. 70–78.
- [3] International Technical Roadmap for Semiconductors, <http://public.itrs.net/>, 2004.
- [4] S. Kumar, A. Jantsch, J.-P. Soininen, M. Forsell, M. Millberg, J. Oberg, K. Tiensyrja, and A. Hemani, "A Network on Chip Architecture and Design Methodology," Proc. of VLSI Annual Symposium (ISVLSI 2002), 2002, pp. 105–112.
- [5] C. Glass and L. Ni, "The Turn Model for Adaptive Routing," Proc. 19th International Symposium on Computer Architecture, May 1992, pp. 278– 287.
- [6] M.D. Schroeder et al. "Autonet: A High-Speed Self-Configuring Local Area Network Using Point-to-Point Links," Journal of Selected Areas in Communications, vol. 9, October 1991.
- [7] A. Jouraku, A. Funahashi, H. Amano and M. Koibuchi, "L-turn routing: An Adaptive Routing in Irregular Networks," Proceedings of International Conference on Parallel Processing, September 2001, pp. 374-383.
- [8] Y.M. Sun, C.H. Yang, Y.C Chung and T.Y. Hang, "An Efficient Deadlock-Free Tree-Based Routing Algorithm for Irregular Wormhole-Routed Networks Based on Turn Model," Proceedings of International Conference on Parallel Processing, vol. 1, August 2004, pp. 343-352.
- [9] U. Ogras, J. Hu and R. Marculescu, "Key Research Problems in NoC Design: A Holistic Perspective," IEEE CODES+ISSS, 2005, pp. 69-74.
- [10] K. Lahiri et al. "Design Space Exploration for Optimizing On-Chip Communication Architectures," IEEE TCAD, vol. 23, no.6, June 2004, pp. 952- 961.
- [11] S. Murali and G. De Micheli, "An Application-Specific Design Methodology for STbus Crossbar Generation," Proc. DATE 2005, 2005, pp. 1176-1181.
- [12] S. Murali and G. DeMicheli, "SUNMAP: A Tool for Automatic Topology Selection and Generation for NoCs," Proceeding of DAC, 2004.
- [13] A. Hansson et al. "A Unified Approach to Constrained Mapping and Routing on Network-on-Chip Architectures," Proceeding of ISSS, 2005, pp. 75-80.
- [14] S. Murali et al. "Mapping and Physical Planning of Networks on Chip Architectures with Quality-of-Service Guarantees," Proc. ASPDAC 2005, 2005.
- [15] R. Ravi et al. "Approximation Algorithms for Degree-Constrained Minimum Cost Network Design Problems," Algorithmica, vol 31, no. 1, 2001, pp. 58-78.
- [16] A. Pinto et al. "Efficient Synthesis of Networks on Chip". ICCD, October 2003, pp. 146-150.
- [17] W.H. Ho and T.M. Pinkston, "A Methodology for Designing Efficient On-Chip Interconnects on Well-Behaved Communication Patterns," HPCA, February 2003, pp. 377-388.
- [18] K. Srinivasan et al. "An Automated Technique for Topology and Route Generation of Application Specific On-Chip Interconnection Networks," Proc. ICCAD 2005, 2005.
- [19] K. Srinivasan and K.S. Chatha, "ISIS: A Genetic Algorithm based Technique for Custom On-Chip Interconnection Network Synthesis," Proceedings of 18th International Conference on VLSI Design, Kolkata, India, 2005, pp. 623-628.
- [20] T. Ahonen et al. "Topology Optimization for Application Specific Networks on Chip," Proc. SLIP 2004, 2004.
- [21] C. Seiculescu, S. Murali, L. Benini, and G. De Micheli, "SunFloor 3D: A Tool for Networks on Chip Topology Synthesis for 3D Systems on Chip," Proc. DATE 2009, 2009, pp. 9-14.
- [22] S. Murali, C. Seiculescu, L. Benini, and G. De Micheli, "Synthesis of Networks on Chips for 3D Systems on Chips," Proc. Asian and South Pacific Design Automation Conference, ASPDAC 2009, 2009, pp. 242-247.
- [23] J. Hu, and R. Marculescu, "Energy-Aware Mapping for Tile-based NOC Architectures under Performance Constraints," ASP-DAC 2003, Jan 2003.
- [24] R.P. Dick, D.L. Rhodes and W. Wolf, "TGFF: Task Graphs for Free," Proc. International Workshop on Hardware/Software Codesign, March 1998.
- [25] Lavina Jain, B.M. Al-Hashimi, M.S. Gaur, V. Laxmi and A. Narayanan, "NIRGAM: A Simulator for NoC Interconnect Routing and Application Modelling, Proc. DATE 2007, 2007.
- [26] <http://www.nirgam.ecs.soton.ac.uk> (Last viewed on November 10, 2010)
- [27] Y.C. Chang, Y.W. Chang, G.M. Wu and S.W. Wu, "B*-Trees: A New Representation for Non-Slicing Floorplans," Proc. 37th Design Automation Conference, 2000, pp. 458-463.
- [28] K. Srinivasan and K.S. Chatha, "Layout Aware Design of Mesh based NoC Architectures," Proc. of 4th International Conference on Hardware Software Codesign and System Synthesis, Seoul, Korea, 2006, pp. 136-141.
- [29] A.E. Eiben and J.E. Smith, Introduction to Evolutionary Computing, Springer-Verlag, Berlin, Heidelberg, 2003.
- [30] T. Cormen, C.Leiserson, and R. Rivest, Introduction to Algorithms, Prentice Hall International, 1990.
- [31] H-S Wang et al. "Orion: A Power-Performance Simulator for Interconnection Network," Proc. International Symposium on Microarchitecture, Nov 2002.
- [32] A.B. Kahng, B. Li, L.S. Peh and K. Samadi, "Orion 2.0: A Fast and Accurate NoC Power and Area Model for Early-Stage Design Space Exploration," Proc. DATE'09, 2009, pp. 423–428.
- [33] J. Duato, S. Yalamanchili and L. Ni, Interconnection Networks : An Engineering Approach, Elsevier, 2003.