



Evaluating Performance of Efficiently Mapped 3D-Mesh using NC-G-SIM Simulator

Kartika Vyas¹, Pranav Wadhvani²

^{1,2}Department of Computer Science and Engineering
Maharana Pratap University of Agriculture and Technology
Udaipur, Rajasthan
INDIA

Abstract: To meet the extensive demand of many-core era, research communities are developing and emerging with diverse NoC solutions which must provide scalable, efficient and reliable communication infrastructure. The arrival of 3D integration technology has the potential to enhance the applicability of all the existing 2D NoCs. Customized NoCs are the need of today's SoCs as they offer energy efficiency and enhanced performances because they are designed to support the specific application behavior. For the evaluation of performance and the impact of the innovative 3D NoC architectures, modeling and simulation tools are required which must provide accurate estimations in terms of quality of service metrics such as energy, latency, throughput etc. This paper, initially presents a heuristic based on branch and bound approach to smartly map applications onto the cores in 3D Mesh NoC architecture. This significantly improves the dynamic communication energy consumption as well as latency. Secondly, the simulator NC-G-SIM is designed using SystemC that is highly generic in nature that supports 3D Mesh, 2D-Mesh and Irregular interconnection networks. The cycle-accurate simulator design has been made modular and flexible and has been used to simulate and analyze the performance of the proposed mapping heuristic in terms of energy and latency.

Keywords: Branch & Bound, Distributed Table-based routing, Energy-efficient mapping, Irregular, Latency, LLC, NC-G-SIM, Simulator, ULC, 3D-Mesh, 2D-Mesh

I. Introduction

Development of the silicon technologies has made the integration of large number of cores possible on a single chip, offering a considerable amount of computation power. Shared or dedicated buses are currently being utilized to interconnect the communication of on-chip resources [1]. However, scalability is a major issue in these buses that needs to be eliminated to develop billion transistor chips. Other factors like energy consumption, power management, wire delays, central arbitration, signal integrity and design choices complicate the entire design operation thereby makes it more time consuming and also increases the time to-market.

To ameliorate these design challenges of the traditional interconnection mediums [2]. Network on-Chip has been proposed. This concept replaces global on-chip wiring and offers choices in selection in terms of structure, performance, and modularity. With Network on Chip approach, building blocks of a system such as processors, memories, peripherals, etc. are mapped on the cores that can communicate by sending packets to one another over the NoC network. The network on chip technology has brought a paradigm shift from computation centric design architecture to communication centric design architecture.

In the deep-sub-micron (DSM) era integration densities have gradually increased due to the continuous shrinking of feature sizes and this has led to the introduction of new challenges in the existing 2D NoC architectures such as increase in interconnect delays, path lengths, area overhead and many others. So, to palliate complex communication problems, 3D NoCs were thought of as a promising solution and has opened the paths for new opportunities in designing NoC architecture suitable for bulky SoCs providing higher efficiency compared to 2D integration.

The complex multimedia applications of current era, integrated in embedded into mobile systems are becoming more computation oriented as numerous functionalities have to be carried out by them. These mobile systems embedding the multimedia applications usually operate on battery and support an extensive range of other applications too hence; these systems must be energy-efficient. By optimizing the application to core mapping process, significant improvement in energy consumption as well as latency can be achieved. In order to verify and analyze the effectiveness of the implemented design architectures a concrete simulation framework is desired. The market is loaded with surplus number of simulators whose functionalities are constrained to cater the needs of specific domains of NoC. All the existing simulators are confined to support small sets of topologies and moreover, they are targeted to simulate specific architectures tailored to satisfy the necessities of

the designers of the particular application domain and so these are not applicable for other architectures. This raises the need for a generic simulation tool to efficiently explore the wide design space [28-31] since, design choices and decisions about NoC architectures are typically taken on the basis of simulation before resorting to implementation since it is cheaper and more flexible.

The rest of the paper is arranged as follows. Section II discusses the related work done to gain knowledge to achieve our objectives; Section III presents a Branch-and-Bound heuristic which includes description of 3D NoC architecture and its associated energy-model and explains the detailed work-flow of the proposed heuristic. Section IV describes the simulation framework design, general architecture as well as the functionality of the NC-G-SIM (NoC-Generic-SIMulator) simulator and Section V presents the experimental results showing comparison between optimized and random mapping on 3D Mesh NoC architectures. Lastly, Section VI concludes the paper and outlines some directions for future work.

II. Related Work

In [3] the impact of 2D and 3D NoC architectural topologies on performance measuring parameters such as network occupancy, latency and throughput evaluated. In [4] the authors have extended their formerly designed 2D OASIS-NoC to third dimension and named it 3D OASIS-NoC. The authors have concluded that 3D OASIS-NoC shows performance enhancement over 2D OASIS-NoC in terms of latency. In [17-18] it is demonstrated that in addition to the footprint reduction, 3D interconnection architectures lead to higher throughput, lesser energy dissipation and reduced network latency in comparison to conventional 2D NoC architectures. In [5] the authors stated that three-dimensional integrated circuits (3D ICs) present extensive improvements in comparison to two-dimensional circuits. 3D ICs have capability of reducing interconnect length as well as footprint without lessening the on chip transistors. In [6] benefits of 3D interconnects together with the fundamental design challenges in 3D ICs are discussed. In [19-23] the authors have illustrated that 3D stacked technologies offer a new prospect for network-on-chip design and since, 3D ICs have several layers of active devices assembled one over another and are capable of boosting up the system's performance characteristics. In [24-25] the authors have shown that even in the absence of scaling 3D ICs consent to performance enhancements.

In [26] the problem of IPs to tile mapping in regular 2D Mesh NoC topology is addressed. In [7] a mapping algorithm called GA-MMAS is proposed with the aim of optimizing NoC energy consumption. In [8] NoCOUT is presented which generates an energy efficient NoC topology customized to meet the requirements of a specific application. In [9] a contention and energy aware mapping algorithm is proposed which results in increased throughput and energy consumption minimization. In [10] multi-objective ant colony algorithm (MOACA) is proposed to map the IPs onto tiles in 2D mesh NoC design structures. This algorithm optimizes the total energy consumption of the NoC architecture as well as it provides a thermal stability in NoC. In [11] an energy-aware run-time mapping scheme is proposed for homogeneous NoC architectures considering the run time arrival and departure of applications into and from the system respectively. In [12] a novel source and distributed routing based methodology in order to optimize the application specific NoCs is proposed. In [13] the problem of application mapping onto NoC architectures is addressed. A communication dependence and computation model (CDCM) is presented that takes the timing as well as communication volume of applications into consideration. In [14] an ant colony optimization based mapping approach is proposed to schedule tasks of an application on NoC architecture such that minimization of total bandwidth requirement can be achieved. In [15] Dependencies-Neighborhood based Lower Energy Consumption approach (LEC-DN) is proposed that facilitates mapping of tasks onto NoC architecture at run time. In [16] the problem of mapping of real time applications onto tiles in mesh NoC architecture is addressed.

Numerous NoC designs have come up however, which one serves as best in which scenario must be first tested and experimented under different conditions. To make this possible numerous NoC simulators have been designed and various the research communities are currently utilizing them to analyze the effect of the designed NoCs. Darsim [32] supports 2D mesh interconnects only and offers various customizable parameters such as routing scheme, network size, VC sizes, packet sizes etc. Similarly, Noxim [33] is another such simulator that models 2D mesh NoC supporting different traffic pattern distribution and 2D routing algorithms. However, unlike DARSIM it does not supports the notion of virtual channels therefore not supporting the bandwidth sharing property of physical channels. One more JAVA based general-purpose purpose simulator known as gpNoCsim simulator [34] evaluates popularly known 2D on-chip architectures such as Butterfly, Mesh, Torus, FatTree etc. while the simulator in [35] only includes the evaluation of Mesh and Butterfly, Fat Tree architectures. The simulator developed by Stanford University, BookSim [36] has limited traffic injection models but embed broad range of two-dimensional topologies together with flattened butterfly and different 2D supportable routing algorithms.

In collaboration with the University of Southampton, UK and MNIT, Jaipur, India NIRGAM [37] is a NoC simulator that is used to analyze regular Mesh and torus topologies. The simulator provides modularity and flexibility feature by easily letting the extensions in terms of traffic, applications, architectural designs and

routings. At present form it only has source, XY and OE (Odd-Even) routing strategies for the 2D infrastructures. Recently, it has been extended to include 3D mesh that is capable of forming n interconnection network with quite less number of tiles [38]. SICOSYS [39] is developed in C++, is a general-purpose interconnection network simulator that forms its basis on the concept of versatility and modularity and that replicates crucial facts of low-level simulation and is integrated in RSIM [40]. RSIM is designed to achieve high instruction level parallelism targeted to simulate shared-memory multiprocessors and uniprocessors. It models contention at different resource levels and also has a multiprocessor coherence protocol included in the simulation environment. Simulators namely NoCsim in [41] and proposed in [43] and Nostrum [42] are some of those simulators that only support synthetic traffic generation and VC routers but do not report router power and energy.

GARNET [44], Ocino_tsim [45], is a cycle-accurate level simulator model that supports either a flexible egress-queued router or a standard virtual channel ingress-queued router with a five-stage pipeline and from the ORION model [47] computes power consumed by the router whereas the SystemC based Nostrum-NNSE [42] does not estimate power and communication energy requirement however, but has the ability to support the regular as well as irregular topologies but with limited number of traffic patterns. Graphite [48] is a particular PIN-based multicore simulator that has the ability to simulate thousands of cores by segmenting among multiple cores on the same die but also among the multi-networked computers. For computer architecture researchers SimpleScalar toolset [49] is utilized to model processor architectures and memory hierarchies. Therefore, it can be clearly noticed that most of the studied simulators are particular to 2D-NoC topology architecture and routings. Also, they do not cater the needs of the ever growing chip complexity and are unable to simulate the performances on the essential performance characteristics such as router energy consumption, total energy dissipated, latency, and throughput simultaneously.

III. Energy Efficient Mapping Heuristic

This section proposes the initial step that now-a-days a must in order to obtain an optimized NoC connection targeted to build application specific SoCs. The whole process is divided into a series of steps as explained below.

A. Platform Characterization

In order to better illustrate the proposed heuristic developed in this research work, a concrete NoC platform is needed. In this paper, the chip is comprised of PxQxR tiles which are arranged to form a 3D Mesh NoC architecture. The paper concentrates on the 3D regular NoC architecture. So, the static routing is preferred over adaptive routing as adaptive routing scheme is complex and requires more resources. Static XYZ routing is adopted for routing data packets among IP cores

B. Energy model for 3D Mesh NoC

The energy model used in this paper is based on the energy model presented in [27]. With XYZ routing, for the 3D Mesh NoC architecture, Eq.1 shows that the average dynamic energy used per bit while transferring it from core i to core j is calculated by the Manhattan distance between these two cores. Equation (1) is given as follows where $E_{bit}^{c_i, c_j}$ denotes the dynamic energy used per bit in transmitting it from core i to core j, E_{Sbit} denotes the energy consumed per bit when it is transported through a switch, E_{Lbit} the energy utilized

$$E_{bit}^{c_i, c_j} = n_{hops} \times E_{Sbit} + (n_{hops} - 1) \times E_{Lbit} \quad (1)$$

per bit on the link between two switches & n_{hops} the number of hops a data bit passes through until it reaches from source tile i to destination tile j (i.e. Core i to Core j). The same energy model is implemented in the proposed NC-G-SIM simulation framework for calculating the total energy consumption.

C. Formulation of energy aware application to core mapping for 3D Mesh NoC

The problem is to find optimal applications to cores mapping in 3D Mesh NoC architecture, so that the overall consumption of dynamic communication energy is minimized.

Communication Routing Graph, CRG, $G = G(C, P)$ also referred as Interconnection Network graph, is a directed graph, where each vertex c_i corresponds to one core in the architecture, and each directed edge $p_{i,j}$ corresponds to the routing path between c_i and c_j which is determined using XYZ routing algorithm. $e(p_{i,j})$ corresponds to the average energy consumed (joule) in transporting a single bit of information from core c_i to c_j , i.e., $E_{bit}^{c_i, c_j}$. $l(p_{i,j})$ corresponds to the group of links that forms the routing path $p_{i,j}$.

Application Communication Graph, ACG, $G = G(A, M)$ is a directed graph, where each vertex a_i corresponds to one application, and each directed edge $m_{i,j}$ corresponds to the communication from a_i to a_j . $V(m_{i,j})$ denotes the communication volume (bits) from a_i to a_j . The minimum bandwidth (bits/sec.) is given by $bw(m_{i,j})$ which the underlying communication architecture should provide.

The formulation of the problem of dynamic communication energy consumption minimization can be done as one to one mapping of applications to cores. The size of CRG must be greater than or equal to the size of ACG so that each application can be mapped onto one and only one core and all the applications in given ACG can be mapped onto CRG. We have derived a one to one mapping relation $bmap()$ [57] from ACG to CRG which reduces to

$$\text{minimize} \left\{ \text{Energy} = \sum_{m_{i,j}} V(m_{i,j}) \times e \left(p_{bmap(a_i), bmap(a_j)} \right) \right\} \quad (2)$$

While considering that traffic load on any link will not cross its allocated bandwidth limit.

D. Proposed Heuristic for 3D Mesh NoC

The proposed heuristic is based on the concept of branch-and bound and it moves through the search tree to derive an optimal mapping that leads to the least communication cost. Every node is assigned a label. For example, node 832xxxxx signifies an internal node where Application number A0, A1 and A2 of ACG are mapped on Core8, Core3 and Core2 of CRG respectively, but applications A3 to A7 of ACG are still unmapped. The communication requirements which involve incoming as well as outgoing data traffic from a specific application to every other application in given ACG are stored in communication Matrix. For a node in the search tree, the energy consumed by communication among mapped applications denotes the cost of that node. The cost of the child nodes is more than cost of their parent nodes and based on this property ineligible tree-branches are pruned later on in the heuristic. If a node meets the bandwidth requirement among the mapped applications, then it is referred as a legal node. This condition can be represented as shown in equation (3) [57].

$$Bw(l_k) \geq \sum_{v_{i,j}, a_i, a_j \in M} bw(m_{i,j}) \times lis_link(l_k, p_{bmap(a_i), bmap(a_j)}) \quad (3)$$

Here, M denotes the set of mapped applications. Illegal parent node yields illegal child nodes. In order to compute as lowest ULC as possible for a node, a greedy approach for mapping the applications onto cores is adopted. The Run time of proposed heuristic scales up with the system size. There exists a trade-off between the quality of solution and run time. The applications ranking and node preference queue are used in the proposed heuristic to reduce the search time by detecting as many unpromising nodes as possible at earlier stages during the search process and then trim away such nodes.

The heuristic iterates in-between the following two steps until it finds out an optimal solution. They are Branch: The next available unexpanded node from node preference queue is expanded to generate new child node and then the mapping of unmapped application with the uttermost communication volume onto to the set of unused cores is done.

Bound: The *upper limit cost (ULC)* of a node denotes a cost that is no less than the minimum communication cost of its legal, descendant child nodes where all the applications have been mapped (i.e. leaf nodes). The *lower limit cost (LLC)* of a node denotes the best possible communication cost that its legal descendant child nodes where all the applications have been mapped (i.e. leaf nodes) can probably attain. The ULC and LLC are taken as the bound parameters. These bound parameters are computed and checked on newly generated child node in the former step to see if it can yield the leaf nodes with the least communication cost on further expansion. This node is pruned if one of the following conditions holds: Mapped applications communication cost > the lowest ULC or LLC > the lowest ULC.

ULC Computation is done through a greedy mapping approach in which the next unmapped application a_k with the uttermost communication volume is chosen and is mapped on an unused core which is mapped with the application that communicates with a_k and is situated at the least Manhattan distance. This step goes on until a single descendent leaf node is recognized. If this leaf node fulfils the bandwidth requirements, then ULC of node under assessment is set equivalent to its cost or else ULC is set to infinity and the node is trimmed away.

To compute LLC, three components: C_{mm} , C_{mu} and C_{uu} are computed first and then are summed up together to give Lower Limit Cost. C_{mm} corresponds to communication cost among mapped applications and can be computed exactly as topological locations of the mapped applications are known. C_{mu} is the communication cost between mapped and unmapped applications. In evaluating C_{mu} all the unmapped applications are assumed to be mapped on an unused core at the nearest topological location of the used core. The last Component is C_{uu} which corresponds to the cost of communication among all unmapped applications. It is computed in two steps: In first step an unmapped application with highest communication volume is chosen and is mapped onto an unused core. Then all the remaining unmapped applications are assumed to be mapped on an unused core at the nearest topological location of the used core in the former step.

IV. NC-G-SIM Simulator

This section discusses the simulation framework of NC-G-SIM simulator. NC-G-SIM simulator is designed to accommodate all the existing requirements of NoCs. It is systemC based, open source, discrete cycle-accurate discrete event simulator that is highly modular and component-based specifically targeted at the

on-chip interconnection research domain. It is generic as it provides substantial support and extensibility primarily in applications attached to cores, topology design, routing algorithms.

IrNIRGAM [50] is extended to develop NC-G-SIM, due to its simplicity in designing the internal interconnection model. NC-G-SIM includes the simulation framework that supports two dimensional as well as three-dimensional mesh based topologies and also irregular NoC framework and their compatible routing schemes also. The simulator is targeted at the NoC research community to experiment with wide design choices with efficient and accurate mechanisms and full convenience. It gives the designer a very important plug-and-play feature through which the designers can attach any desired application library to any core or any router function as suitable to their needs. Therefore, in near future NC-G-SIM can emerge as a powerful tool to model and represent the interconnection on-chip networks for extremely intricate systems having large number of cores providing accurate performance estimation.

A. Traffic generation

NC-G-SIM simulator follows two modes random or fixed to select destination nodes for a source node. Traffic characteristics form an essential part while simulating and evaluating any NoC design. The simulator not only holds the use of CBR and Bursty traffic generator applications which are primitive and known traffic generators but also has incorporated additional traffic generators that are capable in generating concurrent traffics. At any node that has multiple destinations to which it communicates these traffic generators named as BWCBR can be attached that generates multiple traffic among these nodes during the simulation. In such a way simulations become more relative to the actual real-time scenarios where there exists heavy traffic exchange.

B. NoC Topology Framework

NC-G-SIM as said earlier that it is highly generic in nature, it is because that it can easily incorporate any style of topology. Here, the simulator takes in a topology configuration file that guides the simulator the network design that is to be formed. The format of the file is [tileID Neighbor_tileID₁ Neighbor_tileID₂..... Neighbor_tileID_n -1] always starting with the current tile identifier that must be begun from *tileID* zero and so on. Following the *tileID* identifier the *tileIDs* of its neighbouring nodes are specified with a delimiter “-1” marking the end of line and end of the neighbouring nodes for that tileID. Another file that gives the length of the links in the topology of the format [tileID next_tileID link_length_(in mm)] to estimate the energy consumption during simulation.

This feature facilitates the user to design one's own topology according to ones choice and experimentation scenario. The file can therefore, be used to generate the regular two-dimensional as well as three-dimensional and irregular topologies and as per the need of the situation it can be easily altered and modified for further analysis. NC-G-SIM simulator has been made to support maximum number of tiles in any topology type to be 500 with each tile having maximum number of neighbors to be 6. It can further be extended to support more cores per topology.

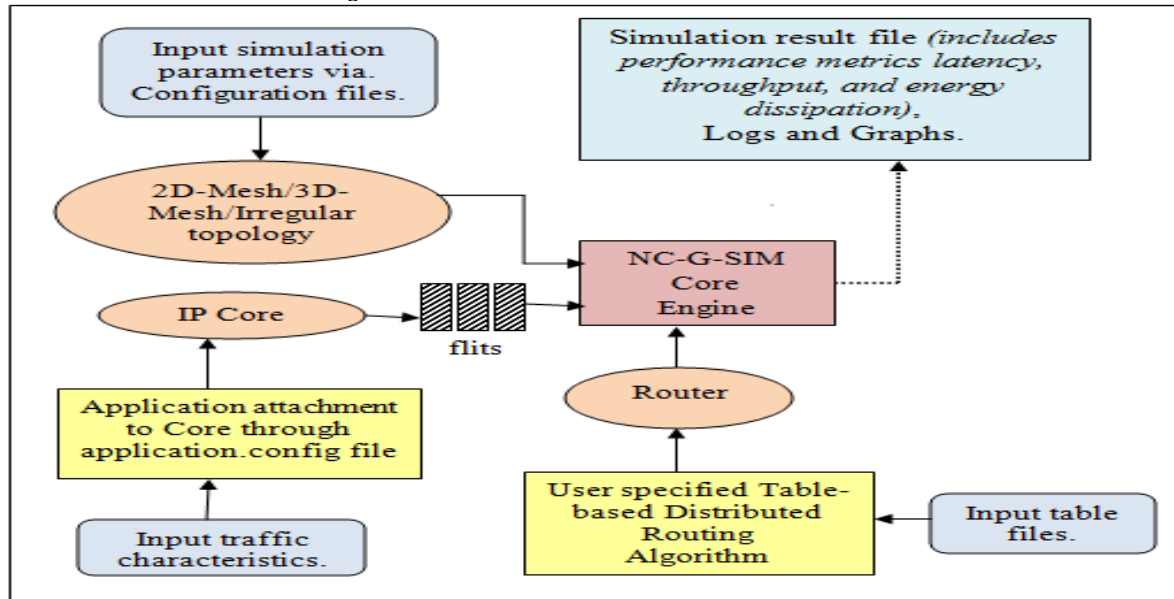
C. Routings Supported

NC-G-SIM simulator supports routings compatible to each topology kind. For two-dimensional Mesh based structure the most popular Dimension Order Routing (DOR) routing scheme, XY is used as well as Odd-Even turn based routing is used which guarantees the routing to be deadlock-free. Moreover, to make the routing more flexible Table-based distributed routing is also implemented to be used in all the three types of topology structures. In this table-based scheme each node in the network has a table associated with it that has the format [source_node, destination_node, next_node, path_id]. The packets are then transmitted on the basis of the decision taken using these table entries at each intermediate node during the transmission. The path_id field is used purposely so that in case of multiple path existence flits can be diverged through multiple routes and so the path_id field keeps a track of the path being followed by the flits. The tables for irregular topology can be filled offline using any deadlock-free irregular archetype supportable routing functions such as up*/down*[52], Left-Right [52], L-turn [53] and prefix-routing [54] etc on the other hand the tables for three-dimensional Mesh architecture can be generated using any of the minimal non-minimal, DOR or adaptive routing such as XYZ, LA-XYZ [55], BDOR[56] etc. The distributed table routing scheme is advantageous in uniform diverging of the traffic over different paths, also is deadlock and livelock free and also provides comparatively decent fault-tolerance without much overhead.

To ensure that no deadlock occurs in the 3D-Mesh and Irregular topologies another routing methodology is included in NC-G-SIM simulator that is based on the concept proposed by [51] in which the virtual channels are divided into two sets with different priorities. The virtual channel set with higher priority is allowed to transmit the newly injected messages. These new channels follow minimal path routing function that provides greater network adaptivity. The packets continue to flow in this set of virtual channels until a deadlock condition or congestion arises which makes them switch to the old lesser priority channels. These channels follow the original routing function, and once a packet acquires an original channel it is not allowed to do

transition to a new channel anymore to avoid deadlock situation. This scheme is said to be using escape channels to avoid congestion and deadlocking. The simulator supports this methodology in the table-based format where the format of the table is similar to that of the format discussed above except that the *path_id* field is divided to denote the two sets of virtual channel.

Figure 1 NC-G-SIM simulation framework overview [46].



NC-G-SIM makes use of flits as the unit of flow of information and implements wormhole switching scheme with multiple virtual channels to transmit the data between the nodes. It supports a variety of customizable features that can be easily modified and are given for carrying out the simulation. These customizable parameters are deciding topology type and routing are the size for the regular 2D-3D topologies, buffer size, packet as well as flit size, packet interval, simulation and traffic generation cycle and others. The simulator operates at the default clock frequency of 1GHz.

The NC-G-SIM control-flow architectural overview is shown in Fig.1. At the end of simulation a log report is generated as per the degree of log level selected by the user. In addition, to the log detail of the simulation another report is generated that contains the detailed output information that includes flit load, energy consumed, throughput, latency and standard deviation. GnuPlot graph plotter can also be used to get the graphical representation of the simulation results as the simulator produces the compatible codes.

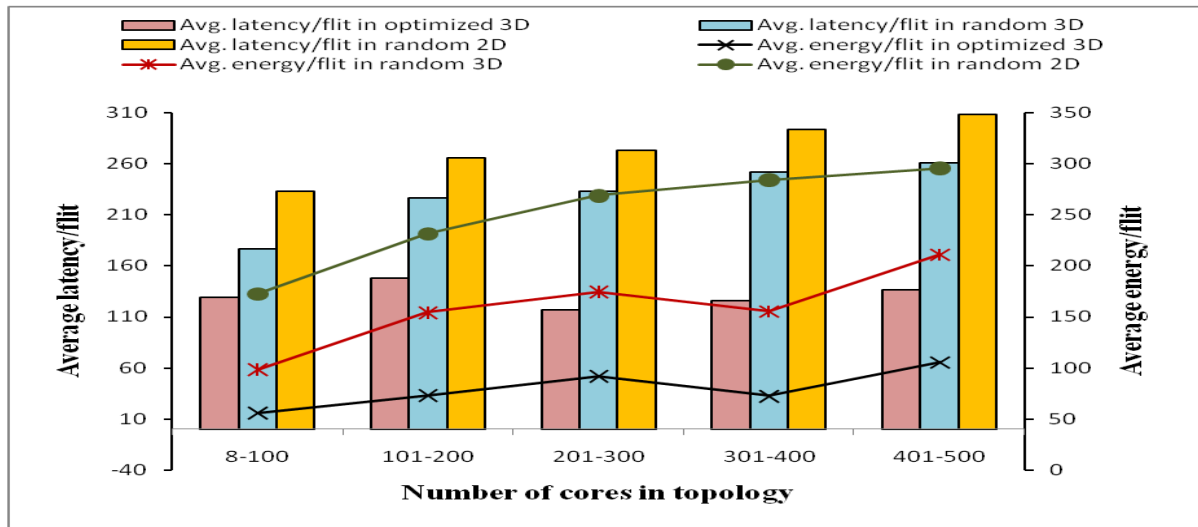
V. Experimental Results

The proposed NC-G-SIM simulator is used to perform the experiments and to generate the output results. The proposed branch and bound heuristic initially produced the efficient application to core mapping which forms the optimized topology sets and they were simulated using the proposed simulator to prove the benefit of applying the heuristic. The heuristic is applied on vast sets of test cases. For 100 different topology dimensions five different cases were generated and tested in order to get accurate information of the different performance parameters under consideration i.e. latency, throughput, and energy used to transmit the data during the simulation. For comparable reason and fair usage, same sized 2D and 3D Mesh topologies under random scenario, as well as under optimized 3D-Mesh based topology were taken in considerations. The traffic generator used is the BWCBR that transmitted concurrent traffic in the network for different source-destination pairs. Using TGFF, according to the specified application scenarios different bandwidth usage and communication volumes are generated for each case within in a particular set in order to check the output diversity.

The number of cores in simulation varies from 8 to 500 and the simulation was carried out for each case upto 5000 simulation cycles at constant throughput rate where the flit interval and packet size was set to two clock cycles and 8 flits/packet respectively. The above explained analytical model was used where the link energy between the routers (E_{Lbit}) and the router switch energy (E_{Sbit}) values, for per bit transmission, were calculated using the ORION tool for 0.18 μ m technology. E_{Lbit} is set to 0.0007 and E_{Sbit} is set to 0.52 and 0.54 for 4-port and 6-port routers respectively. The simulation used offline generated distributed table-based routing for 2D-Mesh and 3D-Mesh topologies filled using the compatible routing schemes such as XY and XYZ routings respectively.

From Fig. 2 it can be seen that the average time taken to reach the destinations has decreased tremendously in the optimized 3D-Mesh network by a range of 26.9% - 47.7% and 44.7% - 55.7% when compared against random 2D-Mesh and 3D-Mesh respectively.

Figure 2 Comparison in terms of latency and energy consumed between optimized 3D-Mesh and random 2D as well as 3D-Mesh topologies.



Here, we can observe that 2D-Mesh offers not much a degree of performances as has much greater node diameter and so takes much duration to travel. Optimally mapped 3D-Mesh structures are more efficient than randomly mapped 3D-Mesh due to the fact that in optimized mapping obtained from heuristic, applications are first ordered according to their communication requirements and all the applications communicating with each other are mapped within the minimum Manhattan distance limit. Optimally mapped 3D-Mesh structures over performs the optimally mapped 2D-Mesh since for the 2D and 3D Mesh structures of same size, the path lengths get reduced in 3D Mesh. The experimental results show that the beneficiary compact design of 3D makes it all the more efficient and also has reduced the total average energy consumed in the optimized design by 66% and from 45% opposite to optimally mapped 2D and randomly mapped 3D. The average total energy consumed is the sum of the average router and average edge energies and is lesser in optimized 3D-mesh as flits reach destinations within fewer hops due to packed in architecture.

We have also given for better evaluation and understanding purpose, the comparison between optimized 3D-Mesh v/s random 2D-Mesh and between optimized 3D-Mesh v/s random 3D-Mesh in terms of average latency/flit and the average total energy per received flits for 16, 32, 64, 100, 140, 196, 256, 300, and 400 number of cores in the topology. From Fig.3 and Fig.4 it can be clearly analyzed that efficiently mapped applications to the core offers better results in every sphere of performance that too with a drastic quality and performance gains.

Figure 3 Optimized 3D-Mesh compared to random 2D-Mesh shown for different selected topologies in terms of latency and total energy consumed.

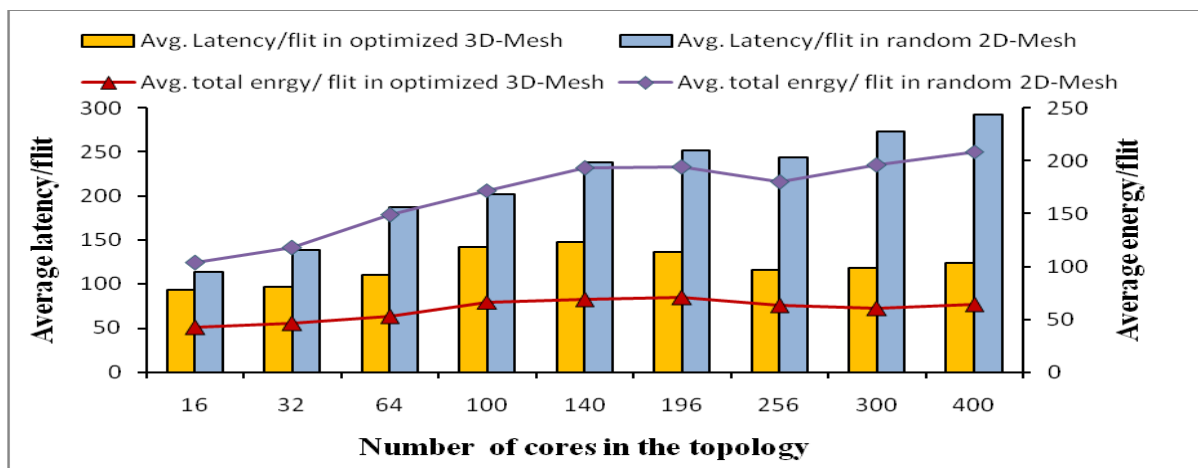
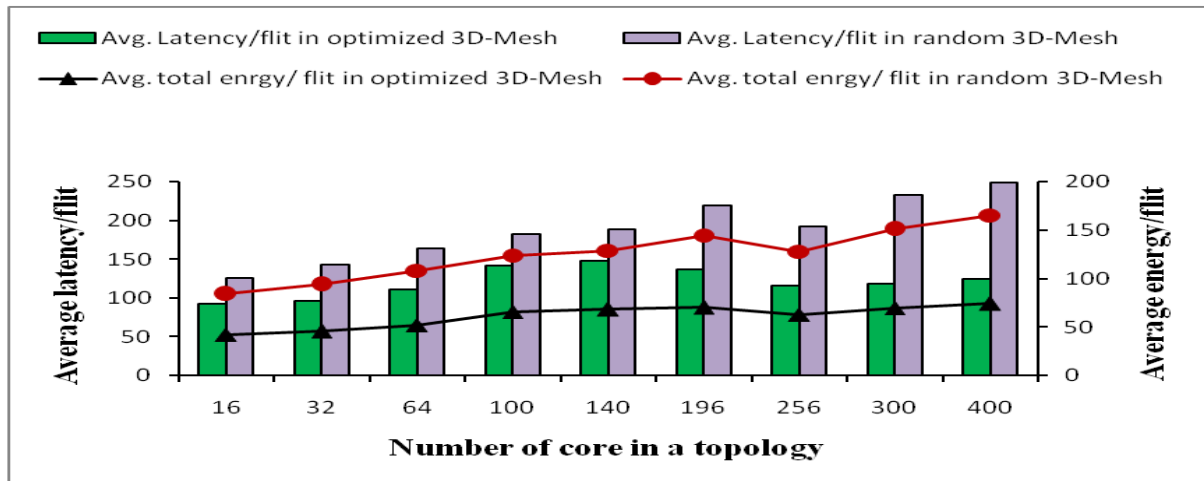


Figure 4 Optimized 3D-Mesh compared to random 3D-Mesh in terms of latency and total energy consumed for specific topologies.



VI. Conclusion

In this paper we have presented a generic SystemC based open-source simulator NC-G-SIM that has the capability to be used by the research community in order to analyze and simulate the various 3D Mesh, 2D Mesh and irregular topologies offering a wide variety of traffic generators and routing strategies particularly the distributed table-based that can be used in any structure. The simulator is flexible enough to support easy plug-and-play feature which helps in its further extension. We are planning to include other topologies, switching mechanisms and routings including fault-tolerant strategies. Also the paper introduces an energy-aware heuristic that is based on branch and bound technique for efficiently mapping the applications to the cores in the 3D-Mesh network that aims at minimizing the latency and the overall energy consumed by the system. It is also planned to be extended to be implemented on the irregular topologies.

References

- [1] M. Ali, M. Welzl, and M.A. Zwicknagl, "Networks on chips: scalable interconnects for future systems on chips," *Proc. 4th European Conf. on Circuits and Systems for Communications (ECCSC)*, 2008, pp. 240-245.
- [2] W. J. Dally and B Towles, "Route packets, not wires: On-chip interconnection networks," *Proc. Design Automation Conference*, 2001, pp. 684-689.
- [3] Y. Ghidini, T. Webber, E. Moreno, I. Quadros, R. Fagundes and C. Marcon, "Topological impact on latency and throughput: 2D versus 3D NoC comparison," *25th Symposium on Integrated Circuits and Systems Design (SBCCI)*, 2012, pp. 1-6.
- [4] A.B. Ahmed, A.B. Abdallah and K. Kuroda, "Architecture and design of efficient 3D network-on-chip (3D NoC) for custom multicore SoC," *Proc. International Conference on Broadband, Wireless Computing, Communication and Applications (BWCCA)*, 2010, pp. 67-73.
- [5] R.S. Patti, "Three-dimensional integrated circuits and the future of system-on-chip designs," *Proc. of the IEEE*, vol. 94, 2006, pp. 1214-1224.
- [6] K. Bernstein et al., "Interconnects in the third dimension: Design challenges for 3D ICs," *Proc. 44th annual Design Automation Conference ACM*, 2007, pp. 562-567.
- [7] N. Wu, Y. Mu, F. Zhou, and F. Ge, "GA-MMAS: an Energy-aware Mapping Algorithm for 2D Network-on-Chip," *Proc. World Congress on Engineering and Computer Science*, 2011, pp. 739-743.
- [8] J. Chan and S. Parameswaran, "NoCOUT: NoC topology generation with mixed packet-switched and point-to-point networks," *Proc. Asia and South Pacific Design Automation Conference*, 2008, pp. 265-270.
- [9] B. Ge, N. Jing, W. He, and Z. Mao, "Contention and energy aware mapping for real-time applications on Network-on-Chip," *Proc. SoC Design Conference (ISOC)*, 2012, pp. 72-76.
- [10] Y. Liu, Y. Ruan, Z. Lai, and W. Jing, "Energy and thermal aware mapping for mesh-based NoC architectures using multi-objective ant colony algorithm," *Proc. 3rd International Conference on Computer Research and Development (ICCRD)*, 2011, pp. 407-411.
- [11] G. Sun, Y. Li, Y. Zhang, L. Su, D. Jin and L. Zeng, "Energy-aware run-time mapping for homogeneous noc," *International Symposium on System on Chip (SoC)*, pp. 8-11, 2010.
- [12] R. Tornero and J. M. Orduna, "Improving topological mapping on NoCs," *International Symposium on Parallel & Distributed Processing, Workshops and PhD Forum (IPDPSW)*, pp. 1-4, 2010.
- [13] C. Marcon, N. Calazans, F. Moraes, A. Susin, I. Reis, and F. Hessel, "Exploring NoC mapping strategies: an energy and timing aware technique," *Proc. Conference on Design, Automation and Test in Europe*, 2005, pp. 502-507.
- [14] J. Wangl, Y. Li, S. Chai, Q. Peng, "Bandwidth-aware application mapping for NoC-based MPSoCs," *Journal of Computational Information Systems*, vol. 7, pp. 152-159, 2011.

- [15] M. Mandelli, L. O. E. Carara, G. Guindani, T. Gouvea, G. Medeiros, and F.G. Moraes, "Energy-aware dynamic task mapping for NoC-based MPSoCs," *IEEE International Symposium on Circuits and Systems (ISCAS)*, 1676-1679, pp. 2011,.
- [16] S. Wildermann and J. Teich, "Run time mapping of adaptive applications onto homogeneous NoC-based reconfigurable architectures," *International Conference on Field-Programmable Technology (FPT)*, pp. 514-517, 2009.
- [17] B. S. Feero, and P. P. Pande, "Networks-on-chip in a three-dimensional environment: A performance evaluation," *IEEE Transactions on Computers*, vol. 58, pp. 32-45, 2009.
- [18] Y. Xu, Y. Du, B. Zhao, X. Zhou, Y. Zhang, and J. Yang, "A low-radix and low-diameter 3D interconnection network design," *15th International Symposium on High Performance Computer Architecture*, pp. 30-42, 2009.
- [19] N. Hassanpour, S. Hessabi, and P.K. Hamedani, "A task migration technique for temperature control in 3D NoCs," AINA, unpublished.
- [20] A.B. Ahmed, "The design of a 3D Network-on-Chip for many-core SoC," University of Aizu, October 2012.
- [21] S. Tyagi, "Extended balanced dimension ordered routing algorithm for 3D-networks," *International Conference on Parallel Processing Workshops*, pp. 499-506, 2009.
- [22] T. Schonwald, J. Zimmermann, O. Bringmann, and W. Rosenstiel, "Fully adaptive fault-tolerant routing algorithm for network-on-chip architectures," *10th Euromicro Conference on Digital System Design Architectures, Methods and Tools*, pp. 527-534, 2007.
- [23] A.M. Rahmani, K. Latif, K.R. Vaddina, P. Liljeberg, J. Plosila, and H. Tenhunen, "Congestion aware, fault tolerant, and thermally efficient inter-layer communication scheme for hybrid NoC-bus 3D architectures," *Fifth IEEE/ACM International Symposium on Networks on Chip (NoCs)*, pp. 65-72, 2011.
- [24] K. Banerjee, S.J. Souri, P. Kapur, and K.C. Saraswat, "3-d ICs: A novel chip design for improving deep-submicrometer interconnect performance and systems-on-chip integration," *Proc. IEEE*, 2001, pp. 602-633.
- [25] M. Ebrahimi, M. Daneshmand, P. Liljeberg, J. Plosila, and H. Tenhunen, "Exploring partitioning methods for 3D Networks-on-Chip utilizing adaptive routing model," *Fifth IEEE/ACM International Symposium on Networks on Chip (NoCs)*, Pittsburgh, pp. 65-72, 2011.
- [26] J. Hu, and R. Marculescu, "Energy-aware mapping for tile-based NoC architectures under performance constraints," *Proc. Design Automation Conference, Asia and South Pacific*, 2003, pp. 233-239.
- [27] J. Hu, and R. Marculescu, "Design methodologies for application specific networks-on-chip," Carnegie Mellon University, 2005.
- [28] W. J. Dally, and B. Towles, "*Principles and practices of interconnection networks*," Amsterdam, London: Elsevier/Morgan Kaufmann, 2004.
- [29] P. J. Fortier, and H. E. Michel, "*Computer systems performance evaluation and prediction*," London: Digital Press, 2003.
- [30] H. Kobayashi, "Modeling and analysis: an introduction to system performance evaluation methodology," London: Addison-Wesley, 1978.
- [31] D. Lilja, "*Measuring computer performance: a practitioner's guide*," Cambridge, New York: Cambridge University Press, 2000.
- [32] L. Mieszko, S.K. Sup, C.M. Hyon, R. Pengju, O. Khan, and S. Devadas, "Darsim: A parallel cycle-level NoC simulator," *MoBS 2010-Sixth Annual Workshop on Modeling, Benchmarking and Simulation*, Online: <http://dspace.mit.edu/handle/1721.1/59832>, 2010.
- [33] M. Palesi, D. Patti, and F. Fazzino, Noxim: The NoC Simulator, Online, <http://noxim.sourceforge.net>, 2010.
- [34] H. Hossain, M. Ahmed, A. Al-Nayeem, T.Z. Islam, and M.M. Akbar, "Gpncsim – A General Purpose Simulator for Network-On-Chip," *International Conference on Information and Communication Technology*, pp. 254-257, 2007.
- [35] G. N. Khan, and V. Dumitriu, "A modeling tool for simulating and design of on chip network systems," *Microprocessors and Microsystems*, vol. 34, pp. 84-95, 2010.
- [36] N. Jiang, G. Micheli, D. Becker, B. Towles, and W. Dally, "Booksim interconnection network simulator," Online: <https://nocs.stanford.edu/cgi-bin/trac.cgi/wiki/Resources/BookSim>.
- [37] L. Jain, B.M. Al-Hashimi, M.S. Gaur, V. Laxmi, and A. Narayanan, "NIRGAM: a simulator for noc interconnects routing and application modeling," *Design, Automation and Test in Europe Conference*, 2007.
- [38] M.K. Puthal, M.S. Gaur, and V. Laxmi, "Performance Comparison of 2D and 3D Mesh NoC," 2011.
- [39] V. Puente, J.A. Gregorio, and R. Bevide, "SICOSYS: an integrated framework for studying interconnection network performance in multiprocessor systems," *Proc. of 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing*, 2002, pp. 15-22.
- [40] V. S. Pai, P. Ranganathan, and S. V. Adve, "RSIM: An execution-driven simulator for ILP-based shared-memory multiprocessors and uniprocessor," *Proc. of Third Workshop on Computer Architecture Education*, 178, 1997.
- [41] M. Jones, "NoCsim: a versatile network on chip simulator, Online: <https://circle.ubc.ca/handle/2429/16550>.
- [42] Z. Lu, R. Thid, M. Millberg, E. Nilsson, and A. Jantsch, "NNSE: Nostrum Network-on-Chip Simulation Environment," *Proc. of Swedish System-on-Chip Conference*, Stockholm, Sweden, 2005.
- [43] C. Grecu, A. Ivanov, R. Saleh, C. Rusu, L. Anghel, P.P. Pande, and V. Nuca, "A flexible network-on-chip simulator for early design space exploration," *Proc. of Conf. on Microsystems and Nanoelectronics Research*, 2008, 33-36.
- [44] N. Agarwal, T. Krishna, L.S. Peh, and N.K. Jha, "GARNET: A detailed on-chip network model inside a full-system simulator," *Proc. of IEEE International Symposium on Performance Analysis of Systems and Software*, 2009, pp. 33-42.
- [45] S. Prabhu, B. Grot, P.V. Gratz, J. Hu, "Ocin tsim- DVFS aware simulator for NoCs," *Proc. of SAW*, 1, 2009.
- [46] K. Vyas, N. Choudhary, and D. Singh, "NC-G-SIM: A Parameterized Generic Simulator for 2D-Mesh, 3D-Mesh & Irregular On-chip Networks with Table-based Routing," *International journal of Global Journal of Computer Science and Technology on Network, Web & Security*, vol. 13, pp. 7-12, 2013.

- [47] H.S. Wang, X. Zhu, L.S. Peh, and S. Malik, "Orion: a power-performance simulator for interconnection networks." *Proc. 35th Annual IEEE/ACM International Symposium on Microarchitecture*, 2002, pp. 294-305.
- [48] J. E. Miller, H. Kasture, G. Kurian, C. Gruenwald, N. Beckmann, C. Celio, and A. Agarwal, "Graphite: A distributed parallel simulator for multicores," *16th IEEE International Symposium on High Performance Computer Architecture*, pp. 1-12, 2010.
- [49] T. Austin, E. Larson, and D. Ernst, "SimpleScalar: An infrastructure for computer system modeling." *IEEE Journal Computer*, vol. 35, pp. 59-67, 2002.
- [50] N. Choudhary, M.S. Gaur, and V. Laxmi, "Irregular NoC simulation framework: IrNIRGAM," *International Conference on Emerging Trends in Networks and Computer Communications*, pp. 1-5, 2011.
- [51] F. Silla, J. Duato, "High-performance routing in networks of workstations with irregular topology," *Journal of IEEE Transactions on Parallel and Distributed Systems*, vol. 11, pp. 699-719, 2000.
- [52] M. D. Schroeder, A. D. Birrell, M. Burrows, H. Murray, R. M. Needham, T. L. Rodeheffer, and C. P. Thacker, "Autonet: A high-speed, self-configuring local area network using point-to-point links," *IEEE Journal on Selected Areas in Communications*, vol. 9, pp. 1318-1335, 1991.
- [53] M. Koibuchi, A. Funahashi, A. Jouraku, and H. Amano, "L-turn routing: An adaptive routing in irregular networks," *Proc. of International Conference on Parallel Processing*, 2001, pp. 383-392.
- [54] J. Wu, and L. Sheng, "Deadlock-free routing in irregular networks using prefix routing," *Proc. of ISCA 12th International Conference on Parallel and Distributed Computing Systems*, 1999, pp. 424-430.
- [55] A. B. Ahmed, "*The design of a 3D Network-on-Chip for many-core SoC*," doctoral diss., University of Aizu, 2012.
- [56] S. Tyagi, and M.A. Khan, "Topologies and routing strategies in MPSoC," *International Journal of Embedded Systems*, vol. 5, pp. 27-35, 2013.
- [57] P. Wadhvani, N. Choudhary, and D. Singh, "Energy Efficient Mapping in 3D Mesh Communication Architecture for NoC," *International journal of Global Journal of Computer Science and Technology on Network, Web & Security*, vol. 13, pp. 1-6, 2013.

Acknowledgement

This work is supported by Department of Science and Technology, Jaipur, Rajasthan, India under the research project "Network-on Chip simulation framework for regular, irregular and 3D-Mesh Interconnection Architecture".