DOI: http://dx.doi.org/10.26483/ijarcs.v8i7.4359

ISSN No. 0976-5697

Volume 8, No. 7, July – August 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

LATENCY EFFICIENT MAPPING IN 3D MESH NOC INTERCONNECTION NETWORK ARCHITECTURE

Neha Jain
Department of CSE
Geetanjali Institute of Technical Studies
Udaipur, India

Mayank Patel
Department of CSE
Geetanjali Institute of Technical Studies
Udaipur, India

Abstract: The progresses and developments in the domain of semiconductor technology is about to take us into the era where there will be thousands of cores available on a System on Chip. The onset of 3D integration technologies has unlocked the doors of novel prospects for design of on-chip networks in SoCs attaining the higher efficiency in contrast to 2D integration by aptly adjusting the increased path lengths of 2D on chip archetypes. The intelligent mapping of the applications to core on a given NoC architecture can result into the improved system's dynamic latency. The paper presents a heuristic based on branch and bound methodology for intelligent mapping of applications to cores in 3D Mesh NoC architecture. The experimental outcomes illustrate that average latency has been immensely decreased in the optimized 3D-Mesh on chip networks when compared against optimized 2D-Mesh network of the same size.

Keywords: NOC, 3D IC, Mesh topology, Latency

I. INTRODUCTION

The network on chip interconnect has resulted into an archetype shift from computation centric design architecture to communication centric design architecture. The use of on-chip network leads to several gains in terms of performance, modularity, and structure. The upcoming generation SoCs will comprise of a huge number of cores and the key challenge to work out will be the NoC bottleneck of these systems which confines scalability. The onset of 3D integration technologies has unlocked the doors of novel prospects for design of on-chip networks in SoCs. The union of two evolving standards, NoC & 3D IC, enables the design of novel design structures with significant performance enhancements in quality metrics upon conventional solutions [8]. One of the anaphases in design flow of on-chip networks is mapping of applications onto cores. This phase has an considerable impact on the quality metrics of the system such as dynamic communication latency, energy consumption etc. To attain this, in this paper we have formulated mapping problem followed by demonstration of the result of several applications to cores mappings on the dynamic communication latency of a given system. In this paper a Branch-and-Bound heuristic to intelligently map the given set of application onto cores in 3D NoC architecture to reduce the average dynamic communication latency of the system is presented. In order to validate efficiency of the proposed approach several experiments have been carried out on several arbitrary scales. In [4][10] it is illustrated that in addition to the footprint reduction in a fabricated design, 3D network structures are more inclined towards leading enhanced performance in terms of smaller latency, lower dissipation of energy and higher throughput in comparison to conventional 2D NoC archetypes. In [6] the mapping problem for 2D regular Tile - based structural designs is addressed.

II. LATENCY EFFICIENT MAPPING HEURISTIC

The proposed approach makes use of the model presented in [6]. The design of chip is comprised of PxQxR tiles which are linked as per the fundamental 3D Mesh structure. Each tile in 3D NoC has IP Core, Virtual Channels (VCs) & seven communications links (East, West, North, South, Front, Rear and Core). The model proposed in [6] takes the Manhattan distance between the cores into consideration while mapping the applications. The basic idea is to map the cores that communicate with each other at the smallest Manhattan distance as possible. Moreover as the NoC architecture is a communication centric design, both these factors ultimately lead to the reduced average latency of the system and energy as well in comparison to a randomly mapped 3D Mesh NoC. Deploying XYZ routing to route packets in a 3D Mesh NoC architecture, Eq.1 illustrates that the cost of transporting a single bit of data from core i to core j is computed by the Manhattan distance between these two cores.

$$C_{bit}^{c_i,c_j} = n_{hops} \times C_{Sb} + (n_{hops} - 1) \times C_{Lb} \qquad (1)$$

Where $C_{bit}^{c_i,c_j}$ corresponds to the cost of communicating a bit of information from core i to core j, C_{Sb} the switching costconsumed on switch when a bit of information while getting transported through a switch, C_{Lb} the link cost consumedby a bit of information on the link between two switches &$n_{hops}$is the number of hops a bit of information goes through on its way from source tile i to destination tile j (i.e. Core i to Core j).

Formulation of latency aware mapping problem:

In order to minimize the overall latency of system, we need to obtain a one to one mapping of applications onto the cores in 3D Mesh NoC.

Core Graph, CG, G = G(C, Rp) is a directed graph consisting of set of vertices C, where c_i represents a core in the architecture, and $Rp_{i,j}$ represents a directed edge that is the routing path computed using XYZ routing algorithm between c_i and c_j. e($Rp_{i,j}$) refers to the consumption of average energy (joule) in carrying a bit of data from core c_i toc_j. Set of links that constitute the $Rp_{i,j}$ is represented by L ($Rp_{i,j}$).

Application Graph, AG, G = G(A, T) is a directed graph consisting of set of vertices A where ap_i refers to an application and each directed edge $t_{i,j}$ representsthat a_i communicates with a_j . Communication volume (bits) between any two applications is represented by V($t_{i,j}$). The least possible bandwidth (bits/sec.) which the underlying communication structural design should offer is denoted by bw ($t_{i,j}$).

The number of cores in the CG must be ample enough to accommodate the applications in the AG following one to one mapping, i.e. one core can accommodate one application only onto it and one application can be mapped onto one core only. This condition can be represented as:

$$size\ (AG) \leq size(CG) \qquad (2)$$

The heuristic we have proposed in this paper is built on the concept of branch and bound approach. The heuristic travels through a search tree that represents the solution space with the aim of obtaining an optimal mapping with the least communication cost. This could be only possible if the two applications communicating with each other can be placed as near as possible. If two cores are placed nearly then the time taken by the data to reach from source to destination will also reduce. A label is assigned to every node in the tree. For example, node 356xxxxx implies an internal node where Core3, Core5 and Core6 of CG are mapped with application number A0, A1 and A2 of AG in that order; and unmapped applications are A3 to A7. A data-traffic matrix is maintained that stores the communication requirements which comprise incoming plus outgoing data traffic from a particular application to every other application in given AG. For a node in the search tree, the cost of communication is defined by the sum of cost consumed by switches (routers)in order to transmit the bits from source to destination following the path that has been computed using XYZ routing algorithm and the cost of transporting the data over links in between source and destination as shown in equation (1).The mapping cost of the child nodes are always greater than that of parent nodes and based on this, unqualified tree-branches are clipped later on. A node is legal if it encounters the requirement in terms of bandwidth amongst the mapped applications [6][9]. Illegal parent node produces illegal child nodes.

The upper end cost (UEC) of a node stands for a cost that is not lower than the minimum communication cost of its legitimate successor child nodes comprising of all the applications that have to be mapped (i.e. leaf nodes). So as to calculate as lowest UEC as achievable for a node, a greedy methodology for the applications to cores mapping is implemented.

The lower end cost (LEC) of a node stands for the best achievable communication cost that its legal descendant child nodes comprising of all the applications that have to be mapped (i.e. leaf nodes) can probably reach.

The execution time of the heuristic proposed in this paper escalates with the increase in the number of on-chip cores in system. There exists a trade-off between the execution time and solution's quality.

The proposed heuristic searches as many unproductive nodes as possible at the earlier stages in the course of the search process and then crops such nodes.

To this end, the total communication requirements of every application is computed and then the applications are assigned a number that is nothing but a rank for the reason that mapping of applications with higher rank can be done at the earlier stages of mapping. The legal nodes which qualify for the further extension are stored in a node preference queue. The node with the lowermost cost is given the highest priority for branching. The queue length significantly affects the execution time of proposed heuristic. When the queue size reaches a threshold value, the child nodes are then inserted into queue based on some firm measures.

The heuristic goes over the following two stages until it attains an optimal solution.

Branch: Under this step, the next unexpanded node from the front of node preference queue is removed and explored further to generate new child nodes by mapping the next application that has not been mapped yet and has the extreme communication requirement; onto a core from the set of vacant cores.

Bound: Under this step, each child node that has been produced in the aforementioned step is examined to comprehend if it leans towards yielding the best leaf nodes later. For this node the UEC and LEC costs are evaluated and this node is clipped if either the communication cost among applications that have already been mapped on cores or results in higher LEC than the lowest UEC that has been estimated all through the exploration.

To reduce the computational time of the bound parameters without affecting the quality of UEC and LEC, the following approach is adopted:

UEC Computation: To discover a legal descending leaf node with the smallest cost, greedy mapping process is employed. The greedy mapping maps the next application a_{um} with the highest communication requirement and its best location (x, y, z) on CG is estimated as follows:

$$x = \frac{\sum_{\forall a_{i\in M}} \left(\mathrm{V}(t_{um,i}) + \mathrm{V}(t_{i,um})\right) \times a_i^x}{\sum_{\forall a_{i\in M}} \left(\mathrm{V}(t_{um,i}) + \mathrm{V}(t_{i,um})\right)}$$

$$y = \frac{\sum_{\forall a_{i\in M}} \left(\mathrm{V}(t_{um,i}) + \mathrm{V}(t_{i,um})\right) \times a_i^y}{\sum_{\forall a_{i\in M}} \left(\mathrm{V}(t_{um,i}) + \mathrm{V}(t_{i,um})\right)} \qquad (3)$$

$$z = \frac{\sum_{\forall a_{i\in M}} \left(\mathrm{V}(t_{um,i}) + \mathrm{V}(t_{i,um})\right) \times a_i^z}{\sum_{\forall a_{i\in M}} \left(\mathrm{V}(t_{um,i}) + \mathrm{V}(t_{i,um})\right)}$$

The coordinates of the core onto which a_i is mapped are denoted by a_i^x, a_i^y *and* a_i^z(row, column and slice number); the set of applications whose mapping has been done is denoted by M. With the mapping of every next unmapped application this set is revised. Application a_{um} is placedonto a core which is vacant and lies at the least Manhattan distance to (xyz). This stage drives into loop till a single descending leaf node is acknowledged. If the leaf node obtained is legal, then UEC of node under exploration is make equivalent to its cost or else UEC is set to infinitely large and the node is clipped.

As every next unmapped application is placed onto such a core that is located at the as closest distance as possible to all the other cores with whom this application is communicating. This leads to reduced delay in transferring packets between two communicating applications.

LEC Computation:

The LEC has three parts: L_{mm} is the cost of communication between applications whose mapping has been done, L_{mu} is the communication cost between mapped and unmapped applications and $L_{uu}$is the communication cost between all applications which are yet to be mapped on CG.

$$LEC = L_{mm} + L_{mu} + L_{uu} \qquad (4)$$

III. Experimental Results and Analysis

NC-G-SIM simulator is used in order to test the performance of the heuristic presented in this paper. NC-G-SIM is a discrete event, cycle accurate simulator which supports Regular 3D, 2D and irregular topology framework with XYZ and distributed table based routing. During experimenting the performance on the 3D Mesh NoC, the number of maximum slices kept is 4 for the reason that in 3D ICs as the number of vertically stacked dies grows, power density/area and the length of heat conduction path also rises [1][3][5].Estimating using Orion [7] for 0.18µm technology C_{LB} is set to 0.0007 andC_{SB} is assumed to be 0.54 and 0.52 for 6 Ports and 4Ports router respectively. The packet size is 8 bytes and flit-interval is set to 2 clock cycles. The heuristic is tested on different topology sizes where number of cores used ranges from 8 to 512. The heuristic is implemented on 5 sets of 100 varying topologies each to get the intelligent along with random application to cores mapping in both 3D and 2D Mesh NoC structural designs of similar sizes with similar traffic settings. With the help of TGFF [2] thousand sets of benchmarks were arbitrarily generated with varied requirement in terms of bandwidth and communication volume of the IP cores in line with the specified distribution. The routing schemes used are XY and XYZ to route the traffic in 2D and 3D Mesh NoC respectively. The average latency/flit is taken as performance metric. The simulation time is taken as 5000 clock cycles with applied packet injection interval.

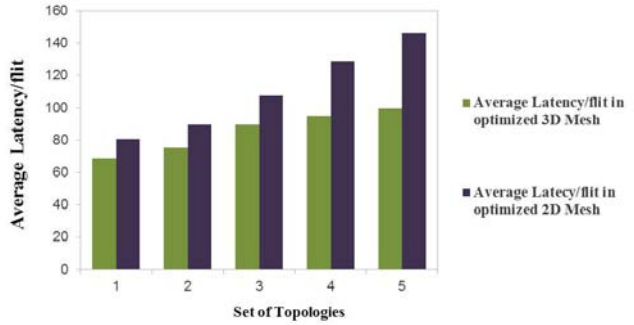

Figure1. Comparison of Latency between optimized mapping same sized 3D and 2D Mesh topologies with similar traffic conditions

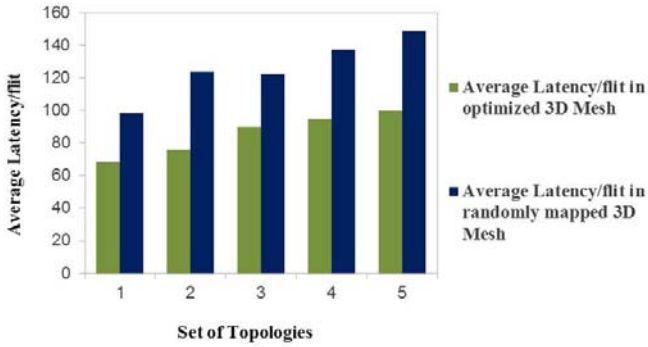

Figure2. Comparison of Latency between optimized mapping and random mapping in 3D Mesh topologies with similar traffic conditions

The graph plotted in fig.1 and fig.2 clearly shows that average latency per flit has reduced by a reasonable amount in the intelligently mapped 3D Mesh NoC in comparison to randomly mapped 3D Mesh and intelligently mapped 2D Mesh NoC as well.

In comparison to randomly mapped the intelligently mapped 3D Mesh NoC shows the significant drop in the average time taken by the packets to reach their destinations by 29.6 to 48.9%. When compared against intelligently mapped 2D Mesh NoC, the proposed heuristic results in 14.9-33.9% savings in average delay in the reaching the packets at their destination(i.e. average latency/flit).

Now, we can perceive that 2D-Mesh presents not much a degree of performances as has comparatively greater node diameter than 3D Mesh of the similar size that leads to take more length of time to complete the journey to the destination node. The optimized 3D-Mesh has employed the intelligent latency efficient mapping heuristic and as all the applications that communicate with each other are placed on the CG within the minimum Manhattan distance perimeter and thus smaller routes is to be covered here.

IV. CONCLUSION

This paper presents an intelligent latency aware mapping heuristic to map the applications in the given AG onto the cores in the CG in 3D Mesh NoC archetype. The results of various experiments being conducted evidently illustrate that the intelligent mapping results into a considerable amount of drop in the average latency of the system. The heuristic runs fast as well. This Branch and Bound based heuristic can be employed to map applications onto the cores in irregular NoC design structures where the application units differ in shape and dimensions and it is challenging to attain a deadlock free routing.

V. REFERENCES

[1] Bernstein, K., Andry, P., Cann, J., Emma, P., Greenberg, D., Haensch, W., & Young, A. (2007, June). Interconnects in the third dimension: Design challenges for 3D ICs.In Proceedings of the 44th annual Design Automation Conference (pp. 562567).ACM.Choudhary, N., Gaur, M. S., &Laxmi, V. (2011).Energy Efficient Network Generation for Application Specific NoC. Global Journal of Computer Science and Technology, 11 (16).

[2] Dick, R. P., Rhodes, D. L., & Wolf, W. (1998, March). TGFF: task graphs for free. In Proceedings of the 6th international workshop on Hardware/software codesign (pp. 97-101). IEEE Computer Society.

[3] Ebrahimi, M., Daneshtalab, M., Liljeberg, P., Plosila, J., &Tenhunen, H. (2011, May). Exploring partitioning methods for 3D Networks-on-Chip utilizing adaptive routing model.In Networks on Chip (NoCS), 2011 Fifth IEEE/ACM International Symposium on (pp. 7380).IEEE.

[4] Feero, B. S., &Pande, P. P. (2009). Networks-onchip in a three-dimensional environment: A performance evaluation. Computers, IEEE Transactions on, 58 (1), 32-45.

[5] Hassanpour, N., Khadem, P., Hessabi,S. (2013). A Task Migration Technique for Temperature Control in 3D NoCs. In 27th IEEE International Conference on Advanced Information Networking and Applications (AINA). Manuscript submitted for publication.

[6] Hu, J., Marculescu, R. (2005). Design methodologies for application specific networks-on-chip. Ph.D. dissertation.Carnegie Mellon University.

[7] Kahng, A. B., Li, B., Peh, L. S., &Samadi, K. (2009, April). Orion 2.0: A fast and accurate noc power andarea model for early-stage design space exploration. In Proceedings of the conference on Design, Automation and Test in Europe (pp. 423-428). European Design and Automation Association.

[8] Rahmani, A. M., Latif, K., Liljeberg, P., Plosila, J., &Tenhunen, H. (2010, November). Research and practices on 3D networks-on-chip architectures. In NORCHIP, 2010 (pp. 1-6). IEEE.

[9] Wadhwani, P., Choudhary, N. and Singh D. (2013). Energy Efficient Mapping in 3D Mesh Communication Architecture for NoC.International journal of Global Journal of Computer Science and Technology on Network, Web & Security (pp. 1-6).

[10] Xu, Y., Du, Y., Zhao, B., Zhou, X., Zhang, Y., & Yang, J. (2009, February). A low-radix and low-diameter 3D interconnection network design.In High Performance Computer Architecture, 2009.HPCA 2009. IEEE 15th International Symposium on (pp. 30-42). IEEE.