Computer Science • 20(3) 2019 https://doi.org/10.7494/csci.2019.20.3.3303

Somasundaram Kanagasabapathi
Chythanya Calicut

A ROUTING ALGORITHM AND A ROUTER ARCHITECTURE FOR 3D NoC

Abstract *In recent years, the enhancement of microchip technologies has enabled large scale Systems-on-Chip (SoC). Due to sharp increase in number of processing elements, SoC faces various challenges in design and testing. Network on Chip (NoC) is an alternative technology to overcome the challenges in SoC design and testing. NoC emerged as a key architecture that allows one to optimize the parameters like power and area. In spite of its applications, NoC faces some real time challenges like designing an optimum topology, routing scheme and application mappings. In this paper, we address the main three issues on NoC, namely, designing of an optimal topology, routing algorithm and a router design for the topology. First, we propose a topology and a routing algorithm. We prove that our recursive network topology is Hamiltonian connected and we propose an algorithm for data packet transmissions, which is free from cyclic deadlocks and the algorithm maximizes the congestion factor. Our experimental results show that the proposed topology gives better performance in terms of average latency and power than the other topologies. Finally, we propose a router architecture for our 3D-NoC. The router architecture is based on shared buffers. Also, our experimental results indicate that the proposed router architecture consumes less area and power than the Virtual Channel architecture.*

Keywords network on chip, 3D topology, routing algorithm, router

Citation Computer Science 20(3) 2019: 369–383

1. Introduction

Traditional bus based System on Chip (SoC) interconnection technologies are unable to meet the requirements of recent multicore processors. Industries are looking various alternative solutions to have better interconnections between cores. Network on Chip (NoC) is an alternative solution to SoC, which will promise the efficient network connections and routing. On-chip system designs have moved towards a more communication based methodology. For many applications, NoCs give optimum structure and connectivity [6]. In recent years, academia and industry have been looking at possible three-dimensional NoC (3D-NoC) technology with high density, high bandwidth, and low power consumption. Even though the 3D-NoC architectures give better performance and energy efficiency against 2D-NoC systems, the issues on chip area, power and reliability are the still open [23]. In a 3D-NoC, the communications between different layers (dies) will be realised by means of Through-Silicon-Via (TSV) technology. Finding an optimal number of TSVs and their placements are vital problems. Wang *et al.* [27] proposed a TSV sharing (TS) scheme to save TSVs in 3-D NoC by enabling neighboring routers to share the vertical channels in a time division multiplexing way. We have to optimize the number of TSVs during the assembly process since thermal issues are directly related to number of TSVs [5]. Eghbal *et al.* [8] addressed various classifications of the potential physical faults of a baseline TSV-based 3D NoC architecture by targeting two-dimensional (2D) NoC components and their inter-die connections. The pros and cons of 3D integrated chips are discussed in [5].

A NoC architecture consists of four basic components, namely, network interfaces, cores, routers and links. The nodes in the NoC network will be connected with links and the links have either static or dynamic bandwidth. Similarly, the routing schemes are also either static or dynamic. Developing an efficient NoC means, the NoC should have an optimum network topology, deadlock free routing scheme, it should consume low power and better application mappings. Various survey reports on recent developments in NoC can be found in [4, 10, 18].

A reconfigurable and adaptive routing method for fault-tolerant mesh-based networks-on-chip is given in [25]. The authors (Valinataj *et al.*) shown that their algorithm can be dynamically reconfigured to support irregular topologies caused by faulty components in a mesh network. An adaptive inter-layer message routing algorithm with partial vertical links is given in [17]. Feero *et al.* [9] showed that the 3D-NoCs based on mesh and tree give better performance than the two dimensional architectures. In [26], the authors proposed a Recursive Network Topology (RNT) and a Modified Mesh Topology (MMT) and they calculated the buffer size, average flits delay and dissipated energy for these two topologies and concluded RNT gives better performance than MMT in terms of the above parameters. Recently, Mamaghani and Jamali [14] showed an adaptive congestion-aware routing algorithm consistent with traffic load for solving the congestion problem of wireless routers.

In [7], Dubois *et al.* proposed a deadlock free routing algorithm for a 3D-NoC with partial TSVs. For irregular 3D-Noc topology, Holsmark *et al.* [11] and Luo and

Xian [13] proposed two different deadlock free routing algorithms for mesh and torus topologies respectively. Somasundaram *et al.* [22] proposed a 3D recursive topology and they proved that the topology is Hamiltonian connected. They also proposed a deadlock free routing algorithm.

Network congestion is another vital issues in NoCs. Any good routing schemes should minimize the congestion. Many soft computing techniques have been proposed for routing the cores with the minimum congestion. For example, L. Silva *et al.* [20] introduced a 3D ant colony routing (3D-ACR) algorithm for three different 3D topologies namely, mesh, torus and hypercube. Ahmed and Abdallah [1] proposed a fault-tolerant routing algorithm, called Hybrid-Look-Ahead-Fault-Tolerant (HLAFT).

The following are the main contributions in the paper: (i) construction of Hamiltonian connected recursive 3D-NoC topology (ii) routing algorithm for maximising the congestion factor during data pocket transmissions, (iii) router architecture for the proposed topology.

This paper is organized as follows: In section 2, we have given the mathematical model for the congestion problem. We have proposed a Hamiltonian connected topology for 3D-NoC. Also, we have given a flow algorithm to maximize the flow congestion factor in the network. In the next section, we have proposed a router architecture for our NoC topology. Our experimental results are given in Section 4.

2. Hamiltonian connected 3D topology and congestion optimization

2.1. Problem formulation

Any network can be realized as a graph $G = (V, E)$, where V and E are respectively set of nodes and links. Here, the cores are considered as nodes and the communication links are taken as edges of the graph. Throughout this paper, we assume n and m are the number of nodes and links in the network graph G, respectively. A path p in the network graph is an alternative sequence of nodes and links $v_1e_1, ..., v_ke_k, v_i \in V$ and $e_i \in E$. A Hamiltonian path is a path that traverse through all the nodes without a repeated node and a graph is said to be a Hamiltonian connected if contains at least one Hamiltonian path between any two nodes. We denote by $P^{(s,d)}$ is the set of all paths in G from a source node s to a destination node d. Each link $e \in E$ is assigned a *length* $l_e, l_e \in \mathbb{Z}^+$ and a *capacity* $C_e, C_e \in \mathbb{Z}^+$. We define the *flow* f_e of a link $e \in E$ as a non-negative quantity such that $f_e \leq C_e$. An *Augmenting path* p is a path from s to d such that $C_e - f_e > 0$, $\forall e \in p$. Given a path p in $P^{(s,d)}$, the length $L(p)$ of p is defined as the sum of lengths of its links. That is, $L(p) = \sum_{e \in p} l_e$. We denoted $\#X$ as the number of elements in the set $\#X$.

2.2. 3D topology and Hamiltonian connectedness

The degree of a node is the number of communication channels linked with that node. If all the degrees are two then the network is called cycle graph. Consider a cycle

graph G with 4 nodes and add a chord between a pair of non-adjacent nodes (shown in Fig. 1). In this paper, we have considered this graph G as our basic 2D module.

Let $l, k \in \mathbb{Z}^+$, we denote by $W(l, G, k)$ a 3D recursive network, where l is the number of layers and k is number of levels. If $l = 1$, then the topology is two dimensional. The basic module $W(1, G, 1)$ and the two level recursive structure $W(1, G, 2)$ are shown in Figures 1 and 2, respectively.

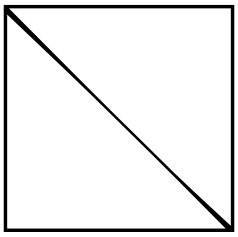

Figure 1. Basic module $W(1, G, 1)$

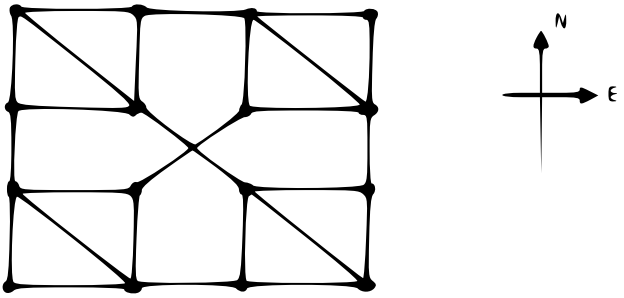

Figure 2. $W(1, G, 2)$

The basic module $W(1, G, 1)$ is also called Diamond graph. We denote NW, NE, SE and SW the four corner nodes of topology. We take four copies of $W(1, G, 1)$ and generated $W(1, G, 2)$. In general, $W(1, G, k)$ is obtained by joining the four copies of $W(1, G, k-1)$ with six links. There are $2^{2(k-1)}$ copies of the basic module $W(1, G, 1)$ in $W(1, G, k)$, for $k \in \mathbb{Z}^+$. Therefore, there are 4^k nodes in $W(1, G, k)$. Figure 3 shows the topology $W(3, G, 2)$.

Theorem 2.1. *Number of links in* $W(1, G, k)$ *is* $7 \times 4^{k-1} - 2$, $k \geq 1$.

Proof. We prove this result by induction on k. From Fig. 1, it is easy to see that the number of links in $W(1, G, 1)$ is 5. Suppose, $\#E(W(1, G, k-1)) = 7 \times 4^{k-2} - 2$. Now, $W(1, G, K)$ is formed by taking 4 copies of $W(1, G, k-1)$ and join the copies with six links. Therefore,
$\#E(W(1, G, k)) = 4 \times \#E(W(1, G, k-1)) + 6 = 4 \times (7 \times 4^{k-2} - 2) + 6 = 7 \times 4^{k-1} - 2$. $\square$

This topology has many advantages in terms of network parameters. Our experimental studies on this topology are shown in Section 4. The vertex connectivity (minimum number of vertices required to remove from the network so that the network becomes disconnected) of $W(l, G, k)$ is 3. This network topology is well connected, in the sense that it has a Hamiltonian path. This is proved in Theorem 2.4. We define $d(u, v)$ as the maximum distance between the two nodes u and v and $G_d = \max\{d(u, v)/u$ and v are any two nodes in $G\}$. This G_d is an important parameter, it ensures a routing path with a given hop count. It is easy to see that $W_d(1, G, k) = 4^k - 1, k \geq 1$.

Theorem 2.2. $W_d(l, G, k) = l \times 4^k - 1$.

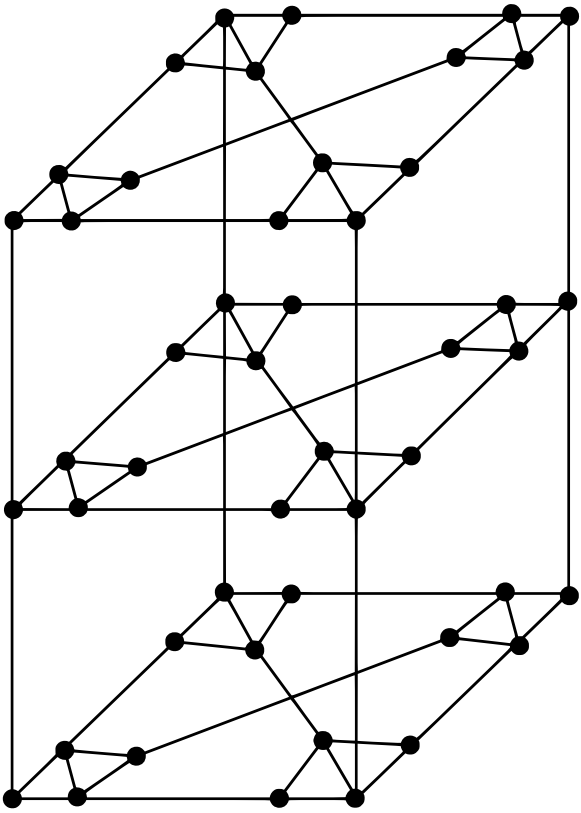

Figure 3. $W(3, G, 2)$

Hamiltonian connectedness is one of the important aspects of a topology. Since $W(l, G, k)$ is a recursive topology, we can easy to prove that $W(l, G, k)$ is Hamiltonian connected. Using mathematical induction on k we can prove that $W(1, G, k)$ is Hamiltonian connected and using the result we can prove that $W(l, G, k)$ is also Hamiltonian (again by induction on l).

Lemma 2.3. *W(1, G, k) is Hamiltonian connected, for* $k \geq 1$.

Theorem 2.4. *W(l, G, k) is Hamiltonian connected, for* $k \geq 1$ *and* $l \geq 2$.

There are $l4^k$ nodes in $W(l, G, k)$ and hence the length of the Hamiltonian path is $l4^k - 1$. That is, the maximum hop count of the network $W(l, G, k)$ is $l4^k - 1$. From the Theorem 2.2, it is easy to see that the bounds of the hop count $H(P)$ is $7 \times 4^{k-1} - 2 \leq H(P) \leq l \times 4^k - 1$. This upper bound is better than the fully mesh based topology.

In a network, the *flow* f_e cannot exceed its capacity C_e, that is, for every $e \in E$, $f_e \leq C_e$. Let $\mu(e) = \frac{f_e}{c_e}$ be the congestion factor during the data transmissions. For any path $p \in P^{(s,d)}$, the congestion factor is $\mu(p) = \sum_{e \in p} \mu(e)$. Our aim is to maximize the factor $\mu(p)$ for all $p \in P^{(s,d)}$. Therefore we are interested in maximizing the congestion factor during the routing in the network.

$$\text{Maximize} \sum_{p \in P^{(s,d)}} \mu(p),$$

Subject to the following constrains:

1. Total outflow at the source node is same as the total inflow at the destination node.
2. Total inflow and outflow are same at any internal nodes (nodes other than source and destination).
3. The congestion factor $\mu(e)$ cannot exceed unity.
4. If we assume the length of each link as unit then $L(p) \leq H(p)$ for all $p \in P^{(s,d)}$.

2.3. Algorithm for maximizing the congestion factor

Let us consider the network with fixed capacity C_e and dynamic data flow f_e. We are interested in maximizing the congestion factor $\mu(e), \forall e \in E$.

Theorem 2.5. *Let G be a network graph. Then maximum* $\mu(e)$ *is equivalent to minimizing the residual* $C_e - f_e$.

Proof.
$$\begin{aligned}
\text{Min } \{C_e - f_e : e \in E(G)\} &= \text{Min}\{\frac{C_e - f_e}{C_e} : e \in E(G)\} \\
&= \text{Min}\{1 - \frac{f_e}{C_e} : e \in E(G)\} \\
&= \text{Min}\{1 - \mu(e) : e \in E(G)\} \\
&= \text{Max } \mu(e).
\end{aligned}$$

Hence, maximizing the congestion factor $\mu(e)$ is equivalent to minimizing the residual $C_e - f_e$. □

Let us consider the network $W(l, G, k)$ as a graph G. Let s and d be respectively the source and destination nodes in $W(l, G, k)$. In the process of minimizing the residual, first we find a set of augmenting paths from s to d. Generally, enumerating all possible paths between two nodes in a graph is computational hard. Particularly, finding multipath from s to d is a NP-hard problem [2].

Here, we propose an algorithm to minimizing the residual. During the process, first we construct different paths from s to d. We use Breadth First Search (BFS) method to find the paths from s to d, in this way we can find augmenting paths with given length restrictions. Second part of our algorithms will optimize the residuals.

Algorithm for Minimizing the Congestion Residual Factor

1. Path Constructions (PC)
s: source node
d: destination node
$l(v)$: label of the node $v, v \in V$
p: augmenting path from s to d
$H(p)$: hop count
$V = \{s\}$
Let $l(s) = 0$ and set the level counter $\lambda = 0$
Initialize label counter $i = 1$
While $d \notin V$
{
Find the neighbors of v from BFS and add to the path p.
If $e = (v, w)$ be a link whose labeled end node v and unlabeled end node w then add to the link e with the path p such that $i \leq H(P)$.
Write $l(w) = i$
$V = V \cup \{w\}$ and $i = i + 1$
}
Return

2. Minimizing the Residual Capacity

Input: $W(l, G, k)$
Initialize: $f_e = 0$ for all links e
Construct a set of paths A_p using PC
While $A_p \neq \phi$
{
If p is a path in A_p
then $\Delta(p) = \min \{C_e - f_e : \text{for all } e \in p\}$
Refinement of the flow f_e :
$f_e + \Delta(p)$ for all links e, $e \in A_p$
$f_e - \Delta(p)$ for all links e, $e \notin A_p$
}
End

Theorem 2.6. *The flow algorithm will minimize the residual capacity during the routing.*

Proof. Let s and d be respectively the source and the destination nodes in the network. Let $P = \{p_1, p_2, ..., p_k\}$ be a set of paths from s to d.

Consider a path p_1 from P. The congestion residual of each link in p_1 is $r_e = C_e - f_e$. This residual is the current bandwidth of each link in path p_1. Now, we update the flow by adding this residual. During this process, capacity and flow are equal for at least one link in the path. We call such a link as a blocked link. Now, we choose another path p_i from P which does not contain the blocked link. We update the flow in the path by adding the congestion residual $r_e = C_e - f_e$. Here also, at least one link becomes a blocked link. By proceeding like this, we can update the flow and end up with some blocked links. The set of blocked links forms a cut set and this cut set will partition the network into two parts. Therefore, we cannot find any augmenting path from s to d. Sum of the flow values of the blocked links is equal to the total congestion residual. □

It is easy to see that the above algorithm takes a polynomial time for its computations since the BFS takes as many as number of links and we require only linear time to calculate $\Delta(p)$.

In a NoC, deadlock is a vital problem during the data packet transmissions. In the above algorithm, we construct paths based on BFS and these paths do not contain any cycles. Hence our algorithm will avoid the cycles and therefore our algorithm is free from cyclic deadlocks.

3. Router architecture with shared queues

Designing an optimal router architecture is one of the challenging problems in NoCs. The buffers are expensive and space consuming and they are not well used. Most of the time, only 40% of the buffer size is efficiently utilized while the remaining 60% is underutilized. All the buffers in the input ports may not have packets for transferring. A few input buffers will get packets while all others are regularly empty. To address this issue, different router architectures have been proposed based on the number of buffers. For example, in a Wormhole (WH) router the buffer is a single queue and in a Virtual-Channel (VC) router the buffer has many queues in parallel. Each of these parallel queues are known as VCs which enables temporary storage of packets from different queues, effectively reducing occurrence of packet stalling.

Gharan and Khan [15] have presented a dynamic VC organization and architecture for NoC systems. They proposed an input-port micro-architecture to support a dynamic virtual channel. A heterogeneous NoC router architecture was proposed by Itzhak *et al.* [3]. This architecture will support different link bandwidths and different number of VCs. Flexible elastic buffering for VC based NoCs (ElastiStore) was proposed by Seitanidis *et al.* [19]. They extended the elastic buffer architectures to support multiple VCs. This is a lightweight architecture with minimal buffering. In [24], Tran and Baas proposed a shared buffer router (RoShaQ) for high performance NoCs. This architecture optimizes the utilization of buffers. Andrew B. Kahng *et al.* [12]

showed an accurate power and area model for network routers (ORION 2.0) that are easily usable by system level designers. They showed a reproducible methodology for extracting inputs to their model from reliable sources. Pavan Poluri and Ahmed Louri [16] proposed a reliable NoC router architecture, which has ability to tolerate both hard and soft errors in the routing pipeline. They used spatial redundancy and exploitation of idle cycles techniques for their router design. In this paper, we modify the RoShaQ architecture with redirecting modules. Also, we implement this architecture with the proposed topology $W(l, G, k)$. The router architecture with shared queue improves the storage capacity and buffer utilization, which is shown in Figure 4.

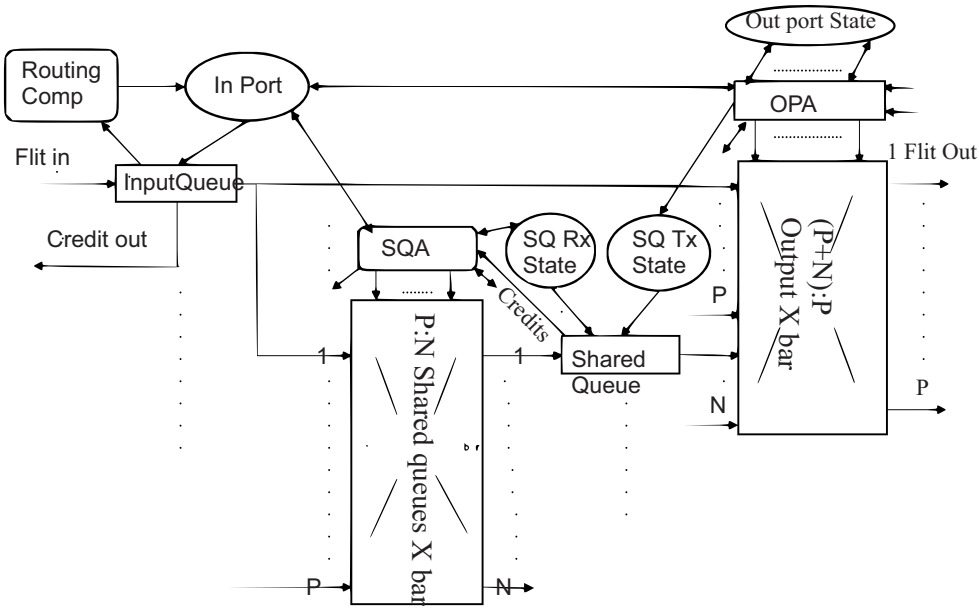

Figure 4. RoShaQ

In this architecture, the inactive queues have the ability to distribute their storage capacity to other input ports. First the packet will reach the input port. This is called queue write or buffer write. After that the packet will be sent to both of the state modules. A state module generally has three main functions. First, it will check whether the packet is a head flit, body flit or tail flit. Second, it will decide a particular output for the packet to traverse the crossbar. Third, it will generate a grant signal. The *Statemodule_one* will decide a shared queue for the incoming packet and the *statemodule_two* will decide a main output for the input packet to traverse the second crossbar. Then the state modules generate and send request signals to the shared queue allocator and output port allocator to indicate that there is a packet present at the input port. The request signal sent by the *statemodule_one* will go to the SQA (Shared Queue Allocator) and this will check whether the selected

shared queue is empty or full. If it is empty it will send a grant signal back. At the same time, the *statemodule_two* will also send a request signal to the OPA (Output Port Allocator) which will then check whether the requested output port is empty or full. If it is empty, it will send a grant signal back to indicate that the packet present in the input port can traverse the cross bar. Sometimes both the OPA and SQA will send their grant signals at the same time; then the priority will be given to the OPA, and the packet will traverse the second crossbar.

The SQA will receive requests from all the input queues and will generate grant signals only if: 1) the shared queues are empty or 2) if more than one input ports is requesting for the same output port. This shared queue packet storing policy ensures deadlock freedom for the networks. During light load conditions packets usually avoid shared queues, then the router operates in the WH mode. At heavy load conditions, a packet presents at the input queue fails to receive a grant signal from the OPA, but it can receive a grant signal from SQA and is then permitted to traverse through the shared-queue crossbar. In the next cycle, it then arbitrates for the selected output port and traverses the output crossbar to the given output channel. RoShaQ operates as an output-buffered router. This is a 3D router with 7 input/output ports.

3.1. Redirecting module

If any physical damage occurs in the physical output channel, then the data present in the input port requested for the output port in question will not receive a grant signal, leading to data loss. To avoid this problem, we introduce here a redirecting module. This module will first check if any output port is physically damaged or not. If a damaged output port is found then the module will send a busy bit. Also it will check if any input port is requested for that particular output port. If this is the case, it will look for another, undamaged empty output port and connect the input port to this output port, thereby avoiding packet loss due to a damaged output port in the router.

In general, it will work in the following way:

- *If any output port is physically damaged*
- *Busy bit will activated*
- *Check any of the input port request for that output port*
- *If so, check any other output port is empty*
- *Connect the input port to that output port and traverse the cross bar.*

4. Experimental results

In light load conditions, the data which is present in the input port directly traverses the second cross bar and reach to the output port. In heavy load conditions, the output port is busy and, data present in the input port traverses the first cross bar and is stored into the shared queue. Then a request signal is sent to the *statemodule_two*, and after getting a grant signal, the data traverses the second crossbar to reach

the output port. To avoid packet loss, we introduce the redirecting module, whose operation is shown in Figure 4. For our experiments, we consider the two topologies $W(3, G, 2)$ and Mesh of Grid topology (MoG), which are shown in Figures 3 and 5 respectively.

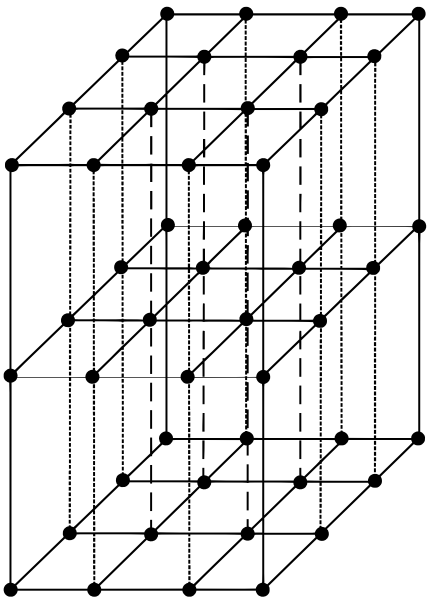

Figure 5. MoG

We simulate the topology $W(3, G, 2)$ to obtain various parameters like, throughput and latency. We assume that the node SW of the bottom layer is the source node and the node NE of the top layer is the destination node. Table 1 shows that throughput is 180.54 kbps and the latency is 0.4478 ns. The packet loss rate is measured as the percentage of dropped packets out of the total number of injected packets. We obtain the packet loss percentage which is 4.1%. These values are calculated for $W(3, G, 2)$, MoG topology and Partial MoG topology (PMoG) (Fig. 6), which are shown in Table 1. We observe that $W(3, G, 2)$ gives better results in throughput, packet loss percentage and latency than the other two topologies. Using the Synopsys Design Compiler, we evaluate the area and power. Table 2 shows the analysis of the 3D router with our topology $W(3, G, 2)$, with different flit sizes. We take the traditional VC router, the router based on ORION2.0 [12] and Shield [16]. The area requirement for the proposed work is 15746.4 μm^2 which is much lesser than other architectures. When we implement this in a $W(3, G, 2)$, the modified RoShaq consumes less than 25 % energy per packet than the traditional VC router.

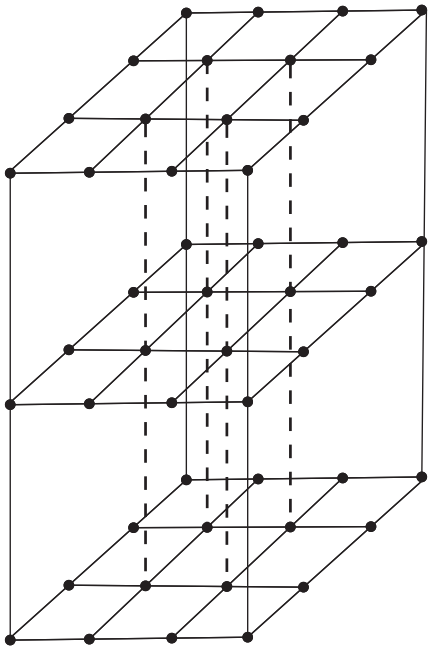

Figure 6. PMoG

Table 1
Performance metrics for PMoG, MoG, $W(3, G, 2)$

Topology	PMoG	MoG	$W(3, G, 2)$
Throughput (kbps)	145.78	72.09	180.54
Latency (ns)	0.4758	0.3830	0.4478
No. of packet sen	4134	4134	4134
No. of packet loss	170	142	130
Packet loss in %	4.1	3.4	3.1

Table 2
Modified RoShaq Design using different flit size

Parameters	32 bit	64 bit	128 bit
Area	65226.7 μm^2	81219.8 μm^2	15746.4 μm^2
Cell Internal Power	1.2944 mW	1.3566 mW	2.3355 mW
Total Dynamic Power	1.378 mW	1.542 mW	2.6669 mW
Cell Leakage Power	245.9647 μW	309.7439 μW	562.2573 μW

Table 3
RoShaq × VC

Topology	VC	Modified RoShaQ
Area	25125.3 μm^2	15746.4 μm^2
Cell Internal Power	6.187 mW	2.3355 mW
Total Dynamic Power	7.4589 mW	2.6669 mW
Cell Leakage Power	923.6941 μW	562.2573 μW

5. Conclusions

In a multi-core system architecture, Network on Chip emerged as a key architecture that optimize the various parameters like, power and area. In this paper, we addressed three main issues on NoC, namely, designing of an optimal topology, routing algorithm and a router design for the topology. First, we proposed a recursive 3D topology and a routing algorithm for data packet transmissions. We proved that our recursive network topology is Hamiltonian connected, our routing algorithm is free from cyclic deadlocks and the algorithm maximize the congestion factor. Our experimental results show that the propose topology gives better performance in terms of average latency and power than the other topologies. Finally, we proposed a router architecture for our 3D-NoC. The router architecture is based on shared buffers. Our experimental results indicated that the proposed router architecture consumes less area and power than the virtual channel architecture.

References

[1] Ahmed A.B., Abdallah A.B.: Graceful deadlock-free fault-tolerant routing algorithm for 3D Network-on-Chip architectures, *Journal of Parallel Distributed Computing*, vol. 74, pp. 2229–2240, 2014.

[2] Banner R., Orda A.: Mutlipath Routing Algorithms for Congestion Minimization, *IEEE Transection on Networking*, vol. 15(2), pp. 413–424, 2007.

[3] Ben-Itzhak Y., Cidon I., Kolodny A., Shabun M., Shmuel N.: Hetrogeneous NoC Router Architecture, *IEEE Transactions on Parallel and Distributed Systems*, vol. 26(9), pp. 2479–2492, 2015.

[4] Bjerregaard T., Mahadevan S., A Survey of Research and Practices of Network on Chip, *ACM Computing Survey*, vol. 38, pp. 1–51, 2006.

[5] Davis W.R., Wilson J., Mick S., Xu J., Hua H., Mineo C., Sule A.M., Steer M., Franzon P.D.: Demystifying 3D ICs: the pros and cons of going vertical, *IEEE Design and Test of Computers*, vol. 22(6), 498–510, 2005.

[6] De Micheli G., Benini L.: *Network on Chips: Technology and Tools*, Morgan Kaufmann Publishers, San Francisco CA, 2006.

[7] Dubois F., Sheibanyrad A., Pétrot F., Bahmani M.: Elevator-First: A Deadlock-Free Distributed Routing Algorithm for Vertically Partially Connected 3D-NoCs, *IEEE Transactions on Computers*, vol. 62, pp. 609–615, 2013.

[8] Eghbal A., Yaghini P.M., Bagherzadeh N., Khayamb M.: Analytical Fault Tolerance Assessment and Metrics for TSV-Based 3D Network-on-Chip, *IEEE Transactions on Computers*, vol. 64(12), pp. 3591–3604, 2015.

[9] Feero B.S., Pande P.P.: Networks-on-Chip in a Three-Dimensional Enviroment: A Performance Evaluation, *IEEE Transactions on Computers*, vol. 58(1), pp. 32–45, 2009.

[10] Hesham S., Rettkowski J., Goehringer D., Abd El Ghany M.A.: Survey on Real-Time Networks-on-Chip, *IEEE Transactions on Parallel and Distribute Systems*, vol. 28(5), pp. 1500–1517, 2017.

[11] Holsmark R., Palesi M., Kumar S.: Deadlock free routing algorithms for irregular mesh topology NoC systems with rectangular regions, *Journal of System Architecture*, vol. 54, pp. 427–440, 2008.

[12] Kahng A.B., Li B., Peh L.-S., Samadi K.: ORION 2.0: A fast and accurate NoC power and area model for early stage design space exploration. In: *Proceedings of 2009 Design, Automation & Test in Europe Conference & Exhibition*, pp. 423–428, 2009.

[13] Luo W., Xiang D.: An Efficient Adaptive Deadlock-Free Routing Algorithm for Torus Networks, *IEEE Transactions on Parallel and Distributed Systems*, vol. 23(5), pp. 800–808, 2012.

[14] Mamaghani S.M., Jamali M.A.J., An adaptive congestion-aware routing algorithm based on network load for wireless routers in wireless network-on-chip, *AEU International Journal of Electronics and Communications*, vol. 97, pp. 25–37, 2018.

[15] Oveis-Gharan M., Khan G.N.: Efficient Dynamic Virtual Channel Organization and Architecture for NoC Systems, *IEEE Transactions on Very Large Scale Integration Systems*, vol. 24(2), pp. 465–478, 2016.

[16] Poluri P., Louri A.: Shield: A Reliable Network-on-Chip Router Architecture for Chip Microprocessors, *IEEE Transactions on Parallel and Distributed Systems*, vol. 27(10), pp. 3058–3070, 2016.

[17] Rusu C., Anghel L., Avresky D.: Adaptive inter-layer message routing in 3D networks-on-chip,*Journal of Microprocessors and Microsystems*, vol. 35, pp. 613–831, 2011.

[18] Sahu P.K., Chattopadhyay S.: A survey on application mapping strategies for Network-on-Chip design, *Journal of System Architecture*, vol. 59, pp. 60–76, 2013.

[19] Seitanidis I., Psarras A., Chrysanthou K., Nicopoulos C., Dimitrakopoulos G.: ElastiStore: Flexible Elastic Buffering for Virtual-Channel-Based Networks on Chip, *IEEE Transactions on Very Large Scale Integration Systems*, vol. 23(12), pp. 3015–3028, 2014.

[20] Silva Jr. L., Nedjah N., De Macedo Mourelle L.: Efficient routing in network-on-chip for 3D topologies, *International Journal of Electronics*, vol. 102(10), pp. 1695–1712, 2015.

[21] Somasundaram K., Plosila J.: Deadlock Free Routing Algorithm for Minimizing Data Packet Transmission in Network on Chip, *International Journal of Embedded and Real Time Communication Systems*, vol. 3(1), pp. 70–81, 2012.

[22] Somasundaram K., Plosila J., Vishwanathan N., Deadlock free routing algorithm for minimizing congestion in a Hamiltonian connected recursive 3D-NoCs, *Microelectronics Journal*, vol. 45, pp. 989–1000, 2014.

[23] Tatas K., Siozios K., Soudris D., Jantsch A.: *Designing 2D and 3D Network-on-Chip Architectures*, Springer-Verlag, New York, 2014.

[24] Tran A.T., Baas B.M., Achieving High Performance On-Chip Networks With Shared-Buffer Routers, *IEEE Transactions on Very Large Scale Integration Systems*, vol. 22(6), pp. 1391–1403, 2014.

[25] Valinataj M., Mohammadi S., Plosila J., Liljeberg P., Tenhunen H.: A reconfigurable and adaptive routing method for fault-tolerant mesh-based networks-on-chip, *AEU International Journal of Electronics and Communications*, vol. 65(7), pp. 630–640, 2011.

[26] Viswanathan N., Paramasivan K., Somasundaram K.: An optimised 3D topology for on-chip communications, *International Journal of Parallel, Emergent and Distributed Systems*, vol. 29(4), pp. 346–362, 2013.

[27] Wang Y., Han Y.-H., Zhang L., Fu B.-Z., Liu C., Li H.-W., Li X.: Economizing TSV Resources in 3-D Network-on-Chip Design, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23(3), pp. 493–506, 2015.

Affiliations

Somasundaram Kanagasabapathi
Amrita School of Engineering-Coimbatore, Department of Mathematics, Amrita Vishwa Vidyapeetham, India, s_sundaram@cb.amrita.edu,
ORCID ID: https://orcid.org/0000-0003-2226-1845

Chythanya Calicut
Amrita School of Engineering-Coimbatore, Department of ECE, Amrita Vishwa Vidyapeetham, India, chythanyacalicut@gmail.com

Received: 16.05.2019
Revised: 20.08.2019
Accepted: 21.08.2019