International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-1 Issue-2, December 2011

Deadlock Free Routing in Irregular Interconnection Networks for Complex SoCs

Naveen Choudhary

Abstract*: *Networks-on-Chip (NoC) is recently proposed as an alternative communication infrastructure/Interconnection Network for address the high communication demands of the complex futuristic SoCs. Most researchers advocate the use of traditional regular networks like meshes as architectural templates which gained a high popularity in general-purpose parallel computing. However, most SoC platforms are special-purpose tailored to the domain-specific requirements of their application. They are usually built from a large diversity of heterogeneous components which communicate in a very specific, mostly irregular way. In such systems the size and nature of cores may vary quite widely making the topology irregular. Moreover regular topologies can become irregular due to faults in links and switches. In such scenario topology agnostic routing algorithms are generally required. In this paper, a survey of various deadlock free table based routing function is presented. The paper presents survey of deadlock free routing function with and without the availability of virtual layers

***Index Terms*: Interconnection Networks, System on Chip, Routing, Deadlock, Virtual Layers**

I. Introduction

Early works in the domain of interconnection networks for SoCs in nano scale regime favored the use of standard topologies such as meshes, tori, k-ary n-cubes or fat trees with the assumption that the communication channels can be well organized in such interconnection networks. These interconnection networks are appropriate for generic systems but as we know most SoCs are heterogeneous in nature, with each core having varying size and communication requirements. Therefore standard regular interconnection networks/topologies may have a undesired interconnection structure for application specific communication, which can result in poor application mapping and increased channel complexity. Moreover A generic regular interconnection network can become irregular due to faults and logical or physical clustering Therefore the use of interconnection network independent routing function is highly desirable for complex futuristic SoC with billions of transistors.

Routing in irregular interconnection networks can be performed by using source routing or routing based on table lookup. For source routing function, the source node specifies the routing keeping in the view the desired deadlock freedom. The computed path is required to be stored in the packet header to reserve channels of the path. In source routing the packet headers tends to be very long as the routing path is required to be stored in the packet header. For efficiency reasons most high end SoCs have proposed the use of table based deadlock free routing. In table based routing, each intermediate router has to make a routing decision depending on the table entries in the router characterizing the local knowledge of the networks. By repeating this process at each intermediate router, the packet is expected to reach its destination. As deadlock-free routing is critical for proper operation of such irregular interconnection networks several interconnection independent deadlock-free routing functions were proposed in this domain such as LASH [1], TOR [2], LASH-TOR [3], DL [4], and multiple virtual networks [5]. All the above mentioned routing functions require virtual channels/layers, a feature that not all technologies may support. Moreover even if the virtual layers are supported there may be a restriction on the number of available virtual layers. For such scenarios there is a different set of routing functions such as up*/down* [6], lturn [7], down/up [8], prefix-routing [9], smart-routing [11], and FX [10], which do not require the availability of virtual layers to provide efficient deadlock free routing in the interconnection networks for SOCs of nano scale regime.

In this paper we present a survey of such routing schemes proposed in this research domain to analyze the functioning, limitations and efficiency of the various deadlock free routing functions proposed in the literature for of interconnection networks for high end SoCs. In Section 2 we present the concept and utility of Virtual Layers in the interconnection Networks. Section 3 presents a prominent deadlock free routing functions proposed in the literature which do not require the availability of virtual layers. Section 4 present a survey of segment based routing functions. Section 5 presents some popular high performance deadlock free routing functions requiring the availability of virtual layers for the interconnection networks and Section 6 we conclude.

II. Virtual Layers

Virtual channels or virtual layers play a prominent role in addressing the various flow control issues in interconnection networks for high end SoCs. In the following, the concept of virtual channels and their use in flow control is discussed. Virtual channels (VCs) [12] are basically meant for sharing of a physical channel by several logically separate channels with individual and independent buffer queues. Generally 2 to 16

Manuscript published on 30 December 2011.
* Correspondence Author (s)

Naveen Choudhary*, Department of Computer Science and Engineering, College of Technology and Engineering, MPUAT, Udaipur, India, (e-mail: naveenc121@yahoo.com).

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC-BY-NC-ND license http://creativecommons.org/licenses/by-nc-nd/4.0/

VCs per physical channel have been proposed for the interconnection networks. Their implementation results in an area and possibly also power and latency overhead due to the cost of control and buffer implementation. There are however a number of advantageous of using Virtual channels/virtual layers as mentioned below.

- Avoiding Deadlocks: Since Virtual channels are not mutually dependent on each other, by adding VCs to links and choosing the routing scheme properly, one may break cycles in the resource dependency graph and thus deadlocks can be avoided.
- Optimizing Wire Utilization: In future technologies, wire costs are projected to dominate over transistor costs. Letting several logical channels share the physical wires, the wire utilization can be greatly increased. Advantages include reduced leakage power and wire routing congestion.
- Improving Performance: VCs can generally be used to relax the inter-resource dependencies in the network, thus minimizing the frequency of stalls. it can be shown that dividing a fixed buffer size across a number of VCs improve the network performance at high loads. Moreover virtual layers can be helpful in implementing adaptive routing protocols.
- Providing QoS Services: Quality-of-service (QoS) can be used as a tool to optimize application performance. VCs can be used to implement such services by allowing high priority data streams to overtake those of lower priority or by providing guaranteed service levels on dedicated connections .

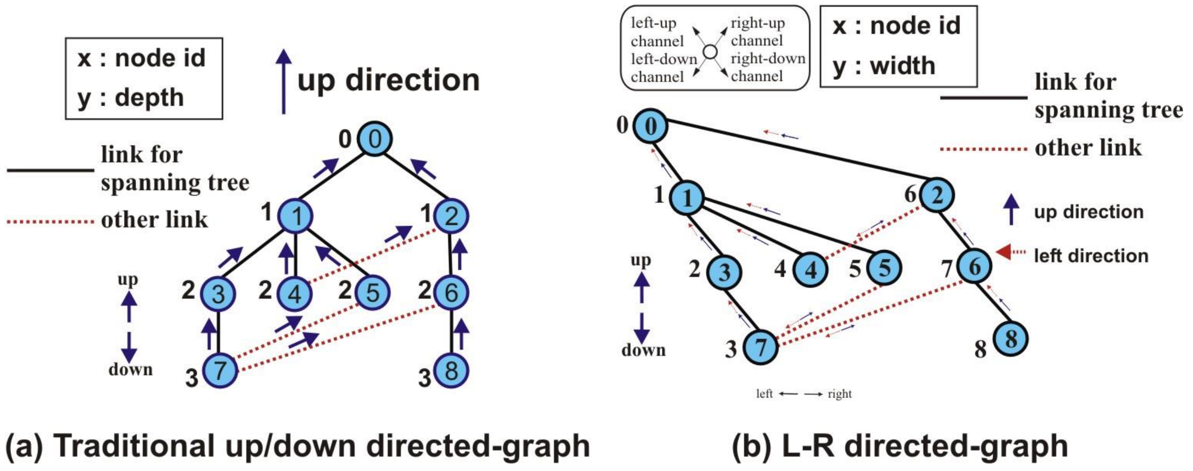
III. Generic Turn Prohibition Based Routing

In order to make the best use of inherent bandwidth, adaptive routing which can avoid the congestion is essential for irregular networks. However, in traditional routing algorithms for irregular networks, available paths are considerably restricted in order to avoid deadlocks. Consequently, traditional routing algorithms such as up*/down* [6] which can not use all of links effectively often cause unbalanced traffic and increases the average hops count for the packets. L-turn routing [7] is based on a logical spanning tree like traditional routing algorithms. However, it distributes the traffic by building a novel directed-graph called the L-R directed-graph. Consequently, all channels are effectively used in various types of topologies. L-turn routing also mitigates a traditional problem by which a traffic concentrates around the root node in up*/down* [6] routing.

In up*/down* based routings, all channels are classified into “up” or “down” channel, and a strict rule between them is applied. So they cannot use all of links effectively, and often cause unbalanced traffics. Especially, the heavy traffic around the root node causes a congestion which degrades the total performance. In L-turn routing, the structure of directed graph itself is modified to L-R tree and L-R graph in order to distribute the traffic evenly. The L-R tree and corresponding L-R graph is constructed using the following steps.

- A BFS (Breadth first) spanning tree is built in the same way as in traditional up/down based routings. Figure 1(a) shows a spanning tree used in traditional up/down based routings. Note that a link provides bidirectional channels.
- An increasing number (width) is assigned to each node in the order such that nodes are visited by the pre-order traversal starting from the root node. In the example shown by Figure 11(b), the order of the traversal is (0, 1, 3, 7, 4, 5, 2, 6, 8). Width represents a horizontal distance from the root node.
- Horizontal direction is assigned to each channel between any two nodes by comparing their widths as follows : Left direction is assigned to the channels whose destination nodes have the smaller widths, and right direction is assigned to the other channels.

Vertical direction (up or down direction) is assigned to each channel between any two nodes by comparing their depths (distances from the root node in vertical direction) as in up*/down* routing. In figure 1(b), an arrow with a thick line shows the up direction, and an arrow with a dotted line shows the left direction. A channel that faces towards the left direction is called left channel, and a channel that faces toward the right direction is called right channel. Unlike up*/down* tree in figure 1(a), an L-R tree has two dimensional directions: up/down and left/right as shown in figure 1(b). An L-R tree does not include any channels which are out of spanning tree in the original irregular network, and



(a) Traditional up/down directed-graph

(b) L-R directed-graph

Fig 1: An example of up/down directed-graph and L-R directed graph

the L-R directed-graph is built by adding such channels into L-R tree. Thus, the L-R directed-graph is topologically equivalent to the original irregular network. Channels with the dotted lines in figure 1 are added in both cases that is an up/down directed-graph and L-R directed graph are built. Eliminating them makes an up/down tree or L-R tree in each case. Note that different L-R trees can be generated from an original network depending on the assignment of width to the nodes in a BFS spanning tree.

Left/Right Routing Algorithm : In left/ right routing we are allowed to choose any path in the graph with the restriction that left channels are not allowed to be used after using the right channel. This routing algorithm is deadlock free as each cycle contains at least one left channel and one right channel and so cyclic channel dependency can not exist. Moreover the path between any pair of nodes is guaranteed by selecting a sequence of left channels (if any) followed by a sequence of right channels (if any) as each left channel and right channel which belongs to the L-R tree corresponds to each up channel and down channel which belongs to the corresponding up/down tree respectively. The left/right routing is also livelock free if a packet must select the path whose length is equal to or shorter than the one using only channels which belong to L-R tree.

The above mentioned conditions satisfy deadlock freedom for the routing function since all possible cycles containing left-up channel are ruled out with the help of condition 1 and all possible cyclic channel dependencies which don't include any left-up channels are based on one of the four patterns shown in figure 2, and always include at least one turn from left-down channel to right-up or right-down channel. So, the condition 2 (restriction 2) breaks all cyclic channel dependencies which don't include any left-up channels.

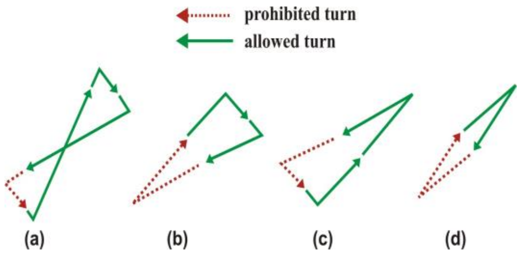


Fig 2: Cyclic channel dependency in specific cases.

In order to reduce redundant prohibited turns among left-down to right-up and left-down to right-down, the specific topology graph in question is looked for channel dependency cycles including these (left-down to right-up and left-down to right-down) turns and then for each node where the turn left-down to right-up or left-down to right-down are not required to be prohibited then such turns are removed from the list of prohibited turns from such nodes in the topology graph. The following algorithm detects the above two type of turns for each node on a given topology graph and then only prohibit the minimum required turns among these (left-down to right-up and left-down to right-down) turns on each node of the topology graph.

- All nodes which satisfy one of the below conditions are selected:
  - (a) Two or more right-up channels exist (generating the turn from left-down to right-up),
  - (b) One or more right-up channels and one or more right-down channels exist (generating the turn from left-down to right-down).
- For each selected node, in the order of the depth-first search, it is checked whether there are cyclic channel dependencies which include selected node and don't include any left-up channels.
- If the selected node satisfies 1-(a), the check is executed as follows. At first, an adjacent node which can move from the selected node by using right-up channel is visited. The output channel used for the visit is marked. Then, an adjacent node is visited by using available output channels recursively. If there are no available output channels or cyclic channel dependency is detected at a visiting node, the checking process returns to the previous node. An available output channel satisfies below conditions: a) Not left-up channel, b) Not marked, and c) Doesn't make turn which is already prohibited by previous search.
- If the selected node is visited again through left-down channel, the cyclic channel dependency which includes the turn from left-down to right-up and no left-up channels is detected. Then, the above turn is prohibited.

The check continues until there are no available output channels or all possible cyclic channel dependencies are detected. If the selected node satisfies 1-(b), the check is executed in almost the same way except for the below conditions.

1) The check starts by using right-down channel.

2) In order to reduce redundant prohibited turns, output channel which makes turn from left-down to right-up is not available.

The presented algorithm basically decreases the redundant prohibited turns. In left/right routing the turns which are opposite to each other are prohibited. This makes the concentration of prohibited turns at the same node and may cause unbalanced traffic. Similar disadvantage also applies to up*/down* based routing. On the other hand, L-turn prevents the above situation by distributing the prohibited turns. Moreover, by defining the channels as left-up, left-down, right-up and right-down prohibited turns at two connected nodes at the same depth are distributed uniformly. The distribution of prohibited turns in L-turn routing will contribute to better traffic balancing and so high throughput routing.

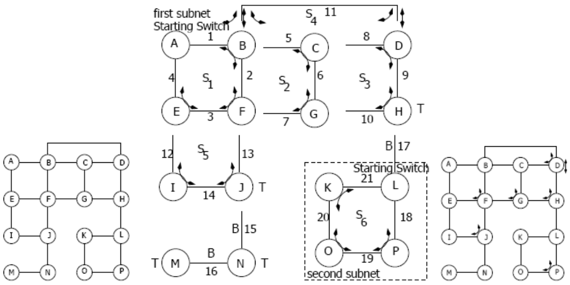
L-turn routing is deadlock free since it breaks all possible cyclic channel dependencies by prohibiting at least one of turns which belongs to each cyclic channel dependency in the channel dependency graph of the topology in question. As each left-up channel and right-down channel which belongs to the L-R tree corresponds to each up channel and down channel which belongs to the corresponding up/down tree respectively. Consequently, the path between any pair of nodes is guaranteed by selecting a sequence of left-up channels (if any) followed

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)
© Copyright: All rights reserved.

by a sequence of right-down channels (if any). To prevent a livelock in L-turn routing, a packet must select the path whose length is equal to or shorter than the one using only channels which belong to L-R tree.

IV. Fault Tolerant Routing

In [13] mejia et al. proposed a new efficient fault tolerant segment based deterministic routing (SR) methodology for tori and meshes, which achieves high performance without the use of virtual channels. Furthermore, it is topology agnostic in nature, meaning it can handle any topology derived from any combination of faults (link, switch failures) when combined with reconfiguration. Segment-based Routing (SR) algorithm works by partitioning a topology into subnets, and subnets into segments. This allows the bidirectional turn restrictions to be placed locally within a segment. As segments are independent, there is freedom to place turn restrictions within a segment independently from other segments. This results in a larger degree of freedom when placing turn restrictions compared to other routing strategies. The Segmented Routing is a three step process. First segmentation of the topology is performed. Then, the selection of the turn prohibitions on each segment is done and finally the most suitable path between every source destination pair is selected. The routing segment and subnets can be defined as follows.



(a) Topology (b) segments, subnets and possible prohibitions (c) Final

Fig 3: Network Segments and subnets for Irregular 2D-Mesh

Routing Segment : A segment is defined as a list of interconnected switches and links. Routing segment can be defined as collection of interconnected switches and links such that no network switch or link belong to more than one routing segment (ie segments are disjoint). Moreover for every routing segment, except the initial segment the routing segment start and end on the switch already part of a previously computed segment. There can be three type of routing segments as mentioned below.

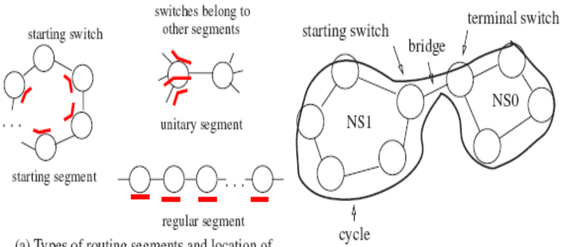
Starting segment : This type of routing segment will start and end on the same switch, thus forming a cycle. This routing segment will be found probably every time a new subnet is initiated.

Regular segment : This type of routing segment will start on a link, will contain at least one switch, and will end on a link.

Unitary segment :: This type of routing segment consists of only one link.

Sebnets: Routing segments are grouped into subnets. A subnet is a set of switches and links (i.e. one or more segments) that is connected to the rest of the network (other subnets) through only one link. Figure 3 shows an example of routing segments and subnets constructed for irregular 2D-Mesh (irregularity is generated due to faulty links) with the help of the algorithm proposed in [13].

Segment Based Routing Restrictions : In order to ensure deadlock-freedom and preserve connectivity, the routing algorithm in segment-based routing must place the routing restrictions in each routing segment as shown in figure 4(a). In particular, for a starting segment, a bidirectional turn prohibition can be placed on any switch except the starting one (as it may introduce a cycle among different subnets). By doing this, we avoid all the cycles in the starting segment of the topology. For regular segments, one bidirectional turn prohibition can be placed on any of the switches belonging to the segment. This is done in order to break any possible loop crossing through the segment. Finally, for unitary segments, all the traffic crossing the link to an already restricted segment must be avoided in order to prevent deadlocks. Thus, in one side of the unitary segment bidirectional turn prohibitions must be placed for every link attached to the switch that connects to already restricted segment. Notice that unitary segments prevent most of the traffic to pass through the link. The unconnected subnets can be connected with the addition of a link called bridge. Whenever a terminal switch is present in a subnet, it will be attached to the starting switch of another subnet and the corresponding link will not belong to any subnet or any segment. This link will be referred to as a bridge. Figure 4(b) shows an example. The starting switch and terminal switch are defined as follow.



(a) Types of routing segments and location of routing restrictions (b) Example of links connecting subnets

Fig 4: (a) Segments and restrictions, and (b) subnets.

Starting switch: A switch is marked as the starting switch if it is the first switch chosen to compute a segment within a subnet.

Terminal switch: A switch is marked as terminal if, through at least one of its links, no new segment is found. In figure 3, the routing segment's switch H can be treated as terminal switch as H has an extra link (link 17) which is not part of the segment S3 and also through switch H, no new segment can be found. Whereas link 9 & link 10 are components of the same segment S3.

The Segment-Based Routing algorithm (SR) concept is extended further in Region based routing [14] with the objective of reducing the size of routing tables. Ali et al in [15], introduces the ideas of dynamic routing in context of NoCs and explains how they could be applied to cope with adverse physical effects of deep sub-micron technology like transient and permanent failures of link, router, signals etc. In particular

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

© Copyright: All rights reserved.

a modified link state routing referred as NoC-LS is proposed in [15] to overcome transient and permanent failures of different components of NoC.

V. Generic Virtual Layer Based Routing

There are a different set of several high performance topology agnostic deadlock-free routing algorithms such as LASH [1], TOR [2], LASH-TOR [3], DL [4], and multiple virtual networks [5] which requires the presence of different virtual channels/layers in the irregular NoC network. LASH [1] and TOR [2] are discussed briefly here as they happen to be the prominent routing methodologies in this set.

The LASH (layered shortest path routing) was proposed in [1]. LASH routing is a deterministic shortest path routing of irregular networks in which all packets are routed minimally for any irregular topology, and in which all packets are delivered inorder. The concept assumes the presence of virtual channels divided into virtual networks (layers) to avoid deadlock. The figure 5 shows an example of network having three virtual layers/networks.

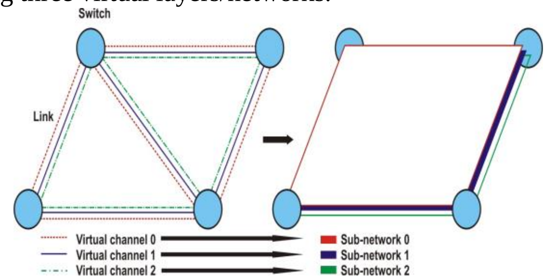


Fig 5: Three virtual networks for a single physical network with the help of three virtual layers per node.

In LAyered SHortest path (LASH) routing, the freedom from deadlocks is attained by dividing the traffic into different virtual layers using virtual channels. The routing function R is defined by two sub-functions - R_{phys} and R_{virt}, respectively. The former defines one minimal physical path for each <source, destination> pair. The latter determines on which virtual layer (set of channels) packets from each <source, destination> pair should be forwarded along the minimal paths specified by R_{phys}.

The number of <source, destination> pairs in a network grows with the square of the number of nodes. The term granular unit (*gu*) is used to designate a set of <source, destination> pairs such that channels dependency graph for the set of <source, destination> pairs is acyclic. In other words a granular unit's channel dependency graph is acyclic and therefore a granular unit is deadlock free.

The Basic idea in LASH [1] is to assign a set of granular units to different virtual channel layer in such a way that each virtual layer remains deadlock free. Moreover no packets are allowed to switch between virtual layers. The theoretical upper bound on the number of virtual channel layers for an interconnection network with n switches is [n/2] + 1 but in LASH [1], it is shown with extensive experimentation that the number of virtual layers practically required is much less. The number of virtual layers required in LASH [1] for 16, 32, 64 and 128 switches were 2, 3, 5 and 9 respectively. Generally the ratio between network size (number of switches) and required number of virtual layer increases as the network size grows.

Sancho et al. in [2] proposed an effective methodology for designing deterministic deadlock-free routing strategies that are able to route packets through minimal paths in irregular networks. The methodology can meet the trade-off between network performance and the number of resources dedicated to deadlock avoidance. The Stages of the Deadlock-free routing table generation methodology for deterministic routing proposed in LASH (layered shortest path routing) [2] can be summarized as follow.

1. At the first stage all the possible minimal paths between every pair of hosts in the network are computed. Notice that although several minimal paths will be computed for the same <source destination pair>, but only one of them will be selected at the later stage. The routing paths selected at this stage do not guarantee deadlock freedom because cyclic channel dependencies may arise.
2. At the second stage, possible cyclic channel dependencies are searched based on up*/down* routing scheme. In this stage every computed minimal path containing any forbidden transition according to the up*/down* rule is marked for later treatment. As we know in up*/down* routing, a forbidden transition occurs when the path uses a link in the "up" direction after having used one in the "down" direction. Here we can note that the cyclic channel dependencies can be removed by using virtual lanes. An easy way of doing this is to route a packet through a new different virtual lane every time a forbidden transition is found. In turn, cyclic channel dependencies between virtual lanes can be removed by forcing their use in a sequential (increasing or decreasing) order.
3. In the third stage of the methodology, those paths are selected that minimizes the number of forbidden transitions. It should be noted that after selecting paths there still may be some pair of hosts with more than one minimal path.
4. At the fourth stage mapping conflicts are resolved by trying to find alternative minimal paths for the conflicting paths. Moreover if finding such alternative paths is not possible due to lack of available number of virtual channels then a non-minimal path is computed according to the up*/down* routing algorithm.
5. After solving the mapping conflicts in stage 4, the methodology has obtained a deadlock-free set of paths with more than one minimal path per each source-destination pair. However, as the required routing is deterministic only one path per source-destination pair is selected in the fifth stage. For this purpose, the traffic balancing algorithm proposed in [16] is used, which tries to balance the number of paths crossing every channel. By doing this, a balanced and deadlock-free set of paths with one minimal path per each source-destination pair are obtained.
6. In stage six the methodology is extended with a step to balance the virtual channels utilization. As per the nature of the above mentioned steps, the virtual layer L1 will be most filled, L2 will be second most filled one, etc and Ln may only

contain very few sub-paths. This could result in poor utilization of the network resources. It is observed that most of the paths assigned to virtual layer L1 are complete paths ie these path need only virtual layer for their complete traversal. Such shortest paths of virtual layer L1 are selected and are reassigned to layer Lj (1 < j <= |L|) if it does not close a channel dependency cycle in layer Lj for better load balancing on virtual layers.

7. Finally, at the seventh stage, the routing information for the routers of the NoC under consideration is generated.

Simulation results presented in [2] show that in the worst case (large networks) up to 3 virtual layers could be necessary to guarantee deadlock-free minimal routing. The needed network resource decreases as the network size decreases. In particular, only 2 virtual layers are needed for 16-switch networks.

VI. Conclusion

Interconnection network for SoC domain of research is still in the early stages. Even though commercial products currently exhibit only a few integrated cores, in the next few years the technology will support the integration of thousands of cores, making a large computational power available.

The scalable and modular nature of interconnection networks for SoCs and their support for efficient on-chip communication potentially leads to SoC-based multiprocessor systems characterized by high structural complexity and functional diversity. It is observed that high end SoC based systems are economically feasible if they can be used in several product variants, and if the design can be reused in different application areas. On the other hand, successful products must provide good performance characteristics, thus requiring dedicated solutions that are tailored to specific needs. Moreover the regular interconnection networks tends to become irregular due to faults for clustering. In such irregular interconnection networks it is generally not efficient to use a fixed regular network structures, instead an irregular interconnection network will be more appropriate. The deadlock free efficient routing for such irregular interconnection networks is a complex issue and has been addressed in the research domain to a good extent. In this paper we have surveyed the proposed routing issues and algorithms in irregular interconnection networks for SoC research domain to give an insight to the readers regarding the routing issues in irregular interconnection networks.

References

1. O. T. Skeie, I.Theiss, "Layered Shortest Path (lash) Routing in Irregular SAN.", In Proceedings of the International Symposium on Parallel and Distributed Processing (IPDPS), pp. 162-169, 2002.
2. J.C.Sancho, A. Robles, J. Flich, P. Lopez, J. Duato, "Effective Methodology for Deadlock-Free Minimal Routing in Infiniband Networks. In International Conference On Parallel Processing, Aug. 2002.
3. T.Skeie, O. Lysne, A. Robles, J. Flich, P. Lopez, J. Duato, " Lash-tor: A Generic Transition-Oriented Routing Algorithm", In IEEE International Conference on Parallel and Distributed Systems, July 2004.
4. M.Koibuchi, A. Jouraku, K. Watanable, H. Amano, "Descending Layers Routing: A Deadlock-Free Deterministic Routing using Virtual Channels in System Area Networks with Irregular Topologies", In International Conference on Parallel Processing, pp. 527-536, 6-9 Oct. 2003.
5. J.Flich, J. C. Sancho, A. Robles, P. Lopez, J. Duato, "Improving Infiniband Routing through Multiple Virtual Networks. In Proceedings of the 4th International Symposium on High Performance Computing, pp. 49-63, 2002.
6. M. D. Schroeder et al., "Autonet: A High-Speed Self-Configuring Local Area Network Using Point-to-Point Links", In Journal on Selected Areas in Communications, vol. 9, Oct. 1991.
7. Jouraku, A. Funahashi, H. Amano, M. Koibuchi, "L-turn routing: An Adaptive Routing in Irregular Networks", In the International Conference on Parallel Processing, pp. 374-383, Sep. 2001.
8. Y.M. Sun, C.H. Yang, Y.C Chung, T.Y. Hang, "An Efficient Deadlock-Free Tree-Based Routing Algorithm for Irregular Wormhole-Routed Networks Based on Turn Model", In International Conference on Parallel Processing, vol. 1, pp. 343-352, Aug. 2004.
9. J. Wu, L. Sheng, "Deadlock-Free Routing in Irregular Networks Using Prefix Routing", DIMACS Tech. Rep. 99-19, Apr. 1999.
10. J.C.Sancho et al., "A Flexible Routing Scheme for Networks of Workstations", In the International Conference on High Performance Computing, Oct. 2000.
11. V. L.Cherkasova, T.Rokicki, "Fibre Channel Fabrics: Evaluation and Design", In Proceedings of the 29th International Conference on System Sciences, Vol. 1, pp. 53-62, 3-6 Jan. 1996.
12. J. Duato, S. Yalamanchili, L. Ni, "Interconnection Networks: An Engineering Approach", Elsevier, 2003.
13. Mejia, J. Flich, J. Duato, S.A. Reinemo, T. Skeie, " Segment Based Routing: An Efficient Fault-Tolerant Routing Algorithm for Meshes and Tori", In IEEE Proceeding of 20th International Conference on Parallel and Distributed Processing Symposium (IPDPS 2006), April 2006.
14. J. Flich, A. Mejia, P. L'opez, J. Duato, "Region-Based Routing: An Efficient Routing Mechanism to Tackle Unreliable Hardware in Network on Chips", In IEEE Proceeding on Networks on Chip, pp.183- 194, May 2007.
15. M. Ali, M. Welzl, S. Hellebrand, "A Dynamic Routing Mechanism for Network on Chip", In IEEE Proceeding of 23rd NORCHIP Conference, pp. 70-73, Nov. 2005.
16. J. C. Sancho, A. Robles, "Improving the Up*/down* Routing Scheme for Networks of Workstations", In the Proceedings of Euro-Par 2000, Aug. 2000.

Dr. Naveen Choudhary received his B.E, M.Tech and PhD degree in Computer Science & Engineering. He completed his M.Tech from Indian Institute of Technology, guwahati, India and PhD from Malviya National Institute of technology, Jaipur, India in 2002 and 2011 respectively. Currently he is working as Associate Professor and Head, department of Computer Science and Engineering, College of Technology and Engineering, Maharana Pratap University of Agriculture and Technology, Udaipur, India.

His research interest includes Interconnection Networks, Network on Chip, Distributed System and Information Security. He is a life member The Indian Society of Technical Education, Computer Society of India and The Institution of Engineers, India. E-mail: naveenc121@yahoo.com