ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, *Vol 9(17), DOI: 10.17485/ijst/2016/v9i17/93120, May 2016* ISSN (Online) : 0974-5645

An Effective on-Chip Network Topology for Network on Chip (Noc) Trade-Offs

M. Venkateswara Rao[1*], T. V. Rama Krishna[1], S. Raaga Sai Sruthi[2], S. Akhila[2], Y. Gopi[2] and L. Bhavani Krishna[2]

[1]Department of Electronics and Communications Engineering, K L University, Vaddeswaram, Guntur – 522502, Andhra Pradesh, India; venkatvlsi@kluniverity.in; tottempudi@kluniversity.in
[2]Department of Electronics and Computer Engineering, K L University, Vaddeswaram, Guntur - 522502, Andhra Pradesh, India; srikaram.sruthi4@gmail.com, akhila11.95@gmail.com, yaddanapudi.gopi123@gmail.com, navya.lankisetti@gmail.com

Abstract

Background/objectives: The major contribution in very large scale integrated circuits is given by System on Chip (SoC) technology. Many fuzzy logics are been implemented on SoC's and are effectively supported by them. In time the Networks came into existence and interconnection of networks became very necessary so Network on Chips (NoC) was implemented and many topologies were also suggested. The main aim of this paper is to find an effective on-chip topology for network on chips. **Methods/Statistical Analysis:** Topologies related to this network on chips gives different routing algorithms which defines the direction of the route. Different algorithms were proposed for different topologies defining the direction, traffic rate at which the packet is entering, latency, and throughput. These were considered as the main trade off's for Network on Chip. **Findings:** in this paper discussion on 4 topologies was done by taking few main properties into account and implementation of the topologies were done in software. It was found that latency and throughput plays a major role in the topology and along with these 2 the packet rate and flit rate also play a key role. The values of all trade-offs for every topology were taken and graphs were plotted showing the variation between the topologies. The efficiency of each topology can be observed in respective graph. **Application/Improvements:** Network on chips is used in routers and these can be further improved in to wireless connections.

Keywords: Interconnection Networks, Latency, Routing algorithm Throughput, Topology

1. Introduction

The new way to make the interconnection in a System on chip[1] internally is Network on Chip. In the past the inter connection were done using a bus architecture. Bus architecture is narrow and it might even block the traffic in some worst cases. In NOC, bus is replaced by a network. Communication is done in segments between two blocks by sending data within packets over a network. Network on chip is just like a computer network where devices and routers are connected with wires internally. Two important things in a network design are routing algorithm and topology. Routing algorithm differs with different requirements. As there are several different requirements for different algorithms, several algorithms were been designed and many features were also included in them. Every network on chip has got its own requirement like minimum latency, guaranteed throughput, and path diversity[2] and so on. Network on chip architecture enables integration of many computational and storages blocks on chip. Network on chips has overcome the disadvantages with system on chips. They are scalable and has high throughput[3] these two features are significant factors that affect net delay of the system. We are implementing four topologies regarding NOC routing algorithms[4] the main aim of this paper is to optimize latency and achieve high throughput. In[5] described the topologies of different routing algorithms in Network on Chips. In[6] detailed cycle-accurate simulator for Network-on-Chips (NoCs) that can also be used to model interconnection

** Author for correspondence*

networks for a variety of other systems. In[7] proposed a Mesh topology which is said to be a popular topology. In this topology the links in the network have similar lengths which make the physical design simple and the area grows linear to the number of nodes. In this topology the size can be measures in rows and columns. Torus is another network topology in interconnecting network which connects the processing nodes in parallel computer systems. The only difference in mesh topology and torus topology is, in mesh the connection between two nodes is made in a linear way and in torus the connection is done in a closed loop.

Evolution of today's technology and increase in pin-bandwidth made people to use high radix routers which are helpful in reducing the latency and throughput of the network. These high radix networks use longer cables to that of low radix ones. The more the cables are the more the network is confusing so usage of lesser cables increases the efficiency of the cables. In[8] introduce the dragonfly topology which is made with a group of high radix routers to increase the efficiency of the radix of the network. Increase in the pin bandwidth of the circuit made an increase in degree of interconnection networks. In introduced the flattened butterfly, a cost-efficient topology for high-radix networks[9].

2. Description

2.1 Dragon Fly

There are two main things in this dragon fly network topology; they are high-radix networks and low-radix network[8]. The diameter of the network is reduced for high-radix networks but there is a need of longer cables to construct the network. Where as in low-radix network, the diameter is more and the cables are short as we can clearly see in Figure 1.

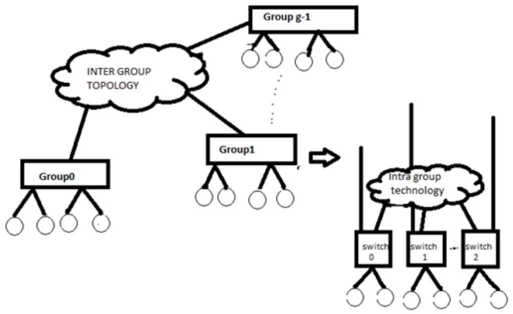

Figure 1. Dragon fly topology.

In this topology there is a network which consists of routers. The routers are again segregated in to individual ones and groups. There are two types of cables like global cables and local cables. Global cables are used to connect routers of different groups and local ones are used to connect routers inside a group. Using two different cables and segregation of routers give a better diversity in path.

2.2 Flattened Butterfly

This flattened butterfly as shown in Figure 2 also has high-radix and low-radix networks where high-radix[9]ones give a better diversity in path than a traditional and old butterfly model and is relatively of low cost also. This network has a less non-linearity and is a little confused comparatively. It is more scalable, but is even dangerous as it exploits the high radix routers in some worst cases.

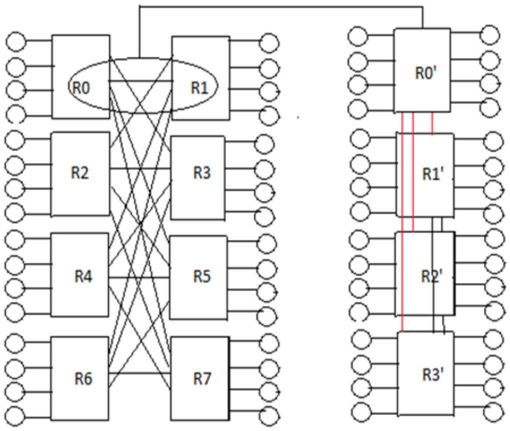

Figure 2. Flattened butterfly.

2.3 Torus Topology

Torus network can be considered as an improved version of mesh. In torus the heads of the column are connected to the tails and same in rows rather than in mesh. This network has efficient path diversity than mesh with minimum number of routers[10]. Torus network is an alternative topology for all large-scale supercomputers because it is rather a cost efficient network than others. There are software's to design a torus network which keeps all the constraints such as reliability and so on into consideration where balance in low latency and high throughput is mainly focused to get good structural properties in network diameter and path diversity[11]. Below Figure 3 depicts a torus network.

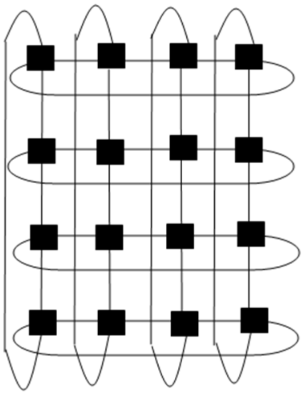

Figure 3. Torus topology.

2.4 Mesh Topology

There are mainly two classifications of routing algorithms they are adaptive routing algorithm and deterministic routing algorithm. In deterministic routing algorithms, the connection or the path from the source to destination is determined and fed prior in the header of the packet. In adaptive or distributive routing algorithms, the header consists only the address of the destination the path is determined automatically by the routers participation while going towards the destination.

There is an advantage in mesh topology[12] that is it has got its own way of routing called source routing. This feature in this topology gives an efficient encoding of path information with only a small number of bits. It is clear that every hop is sufficiently encoded with only two bits. As the packet which is entering the router is fed with a pre-defined decision about the destination port, designing of router is made significantly simple. Since the header is made with only few bits, the design is simple as well as the network not dependent with the size.

Following Figure 4 is an example of typical 16-node mesh. The last element of first row is connected with the first element of the second row.

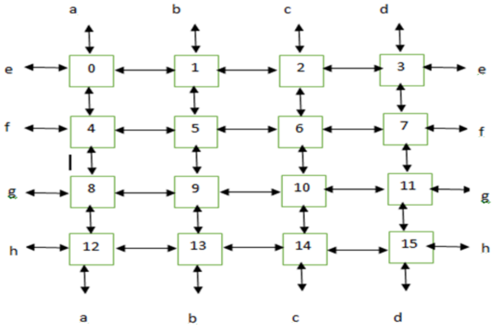

Figure 4. Mesh topology.

3. Results and Discussion

Latency: Latency of a network can be said as the time taken for a packet head to enter the input port and the packet tail to come out of the output port.

$$T = T_h + T_S$$

Where T_h = Head latency.
T_S = Serialization latency
L = Length of the packets.
b = Bandwidth of the channel.

Throughput: Throughput of a network can be defined as the rate of data in bits per second which enters the input port.
Ideal throughput can be given as:

$$\theta_{ideal} = \frac{b}{\gamma_{\max}}$$

Where b = Bits per second
$\gamma_{\max}$ Is unit less

Hop Count: It can be described as number of devices or routers the data may pass while communication.

3.1 Dragon Fly Evaluation

The maximum radix of a dragon fly network is
k = p + a + h-1

Where k = Radix of the router
p is number of terminals connected to each router.
a is number of routers in each group.
h is number of channels within each router used to connect to other group.
And maximum size of the network is:

$$N = ap(ah + 1)$$

Where N= Number of nodes.
H= Hop count.

3.2 Flattened Butterfly Evaluation

Flattened butterfly has a little difference than torus that is k is the Radix but n is the flat i.e; the number of routers fixed in one column.

$$j = i + \left[m - \left(\left| \frac{i}{k^{d-1}} \right| \bmod k \right) \right] k^{d-1}$$

Here N is the size of the network.
d is the diameter
j is the router
m is the range from 0-k-1

We have calculated different latencies for all the four topologies by taking different k values (k=2,4,8). We also took the average hop count of all the four topologies and mentioned them in results[9].

3.3 Torus Evaluation

Torus has two main parameters they are
k = Network Radix
n = dimension of cube
k- ary n-cube is the main theme of torus topology

$$N = k^n, \left(k = \sqrt[n]{N}, n = \log_k^N \right)$$

Where N = number of nodes

3.4 Mesh Evaluation

Mesh also has k and n calibration and
The number of connections in a full mesh is

$\frac{n-1}{2}$ Where n is number of nodes.

3.5 Dragon Fly Topology

For dragon fly topology the hop count average is approximately zero compared with other three topologies as shown in Figure 5. The packet latency is very high that is 192.89 which is quite irrelevant comparatively. Dragon fly topology takes a longer time to compute even for a smaller value of radix that is k=2. The run time for dragon fly is 129.7.

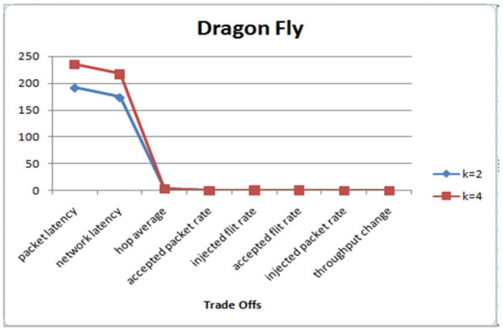

Figure 5. Result for dragonfly topology.

3.6 Flattened Butterfly Topology

For Flattened Butterfly the packet latency is not as big as dragon fly but is having a value of approximately 25. The run time for this topology is 17.5 which is shown in Figure 6.

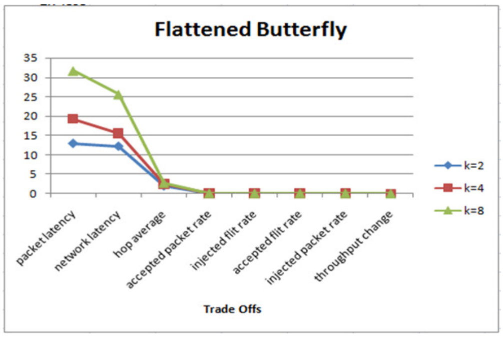

Figure 6. Results for flattened butterfly topology.

3.7 Torus Topology

The below Figure 7 shows the experimental graph for torus network where it is clear that when the radix (k) of the network is increasing the packet latency and network latency also increases. Besides hop average the throughput change is very less. Here the total rum time for one cycle is approximately 5. As we can see the injected rate and the flit rate is zero while number of flit cycles are 355. The packet size is 1.

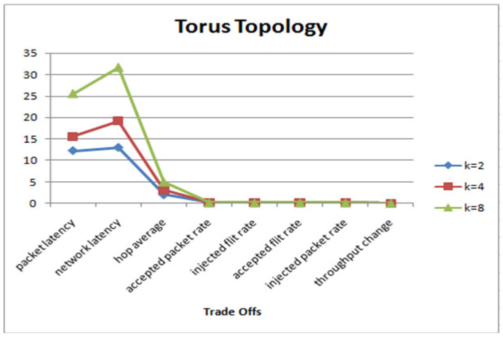

Figure 7. Results for torus topology.

3.8 Mesh Topology

While in mesh topology the packet latency and network latency is a bit high[13] or we can say that it is double when compared to torus. Other value like hop average is maintained same as in torus. The run time for mesh is approximately 4. Here in mesh topology the size of packet is 19 which are quite more compared to torus shown in Figure 8.

Table 1. Comparison of trade-off for different topologies

Network Topology	Torus Topology			Mesh Topology			Dragonfly Topology		Flattened Butterfly Topology		
Radix	K=2	K=4	K=8	K=2	K=4	K=8	K=2	K=4	K=2	K=4	K=8
Packet latency	12.225	15.6014	25.5913	30.2826	38.454	62.828	191.89	235.105	12.953	19.212	31.674
Hop average	1.9754	3.0012	5.0011	2.1304	3.4749	6.0868	3.3037	3.693	2.0090	2.4949	2.7489
Throughput change	0.0019	0.0058	0.004674	0.03025	0.0275	0.0144	0.0017	0.0003	0.0011	0.0004	0.0027

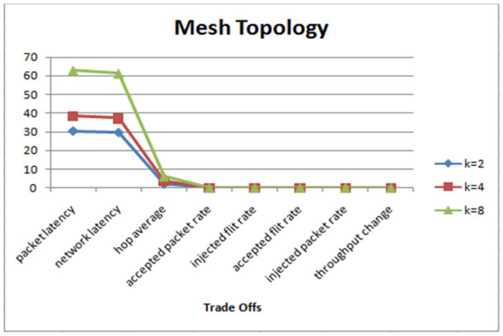

Figure 8. Results for mesh topology.

From Table 1, all the four topologies, finally we can say that torus is having better aspects like less latency and less run time. Experimentally it is clear that torus topology is better when compared with other topologies with latency of 31.67.

4. Conclusion

Network on Chip is a better advancement of system on chip technology. NoC (Network on Chip) is a latest implementation in networks and is still in process to improve than other technologies. So far there were only few implementations using Network on Chip interconnections. We carried out clear comparisons of four different topologies with different figures of trade-offs such as: packet latency, network latency, throughput change, hop average and so on under traffic conditions. We can conclude that Torus Topology is an efficient one that other three topologies as it is adaptive in nature and has a less latency with less time complexity.

5. References

1. Venkateswara Rao M, Rama Krishna TV, Yasaswini G. A frame work on AMBA bus based communication architecture to improve the real time computing performance in MPSoC. International Journal of Computer Applications. 2014; 91(1):1-5.
2. Chang K, Deb S. Performance evaluation and design trade-offs for wireless network on-chip architechtures. ACM Journal on Emerging Technologies in Computing Systems. 2012; 8(3):1-23.
3. Selvaraj G, Kashwan KR. Reconfigurable adaptive routing buffer design for scalable power efficient network on chip. Indian Journal of Science and Technology. 2015 Jun; 8(12):1-9.
4. Rantala V, Lehtonen T, Plosila J. Network on chip routing algorithms. TUCS Technical Report. 2006; 1-38.
5. Jiang N, William J, Dally, Towles B. A detailed and flexible cyclic-accurate network on chip simulator. IEEE. 2013; 86-96.
6. Vaishali Ingle V, Gaikwad MA. Mesh topology of NoC architecture using source routing algorithm. IJEAT. 2013 Aug; 2(6):2249–8958.
7. Bhople SS, Gaikwad MA. Comparative study of different topologies for network-on-chip architecture. International Journal of Computer Applications. 2013; 1-3.
8. Kim J, Dally WJ, Scott S, Abts D. Technology-driven, highly-scalable dragonfly topology. ISCA. 2008 Jun; 36(3):77-88.
9. Kim J, William J, Dally, Abts D. Flattened butterfly: A cost-efficient topology for high-radix networks. ISCA. San Diego, Calfornia, USA. 2007 Jun; 35(2):126-37.
10. Mirza-Aghatabar M, Hessabi S, Koohi S, Pedram H. Network on chip routing algorithms. CA, USA. 2007; 19-26.
11. Vasilis F, Pavlidis Eby G. 3-D topologies for networks on chip. 2008; 58(1):32-45.
12. Potti BMK, Subramanyam MV, Satya Prasad K. Hybrid genetic optimization to mitigate starvation in wireless mesh networks. Indian Journal of Science and Technology. 2015 Sep; 8(23):1-10.
13. Ingle VV, Mahendra A, Gaikwad CZ. Review of mesh topology of NoC architecture using source routing algorithms. International Journal of Computer Applications. 2013.