World Applied Sciences Journal 16 (Special Issue on Recent Trends in VLSI Design): 01-08, 2012
ISSN 1818-4952

Four Squared-Layer Topology for Network-On-Chip

Reza Sabbaghi-Nadooshan, Mahsa Ghorbanian and Hossein Doroud

Department of Electrical Engineering,
Islamic Azad University, Central Tehran Branch, Tehran, Iran

Abstract: This paper proposes four squared-layer topology for Networks-on-Chips. Topology has a significant effect on the most important parameters of a network such as latency and power consumption. The proposed topology is introduced and compared with existing ones in regard of factors such as power and delay. Results of comparisons show that proposed topology perform better than mesh and spidergon topologies. Although the proposed topology imposes the cost near to that of the mesh topology, the proposed topology 1) provides a lower diameter for Network-on-Chip, 2) offers better performance under the uniform and hotspot traffic pattern.

Key words: Four squared-layer · Topology · NoCs · Performance evaluation · Power consumption

INTRODUCTION

The advances in the semiconductor technology and the shrinking feature size in the deep submicron era, have led to an impressive increase in the number of transistors available on a single chip. This huge number of transistors enables us to integrate complex system-on-chip (SoC) designs containing a large number of processing cores together with large amounts of embedded memory and high-bandwidth I/O on a single chip. On the other hand, the applications are becoming more and more complex. The large amount of computational resources together with complicated communication patterns due to complex applications requires higher degrees of support from communication resources. Besides, as technology advances, the delay and power consumption of global interconnect structures (due to a large number of drivers) will be a major bottleneck for SoC design.

Increasing demand for communication between the processors and memories of cores in Systems-on-Chip (SoCs) leads to use of network-on-chip as a way of connections in the core. In the recent years, Network-on-Chip (NoC) has been known as a proper solution in order to deal with constraints of electric buses in SoC design by utilizing communication networks.

Network-on-Chip, a new chip design paradigm concurrently proposed by much research groups [1], [2], [3], is expected to be an important architectural choice for future SoCs. The Network-on-Chip concept has recently become a widely discussed technique for handling the large on-chip communication requirements of complex System-on-Chip (SoC) designs. A traditional bus-based interconnection scheme does not scale well to very large SoCs because many Intellectual Property (IP) blocks must contend with each other to communicate over the shared bus. In contrast, an on-chip network uses the packet-switching paradigm to route information between IP blocks and it can be scaled up to achieve a very large total aggregate bandwidth within the chip.

Although the concepts of NoC are inspired from the traditional interconnection networks, they have some special properties, which are different from the traditional networks. Compared to traditional networks, power consumption is the first-order constraint in NoC design [4][5]. As a result, not only the designer should optimize the NoC for delay (for traditional networks), but also for power consumption.

The key contribution of NoC lies in that it provides a communication infrastructure for the resources, i.e., it separates the communication from computation phase. Resources are connected to each other via the communication infrastructure which is an on-chip interconnect network. The origin of topology in NoC system comes from the research field of parallel computing. However, not all the topologies in parallel computing are suitable for the NoC paradigm.

Corresponding Author: Reza Sabbaghi-Nadooshan, Department of Electrical Engineering,
Islamic Azad University, Central Tehran Branch Tehran, Iran. E-mail: r_sabbaghi@iauctb.ac.ir

The choice of network topology is an important issue in designing a NoC. Different NoC topologies can dramatically affect the network characteristics, such as average inter-IP distance, total wire length and communication flow distributions. These characteristics in turn, determine the power consumption and average packet latency of NoC architectures.

In general, the topologies proposed for NoCs can be classified into two major classes, namely regular tile-based and application-specific. Compared to regular tile-based topologies, application-specific topologies are customized to give a higher performance for a specific application. Moreover, if the sizes of the IP cores of a NoC vary significantly, regular tile-based topologies may impose a high area overhead. On the other hand, this area overhead can be compensated by some advantages of regular tile-based architectures. Regular NoC architectures provide the standard structured interconnects, which ensures well-controlled electrical parameters. In addition, usual physical design problems like crosstalk, timing closure and wire routing and architectural problems such as routing and switching strategies and network protocols can be designed and optimized for a regular NoC and be reused in several SoCs.

The mesh topology is the most popular topology used in today's regular tile-based NoCs. It is well known that the mesh topology is regular, has low cost and consumes low power. Despite such favorable advantages for on-chip implementation, some packets may suffer from long latencies due to lack of short paths between distant nodes in a mesh NoC. A number of previous works tried to tackle this shortcoming, by adding some application-specific links between distant nodes in the mesh [6], bypassing some intermediate nodes by inserting express channels [7], or introducing new topologies with lower diameters [8, 9].

In this paper, we propose a topology called Four squared-layer (FSL for short). Simulation has shown that our topology can achieve better performance than the mesh and is more suitable for NoC with the increase number of nodes.

The Proposed Topology: Considering the comparison between topologies we have defined a new topology called FSL topology. This topology almost inspired from spidergon topology, but its nodes distribute on four squared-layer instead of circles. In this section we first describe spidergon topology and then define FSL topology.

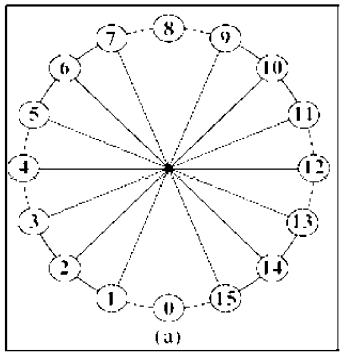

Fig. 1: The spidergon network with 16 nodes [6].

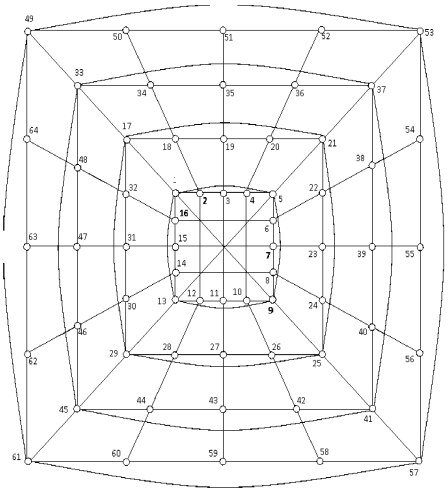

Fig. 2: The FSL network with 64 nodes

Spidergon Topology: The spidergon NoC is a network which is proposed by ST Microelectronics [10]. Fig. 1 is an instance of spidergon topology with 16 nodes. The small circle with numbers stands for a node; every node is composed of a processing element (PE) and a router. The processing element can be memory, IP and so on; the router is used for routing packets from the local processing element to other routers. Each node connects to other nodes in three directions: clockwise, anticlockwise and cross-link. For example, the node 0 connects to node 1 in clockwise, to node 15 in anticlockwise and to node 8 in cross-link. The routing algorithm in the spidergon is quite simple [10].

Structure of FSL Topology: Considering four squares adjacent together, which are connected through 16 nodes and each node of each four layers connected in each single direction also they are connected diagonally. The new topology has 64 nodes and the Fig. 2 represents the FSL topology with 64 nodes.

Table 1: The networks parameters

Topology	Nodes	Diameter	Degree	cost	Average
distanceFSL	2^n	n+1	n	n(n+1)	3.49
Spidergon	2^n	2n+4	n/2	n(n+2)	4.19
mesh	2^n	$2(2^{n/2}-1)$	n-2	$2(2^{n/2}-1)$ (n-2)	3.64

The degree of corner nodes in the inner square is 6. However, for corner nodes are 5. Furthermore, parallel side inner square node has the degree of 4. However, the outer square node has the degree 3. The numbers of edges which are connected together are called degree of each node and the largest degree indicates the degree of network. Therefore, the degree of FSL topology is 6.

Network Parameters: Although high throughput and low latency are still desirable characteristics, other factors such as power consumption constraints and implementation feasibility have to be considered with equal, if not more, efforts. During the past years, a number of different NoC topologies have been proposed such as mesh, torus, spidergon, spin, ring, octagon, fat-tree, butter-fat-tree and other irregular topologies. The network cost parameter for measuring the implementation feasibility of topologies is introduced in [11] and [12]. Degree refers to the number of channels entering and leaving each node in the network. If the degrees of nodes in a network are not identical, the network degree is determined by the maximum node degree. The diameter is the largest, minimal hop count over all pairs of terminal nodes in the network. Thus, in designing a topology, there is always a trade-off between the degree, which relates to the hardware cost and diameter, which relates to the message transmission time.

$$Network\ Cost = Degree * Diameter$$

The networks are compared on Table 1.

The Routing Algorithm: Routing of proposed FSL topology is very simple and each node is connected from each single side to other nodes. The corner nodes in inner squares are connected to corner of same square. Furthermore, we have routes in the front nodes, based on inner layer. For example, routing of inner corner squares which are connected to outer can be written as follows:

$$1 \rightarrow 17 \rightarrow 33 \rightarrow 49$$

Note that we have some routed to reach from node one to the destination node 49. However, we choose the shortest route to reach destination node.

```
Routing  Algorithm
If( j==1)
{
For (int  i=0,i = 16 ,i ++   &   for corner nodes i+4)
& for { i==1 , i=16   i+8
        i==2 ,i=16   i+10
        i==3,i=16   i+8
        i==4,i=16   i+6
        i ==5,i=16   i+7
        i==6 , i=16  i+10
         i==7 ,i=16   i+8
        i==8,i=16    i+9
else
If(j= =2)
{
For(int  i=17 , i=32  i++  & i±16 & for corner nodes i+4)
else
If(j==3)
{
For (int i=33 ,i=48  i++ & i±16 & for corner nodes i+4)
else
if(j==4)
{
For(int i=49 , i=64,i++ &i-16 &for corner nodes i+4)
```

Fig. 3: FSL pseudo code

The paths based on rings can be written as following:

$1 \rightarrow 2$
$1 \rightarrow 2 \rightarrow 3$
$1 \rightarrow 5 \rightarrow 5$
$1 \rightarrow 5 \rightarrow$
$1 \rightarrow 5 \rightarrow 6$
$1 \rightarrow 9 \rightarrow 8 \rightarrow 7$

Routing Pseudo code is as Fig. 3. In the figure j is the number of layer and i is the number of node (we supposed node number is started with 0 in Pseudo code).

Zone-Order Label-Based Routing: There is no unique routing algorithm for this topology and we use Zone-Order label-based routing algorithm that is used for regular and irregular networks. Furthermore, this algorithm was applied to both mesh and spidergon topologies so all topologies are simulated here used the same routing algorithm [13]. This method is based on spanning tree as Fig. 4 and every node can be chosen as root of the tree.

There are three steps that we must indicate to use the algorithm:

- Label cores correctly.
- Specify zones that the links do not have rotational dependence so the zones are dead lock free.
- The determination of sequences forming is in a constant way.

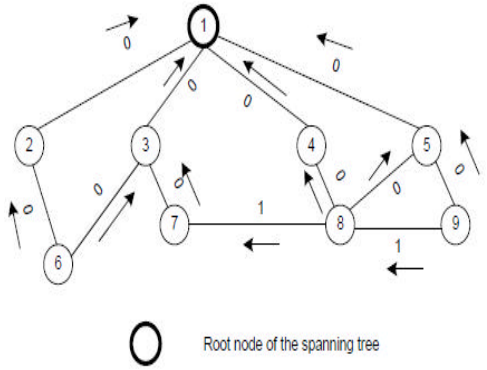

Fig. 4: Label links by spanning tree.

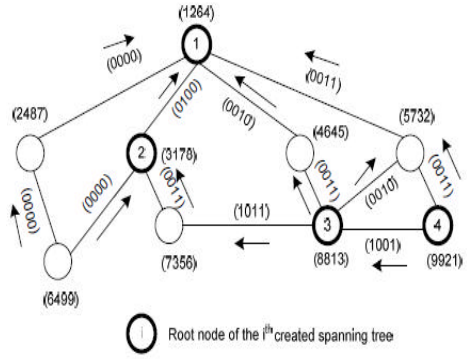

Fig. 5: Label nodes and links by several spanning tree.

To transmit a packet of data between two cores, packet transfer at first in its own zone and then transmit to continue in other zones but moving from upper zone to lower zone is forbidden. For example, X, Y and Z are three zones. X has the first priority, Y and Z have the second and third priority. Packet can't move in X or Y zones after moving in Z zone. As a result the algorithm is deadlock free.

In this algorithm at first we produce spanning tree with BFS method from a noun-uniform graph and the node label ascending. The labels are natural numbers and start from 1. In fact, the first root label is 1 and the final node labels' is 2MN+2. The link's label is 1 if the orient of a link shows the tail's label is smaller than the head's label. On the other hand, if the tail's label is bigger than the head's label, the link's label is 0.

Label by use of several spanning tree lead to links with label that have several bits and each bit is obtained by one of those spanning trees (Fig. 5). If we have P spanning tree, labeled with P bit, the first bit is related to the first spanning tree and the second bit of each link label is related to the second spanning tree and so on.

In the second step, zones are determined and none of the links that exist in a zone have rotational dependence. If all link labels that are in a zone have at least one equal bit, then the zone is deadlock free. For this purpose, at least one same equal bit is enough to define a zone.

At the end of procedure, the sequence between zones must be determined to fulfill continuity limit. Each sequence that adopts this limit can be defined as a deadlock-free routing algorithm. There is an algorithm due to sequences and the domain is deadlock free if and only if all the time 0 comes before 1.

By increasing the number of spanning trees, the usable route increases with 50 percent probability, but as a result complexity of routing algorithm also increases [13].

RESULTS

simulator: To evaluate the performance of FSL architecture, we develop a discrete event simulator operating at the flit level using xmulator [14]. The irregular xmulator is based on events. In general, we have several events and a simulator engine that carries out simulation. The engine holds a queue of events. The sequence of event never finishes because an event is the source of other events and produces them.

The event's queue must be arranged by its time and it is vital for simulator performance. In this simulator, events save a black and red tree and every tree's branch has its own queue and time of the events is equal, in fact events with the same time are in one queue. The name of this simulator consists of XML and Simulator and it is working base on events. In general, we have several events and a simulator engine that carries out simulation. The engine holds a queue of events. Engine works simply. Its internal structure is a While loop. All events are analyzed sequencely by engine and the events that are at head of a queue have priority in scale of events that are at the tail. The queues in the engine never finish at normal situation because of any event is source of other event production. The engine with more simple structure is more powerful and serpentine as view of modularity. In fact that is not any need that engine know the kind of events or their structure, the engine just must know an event is belong to which component.

Queues of events must are arranged by their time. The management of the queues has effective influence on simulator performance. All events are stored in a

Red-Black tree at this simulator and any part of the tree consists of queue of events that their happening time is the same exactly. For example every event that are happened at time 50 are operated parallel at real world so their operating sequence at simulator does not any effect on the final result so the events with the same happening time are stored at a queue.

Simulation Results: We set the networks link width to 128 bits. Each link has the same bandwidth and one flit transmission is allowed on a link. The power is calculated based on a NoC with 90 nm technology whose routers operate at 2.5 GHz. Based on the core size information presented in [15], we set the width of the IP cores to 1 mm and the length of each wire is set based on the number of cores it passes. The simulation results are obtained for 8Í8 mesh NoCs with XY routing algorithm, spidergon and FSL NoCs using the routing algorithms described in the previous section. The message length is assumed to be 16, 32 and 64 flits and one virtual channel per physical channel is used. Messages are generated according to a Poisson distribution with rate 8. The traffic pattern can be *Uniform* and *Hotspot* [16].

With introducing this simulator and checking the advantages of this simulator, the delay and power figures are the following. The x axis of these figures indicates the generation rate and y axis indicates power and delay in our simulations.

In Fig. 6, the average message latency is plotted as a function of message generation rate at each node for the mesh and FSL networks of 64 nodes using deterministic routing for different message sizes. As can be seen in the Fig. 6, the FSL has smaller average message latency with respect to the equivalent mesh network. The reason is that the average inter-node distance of the FSL network is lower than the equivalent mesh network. Fig. 7 compares the total network power in topology mesh and FSL of 64 nodes with different message lengths of 16, 32 and 64.

Fig. 8 shows the average message latency in the 8×8 simple mesh and FSL for different traffic patterns with message length of 16. As can be seen in the Fig. 8, the FSL NoC achieves a reduction in message latency with respect to the simple 2D mesh network for the full range of network load under various traffic patterns (especially in uniform traffic). For hotspot traffic load a hotspot rate of 14% is assumed (i.e. each node sends 14% of messages to the hotspot node (node 36) and the rest of messages to other nodes uniformly). Note that increasing the network size causes earlier saturation in a simple 2D mesh. Fig. 9 demonstrates power consumption of the simple 2D mesh and FSL under deterministic routing scheme with various traffic patterns. It is again the FSL that shows a better behavior before reaching to the saturation point.

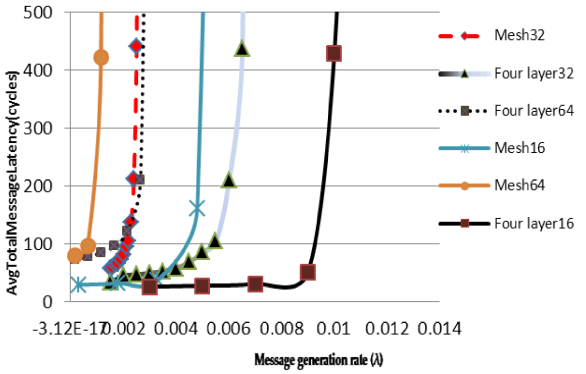

Fig. 6: The average message latency in the 8×8 simple mesh and FSL for different message lengths.

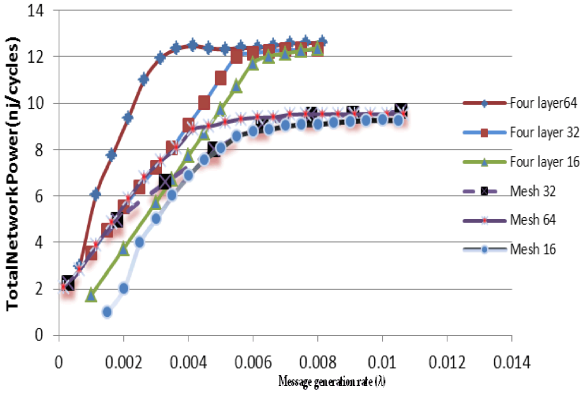

Fig. 7: Total power in the 8×8 simple mesh and f FSL for different message lengths.

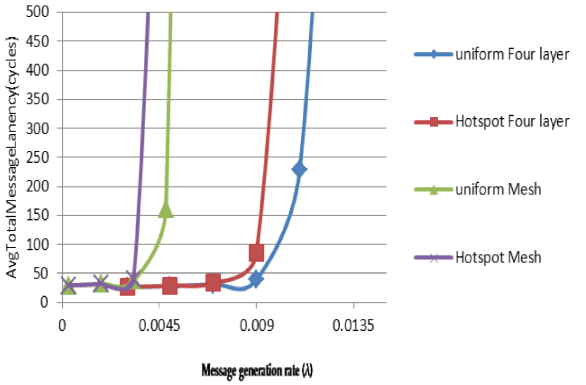

Fig. 8: The average message latency in the 8×8 simple mesh and FSL for different traffics patterns with message length of 16.

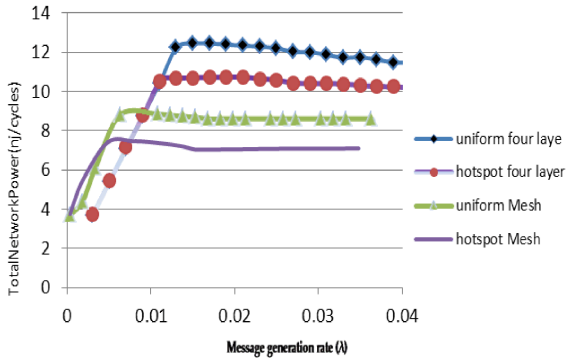

Fig. 9: Totat network power in the 8×8 simple mesh and FSL for different traffics patterns with message length of 16.

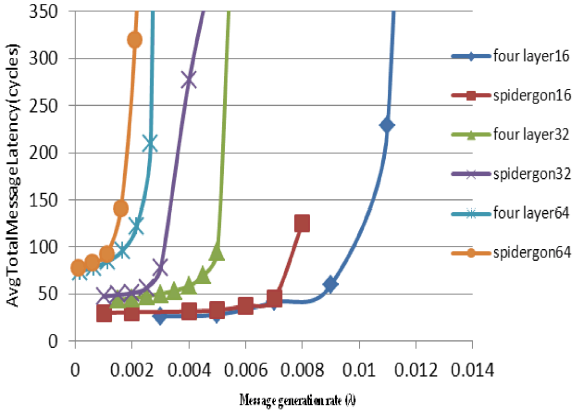

Fig. 10: The average message latency in the 64 nodes spidergon and FSL

Fig. 10 compares the average message latency in spidergon and FSL of 64 nodes with different message lengths of 16, 32 and 64. As can be seen, the FSL has smaller average message latency with respect to the equivalent spidergon network.

In Fig. 11, the average message latency is plotted as a function of message generation rate at each node for the mesh and FSL networks of 64 and 128 nodes using deterministic routing for different message sizes. As can be seen in the Fig. 11, the FSL with 64 nodes has smaller average message latency with respect to the equivalent FSL with 128 nodes network. Fig. 12 compares the total network power in topology FSL of 64 and 128 nodes with different message lengths of 16, 32 and 64.

Fig. 13 shows the average message latency in FSL of 64 and 128 nodes for different traffic patterns with message length of 16. As can be seen in the Fig. 14, for hotspot traffic load a hotspot rate of 14% is assumed.

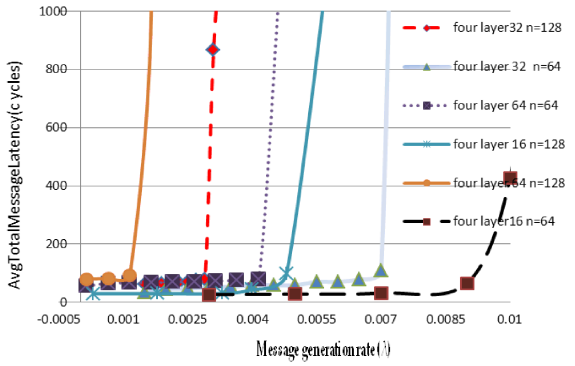

Fig. 11: The average message latency in the FSL with 64 and 128 nodes for different message lengths.

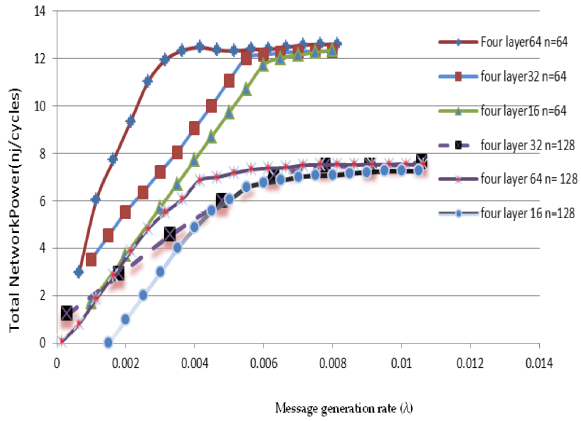

Fig. 12: Total power in the FSL with 64 and 128 nodes for different message lengths.

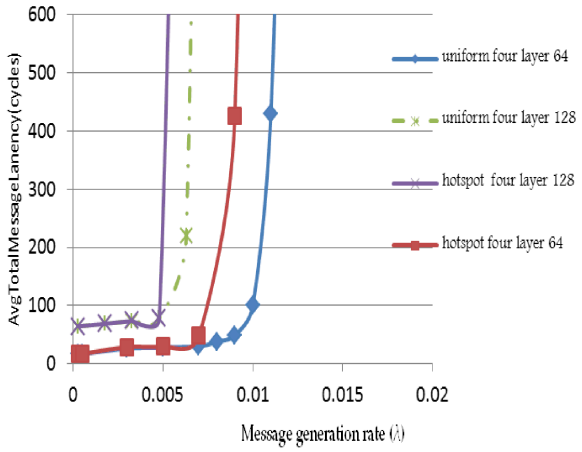

Fig. 13: The average message latency in the FSL with 64 and 128 nodes for different traffics patterns with message length of 16.

Fig. 14 demonstrates power consumption of FSL of 64 and 128 nodes under deterministic routing scheme with various traffic patterns.

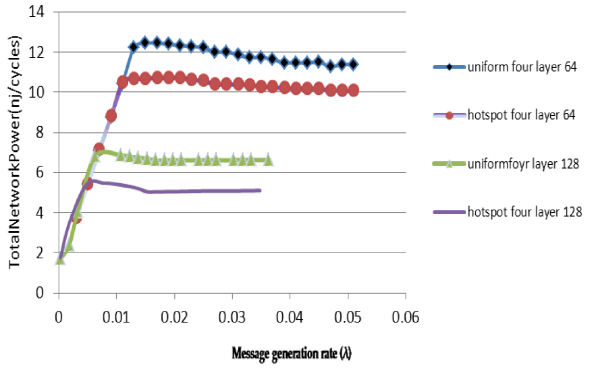

Fig. 14: Totat network power in the FSL with 64 and 128 nodes for different traffics patterns with message length of 16.

The obtained result of xmulator indicates the FSL topology relative to mesh topology and spidergon goes to the saturation later and can send more packages. Therefore has more power also the average distance length of the FSL topology is less than mesh and spidergon topologies.

The results indicate that the power of FSL with 64 nodes network is less for light to medium traffic loads. The main source of this reduction is the long wires which bypass some nodes and hence, save the power which is consumed in intermediate routers in an equivalent mesh and spidergon topologies.

Although for low traffic loads the FSL network provides a better power consumption compared to the simple 2D mesh network, it begins to behave differently near heavy traffic regions. It is notable that a usual advice on using any networked system is *not to take the network working near the saturation region*. Having considered this and also the fact that most of the networks rarely enter such traffic regions, we can conclude that the FSL with 64 nodes network can outperform its equivalent mesh and spidergon networks when power consumption is considered.

CONCLUSION

This paper proposes four squares-layer topology applicable for NoCs. The proposed topology is introduced and discussed in the paper. Simulation experiments were conducted to assess the network latency and power consumption of the proposed topology. Results showed that the proposed networks improved the performance of the NoC in comparison with the equivalent-sized mesh and spidergon NoC(64 and 128 nodes) under light and moderate traffic loads.

In future work, we are trying to increase the size and number of node square topology in order to simulate the power, delay and other specific parameter for topologies. Furthermore, we will try to propose 3D layout and improvement of routing for four squared-layer topology.

REFERENCES

1. Sgroi, M., *et al.,* 2001. Addressing the System-on-a-Chip Interconnect Woes Through Communication-based Design, 38th Design Automation Conference, June, 2001.
2. Luca Benini and Giovanni De Micheli, 2002. Network on Chips: A new SoC Paradigm, IEEE computer, Jan., 2002.
3. Shashi Kumar, *et al.*, 2002. A Network on Chip Architecture and Design Methodology, IEEE Computer Society Annual Symposium on VLSI, Pittsburgh,Pennsylvania, USA, April 2002.
4. Pande, P.P., C. Grecu, M. Jones, A. Ivanov and R. Saleh, 2005. Performance Evaluation and Design Trade-Offs for Network-on-Chip Interconnection Architectures, IEEE Transactions on Computers, 54: 8, August 2005.
5. Hu, J. and R. Marculescu, 2005. Energy-performance aware mapping for regular NoC architectures, IEEE TCAD.
6. Ogras, U.Y. and R. Marculescu, 2005. Application-specific network-on-chip architecture customization via long-range link insertion, In: ICCAD05: IEEE/ACM Intl. Conf. on Computer Aided Design, pp: 246-253.
7. Dally, W.J., 1991. Express cubes: improving the performance of K-ary N-cube interconnection networks", IEEE Trans Comput, 40(9): 1016-1023.
8. Kim, J., J. Balfour and W.J. Dally, 2007. Flattened butterfly topology for on-chip-networks, In: Micro07: 40th IEEE/ACM international symposium on microarchitecture, pp: 172-182.
9. Grot, B., J. Hestness, S.W. Keckler and O. Mutlu, 2009. Express cube topologies for on-chip interconnects, In: HPCA09: 15th IEEE international symposium on high performance computer architectures, pp: 163-174.
10. Coppola, M., *et al*., 2004. Spidergon: a novel on chip communication network, proc. Int'l Symposium on System on Chip 2004, Tampere, Finland, Nov. 2004.
11. Efe, K., 1991. A variation on the hypercube with lower diameter, IEEE Transaction of Computers, 40(11): 1312-1316. Nov. 1991.

12. Dally, W.J. and B. Towles, Eds., Principles and Practices of Interconnection Networks. Morgan Kaufmann Publishers, 2004.
13. Moraveji, R., P. Moinzadeh, H. Sarbazi-Azad and A. Y. Zomaya, 2011. Multispanning Tree Zone-Ordered Label-Based Routing Algorithms for Irregular Networks, IEEE Transactions on Parallel Distributed Systems, 22(5): 817-832.
14. Nayebi, A., S. Meraji, A. Shamaei and H. Sarbazi-Azad, 2007. Xmulator: A listener-Based Integrated Simulation Platform for Interconnection Networks, Proc. of Asian Int. Con. on Mod. Sim., pp: 128-132.
15. Mullins, R., A. West and S. Moore, 2006. The Design and Implementation of a Low-Latency On-Chip Network, Asia and South Pacific Design Automation Conference, pp: 164-169.
16. Duato, J., S. Yalamanchili and L.M. Ni, 2003. Interconnection Networks", Morgan Kaufman.