A New Irregular Fault-Tolerant Routing Algorithm in Network-on-Chip

Davood Abednezhad†, Seyed Enayatallah Alavi††

†Department of Computer Engineering Ahvaz Branch, Islamic Azad University Ahvaz, Ahvaz, Iran
††Shahid Chamran university, Ahvaz, Iran

Summary
Many Network-on-Chip (NoC) routing algorithms have been introduced to increase performance of the system and decrease latency. Furthermore, some algorithms work under faulty conditions and some of them can tolerate in presence of oversized nodes. In this paper, we propose a deadlock-free routing algorithm, Fault Tolerant enhanced odd-even XY (FT-OED-XY), in which tolerate under faulty conditions and also can pass oversized nodes simultaneously in an irregular mesh interconnection network. In the absence of faulty nodes/links and oversized nodes, FT-OED-XY uses regular XY routing algorithm and if a packet encounters a faulty link or node, the algorithm acts as enhanced odd-even mechanism try to pass the faulty region by dividing columns into two classes; odd and even. Simulations have been done to illustrate the performance of proposed routing algorithm. Moreover, we compared FT-OED-XY with fault tolerant FT-DyXYZ and Odd-Even algorithms, in terms of average packet latency and throughput. Results show higher rates of throughput and lower average message latency in FT-OED-XY algorithm.

Key words:
3D NoC, System-on-Chip, irregular, Fault tolerant routing, Congestion, Enhanced odd-even

1. Introduction

With shrinking geometries, global interconnects are becoming the principal performance bottleneck for high-performance Systems-on-Chip (SoCs) [1] [2] [3]. 3D ICs, which contain multiple layers of active devices, have the potential for enhancing system performance [4] [5] [6] [7].
Network-on-Chips (NoCs), as scalable and efficient communication structures for complex SoCs [3] [1], together with short interconnects in 3D ICs have led to appearance of 3D NoCs for future complex and high performance SoCs. Thus, utilizing some methods which introduce fault tolerance is essential for system reliability requirements. In NoC-based SoCs, fault tolerant routings are promising solution to increase reliability of communications between devices on the chip against faults that occur in manufacturing phase or system runtime [8].
In this paper, we have been proposed an irregular fault-tolerant routing algorithm for 3D mesh based Network-on-Chip. First, we overview basic routing algorithm and then in the next section two fault tolerant algorithm called FT-DyXYZ and odd-even introduced. Then, compare with the proposed algorithm followed by conclusion.

2. Related Work

2.1 XY routing algorithm

XY routing is a dimension order routing which routes packets first in x- or horizontal direction to the correct column and then in y- or vertical direction to the receiver. XY routing suits well on a network using mesh or torus topology. Addresses of the routers are their xy-coordinates. XY routing never runs into deadlock or livelock [9].
There are some problems in the traditional XY routing. The traffic does not extend regularly over the whole network because the algorithm causes the biggest load in the middle of the network. There is a need for algorithms which equalize the traffic load over the whole network [10].

2.2 DyXY routing algorithm

The With the DyXY routing algorithm, each packet only travels along a shortest path between the source and the destination (this guarantees the deadlock-free feature of the routing algorithm) if there are multiple shortest path available, the routers will help the packet to choose one of them based on the congestion condition of the network [11].
The detailed routing algorithm can be summarized as follows:
* Read the destination of an incoming packet.
* Compare addresses of the destination and the current router.
- If the destination is the local core of the current router, send the packet to the local core;
- Else
* If the destination has the same x (or y) address as the current router, send the packet to the neighboring router on the y-axis (or x-axis) towards the destination;

Manuscript received April 5, 2017
Manuscript revised April 20, 2017

* Else, check the stress values of current router's neighbors towards the destination, and send the packet to the neighbor with the smallest stress value [11].
The stress value is a parameter representing the congestion condition of a router. Here, DyXY use the 'instant queue length' of each router (i.e., the number of occupied cells in all input buffers) as the stress value, since it achieves the best results among all kinds of average or flow-control types we have attempted. Each router stores instant stress values for all neighbors, and each stress value is updated based on an event-driven mechanism [11].

2.3 FT-DyXYZ routing algorithm

In the absence of faulty links in NoC, packets are routed using DyXYZ that is a minimal and congestion aware routing. But when there are some faulty links in the network, we use fault tolerant routing, FT-DyXYZ [8].
If there are some permanent faulty links (Through Silicon Vias based vertical links or horizontal links), after determination of preferred output ports to route packet closer to the destination, faulty links are removed from preferred link set and among the rest, one output port is chosen based on congestion parameter [8].
But, if after removing of faulty links, there is no output port left to route packet on minimal path, misrouting should be done. In this case, to tolerate fault and route packet to its destination, utilizing non minimal paths is unavoidable [8].
In the case of misrouting, one intermediate node is randomly chosen among the nodes in the current layer or neighbor layers (up or down layers) [8]. Then routing will be continued in two phases: first, from current node to the intermediate node and then, from intermediate node to destination node of packet. In each phase, the minimal and normal FT-DyXYZ is used and removing faulty links and congestion parameter is considered in each hop of routing packets [8]. In each phase of misrouting, if the packet reaches to a router with no fault-free output port toward temporary or real destination, another intermediate node will be chosen randomly [8]. This new intermediate node will be saved in header flit and misrouting will be started again from that router. This scenario will be repeated when it is necessary, until delivering the packet to destination node [8].
Figure 1 shows an example of routing packet from the source node of A to destination of J [8].
The packet is routed based on normal FT-DyXYZ routing to nodes B, C and D. In node D, there is just one preferred output port (east) that is faulty, so misrouting should be done [8]. We assume that node E is randomly determined as intermediate node, so in the first phase, the packet is routed based on normal FT-DyXYZ through D–F–G–E path and in the second phase, the packet reaches to destination through E–H–I–J path and in the node of J it is delivered to the local core [8].

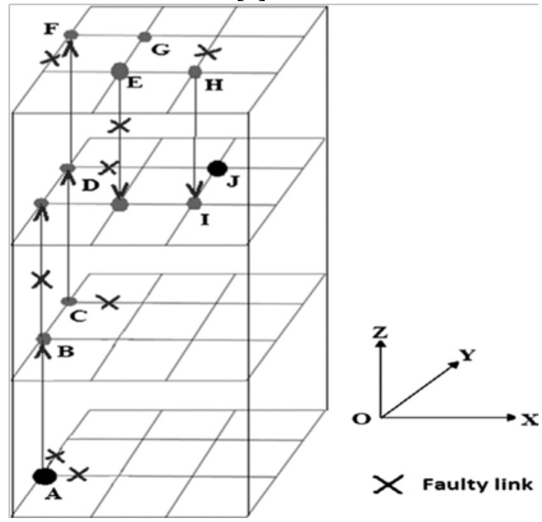

Figure 1: An example of routing packet using FT-DyXYZ algorithm

3. Proposed routing algorithm, Fault Tolerant Enhanced Odd-Even XY (FT-OED-XY)

In absence of faulty link in NoC, packets are routed using XY base that is deadlock and livelock free routing. It first send packet in vertical direction, when X value of packet location equal to X of destination node, algorithm change direction to horizontal .In faulty situation, algorithm use enhanced odd-even model to routed packet to its destination. Figure 2 shows FT-OED-XY algorithm and Figure 3 describe the algorithm enhanced odd-even.

```
routing_mode = Get_routing_mode();
If (routing_mode = normal) {
    Routing using XY algorithm with vertical move first;
}
Else {
    packet.temp_destination =
    find_Next_Node_base_on_Enhanced_odd_even_routing ();
    routing_mode = normal;
}
```

Figure 2: pseudo code of proposed routing algorithm

```
find_Next_Node_base_on_Enhanced_odd_even_routing (pck As
NetworkPacket) As Node{

    pDirect // Packet Direction
    Cur_x, Cur_y //packet Current location
    Dest_x, Dest_y //packet destination
    Second_Node ,tmp // Network Node
    dx = Dest_x - Cur_x ; dy = Dest_y - Cur_y;

    Select Case pDirect
      Case SouthToNorth
        If (Abs(dx) > 2 And Cur_x - 1 >= 0 ){
          Second_Node = NetworkNodes(Cur_x - 1, Cur_y)
        }
      Case NorthToSouth
        If (Abs(dx) > 2 And Cur_x + 1 <= NetworkHeight) {
          Second_Node = NetworkNodes(Cur_x + 1, Cur_y)
        }
      Case EastToWest
        If (Abs(dy) > 2 And Cur_y - 1 >= 0 ){
          Second_Node = NetworkNodes(Cur_x, Cur_y - 1)
        }
      Case WestToEast
        If (Abs(dy) > 2 And Cur_y + 1 <= NetworkWidth Then){
          Second_Node = NetworkNodes(Cur_x, Cur_y + 1)
        }
    End Select

    Return Second_Node

  }
```

Figure 3: Routing using enhanced odd-even algorithm

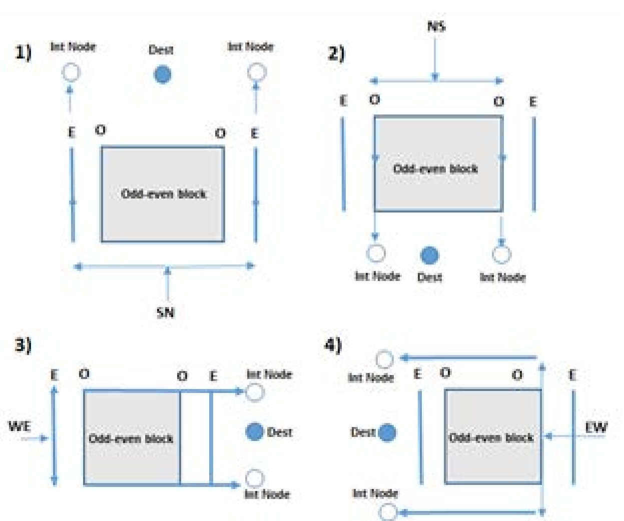

Figure 4: enhanced odd-even and intermediate node

In Figure 5 shown an example of proposed routing algorithm. Packet from source node with (2, 4) address pass through faulty region to (2, 0)

As shown in Figure 5, if packet direction was north to south (NS) or south to north before reach to the destination, algorithm uses odd-even right or left column to pass the faulty region and try to reach router with X address set to destination node X value, and Y value set to odd-even column. If packet direction is west to east, it uses top or bottom edge of the odd-even block to cross the faulty-link region and try to reach router with X address value set to top edge of odd-even block if current X address lower than half width of network, otherwise set to bottom edge and Y dimension set to destination Y address. If packet direction was east to west, when current X address lower than half width of network, it use above row of top edge of odd-even block, otherwise bottom row of bottom edge of odd-even block to cross the faulty-region and Y value set to destination Y address.

In faulty conditions FT-DyXYZ uses a random intermediate node mechanism to pass packet through the faulty-region and this solution cause to increase the distance to destination and also delay increment. Odd-Even algorithms pass the packet trough the faulty-region so they are crowded around the fault zone. This leads to a massive delay of packets which have to pass through this region [8]. In contrast, FT-OED-XY routing algorithm try to pass packet to nearest (best) node to destination and divide the traffic around the faulty region specially based on routing direction of packet.

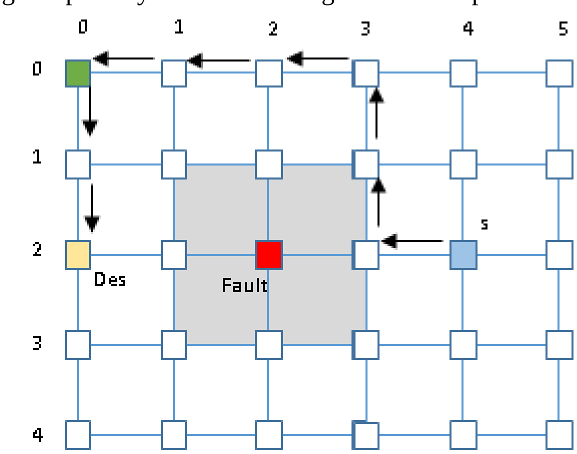

Figure 5: an example of proposed routing algorithm

4. Simulation result

To simulate under the simulate condition, we use 2 and 3 virtual channels, 8 x 8 mesh network and compare FT-EOD-XY algorithm with basic odd-even, 2D version of FT-DyXYZ . In our simulation, each packet consist 32 flits. We evaluated FT-EOD-XY algorithm with FT-DyXYZ and odd-even routing in terms of average packet latency. Figure 6 demonstrates average packet latency and Figure 7 shows average throughput of

FT-EOD-XY, FT-DyXYZ and odd-even routing for 8% fault in uniform traffic.

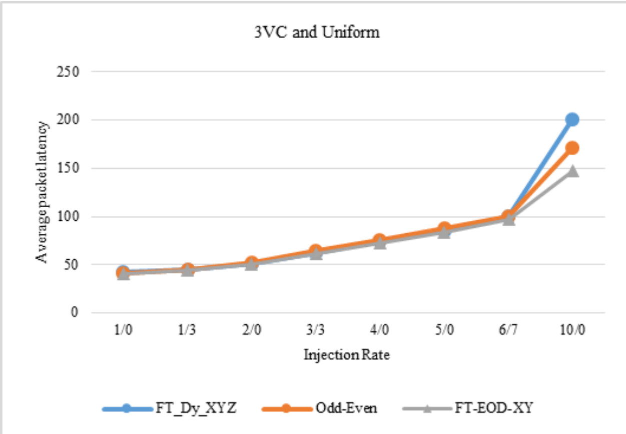

Figure 6: average packet latency of our algorithm, FT-DyXYZ and odd-even in uniform traffic

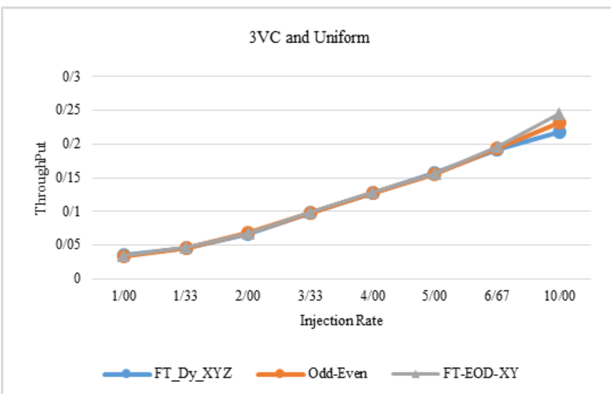

Figure 7: average Throughput of our algorithm, FT-DyXYZ and odd-even in uniform traffic

We used three kinds of traffics for evaluation, uniform, hotspot and transpose traffic.

- Uniform traffic

In this traffic, each node sends several messages to destination nodes in the network where a uniform distribution is used to adjust the destination set of each message [12].

In Figure 6, 7 the average packet latency and throughput as a function of injection rate has been showed. In this case, by use of enhanced Odd-Even technique, FT-EOD-XY has lower latencies because uniform traffic is balanced under XY routing [13] and higher throughput. Figure 6,7 demonstrated improvement of delay and throughput with increase of injection rate in FT-EOD-XY to the other algorithm in uniform traffic pattern and 8% fault of total link.

FT-DyXYZ needs two of three VCs to avoid deadlock [8], but enhanced Odd-Even method in our algorithm uses three VCs to improve performance rather than deadlock avoidance. So as simulation demonstrates, using FT-OED-XY, network saturates in high injection rates in contrast to FT-DyXYZ algorithm that is saturated in low injection rates.

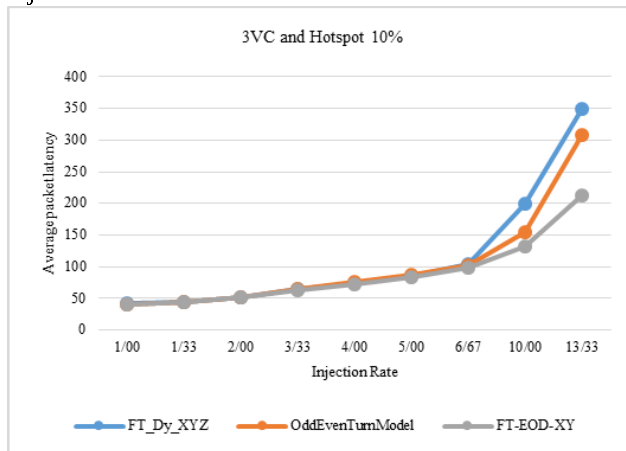

Figure 8: average packet latency of our algorithm, FT-DyXYZ and odd-even in hotspot 10% traffic

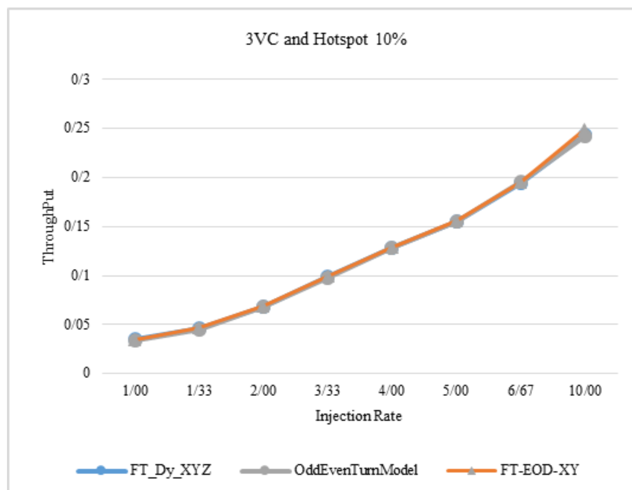

Figure 9: average throughput of our algorithm, FT-DyXYZ and odd-even in hotspot 10% traffic

- Hotspot traffic

In this traffic node one node receives 10% of packet. Figure 8 demonstrates average packet latency and Figure 9 shows average throughput of proposed algorithm, FT-DyXYZ and odd-even routing. Figure 8,9 demonstrated improvement of delay and throughput with increase of injection rate in FT-EOD-XY to the other algorithm in 10% hotspot traffic pattern and 8% fault of total link.

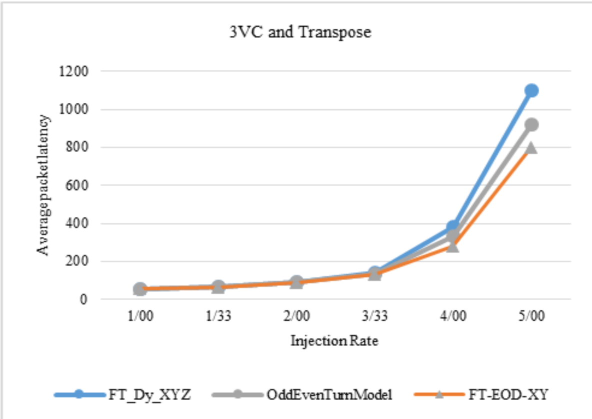

Figure 10: average packet latency of our algorithm, FT-DyXYZ and odd-even in transpose traffic

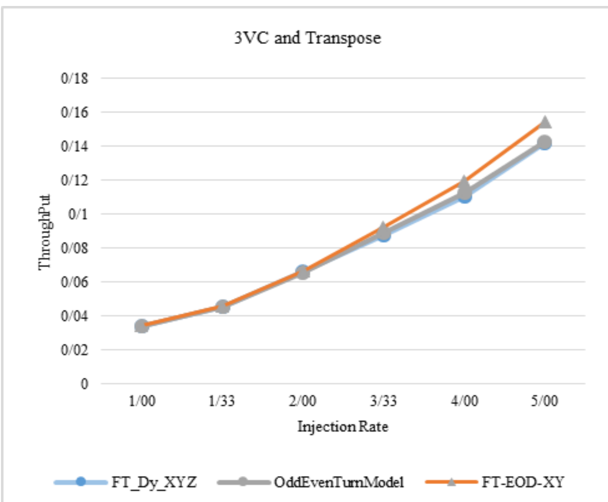

Figure 11: average throughput of our algorithm, FT-DyXYZ and odd-even in transpose traffic

- Transpose traffic

Figure 10 demonstrates average packet latency and Figure 11 shown as average throughput of proposed algorithm, FT-DyXYZ and odd-even routing for 8% fault in transpose traffic

By using enhanced Odd-Even and divide the traffic around the faulty region base on routing direction of packet, FT-EOD-XY shows significant performance in none uniform traffic patterns like hotspot and transpose traffic in contrast to FT-DyXYZ and Odd-Even. It cause lower latency in FT-EOD-XY with increased of injection rate against the other algorithm.

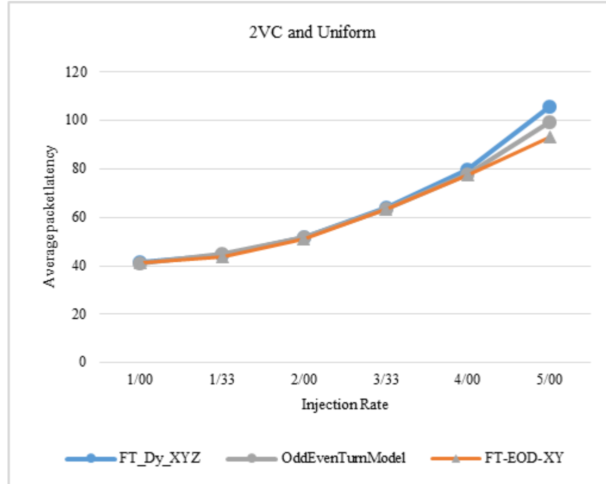

Figure 12: average packet latency of our algorithm, FT-DyXYZ and odd-even in uniform traffic with 2VC

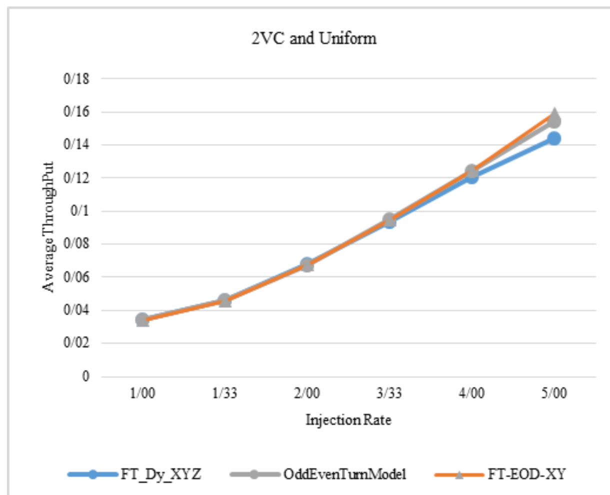

Figure 13: average throughput of our algorithm, FT-DyXYZ and odd-even in uniform traffic with 2VC

Figure 12 demonstrates average packet latency and Figure 13 shows average throughput of FT-DyXYZ algorithm and odd-even routing for 8% fault in uniform traffic with 2 virtual channel. As demonstrated in Figure 12, 13 after decrease number of VCs to 2, FT-EOD-XY because of use XY routing has better latency and throughput against FT-Dy-XYZ and Odd-Even algorithm in uniform traffic pattern.

To evaluate the effect of number of VCs on the performance of our algorithm, we have simulated FT-DyXYZ in 2D dimension and odd-even routing mode under uniform, hotspot and transpose traffic with 2 VCs, 3 VCs. Figure 6-13 demonstrate these situations.

Figure 6-13 demonstrated improvement of saturation injection rate in FT-EOD-XY to the other algorithm in different traffic pattern and 8% fault of total link.

5. Conclusion

In this paper, a routing algorithm, called FT-EOD-XY, was proposed. It is congestion-aware and more link failure tolerant compared to the odd-even and FT-DyXYZ routing technique. Our algorithm based on XY routing and an enhanced odd-even mechanism that divide the traffic around the faulty region base on routing direction of packet. It just need local information about fault and every node must only be aware of its own faulty link, so it can handle runtime faults

Simulation results demonstrate significant improvement of latency, saturation point and reliability in FT-EOD-XY compared to FT-DyXYZ and odd-even for 4 fault (8% of total links failure) in uniform, hotspot and transpose traffics. FT-EOD-XY routing algorithm also can uses no virtual channel to route packets.

Acknowledgment

The authors would like to express their cordial thanks to Arshin Rezazadeh for his valuable advice.

References

[1] Benini and G. De Micheli, "Networks on chips: a new SoC paradigm," Computer, vol. 35, no. 1, pp. 70 - 78, Jan 2002.
[2] P. Magarshack and P. Paulin, "System-on-Chip beyond the Nanometer Wall," in Proc. 40th Design Automation Conf, 2003.
[3] W. Dally and B. Towles, "“Route Packets, Not Wires: On-Chip Interconnection Networks," in Proc. 38th Design Automation Conf, June 2001.
[4] A. Topol and et al, "Three-Dimensional Integrated Circuits," IBM J. Research and Development, vol. 50, July-Sept. 2006.
[5] W. Davis and et al, "Demystifying 3D ICS: The Pros and Cons of Going Vertical," IEEE Design and Test of Computers, vol. 22, Nov./Dec. 2005.
[6] Y. Deng and et al, "2.5D System Integration: A Design Driven System Implementation Schema," in Proc. Asia and South Pacific Design Automation Conf, 2004.
[7] M. Ieong and et al, "Three Dimensional CMOS Devices and Integrated Circuits," in Proc. IEEE Custom Integrated Circuits Conf, 2003.
[8] H. Naghibi Jouybari and K. Mohammadi, "A low overhead, fault tolerant and congestion aware routing algorithm for 3D mesh-based Network-on-Chips," Original Research Article Microprocessors and Microsystems, vol. 38, no. 8, pp. 991-999, November 2014.
[9] M. Dehyadgari, M. Nickray, A. Afzali-kusha and Z. Navabi, "Evaluation of Pseudo Adaptive XY Routing Using an Object Oriented Model for NOC," in The 17th International Conference on Microelectronics, 13–15 December 2005.
[10] V. Rantala, T. Lehtonen and J. Plosila, "Network on Chip Routing Algorithms," August 2006.
[11] M. Li, Q.-A. Zeng and W.-B. Jone, "DyXY - A Proximity Congestion-Aware Deadlock-Free Dynamic Routing Method for Network on Chip," in 43rd ACM/IEEE Design Automation Conference, 2006.
[12] X. Lin and L. Ni, " Multicast communication in multi-computer networks," IEEE Transactions on Parallel and Distributed Systems, pp. 1105-1117, 1993.
[13] P. Gratz, B. Grot and S. Keckler, "Regional congestion awareness of load balance in network-on-chips," in International Symposium on High Performance Computer Architecture , February 2008.
[14] P. Lotfi-Kamran, A.-M. Rahmani, A. Afzali-Kusha, M. Daneshtalab and Z. Nabavi , "EDXY – A low cost congestion-aware routing algorithm for network-on-chips," Journal of Systems Architecture, vol. 56, no. 7, pp. 256-264, July 2010.

Davood abednezhad received the BS. degrees, from Jahad Daneshgahi Ahvaz Univ. in 2009

Seyed Enayatallah Alavi BS. degrees from Isfahan University of Technology in 1992, MSc. degrees from Shiraz University and Dr. Eng. degrees from Belarusian National Technical University in 2012. After working as an assistant professor in the Dept. of Computerl Engineering, the Shahid Chamran University of Ahvaz.