Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 57 (2015) 232 – 240
Procedia Computer Science

2015 International Conference on Recent Trends in Computing

Generation and Performance Evaluation of Reconfigurable Fault Tolerant Routing Algorithm for 2D-Mesh NoC

Sandeep Kr Singh[a], Abir J Mondal[b,]*, Alak Majumder[b]

[a]NIT Arunachal Pradesh, Department of Computer Science and Engineering, Yupia and 791112, India
[b]NIT Arunachal Pradesh, Department of Electronics & Communication Engineering, Yupia and 791112, India

Abstract

Recent design techniques are integrating 10 to 100 embedded functional and storage blocks in a single system on chip and the number is growing to increase with further advancements. The bus based interconnections are not a suitable alternative for Massively Parallel Multi-Processors Systems on Chip (MPSoCs) because of power and latency issue. The communication requirements of many-core embedded systems are addressed by the Networks on Chip (NoC) paradigm. In this paper, a minimal and fault tolerant routing algorithm is proposed so as to route packets adaptively through the shortest path in the presence of faulty nodes. Using fault-tolerant routing algorithm to reroute packets around faulty nodes will increase latency. Besides, the performance of NoC is heavily affected by network congestion. Congestion in the network increases the time to traverse a packet from a source to a destination. The proposed routing algorithm adaptively chooses the next node where to send packet, so as to avoid packet drop in presence of congestion. The algorithm does handle both single and multiple busy nodes using reconfigurable paths (minimal and/or non-minimal).

Peer-review under responsibility of organizing committee of the 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015)

Keywords: NoC; Fault Tolerance;Routing Algorithm; Deadlock; Packet Drop.

1. Introduction

According to Moore's law, billions of transistors could be integrated on a single chip as we advance towards deep submicron technologies [1]. Inside these chips are embedded hundreds of functional intellectual property

* Corresponding author. Tel.: +919436288558.
E-mail address: abir_jm@hotmail.com.

Peer-review under responsibility of organizing committee of the 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015)
doi:10.1016/j.procs.2015.07.471

(IP) blocks and memory modules to form MPSoCs [1]. The traditional bus based architecture in MPSoCs fails to address the present complexity due to increase in the number of processing elements in a single chip and hence a new communication backbone is required. NoC has been a promising solution for on chip interconnection in many-core Systems-on-Chip (SoC) due to its reusability and scalability [2-4].

When implemented in deep submicron technology, on chip interconnects are prone to failures [5-6]. Due to this aggressive device scaling, the chances of failure increases [7]. Two different types of faults that are predominant in NoC are transient and permanent. Unpredictable causes are associated with transient faults and are often difficult to be corrected and detected. Permanent faults are caused by physical damages. In this work, permanent fault has been discussed. Faults can be tolerated by many methods and majority of them are based on fault tolerant routing algorithms. Majority of routing algorithms are subdivided into deterministic and adaptive types [8-9]. A deterministic routing algorithm uses a fixed path for each pair of node, resulting in increased packet latency in congested networks. Deterministic routing algorithms are easy to implement but they are unable to balance load across the nodes. Unlike deterministic routing algorithms, adaptive routing algorithms could avoid congestion in the network and provide better fault-tolerant characteristics by utilizing alternative routing paths.

Wormhole routing is the most commonly used flow control technique, where packet is divided into smaller flow control units called flits and traversed through the network in a pipelined fashion. This approach eliminates the requirement to allocate large buffers in intermediate switches along the path [10]. It should be used with special care to avoid deadlock and packet drop in the network. Conventional fault tolerant routing algorithms reroute packets around faulty regions, so the selected paths are not always the shortest ones. Rerouting is an expensive solution and considerably increases packet's latency and router's complexity. The information about faulty nodes is insufficient. In this paper, we have presented a fault tolerant routing algorithm that has solved problems associated with deadlock and packet drop. The proposed algorithm has solved the associated problems in three ways. First, it can tolerate all one-faulty nodes using a minimal path between source and destination, if a minimal path exits. Second, to avoid congestion, output channels can be adaptively chosen whenever the distance from the current to destination node is greater than one hop. Third, situations are depicted where the proposed algorithm handles multiple busy nodes.

2. Related Work

Fault tolerant routing algorithms can be separated into two groups: one that uses convex or concave regions [11-14] and the other utilizes contour strategy for addressing faults [15-16]. It can also be classified into two classes: the methods using virtual channels [16-18] and those without using virtual channels [19-20]. It is also possible to implement routing algorithms as either table-based or in algorithmic form [21-22]. In algorithmic routing mechanism, an algorithm is executed using hardware circuits using FSM to compute appropriate router port. It is generally suitable for one topology. Table based mechanism is used to deal with regular as well as irregular topologies. The table based methods cannot scale well since the table size increases with the size of network and may become impractical. In the application specific platforms where communication transactions among IP cores are known in advance, it is quite possible to use compression techniques [23] to reduce the size of tables instead of straight forward table based implementation. In [24], authors discussed efficient implementation of distributed routing algorithms for partial 2D meshes without using routing tables. Most of the fault tolerant routings use either virtual channels [18] or turn models [25] based strategies to achieve deadlock freedom. In this paper, we present a reconfigurable, deadlock free, cost efficient routing algorithm without using virtual channels.

3. Routing Algorithm

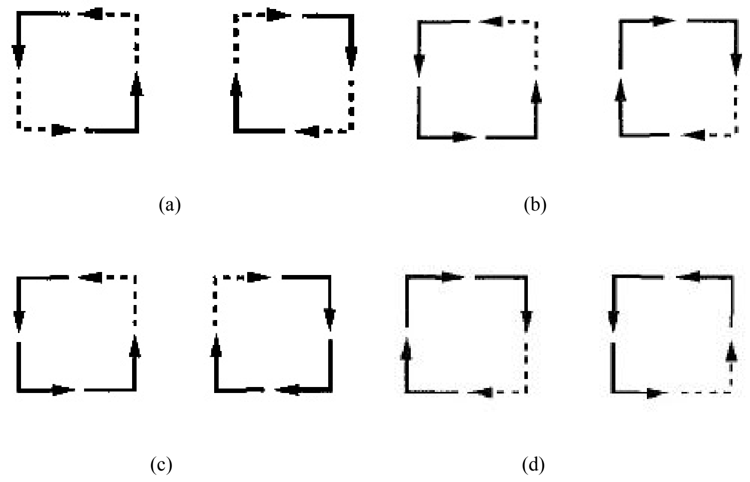


Fig. 1. (a) xy; (b) West-First; (c) North-Last; (d) Negative-First

Routing involves selecting a path from source node to destination node in a particular topology. Topology determines the ideal performance of a network and routing is one of the two key factors that determine how much of the potential is realized. The simplest deadlock free routing algorithm for mesh NoCs is xy routing. The packets are first routed along x dimension to the correct column and after that in the y dimension to the correct row. West-first, north-last and negative-first are turn model that prohibit minimum number of turns. The allowed and prohibited turns in xy and turn model are shown in figure 1.

3.1. Fault Distribution Mechanism in the Proposed Methodology

A unique fault distribution methodology is described which avoids redundant non-minimal paths. The fault information is shared in such a way that each router is informed about the fault condition in its immediate neighbor and also at multiple hops through its neighbor. Using this information, unnecessary paths are avoided to prevent packet drop in case destination is unreachable and deadlock. Figure 2 depicts the proposed fault distribution methodology. The current router is aware of the faulty nodes in one hop distance. E, W, N and S stand for the East, West, North and South directions. In figure 2(a), the neighboring nodes share their condition with current node (C). In figure 2(b), the node in the East direction is aware of the faulty nodes in its neighbor (NE, EE and SE). Similarly, in figure 2(c), the neighboring nodes (SE, SS, SW) share their condition with South directed node. Finally, in figure 2(d), the node in the West direction has knowledge about its neighboring nodes (SW, WW and NW). In this manner the current node is not only aware about its immediate neighbor, but also possesses information about nodes in two hop distances through its immediate neighbor.

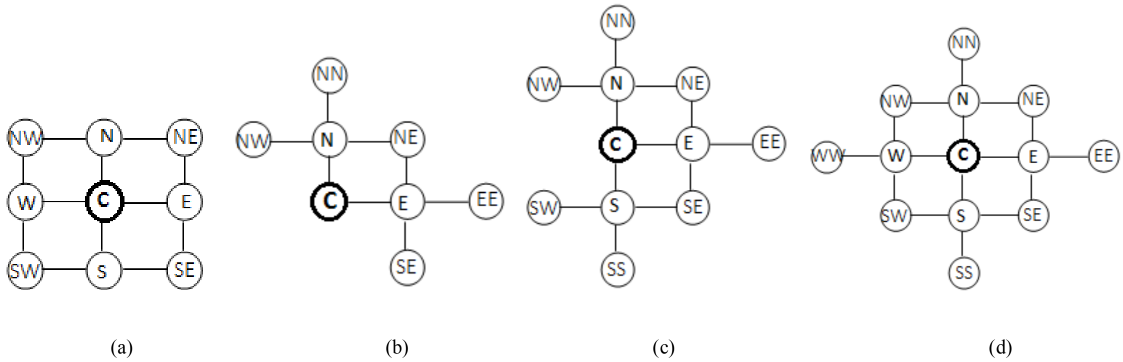


Fig. 2. The fault distribution methodology

For routing a packet in the northeast direction, the router uses information about faulty nodes from its immediate neighboring nodes in the north and east directions because they are aware about the condition of their neighbors (NE). Similarly, for a southwest packet, the information on south and west directed nodes is beneficial for making a reliable routing. Using this information, packets are routed through minimal and non-faulty nodes which avoids making unnecessary routing around faulty nodes.

3.2. Bypassing Faulty Nodes

According to the relative position of the source and destination nodes, a packet can be sent in eight directions: north, south, east, west, northeast, northwest, southeast and southwest. By using the proposed distribution methodology, we have shown that packets destined for northeast, northwest, southeast and south west directions take only the shortest paths in the presence of the faulty nodes in the network. As a consequence, rerouting does not take place and the algorithm avoids deadlock and packet drop. In case of eastward, westward, northward and southward packets, non-minimal paths must be traversed if faulty nodes are present.

3.2.1. Northeast, Southeast, Northwest and Southwest Directions

When the destination is in the southeast position of the current node, the packet can be traversed in either the south (S) or east (E) direction. As depicted in figure 3(a), the distances along both east and south directions are one

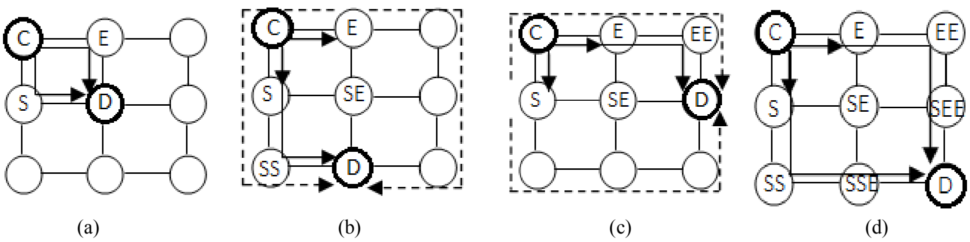


Fig. 3. Bypassing faulty nodes when the destination is located in the southeast position of the source node

. Using the fault distribution methodology, the current node is informed about faulty nodes in its immediate neighbor along east and south directions. Using this information, if a node turns out to be faulty or busy, the other node is selected by the routing algorithm. The packet is always routed towards destination. In figure 3(b), the current node is aware about its immediate neighbor (E and S) . The packet can be traversed in the east direction if the node in the east is non-faulty. Now the east directed node has knowledge about its immediate neighbor (SE). If the node in the south direction (SE) is busy or faulty, the packet must take either of the non-minimal paths along the dotted line to bypass the faulty node. This is not an optimal solution. The proposed algorithm avoids rerouting because the current router possesses information not only about its immediate neighbors (S and E) but also about nodes (SE and SS) at two hops distances with the help of immediate neighbors. As a consequence, the packet will be routed along the south of the current node. On reaching the south node the situation is identical to that of figure 3(a). Since the east directed node (SE) is faulty the packet is again routed towards south node (SS) and finally along east towards destination. In another case of figure 3(b), if the packet is routed along south of the current node and if either the east (SE) or further south bound node (SS) turns out faulty, the packet is always traversed towards destination along minimal path. Similarly in figure 3(c), the current node has equal chances to send packet either to east (E) or south (S) direction. If the south node is non-faulty, packet can be routed in the south direction. The south directed node has information about its immediate neighbor (SE). On reaching the south node, if the east node (SE) turns out faulty rerouting along dotted path has to be done. Similarly, if the packet is first routed along the east of current node and if either neighbor (SE and EE) turns out faulty, the situation is identical to figure 3(a). Finally, in figure 3(d), the packet can be delivered either through east bound or north bound nodes depending on the information about neighboring nodes.

3.2.2. East, West, North and South Directions

When a packet is eastward, westward, northward or southward and there is a faulty node in the path, the packet must be routed through a non minimal-path. As illustrated in figure 4(a), for the eastward packet, at first the east node is checked and if it is non-faulty, the packet is sent through this node. However, if the node is faulty, the packet is delivered to the north or south direction with same priority. The situation is similar for westward packet. When the packet is north bound and the north node is faulty, depending on the location of destination either the east or the west node is checked earlier. A similar explanation applies to southward packet.

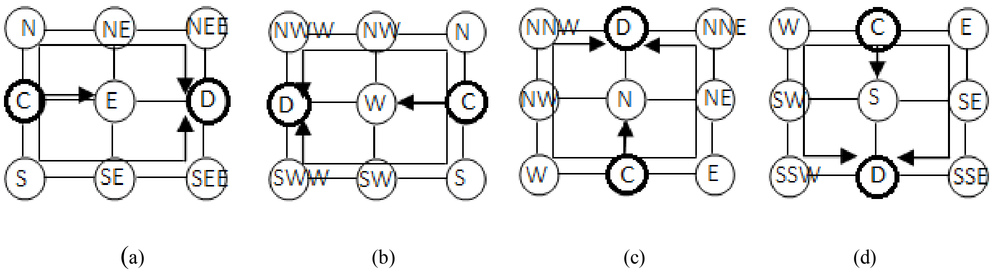
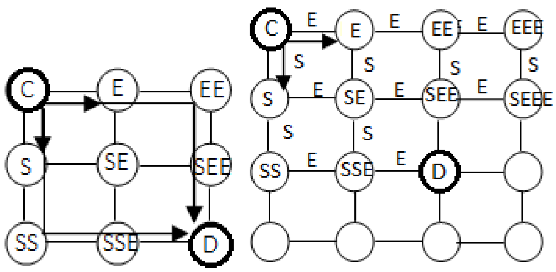


Fig. 4. Bypassing faulty nodes when the destination is located in the (a) east; (b) west; (c) north and (d) south positions of the source node

4. Adaptive Fault Tolerant Routing Algorithm

A deterministic routing algorithm is a known method used in traditional fault tolerant methods because the path to be traversed is predictable. Based on the proposed algorithm, the next node to be traversed can be adaptively chosen because the current node is not only informed about immediate neighboring nodes but also about nodes at two-hop distances. Using the fault distribution methodology described above, the fault tolerant routing algorithm is defined as follows;

- If the packet's destination is connected to the router, forward the packet to the processing element (PE)
- If source and destination are at the same row/column traverse perpendicularly.
- The current and destination nodes are located in such a manner that there is only one busy node between them. The current node is informed about this faulty node through its neighbors and routes packets using minimal path. As illustrated in figure 5(a), the current node has information about neighboring nodes (E and S). If the east node is non-faulty, the packet is routed towards it. The east node is also aware about its neighbor (EE and SE). If either one is faulty, the packet is routed through minimal path. The same situation is prevalent if the packet was routed along south node.
- The current and destination nodes are located in such a manner that there are multiple faulty nodes present in the network. The current node is aware about the neighboring nodes and also about the nodes at more than one hop distance through neighboring nodes. Using this information, the packet is always routed through minimal path (rerouting is not required). In figure 5(b), there are multiple busy nodes in between current and destination. The current node has equal probability to route packet either to east (E) or south (S). If the east bound node is non-faulty, packet is routed east. The east node is informed about its neighbor (EE and SE). If either one is faulty, the packet is routed through the other path. If the south bound node (SE) is non-faulty, packet is routed south. The south node is also aware about its neighbor (SSE and SEE). Again, if either one is faulty, the packet is directed through the other one. If the node SEE is not busy packet is directed towards destination through SEE. Similarly, if the south bound node is non-faulty, packet is routed south. The south node is aware about the neighboring nodes (SS and SE). If the packet is routed through SS rerouting is required because SSE is faulty. Using this information packet is directed towards SE. Again, the node SE is aware about neighbor and routes packet accordingly to destination. In the case of figure 5(b), similar discussions are possible for any arbitrary location of faulty nodes.
- Take u-turn if all neighboring nodes are busy.

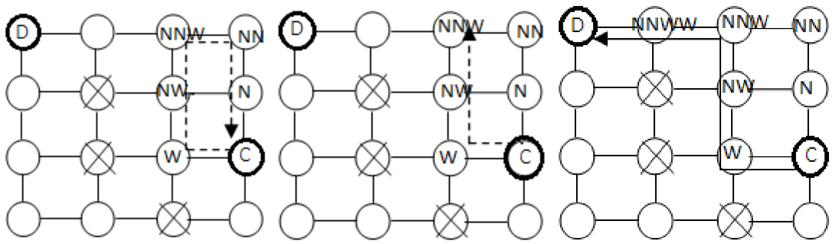


(a) (b)

Fig. 5. Cases showing occurrence of (a) single faulty node and (b) multiple faulty nodes

4.1. Comparison of proposed algorithm with traditional methods

Figure 5 shows an example comparing the proposed fault tolerant algorithm with traditional model. All the



(a) (b) (c)

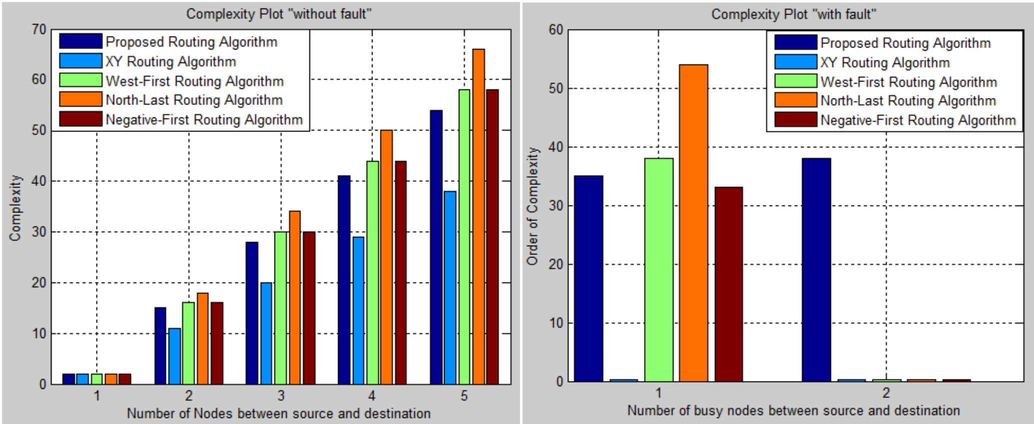
Fig.6. Comparison of (a) west-first, (b) north-last with (c) proposed method

models select next node based on fault information. In figure 6, packet is sent from the current node to destination with the nodes marked cross denote faults. In west-first routing algorithm, if a packet has to be routed west, it has to be done first. Later on it cannot be routed west, if it has taken other turns (N,S and E). In figure 6(a), packet is first directed west (W). On reaching west, if finds the neighbor node faulty and routes adaptively to north node (NW). Once north turn has been taken it cannot further take west and keeps on moving along the dotted path. Similarly, in north-last routing algorithm, if a packet has to be routed north, it has to be done at last. Figure 6(b), shows the situation where north turn is taken early because of faulty nodes in the network. Consequently, the packet travels along dotted path and never makes to destination using north-last algorithm. Lastly, figure 6(c) depicts the proposed method. Using the fault distribution methodology described above, the current node is aware about the neighbors and also at 2 hop distances using neighbors. Since the west node (W) is non-faulty, the packet is first routed west. The west node is informed about its neighbor (NW and ⊗). The packet is next directed towards north bound node (NW). The node NW is also aware about its neighbor (NNW and ⊗). So it routes packet towards NNW and finally through node NNWW to destination.

5. Analysis and Results

We have evaluated the proposed routing algorithm in terms of complexity and successful arrival rate. The term complexity refers to number of comparisons required by routing algorithm to reach destination. To determine complexity we have assumed two cases; mesh without faulty nodes and with faulty nodes. We have considered a 2D mesh with predefined source and destination locations. The performance of routing algorithms is measured as the number of nodes between source and destination are increased. It is observed from figure 7(a), that the proposed routing algorithm has better complexity compared to turn model and surrounding-xy whereas the performance is little higher in comparison to xy. Next we have introduced faulty nodes in the mesh and evaluated the performance for different positions of faulty nodes. Figure 7(b) illustrates that for a particular location of faulty nodes, xy algorithm fails to traverse packet. But the proposed methodology depicts better performance as compared to turn model. In another case of figure 7(b), the position of busy nodes make it impossible for turn model and xy to deliver packet to destination, whereas the proposed method shows much better results.

We have assumed initially that there are no busy nodes in the beginning and then faulty nodes are uniformly increased at a progressive rate. In the case of xy algorithm, the probability of traversing packet to destination is 1 when the fault rate is 0%. So, successful arrival rate of algorithm is 100%. When the fault rate is 1%, the probability of occurrence of busy node is 1/100.The total probability of occurrence of busy nodes is 4/100 i.e. 1/25. But the packet can traverse in only 3 directions because one path is meant for entering. Then the chances of



(a) (b)

Fig.7. Plot of complexity with (a) fault and (b) without fault

occurrence of busy nodes are 3/25; the failure rate of algorithm is 12% and successful arrival rate is 88%. The analysis is done upto 5% fault rate and the corresponding arrival rates are obtained. Similarly, the probability of traversing packet to destination in west first routing algorithm is 1 when the fault rate is 0%. This indicates that successful arrival rate is 100%. Since in west first algorithm packet is first directed west if necessary else it cannot be routed later, the probability of occurrence of busy node is 1/100 when the fault rate is 1%. So the failure rate is 1% and successful arrival rate is 99%. The same analysis holds true for north last algorithm also. Figure 8 depicts the desired plot of successful arrival rates as the fault rate is increased. It is observed that the proposed method has better arrival rates compared to traditional model except when there are faulty nodes in all the neighbors of destination.

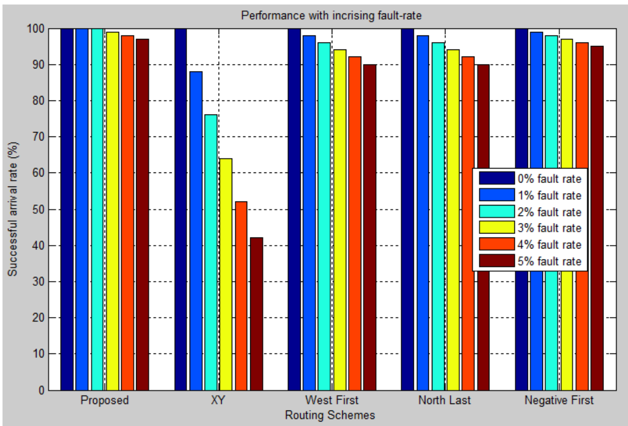


Fig. 8. Performance with increasing fault rate

6. Conclusion

In this paper we have proposed a fault tolerant routing algorithm that has solved deadlock and packet drop issues associated with traditional algorithms. We do not restrict our algorithm to be minimal since the target is to specify a deadlock-free routing algorithm in fault tolerance perspective. We are working on hardware realizations of the proposed algorithm and also on router architecture so as to verify the performance in real scenario.

References

[1]Xu, Jing et al. ,2005, "A methodology for design, modeling and analysis for network-on-chip, " in Proc. IEEE International Symposium on Circuits and Systems, pp.1778-1781.
[2]W. Tsai et al. , 2011, "A fault-tolerant NoC scheme using bidirectional channel", in Proc. DAC, pp.918-923.
[3]E. Rijpkema et al. , 2003, "Trade offs in the design of a router with both guarantee and best-effort services for networks on chip", in Proc. DATE'03, pp. 350-355.
[4]M. Cuviello et al. , 1991, "Fault modeling and simulation for crosstalk in system-on-chip interconnects", in Proc. ICCAD, pp.297-303.
[5]M.H Neishaburi et al. , 2007, "HW/SW architecture for soft-eError calculation in real-time operating system", Journal of the Institute of Electronics, Information and Communication Engineers, Vol. 4, No. 23, pp. 755-761.
[6]J. Duato et al. , 2003, "Interconnectin networks: an engineering approach", Morgan Kaufmann publishers.
[7]L.M. Ni et al. , 1993, "A surway of wormhole routing techniques in direct networks", in Proc. IEEE Computer , V. 26, I.2, pp. 2-7.
[8]D. Fick et al. , 2009, "Vicis: a reliable network for unreliable silicon", in Proc. of Design Automation Conference, pp. 812-81.
[9]S. Chalasani et al. , 1995, "Fault-tolerant worhmhole routing algorithms for mesh networks", IEEE Trans on Computers, 44(7):848-64.
[10]PH. Sui et al. , 2011, "An improved algorithm for fault-tolerant wormhle routing in meshes", IEEE Trans on Computers x;46(9): 1040-2.
[11]S. Park et al. , 2008, "Fault-tolerant wormhole routing algorithms in meshes in the presence of concave faults", in roc. of International Parallel and Distributed Processing Symposium (IPDPS), pp. 441-446.
[12]Z. Zhang et al. ,2008, "A reconfigurable routing algorithm for fault-tolerant 2D-mesh Network-on-chip", in Proc. DAC, pp. 441-446.
[13]M. Valinataja et al. , 2011, "A reconfigurable and adaptive routing method for fault-tolerant mesh-based Network-on-Chip", in Proc. International Journal of Electronics and communications (AEU), V. 65, I.7, pp. 630-640.
[14]M. Koibuchi et al. , 2008, "A lightweight fault-tolerant mechanism for Network-on-Chip", in Proc. NOCS, pp. 13-22.
[15]J. Wu, 2003, "A fault-tolerant and deadlock-free routing protocol in 2D meshs based on dd-even turn model", IEEE Transation on computers, V. 52, pp. 1154-119.
[16] D. Fick et al. , 2009, "A highly resilient routing algorithm for fault-tolerant NoCs", in Proc. DATE, pp. 21-26.
[17] W. J. Dally and B. P. Toweles, 2004, "Principles and Practices of Interconnection Netwrks", Morgan Kaufmann.
[18] J. Duato, S. Yalamanchili, and L. Ni, 2003, "Intercnnection Networks – An Engineering Approach", Morgan Kaufmann.
[19]M. Palesi, R. Holsmark, S. Kumar, and V. Catania, 2009 ,"Application specific routing algorithms for network-on-chip", Prallel and Distributed Systems, IEEE Transactions on, vol. 20, no. 3, pp. 316-330.
[20]J. Flich and J. Duato, 2008, "Logic-based distributed routing for NoCs", IEEE Computer Architecture Letter, Vol. 7, No. 1, pp. 13-16.
[21]C. Glass and L. Ni, 1992, "The turn model for adaptive routing", in Proc. 19th Internatinal Symposium on Computer Architecture, pp. 278-287.
[22]Z . Zhang, A. Grener, and S. Taktak ,2008, "A reconfigurable routing algorithm for fault-tolerant 2D-mesh Network-on-chip", in Proc. 45th DAC, pp. 441-446.
[23]G. M. Chiu, 2000, "The odd-even turn model for adaptive routing", IEEE Transactions on, vol. 11, no. 7, pp. 729-738.
[24]B. Fu, Y. Han, J. Ma, H. Li, and X. Li, 2011, "An abacus turn model for time/space-efficient reconfigurable routing", in Proc. 38th International Symposium on Comuter Architecture, pp. 249-270.
[25] C. J. Glass and L. Ni, 1993, "Fault-tolerant wormhole routing in meshes", in Proc. 23rd Internatinal Symposium on Fault-tolerant Computing, pp. 240-249.