Design and implementation of secured agent based noc using shortest path routing algorithm

Kendaganna Swamy S, Anand Jatti, Uma B. V.
Department of Electrical & Instumentation Engg, R.V College of Engineering, Bangalore, India

Article Info

Article history:

Received Jul 9, 2018
Revised Oct 10, 2018
Accepted Nov 19, 2018

Keywords:

Congestion
Fault tolerant
Network on chip
Permanent fault
Random arbiter
Routing algorithm
Security
Shortest path routing

ABSTRACT

Network on chip (NoC) is a scalable interconnection architecture for every increasing communication demand between many processing cores in system on chip design. Reliability aspects are becoming an important issue in fault tolerant architecture. Hence there is a demand for fault tolerant Agent architecture with suitable routing algorithm which plays a vital role in order to enhance the NoC performance. The proposed fault tolerant Agent based NoC method is used to enhance the reliability and performance of the Multiprocessor System on Chip (MPSoC) design against faulty links and nodes. These agents are placed in hierarchical manner to collect, process, classify and distribute different fault information related to the faulty links and nodes of the network. This fault information is used for further packet routing in the network with the help of shortest path routing algorithm. In addition to this the agent will provide the security for the node by setting firewall, which then decides whether the packet has to be processed or not. This intern provides high performance, low latency NoC by avoiding deadlock and live lock with low area overhead.

Corresponding Author:

Kendaganna Swamy S,
Department of Electrical & Instumentation Engg,
R.V College of Engineering, Bangalore, India.
Email: kendagannaswamys@rvce.edu.in

1. INTRODUCTION

As a feature size of the transistor scaling down, the number of transistor on a single die increases whose result improves the number of IP cores on the SoC. As number of IP cores increases on SoC the networking becomes a bottleneck issue. The conventional crossbar and switches will not be able to support the communication between more number of IP core based SoC because of their performance degradation. In order to improve the communication performance on SoC a new method has been introduced that is Network on Chip. Such NoC is capable to improve the networking performance on MPSoC [1] when compared to conventional method if and only if network is fault free.

If there is any local fault on the network, it may be in router or link or Processing Element (PE) or in Network Interface (NI) element. Due to this fault [2], [3], there may be a chance for the packet to enter into the deadlock or live lock or packet loss. Then packet has to remap once again which leads to performance degradation. These local faults [4] may occur during the fabrication process or down the length of year of usage. Hence there is a demand to create awareness of local fault in NoC before mapping any packet into any of the node in the network [5]. Hence there should be an intelligence unit on NoC to be aware about local fault before mapping any packet.

The Background work in [6-8] the author addresses the local fault awareness using fault tolerant routing algorithm and it is software in nature which may lead to long routing path hence there is a more packet latency and throughput degradation [9]. In [10]-[12] the author introduces the hardware based hierarchical agents on the NoC in order to gather and classify the fault information and transfer that message to upper layers [13]. Such agent based NoC method uses the odd even based routing algorithm. This method will improve the network performance with an area overhead and these agents are unsecure in nature [14]. Hence there is research scope to enhance the feature of fault tolerant agents.

1.1. The problem

The significant research problems are as follows:

a. Existing agent based NoC systems focused only an XY routing algorithm.
b. Conventional research towards fault tolerance doesn't emphasize on the scalability while evolving up with fault tolerant protocol over network design.
c. Although existing studies have worked on fault identification but there are less number of studies towards classifying the faults existing over the networks.
d. None of the existing studies towards NoC has highlighted any design issues with its processing elements that offer latent faults in any network architecture.
e. The mechanism of formulating the decision in ensuring better performance of fault tolerance network is not clearly defined in any existing studies.

Therefore, the problem statement of the proposed study can be stated as "*Developing a cost effective modeling to encapsule comprehensive network faults with equivalent focus on packet-level controlling mechanism in chip architecture with different routing algorithm is computationally challenging.*"

1.2. The proposed solution

The prime aim of the proposed system agents are not only fault and congestion information provider but also takes the decision whether packet has to pass or not to the Processing Element by setting the firewall. Such secured agents will give two benefits such as placing the firewall on the chip is safer from the hackers compare to off chip firewall. Second is, if the specific packets can be able to execute only through the specific PE that time by securing such node we can overcome the waiting of highest priority packet looking for the specific PE and overcome the live lock situation [15]. In the proposed design in order to route the packets, shortest path routing algorithm is used which will improve the network performance with less area overhead as compare to [16]-[20] the proposed stsem is described in detail in Section 2.

2. PROPOSED HIERARCHICAL AGENTS BASED MONITORING SYSTEM

The performance of NoC based multiprocessor system on chip depends on the packet switching and processing rate on the network. The network has fault or congestion because of a faulty link or a router or a PE which may occurs in manufacturing or in a operational phase. If the upper layer is not aware of such local fault and congestion information then the packet enters into the dead lock and live lock situation which leads to performance degradation. Then the proposed agents need to be placed in hierarchical manner as shown in Figure 1. These agents will collect and classify the local fault [21], [17] and congestion information and send them to the upper layer before mapping any packet into the node from the application layer. In addition to this the proposed agent provides security to decide whether the received packet as to pass or not to the processing element by setting the firewall.

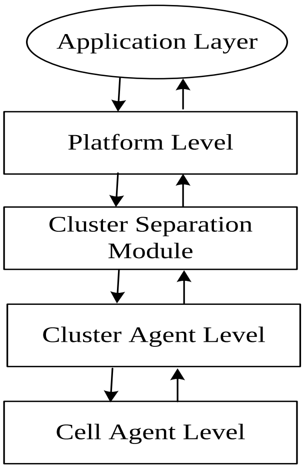


Figure 1. Proposed hierarchical agent monitoring system flow

The proposed hierarchical agent system has five layers namely application layer, platform level, cluster separation module, cluster agent and the cell agent. The platform level will receive the packet from the application layer and try to produce the error free packet to the network layer by considering all network erroneous scenarios [22] Cluster separation module decides the received packet belongs to which cluster under N number of clusters. These cluster agents have N number of cell agents. Cluster agent will collect fault information from the cell or node agents and update to the upper layer. The node or cell agent dedicated to one node will collect, accumulate and distribute the fault and the congestion information of its own node and neighboring node by updating the local fault register [LFR] and regional fault register [RFR] respectively.

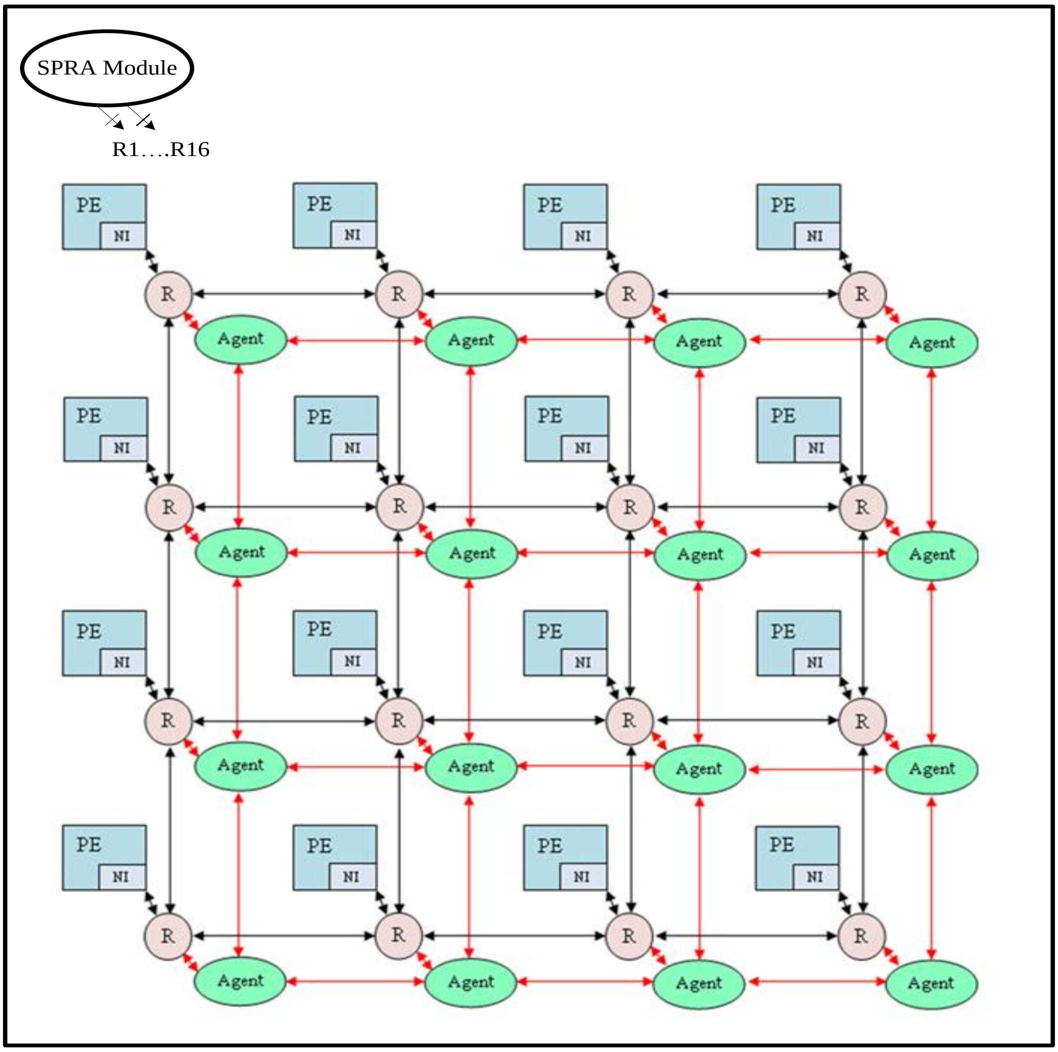


Figure 2. 4x4 hierarchical agent based NoC

A 4x4 hierarchical agent based NoC is as shown in Figure 2, which consist of PE, router [23], NI and the agents. All the agents are connected bidirectional in order to perform pear to pear communication and there is a one bit information exchange between the agents to update the RFR. These agents are connected to router network [24]. The packets form the upper layer enters into the router via cell agent in order to check the security parameters which will be elaborated in the next section. The base line data communication takes place between the routers in the network. In addition to this, the SPRA module is added to the network to perform packet routing using shortest path routing algorithm [16]. The SPRA module is connected to each and every node of the network.

3. PROPOSED SECURED CELL AGENT DESIGN

The proposed cell agent is as shown in Figure 3 which will perform the following tasks: fault detection, security, session monitoring and congestion detection hence the proposed design called as secured agent.

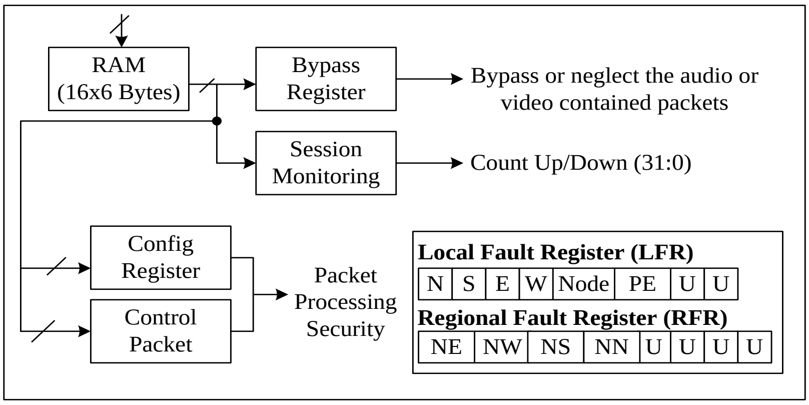


Figure 3. Proposed cell agent design

3.1. Fault information classification

The network fault information classification is useful for the router to perform the routing process. The fault detection circuitry in the agent will provide the appropriate signals. These signals provide the faulty component information such as links, router and processing element. The cell agent accumulates, manage and distribute the fault information of its own components with the help of LFR and update the neighboring node faults with the help RFR.

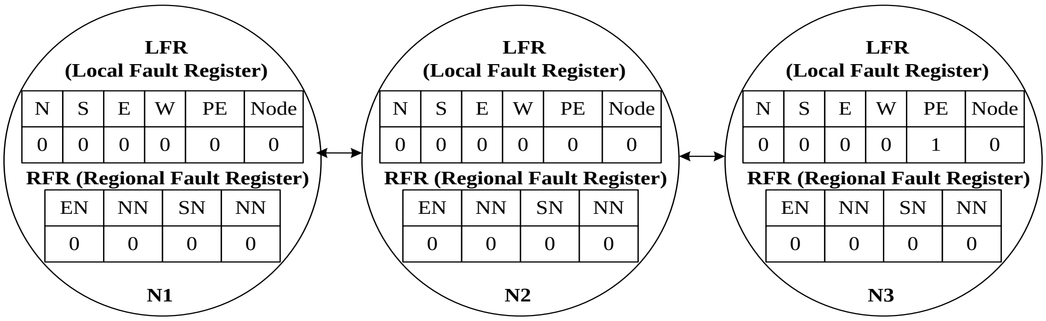


Figure 4. Fault information exchange between the agents

Figure 4 shows that how the cell agents update the fault information between the neighboring cell agents. Assume N2 is the current node; its LFR gets updated based on its own component faulty status of its PE, NI and the router. The RFR gets update based on faulty status received from the E, W, N and south side neighboring agents. Assume Node3 PE is faulty, then the N3 LFR PE bit changes from status ‘0’ to ‘1’ this faulty status gets updated in the neighboring agents RFR. The N2 is the neighbor of N3, the N2 RFR east Node [EN] bit status changes from 0 to 1 to indicate east side node [N3] is faulty. This indicates that the east side node is unhealthy and the packet from N2 will be routed towards south side rather than east side by considering the RFR status to reach the destination of bottom right side of the node in the network. The congestion information or the fault information is determined by the agents with the help of encoding and decoding process.

3.2. Agent security

This cell agent will provide the security to the processing element using config register and control packet stage. Config register is used for source port configuration (using lookup table concept) in order to block the unwanted and unrelated packets to give security (like blocking the website or virus packets).

Control packet will get the authorized packet information from the config register and decides whether packet must be passed or not to the processing element. In general, people can hack the secured firewall, but in the proposed design, some of the port addresses are itself blocked in the hardware (i.e inside the chip), which avoids the intruder by hacking the firewall. The cell agent will ignore or bypass some of the packets, if those packets contain video or audio related data using bypass register. The agents will also monitor the maximum sessions per node using session monitoring stage. This session monitoring stage will take care of start session and close session (limited to 0 - 31 sessions) after performing the task.

4. SHORTEST PATH ROUTING ALGORITHM

In the proposed system in order to rote the packet on the network, shortest pate routing algorithm (SPRA) is implemented. This SPRA module is connected to each and every node of the network and exchanges the one bit information bidirectional as shown in Figure 2. The SPRA will assign the weightage to each and every node from 0 to 7; this weightage assignment indicates how busy the node is. If the weightage assigned to the node is 7 means, the node is highly busy or if the weightage assigned to node is 0 means, the node is idle. The SPRA module assigns two registers for each and every node, one is permanent weightage assignment register and another one is temporary weightage assignment register. The value of temp weightage register keeps on updating for each and every movement of the packet where as the permanent weightage register will never change its value. The SPRA module will route the packet on the network based on the weightage. The working flow of SPRA is as shown in Figure 5.

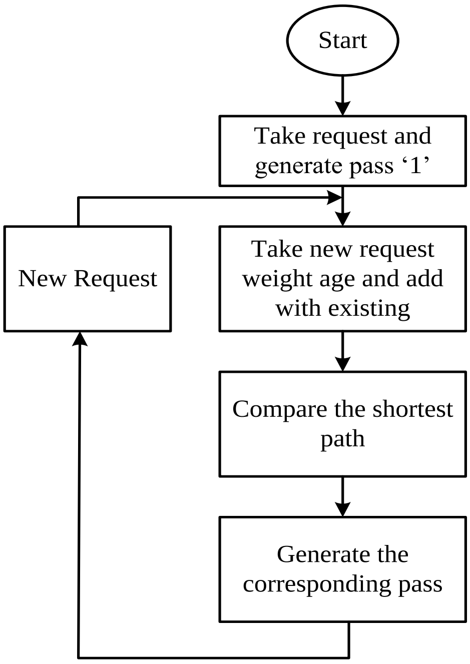

Start
Take request and generate pass ‘1’
New Request
Take new request weight age and add with existing
Compare the shortest path
Generate the corresponding pass

Figure 5. SPRA flow chart

For example if the packet generated in node1 as to reach Node16 using SPRA module, initially node 1 sends the request to SPRA module and SPARA module will take the request and check the weightage status on its internal database and generate the one bit grant information to node1. Then it will check the shortest path based on weightage like node1 can move ether move to node 5 or Node2 which are its neighboring nodes respectively. The SPRA will add the node 1 weightage with Node2 weightage and update the node2 temp weightage register value to new weightage value (i.e. 1+2 = 3). Similarly SPRA will add node1 weightage with Node5 and update the node5 temp weightage register value to new weightage value (i.e. 1+4 = 5). Then SPARA will compare both the weightage value and find the shortest path. Then SPAR will generate the corresponding pass, it means that the packet will be sent from node1 to node2 side because this path is having less weightage compare to node5 side. Ones it reaches the ndoe2 the proposed router will crosscheck the destination location using destination bits on the packet, if this matches, it will stop the routing process else further routing process takes place by generating the new request from node2 and it will be continue till the packet reaches node16 (destination node). Figure 6 shows the shortest path direction from node 1 to 16 with updated weightage assignment. The updated weightage on temp register is as shown in brackets in the Figure 6 and the original weightage assigned in the register will not be affected by the new value.

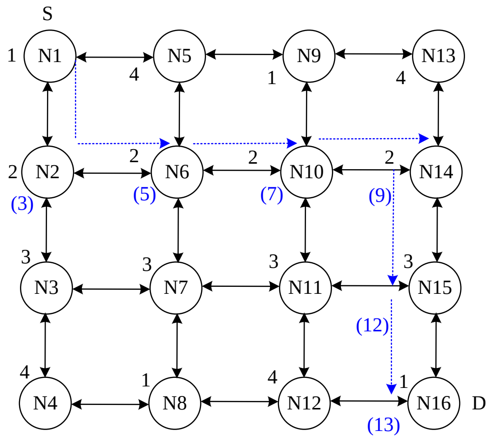
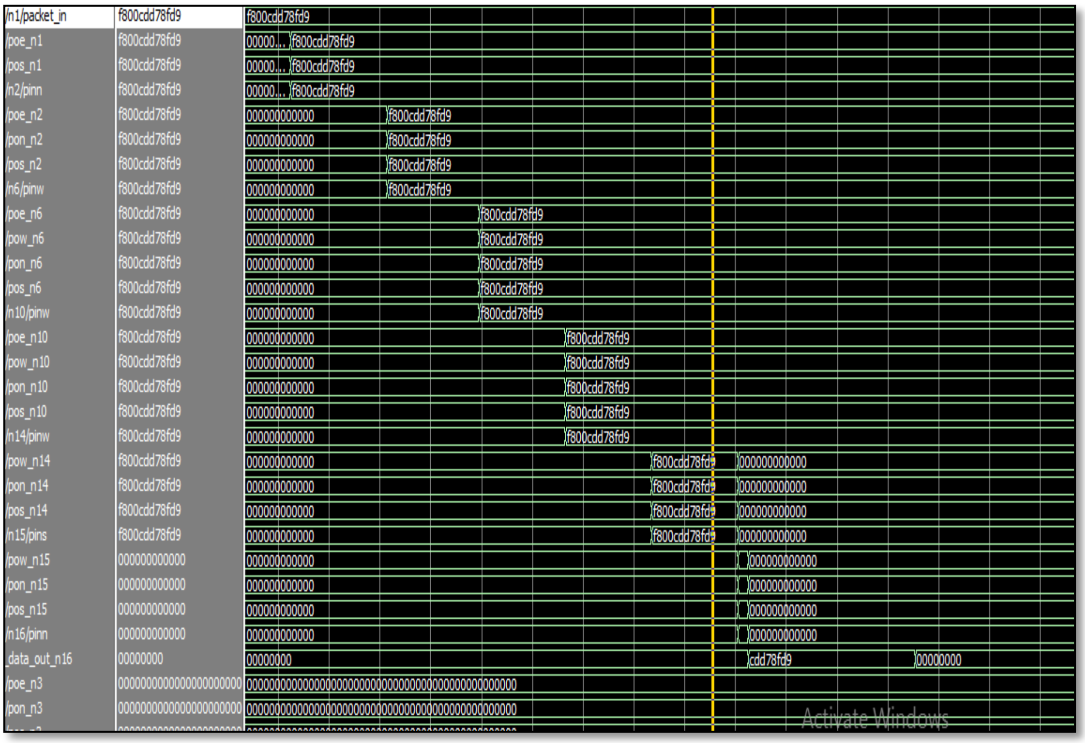


Figure 6. Shows the shortest path direction from node 1 to 16

The Figure 7 shows the simulated waveform with respect the network configuration in Figure 6. The packet moving path N1-N2-N6-N10-N14-N15-N16. In the waveform data enters into the N1 from its own PE and virtually the data is sent out in all the direction first then actual packet is sent out based on the decision of the SPRA module and finally it will send the packet data out to south (PoS) from node1 and data_in to Node2 of north side i.e PinN based on weightage calculation. This will continues till the packet reaches the destination node16.

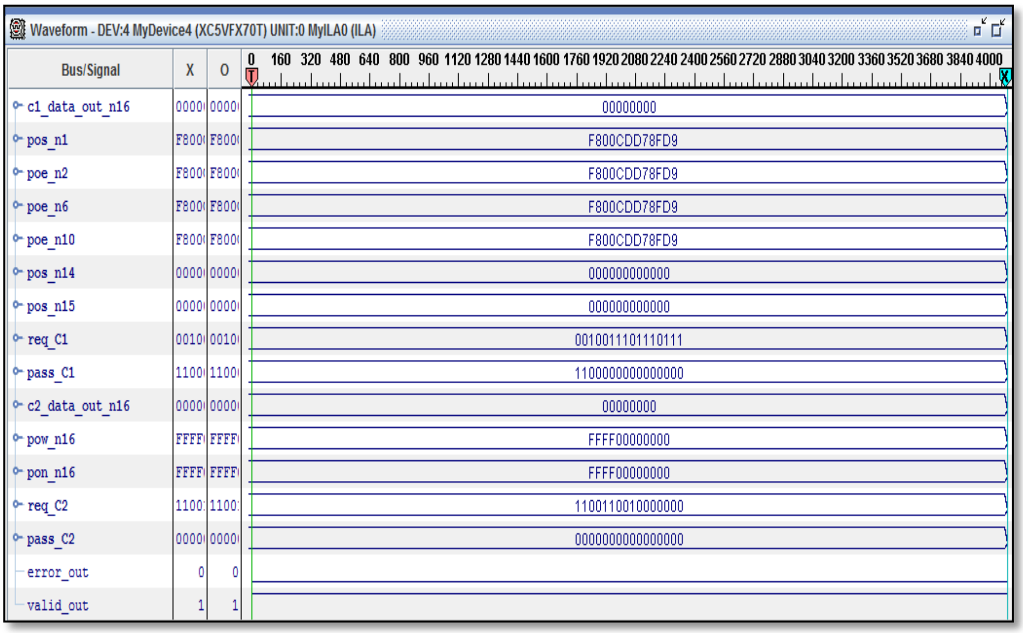


```
/n1/packet_in     f800cdd78fd9
/poe_n1           f800cdd78fd9
/pos_n1           f800cdd78fd9
/n2/pinn          f800cdd78fd9
/poe_n2           f800cdd78fd9
/pon_n2           f800cdd78fd9
/pos_n2           f800cdd78fd9
/n6/pinw          f800cdd78fd9
/poe_n6           f800cdd78fd9
/pow_n6           f800cdd78fd9
/pon_n6           f800cdd78fd9
/pos_n6           f800cdd78fd9
/n10/pinw         f800cdd78fd9
/poe_n10          f800cdd78fd9
/pow_n10          f800cdd78fd9
/pon_n10          f800cdd78fd9
/pos_n10          f800cdd78fd9
/n14/pinw         f800cdd78fd9
/pow_n14          f800cdd78fd9
/pon_n14          f800cdd78fd9
/pos_n14          f800cdd78fd9
/n15/pins         f800cdd78fd9
/pow_n15          000000000000
/pon_n15          000000000000
/pos_n15          000000000000
/n16/pinn         000000000000
_data_out_n16     00000000
/poe_n3           000000000000000000000000
/pon_n3           000000000000000000000000
```

Figure 7. Shows the simulated waveform

5. IMPLEMENTATION AND PERFORMANCE EVALUATION

The proposed secured agent based NoC using shortest path routing algorithm is designed using HDL code and simulated using Xilinx ISE 14.2 tool with modelsim 6.3f respectively.



Waveform - DEV:4 MyDevice4 (XC5VFX70T) UNIT:0 MyILA0 (ILA)

Bus/Signal	X	O	
c1_data_out_n16	0000	0000	00000000
pos_n1	F800	F800	F800CDD78FD9
poe_n2	F800	F800	F800CDD78FD9
poe_n6	F800	F800	F800CDD78FD9
poe_n10	F800	F800	F800CDD78FD9
pos_n14	0000	0000	000000000000
pos_n15	0000	0000	000000000000
req_C1	0010	0010	0010011101110111
pass_C1	1100	1100	1100000000000000
c2_data_out_n16	0000	0000	00000000
pow_n16	FFFF	FFFF	FFFF00000000
pon_n16	FFFF	FFFF	FFFF00000000
req_C2	1100	1100	1100110010000000
pass_C2	0000	0000	0000000000000000
error_out	0	0	
valid_out	1	1	

Figure 8. 4x4 NoC implementation result

The proposed design synthesized and implemented on VERTX – 5 FPGA (XC5VFX70T) Kit, which uses is 45nm technology with 100 MHz Operating frequency. Figure 8 shows the waveform recorded at the output of the FPGA kit.

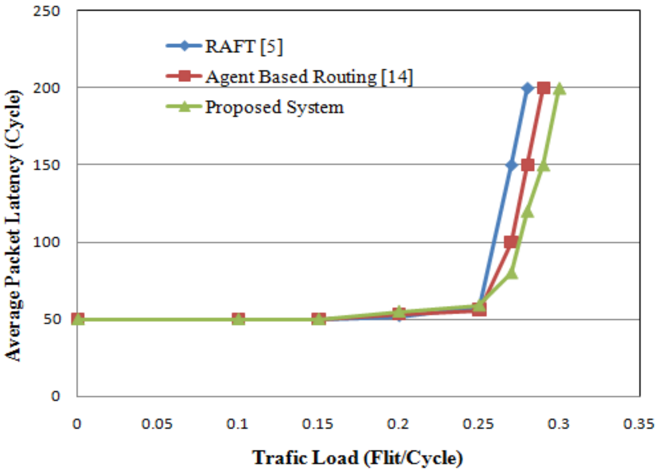


Figure 9. Average packet latency

The performance of the proposed system is compared with the existing method. Figure 9 shows the comparison of average packet latency with existing system. In [5] it does not have any means to reliably send all the packets to their destination in the faulty situation, there may be chance of packet entering in to deadlock in the faulty node. Then packet will be resent from the top level, which leads to performance degradation by increasing latency. The proposed system has prior knowledge of all the faulty links and nodes. Hence the packet will reach the healthy node with a reliable time as compare to existing agent based system. The proposed system graph has high performance with saturation point compare to existing system [14]. In the graph, proposed method saturation point is high i.e packet injection rate will be more and it will reach the destination with a less number of clock cycles. From the graph existing method input packet data rate lies between 0.25 and 0.3 by taking 100 clock cycles. The proposed method takes 77 clock cycles to achieve the same data rate. Hence in the latency graph the line one which has higher saturation point with a wider slanting have better performance hence proposed system have higher performance compare to [14].

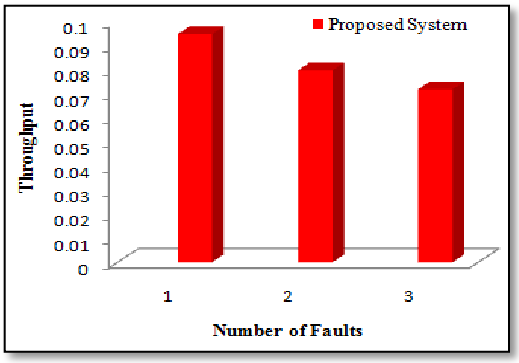
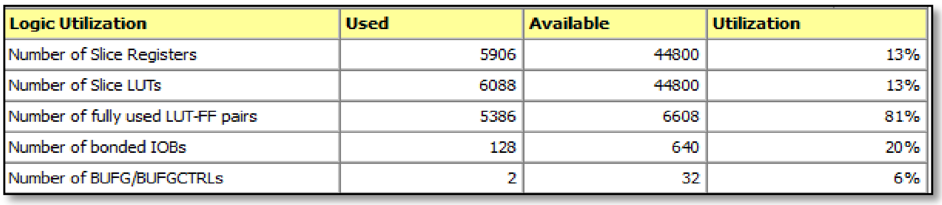


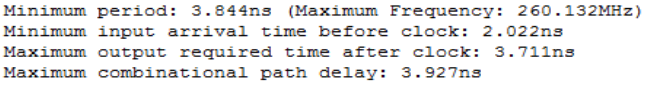
Figure 10. Throughput of the proposed system

Figure 10 shows the throughput of the proposed system. The proposed system takes care of receiving error free packets from the platform level itself hence packet remapping can be avoided and also proposed system have prior knowledge of all faulty links and nodes which will give higher throughput under different fault conditions and in the existing agent based system [14] author haven't discussed on throughput. Table 1 shows the proposed system hardware utilization summary with gate count and the simulated operating frequency was found to be 260.132 MHz with 3.927 delay path in the operation. The value of the set-up time was observed as 2.022ns whereas that of the hold-time was 3.711ns. Both set-up time and hold-time are slack values and do not result in time violation.

Table 1. Proposed design area utilization summary



Logic Utilization	Used	Available	Utilization
Number of Slice Registers	5906	44800	13%
Number of Slice LUTs	6088	44800	13%
Number of fully used LUT-FF pairs	5386	6608	81%
Number of bonded IOBs	128	640	20%
Number of BUFG/BUFGCTRLs	2	32	6%



```
Minimum period: 3.844ns (Maximum Frequency: 260.132MHz)
Minimum input arrival time before clock: 2.022ns
Maximum output required time after clock: 3.711ns
Maximum combinational path delay: 3.927ns
```

The proposed agent based NoC system using shortest-path-Algorithm is compared with previous similar NoC systems like RAFT [5] and traditional agent based NoC [14] in Table 2 to validate hardware utilization improvements by 51.82 %, 55.50% and 55.50 % respectively. In the Hierarchical cell agent using shortest-path, the routing process is controlled using Shortest Path Data Pass (SPDP) unit, which generates the pass signal to each node that allows the shortest route to be followed. In routing, based on Pass signal, the packet will traverse in east, west, south and north direction along with local output. In this module there is no arbitration and packet will not traverse as per XY algorithm [14]. Because of these reasons, Hierarchical Cell Agent using shortest-path consumes lesser area than Hierarchical Cell Agent using XY algorithm.

Table 2. Agent based NoC system using shortest-path-Algorithm is compared with previous similar NoC systems

Routing Method	Area utilization (Gate Count) for 5 Port	Area Overhead Comparison (%) of RAFT[5]	Area Overhead Comparison (%) of Agent based NoC [14]
RAFT[5]	39355	NA	NA
Agent-based Routing[11]	41574	5.6	NA
Proposed agent based NOC system using shortest-path-Algorithm	17510	No Area Overhead	No Area Overhead
Proposed agent based NOC system using shortest-path-Algorithm Area reduced % compare to exisiting NoC system.		55.50%	55.50%

6. CONCLUSION

The proposed secured agent based on chip system design using shortest path algorithm have hierarchical Agents. These agents secure the node by providing security and healthy status of the node to upper layer. This healthy status information helps the upper layer to map the packet into healthy node. This improves the performance by avoiding packet remapping and reduces the packet latency against faulty links and nodes. In addition to this, the agent will provide the security to PE which avoids the unrelated and unwanted packet for the dedicated PE. This intern overcomes the live lock situation of the high priority packet. From the simulation and synthesized result, the proposed design provides better performance compare to existing method using shortest path routing algorithm with less hardware overhead.

REFERENCES

[1] M. Valinataj, S. Mohammadi, and S. Safari, “Fault-aware and reconfigurable routing algorithms for Networks-on-Chip,” *IETE Journal of Research*, vol. 57, no. 3, pp. 215–223, 2011.

[2] T. Lehtonen, D. Wolpert, P. Liljeberg, J. Plosila, and P. Ampadu, “Self-adaptive system for addressing permanent errors in on-chip interconnects,” *IEEE Trans. Very Large Scale Integr. Syst*., vol. 18, no. 4, pp. 527–540, Apr. 2010.

[3] M. Kakoee, V. Bertacco, and L. Benini, “At-speed distributed functional testing to detect logic and delay faults in NoCs,” *IEEE Trans. Comput.*, vol. 63, no. 3, pp. 703–717, Mar. 2014.

[4] G. Schley, N. Batzolis, and M. Radetzki, “Fault localizing end-toend flow control protocol for networks-on-chip,” in *Proc. 21st Euromicro Int. Conf. Parallel Distrib. Netw.-Based Process*., 2013,pp. 454–461.

[5] M. Valinataj, S. Mohammadi, J. Plosila, P. Liljeberg, and H. Tenhunen, “A reconfigurable and adaptive routing method for fault-tolerant meshbased networks-on-chip,” *Elsevier, Int. J. Electronics and Communications (AEÜ)*, vol. 65, no. 7, pp. 630–640, 2011.

[6] C. Feng, Z. Lu, A. Jantsch, J. Li, and M. Zhang, “FoN: Fault-on-Neighbor aware routing algorithm for Networks-on-Chip,” *Proc. 23th IEEE Int. System-on-Chip Conf. (SOCC)*, pp. 441–446, 2010.

[7] Chen, Yu-Yin, En-Jui Chang, Hsien-Kai Hsin, Kun-Chih Chen, and An-Yeu Wu, “Path-Diversity-Aware Fault-Tolerant Routing Algorithm for Network-on-Chip Systems,” *IEEE Transactions on Parallel and Distributed Systems*, Volume 28 Issue 3, March 2017.

[8] O. Cesariow *et al.*, “Multiprocessor SoC platforms: a component-based design approach,” *IEEE Design and Test of Computers*, vol. 19, no. 6, pp. 52–63, 2002.

[9] A. Kohler, G. Schley, and M. Radetzki, “Fault tolerant network on chip switching with graceful performance degradation,” *IEEE Trans. On Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no.6, 2010.

[10] P. Rantala, J. Isoaho, and H. Tenhunen, “Novel agent-based management for fault-tolerance in network-on-chip,” *Proc. 10th Euromicro Conf. on Digital System Design (DSD)*, pp. 551–555, 2007.

[11] L. Guang, E. Nigussie, P. Rantala, J. Isoaho, and H. Tenhunen, “Hierarchical agent monitoring design approach towards self-aware parallel systems-on-chip,” *ACM Trans. on Embedded Computing Systems*, vol. 9, no. 3, article 25, 2010.

[12] A. W. Yin et al, “Hierarchical agent monitoring NoCs: a design methodology with scalability and variability,” *Proc. 26th NORCHIP Conf.*, pp. 202–207, 2008.

[13] L. Guang, B. Yang, J. Plosila, K. Latif, and H. Tenhunen, “Hierarchical power monitoring on NoC - a case study for hierarchical agent monitoring design approach,” *Proc. 28th NORCHIP Conf.*, 2010.

[14] M.Valinataj, P.Liljeberg, J.Plosila,“ Reliable On-Chip Network Design Using an Agent-based Management Method”, *19th International Conference on "Mixed Design of Integrated Circuits and Systems"*, May 24-26, 2012.

[15] M. Li, Q. Zeng, and W. Jone, “DyXY- a proximity congestion-aware deadlock-free dynamic routing method for Network on Chip, ” *Proc. 43th Design Automation Conference (DAC)*, pp. 849–852, 2006.

[16] En-Jui Chang, Hsien-Kai Hsin, Shu-Yen Lin, and An-Yeu Wu. Path-congestionaware adaptive routing with a contention prediction scheme for network-onchip systems. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions* on, 33(1):113–126, 2014.

[17] Feiyang Liu, Huaxi Gu, and Yintang Yang, “Dtbr: A dynamic thermal-balance routing algorithm for network-on-chip,” *Computers & Electrical Engineering*, 38(2):270–281, 2012.

[18] Yeong Seob Jeong and Seung Eun Lee, “Deadlock-free xy-yx router for on-chip interconnection network,” *IEICE Electronics Express*, 10(20):20130699–20130699,2013.

[19] [20] G. Siva Nageswara Ra, N. Srinivasu, S.V.N. Srinivasu3, G. Rama Koteswara Rao, “Dynamic Time Slice Calculation for Round Robin Process Scheduling Using NOC,” *International Journal of Electrical and Computer Engineering (IJECE)*, Vol. 5, No. 6, pp. 1480~1485, 2015.

[20] Adam Hendra Brata, Deron Liang, and Sholeh Hadi Pramono, et al, “Software Development of Automatic Data Collector for Bus Route Planning System,” *International Journal of Electrical and Computer Engineering (IJECE)*, Vol. 5, No. 1, pp. 150-157, 2015.

[21] A. Kohler, G. Schley, and M. Radetzki, “Fault tolerant network on chip switching with graceful performance degradation,” *IEEE Trans. On Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no.6, 2010.

[22] Kendaganna Swamy S, Anil N, A. Jatti and Uma B V, "Platform level design for Network on Chips," *2015 IEEE International Advance Computing Conference (IACC)*, Banglore, pp. 16-19 2015.

[23] S. K. Swamy, A. Jatti and B. V. Uma, “Random arbiter and platform level design for improving the performance on 4×4 NoC,” 2016 *International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, Chennai, pp. 978-983,2016.

[24] Anala M R, Amit N Subrahmanya, Allbright D’Souza, “Performance Analysis of Mesh-based NoC’s on Routing Algorithms,” *International Journal of Electrical and Computer Engineering (IJECE)*, Vol. 8, No. 5, 2018.

BIOGRAPHIES OF AUTHORS

Prof. Kendaganna Swamy. S, Assistant Professor Department of Electronics and Instrumentation engineering, R.V College of engineering, Bangalore. He is having 6yrs of teaching experience and 2 years of industry; Area of interest VLSI design, FPGA and NoC. He is published 19 papers along with one national level patent published.

Dr. Uma B.V., working as Professor & Head in Department of electronics and communication engineering, R.V College of engineering, Bangalore. She is having 25yrs of teaching experience, Area of interest VHDL, VLSI design, Digital Electronics Circuits, Synthesis and optimization of digital circuits, CAD tools for VLSI, CMOS VLSI design. She is published 48 papers.

Dr. Anand Jatti, working as an Associate Professor Department of electronics and instrumentation engineering, R.V College of engineering, Bangalore. He is having 14yrs of teaching experience; Area of interest image processing, signal processing and VLSI. He is published 23 papers.