ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3 , Issue 4 , April 2014

Addressing Fault in Noc With On-Line Diagnosis Mechanism

V.Dhivya[1], C. Sumathy[2], M. Saranya[3]

Student, Department of ECE, SyedAmmal Engineering College, Ramanathapuram, Tamil nadu , India[1,3]

Assistant Professor, Department of ECE, Syed AmmalEngineering College, Ramanathapuram, Tamil nadu, India[2]

Abstract:This project proposes a fault-tolerant solution for a buffer less network-on-chip, including an on-line fault-diagnosis mechanism to detect both transient and permanent faults, a hybrid automatic repeat request, and forward error correction link-level error control scheme to handle transient faults and a reinforcement-learning-based fault-tolerant deflection routing (FTDR) algorithm to tolerate permanent faults without deadlock and live lock. A hierarchical-routing-table-based algorithm (FTDR-H) is also presented to reduce the area overhead of the FTDR router. The Ant Colony Algorithm is used to find the minimum distance between sender to receiver. Modelsim simulator used to verify a existing and proposed method.

Keywords: Deflection Routing, Fault Tolerance, On-line Fault diagnosis, Permanent fault

I.INTRODUCTION

The design of a chip is based on four distinct aspects: computation, memory, communication and I/O. The increase of the processing power and the emergence of data intensive applications has attracted major attention on the challenge of the communication aspect in single-chip systems (SOC).Fault-tolerance or graceful degradation is the property that enables a system to continue operating properly in the event of the failure of (or one or more faults within) some of its components. If its operating quality decreases at all, the decrease is proportional to the severity of the failure, as compared to a naively-designed system in which even a small failure can cause total breakdown.Fault-tolerance is not just a property of individual machines; it may also characterize the rules by which they interact. For example, the Transmission Control Protocol (TCP) is designed to support reliable two-way communication in a packet-switched network, even in the presence of communications links which are imperfect or overloaded. Within the scope of an individual system, fault-tolerancecan be achieved by anticipating exceptional conditions and building the system to cope with them, and, in general, aiming for self-stabilization so that the system converges towards an error-free state.

communications links which are imperfect or overloaded. Within the scope of an individual system, fault-tolerancecan be achieved by anticipating exceptional conditions and building the system to cope with them, and, in general, aiming for self-stabilization so that the system converges towards an error-free state. Nowadays, the SOC design challenges concern at first the design complexity; the goals are the separation of computation from communication, and the use of structured communication means. It is also important to achieve design reliability in order to cope with process variability, and to guarantee resilience against soft and hard errors. Fault-Tolerant Routing Algorithms to Handle Permanent Faults for NOC Two kinds of fault-tolerant routing, which are known as stochastic and deterministic, have been proposed for NOC to handle permanent faults. Stochastic communication transfers redundant packets through different paths to avoid faults Depending on the shape of the fault region, deterministic fault-tolerant routing algorithms can be categorized intotwo classes: one can handle regular fault regions (e.g., convex and concave shapes) and the other, which is also known as topology-agnostic, can handle irregular fault regions.

II.FTDR ALGORITHM

In order to distinguish transient faults from permanent faults, the fault diagnosis process is shown in Fig. 3. The decoder will generate a syndrome, which contains the error information of the packet. If the decode detects a single-bit error in any one part of the

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3 , Issue 4 , April 2014

encoding packet, it will correct it no matter which kind of faults it is. If it detects a two-bit error in any one part of the encoding packet for one cycle, which is considered as a transient fault, it will require the upstream router to retransmit the packet. If the syndromes of two consecutive received packets are the same, which means the retransmitted packet contains the same two-bit error, in order to check whether the link contains real permanent faults,If all tests pass, the link will be enabled again. In the test mode, the other links of the upstream and downstream routers can still transmit packets as normal. From this test process, transient faults can be distinguished from permanent faults.

	North	East	South	West
Number of hops to R1	2	4	4	2
Number of hops to R2	1	3	3	3
Number of hops to R3	2	2	4	4
Number of hops to R4	3	3	3	1
Number of hops to R5	0	0	0	0
Number of hops to R6	3	1	3	3
Number of hops to R7	4	4	2	2
Number of hops to R8	3	3	1	3
Number of hops to R9	4	2	2	4

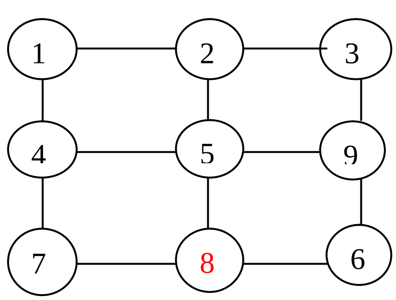

FIG1:ROUTING TABLE

The routing table update function is shown in Algorithm 1. If there is no fault in the network, the routing table cannot be updated. If one link of the router is broken or temporarily disabled during testing, all table entries corresponding to this direction are set to "∞" (steps 2–4). After a learning period, the table entries will converge to a fixed value which denotes the minimum number of hops from m each port to each destination. Additionally, we use the two-hop fault information to reduce the average hop counts. If a router detects that one of its neighborsy along direction d has only one link not faulty based on the two-hop fault information, the table entries from d to all destinations except y are set to "∞"

FIG 2 :UPDATE FUNCTION

Routing table update(des t_ ID,Q_ Value_ from, FON_ from)
For dir £ {North, East, South, West}
If link(dir) is faulty then
For i=1 to N and I ≠ Switch_ ID
Table _entry(i)(dir) –∞
If Neighbor(dir) has only one link not faulty then
For I =1toNand(I ≠ switch_ ID and I ≠ Neighbor_ ID(dir) then
Table entry (i)(dir)- ∞
If j £ {North, East ,South, West}
If FON_ from (dir)(j)=1 then
for n £ {all switches along j through Neighbor(dir)}
Table_ entry(n)(dir)=table_ entry(n)(dir)+2
If link(n)(dir)from disable to enable then
For I =1 to N and I ≠ switch_ ID
Table_ entry(des t _ ID)(dir)-q_ value_ from(dir, des t_ ID)

The pseudo code of the algorithm is shown in Algorithm 2. There are at most four packets reaching a router at the same time. The router makes routing decision from the highest priority packet to the lowest. The router first calculates the destination ID of the packet and looks up the routing table to check if the packet has reached the destination (steps 1–4). If the packet has not reached the destination, the router looks up the productive direction(s) with the minimum number of hops to destination from the routing table and then chooses a free productive port with the smallest stress value to route the packet (steps 7–9). If there is no free productive port, the router chooses a free port with the smallest stress value to route the packet, which can balance the network traffic loads(steps 10 and 11).

III.ANT COLONY OPTIMIZATION

The ant colony optimization algorithm (ACO) is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs. This algorithm is a member of the ant colony algorithms family, in swarm intelligence methods, and it constitutes some optimizations. the first algorithm was aiming to search for an optimal path in a graph, based on the behavior of an ants seeking path between their colony and a source of food. The original idea has since diversified to solve a wider class of numerical problems, and as a result, several problems have emerged, drawing on various aspects of the behavior of ants. ACS was the first algorithm inspired by real ant's behavior. The merit is used to introduce the ACO algorithms and to show the potentiality of using artificial pheromone and artificial ants to drive the search of always better solutions for complex optimization problems. In ACS once all ants have computed their tour (i.e. at the end of each iteration) AS updates the pheromone trail using all the solutions produced by the ant colony. Each edge belonging to one of the computed solutions is modified by an amount of pheromone proportional to its solution value. At the end of this phase the pheromone of the entire system evaporates and the process of construction and update is iterated. On the contrary, in ACS only the best solution computed since the beginning of the computation is used to globally update the pheromone. As was the case in AS, global updating is intended to increase the attractiveness of promising route but ACS mechanism is more effective since it avoids long convergence time by directly concentrate the search in a neighborhoods of the best tour found up to the current iteration of the algorithm.

IV.EXPERIMENTAL RESULTS

The proposed router system, with ANT COLONY algorithm are simulated by using Xilinx ISE 12.1i and implemented in spanten FPGA processor.

s.no	parameter	used
1	Number of Slices	28
2	Number of 4 input LUTs	50
3	Number of bonded IOBs	10

V.CONCLUSION

In this paper, we provided a fault-tolerant solution for a buffer less NOC to protect it from both transient and permanent faults on the links based on Ant Colony Optimization based routing algorithm was implemented The experimental results showed that FTDR and FTDR-H routers are high-reliability buffer less routers, which can protect against any fault distribution pattern, as long as the network is not cut into two or more disconnected sub-networks.

REFERENCES

[1] W. J. Dally and B. Towles, "Route packets, not wires: On-chip interconnection networks," in *Proc. 38th Annu. Design Autom. Conf.*, 2001,pp. 684–689.

[2] C. Constantinescu, "Trends and challenges in VLSI circuit reliability,"*IEEE Micro*, vol. 23, no. 4, pp. 14–19, Jul.–Aug. 2003.

[3] Y. H. Kang, T.-J. Kwon, and J. Draper, "Fault-tolerant flow control in on-chip networks," in *Proc. 4th ACM/IEEE Int. Netw.-Chip Symp.*, May 2010, pp. 79–86.

[4] S. Murali, T. Theocharides, N. Vijaykrishnan, M. J. Irwin, L. Benini, and G. De Micheli, "Analysis of error recovery schemes for networks on chips," *IEEE Design Test Comput.*, vol. 22, no. 5, pp. 434–442, Sep.– Oct. 2005.

[5] S. Pasricha, Y. Zou, D. Connors, and H. J. Siegel, "OE+IOE: A novel turn model based fault tolerant routing scheme for networks-on-chip," in *Proc. 8th IEEE/ACM/IFIP Int. Conf. Hardw./Softw. Codesign Syst. Synth.*, Oct. 2010, pp. 85–94.

[6] A. Patooghy and S. G. Miremadi, "XYX: A power & performance efficient fault-tolerant routing algorithm for network on chip," in *Proc.17th Euromicro Int. Parallel, Distrib. Netw.-Based Process. Conf.*, 2009,pp. 245–251.

[7] Z. Lu, M. Zhong, and A. Jantsch, "Evaluation of on-chip networks using deflection routing," in *Proc. 16th ACM Great Lakes Symp. VLSI*, 2006, pp. 363–368.

[8] T. Moscibroda and O. Mutlu, "A case for bufferless routing in on-chip networks," in *Proc. 36th Annu. Int. Symp. Comput. Arch.*, 2009, pp. 196–207.

[9] M. Hayenga, N. E. Jerger, and M. Lipasti, "SCARAB: A single cycle adaptive routing and bufferless network," in *Proc. 42nd Annu.IEEE/ACM Int. Symp. Microarch.*, Dec. 2009, pp. 244–254.

[10] C. Feng, Z. Lu, A. Jantsch, J. Li, and M. Zhang, "A reconfigurable faulttolerant deflection routing algorithm based on reinforcement learning for network-on-chip," in *Proc. 3rd Int. Workshop Netw. Chip Arch.*, 2010, pp. 11–16.

[11] H. Zimmer and A. Jantsch, "A fault model notation and error-control scheme for switch-to-switch buses in a network-on-chip," in *Proc. 1st IEEE/ACM/IFIP Int. Conf. Hardw./Softw. Codesign Syst. Synth.*, Oct. 2003, pp. 188–193.

[12] C. Feng, Z. Lu, A. Jantsch, J. Li, and M. Zhang, "FoN: Fault-onneighbor aware routing algorithm for networks-on-chip," in *Proc. 23rd IEEE Int. SoC Conf.*, Sep. 2010, pp. 441–446.

[13] A. Kohler, G. Schley, and M. Radetzki, "Fault tolerant network on chip switching with graceful performance degradation," *IEEE Trans.Comput.-Aided Design Integ*