

ADAPTIVE WEIGHTED ON DEMAND VECTOR ROUTING PROTOCOL IN FAULT TOLERANT NOC

¹NAIR RAJESH RAJSEKARAN,²A.BALAKUMARAN

¹P.G Scholar Department of ECE, ²Assistant Professor Department of ECE, Joe Suresh Engineering College
Tirunelveli,India

Abstract—Unicast routing, which discovers a path between a source-destination pair, is a challenging problem. In this paper, we propose a method to improve fault tolerance and load balancing using weighted on demand vector routing algorithm to which discovers its route for sending packets based on the stability of the path. The proposed algorithm is presented to reduce the energy consumption and also to increase its efficiency. The aim of the experiments is to show that networks can be trained in advance to be fault tolerant; to be able to compensate for hardware errors after being downloaded onto a hardware platform. Using simulation we show that our protocol is smart enough to cope with the network. For transient faults, the performance of the weighted on demand vector routing can achieve graceful degradation even at a high fault rate.

Index Terms—Deflection routing, fault-tolerance, on-line fault diagnosis, transient fault.

I. INTRODUCTION

NETWORK-ON-CHIP (NoC) approach has emerged as a promising solution for on-chip communications to enable integrating various processors and on-chip memories into a single chip [1]. As the CMOS technology scales down to the nanometer domain, smaller feature size, lower voltages and higher frequencies increase the number of occurrence of intermittent and transient faults, besides manufacturing defects and wear out effects which lead to permanent faults are also inevitable [2]. As power/energy consumption has already become a limiting constraint in the design of high-performance processors [26] and future on-chip networks in many-core processors are estimated to consume hundreds of watts of power [27], simple energy- and area-efficient interconnection network designs are especially desirable. The inherent structure redundancy of NoC provides the potential to design a fault-tolerant routing algorithm to enhance the reliability. NoC designs often considered Buffers in the buffers consume significant energy/power: dynamic energy when read/written and static energy even when they are not occupied. Second, having buffers increases the complexity. Third, buffers can consume significant chip area. Routing without buffers significantly reduces the energy consumption of the on-chip cache/processor-to-cache network, while providing similar performance to that of existing buffered routing algorithms at low network utilization (i.e., on most real applications). Recently, bufferless router has been studied in NoC to achieve higher speed and lower cost than a wormhole or virtual channel router [7]–[9]. The fully adaptive feature of deflection routing provides the potential to route packets to avoid faulty links/routers

and achieve fault tolerance. A reconfigurable fault-tolerant deflection routing (FTDR) algorithm based on reinforcement learning has been proposed for 2-D mesh NoC [10]. The advantage of the FTDR algorithm is the topology-agnostic feature, which is insensitive to the shape of the faulty region. The routing table can be reconfigured during packets transmission. A fault-tolerant solution, including an on-line fault diagnosis mechanism, a link-level error control scheme, and a fault-tolerant routing algorithm, is proposed for the bufferless NoC to tolerate both transient and permanent faults. The fault diagnosis mechanism uses the single-error-correcting and double-error-detecting (SECDED) Hamming code [11] to detect both transient and permanent link faults. The basic idea of “bufferless routing” is to always route a packet (or a flit) to an output port regardless of whether or not that output port results in the lowest distance to the destination of the packet. In other words, packets are deflected or “misrouted” by the router to a different output port if an output port that reduces the distance to the destination node is not available. Bufferless routing has also been called “hot-potato” routing alluding to the scenario that the router immediately needs to pass the potato (i.e. the packet) on to some other router as the potato is too hot to keep (i.e. buffer). With the use of bufferless method it happens that the route adopted may sometimes be longer than the actual easy and reliable one.

This would eventually affect the total time delay and Design. Buffering has several disadvantages. First, this problem in Bufferless NoC Design. The rest of paper is organized as follows: Related work is reviewed in Section II. Section III describes the NoC architecture and fault model. The detailed routing Algorithm and implementation are proposed in section IV. In section V, simulation experimental

results are analyzed, followed by the conclusion and future work in section VI.

II. RELATED WORKS

A. 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. Dumitras et al. have proposed a probabilistic broadcast scheme that is derived from the randomized gossip protocol. In order to limit the number of packet copies, redundant random walk only replicates packets at the source node. Although the stochastic communication can be highly fault-tolerant, it wastes much bandwidth to transfer redundant packets, which reduces the throughput of the network significantly. A reconfigurable routing algorithm is proposed to route packets surrounding a faulty router in a 2-D mesh NoC

The difference from the ordinary 2-D mesh is that the boundary output is connected to the input of the same router. This can be viewed as an additional packet buffer. All incoming packets are prioritized according to their hop counts, which record the number of hops the packet has been routed. The router makes routing decision for each arriving packet from the highest priority to the lowest. If a desired output port has already been occupied by a higher priority packet, a free port with the smallest stress value will be chosen, which means the packet has to be deflected

For deflection routing, a fault adaptive routing algorithm, based on a cost function to make routing decision, has been proposed in [13]. The cost function takes the route length and local fault status into consideration. Because the routing decision is only based on the fault information of the current router, the hop count field of the packet can easily overflow even in some simple fault patterns, which can lead to livelock.

A FoN aware deflection router [12], which can tolerate convex and concave fault regions without deadlock and power consumption. We propose a method to overcome livelock.

B. A Case for Bufferless Routing in On-Chip Networks No Buffers: it helps reduce both complexity and, as we show in our evaluation, energy consumption. Simplicity and Router Latency Reduction: Buffered routing algorithms employ virtual channels to improve buffer performance and flow control mechanisms (e.g. credit based flow control) to control buffer management in virtual channels. Since bufferless routing eliminates buffers, there are no virtual channels and there is no need to manage buffer allocation/deallocation. This reduces complexity in the router and can enable router latency

Area saving: , removing buffers from routers can result in significant area savings.

Absence of Deadlocks: Deflection-based bufferless routing is free of deadlock. Since the number of input and output ports are the same, every packet that enters a router is guaranteed to leave it.

Absence of Livelock: One of the potential challenges in bufferless routing is livelocks that could arise if a packet continuously gets deflected

C. BiNoC: A Bidirectional NoC Architecture with Dynamic Self-Reconfigurable Channel BiNoC architecture can significantly reduce the packet delivery latency at all levels of packet injection rates.

Compared to the conventional NoC, the bandwidth utilization of our BiNoC also exhibits higher efficiency. The BiNoC allows each communication channel to be dynamically self-configured to transmit flits in either direction in order to better utilize on-chip hardware resources. This added flexibility promises better bandwidth utilization, lower packet delivery latency, and higher packet consumption rate at each on-chip router

III. NOC ARCHITECTURE

The NoC architecture is based on a 2-D mesh topology, Nostrum NoC [17]. Each processing element is attached

to a router (R), as shown in Fig. 1.

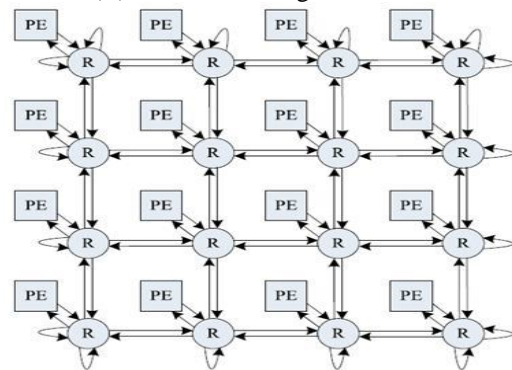


Fig. 1 NoC architecture

Each processing element is attached to a router (R). The boundary output is connected to the input of the same router. All incoming packets are prioritized according to their hop counts, which record the number of hops the packet has been routed. The router makes routing decision for each arriving packet from the highest priority to the lowest. If a desired output port has already been occupied by a higher priority packet, a free port with the smallest stress value will be chosen, which means the packet has to be deflected. The stress value, which can be used to balance traffic load, is the number of packets processed by neighboring routers in the last four cycles.

IV ALGORITHM AND IMPLEMENTATION

a) Forward error correction

The FEC is used to perform error control to tolerate transient faults during packets transmission. In the case of a single-bit error in any part of the packet, the error can be corrected after the packet has been decoded. If any part of the packet contains a two-bit error for one cycle, the router which receives the packet will require the router, which sends the packet, to retransmit the packet.

b).Q-learning (FTDR-H)

Q-routing or learning is a table-based routing algorithm. The 2-D routing table stores the lowest estimated delivery time to each router in the network from each output of the current router. A hierarchical routing- table-based deflection routing algorithm is used to reduce the table size.

The $n \times n$ mesh can be divided into several subregions with equal size. Each router contains a local and a region routing table. The local routing table stores the minimum number of hops from all ports of the current router to other routers in the same region, while the region routing table stores the minimum number of hops from all ports of the current router to the nearest boundary routers in other regions.

When a packet reaches a router, the router first checks whether the destination is in the same region as the current router or not. If it is, the router makes routing decision according to the local routing table. If the destination is not in the same region, the router makes routing decision according to the region routing table.

The local and region routing tables are also updated

c) Dijkstra's algorithm

Dijkstra's algorithm is a graph search algorithm that solves the single-source shortest path problem for a graph with nonnegative edge path costs, producing a shortest path tree. This algorithm is often used in routing and as a subroutine in other graph algorithms. For a given source vertex (node) in the graph, the algorithm finds the path with lowest cost (i.e. the shortest path) between that vertex and every other vertex. It can also be used for finding costs of shortest paths from a single vertex to a single destination vertex by stopping the algorithm once the shortest path to the destination vertex has been determined.

for all v in G :

$d(v) = \text{infinity}$;

$D = \{s\}$;

$U = \{G-s\}$; // all other nodes in G

for all u adjacent to s :

$d(u) = w(u, s)$;

while U is not empty {

Let v be the node from U with minimal $d(v)$;/*

parallelization takes

place here */

$U = U / v$;

$D = D \cup v$;

for all u adjacent to v { /* **another possible place for**

parallel */

if $d(u) > w(u,v) + d(v)$ then

$d(u) = w(u,v) + d(v)$;

} }

d)Bi-Directional Search Algorithm

A technique that has proved useful in shortest path and other discrete optimization computations has been bidirectional search. The method has been well tested in the two-node shortest-path problem providing substantial computational savings. A natural impulse is directional algorithms, the search proceeds from an initial node forward until the goal node is encountered Problems for which the goal node is explicitly known can be searched backward from the goal node. An algorithm combining both search directions is bidirectional.

V. EXPERIMENTAL RESULTS

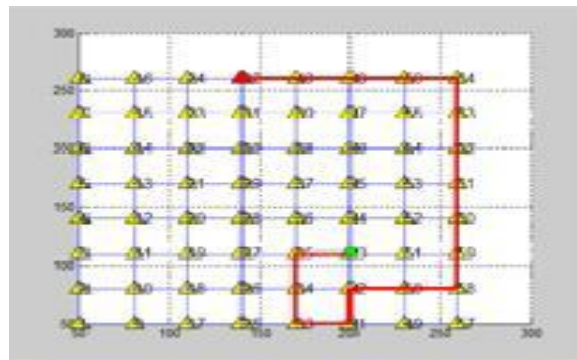


Fig.2 8*8 NoC and the Fault Tolerant Routing path with source and destination using existing method.

The above figure (2) shows the path routing from the source to the destination as per the fault tolerant route available. Thus making the availability fault tolerant, but as can be seen the path even if fault tolerant may take more delay and time..

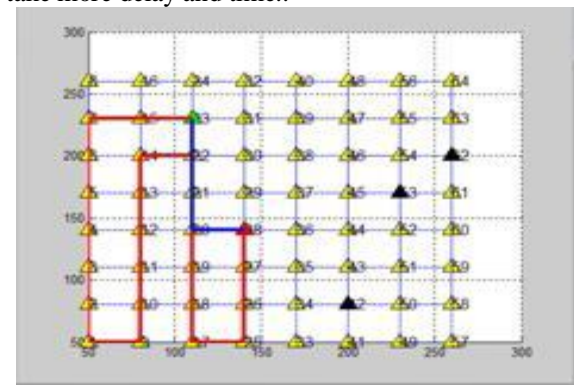


Fig.3 8*8 NoC and the Fault Tolerant Routing path with source and destination avoiding the faulty link using Bidirectional

Routing (blue line).to extend its benefits to heuristicsearch. In the uni-.

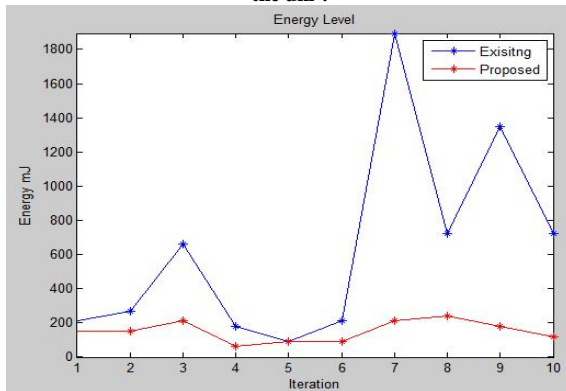


Fig.4. Energy Response

As shown in the figure 3 , the proposed method provides fault tolerant path with minimum delay and time. The figure 4 shows the measure of approximated amount of power saved in the proposed method as a result quick path routing and maintaining the link due to bidirectional routing ..

The figure 5 shows the responses of the amount of delay taken by the Existing method to the proposed one . The delay is being reduced and as a result the routing of the path can be made easily and speedily with minimum delay and power consumption

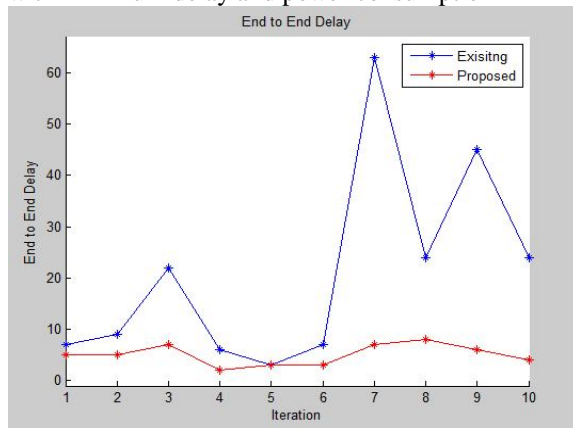


Fig.5. Time Delay Response

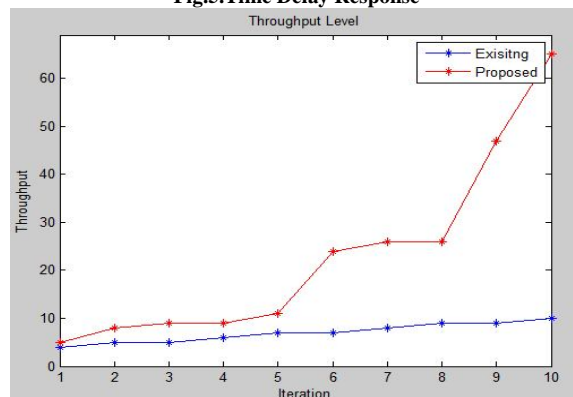


Fig.6. Throughput Response

CONCLUSION

In this project, a fault-tolerant solution for a bufferless NoC is provided to protect from both

transient and permanent faults. a fault-tolerant solution for a bufferless network-on-chip, a reinforcement-learningbased routing algorithm to tolerate permanent faults without dead-lock and livelock. Using our proposed method we have constructed a shortest restoration path for link fault.To reduce the area overhead of the router a hierarchical-routing-table-based algorithm (FTDR-H) is presented. The experimental results show for a bufferless approach with bidirectional routing the delay time in establishing the link as well as the powerconsumption can be minimised. .

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– tolerant network on chip switching with graceful performance degradation," IEEE Trans.Comput.-Aided Design Integr. Circuits Syst., vol. 29, no. 6, pp. 883–896, Jun. 2010.
- [3] Y. H. Kang, T.-J. Kwon, and J. Draper, "Fault- 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 onchip 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-neighbor 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 Integr. Circuits Syst.*, vol. 29, no. 6, pp. 883–tolerant flow control in on-chip networks," in *Proc. 4th 896*, Jun. 2010.
- [14] D. Bertozzi, L. Benini, and G. De Micheli, "Error control schemes for on-chip communication links: The energy-reliability tradeoff," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 24, no. 6, pp. 818–831, Jun. 2005.
- [15] C. Grecu, A. Ivanov, R. Saleh, E. S. Sogomonyan, and P. P. Pande, "Online fault detection and location for NoC interconnects," in *Proc. 12th IEEE Int. On-Line Test. Symp.*, Jul. 2006, pp. 145–150.
- [16] R. Holsmark, M. Palesi, and S. Kumar, "Deadlock free routing algorithms for irregular mesh topology NoC systems with rectangular regions," *J. Syst. Arch.*, vol. 54, nos. 3–4, pp. 427–440, 2008.
- [17] E. Nilsson, M. Millberg, J. Oberg, and A. Jantsch, "Load distribution with the proximity congestion awareness in a network on chip," in *Proc. Design, Autom. Test Eur. Conf. Exhibit.*, 2003, pp. 1126–1127
- [18] Q. Yu and P. Ampadu, "Transient and permanent error co-management method for reliable networks-onchip," in *Proc. 4th ACM/IEEE Int.Netw.-Chip Symp.*, May 2010, pp. 145–154.
- [19] A. Ejlali, B. M. Al-Hashimi, P. Rosinger, and S. G. Miremadi, "Jointconsideration of fault-tolerance, energy-efficiency and performance in on-chip networks," in *Proc. Design, Autom. Test Eur. Conf. Exhibit.*, 2007, pp. 1–6.
- [20] M. Ali, M. Welzl, and S. Hessler, "An end-to-end reliability protocol toaddress transient faults in network on chips," in *Proc. Workshop Diag.Serv. Netw.-Chips*, 2007, pp. 376–380.
- [21] D. Park, C. Nicopoulos, J. Kim, N. Vijaykrishnan, and C. R. Das,"Exploring fault-tolerant network-on-chip architectures," in *Proc. Int.Conf. Dependable Syst. Netw.*, 2006, pp. 93–104.
- [22] T. Dumitras, S. Kerner, and R. Marculescu, "Toward on-chip fault-tolerant communication," in *Proc. Asia South Pacific Design Autom.Conf.*, 2003, pp. 225–232.
- [23] M. Pirretti, G. M. Link, R. R. Brooks, N. Vijaykrishnan, M. Kandemir, and M. J. Irwin, "Fault tolerant algorithms for network-on-chip interconnect," in *Proc. IEEE Comput. Soc. Annu. Symp.VLSI*, Feb. 2004, pp. 4
- [24] Z. Zhang, A. Greiner, and S. Taktak, "A reconfigurable routing algorithm for a fault-tolerant 2Dmesh network-on-chip," in *Proc. 45th ACM/IEEE Design Autom. Conf.*, Jun. 2008, pp. 441–446.
- [25] R. Holsmark, M. Palesi, and S. Kumar, "Deadlock free routing algo-rithms for irregular mesh topology NoC systems with rectangular regions," *J. Syst. Arch.*, vol. 54, nos. 3–4, pp. 427–440, 2008.
- [26] M. K. Gowan, L. Biro, and D. Jackson. Power considerations in thedesign of the Alpha 21264 microprocessor. In *DAC*, 1998.
- [27] S. Borkar. Thousand core chips: A technology perspective. In *DAC*, 2007

★ ★ ★