Fault-Resilient Routing Unit in NoCs

Xiaofan Zhang[1], Masoumeh Ebrahimi[2], Letian Huang[1], Guangjun Li[1]
[1] University of Electronic Science and Technology of China, China
[2]KTH Royal Institute of Technology, Sweden and University of Turku, Finland

***Abstract*—With aggressive technology scaling in deep submicron era, burgeoning transistors make chips more susceptible to failures. It is inevitable that process variation is gradually becoming a crucial challenge in the IC design. In addition, aging leads to faults, shortening the lifetime of the circuits. Networks-on-chip also come to the problems caused by variations and aging, leading to degraded performance and erroneous behaviors. Faults may occur in numerous locations of the on-chip networks and once they occur in the control path, more severe effects such as deadlock and livelock are expected. In this paper, we present a fine-grained mechanism to tolerate faults in the routing computation units without disabling the faulty routers. By applying this mechanism, routing and packet-receiving services are separated. The faulty routing computation unit is replaced by a light-weight redundant circuit, providing static but reliable routing services. The other components in this router are still functional retaining the on-chip performance. Experimental results indicate that the on-chip network with the proposed mechanism is fault-tolerant when 14% of all routing computation modules are suffering from faults. The area overhead and power consumption of the proposed method is around 7.29% and 6.20% over the baseline approach.**

Keywords-Network-On-Chip; Fault Tolerance; Redundant Routing Unit Design

I. INTRODUCTION

With the characteristic of distributed and shared computing resource, Network-on-Chip has shown its remarkable advantages of performance and scalability in on-chip communication [1]. As feature sizes of integrated circuits decrease aggressively, chips are getting more susceptible to faults [2][3]. It is inevitable that on-chip network will also suffer from the same problems. Among the others, faults may occur because of Process Variation (PV) as a result of manufacturing imperfections [4] or time-dependent variation such as the Negative Bias Temperature Instability (NBTI) [5][6]. Therefore, fault-tolerance in NoC is more than a feature but a necessity.

Generally, a router in NoCs consists of five stages as routing computation (RC), virtual channel allocation (VA), switch allocation (SA), crossbar (Xbar), and link traversal (LT) phases. The router architecture in this paper utilizes one RC unit per input port similar to the designs in [7].

As shown in Fig. 1, components inside the router can be divided into two parts, the control path and the data path [7]. The RC, VA and SA units are part of the control path by directly affecting the routing decision while other parts of the router are categorized as the units in the data path, determining the path taken by the packets to pass through the router. When faults occur in the data path, information carried by a packet may be corrupted during the transmission. To protect the data path against faults, Error Detecting Code (EDC) and Error Correcting Code (ECC) can be designed to detect and to correct the faults in the packet [8][9]. Also the data redundancy mechanism, such as retransmission, can also be applied [10]. In contrast, faults in the control path of NoC routers are hard to be detected and even harder to be tolerated. Control path faults are such as those affecting the internal logics of the routing unit leading to miscalculations of the routing path or wrong matching pairs between the input and output port in the crossbar units [11]. When the control path faults are introduced, packets may be forwarded to wrong directions and eventually may lead to the deadlock or livelock.

In this paper, we propose a fault-resilient routing unit for NoCs which brings redundancy to the traditional routing computation unit. The default routing unit supports a fully adaptive routing algorithm while the redundant unit is a very light-weight unit capable of delivering packets through limited paths.

This paper is organized as follows. Section II reviews the related work. In Section III, the router architecture is introduced and the proposed mechanism is presented. The experimental results are reported in Section V while the summary and conclusion are given in the last section.

II. RELATED WORK

It is necessary to consider a fault-tolerant design for NoCs to better satisfy the need of future on-chip interconnection while facing the continued shrinkage of the semiconductor process feature sizes and the resulting unreliable integrated circuits [12].

The PV-induced effects on the major components of the NoC architecture are analysed in [13] where the authors proposed an improved NoC router architecture with a high PV resilience. Effects of aging and wear-out mechanisms on the NoC based routers and links are investigated in various levels such as the circuit and system level.

An investigation on the multiple aging degradation mechanism on routers and links is performed in [14]. The ReliNoC architecture is proposed in [15] which utilizes the redundant components as the replacements of the faulty ones. In [16], Vicis is presented, combining different schemes as built-in-self-test (BIST), Error Correcting Code (ECC), port-swapping mechanism and a crossbar bypass bus, aiming at tolerating wear-out induced faults.

Beside the well-design router architecture, current efforts in literature also focus on the robust routing algorithms. Aging-aware routing algorithms are proposed in [17] and [18] with the goal of reducing the influence of NBTI. In addition, several minimal and non-minimal approaches have been previously investigated to tolerate a number of faults in an on-chip network [19][20][21][22].

978-1-4673-9094-1/15/$31.00 ©2015 IEEE

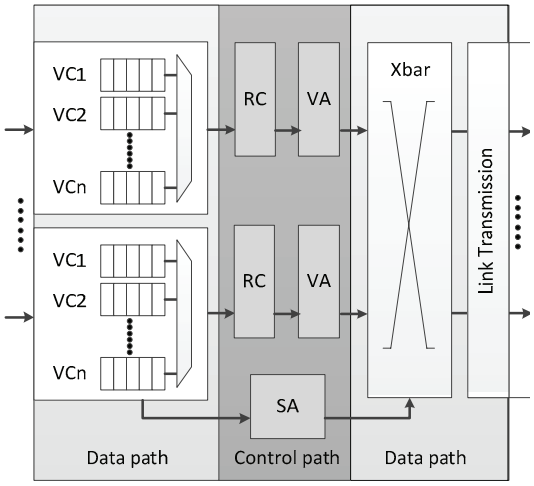


Fig. 1 Default router architecture

These methods are mostly based on disconnecting both the faulty router and the core from the network and routing packets around the faulty areas. However, this is an unrealistic assumption that the entire router should be disabled in the case of a single fault. This assumption may leave a severe impact on the functionality of the entire system (i.e. by disconnecting a core) or the performance (i.e. traffic highly increases around the faulty area).

In this paper, we aim at tolerating faults in the routing computation unit without disabling the router. To achieve this goal, a simple redundant routing unit is designed to replace the original one when faults occur and the additional packet receivers are aiming at providing reliable receiving services. With the proposed mechanism, faults in the routing computation unit will not lead to packet misrouting and thus avoiding the consequences of threatening the whole network.

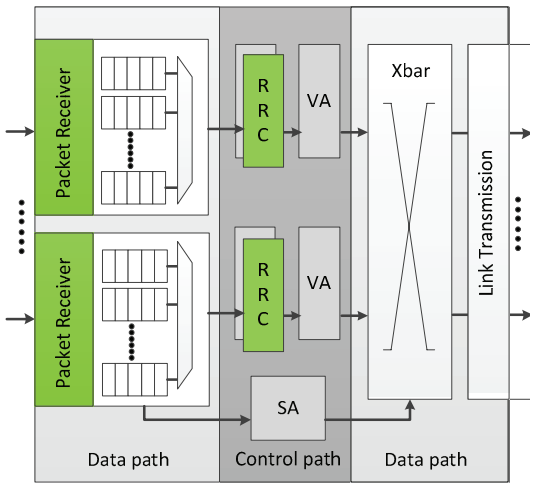


Fig. 2 The proposed router architecture

III. THE PROPOSED MECHANISM

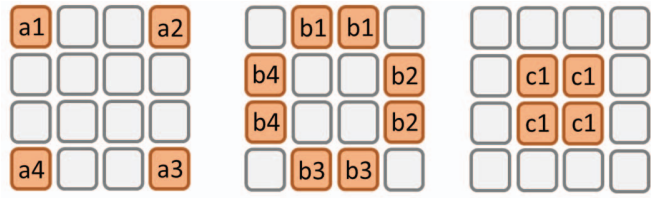
A. The Router Architecture

The proposed router architecture is shown in Fig. 2. In this new architecture, each routing unit is equipped with a redundant routing computation unit (RRC) and each input port of the router is connected with a packet receiver. Once faults occur in the RC module, RRC is activated to route packets. RCC consists of a very small logic circuit with a static routing decision. The small logic of RCC leads to a low probability of faults as compare to the RC unit itself.

Packet receiver (PR) is designed to tackle the packets with the local core as the destination. Because of the miscalculation in the faulty RC module, packets destined the local core may never reach it and thus blocking other packets. In the proposed architecture, packets will be first processed by the PR to make sure that they can reach the local port prior making the calculation in the RC unit. In PR, the address information carried by the packet is checked and once there is a match, packet will be delivered to the local core.

B. The Basic Routing Algorithm

The non-minimal routing algorithm proposed in [23] is selected as the basic algorithm, running in the routing computation units of this paper. In the absence of faults, this algorithm only utilizes the minimal paths and acts similar to the DyXY routing algorithm, which is realized as a high-performance algorithm in NoC-based routing [24]. This non-minimal algorithm uses two virtual channels in the Y dimension and no virtual channel in the X dimension. Thereby,



(a) (b) (c)

Fig. 3 Routers highlighted in the corner (a), borderline (b), and central parts (c)

The Basic Routing Algorithm

```
IF Position = {L}
  THEN OutChannel(L) <= '1';
IF InChannel != {E} AND Position = {E, NE, or SE}
  THEN OutChannel(E) <= '1';
IF InChannel = {L, N1, S1, or E}
  THEN OutChannel(W) <= '1';
IF InChannel = {L, S1, or E}
  THEN OutChannel(N1) <= '1';
IF InChannel = {L, N1, E, or S1}
  THEN OutChannel(S1) <= '1';
IF InChannel != {N2} AND Position = {N, E, NE, or SE}
  THEN OutChannel(N2) <= '1';
IF InChannel != {S2} AND Position = {S, E, NE, or SE}
  THEN OutChannel(S2) <= '1';
```

Fig. 4 The basic routing algorithm

```
Redundant Routing Paths for the Corners
Router (a1)
  IF     InChannel = {L}  THEN OutChannel(E) <='1';
  ELSEIF InChannel = {E}  THEN OutChannel(S1)<='1';
  ELSEIF InChannel = {S1} THEN OutChannel(E) <='1';
  ELSEIF InChannel = {S2} THEN OutChannel(E) <='1';
Router (a2)
  IF     InChannel = {L}  THEN OutChannel(S1)<='1';
  ELSEIF InChannel = {W}  THEN OutChannel(S2)<='1';
  ELSEIF InChannel = {S1} THEN OutChannel(W) <='1';
  ELSEIF InChannel = {S2} THEN OutChannel(L) <='1';
Router (a3)
  IF     InChannel = {L}  THEN OutChannel(W) <='1';
  ELSEIF InChannel = {W}  THEN OutChannel(N2)<='1';
  ELSEIF InChannel = {N1} THEN OutChannel(W) <='1';
  ELSEIF InChannel = {N2} THEN OutChannel(L) <='1';
Router (a4)
  IF     InChannel = {L}  THEN OutChannel(N1)<='1';
  ELSEIF InChannel = {E}  THEN OutChannel(N1)<='1';
  ELSEIF InChannel = {N1} THEN OutChannel(E) <='1';
  ELSEIF InChannel = {N2} THEN OutChannel(E) <='1';
```

Fig. 5 The algorithm for the routers in corners

```
Redundant Routing Paths for Central Routers
  IF     InChannel = {L} THEN  OutChannel (E)<='1';
  ELSEIF InChannel = {E} THEN  OutChannel (W)<='1';
  ELSEIF InChannel = {W} THEN  OutChannel (E)<='1';
  ELSEIF InChannel = {N1} THEN OutChannel (S1)<='1';
  ELSEIF InChannel = {N2} THEN OutChannel (S2)<='1';
  ELSEIF InChannel = {S1} THEN OutChannel (W)<='1';
  ELSEIF InChannel = {S2} THEN OutChannel (N2)<='1';
```

Fig. 6 The algorithm for the routers in central parts

there are 7 channels in each router called East (E), West (W), North1 (N1), North2 (N2), South1 (S1), South2 (S2), and Local (L). The basic routing algorithm is summarized as the pseudo-code in Fig. 4. As can be seen in this pseudo code, several output channels are eligible to deliver a packet. For example, when the input port is N1 and the destination is in the NE quadrant, the possible output channels are as E, W, S1, N2, and S2. The capability of this algorithm to support non-minimal paths helps to design the redundant routing unit.

C. Redundant Routing Computation Unit

The redundant routing computation (RRC) module will be activated and replaced the original routing when faults occur in the RC module, providing much simpler and more robust routing to the on-chip network. The routing algorithms, shown between Fig. 5 and Fig. 7, are running in RRC units. Depending on the location of routers in the network, the routers are classified into three groups and the redundant routing paths for the routers in the corner, borderline, and central parts are shown in Fig. 3 (a), Fig. 3 (b), and Fig. 3 (c), respectively. For example, if the router a1 in Fig. 3 (a) is faulty, the default adaptive routing unit is replaced by the algorithm indicated in Fig. 5, Router (a1).

D. The packets receiver

In general, packets will be processed in the RC module to verify whether they should be delivered to one of the neighbouring routers or to the local core. If the routing unit is

```
Redundant Routing Paths for Borderline
Router (b1)
  IF     InChannel = {L}  THEN OutChannel (S1)<='1';
  ELSEIF InChannel = {E}  THEN OutChannel (W)<='1';
  ELSEIF InChannel = {W}  THEN OutChannel (E)<='1';
  ELSEIF InChannel = {S1} THEN OutChannel (W)<='1';
  ELSEIF InChannel = {S2} THEN OutChannel (E)<='1';
Router (b2)
  IF     InChannel = {L}  THEN OutChannel (W)<='1';
  ELSEIF InChannel = {W}  THEN OutChannel (S2)<='1';
  ELSEIF InChannel = {N1} THEN OutChannel (S1)<='1';
  ELSEIF InChannel = {N2} THEN OutChannel (S2)<='1';
  ELSEIF InChannel = {S1} THEN OutChannel (W)<='1';
  ELSEIF InChannel = {S2} THEN OutChannel (N2)<='1';
Router (b3)
  IF     InChannel = {L}  THEN OutChannel (N1)<='1';
  ELSEIF InChannel = {E}  THEN OutChannel (W)<='1';
  ELSEIF InChannel = {W}  THEN OutChannel (E)<='1';
  ELSEIF InChannel = {N1} THEN OutChannel (W)<='1';
  ELSEIF InChannel = {N2} THEN OutChannel (E)<='1';
Router (b4)
  IF     InChannel = {L}  THEN OutChannel (E)<='1';
  ELSEIF InChannel = {E}  THEN OutChannel (S1)<='1';
  ELSEIF InChannel = {N1} THEN OutChannel (S1)<='1';
  ELSEIF InChannel = {N2} THEN OutChannel (S2)<='1';
  ELSEIF InChannel = {S1} THEN OutChannel (E)<='1';
  ELSEIF InChannel = {S2} THEN OutChannel (N2)<='1';
```

Fig. 7 The algorithm for the routers in borderlines

faulty, packets may never reach the destinations. To solve this problem, the packet receiver unit is designed to detect the packets with the local core as the destination before forwarding the packets to the RC module. As shown in Fig. 8 for this purpose, the address segment of packets is compared with the address of the local router using a comparator. Once the address matches the current node address, the packet will be forwarded to the local core.

IV. Result and Discussion

We evaluate the proposed mechanism in terms of reliability and performance. The reliability is the ratio of the number of well-running cases over the total number of cases. In each case, faults will be randomly injected to RC modules and we force packets to take a random turns despite the decision of the routing computation units in the faulty RC modules. This operation is similar to the behaviour when a fault occurs in the circuit level of routers such as stuck-at-1 and stuck-at-0 and eventually faults lead to illegal turns. Those cases with equal number of send/receive packets are defined as the well-running cases meaning that the NoC is fully reliable.

For measuring performance, the time takes for packets to reach from a source node to a destination node is considered under various packet injection rates. The experiments are performed on a 2D 8×8 mesh network using wormhole switching with a constant packet size of 4 flits and different packet injection rates in the POPNET simulator. Faults are injected randomly to each RC module of the on-chip network.

A. Reliability Analysis

In this set of experiments, we evaluate the reliability of the

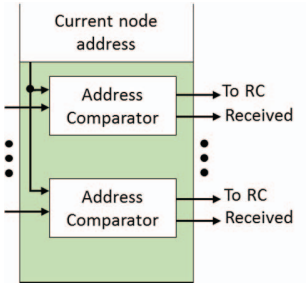


Fig. 8 The packets receiver

network for different fault injection percentage between 0% and 20% under Packet Injection Rate (PIR) 0.05 (Packet/Router/Cycle). Since the reliability of the proposed mechanism decreases below 90%, we stop the fault injection percentage at 20%. It means that 58 out of totally 288 RC modules in the 2D 8×8 mesh network is faulty in the most severe case of this experiment. (Note that the routers located in the central part of the network have 5 RC units while the others in the corners and borderlines own 3 and 4 RC units respectively so that we have 288 RC units in total). The proposed mechanism is compared with the DyXY routing algorithm and Triple Modular Redundancy (TMR). In TMR, each RC module with the DyXY routing algorithm is triplicated in order to achieve fault tolerance (i.e. the number of RC units is 864) while in the proposed mechanism the faulty RC module is replaced with a redundant unit with a very simple logic circuit.

Fig. 9 shows the reliability of the proposed method in comparison with DyXY and TMR. As can be seen in this figure, the proposed mechanism is reliable under 14% RC unit failure. DyXY, however, fails by faults in any of the RC units. TMR on the other hand, is reliable even with the same routing algorithm as DyXY when the faulty units are less than 5%. The reason for the efficiency of the proposed method is that the RRC unit is very light-weight and the probability of faults is nearly zero in them. On the contrary, the redundant RC modules in TMR have the same probability of fault as the original module.

The fluctuation in the figures is due to the fact that all random patterns cannot be practically examined. The reliability value for each dot is calculated based on the average of 1000 random selections (e.g. 1000 different random patterns of 2% faulty RCs among more than 700 billion possible patterns).

B. The Performance

The performance of the proposed mechanism under different fault percentage is shown in Fig. 10 to Fig. 14 under 5 traffic patterns as Uniform, Transpose1, Transpose2, Bit-Reversal and Shuffle traffic. The orientation of choosing source and destination nodes generates the difference between Transpose1 and Transpose2. In these figures, the DyXY algorithm is selected as the baseline showing the optimal performance. It is a fully adaptive routing algorithm using the same number of virtual channels as the proposed mechanism but supporting only the minimal directions. The X axis represents the packet injection rate while the Y dimension shows the average latency. As can be seen in these figures, the average latency of the proposed mechanism is nearly the same as those using DyXY when no fault occurs. The reason is that, in fault-free conditions, both methods follow the minimal paths.

In general, faults in central areas have more severe effects on the latency than those located in corners and along the edges. By injecting faults with the percentages of 2, 4, 6, 8, and 10, it can be observed that the network is saturated earlier. The reason is that when increasing the percentage of faults, more packets follow the non-minimal paths and thus experience higher latency.

C. Area Overhead and Power Consumption

To evaluate the area overhead and power consumption, an on-chip network router with the proposed mechanism, DyXY, and TMR are synthesized by Synopsys Design Compiler. For synthesizing, the TSMC45nm technology at the operating frequency of 1GHz and supply voltage of 0.9V is applied. We keep these three mechanism in a highly similar router structure except the PR modules and RC modules to achieve reasonable comparison.

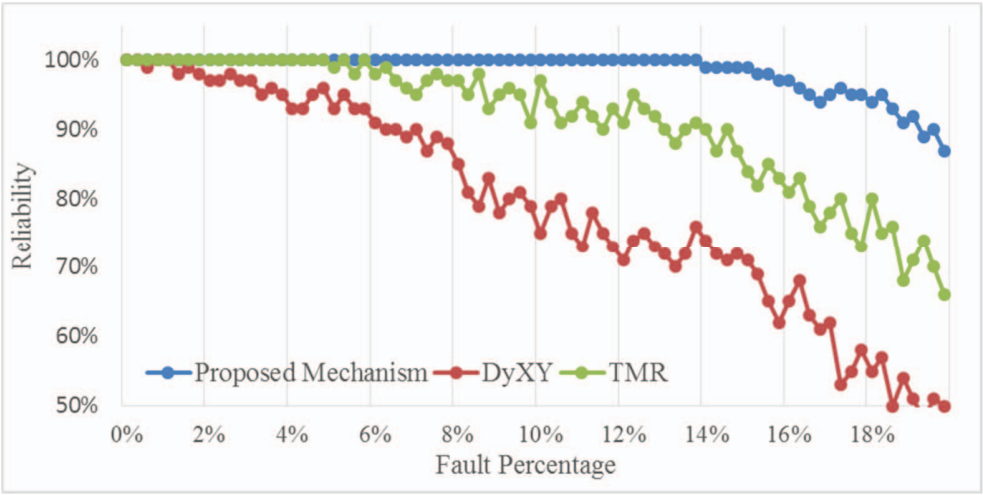


Fig. 9 The reliability of proposed mechanism

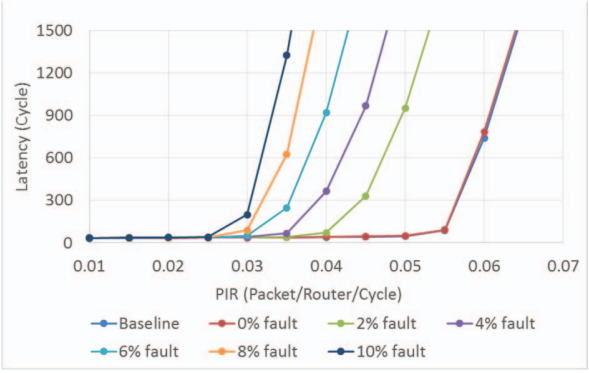


Fig. 10 The latency under uniform traffic

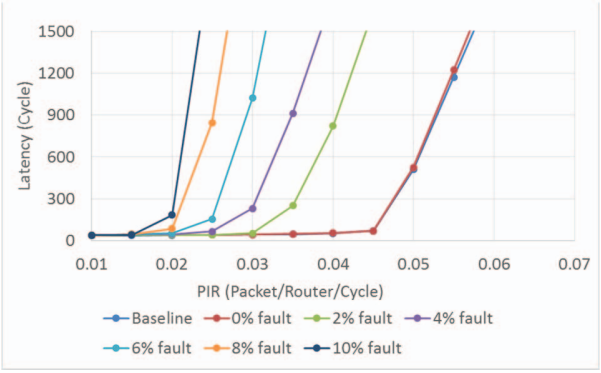


Fig. 11 The latency under Transpose1 traffic

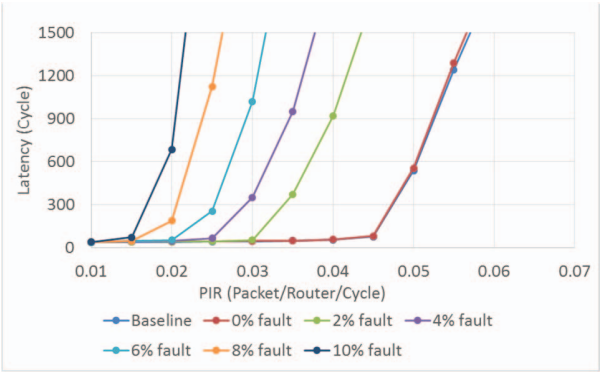


Fig. 12 The latency under Transpose2 traffic

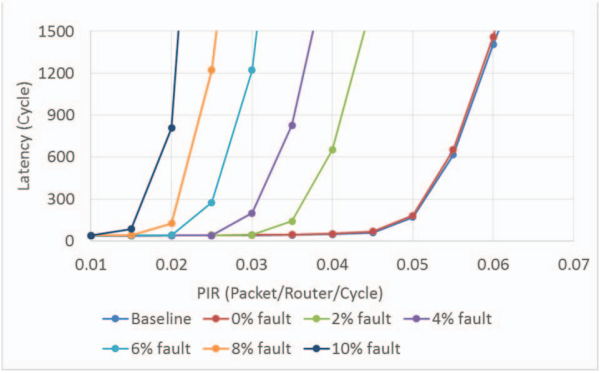


Fig. 13 The latency under Bit-Reversal traffic

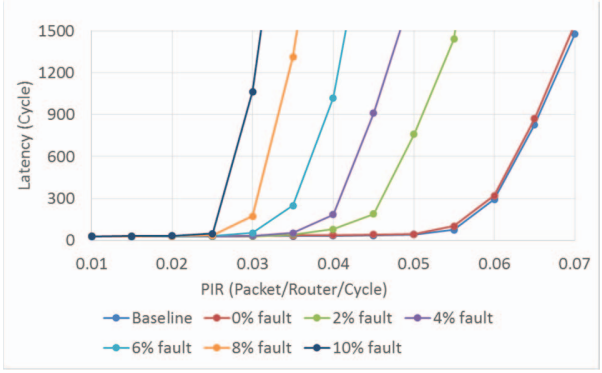


Fig. 14 The latency under Shuffle traffic

For example, they share the same number of physical ports and virtual channels.

As indicated in Table 1, the power consumption and area overhead of the router with the proposed mechanism is comparable with the default router using DyXY routing as well as the default router with the TMR solution. The hardware overhead of the proposed mechanism is between the other two models and the comparison among these models is listed in the table where a router with DyXY is chosen as the baseline. It is shown that the proposed mechanism has an additional 7.29% area and 6.20% power overhead than DyXY while TMR has 38.3% and 24.21% additional overhead.

V. CONCLUSION

We proposed a method to tolerate faults in the routing computation units. It is a fine-grained solution to the control path faults including the basic routing algorithm, the redundancy routing computation (RRC) and the packet receiver unit (PR). By applying this mechanism, the faulty RC module can be replaced by the redundancy routing module which is lightweight and robust, offering reliable services with a low probability of faults. In this case, the other components are still functional which maintain the performance of on-chip network. Experimental results show that the average latency of transmission in the proposed mechanism is almost the same as DyXY routing when no fault occurs. In addition, more reliability can be achieved in the proposed mechanism comparing with those using TMR in RC modules. In the proposed mechanism, the on-chip network is reliable when nearly 14% of the RC modules suffer from transient faults simultaneously. The work presented in this paper concentrated solely on the routing computation unit. For future work, we intend to expand the mechanism focusing on the design of a simple redundant module to the whole control path in NoC router, offering comprehensive solution to faults.

ACKNOWLEDGMENT

This work is supported by the NSFC under grant No.61176025 and No.61006027 and the Oversea Academic Training Funds (OATF), UESTC. It is also supported by VINNOVA-MarieCurie within the ERoT project and Academy of Finland.

Table 1. Area overhead and power consumption

	Default Router(DyXY)	Default Router(TMR)	Proposed Mechanism
Cell area	1848.46um^2	2551.44 um^2	1983.24 um^2
Cell power	264.56uW	328.60uW	280.97uW
vs in area	+0.00%	+38.03%	+7.29%
vs in power	+0.00%	+24.21%	+6.20%

REFERENCES

[1] A. Jantsch, H. Tenhunen, Networks on chip, Vol. 396, Dordrecht: Kluwer Academic Publishers, 2003.

[2] K. Bernstein, et al, "High-performance CMOS variability in the 65-nm regime and beyond," IBM journal of research and development, vol. 50, no. 4/5, pp. 433-449, 2006

[3] J. D. Owens, et al, "Research challenges for on-chip interconnection networks," IEEE micro, vol. 27, no. 5, pp. 96-108, 2007.

[4] S. R. Sarangi, et al, "VARIUS: A model of process variation and resulting timing errors for microarchitects," IEEE Transactions on Semiconductor Manufacturing, vol. 21, no.1, pp 3-13, 2008.

[5] M. Orshansky, S. Nassif, D. Boning, Design for manufacturability and statistical design: a constructive approach, Springer Science & Business Media, 2007.

[6] X. Fu, T. Li, J. A. Fortes, "Architecting reliable multi-core network-on-chip for small scale processing technology," IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2010.

[7] J. Howard, et al, "A 48-core IA-32 processor in 45 nm CMOS using on-die message-passing and DVFS for performance and power scaling," IEEE Journal of Solid-State Circuits, vol. 46, no.1, pp. 173-183, 2011.

[8] A. Ganguly, P. P. Pande, B. Belzer, "Crosstalk-aware channel coding schemes for energy efficient and reliable NOC interconnects," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 17, no. 11, pp. 1626-1639, 2009.

[9] L. K. Loo, et al. "Packet logging mechanism for adaptive online fault detection on Network-on-Chip," IEEE International Symposium on Circuits and Systems (ISCAS), 2014.

[10] G. Schley, N. Batzolis, M. Radetzki, "Fault localizing end-to-end flow control protocol for Networks-on-Chip," Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), 2013.

[11] A. Prodromou, A. Panteli, C. Nicopoulos, Y. Sazeides, "Nocalert: An on-line and real-time fault detection mechanism for network-on-chip architectures," IEEE/ACM International Symposium on Microarchitecture (MICRO), 2012.

[12] P. P. Pande, C. Grecu, R. Saleh, G. Demicheli, "Design, synthesis, and test of networks on chips," IEEE Design & Test of Computers, vol. 22, no. 5, pp. 404-413, 2005.

[13] C. Nicopoulos, V. Narayanan, C. R. Das, On the effects of process variation in Network-on-Chip architectures, Springer Netherlands, 2010.

[14] D. M. Ancajas, K. Bhardwaj, K. Chakraborty, S. Roy, "Wearout resilience in NoCs through an aging aware adaptive routing algorithm," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 23, no. 2, pp. 369-373, 2015.

[15] M. R. Kakoee, V. Bertacco, L. Benini, "ReliNoC: A reliable network for priority-based on-chip communication," Conference on Design, Automation & Test in Europe (DATE), 2011.

[16] D. Fick, et al, "Vicis: a reliable network for unreliable silicon," Design Automation Conference (DAC), 2009.

[17] K. Bhardwaj, K. Chakraborty, S. Roy, "Towards graceful aging degradation in nocs through an adaptive routing algorithm," Design Automation Conference(DAC), 2012.

[18] D. M. Ancajas, K. Chakraborty, S. Roy, "Proactive aging management in heterogeneous NoCs through a criticality-driven routing approach," Conference on Design, Automation and Test in Europe (DATE), 2013.

[19] M. Ebrahimi, M. Daneshtalab, J. Plosila, H. Tenhunen, "MAFA: adaptive fault-tolerant routing algorithm for networks-on-chip," Euromicro Conference on Digital System Design (DSD), 2012.

[20] M. Ebrahimi, M. Daneshtalab, J. Plosila, "High performance fault-tolerant routing algorithm for NoC-based many-core systems," Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), 2013.

[21] C. Iordanou, V. Soteriou, K. Aisopos, "Hermes: Architecting a top-performing fault-tolerant routing algorithm for Networks-on-Chips," IEEE International Conference on Computer Design (ICCD), 2014.

[22] B. Fu, Y. Han, H. Li, X. Li, "Zonedefense: a fault-tolerant routing for 2-d meshes without virtual channels," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 22, no. 1, pp. 113-126, 2014.

[23] M. Ebrahimi, M. Daneshtalab, P. Liljeberg, J. Plosila, H. Tenhunen, "LEAR--A low-weight and highly adaptive routing method for distributing congestions in on-chip networks," Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), 2012.

[24] M. Li, Q. A. Zeng, and W. B. Jone, "DyXY: a proximity congestion-aware deadlock-free dynamic routing method for network on chip." Design Automation Conference (DAC), 2006.