Detection of Faults in NOC with Efficient Fault-Tolerent Deflection Router

L. Vinoth[1], A. Abdul Khadar[2]
[1, 2] Department of ECE
[1, 2] Gnanamani College of Technology, Anna University, Chennai

Abstract- *Networks-on-Chips (NoCs) provide inherent structural redundancy of on-chip communication pathways. The Based on a fine-grained functional fault model, error-detecting circuitry, and distributed on-line fault diagnosis, we determine the fault status of NoC switches, including their adjacent links. This project proposes a fault-tolerant solution for a bufferless 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. Simulation results demonstrate that under synthetic workloads, in the presence of permanent link faults, the throughput of an 8 × 8 network with FTDR and FTDR-H algorithms are 14% and 23% higher on average than that with the fault-on-neighbor (FoN) aware deflection routing algorithm and the cost-based deflection routing algorithm, respectively. For transient faults, the performance of the FTDR router can achieve graceful degradation even at a high fault rate.*

Keywords- Networks-on-Chips, Fault-tolerant deflection routing, Fault-on-neighbor.

I. INTRODUCTION

A network-on-chip (NoC) is an on-chip communication infrastructure that implements multi-hop and predominantly packet-switched communication. Through pipelined packet transmission, NoCs permit a more efficient utilization of communication resources than traditional on-chip buses. Regular NoC structures reduce VLSI layout complexity compared to custom routed wires.

In future chip generations, faults will appear with increasing probability due to the susceptibility of shrinking feature sizes to process variability, age-related degradation, crosstalk, and single-event upsets. To sustain chip production yield and reliable operation, very large numbers of faults will have to be tolerated. In order to take countermeasures against NoC faults, we must first detect and diagnose them.

To this end, we propose to equip data packets with CRC checksums and to augment the switch circuit architecture with error detecting units. Based on the information from these units and additional directed test patterns, a distributed diagnosis of switches and their immediate neighborhood is performed. Diagnosis results are stored in on-chip structures that represent information of a detailed functional fault model. This information requires special protection so as not to become corrupted by faults, which would jeopardize switch operation. It is used by a new fault-adaptive deflection routing algorithm to avoid defective parts of the switch but still utilize its remaining functionality. Thereby, a more graceful degradation of network operation, even in presence of large numbers of faults, is achieved.

Recently, buffer less router has been studied in NoC to achieve higher speed and lower cost than a wormhole or virtual channel router. Except one input register for each input port, there are no other buffers in the buffer less router. Due to the lack 2of buffers, deflection routing is utilized in the buffer less router to route packets to neighboring routers immediately without buffering in the router. The fully adaptive feature of deflection routing provides the potential to route packets to avoid faulty links/routers and achieve fault-tolerance. In our previous works, a reconfigurable fault-tolerant deflection routing (FTDR) algorithm based on reinforcement learning has been proposed for 2-D mesh NoC. 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. In, we only focus on the routing algorithm to handle permanent faults.

Recently, buffer less router has been studied in NoC to achieve higher speed and lower cost than a wormhole or virtual channel router. Except one input register for each input port, there are no other buffers in the buffer less router. Due to the lack of buffers, deflection routing is utilized in the buffer less router to route packets to neighboring routers immediately without buffering in the router. The fully adaptive feature of deflection routing provides the potential to route packets to avoid faulty links/routers and achieve fault-tolerance. In our previous works, a reconfigurable fault-tolerant deflection routing (FTDR) algorithm based on reinforcement learning has been proposed for 2-D mesh NoC. 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. In, we only focus on the routing algorithm to handle permanent faults.
A hybrid automatic repeat request (ARQ) and forward error correction (FEC) link-level error control scheme using retransmission is proposed to handle transient faults. The FTDR algorithm guarantees "0 lost packet" as long as the fault pattern does not cut the network into two or more disconnected parts. This project proposes a fault-tolerant solution for a buffer less NoC.

II. FUNCTIONS OF LINK-LEVEL ERROR CONTROL SCHEME SYSTEM

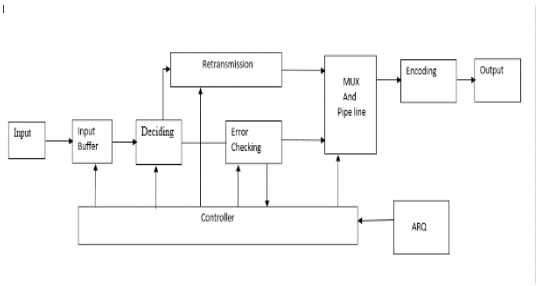

Fig. 1 Block Diagram of Link-Level Error Control Scheme System

The ARQ scheme using retransmission performs well without incurring much latency for low error rates; however, at higher error rate the hybrid ARQ/FEC scheme with slight hardware overhead provides better performance than the pure ARQ scheme. We propose a hybrid ARQ/FEC scheme 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. The hardware structure of the ARQ scheme for one input port North, East, South, West, Local. Each input port of the router has an input buffer with two entries instead of the original one and the boundary input port of the boundary router has an input buffer with three entries. Additionally, a retransmission buffer (RBi) is used to buffer the packet which may be retransmitted. After decoding, the packet will be written to RBi. A 2-to-1 multiplexer is used to select to send a new packet or retransmit the last packet. A request signal arq is introduced between two neighboring routers to indicate whether the last packet should be retransmitted or not. The fault information transmission signal is used to disable the outgoing link i of the upstream router temporarily.

2.1 NETWORK ON CHIP

Network on chip or network on a chip (NoC or NOC) is a communication subsystem on an integrated circuit (commonly called a "chip"), typically between IP cores in a system on a chip (SoC). NoCs can span synchronous and asynchronous clock domains or use unclocked asynchronous logic. NoC technology applies networking theory and methods to on-chip communication and brings notable improvements over conventional bus and crossbar interconnections. NoC improves the scalability of SoCs, and the power efficiency of complex SoCs compared to other designs.

2.2 PARADIGM

Network on chip is an emerging paradigm for communications within large VLSI systems implemented on a single silicon chip. Sgroi et al. call "the layered-stack approach to the design of the on-chip inter-core communications the network-on-chip (NOC) methodology." In a NoC system, modules such as processor cores, memories and specialized IP blocks exchange data using a network as a "public transportation" sub-system for the information traffic. A NoC is constructed from multiple point-to-point data links interconnected by switches (a.k.a. routers), such that messages can be relayed from any source module to any destination module over several links, by making routing decisions at the switches.

A NoC is similar to a modern telecommunications network, using digital bit-packet switching over multiplexed links. Although packet-switching is sometimes claimed as necessity for a NoC, there are several NoC proposals utilizing circuit-switching techniques. This is somewhat confusing since all above mentioned are networks (they enable communication between two or more devices) but they are not considered as network-on-chip approaches.

2.3 PARALLELISM AND SCALABILITY

The wires in the links of the NoC are shared by many signals. A high level of parallelism is achieved, because all links in the NoC can operate simultaneously on different data packets. Therefore, as the complexity of integrated systems keeps growing, a NoC provides enhanced performance (such as throughput) and scalability in comparison with previous communication architectures (e.g., dedicated point-to-point signal wires, shared buses, or segmented buses with bridges). Of course, the algorithms must be designed in such a way that they offer large parallelism and can hence utilize the potential of NoC.

2.4 BENEFITS OF ADOPTING NoCs

Traditionally, ICs have been designed with dedicated point-to-point connections, with one wire dedicated to each signal. For large designs, in particular, this has several limitations from a physical design viewpoint. The wires occupy much of the area of the chip, and in nanometer CMOS technology, interconnects dominate both performance and dynamic power dissipation, as signal propagation in wires across the chip requires multiple clock cycles. (See Rent's rule for a discussion of wiring requirements for point-to-point connections).

NoC links can reduce the complexity of designing wires for predictable speed, power, noise, reliability, etc., thanks to their regular, well controlled structure. From a system design viewpoint, with the advent of multi-core processor systems, a network is a natural architectural choice. A NoC can provide separation between computation and communication, support modularity and IP reuse via standard interfaces, handle synchronization issues, serve as a platform for system test, and, hence, increase engineering productivity.

2.5 RESEARCH ON ON-CHIP NETWORKS

Although NoCs can borrow concepts and techniques from the well-established domain of computer networking, it is impractical to blindly reuse features of "classical" computer networks and symmetric multiprocessors. In particular, NoC switches should be small, energy-efficient, and fast. Neglecting these aspects along with proper, quantitative comparison was typical for early NoC research but nowadays they are considered in more detail. The routing algorithms should be implemented by simple logic, and the number of data buffers should be minimal. Network topology and properties may be application-specific.

To date, several prototype NoCs have been designed and analyzed in both industry and academia but only few have been implemented on silicon. However, many challenging research problems remain to be solved at all levels, from the physical link level through the network level, and all the way up to the system architecture and application software. The first dedicated research symposium on networks on chip was held at Princeton University, in May 2007. Research has been done on integrated optical waveguides and devices comprising an optical network on a chip (ONoC).

III. NOC TOPOLOGY AND PACKET FORMAT

The NoC architecture is based on a 2-D mesh topology, Nostrum NoC.Each processing element is attached to a router (R), as shown in Fig.. 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.

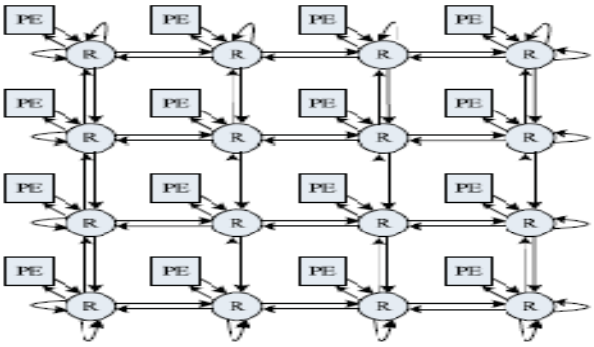

Fig. 2 (a) Noc Architecture

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.

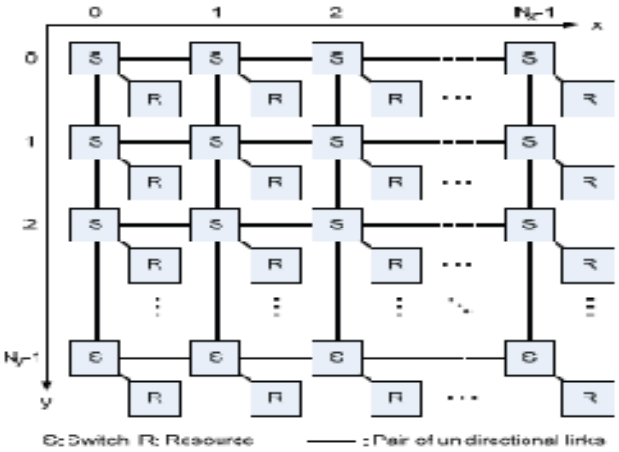

Fig. 2 (b) Network Topology

The Nostrum packet format includes a flag, V, to mark valid packets. If that flag is set to zero, all other bits are ignored, whereby the absence or invalidity of a packet is represented. The HC field holds the current hop count of the packet. A relative addressing scheme is employed that encodes the Δx and Δy distances to the target with a sign-magnitude representation instead of using absolute (x, y) coordinates. This relieves switches from having to know their position and allows routing decisions to be made based on the sign bits only, but requires implementation of address updating (±1). Address bit width is chosen in accordance with a reference

design for sake of comparison, but could be reduced to the benefit of the payload in case of moderately sized NoCs.

IV. ENCODEDING OF PACKET

The basic data transfer unit in this project is a packet. The original packet format, which is compatible with a multicore NoC platform is shown in Fig. A packet, which has 114 bits, contains a 34-bit head and an 80-bit payload. A valid bit (V) is used to mark a packet valid or not. Relative addressing is used for the source and destination address fields (SA and DA) which have 12 bits (six bits for row/column address), respectively.

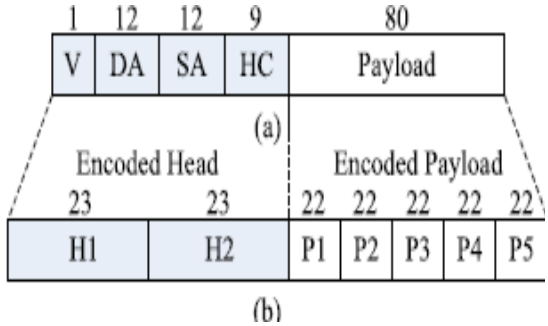

Fig.3 Original Packet Format And Encoded Packet Format.

The hop counter field (HC, nine bits) records the number of hops the packet has been routed. In order to detect and correct errors in transmission, SECDED Hamming codes are used to encode the head and payload, respectively. The encoded packet format, which has 156 bits, is shown in Fig. 2(b). The head is divided into two parts and Hamming (23, 17) code is used to encode each part. The payload is divided into five parts, each of which is encoded with Hamming (22, 16) code.

4.1 MODIFIED PACKET

The packet format by using eight payload bits to store a CRC checksum for error detection. The CRC polynomial used is g(x) = x8 + 1. It detects all packets with an odd number of erroneous bits or a single burst of up to 8 erroneous bits. The latter property ensures that spatially correlated errors, e.g. from crosstalk between adjacent wires, are covered.
The choice of the polynomial is primarily driven by the need to compute the CRC in a combinational way, without using LSFRs. The parity trees employed for that purpose grow with the complexity of the polynomial.

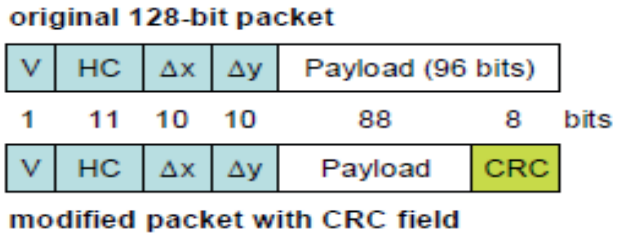

Fig.4 Modified Packet With Crc Field

The area cost of using polynomials with additional properties such as detection of all independent double errors is prohibitive, as s While we cannot guarantee the detection of such errors, the probability of missing arbitrary bit faults (regardless of their number and position) is less than 0.004.In the following, beyond error detection, we use the CRC field to fully diagnose NoC links and switch datapaths.

4.2 FAULT DIAGNOSIS

The applied fault model differentiates between errors that occur during the transmission of packets between adjacent switches and errors that are introduced locally at a switch in the forwarding process. For local faults, all elements of the switch data path are accounted for. However, errors provoked by router malfunction are currently not taken care of because in case of a defective router the switch as a whole unit needs to be avoided or shut down. The respective diagnosis is not in scope of this work; it can be performed on higher network layers or by gate-level structural diagnosis of the router logic. For transmission errors, both the link wires and the input registers are taken into account, as a fault in any of these two components leads to corrupted data at the output of the input register. In order to determine the source of detected errors, each input and output except those connected to the local resource features a combinational CRC unit . On reception of a packet as well as before being sent out to the next switch, its checksum is verified by the CRC units assigned to the input and output, respectively. Errors induced internally by the crossbar are detected by a matching checksum at an input and a mismatch at the output, whereas errors caused by switch to switch transmission are characterized by a CRC mismatch at the input. The connections to and from the local resource L have no CRC checks as there is no redundancy that could be used to bypass diagnosed faults.

As the packet header is changed during the routing process, the checksum also needs updating. The recalculation is performed by a CRC instance right after the router that employs an own CRC tree for taking the new header bits into account. For the payload bits the intermediate results from the CRC unit after the input register are used. This not only reduces area consumption, but also ensures that packets

altered between the input register and the crossbar input are not assigned a valid checksum by mistake.

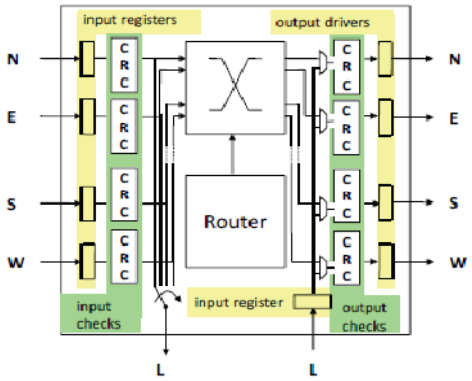

Fig.5 Switch With CRC Checks

The precondition for a fault-tolerant system running smoothly is to detect the location of the faults first. The fault detection mechanism should also distinguish transient faults from permanent faults. Transient link errors can be detected via error coding techniques, such as cyclic redundancy check and parity codes. Few works focus on detecting transient faults and permanent faults at the meantime. However, the error control coding (ECC) scheme which is selected based on the noise condition can only detect transient errors.

4.3 LINK-LEVEL FAULT DETECTION AND PROTECTION

SECDED Hamming code, which can correct single error and detect double errors, is used to encode the packet to perform fault diagnosis. To make a compromise among performance, area and power consumption, we compare two ECC strategies: 1) encode the whole packet with Hamming (122,114) code and 2) encode the head with two Hamming (23, 17) codes and the payload with five Hamming (22, 16) codes. For the first encoding strategy, eight parity bits are used to encode the 114 bits packet into 122 bits. It can only correct onebit error and detect two-bit error in the packet. The second strategy divides the packet into seven parts: two for head and five for payload, encoded with Hamming (23, 17) and Hamming (22, 16) codes, respectively. The encoded packet length has 156 bits with 42 parity bits. It can correct seven simultaneous single-bit errors in each part and detect at most 14 error bits in the case of a two-bit error in each part. If any of the seven parts contains a two-bit error for one cycle, it will lead to a retransmission. If the retransmitted packet has the same two-bit error in any part, the link will be tested to check if it is considered as a permanent faulty link. The input register is followed by the decoder, so the ECC mechanism can detect and correct both link and input register errors. Table I shows the latency, area, and power comparison of encoders and decoders for Hamming (122, 114) and (22, 16) codes, respectively (developed in VHDL and synthesized with TSMC 65-nm technology). The results for Hamming (23, 17) code are similar as Hamming (22, 16) code. As the table illustrates, the latency, area, and power consumption of the encoder and decoder for Hamming (122, 114) code are much larger than those of Hamming (22, 16) code.

4.4 LINK-LEVEL ERROR CONTROL SCHEME

The ARQ scheme using retransmission performs well without incurring much latency for low error rates, however, at higher error rate the hybrid ARQ/FEC scheme with slight hardware overhead provides better performance than the pure ARQ scheme. We propose a hybrid ARQ/FEC scheme 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. A retransmission buffer (RBi) is used to buffer the packet which may be retransmitted. After decoding, the packet will be written to RBi. A 2-to-1 multiplexer is used to select to send a new packet or retransmit the last packet. A request signal arq is introduced between two neighboring routers to indicate whether the last packet should be retransmitted or not. The fault information transmission signal (fault_to[i]) is used to disable the outgoing link i of the upstream router temporarily.

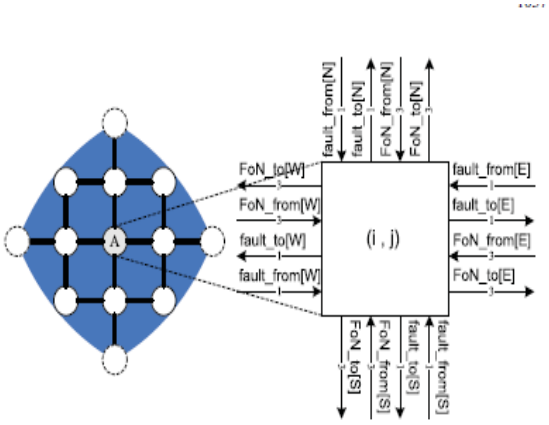

Fig.6 Fault Information Transmission Mechanism

After detecting this signal valid, the router n–1 will disable the output i temporarily for one cycle to stop sending a packet to router n. If the router n-1 is fully loaded, which means it has four packets to handle, it will also set the signal fault_to[i] to disable the output i of the router n–2 for one cycle. This process will be repeated along the direction i until finding a router, which is not fully loaded or reaching the boundary router.

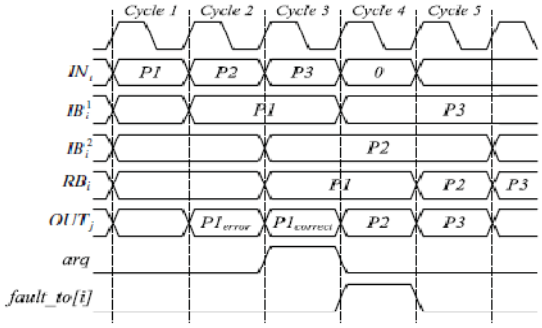

Fig.7 Signal Timing Of Arq Scheme.

Different from the ARQ scheme described if P2 at Cycle 4 contains a transient fault, our scheme can retransmit it at Cycle 5. However, the ARQ cannot handle this situation and the packet will be lost, because it uses only one input register and one retransmission register. Actually, our proposed error control scheme can also be extended to contain more input buffer entries to handle transient faults lasting for more than one cycle easily. Without loss of generality, to handle transient faults lasting for n cycles, the bufferless router needs an input buffer with at least n + 1 entries.

V. RESULT

Simulation Result

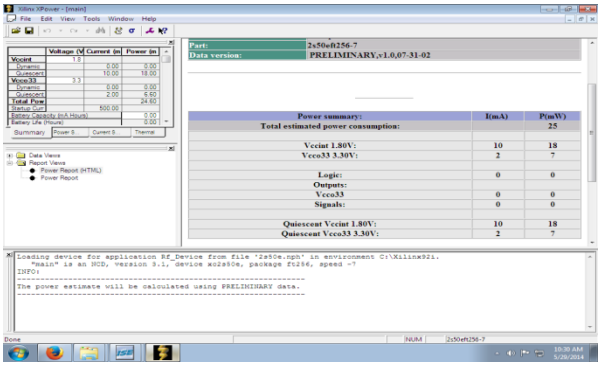

Fig.7 Proposed System Simulation Result

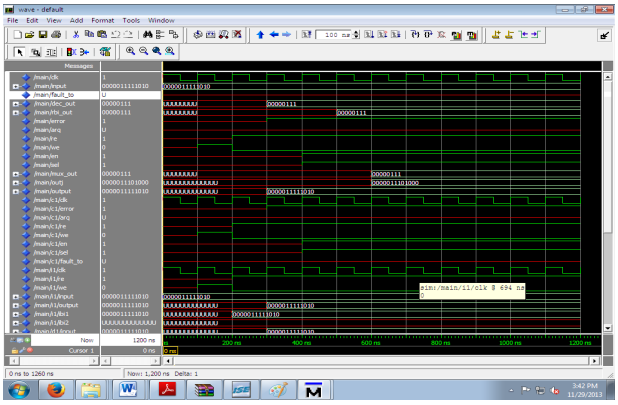

Fig.8 Simulation Result

VI. CONCLUSION

I have presented a fault-adaptive deflection routing mechanism that takes the detailed fault status of NoC crossbar connections into account. The fault status is obtained through distributed online diagnosis that distinguishes between permanent and transient faults. The combination of routing and diagnosis results in a highly reliable transmission of injected packets to their destinations even under the effect of a large number of faults and thus minimizes costly retransmissions, while keeping the increase in area and latency in an acceptable range. 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. The FTDR router is cost-efficient for small networks (e.g., less than 64 nodes), while the FTDR-H router has good scalability and is a feasible solution for at least several hundreds of nodes.The time is 0.36ns and my proposed system reduce time duration about approximately 0.10ns to 0.16ns.

REFERENCES

[1] Bertozzi.D, Benini.L, and De Micheli.G, Jun. 2005 “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.

[2] Constantinescu.C, Jul.–Aug. 2003 “Trends and challenges in VLSI circuit reliability,” IEEE Micro, Vol. 23, No. 4, pp. 14–19,.

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

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

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

[6] Grecu.C, Ivanov.A, Saleh.R, Sogomonyan.E. S., and Pande. P. P, Jul. 2006 “Online fault detection and

location for NoC interconnects," in Proc. 12th IEEE Int. On-Line Test. Symp., pp. 145–150.

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

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

[9] Kohler, Schley.G, and Radetzki.M, Jun. 2010 "Fault tolerant network on chip switching with graceful performance degradation," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., Vol. 29, No. 6, pp. 883–896.

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

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

[12] Pasricha.S, Zou.Y, Connors.D, and Siegel.H. J, Oct. 2010 "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, pp. 85–94.

[13] Patooghy.A and Miremadi.S.G, 2009 "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.

[14] Zimmer.H and Jantsch.A, Oct. 2003 "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, pp. 188–193.