AN EFFICIENT SOLUTION FOR ADDRESSING FAULTS IN SOC

[1]B.Mahalakshmi, [2]P.Sowmiya
[1]PG Scholar, Department of Electronics and Communication Engineering, Shivani College of Engineering and Technology, Trichy
[2]Assistant Professor, Department of Electronics and Communication Engineering, Shivani College of Engineering and Technology, Trichy
[1]mahalakshmikrishece@gmail.com, [2]sowmiyanst@gmail.com

Abstract: As technology scales chips are becoming less reliable, harder to keep defect density constant and the variations becomes more severe. Soft error rate also grows exponentially. Reliability concerns for the future technologies can arise in the form of permanent and transient faults. This paper proposes a fault tolerant solution for a bufferless system on chip. In particular, recent work proposes bufferless deflection routing to eliminate router buffers. A hybrid automatic repeat request and forward error correction link level error control scheme is used to handle transient faults. A reinforcement learning based Fault Tolerant Deflection Routing (FTDR) is also proposed in order to handle permanent faults. We also assume that a hierarchical based FTDR algorithm (FTDR-H) can reduce the area than FTDR.

Index Terms: Deflection Routing, Fault tolerance, Bufferless, Permanent fault, Transient fault.

1. INTRODUCTION

Continuing decrease in future size of Very Large Scale Integrated Circuits (VLSI) enables the integration of dozens, and in future hundreds of processing elements on a single chip. SoC approach has emerged as a promising solution for on-chip communication. Two kinds of faults need to be addressed in SoC architecture. End to end flow control based methods [1] and [2] combines the error control code with retransmission mechanism inorder to tolerate transient faults and the fault tolerant routing utilizes the structure of SoC to route signals around the fault tolerant processing elements and faulty routers or links / switches to achieve fault tolerance. A good fault tolerant routing should ensure „0" lost packet in whatever fault patterns exist but many of them [2] and [3] only improves the successful arrival rate of packet. Bufferless design is used to achieve higher speed and lower cost. It has recently been evaluated as an alternative to traditional virtual channel buffered routers [4]. Buffers consume significant portion of this power. A recent work [5] reduced network energy by 40% by eliminating buffers. Deflection routing is utilized on the buffer less router to route packets to neighboring immediately without buffering.

In previous works [6], a reconfigurable Fault Tolerant Deflection Routing (FTDR) algorithm based on reinforcement learning has been proposed. FTDR algorithm is a topology agnostic feature may be applied to regular and irregular topologies. The routing table can automatically reconfigure during signal transmission.

The fault diagnosis mechanism uses the single error correcting and double error detecting (SECDED) Hamming code [7] to detect both transient and permanent faults. A hybrid ARQ and FEC is proposed to handle transient faults and FTDR guarantees „0" lost packet as long as fault pattern exist. A hierarchical based algorithm is assumed to reduce the area overhead.

Few previous works provides a fault tolerant solution to handle faults for NoC. This paper proposes a fault tolerant solution for a bufferless SoC. By comparing the existing and proposed system, the following aspects are provided below.

1) A link level error control scheme used to handle transient faults through hybrid ARQ/FEC method. The previous works only proposed the solution to handle permanent faults.

2) A fault diagnosis mechanism using SECDED hamming code is proposed to distinguish transient from permanent faults. While the previous works does not include this topic.

3) A test method is assumed to check whether a link contains permanent or transient fault. It does not interfere with regular signal transmission.

The rest of the paper is organized as follows. Related works are reviewed in section II. Section III describes fault injection method and section IV

describes about Fault diagnosis mechanism and Fault tolerant deflection routing is proposed here

2. Related works

A diagnosis method that injects test patterns at the boundaries of a 2D mesh network has been proposed in [8], to locate faults based on different test patterns. Transient faults in NoC can be handled at both link and transport level by three schemes, ARQ, FEC and Hybrid ARQ/FEC [9]. A link level ARQ scheme, which is called hop by hop [10], utilizes a three flit deep retransmission buffer to handle transient faults. Due to the lack of buffers in the deflection router, implementing link level error control scheme is different from wormhole / virtual channel router with more input buffers.

In general, two kinds of fault tolerant routing are known as stochastic and deterministic. 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 into two classes: one can handle regular and irregular fault regions. A reconfigurable routing algorithm is proposed to route packets surrounding a faulty router A FON aware deflection router [11], which can tolerate convex and concave fault regions without deadlock and livelock.

3. Fault injection

3.1 Fault Model

Faults are assumed as faulty links which may be transient or permanent faults. For deflection router, the number of input ports should be equal to the number of output ports. In each router, a six bit fault vector is used to represent the fault state of six links. The faulty region can be any shape as long as it disconnects the network.

3.2 Fault Category

A fault as a deviation in a hardware or software component from its intended function can arise during all stages in a computer system design process: specification, design, development, manufacturing, assembly, and installation throughout its operational life. Most faults that occur before full system deployment are discovered and eliminated through testing. Faults that are not removed can reduce system‟ s dependability when it is embedded into the system. Hardware/Physical Fault that arise during system operation are best classified by their duration: Permanent, transient or intermittent.

- Permanent faults: Caused by irreversible component damage, such as a semiconductor junction that has shorted out because of thermal aging, improper manufacture, or misuse. Since it is possible that a chip in a network card that burns causing the card to stop working, recovery can only be accomplished by replacing or repairing the damaged component or subsystem.
- Transient faults: Triggered by environmental conditions such as power-line fluctuation, electromagnet tic interference, or radiation. These faults rarely do any lasting damage to the component affected, although they can induce an erroneous state in the system. According to several studies, transient faults occur far more often than permanent ones, and are also far harder to detect.

3.3 Fault injection

Fault Injection is defined by Arlat [12] as the validation technique of the dependability of fault tolerant systems which consists in the accomplishment of controlled experiments where the observation other system‟ s behavior in presence of faults is induced explicitly by the writing introduction (injection) of faults in the system. The fault injection techniques have been recognized for a long time as necessary to validate the dependability of a system by analyzing the behavior of the devices when a fault occurs. Several efforts have been made to develop techniques for injecting faults into a system prototype or model. Most of the developed techniques fall into five main categories: hardware based, software based, simulation based, emulation based and hybrid fault injection technique. Here we assumed the simulation based fault injection. Consists in injecting the faults in high-level models (most often, VHDL models). It allows early evaluating the system dependability when only a model of the system is available. Then it addresses different abstraction levels by using distinct description languages. A coherent environment should be provided to favor interoperability between the successive abstraction levels and to integrate the validation in the design process. Fault injection techniques provide a way for fault removal (the correction of potential fault tolerance deficiencies in the system) and fault forecasting (the evaluation of the coverage distribution – coverage factor and latency – provided by the tested system).Regarding the fault removal objective, the test should be directed to achieve a high coverage of the

possible configurations of the system to be validated. In this case, the selection of the faults/errors to apply and errors to propagate is primarily based on the analysis of the model describing the system and the information flow in the simulation of the system. Regarding the fault forecasting objective, the main alternatives are either to rely on statistical testing simulating a priorities relative distribution of the classes of faults/errors or to statistically process a posteriori the results of the test sequence. The data used to carry out the statistical processing may result from available file data on the distributions and/or from results of simulation experiments.

4. FAULT DIAGNOSIS

4.1 Fault Detection

SECDED hamming code , which can correct single error and detect double errors is used to encode the packet to perform fault diagnosis .partitioning the signals into smaller blocks and encoding separately is clearly a better strategy for improving performance, area and power consumption. The ECC storage array overheads are shown below [13]. From the table we understood that SEC- DED Hamming code is a better solution than other hamming codes based on the check bits and parity bits.

Table 1: ECC storage array overheads

	SEC – DED		SNC - DND		DEC – TED	
Data bits	Check bits	overhead	Check bits	overhead	Check bits	Overhead
16	6	38%	12	75%	11	69%
32	7	22%	12	38%	13	41%
64	8	13%	14	22%	15	23%
128	9	7%	16	13%	7	13%

The decoder will generate a syndrome, which contains the error information of the packet. If the decoder detects a single bit error in any one part of the encoding packet, it will correct it no matter which kind of fault it is. If it detects two bit error in any one part of the encoding packet for one cycle, which is considered as transient fault, it will require the router to retransmit the packet. If the syndromes of two consecutive received packets are same, which means the retransmitted packet contains the same two bit error, in order to check whether a link contains a real permanent fault or not. The router enters into test mode by applying six test vectors.

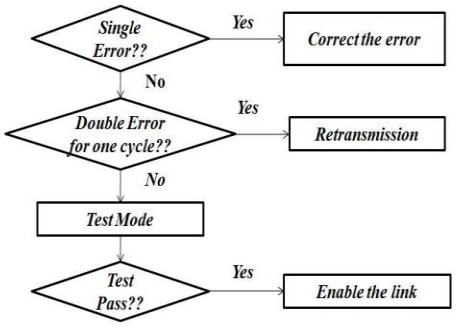

Figure 1: Fault diagnosis

The test process will be conducted atmost twice. The aim of detecting faults is to establish defined paths faster as possible. Single error can be corrected easily but when double error occurs one may need to take decision to establish a defined path. Occurring double error is considered as permanent fault. The final parameter considered in turn work is time taken to fault detection, hardware requirements and power. If the tests pass, the link will be enabled again.

4.2 Routing Design

In deflection routing, an incoming packet is always routed to a free output port even though it is far away from the destination. Because of its non-minimal routing characteristic, deflection routing can be easily modified to achieve fault tolerance. The reconfigurable fault tolerant deflection routing algorithm (FTDR) is based on a kind of reinforcement learning. It is a table based routing algorithm which reconfigures the routing table through Q learning and uses two hop fault information transmission to make efficient routing decision.

The algorithm can be divided into two parts: packets routed on the same layer and packets routed across layers. A reinforcement-learning-based deflection routing algorithm is used to route packets on the same layer [6]. Each switch contains an $n \times 4$ routing table which is constructed by the minimum number of hops to all destinations on the layer from four output ports (North, East, South, West)..The routing table is reconfigured by equation (1). Qx(d, y)denotes the minimum number of hops from x to d

through neighbor y. When x sends a packet to d through y, y will return 1 plus the minimum number of hops from itself to d (minzQyt−1(d, z)) back to x to reconfigure the corresponding routing table entry of x. Through this reinforcement learning method, after a learning period the routing table will be reconfigured to achieve fault-tolerance.

$$Qx(d, y) = 1 + \min zQyt-1(d, z)$$

For packets routed across layers, the switch makes routing decision based on the TSV state vectors. When a packet reaches a switch with the same row and column addresses but different layer as the destination switch, if the up/downlink of the switch is faulty, it will try to find an intermediate switch with a healthy vertical link at the same layer, which has a minimal Manhattan distance to the current switch, based on the TSV state vector. Then the packet will be routed to the intermediate switch according to the routing table of the layer. The FTDR switch is shown below [6].

The routing table is the main overhead of FTDR. As the size of system increases, the routing table size will increase significantly. In order to reduce the table size, we assumed the hierarchical based deflection routing (FTDR-H). The nxn mesh can be divided into several sub-regions with equal size. Each switch contains a local and region routing table. When a packet reaches a switch, the switch checks whether the destination is in the same region as the current switch or not. If it is, the switch makes routing decision based on the local routing table. If the destination is not in the same region, the switch makes routing decision based on the region routing table. The local and region routing tables are also updated.

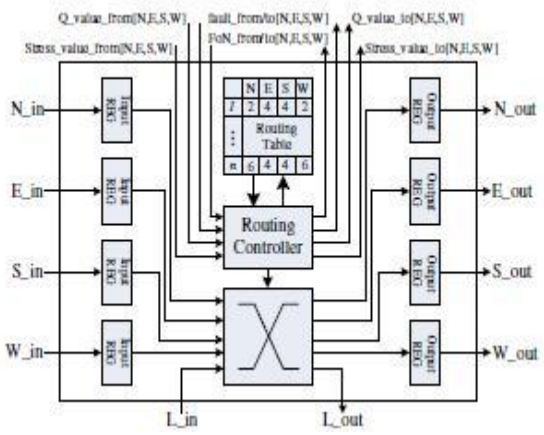

Figure 2: FTDR switch.

In FTDR algorithm, the routing table entry will converge to the minimum number of hops to destination in the presence of fault regions which do not disconnect the network. The structure of FTDR switch is shown in Fig. 2. Each switch contains an n x 4 routing table. Each entry of the routing table contains 6 bits, so the size of the whole table is n x 24 bits. An entry of all '1' denotes 1. The stress value, fault information and Q-value are transmitted between two switches. Routing controller makes routing decision based on the above three kinds of information.

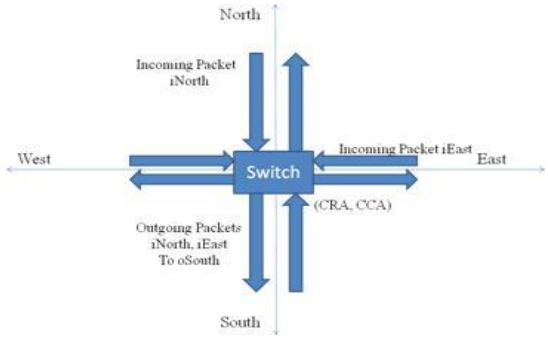

Figure 3: Example case for deadlock in direction

The routing table is the main overhead of FTDR. As the size of system increases, the routing table size will increase significantly. In order to reduce the table size, we assumed the hierarchical based deflection routing (FTDR-H). The nxn mesh can be divided into several sub-regions with equal size. Each switch contains a local and region routing table. When a packet reaches a switch, the switch checks whether the destination is in the same region as the current switch or not. If it is, the switch makes routing decision based on the local routing table. If the destination is not in the same region, the switch makes routing decision based on the region routing table. The local and region routing tables are also updated.

4.3 Avoiding Deadlock and Livelock

Bufferless design has been evaluated as an alternative to traditional virtual channel buffered routers. It is mainly due to two reasons: reduced hardware cost and simplicity in design. It may assume that energy consumption will be reduced by nearly 40% than buffered networks. FTDR is based on deflection routing which is inherently deadlock free since signals never have to wait in a router. Signals are never blocked due to the fact that there is less buffer queue to store the

signal. In deflection routers, the no of input ports is equal to the number of output ports. Thus an incoming signal will always find a free output port to go. It may always be deflected to a no preferred path but will never be blocked.

Livelock has to be avoided by limiting the number of misrouting. The FTDR makes routing decision according to the priority of a signal and routing. It will always give the highest priority to the previous signal.

5. CONCLUSION

The faults such as transient fault and permanent faults are addressed in System on Chip by considering one bit fault as transient fault and two bit fault as permanent fault. Here permanent fault is considered as link fault. And the one bit fault such as transient fault, the fault is considered to be present in data. We have found the optimum solutions for detecting and correcting the transient fault and permanent faults in System on chip. This paper also presented a method for fault tolerant solution for a buffer less system on chip, compliant with the following features: online fault diagnosis mechanism, scalability and Hybrid ARQ and FEC link level error control schemes. Due to this the proposed routing method may easily integrated into another SOC design that require fault tolerance.

REFERENCES

[1] 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., May2010, pp. 79–86

[2] 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.

[3] 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.

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

[5] T.Moscibroda and O.Mutlu, A case for bufferless routing in on-chip networks, ISCA – 36, 2009.

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

[7] 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. 1stIEEE/ACM/IFIP Int. Conf. Hardw./Softw. Codesign Syst. Synth., Oct.2003, pp. 188–193

[8] J. Raik, R. Ubar, and V. Govind, "Test configurations for diagnosing faulty links in NoC switches," in Proc. IEEE Eur. Test Symp., May2007, pp. 29–34

[9] A. Ejlali, B. M. Al-Hashimi, P. Rosinger, and S.G. Miremadi, "Joint consideration of fault-tolerance, energy-efficiency and performance inon-chip networks," in Proc. Design, Autom. Test Eur. Conf. Exhibit., 2007, pp. 1–6.

[10] 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.

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

[12] Folkesson P., Svensson S., and Karlsson J., "A Comparison of Simulation Based and Scan Chain Implemented Fault Injection," in Proceedings of28th International Symposium on Fault-Tolerant Computing (FTCS-28), Munich, Germany, pp.284-293, June 1998.

[13] Ikhwan Lee, Mehmet Basoglu, "Survey of Error and Fault Detection Mechanisms", April 2011.

[14] 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.