

Static and transient fault isolation in NOC using Error correction code and inbuilt test

D.Jagadeeswari, student M.E applied electronics
velammal engineering college, chennai, India

Mrs.s.lakshmikantham, Assistant professor, velammal
engineering collge, chennai, India

Abstract— Decline feature sizes have aggravated the effects of transient faults on Network-on-Chips (NoCs). Therefore, it is imperative to incorporate run-time transient fault detection and correction into the design of a NoC. In this paper 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 livelock. A reinforcement learning agent interacts with its environment in discrete time steps. The routing table to analyse the datas under real application workloads. To detect the multibit error the bch code is used.

Index Terms—Deflection routing, fault-tolerance, on-line fault diagnosis, permanent fault, 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. **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. There are two types of faults transient and permanent faults.

Fault model notation and error control scheme for switch to switch buses in a NoC scheme uses forward error correction and link level error control scheme using retransmission is proposed to handle the transient faults. Evaluation of on chip networks using deflection routing scheme route the packets to

neighboring routers immediately with buffering in the router. Fault tolerant flow control in on-chip networks schemes uses the flow control based methods combines the error control code with the retransmission mechanism to tolerate the transient faults. Transient and permanent error co-management method for reliable networks on chip scheme uses the detecting both transient and permanent faults at the mean time. Support efficient and fault tolerant multicast in bufferless network on-chip uses the three deflection routing based multicast schemes provide fault tolerant supporting schemes through the reinforcement learning to reconfigure the routing table to tolerate the permanent faults.

II. PROPOSED SYSTEM

Our objective is to find the transient and permanent faults at the run time. By using online fault diagnosis mechanism. Fig.1 shows the flow diagram of whole process.

Flowchart of online fault diagnosis system

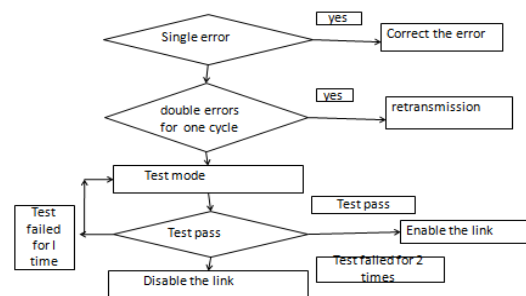


Fig. 1 Flowchart of the process

The decoder detects the single bit error in anyone of the encoding packet it will correct it. If it detects a two bit error in any one of the encoding packet, which is considered as transient fault and retransmit the packet. If the two

consecutive received packets are same, which is considered as permanent faults.

III. NOC ARCHITECTURE 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. 2. 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 value.

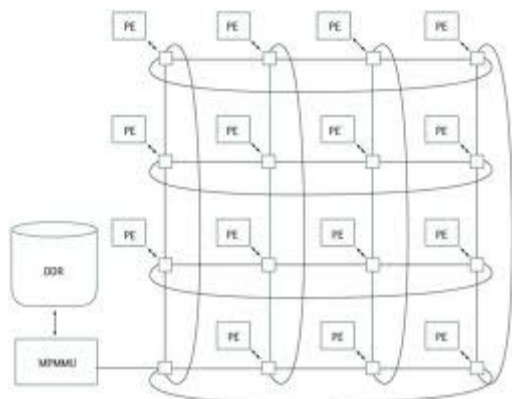


Fig 2. Noc architecture

The basic data transfer unit in this paper is a packet. The original packet format, which is compatible with a multicore NoC platform, is shown in fig 3.

1	12	12	9	80															
v	D	SA	H	payload															
Encoded head																			
23	23		22	22	22	22	22												
H1	H2		p1	p2	p3	p4	p5												
			11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11
			P1	p2	p3	p4	p5												

Fig.3 packet format

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. 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. 3. 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. For easy transmission the encoded with hamming the payload is again divided into the equal parts of 11 bits. This makes the transmission is easy.

IV. SECDED HAMMING CODE AND LINK LEVEL FAULT

Good error control performance requires the scheme to be selected based on the characteristics of the communication channel. Common channel models include memory-less models where errors occur randomly and with a certain probability, and dynamic models where errors occur primarily in bursts. Consequently, error-detecting and correcting codes can be generally distinguished between random-error-detecting/correcting and burst-error-detecting/correcting. Some codes can also be suitable for a mixture of random errors and burst errors.

If the channel capacity cannot be determined, or is highly variable, an error-detection scheme may be combined with a system for retransmissions of erroneous data. This is known as automatic repeat request (ARQ), and is most notably used in the Internet. An alternate approach for error control is hybrid automatic repeat request (HARQ), which is a combination of ARQ and error-correction coding.

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 one bit 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 correct the seven bit error and detect the 14 bit error encoded payload again divided for further codes.

VI. RESULTS

V. LINK LEVEL ERROR CONTROL SCHEME AND HARDWARE ARCHITECTURE

ARQ : This is an error control technique whereby an error detection scheme is combined with requests for retransmission of erroneous data. Every block of data received is checked using the error detection code used, and if the check fails, retransmission of the data is requested – this may be done repeatedly, until the data can be verified.

FEC: The sender encodes the data using an *error-correcting code (ECC)* prior to transmission. The additional information added by the code is used by the receiver to recover the original data. In general, the reconstructed data is what is deemed the "most likely" original data.

ARQ and FEC may be combined, such that minor errors are corrected without retransmission, and major errors are corrected via a request for retransmission.

The hardware structure of the ARQ scheme for one input port i ($i \in \{North, East, South, West, Local\}$) is shown in Fig. 4.

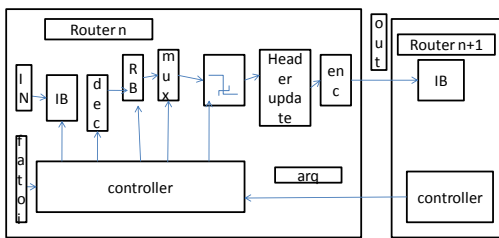
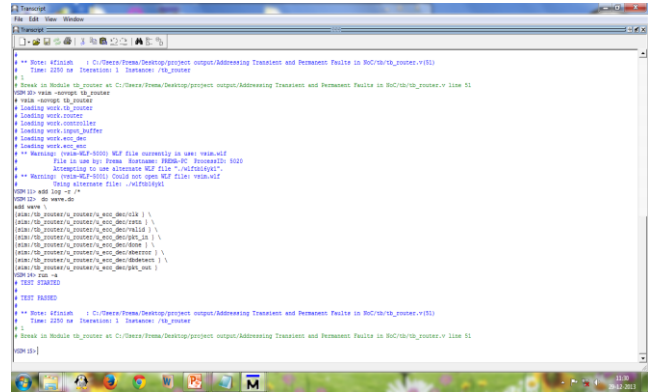


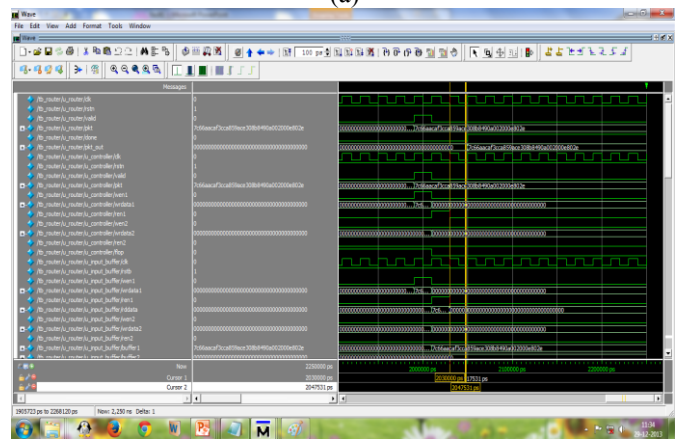
Fig.4 Hardware structure of link level error control scheme

Each input port of the router has an input buffer (IB_i) 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 (RB_i) is used to buffer the packet which may be retransmitted. After decoding, the packet will be written to RB_i . 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.

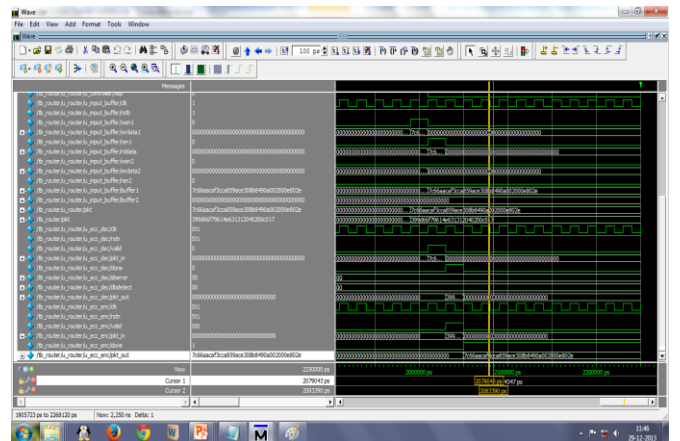
The experimental results of transient and permanent fault and encoded packet type as shown in the fig. (a)test pass or failed, (b) input buffer, controller, router packet.(c) ecc encoder and ecc decoder packets



(a)



(b)



(c)

VII. CONCLUSION AND FUTURE WORKS

In this paper, we provided a fault-tolerant solution for a bufferless NoC to protect it from both transient and permanent faults on the links. Specific contributions of this paper can be described as follows.

1) An on-line fault diagnosis mechanism utilizes SECDED Hamming code to detect both transient and permanent faults, and the encoding scheme can silently correct between 1–7 faulty bits and detect between 2–14 faulty bits, depending on the distribution of the faults over the link.

2) A hybrid ARQ/FEC scheme, which can achieve graceful degradation even at a high fault rate, is proposed to tolerate transient errors during transmission.

In future work BCH code to find the multi bit error

VIII. REFERENCES

- [1]. C. Constantinescu, "Trends and challenges in vlsi circuit reliability," *IEEE Micro*, vol.23, no.4, pp.14–19, 2003.
- [2] T. Moscibroda and O. Mutlu, "A case for bufferless routing in onchip networks," *Proc. IEEE Computer Society, Int. Symposium on Computer Architecture*, pp.196–207, 2009.
- [3]. M. Hayenga, N.E. Jerger, and M. Lipasti, "Scarab: a single cycle adaptive routing and bufferless network," *Proc. IEEE Computer Society, Int. Symposium on Microarchitecture*, pp.244–254, 2009.
- [4]. L.Wang, Y. Jin, H. Kim, and E.J. Kim, "Recursive partitioning multicast: a bandwidth-efficient routing for networks-on-chip," *Proc. IEEE Int. Symposium on Networks-on-Chip*, Washington, DC, USA, pp.64–73, 2009.
- [5]. P. Abad, V. Puente, and J.A. Gregorio, "Mrr: enabling fully adaptive multicast routing for cmp interconnection networks," *Proc. IEEE Computer Society, Int. Symposium on High Performance Computer Architecture*, pp.355–366, 2009.
- [6] Z. Zhang, A. Greiner, and S. Taktak, "A reconfigurable routing algorithm for a fault-tolerant 2d-mesh network-on-chip," *Proc. ACM/IEEE Design Automation Conference*, pp.441–446, 2008.
- [7] D. Fick, A. DeOrio, G. Chen, V. Bertacco, D. Sylvester, and D. Blaauw, "A highly resilient routing algorithm for fault-tolerant nocs," *Proc. Design, Automation and Test in Europe Conference and Exhibition*, pp.21–26, 2009.
- [8] A. Kohler, G. Schley, and M. Radetzki, "Fault tolerant network on chip switching with graceful performance degradation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol.29, no.6, pp.883–896, 2010.
- [9] D. Xiang, Y. Zhang, and J.G. Sun, "Unicast-based fault-tolerant multicasting in wormhole-routed hypercubes," *Journal of Systems Architecture*, vol.54, no.12, pp.1164–1178, December 2008.
- [10] J. Wu and X. Chen, "Fault-tolerant tree-based multicasting in mesh multicomputers," *Journal of Computer Science & Technology*, vol.16, no.5, p.17, 2001.