Smart Computing Review

Fault-Aware Dual-Layer Adaptive Error Control Technique for NoC

Waqar Amin, Naveed Khan Baloch, Jawad Ali Khan, and Muhammad Iram Baig

Department of Computer Engineering, University of Engineering and Technology Taxila, Pakistan/
waqar.ameen45@gmail.com, Naveed.khan@uettaxila.edu.pk, jaykhangr8@gmail.com,
ram.baig@uettaxila.edu.pk

*Corresponding Author: Naveed Khan Baloch

Received October 15, 2015; Revised November 21, 2015; Accepted December 21, 2015; Published December 31, 2015

Abstract: The network-on-a-chip (NoC) appeared as a promising solution to handle the communications requirements of the multiprocessor system-on-a-chip (MPSoC). As the complexity of designs rises and the technology scales down into the deep-submicron domain, the probability of errors in the NoC components increases. Fault tolerance is a vital aspect in designing NoC architectures for future MPSoCs. This paper proposes an adaptive fault-tolerant technique that is a hybrid end-to-end and hop-to-hop, offering benefits of both error control schemes, and introduces a fault-aware adaptive selective hop-to-hop error correction scheme. The proposed technique ensures improvement in reliability by reducing the latency of the network in low transient–noise conditions.

Keywords: Network-on-A-Chip, Dual Layer, Fault Tolerance, Error Control Code (ECC)

INTRODUCTION

In a system-on-a-chip (SoC), because of the technological advancement, many cores can be integrated onto a single chip, which amplifies concurrent memory access. In bus-based architectures, more than two cores are unable to communicate at the same time. This problem in bus-based architectures raises the need to work on the communications infrastructure. Researchers borrowed the concept of packet-based networks from computer communications to fulfill the high-performance requirements for communication between multiple cores. The concept thus formed was named network-on-a-chip (NOC), which provides modularity, scalability and ease in multiprocessor SoC (MPSoC) development [2].

In a NOC, the traditional bus-based architecture is replaced with a network, which comprises routers and a network interface (NI), to connect multiple intellectual property cores. NOCs establish the interconnection architecture of parallel routers, which combines hundreds to thousands of processing cores on a single chip. A NOC provides the advantages of structural regularity, modularity, scalability and high data throughput. The NOC comes with the problem of increased vulnerability to failure (because of some different noise sources) and reliability issues, which become more crucial in both current and upcoming technologies.

DOI: 10.6029/smartcr.2015.06.007

Progress in the future ultra-deep submicron field has assisted with the integration of a billion transistors onto a single chip. However, the shrinking size of transistors, their complexity, the lower supply voltages, the high frequencies and integration density cause faults in the network-on-a-chip [2]. Although most of the faults can be detected after manufacturing with the help of some special offline integrated circuit tests, a large proportion of faults occur during runtime. A fault-tolerant architecture is the best solution to handle faults at runtime. Depending upon duration, faults can be classified into different types, such as permanent, intermittent and transient faults. Permanent faults caused by physical changes, where wires and transistors are permanently shorted or open, also cause delay in communications, which eventually results in incorrect logic. The factors that cause transient faults are alpha-particles, electrostatic discharge, power supply, electromagnetic interference, and interconnect noise. Intermittent faults are more like transient faults and are caused by the same environmental changes. The criteria to distinguish intermittent faults from transient faults are that intermittent faults occur repeatedly at the same location, and errors created by intermittent faults occur in bursts. There are many techniques to correct transient errors that occur in a NOC.

The reliability requirement can be addressed with a hop-to-hop technique in the data link layer and in an end-to-end fashion in the transport layer. Both end-to-end and hop-to-hop have both benefits and drawbacks. An end-to-end scheme is more reliable in low transient–noise conditions and achieves low latency, compared to hop-to-hop, which is highly reliable in high noise conditions but increases latency, even if there is no error in the network. This research work proposes a technique that uses the benefits of both hop-to-hop and end-to-end error control and minimizes the drawbacks with both techniques by introducing a selective hop-to-hop error correction scheme, which decodes and encodes packets after every second hop. The proposed technique is also adaptive, and can change error correction mode according to noise conditions. Utilizing the characteristics of a dual layer provides improvement in reliability of the NoC by correcting three error bits in the network.

The rest of the paper is organized as follows. The section on related work reviews the research work of different authors in hop-to-hop and end-to-end error correction. The proposed technique is explained in the Methodology and Research section. The results are given in the section titled Experimental Analysis, while the last section summarizes and concludes this research work.

RELATED WORK

Yu and Ampadu [1] proposed a framework that allows dual-layer cooperative error control in a network-on-a-chip in order to simultaneously improve reliability, performance and energy efficiency. They combined a configurable routing algorithm in the network layer and an adaptive error control coding scheme in the data link layer. A multi-layered error correction and detection scheme with adaptive and configurable encoding and decoding on the switch-to-switch (s2s) layer, avoiding faults from accumulating alongside the route of the packet, was presented [2]. Shamshiri [3] proposed a complete solution for end-to-end error correction and runtime defect diagnosis for a NoC. An interleaved error-locality-aware code that efficiently corrects both random and burst errors was presented. For 64-bit wide network links, the authors used interleaving with the proposed code, each of which supports 16-bit data, to correct as many as two random errors or 16 adjacent errors.

Murali [4] conducted research to investigate the power–performance efficiency of several error recovery schemes applied in the network and data link layers to handle transmission errors in on-chip networks. Comparison of transient fault effects in an asynchronous NoC router and a synchronous one was presented [5]. The experiment was a simulation-based fault injection method to assess the fault-tolerant behavior of both architectures. Different fault models, such as Crosstalk, SEU, and SET, have been applied in both architectures to evaluate their robustness. Ebrahimi [6] presented an adaptive routing algorithm for on-chip networks that provides a wide range of alternative paths between each pair of source and destination switches.

The implementation of the algorithm provides the best usage of all allowable turns to route messages more adaptively in the network. A fault-tolerant approach based on using the shortest paths was also described by Ebrahimi [7]. This method maintains the performance of networks-on-a-chip in the presence of faults. To avoid using non-minimal paths, the router architecture is slightly modified. According to this algorithm, only the shortest paths are used by packets in the presence of a fault. This results in retaining the performance of NoCs in faulty situations. Lehtonen [8] discussed fault-tolerant methods for NoCs and the origin of faults in modern technologies, and explained the classification of transient, intermittent and permanent faults. Lehtonen also discussed different error control coding techniques used in both network and data link layers. Fault-tolerant methods in networks-on-a-chip are comprehensively discussed by Radetzki [9]. Jantsch [10] analyzed link-level low-power encoding techniques, concluding that they expend several times more power than no encoding at all, if normalized for the same performance.

▪ End-to-End Error Correction

Network-layer error control code (ECC) is implemented only in the network interface and does not detect or correct errors during each hop on the route. End-to-end error control schemes do not increase link width between hops and router complexity; therefore, power consumption of end-to-end ECC is less than that of hop-to-hop ECC when the number of retransmissions, route length and error rate are small. Because the link switching power portion of total power rises as the technology scales down, the average power consumed by effectively transferring a packet over a long path may surpass the power consumed by hop-to-hop error control. Furthermore, a longer path may accumulate more errors because of the absence of hop-level error recovery. Thus, end-to-end ECC requires strength to achieve the same performance as hop-to-hop ECC. End-to-end ECC schemes require a possibly large codec overhead in the network interface. Waiting for acknowledgement packets in end-to-end control also results in an increase in average latency. End-to-end hybrid error detection and correction schemes correct packets having single errors, and requests retransmission only when multiple errors are detected.

▪ Hop-to-Hop Error Correction

Hop-to-hop error detection and correction is accomplished in the data link layer. In hop-to-hop ECC, error detection and correction of each packet/flit is performed during each hop. The output/input port of each router contains an ECC encoder/decoder. In hop-to-hop ECC, no codec is required in the NI. The router is for a mesh/torus topology, and the details of other routing blocks and the buffers are not emphasized here. Hop-to-hop schemes detect the error and recover the corrupted packet/flit within the present hop. This instantaneous error recovery avoids error accumulation, minimizing the need for powerful error control codes. However, hop-to-hop ECC increases complexity in router design. The communications energy for long on-chip paths becomes comparable to computation energy. The redundant encoding/decoding at each hop wastes energy.

▪ Router Design

Figure 1a, illustrates a block-level architecture for a five-port router. It generally consists of a crossbar, input ports and output ports, a routing control unit and input/output first-in, first-out (FIFO) buffers. Input ports on each side save data into a buffer, and each output port picks data from three input buffers and the processor core, based on the routing control unit. Router design for the proposed technique is shown in Figure 1b. To perform end-to-end ECC at the network interface, an end-to-end ECC codec is added, accompanied by a unit that contains counters to accumulate error detection results and a mode switch controller. Similarly, in the data link layer, a hop-to-hop ECC codec is added to perform ECC on each hop, plus counters with a mode switch controller. The counters count the packets with a particular number of errors, and the mode switch controller switches the mode when the counters reach a predefined threshold.

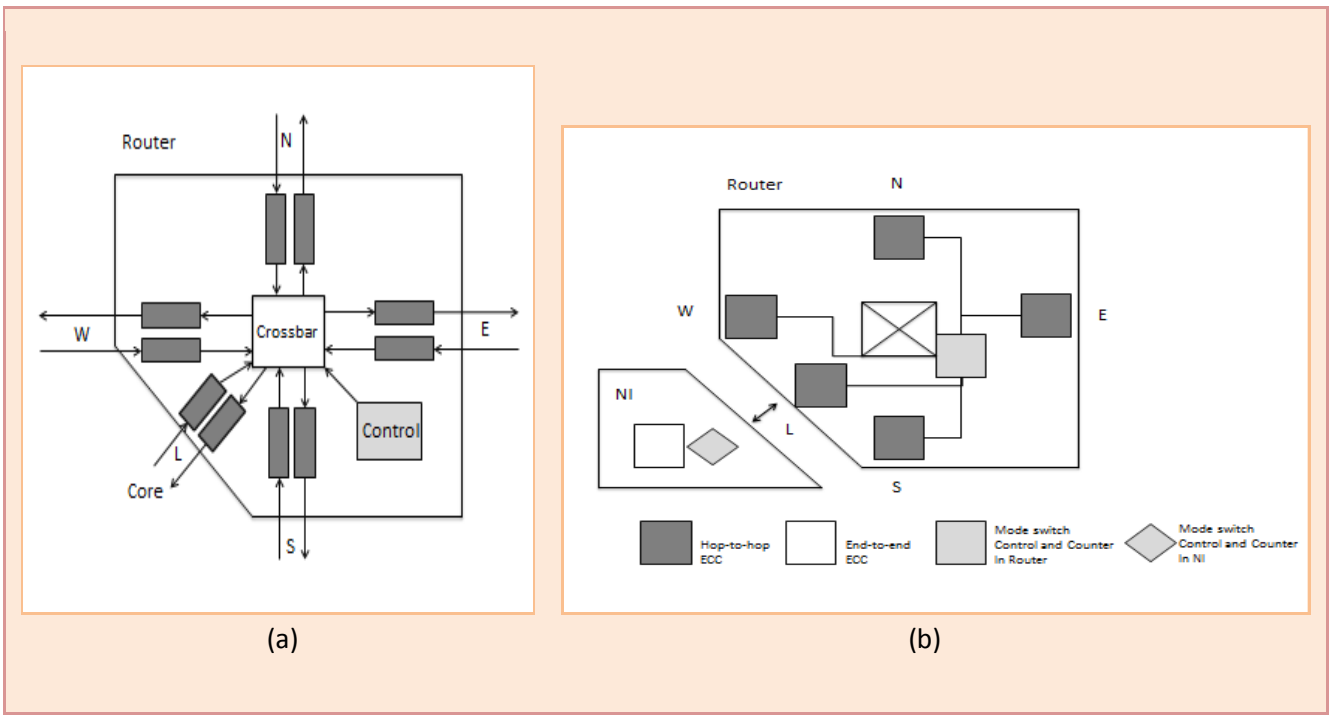


Figure 1. (a) Generic router design, and (b) router design for the proposed technique

METHODOLOGY AND RESEARCH

In the proposed work, the available resources of different NoC layers are combined to explore the advantages of both end-to-end and hop-to-hop ECC under inconsistent noise conditions. The proposed work presents a "fault-aware dual-layer adaptive error detection and correction scheme" for NoCs. A JTEC (Joint Triple Error Correction) technique [12] is used for error detection and correction. The JTEC error correction technique is an efficient triple-error correction technique with low encoder/decoder delay and low latency. In Figure 2, a state diagram of the proposed technique is given. The end-to-end ECC schemes are implemented in the network layer and only executed in the NI. If the network operates in end-to-end mode, and the number of errors detected at any network interface is equal to 1, then the network continues to operate in end-to-end mode. If the network operates in end-to-end mode and the number of errors detected at any NI equals 2, a counter starts recording the number of packets with two errors.

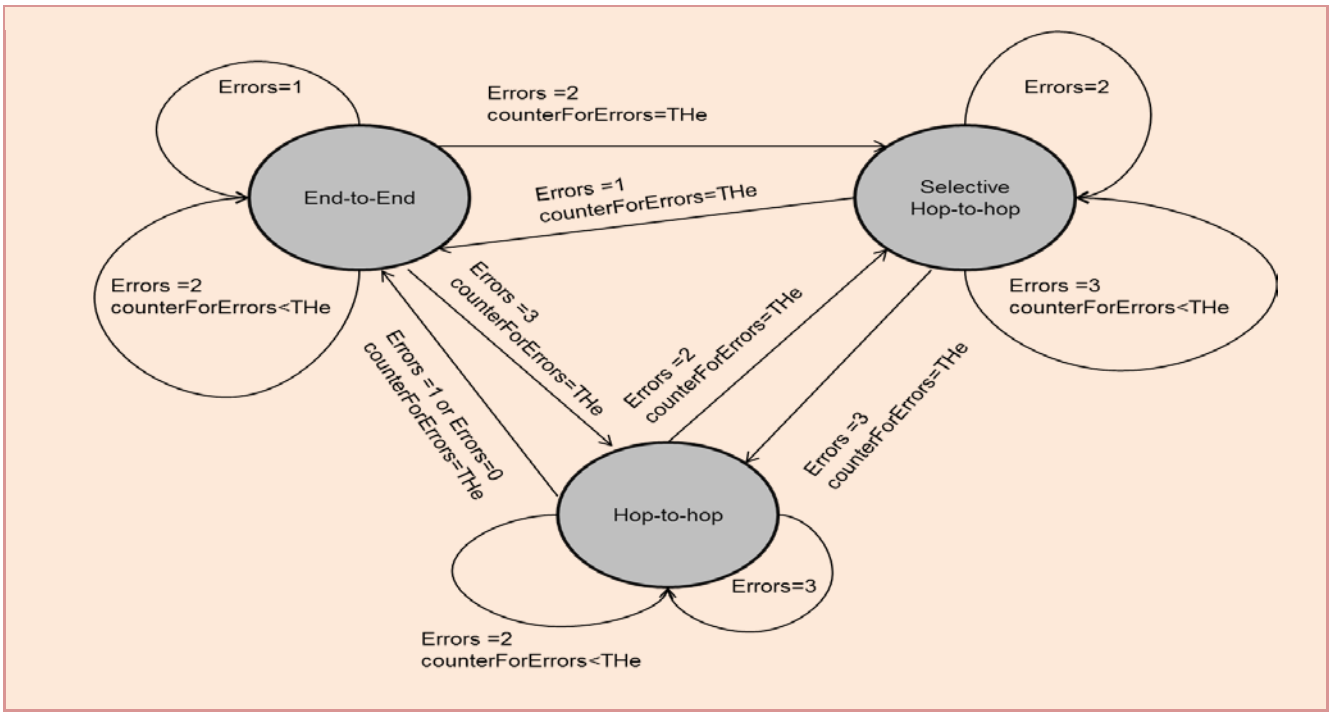


Figure 2. State diagram of the proposed technique

When the counter for double errors reaches a specific threshold, end-to-end mode is switched to selective hop-to-hop mode. Similarly, if there is triple-error detection in packets, and this situation persists for a predefined time, then end-to-end mode switches to hop-to-hop mode to deal with the high noise conditions. The counters prevent the network from switching immediately and ensure that the network only changes its ECC mode if the noise condition persists for a given time, because unnecessary switching may cause extra latency and energy consumption in the network. When the network is in selective hop-to-hop ECC mode, it can switch to either end-to-end or hop-to-hop ECC mode, or keep on operating in selective hop-to-hop mode, depending upon the noise conditions. If the network is error-free, or the number of errors detected is equal to 1 for a specific threshold, then the network switches the mode to end-to-end from selective hop-to-hop ECC mode. Similarly, if the network encounters high noise conditions for a given threshold, the network switches from selective hop-to-hop to hop-to-hop ECC mode in order to achieve more reliability. To ensure the reliability of the network, it keeps operating in hop-to-hop mode as long as there are high noise conditions. As the noise in the network decreases, the network gradually switches its ECC mode to selective hop-to-hop and then to end-to-end ECC mode. Figure 3a illustrates a situation where the network experiences only one error for a specific threshold and operates in end-to-end mode.

In Figure 3b, the network operates in hop-to-hop mode when it experiences high noise conditions and the number of errors detected is 3. Similarly, Figure 3c illustrates selective hop-to-hop mode where errors are detected and corrected on alternate hops.

The proposed work provides reliability by using a triple-error correction technique in the form of JTEC. Typically, end-to-end mode provides satisfactory reliability for low transient noise conditions at low energy consumption. In comparison, hop-to-hop mode prevents error accumulation through the long path between source and destination, and is more efficient than end-to-end mode in high transient noise conditions. Selective hop-to-hop mode provides reliability in the network and also improves latency by performing error detection and correction on alternate hops. Therefore, in variable transient noise

conditions, collaboration between end-to-end, selective hop-to-hop and hop-to-hop modes improves average latency, energy consumption and reliability.

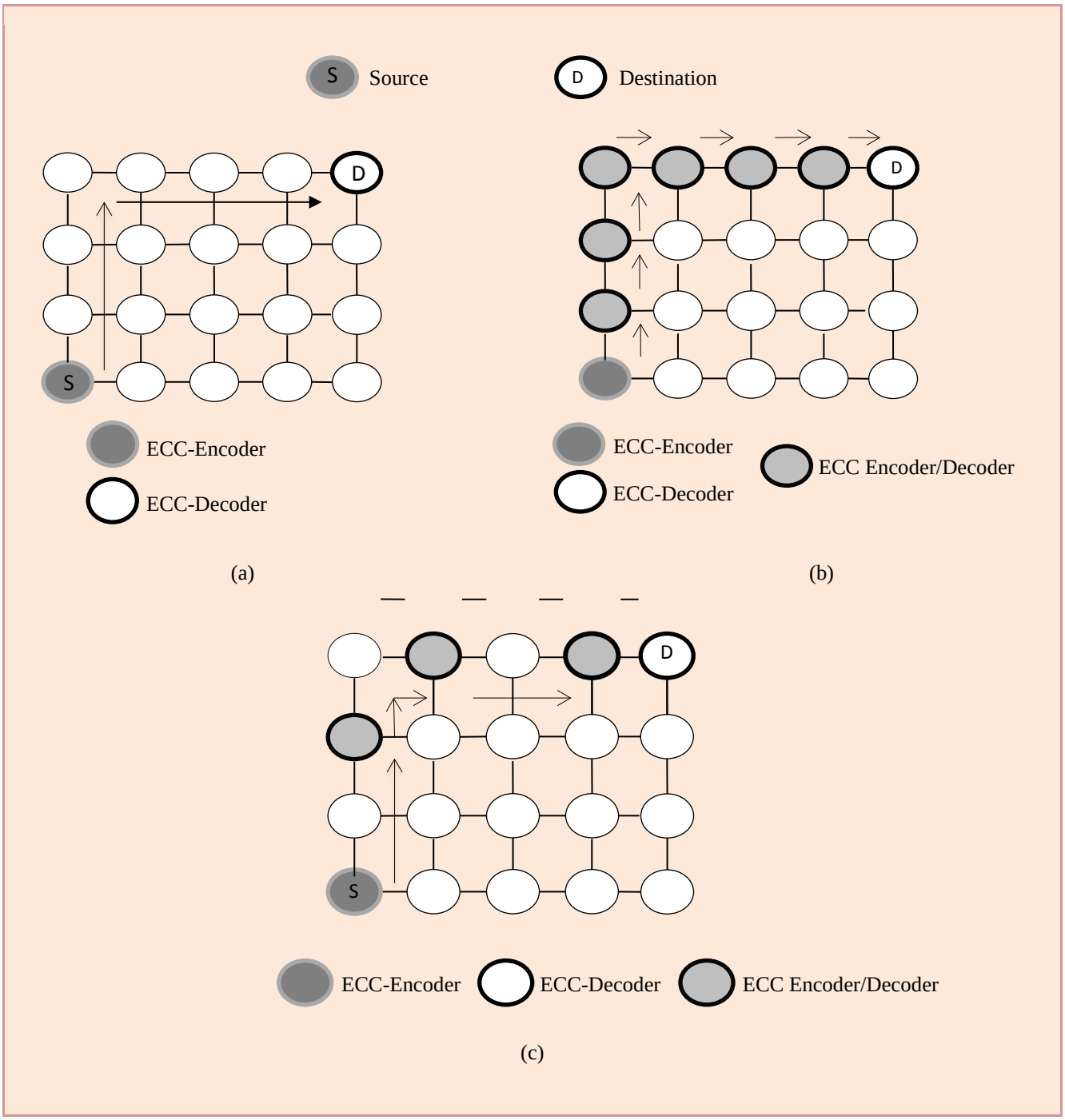


Figure 3. (a) End-to-end ECC when errors=1; (b) hop-to-hop when errors=3, and (c) selective hop-to-hop when errors=2

EXPERIMENTAL ANALYSIS

- ### Average Latency

Latency is defined as the time interval between a packet header leaving the source and that packet's tail reaching the destination. Average latency per useful packet is a metric to evaluate the latency of different methods. A useful packet is a

packet received at the destination error-free after error control coding. Latency is affected by both the packet injection rate and error, since both are likely to raise the number of packets that fight for the same resources. Latency for end-to-end, hop-to-hop and selective hop-to-hop ECC is given in 5.1, 5.2 and 5.3, respectively.

$$\boldsymbol{Avg\ Le2e} = \frac{(\boldsymbol{L_{NA}}+\boldsymbol{h}*\boldsymbol{L_r}+\boldsymbol{h}*\boldsymbol{L_{ecc}}+\boldsymbol{L_{link}})}{\boldsymbol{No\ of\ errors}} \quad \textbf{5.1}$$

$$\text{Avg Lh2h} = \frac{(\text{h*L}_\text{r}+\text{h*L}_\text{ecc}+\text{L}_\text{link})}{\text{No of errors}} \quad \textbf{5.2}$$

$$\boldsymbol{Avg\ Lsh2h} = \frac{(\boldsymbol{h}*\boldsymbol{L_r}+\boldsymbol{h}*\boldsymbol{L_{ecc}}+\boldsymbol{L_{link}})}{\boldsymbol{No\ of\ errors}} \quad \textbf{5.3}$$

where

$\boldsymbol{L_{NA}} = \boldsymbol{Latency\ in\ the\ Network\ Adapter}$

$\boldsymbol{h} = \boldsymbol{Number\ of\ hops}$

$\boldsymbol{L_r} = \boldsymbol{Latency\ in\ the\ router}$

$\boldsymbol{L_{ecc}} = \boldsymbol{Latency\ in\ the\ encoder/decoder}$

$\boldsymbol{L_{link}} = \boldsymbol{Latency\ in\ the\ link}$

The processing elements inject packets separately, and the destination of a packet is determined at random, which results in a uniform traffic pattern. Each router takes five clock cycles to process and route a flit, and a Golay encoder/decoder takes one clock cycle to generate encoded/decoded output. When a flit is injected by a resource through a resource link into the NoC, it takes one clock cycle to encode the payload of the flit, and similarly, one clock cycle on the destination side to decode it. Overall latency in the presence of coding is increased by only two clock cycles. The number of extra clock cycles used in the selective hop-to-hop scheme is given in 5.4.

$$\text{Extra clock cycles for sh2h} = ((\text{round}(\text{hop\_count}/3)*2)-2) \quad \textbf{5.4}$$

With an increasing injection rate, latency increases due to the congestion of routers, as shown in Figure 4.

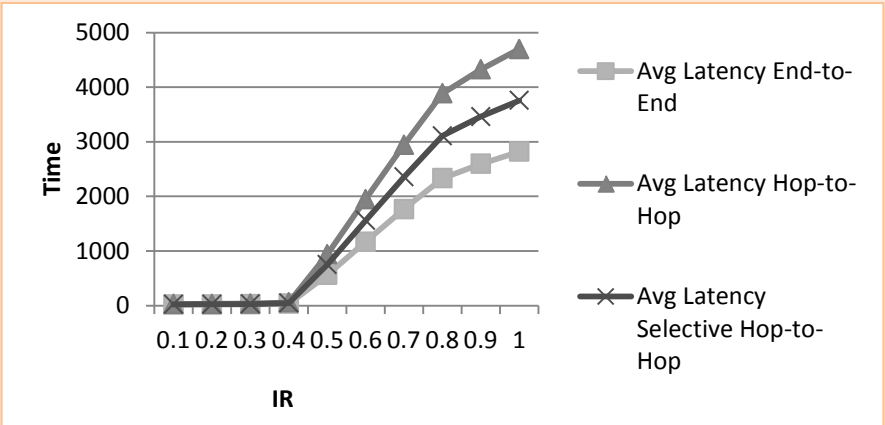


Figure 4. Average latency comparisons

▪ Average Throughput

The experiment evaluated the packet count with different numbers of link failures. Packet count includes the percentage of correct packets received at the destination out of the total number of packets injected. Packets were injected at different injection rates. With nonexistence of faulty links, the correct packet count is 100%. An increase in the percentage of faulty links results in degradation of the correct packet count. The correct packet count for IR 1.0 with increasing numbers of faulty links (%) is presented in Figure 5.

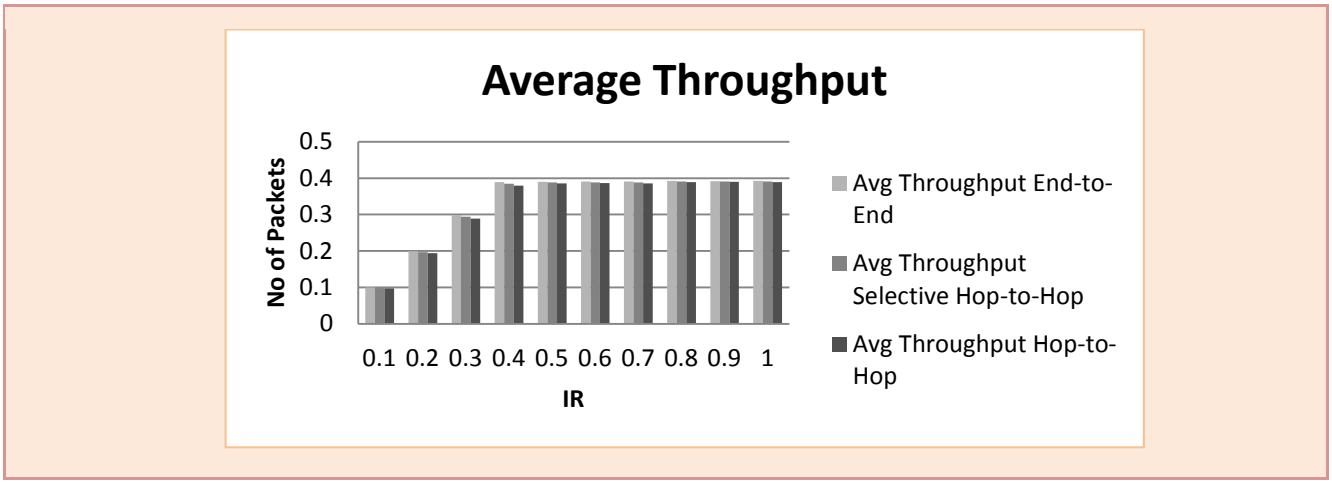


Figure 5. Average throughput comparisons

CONCLUSIONS

The network-on-a-chip is increasing in importance as an innovative solution to integrate several cores in a SoC. The presence of faults in NoC interconnects influences its performance. A fault-aware adaptive dual-layer error correction coding technique is proposed in this work to enhance the reliability of network-on-a-chip design by reducing the probability of the occurrence of errors. The outcome of an analysis of correct packet counts and average latency in an end-to-end NoC and a hop-to-hop NoC using the proposed scheme is presented. The proposed scheme has the ability to successfully correct three errors. Overall latency is increased by only two clock cycles, and for 4.1% faulty links, up to an 88% correct packet count is achieved. Link utilization, hop count and throughput remain constant throughout the simulations. Future work includes study into mitigating the effects of different fault models in NoC design. The proposed technique can be made more efficient by reducing the number of encoding bits.

References

[1] Q. Yu, P. Ampadu, "Dual-Layer Cooperative Error Control for Reliable Nanoscale Networks-on-Chip," *22nd IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, 2011.

[2] Travis H. Boraten, "Runtime Adaptive Scrubbing in Fault-Tolerant Network-on-Chips (NoC) Architectures," *IEEE 31st International Conference on Computer Design(ICCD)*. Article (CrossRef Link)

[3] S. Shamshiri, "End-to-End Error Correction and Online Diagnosis for On-Chip Networks," *International Test Conference* , 2011. Article (CrossRef Link)

[4] S. Murali, "Analysis of Error Recovery Schemes for Networks-on-Chips," 2012.

[5] Pooria M. Yaghini, "Investigation of transient fault effects in synchronous and asynchronous Network on Chip router" *Journal of Systems Architecture*, 2010

[6] M. Ebrahimi, "HARAQ: Congestion-Aware Learning Model for Highly Adaptive Routing Algorithm in On-Chip Networks," 2012.

[7] M. Ebrahimi, "Minimal-Path Fault-Tolerant Approach Using Connection-Retaining Structure in Networks-on-Chip," *IEEE*, 2013. Article (CrossRef Link)

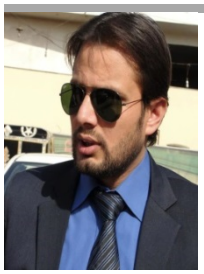
[8] T. Lehtonen, "On Fault Tolerance Methods for Networks-on-Chip," *TUCS Dissertations*, no. 122, Oct. 2009

[9] Martin Radetzki, "Methods for Fault Tolerance in Networks-on-Chip," *ACM Computing Surveys*, vol. 46, no. 1, Article 8, Oct. 2013. Article (CrossRef Link)

[10] A. Jantsch, "Power analysis of link level and end-to-end data protection in networks on chip," 2005.

[11] V. Rantala, T. Lehtonen, "Network on Chip Routing Algorithms," *TUCS Technical Report*, No. 779. Aug. 2006.

[12] A. Ganguly, "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, Nov. 2009. Article (CrossRef Link)



Waqar Amin completed his bachelors & Masters in Computer Engineering from University of Engineering and Technology Taxila Pakistan, in 2007 and 2015, respectively. He has vast experience of embedded system designing and currently his research interests include reliable communication for on-chip networks.

Naveed Khan Baloch is currently working as Assistant Professor, Department of Computer Engineering, University of Engineering and Technology Taxila. He completed his Bachelors in Computer Engineering and Masters in Computer Engineering, in 2007 and 2011, from University of Engineering and Technology Taxila Pakistan. His research interests include embedded systems, digital design and Network-on-a-chip.

Jawad Ali khan is currently working as lecturer in COMSATS Institute of Information and Technology Wah, Pakistan. He completed his Bachelors in Computer Engineering from EME College Islamabad and Masters from University of Engineering and Technology Taxila Pakistan. His research interests include mobile App development, Web development, Desktop applications development and on-chip networks.

Muhammad Iram Baig is currently working as Chairman of Computer Engineering Department, University of Engineering and Technology Taxila. He completed his Bachelors, Masters and PhD in Electrical Engineering, from University of Engineering and Technology Taxila Pakistan. His research interests include embedded systems, digital design and microprocessor system design. He is also the co-founder of ASIC research center UET Taxila. He has supervised many Masters and PhD thesis and published papers in reputed international conferences and journals.