Research Journal of Applied Sciences 9 (1): 48-52, 2014
ISSN: 1815-932X


# FPGA Based Adaptive Resource Efficient Error Control Methodology for Network on Chip

[1]M. Deivakani and [2]D. Shanthi
[1]Department of Electronics and Communication Engineering,
[2]Department of Computer Science Engineering,
PSNA College of Engineering and Technology, Dindigul, India

**Abstract:** This research work proposes resource efficient and secured network on chip router using error control schemes. The proposed method combines the Cipher block encryption based parallel crossbar methodologies of the NoC data link and network layers to efficiently gives error control strength in variable network topology conditions. The proposed method significantly minimizes hardware utilization when compared to other earlier research. This can be achieved by implementing parallel cross bar architecture with Cipher block based ECC Coding Method in NoC. The proposed system uses Modelsim Software for simulation purposes and Xilinx Project Navigator for synthesis purposes.

**Key words:** Error control, cipher, data link, residual packet, interleaving

## INTRODUCTION

Network on Chip (NoC) provides a flexible communication infrastructure to be adapted for various applications with various workloads at the same time maintaining performance and simplicity. But System on Chip (SoC) requires a longer time to respond and to map the applications. Also, SoC lacks an efficient usage of the bus systems in its design. New system has to depend on reuse of components and architectures. NoC provides a good solution for flexible products which may be reconfigurable for applications with a combined task for applications with strict time to market requirements. There is always a trade-off between performance and simplicity. Generality provides hardware reuse while performance is achieved by using application-specific structures to increase efficiency and decrease the processing time. Implanting a network instead of global wiring imparts certain advantages in performance and structure. Hence, NoC is the solution for several technological, economical and productivity problems.

In reconfigurable embedded systems, NoCs provide links which interconnect processors, memories and other Intellectual Property (IP) components. NoCs combine performance with design modularity, allowing the integration of many design elements on single chip die. NoCs have been applied to SoC to avoid the complexity of on-chip communications. On-chip communications become more liable to transient noises, like cross-talk and external radiations.

In state of art work, they introduced error minimizing methodologies to reduce the hardware errors in network router. This type of errors varies linearly in accordance with time unit. This also affects the bandwidth optimization and power consumption. In other ways, an adaptive error control techniques are used to access the network utilization efficiency. Earlier researches (Lehtonen *et al*., 2007; Ganguly *et al*., 2008; Duan *et al*., 2009; Fu and Ampadu, 2010) employs ECC at the data link layer (hop to hop ECC) which detects errors at each link. This type of error control techniques reduces the residual error rate but it also reduces the available utilization bandwidth and increases the power consumption.

In the modification to existing error control techniques, researchers propose a network layer error control methodology. This type of error control mechanism preserves the packet transmission and reception instead of error detection and correction of such type of errors at various noise conditions. Researchers propose an efficient framework which adapts ECC strength across data link and network layers thus improving the energy efficiency and link reliability whilst maintaining performance.

**Literature rewiew:** Murali *et al*. (2005) presented a Dynamic Error Recovery algorithm to be used on NoCs. Their technique provided a large error rate and also supports only a less baud rate. Banerjee *et al*. (2009) proposed an energy-aware routing algorithm for NoCs. The algorithm uses a complex architecture which is the main drawback.

**Corresponding Author:** M. Deivakani, Department of Electronics and Communication Engineering,
PSNA College of Engineering and Technology, Dindigul, India

Fu and Ampadu (2009) proposed a Hybrid ARQ scheme for the network interconnects. Their methodology used type2 hybrid ARQ features and employed hamming product codes. The experimental results showed low transmission speed and requires larger gates for a better performance.

Lee *et al.* (2006) suggested an efficient and low power NoC design to be used for low power applications. Their design made use of a hierarchical star topology based NoC for low power applications. The design used a complex architecture.

Xu *et al.* (2009) presented a wave-pipelined interconnect architecture for NoCs. Their results show that their design consumed high power and required more number of gates and slices. Yu and Ampadu (2010) presented a simulator design for NoCs which employed Parallel Error Control Algorithm. Their method required higher power consumption.

Yu and Ampadu (2010) has proposed error controlled network router and simulated using flexible parallel simulator. But they focused only four ports configuration and buffer allotment for each port became variable memory unit and it consumed high power consumption.

Lehtonen *et al.* (2007) has proposed error control schemes such as forward error correction methodologies for large scale network on chip routers. This study only concentrated on error detection and correction process.

Ganguly *et al.* (2008) has designed a low power network on chip router for reliable network architecture. It also supported multiple error correction coding for this design and verification. Duan *et al.* (2009) has designed and verified a CODEC based routing schemes which avoids cross talk and other interference.

## MATERIALS AND METHODS

The proposed coding techniques merges Cipher block encryption with inter-leaving methods to adaptively adjust to the error control operation during varying noise conditions. The proposed method lowers the residual packet error rate by implementing interleaving techniques with Cipher block based ECC Coding Method in NoC.

**NI architecture:** The Network Interface (NI) is a part of the proposed ECC mode switching protocol. It counts the number of errors found during a time interval $T_c$ and requests for an ECC mode switch if necessary. The switching request architecture is shown in Fig. 1. If the node is present in the Single Layer (SL) state, the total number of detected errors after a time interval $T_c$ beyond the end error threshold makes the request. To distinguish

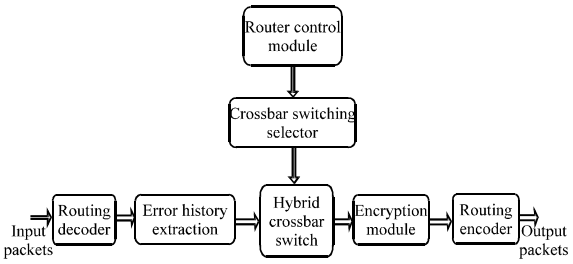


Fig. 1: Architecture of router for a dual layer ECC

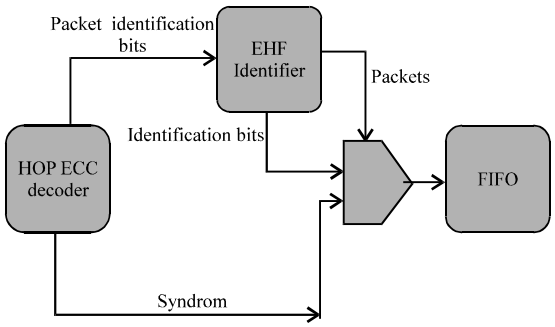


Fig. 2: The EHF update system

real favourable noise conditions and the consequence of adding hop to hop ECC, researchers check logic "1" in the Error History Flit (EHF) which stores the error detected at every hop along the route. The EHF is created by the NI when it packetizes the bit stream into a packet. This EHF starts with a unique header bit to be recognized by the router and is protected with the same error protection code. The remaining elements of EHF are written at each hop as "1" for error detection and "0" for no error. Whenever a non-zero EHF is present, the value of ripple counter is incremented. The value of modular counter is incremented by 1 at each clock cycle and this resets the ripple counter every $T_c$ cycles. The number of D-flip flops needed in the modular counter and the ripple counter are equal to $[\log_2(T_c)]$ and $[\log_2(\text{end error threshhold})]$, respectively.

**Router architecture:** Researchers have designed a router which supports the proposed modified dual layer ECC and is shown in Fig. 2. The designed router consists of ECC mode switch, mode propagation counter and Error History Flit (EHF) update system. A router typically has 5 input and output ports, namely, North, South, West, East and local. The information extractor obtains the destination address from the packet header flit. The arbiter provides connection between input and output ports. The arbiter makes use of NACK feedback from neighbouring routers (NACK_in) to control the First In First Out (FIFO) packet.

In the Dual Layer (DL) state and pre-SL state, $S_1 = 1$ is transmitted to neighbouring routers to make sure all the routers use hop to hop ECC. The SL and Pre-DL states propagate $S_1 = 0$ to turn off the hop to hop ECC. The propagation end signal comes from the mode propagation counter. The input width for that counter is $[\log_2(T_{propagation})]$. The width of EHF is the same as that for an uncoded flit. For a 32-bit flit width, the EHF can store the error history for a route length of up to 24 hops. In the network interface, the error history flit is decoded with an OR operation masked by the hop to hop ECC.

The EHF update system is explained in ECC switching algorithm. The router consists of a crossbar switch mechanism as shown in Fig. 3a. The 2×2 crossbar architecture consists of Input Port Controllers (IPC), Output Port Controllers (OPC) and a control unit. Each IPC adds an Internal Header (IH) to the incoming packet. The IH contains the forwarding route of the packet. If IH = 0 then the crossbar acts as a parallel architecture. Similarly, if IH = 1 then the crossbar acts as a cross-over crossbar. The architectures of a crossbar working in parallel and cross-over mechanism are shown in Fig. 3b. To avoid mode switching oscillation, only network-layer requests can change the DL state to the Pre-SL state. The time spent in the Pre-SL state is used to propagate the request through the rest of the network using the dual-layer error control. If no dual-layer mode is requested by the local node and neighbour nodes, the network enters the SL state. The dual-layer ECC mode is triggered when the total number of errors detected by any node in $T_c$ cycles exceeds the error threshold. $T_c$ is the time interval in which the NI counts the total number of erroneous packets. The number of error-packets is given by:

$$P_e = 1-(1-e)^{\omega_p h} \quad (1)$$

Where:

$\omega_p$ = The total number of bits per packet

$\varepsilon$ = The Bit Error Rate (BER)

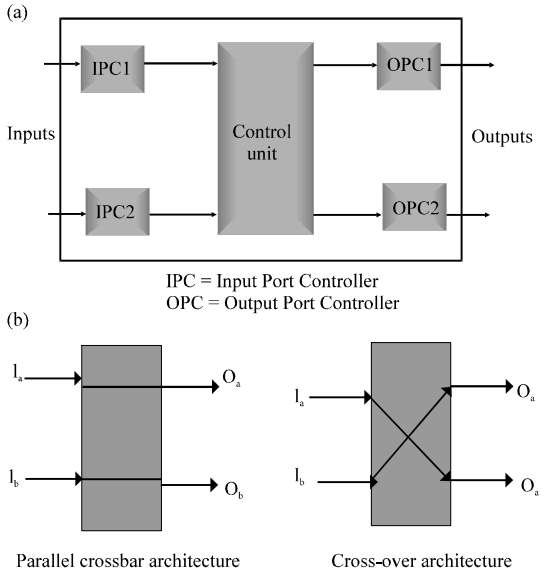


Fig. 3: a) Architecture of 2×2 crossbar; b) Operation of a crossbar based on IH

**Dual layer ECC encoder:** The end to end ECC encoder is located in the network interface. Figure 4 shows the overall process of the proposed encoder architecture. The binary bit stream of each packet is arranged into an array where each flit is a column. There are 3 stages for an encoding process.

**Stage 1:** Each flit is encoded by a column encoder (in this research, researchers use a systematic linear code) to generate the Flit Check Bits (FCB).

**Stage 2:** The row vector is encoded by a row encoder to produce the Packet Check Bits (PCB) using systematic linear codes.

**Stage 3:** To generate the Checks on Checks (CoC), researchers encode the PCB by the column encoder.

The product codes in the dual-layer ECC require an additional step to packetize the PCB and CoC into a new packet. To reduce the energy consumption for transmitting parity check bits, researchers transmit a coded packet first and then transmit the packet composed of PCB and CoC if a retransmission request is received.

**Key generation:** This Security algorithm for router architecture uses the 10 bit length of initial key for encoding or encrypting the receiving packets in each port of the router. The two corresponding sub keys are generated by using this initial key. The generation of key

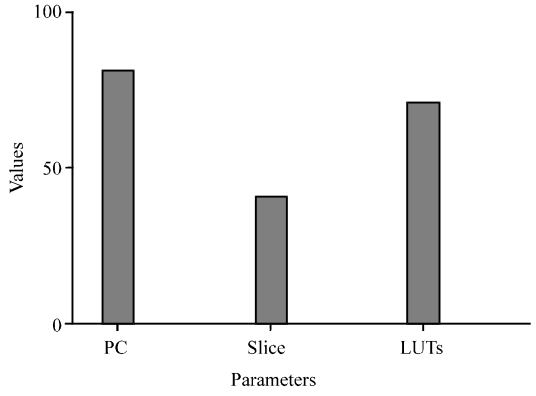


Fig. 4: Graphical illustration of hardware utilizations

and sub-keys are clearly illustrated in Fig. 5. Initially, the given packet data is permuted and this permuted data is divided in to two sub blocks of same size.

Initially the 10 bit key is permuted using permutation table P10 and this 10 bit key is divided in to 2 five bits of equal length, the first five bits are left shift by 1 and next corresponding five bits are left shift by 1. After this, the left shifted 10 bits are applied to permutation table P8 which produces first 8 bit key called as K1. The same process is applied for remaining portions in order to get another 8 bit key length called as K2.

**Pseudo code for encryption of data in receiving packet:**

Input: receiving packet with data
Output: encrypted packet with data
1. Upon extracting data from receiving packet
2. If the data is available then
3. Begin
4. Permute the data using P10 permutation table
5. P10 output is xor with K1
6. Switching operation is done
7. End
8. If the Switching operation is done then
9. It is xor with K2
10. End
11. Perform inverse permutation
12. End

**Dual layer ECC decoder:** During the packet decoding procedure, initially the first received packet is decoded using routing decoder which is available at the input port side of the router. If the received packet may have errors during transmission and these errors can be detectable and correctable with in the range limit then these packets are stored in the router internal buffer or else the received packet will be forwarded to its corresponding internal port of the router. In the mean time, if researchers receive the second packet from external router, these second packets can be combined or integrated with first received packet and the corresponding error patterns are decoded using such decoder.

The decoding process as shown in study is different from the hop to hop level product code. The entire message comes in each cycle in the hop to hop level application of the product code. In contrast, one flit arrives at the network interface is enough to execute the first decoding step. If there are uncorrectable errors in the first transmission packet, an array of column decoders is needed to obtain the final uncoded packet. To obtain the whole packet or the check bit packet, two NI buffers are necessary. Since these buffers can be shared with the existing NI data buffers, the cost is acceptable.

## RESULTS AND DISCUSSION

To evaluate the performance of the proposed scheme, researchers consider erroneous port which creates errors in other ports of the corresponding router itself. The proposed system architecture has minimal hardware utilities which provides low power consumption and reduces the routing latency in the router network on chip.

Researchers have used modelsim 5.5e as Simulation Software and Xilinx Project Navigator 10.1 as Synthesis Software to evaluate the proposed network on chip router. By analyzing the results from simulation, it shows that it produces the tolerable packet error rate and also the synthesis result shows that low hardware utilization such as 41 slices, 72 LUTs and 81 mW power consumption, respectively.

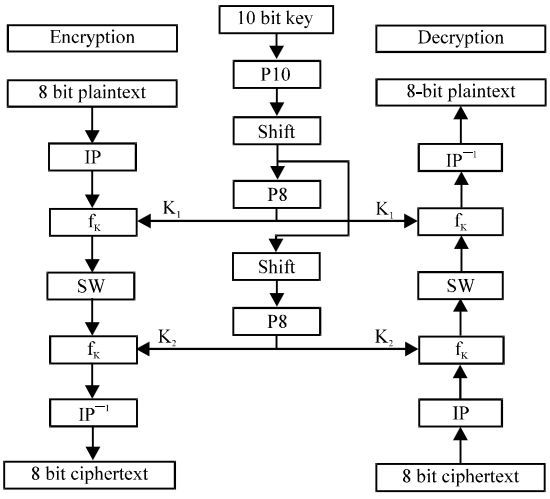


Fig. 5: The overall key generation process

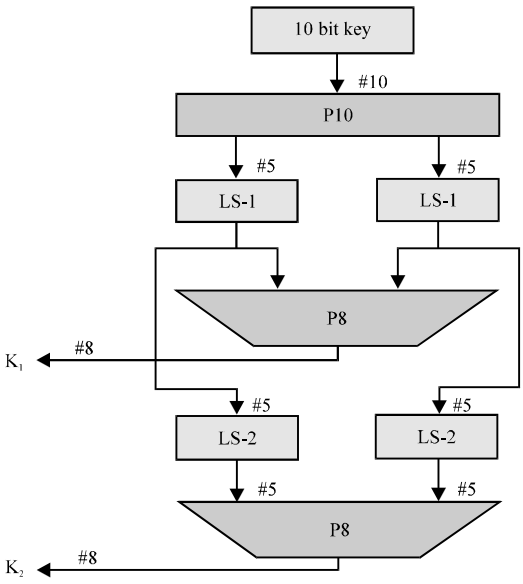


Fig. 6: Key generation flow

Table 1: Performance evaluation of power consumption and hardware utilizations

| Evaluation parameters | Banerjee *et al.* (2009) | Proposed methodology |
|---|---|---|
| Power consumption | 92 mW | 81 mW |
| Slices | 75 | 41 |
| LUTs | 102 | 72 |

The performance evaluation parameters considered here are power consumption, slices utilization and LUTs utilization in FPGA platform. These evaluation parameters are computed for our proposed methodology based on 3s500epq208-5 FPGA package. This performance evaluation between Banerjee *et al.* (2009) and the approach are given in Table 1 and the same is illustrated in Fig. 6.

## CONCLUSION

This research work proposes energy efficient and secured network on chip router using error control schemes. The proposed method significantly minimizes the packet loss when compared to other previous works. This can be achieved by implementing interleaving techniques with Cipher block based ECC Coding Method in NoC. The proposed system uses Modelsim Software for simulation purposes and Xilinx Project Navigator for synthesis purposes. The proposed system consumes 81 mW than the existing system and utilizes less hardware components.

## REFERENCES

Banerjee, A., P.T. Wolkotte, R.D. Mullins, S.W. Moore and G.J.M. Smit, 2009. An energy and performance exploration of network-on-chip architectures. IEEE Trans. Very Large Scale Integr. Syst., 17: 319-329.

Duan, C., V.H.C. Calle and S.P. Khatri, 2009. Efficient on-chip crosstalk avoidance CODEC design. IEEE Trans. Very Large Scale Integr. Syst., 17: 551-560.

Fu, B. and P. Ampadu, 2009. On hamming product codes with type-II hybrid ARQ for on-chip interconnects. IEEE Trans. Circuits Syst. I: Regul. Pap., 56: 2042-2054.

Fu, B. and P. Ampadu, 2010. Exploiting parity computation latency for on-chip crosstalk reduction. IEEE Trans. Circuits Syst. II: Express Briefs, 57: 399-403.

Ganguly, A., P.P. Pande, B. Belzer and C. Grecu, 2008. Design of low power and reliable networks on chip through joint crosstalk avoidance and multiple error correction coding. J. Electron. Test, 24: 67-81.

Lee, K., S.J. Lee and H.J. Yoo, 2006. Low-power network-on-chip for high-performance SoC design. IEEE Trans. Very Large Scale Integr. Syst., 14: 148-160.

Lehtonen, T., P. Lijieberg and J. Plosila, 2007. Analysis of forward error correction methods for nanoscale networks-on-chip. Proceedings of the 2nd International Conference on Nano-Networks, September 24-26, 2007, Catania, Italy, pp: 1-5.

Murali, S., T. Theocharides, N. Vijaykrishnan, M.J. Irwin, L. Benini and G. De Micheli, 2005. Analysis of error recovery schemes for networks on chips. IEEE Des. Test Comput., 22: 434-442.

Xu, J., W. Wolf and W. Zhang, 2009. Double-data-rate, wave-pipelined interconnect for asynchronous NoCs. IEEE Micro, 29: 20-30.

Yu, Q. and P. Ampadu, 2010. A flexible parallel simulator for networks-on-chip with error control. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., 29: 103-116.