An Efficient Error Correction Code for A Smart Reliable Network-on-Chip

KRITHIKA ANAND[1] MUKTHI S L[2]
[1]P.G Scholar [2]Assistant Professor
[1]Department of Electronics and Communication Engineering
[1]Bangalore Institute of Technology, Bangalore, India

Abstract— Networks on Chip plays a vital role in development of VLSI design and it is one of the solutions for faster on chip communication. In this paper a new network-on-chip is proposed that handles accurate localizations of the faulty parts of the NoC. The proposed NoC handles new error detection mechanisms suitable for dynamic NoCs, and the errors are detected and corrected efficiently in order to have a fault free data transmission. The presented mechanisms are able to distinguish permanent and transient errors and localize accurately the position of the faulty blocks such as data bus, input port, output port in the NoC routers, while preserving the throughput, the network load, and the data packet latency. The proposed method uses Decimal Matrix Code concept which utilizes decimal algorithm to obtain the maximum error detection capability. The proposed DMC is compared to the existing Hamming code which corrects only one bit error, whereas DMC can correct up to 16 bits error. As a result this improves the error correction capability and enhances data reliability.

Key words: Network-On-Chip (NoC), Decimal Matrix Code (DCM), Dynamic NoCs

I. INTRODUCTION

The intricacy of each component in a system increases rapidly as the volume and density of VLSI design increases. In order to meet the requirements of real-time applications the trend of embedded systems has been moving toward multiprocessor systems-on-chip (MPSoCs). The traditional bus-based communication methods are not able to keep up with the increasing requirements of future SoCs in terms of performance, timing closure, power, scalability etc. The complexity of these SoCs is increasing and the communication medium is becoming a major issue of the MPSoC [6]. To satisfy the design productivity and signal integrity challenges of next-generation system designs, a structured and scalable interconnection architecture, Network on-Chip, has been proposed recently to reduce the complex on-chip communication problem. The integration of a network-on-chip (NoC) into the SoC gives an effective means to interconnect several processor elements (PEs) or intellectual properties (IP) (processors, memory controllers, etc.). Although NoCs can adopt concepts and methods from the well-established platforms of computer networking, it is impractical to blindly reuse features of existing computer networks and symmetric multiprocessors. In particular, NoC switches should be energy-efficient, small and fast. For early NoC research neglecting these views along with proper comparison was typical but nowadays they are considered in more detail. The NoC medium features a high level of modularity, flexibility, and throughput. An NoC comprises routers and interconnections allowing communication between the PEs and/or IPs. The NoC relies on data packet exchange. The path for a data packet between a source and a destination through the routers is defined by the routing algorithm. Therefore, the path that a data packet is allowed to take in the network depends mainly on the adaptiveness permitted by the routing algorithm, which is applied locally in each router being crossed and to each data packet.

The partial dynamic reconfiguration of FPGAs with varying position and the number of implemented PEs and IPs, higher adaptiveness is allowed in MPSoCs during runtime. Hence dynamically reconfigurable 2-D mesh NoCs (DyNoC, CuNoC, QNoC, ConoChi, etc.) are suitable for field programmable gate array (FPGA)-based systems. To achieve a reconfigurable NoC, an efficient dynamic routing algorithm is required for the data packets. The goal is to preserve flexibility and reliability while providing high NoC performance in terms of throughput.

Fig. 1(a) depicts the communication between different processor elements (PE's) of a dynamic reliable NoC. Fig 1(b) and Fig 1(c) depicts the placement of PE's dynamically and the occurrence of nodes with faults. In both the cases the bypass through the routers is determined by the dynamic routing algorithm. Fig 1(c) shows regions with faulty nodes which make the communication impossible with routing algorithms that are not adaptive. This shows the need for algorithms that are fault tolerant and highly adaptive that can be used dynamic NoCs during runtime.

MPSoCs are becoming more sensitive to phenomena that generate permanent, transient, or intermittent faults [10] because of the increasing complexity and the reliability evolution of SoCs. These faults may generate data packet errors, or may affect router behavior leading to data packet losses or permanent routing errors. Indeed, a fault in a routing logic will often lead to packet routing errors and might even crash the router. To detect these errors, specific error detection blocks are required in the network to locate the faulty sources. Indeed, the precise location of permanent faulty parts of the NoC must be determined, in order for them to be bypassed effectively by the adaptive routing algorithm. For handling message routing errors in dynamic networks, a new faulty switch detection mechanism is required for adaptive or fault-tolerant routing algorithms. The error correction codes (ECCs) are implemented inside the NOC components in order to protect data packets against errors.

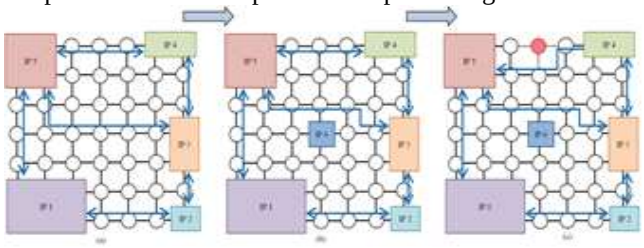

Fig. 1: Illustration Of A Dynamic Reliable Noc. (A) Normal Operation. (B)Dynamic Implementation Of An IP. (C) Online Detection Of A Faulty Router.

In this paper, a novel decimal matrix code (DMC) [2] based on divide-symbol is proposed to provide enhanced memory reliability. The proposed DMC utilizes decimal algorithm which involves decimal integer addition and decimal integer subtraction to detect and correct errors. The advantage of using decimal algorithm is that the error detection capability is maximized so that the reliability of memory is enhanced.

This paper is divided into the following sections. The existing system is presented in Section II. Section III illustrates the proposed system and the decimal matrix code concept. In Section IV the simulation results are illustrated and Section V gives the conclusion of the paper.

II. EXISTING RKT-SWITCH CONCEPT

The architecture of RKT-NoC switch [1] is as shown in the Fig 2. The RKT-NoC is a packet- switched network based on intelligent independent reliable routers called RKT-switches. The switch has four directions i.e., North, South, East, and West which is needed for a 2-D mesh NoC. The IPs and PEs can be linked directly to any side of the router and thus there is no need of specific connection. This proposed mechanism can be applicable to NoCs using five port routers with a local port dedicated to an IP but when this local port has permanent error the IP connected to this port is permanently lost or it needs to be moved in the chip dynamically. Hence this is the major drawback of using five port routers. On the other side, that is by using four port RKT-NoC the IP can be replaced by several routers by having several input ports. Therefore there are strongly connected in the network. The switch has two unidirectional data buses (input and output ports). Each input port has first-input, first-output (FIFO) and a routing logic block. The switching strategy used is the store-and-forward switching technique which is suitable for dynamically reconfigurable NoC. This is because at any instant with the store- and-forward technique, each data packet is stored only in a single router. Hence, when a router needs to be reconfigured, the router is only required to empty its buffers.

The automatic repeat request is the type of error control method used here is which uses error detecting codes. The Ack/Nack solution is used in the existing architecture which handles fault-tolerant transmissions effectively. The packet is saved locally for packet retransmission until an Ack or Nack is received. Once a flit with an error fails to be corrected by ECC is found by the neighbouring router, a Nack is sent back and the whole packet is retransmitted., an Ack is sent by receiver signifying that full packet is being received correctly as a data frame as a result latency is reduced. Otherwise

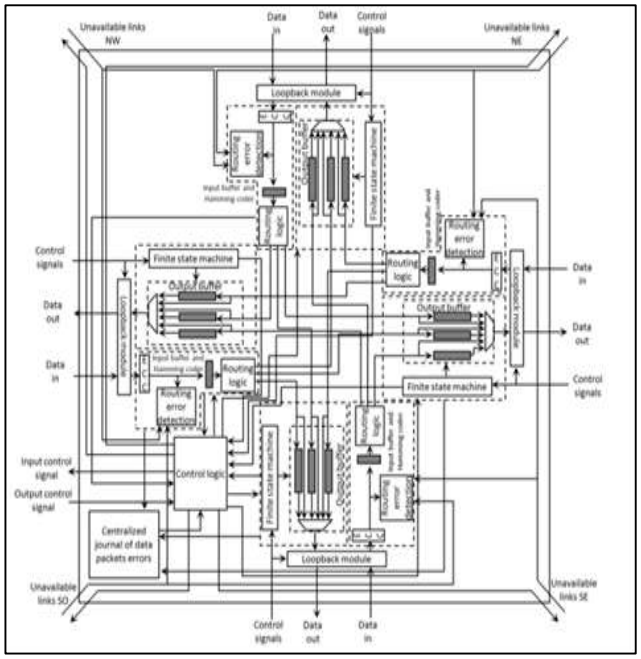

Fig. 2: Architecture of Reliable Router RKT Switch

The Hamming ECC is considered for our RKT-switch, in order to provide a convenient tradeoff between area overhead and error correction capacity. This choice permits the correction of single event upset (SEU) errors (one bit flip in a flit) and the detection of multiple event upset (MEU) errors (two bit flips in a flit). Moreover, the Hamming code is more suitable for NoCs based on Ack/Nack flow control than the parity bit check. Indeed, on a single bit-flip error occurrence, error correction is possible with the Hamming ECC, whereas the single parity check would require packet retransmission and hence an increased transmission latency.

III. PROPOSED DMC METHOD

In the proposed system we use Decimal Matrix Code to maximize the error detection and correction capability. Decimal Matrix Codes (DMC) are mainly used to avoid soft errors due to multiple cell upsets in memory unit. DMC make use of decimal algorithm that is decimal integer addition and decimal integer subtraction to detect errors. In this the information bits are provided to the DMC encoder during the encoding process. As a result, the horizontal redundant bits, vertical redundant bits and the information bits are obtained. Once encoding process gets completed, this codeword is stored to the memory. If multiple cell upset occurs in memory, this can be corrected during the decoding process. Thus the error detection and correction capability is maximized due to the usage of decimal matrix code.

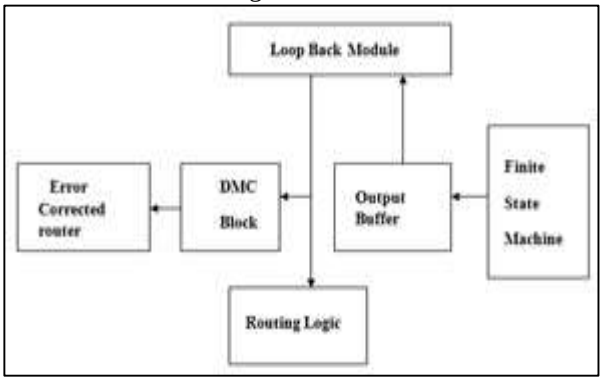

Fig. 3: Proposed System

The existing system used hamming code concept for error detection and correction which corrects one bit error and detects maximum of two bit error, whereas the proposed Decimal Matrix Code detects and corrects up to 16 bits of data. This increases the data reliability and enhances the fault free data transmission between different directions of the NoC router. The proposed system is as shown in the Fig 3, where the DMC concept is implemented as shown.

In the proposed DMC [2], we first divide-symbol and arrange them in matrix form i.e., the N-bit word is divided into k symbols of m bits (N = k × m), and these symbols are arranged in a k1 ×k2 2-D matrix. By performing decimal integer addition of selected symbols per row the horizontal redundant bits H are produced and thee vertical redundant bits V are obtained by binary operation among the bits per column. These operations are implemented in logical instead of in physical. Hence, the proposed DMC does not require changing the physical structure of the memory.The Fig 4 illustrates an example to understand the proposed DMC scheme. Here we take a 32 bit word i.e. D0 to D31 which are the information bits. We then divide the informaton bits into eight symbols of 4-bit. The horizontal check bits are H0–H19 and V0 through V15 are vertical check bits. The k and m should be carefully adjusted to maximize the correction capability and minimize the number of redundant bits.

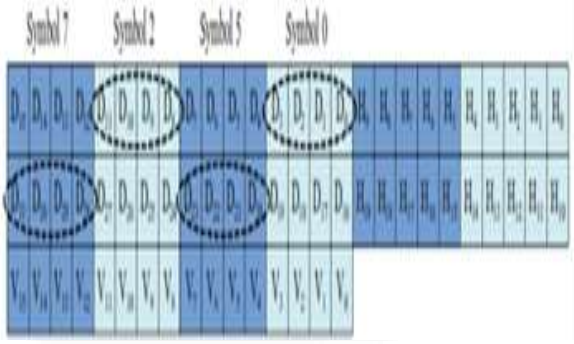

Fig. 4: 32-bits DMC logical organization (k = 2 × 4and m = 4).

The horizontal redundant bits H can be obtained by decimal integer addition as follows:

$$H_4H_3H_2H_1H_0 = D_3D_2D_1D_0 + D_{11}D_{10}D_9D_8 \quad (1)$$

$$H_9H_8H_7H_6H_5 = D_7D_6D_5D_4 + D_{15}D_{14}D_{13}D_{12} \quad (2)$$

and similarly the rest of the horizontal redundant bits are obtained where "+" represents decimal integer addition.

For the vertical redundant bits V, we have

$$V0 = D0 \oplus D16 \quad (3)$$

$$V1 = D1 \oplus D17 \quad (4)$$

and similarly for the rest vertical redundant bits.

The decoding process is required to obtain a word being corrected. For example, first, the received redundant bits H4H3H2H1H0′ and V0′−V3′ are generated by the received information bits D′. Second, the horizontal syndrome bits ΔH4H3H2H1H0 and the vertical syndrome bits S3 −S0 can be calculated as follows:

$$\Delta H_4H_3H_2H_1H_0 = H_4H_3H_2H_1H_0' - H_4H_3H_2H_1H_0 \quad (5)$$

$$S_0 = V_0' \oplus V_0 \quad (6)$$

and similarly for the rest vertical syndrome bits, where "−" represents decimal integer subtraction.

When ΔH4H3H2H1H0 and S3 −S0 are equal to zero, the stored codeword has original information bits in symbol 0 where no errors occur. When ΔH4H3H2H1H0 and S3 −S0 are nonzero, the induced errors are detected and located in symbol 0, and then these errors can be corrected by

$$D0correct = D0 \oplus S0. \quad (7)$$

Finally, these errors can be corrected by inverting the values of error bits. The Fig 5 shows the flow chart of the proposed DMC method.

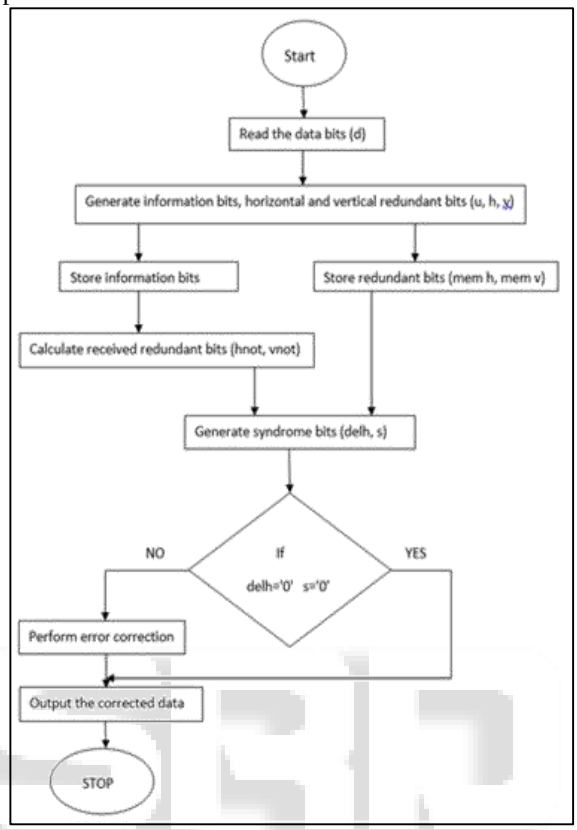

Fig. 5: Algorithm of DMC

IV. SIMULATION RESULTS

Simulation was done in ModelSim and synthesis report was summarized in Xilinx ISE 13.2. The Fig 6 below shows the simulation of the DMC concept used in the ECC block of the NoC router. The simulation result of the NoC router as top module is shown in Fig 7.

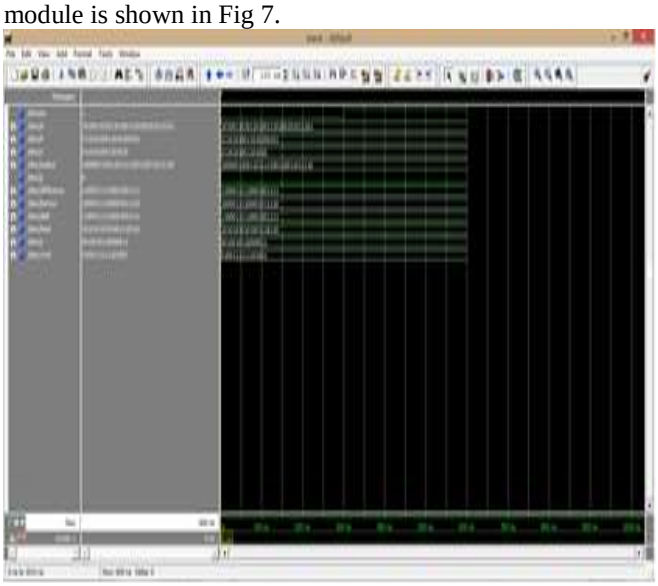

Fig. 6: Snapshot of DMC used in ECC

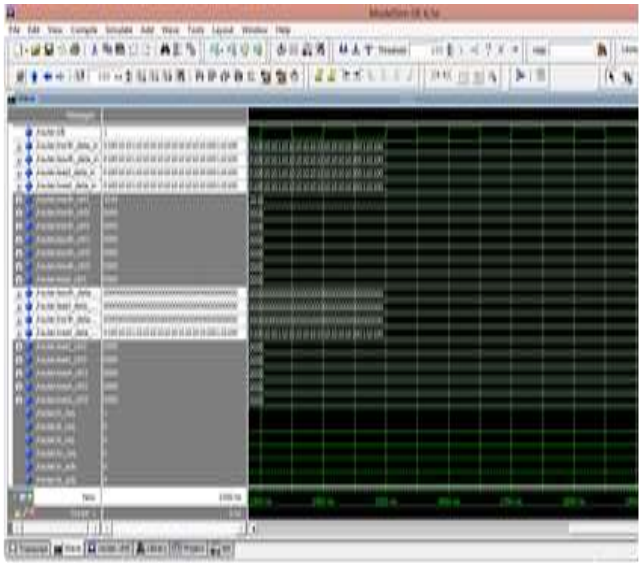

Fig. 7: Snapshot of NoC router operation

V. CONCLUSION

In this paper, we proposed new error detection mechanisms for dynamic NoCs. The proposed routing error detection mechanisms allows the accurate localization of permanent faulty routing blocks in the network. The proposed DMC method , corrects up to 16 bit error and can detect more than 16 bit error. It increases the error correction rate up to 10 times more than the existing method. Thus the error detection and correction capability is maximized due to the usage of decimal matrix code.

REFERENCES

[1] Cédric Killian, Camel Tanougast, Fabrice Monteiro, and Abbas Dandache " A Smart Reliable Network-on-Chip", IEEE TRANSACTIONS ON VERYLARGE SCALE INTEGRATION (VLSI) SYSTEMS 2013.
[2] Jing Guo, Liyi Xiao, Zhigang Mao, and Qiang Zhao "Enhanced Memory Reliability Against Multiple Cell Upsets Using Decimal Matrix Code", IEEE Transaction on Very Large Scale integration (VLSI) Systems 2013.
[3] S. Liu, P. Reviriego, and J. A. Maestro, "Efficient majority logic fault detection with difference-set codes for memory applications," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 20, no. 1, pp. 148–156, Jan. 2012.
[4] M. Hosseinabady, M. Kakoee, J. Mathew, and D. Pradhan, "Low latency and energy efficient scalable architecture for massive NoCs using generalized de Bruijn graph," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 19, no. 8, pp. 1469–1480, Aug. 2011.
[5] A. Ejlali, B. Al-Hashimi, P. Rosinger, S. Miremadi, and L. Benini, "Performability/energy tradeoff in error-control schemes for on-chip networks," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 18, no. 1, pp. 1–14, Jan. 2010.
[6] K. Sekar, K. Lahiri, A. Raghunathan, and S. Dey, "Dynamically Configurable bus topologies for high-performance on-chip communication," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 16, no. 10, pp. 1413–1426, Oct. 2008.
[7] C. Grecu, A. Ivanov, R. Saleh, E. Sogomonyan, and P. Pande, "On-line fault detection and location for NoC interconnects," in Proc. 12th IEEE Int. On-Line Test. Symp., Jul. 2006, pp. 145–150.
[8] J. Han, "Toward hardware-redundant, fault-tolerant logic for nanoelectronics," IEEE Design Test Comput., vol. 22, no. 4, pp. 328–339, Jul.– Aug. 2005.
[9] M. Majer, C. Bobda, A. Ahmadinia, and J. Teich, "Packet routing in dynamically changing networks on chip," in Proc. 19th IEEE Int. Parallel Distrib. Process. Symp., Apr. 2005, p. 154b.
[10] J. Wu, "A fault-tolerant and deadlock-free routing protocol in 2d meshes based on odd-even turn model," IEEE Trans. Comput., vol. 52, no. 9, pp. 1154–1169, Sep. 2003.

