Indonesian Journal of Electrical Engineering and Computer Science
Vol. 29, No. 3, March 2023, pp. 1336~1344
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v29.i3.pp1336-1344

Proficient matrix codes for error detection and correctionin 8-port network on chip routers

Neelima Koppala, Nagarajan Ashok Kumar, Satyam, Neeruganti Vikram Teja
Department of Electronics and Communication Engineering, Sree Vidyanikethan Engineering College, Tirupati, India

Article Info

Article history:

Received May 21, 2022
Revised Nov 16, 2022
Accepted Nov 24, 2022

Keywords:

Dynamic networks
Matrix codes
Network on chip
Router
Throughput

ABSTRACT

This paper verifies the applicability of the proposed code to dynamic Network on Chips that have variable faulty blocks with runtime suggesting an online error detection mechanism with adaptive routing algorithm that bypasses faulty components dynamically and the router architecture uses additional diagonal state indications for the reliable network on chip (NoC) operation. In NoC, the permanently faulty routers are disconnected to enable high runtime throughput as data packets are not lost due to self-loopback mechanism. The proposed proficient matrix codes use the capabilities of decimal matrix code technique with minimum check bits for maximum error correction capability. The proposed code is compared with existing codes such as decimal matrix codes, modified decimal matrix codes and parity matrix codes. The codes are developed in verilog hardware description language and simulated in the Xilinx ISE 14.5 tool. This proficient matrix code proves to be efficient for multiple adjacent error detection and correction with trade off in delay. Also 65% code rate is achieved with 22.73% less redundant bits that occupy less area by atleast 11.78%. The codes when used for increased data sizes like 8, 16, 32, and 64 bits, the power delay product decreased by atleast 1.74%.

Corresponding Author:

Neelima Koppala
Department of Electronics and Communication Engineering, Sree Vidyanikethan Engineering College
Tirupati, Andhra Pradesh, India
Email: neelima.k@vidyanikethan.edu

1. INTRODUCTION

As a result of rapid advancements in very large-scale integration (VLSI), billions of transistors are expected to be fabricated on a single chip. For establishing communication among these elements, traditional wiring becomes less reliable. The best available solution for these complex wiring issues in System on Chip (SoC) is network on chip (NoC) [1]-[3]. NoC has established itself as the most dependable and legitimate method of connecting multiple cores in a SoC. It also increases the throughput without using further channels for interconnection. A routing-algorithm should also be developed to specify which path a data packet should take from source through routers. The deadlock and live lock both must be free [4]. The configuration of routers and processor connections on a device is referred to as networktopology [5]-[7]. The most common topology is 2-D mesh topology which looks like a tile arrangement. The router is typically made up of first in first out (FIFO) buffers, control logic, and an arbiter.

The system architecture is a NoC router [8], [9] capable of overcoming both dead and live locks. Figure 1 depicts the router's block diagram. It is appropriate for a two dimensional mesh NoC with four directions (east, west, north and south) [10]-[12]. Router can connect provider edge router (PES) and IPS directly to any side. Therefore, a PE or an intellectual properties (IP) does not require a particular connection port [13]-[16]. It includes four loopback modules, routing error detection, routing logic, I/O buffers, I/O ports,

ECC, and control signals. It uses the store and forward method of operation [17]. Among these, routers are the elements that induce errors into the data stored in buffers. The memories capacity in recent years has increased rapidly, and radiation effects in a variety of spaces, including aeronautical, nuclear and chemical equipment, whose memory operates at very high temperatures or in a radiation environment, face soft error, then the error detection and correction (EDAC) codes are necessary. In the previous methods, hamming code was used to correct errors in the transmitted bits. The code architecture is liable to errors and doesn't represent a breakdown of errors even though the path and therefore the data are properly managed. The term hamming code is employed to detect and rectify individual bit errors [18], [19]. Further, there esixt many error detection and correction codes in literature that are capable of correcting single bit to single byte errors.

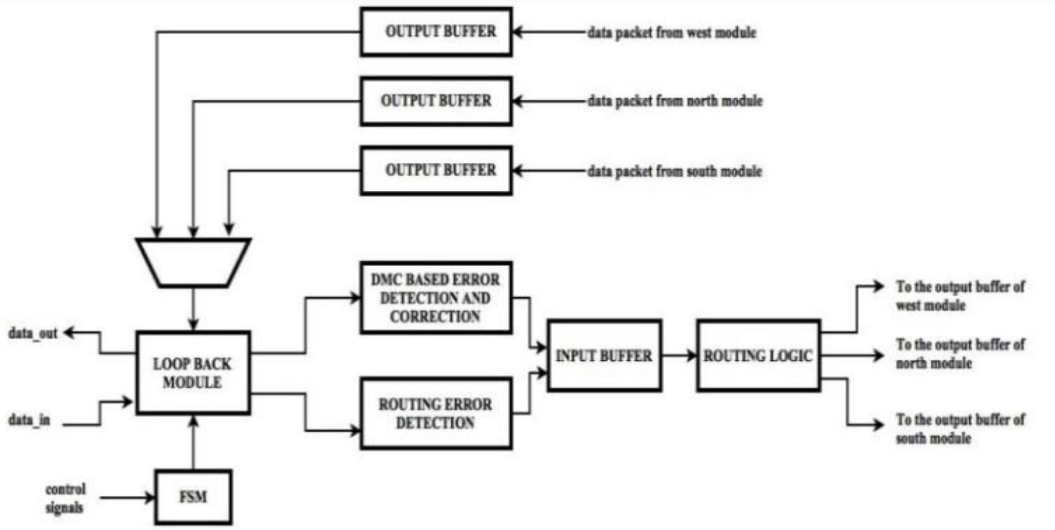

Figure 1. Block diagram of NoC router

In the existing system, any number of errors can be detected in a single byte via the code for single error correction-double error detection-single byte detection (SEC-DED-SBD). These codes are especially useful to protect byte-organized memories than traditional single error correction-double error detection (SEC-DED) codes [20]-[22]. For several other codes capable of resolving multiple errors, additional error correction codes, single byte correction-double byte detection (SBC-DBD) and triple error detection code are included. In addition, the reconfiguration NoC structure can include bypass areas where the switches use a rotation algorithm to take various routing choices. It has a low overhead and can be detected multiple times, but it cannot be fixed for multiple errors. Also, the Turn model algorithm is inappropriate to detect multiple upsets or errors.

The ECCs based on decimal algorithms are implemented within NoC components such as router switch to protect data packets from errors. Four-way architecture characterises the switch as north, south, east and west, while the eight-way architecture characterises the switch for as north, south, east, west, north-east, north-west, south-east and south-west as shown in Figure 2. Processing elements (PE) and intellectual properties (IP) are directly connected to the router. Thus there is no specific PE or IP connection port. There are two unidirectional data buses in each port direction (input and output ports). A FIFO (primaryinput, first out buffer) and a logical routing block are connected to each input port. The storage and forward switching techniques are used during the switching operation. All data packets are saved on a single router at any given time using the storage and transmission technique. If a router is reconfigured, the buffers must only be emptied. As the switch uses non-bouncing routers, the other three neighbours around a router remain unavailable. If a router with three nodes, then it cannot be routed with a data packet.

Error correction code memory (ECC) block checks its contents to ensure that the data is correct if a router receives a neighbour's data packet. Data errors based on decimal ECC effectiveness are detected and fixed by this process. The NOC is a mesh router structure that detects routing error using the XY algorithm for adaptive routing. This solution helps to correct individual errors (one bit in one flit) in event disorder (SEU) and to identify several errors in the event disruption (MEU). An online fault detection mechanism includes the proposed reliable switch. The XY routing algorithm can be used in conjunction with adaptive algorithms. The reliable router depends upon real-time diagonal information, additional flow information on the header, and

routing error blocks to detect routing errors for each port. The underlying principle uses previously crossed switch to confirm every route receiving a data packet. In combination with decimal ECC, this error routing detection is conducted.

Among the existing codes like decimal marix code, modified decimal matrix code and parity matrix codes suit this application due to their organization of matrix like structure which enables to correct multiple data at a time. But the problem of using these codes is they have less code rate and use more number of redundant bits for error detection and correction. In this paper, we propose proficient matrix code that is capable of correcting maximum number of adjacent errors with less number of redundant bits and maximum possible code rate.

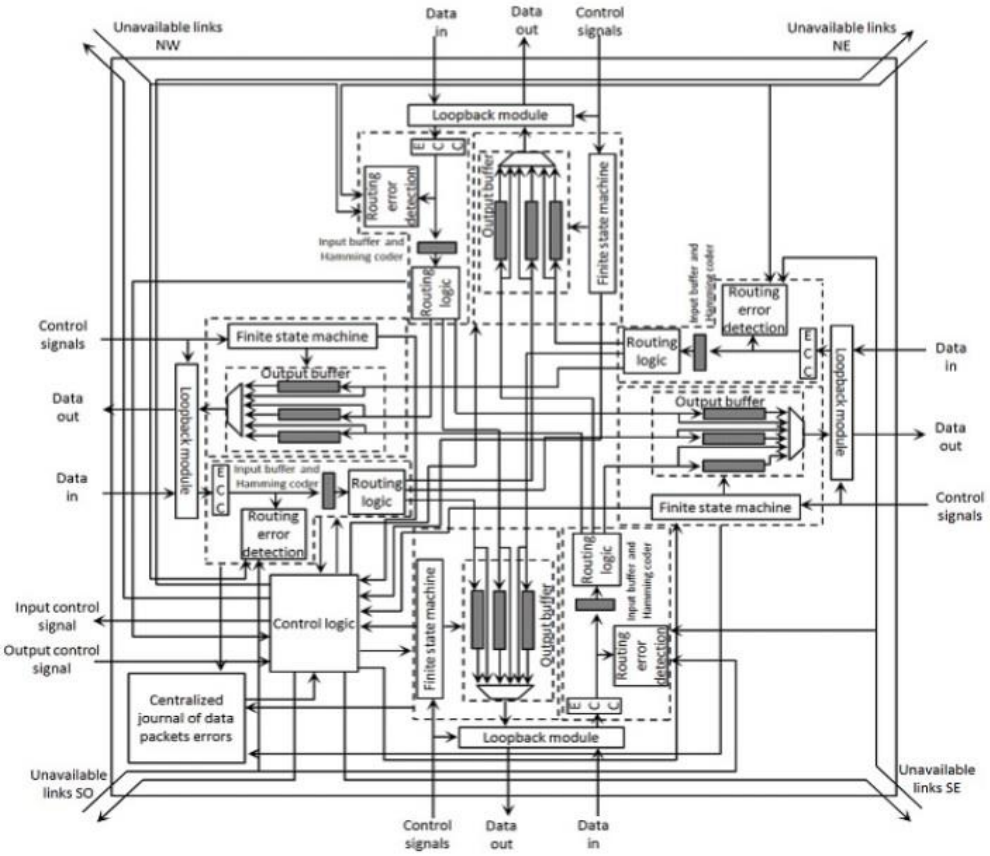

Figure 2. Architecture of the router switch

2. METHOD

The codes considered in this paper for evaluating the proposed proficient matrix code are decimal matrix code, modified decimal matrix code and parity matrix code. The data matrix codes (DMC) code is based on a proportional divider symbol and uses the decimal integer addition and subtraction for detecting errors [23], [24] as shown in Figure 3. This encoding approach in each row and column assumes a symbol treated as decimal integer to obtain a corrected word.

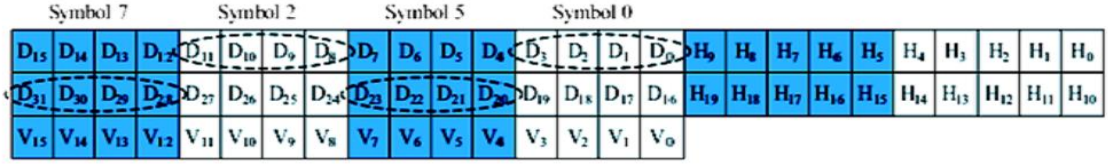

Figure 3. DMC local organization (k=2 x 4, m=4)

Information bits are included in the D0 to D31 numbered cells. The 32-bit word was divided into eight four-bit symbols i.e., K1=2 and K2=4 are chosen at the same time. The horizontal parity bits are H0-H19, while vertical check bits are V0-V15. The maximum correcting capabilities (i.e., maximum correction size of the corrected MCU) and the number of check bits differ when different values are used for k and m. To maximise the correction capability, k and m need to be chosen carefully, while using less number of overhead bits.The full adders are used for horizontal check bits, and vertical check bits are obtained by modulo-2 addition. Ideas for dividing a symbol and arranging a matrix are implemented in the system. To generate the defective information bits, the failure is injected into the decoding data bit. If an error in a data bit is discovered, a calculation of the syndrome will correct the corresponding bit. Finally it generates the corrected output data as shown in Figure 4.

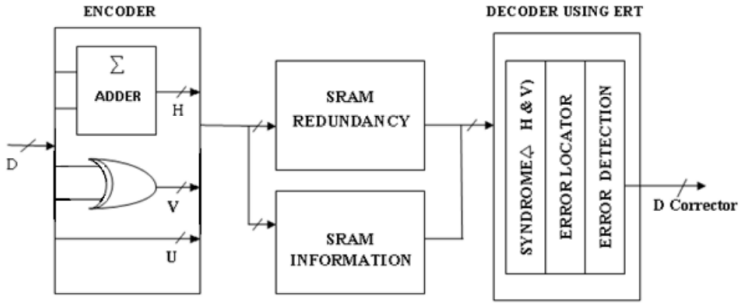

Figure 4. DMC code

During encoding, the data bits D are used to obtain H, V, and U (redundant data bits) are obtained and the codeword is saved in memory. During decoding, the MCUs found in the memory are corrected using syndrome bits. Thus, DMC has a higher fault tolerant capacity with lower overhead performance and uses less overhead by encoder reuse technique (ERT).

The modified DMC (MDMC) is similar to the DMC in that it calculates horizontal redundant bits using higher order adders and subtractors. It uses a Hamming encoder and XOR gates for encoding. For an N-bit data, k-symbols of m-bits size are subdivided, i.e., N = k x m. The k-symbols are arranged in a k1 x k2 2-D matrix. For each symbol treated as decimal integer in the first row, the horizontal check bits are calculated using Hamming codes. The vertical bits for each column are calculated using binary Ex-OR operation. The electrical resistivity tomography (ERT) technique is used for decoding [25].

Consider 32-bit data i.e., D0-D31 cells contain bits of information then N=32 has k=8 symbols each with m=4 bits. P0-P11 as horizontal check bits and V0-V15 as vertical check bits. For k = 2 x 4 and m = 4, then, up to 16-bit errors and 28 redundant bits can be corrected [26] as shown in Figure 5.

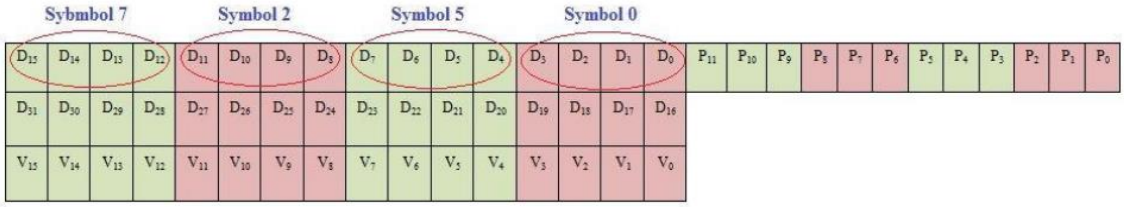

Figure 5. 32-bit Modified DMC logical representation

When compared to the MDMC, the parity matrix code (PMC) has a higher correction capability with the same detection capability [27] as shown in Figure 6. The inadequacy of H ensures that the many-sided translation performs as well as that the code length is linear and increases in proportion with codeword length. The H matrix is a sparse matrix consisting of identity matrix and parity matrix. In comparison with other coding methods, the most significant advance here is the method by which the code is decoded to minimise overhead cost.In encoder, this method uses a matrix multiplier and addition along with a generator matrix to detect errors.

$$G = \begin{bmatrix} 1 & 0\ 1 & 0 \\ 0 & 1\ 1 & 1 \end{bmatrix}$$

The generator matrix is multiplied by the input divided into two bits. The matrix is multiplied by four bits, with the first two bits represent input and the last two as overhead bits. Both of these data sets are stored separately in static random-access memory (SRAM). The matrices of identity and equality are shown in the first and second columns respectively. The proportion of a single column or row, regardless of its matrix order must always be determined to divide the matrix in two equal parts, one as identity matrix and the other as parity matrix [28].

The PMC decoder uses H-matrix, i.e., a transpose of the generator matrix. The data is increased within the matrix once more, and the matrix is added, allowing us to detect and correct errors within the stored bits. The correction is made by multiplying the redundant bit by adding the multiplied data, which leads to the correction of the original data or the error.

$$\mathrm{H} = \begin{bmatrix} 1 & 1\ 1 & 0 \\ 0 & 1\ 0 & 1 \end{bmatrix}$$

Data are used to check the number of corrected data compared to the input data. Throughout the flow, as the divide is two, the decoder detects four bits in both cells and, when the number of split bits is two, it is able to correct one, resulting in a defect correction ratio of one-half as shown in Figure 7. This means that in any cases of detection of memory cells, we will obtain a most 32-bit correction rate for 64-bit input information. PMC has an advantage over improved DMC because the detection remains constant till 64-bit data is used [29].

The proposed proficient matrix code is similar to DMC [18] with a difference that its horizontal parity bits can predict only the position of the error, while the remaining bits remain unchanged and the way they are designed is as shown in Figure 8. One partial encoder is reused for syndrome bit analysis, with only the H and V bits shown in Figure 9. The data in the 64-bit range is logically identical [19]. Again, the horizontal non-zero syndrome bits indicate where the error should be located, and the vertical non-zero syndrome bits are used to resolve incorrect results [20].

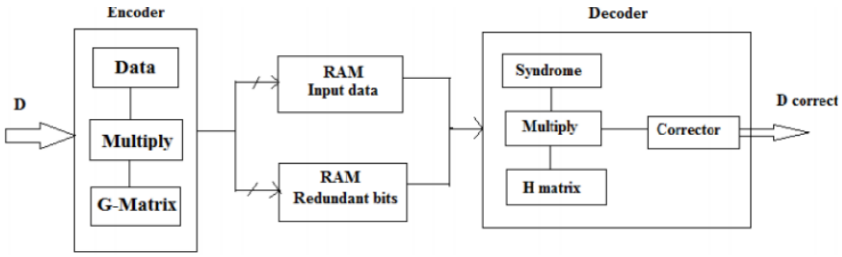

Figure 6. Block diagram of PMC

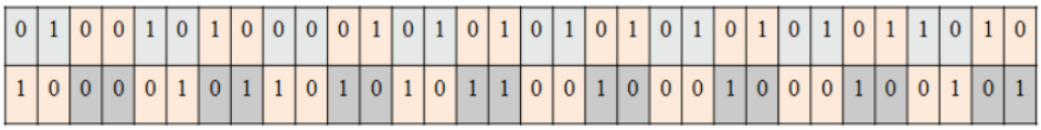

Figure 7. Logical PMC organisation 64-bit, divided into two symbols

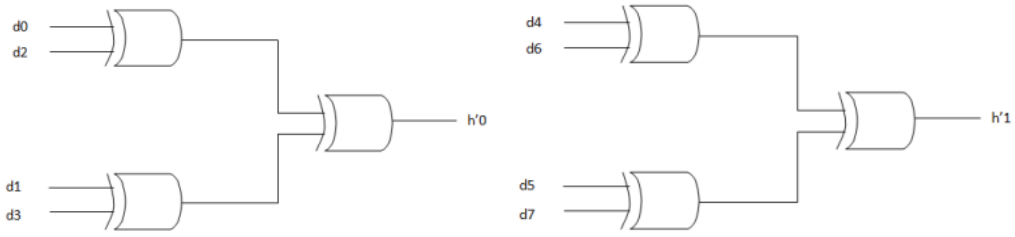

Figure 8. Horizontal redundant bits

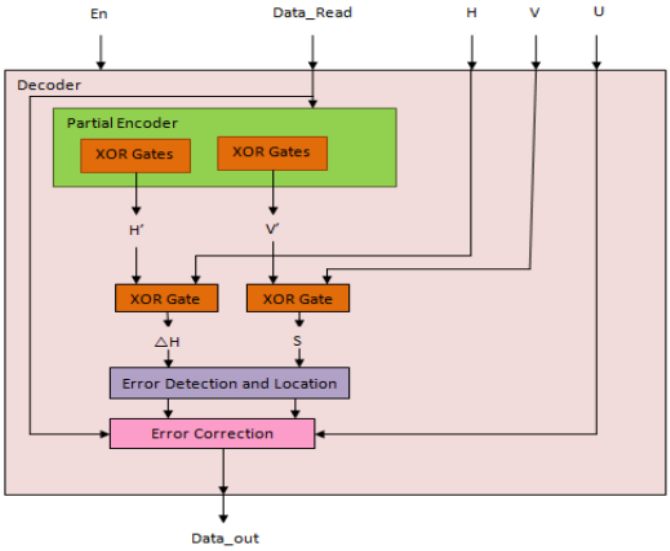

Figure 9. Proficient matrix code decoder

3. RESULTS AND DISCUSSION

The codes are written in Verilog HDL and tested with Zynq (XC7Z100- 1FFG1156) FPGA in Xilinx ISE 14.5 Tool. The power is evaluated using Xpower Analyzer for a toggle rate of 12.5%. Table 1 and Table 2 show that when compared to DMC, MDMC, and PMC, the PrMC code uses less code word bits by at least 22.22%, 18.75%, 13.79%, and 9.26%, and less redundant bits by at least 4%, 37.5%, 22.76%, and 22.73%, and a high code rate by at least 22.23%, 18.75%, 13.7%, and 9.26%, respectively. The simulation result of 8-port NoC with PrMC is shown in Figure 10, where the maximum correction capability of adjacent erroneous bits depends on the length of the vertical syndrome. As XOR gates alone cannot ideally help in error correction, XNOR gates are also necessary to correct even number of erroneous bits in data.

Table 1. Comparison of existing and proposed matrix codes for 8 and 16 bit data sizes

Parameters	DMC	MDMC	PMC	PrMC	DMC	MDMC	PMC	PrMC
#Data Bits, k	8	8	8	8	16	16	16	16
#Parity Bits, r	12	10	10	6	20	18	16	10
#Code Word, n=r+k	20	18	18	14	36	34	32	26
Bit Overhead, r/k in %	150	125	125	75	125	112.5	100	62.5
Code Rate, k/n in %	40	44.44	44.44	57.14	44.44	47.06	50	61.54
Code Efficiency, r/n in %	60	55.55	55.55	42.86	55.55	52.94	50	38.46

Table 2. Comparison of existing and proposed matrix codes for 32 and 64 bit data sizes

Parameters	DMC	MDMC	PMC	PrMC	DMC	MDMC	PMC	PrMC
#Data Bits, k	32	32	32	32	64	64	64	64
#Parity Bits, r	36	34	26	18	68	66	44	34
#Code Word, n=r+k	68	66	58	50	132	130	108	98
Bit Overhead, r/k in %	112.5	106.25	81.25	56.25	106.25	103.125	68.75	53.125
Code Rate, k/n in %	47.06	48.48	55.17	64	48.48	49.23	59.26	65.31
Code Efficiency, r/n in %	52.94	51.51	44.83	36	51.51	50.77	40.74	34.69

The codes for various data bits, such as 8, 16, 32, and 64-bit data are compared with respect to area and power delay product when they are used in 8-port router as shown in Figures 11 and 12. From Figure 11, the PrMC decreases area occupied by 35.27%, 22.89% and 11.78% when compared with DMC, MDMC and PMC respectively. Similarly from Figure 12, the proficient matrix codes of 64-bit data size use increased power delay product of 19.46% and 12.11% when compared with DMC and MDMC codes due to increase in number of Ex-OR gates used for implementation, and 1.74% decrease when compared with PMC Codes.

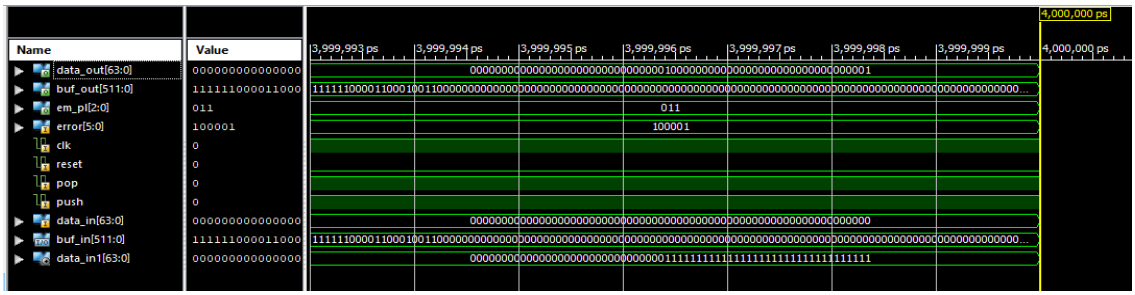

Figure 10. Simulated result of PrMC for 8 port NoC router

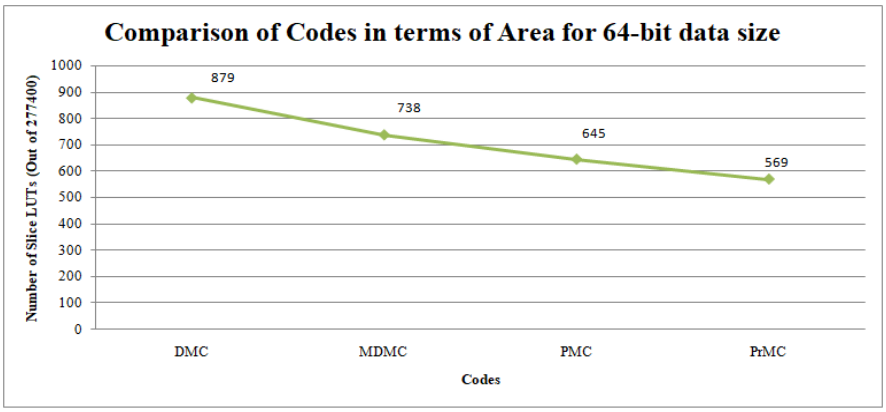

Figure 11. Comparison of 8-port router with various codes in terms of area

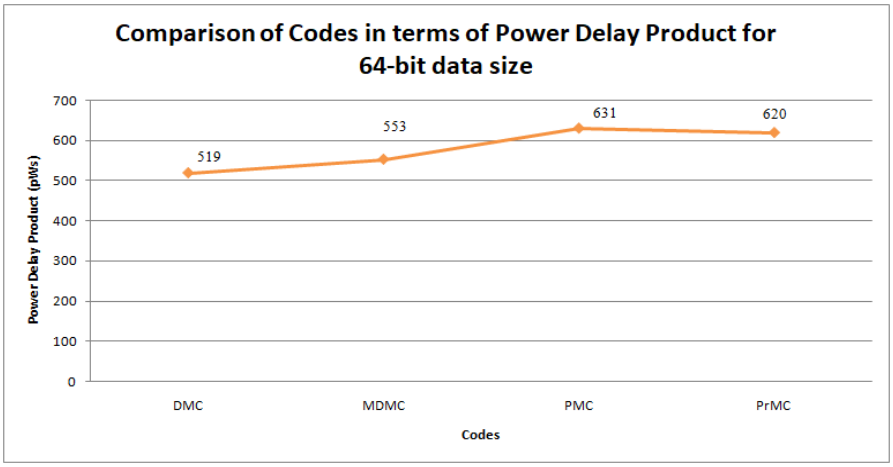

Figure 12. Comparison of 8-port router with various codes in terms of power delay product

4. CONCLUSION

In network on Chips, routers are the most prone to error, resulting in corruption of data or deadlock necessitating packet retransmission in end-to-end and hop-to-hop configurations. To prevent retransmission and ensure reliability, EDAC codes are proposed. Matrix codes are the main aspect of this project. The proposed matrix code, PrMC is compared with existing DMC, MDMC and PMC codes. The codes are written in Verilog HDL and tested with Zynq (XC7Z100-1FFG1156) FPGA in Xilinx ISE 14.5 Tool. The proposed code is efficient in many ways, with the only drawback being a slight increase in delay. Also 65% code rate is achieved with 9.26% increase in code word, 22.73% less redundant bits and 6.05% high code rate. The proficient Matrix codes prove to be a better choice for implementing EDAC codes in NoC routers as the area decreases by atleast 11.78% and with a decrease of atleast 1.74% in power delay product.

ACKNOWLEDGEMENTS

We thank the management and faculty of Electronics and Communication Engineering, Sree Vidyanikethan Engineering College, Tirupati for providing the tool and enormous support.

REFERENCES

[1] M. Katta, T. K. Ramesh, and J. Plosila, "SB-router: A swapped buffer activated low latency network-on-chip router," *IEEE Access*, vol. 9, pp. 126564–126578, 2021, doi: 10.1109/ACCESS.2021.3111294.

[2] K. Khalil, O. Eldash, A. Kumar, and M. Bayoumi, "Self-healing router approach for high-performance network-on-chip," *IEEE Open Journal of Circuits and Systems*, vol. 2, pp. 485–496, 2021, doi: 10.1109/ojcas.2021.3095000.

[3] A. Sarihi, A. Patooghy, A. Khalid, M. Hasanzadeh, M. Said, and A. H. A. Badawy, "A survey on the security of wired, wireless, and 3D network-on-chips," *IEEE Access*, vol. 9, pp. 107625–107656, 2021, doi: 10.1109/ACCESS.2021.3100540.

[4] R. Uma, H. Sarojadevi, and V. Sanju, "Network-On-Chip (NoC) - Routing techniques: a study and analysis," in *2019 Global Conference for Advancement in Technology, GCAT 2019*, Oct. 2019, pp. 1–6, doi: 10.1109/GCAT47503.2019.8978403.

[5] A. Naik and T. K. Ramesh, "Efficient network on chip (NoC) using heterogeneous circuit switched routers," in *2016 International Conference on VLSI Systems, Architectures, Technology and Applications, VLSI-SATA 2016*, Jan. 2016, pp. 1–6, doi: 10.1109/VLSI-SATA.2016.7593043.

[6] K. N. Dang and X.-T. Tran, "An adaptive and high coding rate soft error correction method in network-on-chips," *VNU Journal of Science: Computer Science and Communication Engineering*, vol. 35, no. 1, Jun. 2019, doi: 10.25073/2588-1086/vnucsce.218.

[7] C. Killian, C. Tanougast, F. Monteiro, and A. Dandache, "Smart reliable network-on-chip," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 2, pp. 242–255, Feb. 2014, doi: 10.1109/TVLSI.2013.2240324.

[8] N. Ashokkumar and A. Kavitha, "Nova 3D NoC shema za unicast i multicast usmjerivačke protokole velike propusne moći," *Tehnicki Vjesnik*, vol. 23, no. 1, pp. 215–219, Feb. 2016, doi: 10.17559/TV-20141230061413.

[9] N. A. Kumar and A. Kavitha, "Soft-error-resilient LDPC architecture on SoC using NoC decode algorithm," *Australian Journal of Basic and Applied Sciences*, vol. 8, no. 18, pp. 106–111, 2014.

[10] N. Y. Phing, M. N. M. Warip, P. Ehkan, R. B. Ahmad, and F. W. Zulkefli, "Performances analysis of reducing router in ring and mesh topology for network-on-chip (NoC) architecture," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 14, no. 2, p. 802, May 2019, doi: 10.11591/ijeecs.v14.i2.pp802-809.

[11] M. Huang, W. Liu, T. Wang, H. Song, X. Li, and A. Liu, "A queuing delay utilization scheme for on-path service aggregation in services-oriented computing networks," *IEEE Access*, vol. 7, pp. 23816–23833, 2019, doi: 10.1109/ACCESS.2019.2899402.

[12] G. Xu, Y. Shi, X. Sun, and W. Shen, "Internet of things in marine environment monitoring: a review," *Sensors*, vol. 19, no. 7, p. 1711, Apr. 2019, doi: 10.3390/s19071711.

[13] G. M. Walunjkar and K. R. Anne, "Performance analysis of routing protocols in MANET," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 17, no. 2, pp. 1047–1052, Feb. 2020, doi: 10.11591/IJEECS.V17.I2.PP1047-1052.

[14] M. Aqib, R. Mehmood, A. Alzahrani, I. Katib, A. Albeshri, and S. M. Altowaijri, "Smarter traffic prediction using big data, in-memory computing, deep learning and gpus," *Sensors (Switzerland)*, vol. 19, no. 9, p. 2206, May 2019, doi: 10.3390/s19092206.

[15] S. D. Lakshmi, K. Neelima, and C. Subhas, "Proficient matrix codes for NOC applications," in *4th International Symposium on Devices, Circuits and Systems, ISDCS 2021 - Conference Proceedings*, Mar. 2021, pp. 1–4, doi: 10.1109/ISDCS52006.2021.9397914.

[16] S. Leonelli and N. Tempini, *Data journeys in the sciences*. Cham: Springer International Publishing, 2020.

[17] Q. Song, H. Ge, J. Caverlee, and X. Hu, "Tensor completion algorithms in big data analytics," *ACM Transactions on Knowledge Discovery from Data*, vol. 13, no. 1, pp. 1–48, Jan. 2019, doi: 10.1145/3278607.

[18] J. Guo, L. Xiao, Z. Mao, and Q. Zhao, "Enhanced memory reliability against multiple cell upsets using decimal matrix code," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 1, pp. 127–135, Jan. 2014, doi: 10.1109/TVLSI.2013.2238565.

[19] K. Neelima and C. Subhas, "Multiple adjacent bit error detection and correction codes for reliable memories: a review," in *Lecture Notes in Electrical Engineering*, vol. 722 LNEE, 2021, pp. 357–371.

[20] K. Neelima and C. Subhas, "Half diagonal matrix codes for reliable embedded memories," *International journal of health sciences*, pp. 11664–11677, May 2022, doi: 10.53730/ijhs.v6ns2.8117.

[21] G. S. Lakshmi, K. Neelima, and C. Subhas, "Error detection and correction methods for memories used in system-on-chip designs," *International Journal of Engineering and Advanced Technology*, vol. 8, no. 2, pp. 60–66, 2019.

[22] N. Nimisha, P. Rajkumar, and S. Rajkumar, "Error correction codes using burst and random errors for multiple cell upsets in space application," *International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineer*, vol. 8, no. 5, 2020, doi: 10.17148/IJIREEICE.2020.8507.

[23] N. Koppala and C. Subhas, "Low overhead optimal parity codes," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 20, no. 3, pp. 501–509, 2022, doi: 10.12928/TELKOMNIKA.v20i3.23301.

[24] S. Manoj and C. Babu, "Improved error detection and correction for memory reliability against multiple cell upsets using DMC & PMC," in *2016 IEEE Annual India Conference, INDICON 2016*, Dec. 2017, pp. 1–6, doi: 10.1109/INDICON.2016.7839094.

[25] G. P. Acharya, M. A. Rani, G. G. Kumar, and L. Poluboyina, "Adaptation of march-SS algorithm to word-oriented memory built-in self-test and repair," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 26, no. 1, pp. 96–104, Apr. 2022, doi: 10.11591/ijeecs.v26.i1.pp96-104.

[26] K. Neelima and C. Subhas, "Efficient adjacent 3D parity error detection and correction codes for embedded memories," in *Proceedings of CONECCT 2020 - 6th IEEE International Conference on Electronics, Computing and Communication Technologies*, Jul. 2020, pp. 1–5, doi: 10.1109/CONECCT50063.2020.9198452.

[27] A. Y. Ardiansyah and R. Sarno, "Performance analysis of wireless sensor network with load balancing for data transmission using xbee zb module," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 18, no. 1, pp. 88–100, Apr. 2019, doi: 10.11591/ijeecs.v18.i1.pp88-100.

[28] A. Samy, A. Y. Hassan, and H. M. Zakaria, "Improving bit error-rate based on adaptive Bose-Chaudhuri hocquenghem concatenated with convolutional codes," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 2, pp. 890–901, Aug. 2021, doi: 10.11591/ijeecs.v23.i2.pp890-901.

[29] Y. N. S. Kumar and P. Dinesha, "TFI-FTS: An efficient transient fault injection and fault-tolerant system for asynchronous circuits on FPGA platform," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 3, pp. 2704–2710, Jun. 2021, doi: 10.11591/ijece.v11i3.pp2704-2710.

BIOGRAPHIES OF AUTHORS

Neelima Koppala received her M.Tech degree in Embedded Systems from JNTUA Ananthapuramu in 2015, and is now currently working as Assistat Professor in ECE department, Sree Vidyanikethan Engineering College, Tirupati. Her current research interests include information theory, VLSI, and image processing. She can be contacted at email: neelumtech17@gmail.com.

Dr. Ashokkumar Nagarajan received B.E degree in Electronics and CommunicationEngineering from Anna University, Chennai, in 2007, M.E.degree in Applied Electronics from, Anna University Trichy in 2010 and Ph.D. Degreein the title of Performance Analysis of 2D and 3D Network-On-Chip Architectures from Anna University, Chennai in 2017. Currently, he is an Associate Professor in the department of Electronics and Communication Engineering at Sree Vidyanikethan Engineering College, Tirupati, Andra Pradesh, India. His research interests include network-on-chip, system-on-chip and low-power and digital VLSI design. She can be contacted at email: ashoknoc@gmail.com.

Satyam is an Assistant Professor at Sree Vidyanikethan Engg. College, Tirupati, Andhra Pradesh, India. He holds a Master degree in Wireless Communications. His research areas are electromagnetics, antenna design, microwave engineering, communication, wireless routing protocols and mobile Adhoc networks. He is a life time editorial member of Edwin Group of Journals. He has many research papers in International Conferences and Journals to his credit. He is a recipient of Best Research Paper award at IEEE International Conference on Emerging Smart Computing and Informatics (ESCI 2022-Virtual Mode). He has recently filed a patent on his innovative ideas. He can be contacted at email: mesatyam0101@gmail.com.

Neeruganti Vikram Teja is Assistant Professor at college of Electronics and Communication Engineering, Sree Vidyanikethan Engineering College, Tirupati, Andhra Pradesh, India. He Holds M.Tech. degree in VLSI. His research areas are VLSI, Nano electronics, and Nanofabrication. His research interests include in VLSI, device fabrication of Nano devices. He can be contacted at email: vikramtejan@gmail.com.