Design and performance analysis of asynchronous network on chip for streaming data transmission on FPGA

Trupti Patil, Anuradha M. Sandi
Department of Electronics and Communication Engineering, Guru Nanak Dev Engineering College, Bidar, India

Article Info

Article history:

Received May 27, 2023
Revised Sep 17, 2023
Accepted Sep 29, 2023

Keywords:

BT-based power optimization and streaming data
Label switching
Network interface
Network on chip
NoC manager
UART

ABSTRACT

The majority of the system on chip (SoC) uses the network on chip (NoC) as routing ports for data transfer from node-to-node with minimal power consumption and low latency and high throughput. This paper concentrates on the ability to model the asynchronous NoCs on the asynchronous circuits on field programmable gate arrays (FPGAs). A 3×3 NoC and its universal asynchronous receiver transmitter (UART) protocol is designed and its simulation of the Verilog hardware description language (VHDL) code is done and tested on the Artix-7 FPGA kit, the testing processes in done using the Chipscope tool. In order to meet target requirements in terms of power consumption and latency, the label switching (LS) technique is used as routing. The proposed LS-NoC with level-encoded dual-rail (LEDR) encoding technique provides throughput by registering the packet between the different routers and it helps to improve throughput and speed. The effectiveness of the data transfer is measured and analyzed through a synthesis summary in terms of lookup table's (LUT's), slice registers, flip flops's (FF's), latency, and packet delivery ratio (PDR) for the traffic pattern generator. The proposed NoC is designed for 8×8 and each port size is 21 bits including ID's of source and destination routers. The results can be justified by following results: improvement of LUTs is about 12%, flip-flops are 7%, improvement of throughput is 23% and delay is reduced by 26%.

Corresponding Author:

Trupti Patil
Department of Electronics and Communication Engineering, Guru Nanak Dev Engineering College
Bidar, 585403, Karnataka, India
Email: patiltrupti11@gmail.com

1. INTRODUCTION

One of the developing fields contained by the system on chip (SoC) analysis area is network on chip (NoC). There are several papers published by researchers about the developing fields of NoC systems and their application [1]. In the meantime, there are several improvements and developments in the structure of asynchronous and asynchronous NoCs [2]. Message-passing asynchronous NoC is guaranteed service over open core protocol (OCP) interfaces and is developed to a fully grown network in high speed NoC [3], [4]. The favorable services offered by the asynchronous message-passing asynchronous NoC providing guaranteed services over OCP interfaces (MANGO) are bounded services [5], [6]. The interfacing of OPC collaborates with NoC, this is associated with the core. The global science (GS) network and the built environment (BE) network are the two main components of any NoC network [7], [8]. The virtual channels support the connection-oriented GS services, these services are measured with the latency and hard information that promises better utilization. The BE network is empowered with the packets that are routed within the wormhole routers [9]. In the initial research, we often find the execution of the asynchronous circuits on field programmable gate array (FPGA) is very narrow and confined [10], [11]. So, here we are eager to implement a well-sophisticated approach that makes the implementation a better way in the execution. The NoC has been initiated with the best-effort NoC, in the elementary asynchronous mode [12]. A router, master, network adapter (NA), and slave NA are the three components of the NoC. The routers are interconnected in a mesh topology and the warn hole routing is used for the communication. The use of supply routing and the XY-routing was issued to avoid the deadlocks [13]. The number of lists can be unlimited in the packets [14].

The four basic elementary units of NoC are intellectual property (IP) cores, NA, routers (R0 to R8), and links. Figure 1 shows the outline of a 3×3 NoC module. NoC is a super technique where we can see the cores within can easily communicate with each other in a very accurate way [15]. The execution and implementation of the FPGA are completely theoretical, so it is preferred to execute the BE NoC that has been performed. The primary concern of the thesis is availability, and the least prioritized issue is execution [16]. The area of the complete structure is low, this is result of the accessible logical resources on the given FPGA [17]. The next part of the thesis will eventually show the correct model for the selection of NoC design [18]. The topology selected should be suitable for the outlines that are specified by the FPGA. The conditions of the topology that are to be concerned are listed in [19]. The successor of the next node shall always be a one-directional link like a torus or a K-Ary 2 cube mesh or the torus topology. At the stage of selection, a two-way link of A K-Ary 2 cube network is selected. The basic reason for the selection is to be free from the deadlock that occurs, whereas the torus has a huge abundant number of links [20]. If the topology is integrated with XY routing, the deadlocks can be removed without the simulations of virtual channels the architecture of FPGA has a well onto the structure of topology in two dimensions. The further needs of a K-Ary 2 cube network topology are: there are four ports for network connections, one port for a core affiliation, and p described in [21].

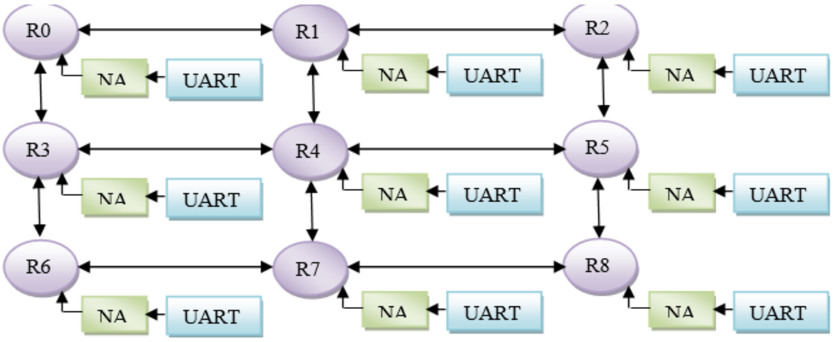


Figure 1. General structure of NoC connected in a 3-by-3 mesh topology

2. PROPOSED LOW POWER ROUTER DESIGN FOR LABEL SWITCHING-NOC’S

Label switching (LS) technique is used in many networks such as automatic teller machines (ATMs) and banking applications since it is purely dependent on packet relaying because LS will carry route information in the form of labels within the network. Another function of LS is to change the direction in X-Y coordinates for transmission of a packet from one route to another route by identifying the next router through forwarding information, quality of service (QoS), guarantee, and traffic priority and finally, it assigns to nest route label [22]. The LS is applied for the transmission of screaming data with more area consumption and high power utilization. The microarchitecture of the single router is shown in Figure 2 and it consists of first in first out (FIFO) and its control block, NoC manager, crossbar switch, and arbiter. This proposed work is mainly concentrated on reducing power using the bit transition encoder and decoder (BTED) technique as shown in (3) and (4). The existing LS-based NoC is for streaming applications that limit latency and hardware utilization. These limits are mainly addressed in this research work with the help of a NoC manager which can monitor and control bandwidth sharing and its adjustment automatically. The NoC manager uses a flow graph (FG) to represent communication between source and destination nodes which are updated and stores the packet and updates their bandwidths in a table known as the routing table, the source router present in FG is to process the packet which is generated through traffic generator is processing engine through input and output ports. This engine and input and output ports receive the data to form sink nodes in the FG from source to destination and intermediate nodes are represented as edges and stored in the FG is given in Figure 2 and its edges are shown in Table 1.

$$E_{ij} = \{N_i, N_j, u_{ij}, A_{ij}, LP_{ij}^{used}, LP_{ij}^{unused}\} \quad (1)$$

Where E_{ij} is the edges connected between the source (s) and destination (d) and these can be any nodes out of 64 nodes. The N_i is source node, N_j is the destination node, u_{ij} is utilized (already used by another router), A_{ij} is the available node (or free node in FG), LP_{ij}^{used} is node present in the list of labels used in the pipe through E_{ij} and LP_{ij}^{unused} is node present in FG which is not used by any other router.

During transmission of the packet, LP_{ij}^{used}, LP_{ij}^{unused} are equal to "NULL" when no data is available and their capacity or bandwidth are completely utilized and not available to serve further with any other router for data transmission. During the transmission of data, A_{ij} will have maximum capacity or bandwidth and LP_{ij}^{unused} is not used by any other router and it will be free for serve and available for data transmission. For effective data transmissions, the pipe should have maximum capacity 'c' and it will establish communications between source (s) and destination (d).

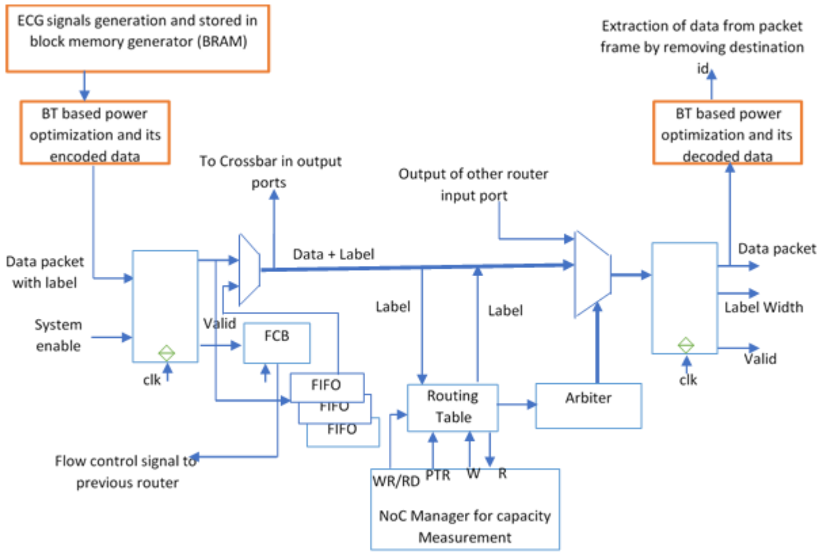

ECG signals generation and stored in block memory generator (BRAM)
BT based power optimization and its encoded data
Extraction of data from packet frame by removing destination id
BT based power optimization and its decoded data
To Crossbar in output ports
Output of other router input port
Data packet with label
Data + Label
Data packet
System enable
Valid
FCB
Label
Label
Label Width
Valid
clk
clk
FIFO
FIFO
FIFO
Routing Table
Arbiter
Flow control signal to previous router
WR/RD
PTR
W
R
NoC Manager for capacity Measurement

Figure 2. Proposed label switched-based microarchitecture of single router with single-cycle flit traversal and their internal micro blocks including power optimization using BTED

Table 1. LS-NoC and BTED based NoC and its routing table's from 33 to 16 and 17 to 5

Ei	Ni	Nj	uij	Aij	LP_{ij}^{used}	LP_{ij}^{unused}
25_26	1	6	10	0	{0}	{0..64}
27_28	2	8	10	0	{0}	{0..64}
33_34	3	4	0	10	{}	{1..64}
34_35	4	5	0	10	{}	{1..64}
36_37	5	7	0	10	{}	{1..64}
38_39	6	9	0	10	{}	{1..64}
30_31	7	1	10	0	{0}	{0..64}
39_40	8	1	10	0	{0}	{1..64}
40_32	9	3	10	0	{0}	{1..64}
32_24	10	5	10	0	{0}	{1..64}
24_16	11	6	10	0	{0}	{1..64}

In the proposed design, the major sub-systems are routers, network adaptors, switching algorithm, label-based routing technique and power optimization method. These all are integrated as SoC level to meet requirements of IP with optimal power, area, latency and throughput. All these sub-systems are part of NoC and it is integrated as SoC systems for interfacing with high speed Cortex-M33 processors and other controllers through different protocols [23]. In Algorithm 1, the first step is for FG creations, from which, the input packet includes both information data and destination id as labels and FG contains the number of edges and capacity determination [24]. All edges in FG will change their directions from E_{ij} to $E_{ji,}$ based on the next router which is depending on the destination node. The FG also stored the capacity of each and every link (path). The third step is to monitor the number of packets transmitted and received between routers. In the fourth step, the data is stored in output ports when the source and destination node is the same. The remaining steps are to perform the data transmission based on the bandwidth available at every node and finally, the received data is stored in Ps. Once the packet is reached the destination, FG will update its edges and push the packet data to output ports, and also store it in stack pointer (sp). After the destination node and pipe is identified, the SP is used for updating the used list and pipe (LP_{ij}^{used}) and available (LP_{ij}^{unused}) flows based on capacity or bandwidth. Once FG is updated, the NoC manager will configure the routing table, and label updating is performed in each and every intermediate router. Along with FG updating, the LP_{ij}^{used} of each edge will check for conflicts, if it is a conflict then the NoC manager will identify the alternative port or router which is unused in the routing table. This table data structure at a node can be written as shown in (2):

$$N_i, P_{old} \rightarrow N_j E_{new} \tag{2}$$

where P_{old} is the pipe label in the edge ending at N_i and E_{new} is the pipe label in the edge in E_{ij}.

Algorithm 1. NoC manager: identification of pipe, ps: pipe stack

```
Define the source node as the pipe in the NoC:
  Input required: $E_{ij} = \{N_i, N_j, u_{ij}, A_{ij}, LP_{ij}^{used}, LP_{ij}^{unused}\}$, s, d and c
Define flow graph (FG) ={$E_{ij}$}=$\{N_i, N_j, u_{ij}, A_{ij}, LP_{ij}^{used}, LP_{ij}^{unused}\}$
Initialize counter value to ‘0’ i.e k=0
Suppose s=d then
  Do not perform any process and store source packet into output ports of same router and
update ps=s{data}
if s != d then
  For all edges starting from s, s+1,… to d  perform loop
          If $A_{ij}$> c then {If available node capacity is greater than c}
                    Data_out = data packet is sends to next router input port and update the
                    FG based on node id and then push data into sp
          If $A_{ij}$<c then
Search for alternative router and its free input and output ports and then push into sp. If
d= destination id then
Ps = d{data} and update FG and extract data packet by removing the label bits
End
```

The Figure 3 shows 8×8 LS-NoC in 2D mesh topology. The NoC manager is part of every router and each router has five input and output ports (East, North, South, West, And Local) and processing elements along with IP blocks that store the received packet at destination node. The single LS-based router is designed using combinational circuits between input and output ports. The received data from the source system i.e. the device which is generating electrocardiogram (ECG) signals are stored in FIFO if other flits are awaiting traversal or if the arbiter does not provide grant access to the output port [25]. The FIFO control block (FCB) will take care of the FIFO pointer arithmetic and control the corresponding input port’s signal flow.

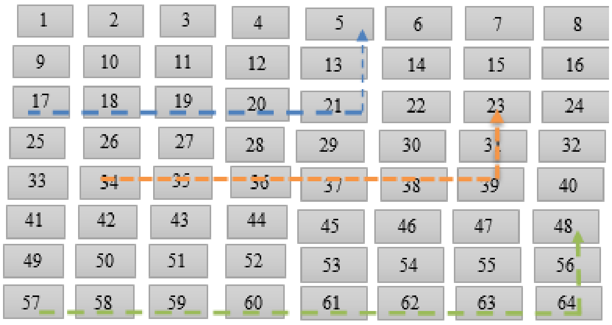


Figure 3. FG of the LS-NoC and BTED architectures during data packet transmission with two sources and two destinations marked as red and green

The FIFO-based router design of LS-NoC can handle multiple clock domains which are asynchronous. Multi-clock buffers can be used in place of the buffers at the router's output port, and they can be linked to dual clock interfaces. Because of the nature of the pipe formation process, LS-NoC has built-in fault tolerance. Following the detection of a defective connection, the LS-NoC manager takes the following steps:

– When a bad connection is detected, the LS-NoC Manager sets the capacity of that link to 0 in the FG.
– Existing pipes connected to the connection are deactivated. The pipes have been renamed, and the routing tables have been modified.
– After pipes are configured, the FG is updated.

The NoC manager's overhead is made up of two parts: computation and configuration. Identifying a pipe with a flow-based method (Algorithm 1) incurs computational cost. Routing table configuration is transmitted across the network and routing tables are updated as part of the configuration overhead (Table 1).

In the proposed design, the major sub-systems are routers, network adaptors, switching algorithm, label-based routing technique and power optimization method. These all are integrated as SoC level to meet requirements of IP with optimal power, area, latency and throughput. All these sub-systems are part of NoC and it is integrated as SoC systems for interfacing with high speed Cortex-M33 processors and other controllers through different protocols.

3. BIT TRANSITION ENCODER/DECODER FOR POWER OPTIMIZATION IN NoC

The power consumption and its optimization in NoC is major challenging task and it will degrade the performance level. In this work as shown in Figures 4 and 5, bit transition encoder technique is applied before transmission of packet to source router and after receiving packet at destination router for power optimization. In any on chip memory or networks, power consumption is depending on number transitions such as bit 1 to bit 0 (formally known as type 1) or bit 0 to bit 1 (formally known as type 2), there is not bit transition if both bits are same like bit 0 to bit 0 (formally known as type 3) or bit 1 to bit 1 (formally known as type 4). The power reduction technique will work only on type 1 and type 2. The power optimization purely works based on number bits transitions in packet data, if there are more number of transition bits then encoding techniques is going minimize before sending the packet to next router. The generalized logical expression for encoding are given in (3) and (4).

$$Encoded_i = data_i \oplus data_{i-1} \oplus FIforalloddnumber \in i \tag{3}$$

$$Encoded_i = data_i \oplus data_{i-1} \oplus FI \oplus HIforallevennumber \in i \tag{4}$$

Where FI is full invert, it can be either 1 or 0 and HI is half invert, its bit is same as FI, $data_i$ is present bit in given packet and $data_{i-1}$ is previous bit in given packet, between these two bits, the XOR operations is performed to reduce number transitions, for example, let consider number of bits in packet is 16 bits (let say: 1010101010101010), the number of transitions are 15. After performing bit transition encoder on packet through XOR operation, encoded bits are 1111111111111111 as shown in Figure 3, number of transitions in encoded bits are 0, therefore number transitions are reduced from 15 to 0.

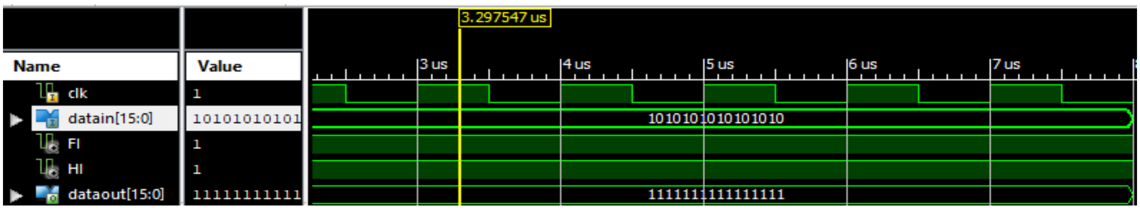


Figure 4. Simulated results of power reduction through bit transition encoder technique, here input is 16 bits (1010101010101010) and output is 16 bits (1111111111111111)

The encoded packet is transmitted from source router to destination router, at destination router, after receiving packet before decoding, the bit transition decoder is applied to decode the original packet. The generalized logical expression for decoding are given in (5) and (6):

$$Decode_i = Rc_{data_i} \oplus Rc_{data_{i-1}} \oplus FIforoddbitsofi \tag{5}$$

$$Decode_i = Rc_{data_i} \oplus Rc_{data_{i-1}} \oplus FI \oplus HIforevenbitsofi \tag{6}$$

after applying (3) and (4), the simulated results are shown in Figure 3, the decoded packet bits are same as packet bits which is transmitted at source node.

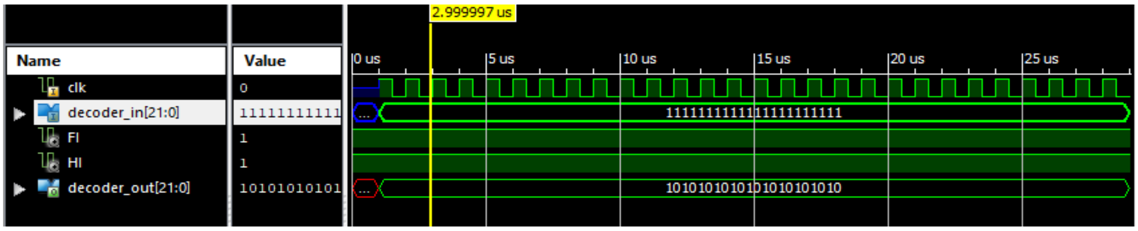
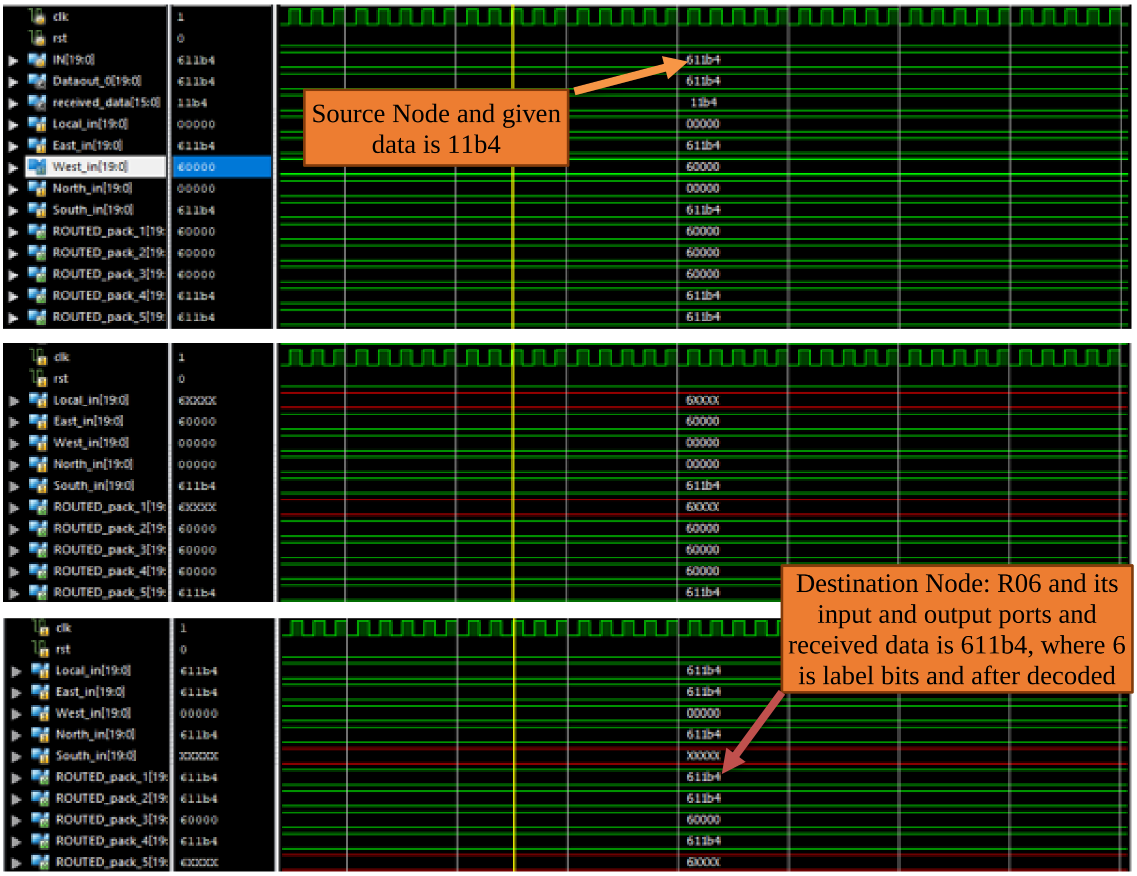


Figure 5. Simulated results of power reduction through bit transition decoder technique, here input is 16 bits (1111111111111111) and output is 16 bits (1010101010101010)

4. RESULTS AND DISCUSSIONS

The proposed LS-NoC with power optimization technique is successfully synthesized using Xilinx Design Suite 14.7 software tool and implemented on Artix-7 FPGA development. The delay and throughput and figure of merit are analyzed between source (R_{00}) and destination (R_{06}) nodes through simulated results shown in Figure 5. In order to proof the latencies between different routers, considered first router is always source routers and others are destination routers, the latency is measured from source to any other routers as shown in Table 2. The second column in the Table 2 shows different latencies, for example 10 and 15 is latency from router 3 to router 6 (shown in destination node column). Similarly, for throughput and frequencies are shown in Table 2. In Figure 6, the very first signal is clock of 100 MHz followed by input and output data of source router and destination routers and they are highlighted in the separate box.


Source Node and given data is 11b4

Destination Node: R06 and its input and output ports and received data is 611b4, where 6 is label bits and after decoded

Figure 6. Simulated results of 3×3 LS-NoC and their received data at each input and output ports

Throughput (thp): The proposed 8×8 NoC system's throughput is calculated as the ratio of the total amount of bits to be transmitted by the simulation time, in seconds, to the total number of bits to be conveyed in a given time, per sec, and is represented as (7).

$$thp = \frac{np_{ck}*packet_{flit_{zise}}}{nsim_{pkt}*T} \quad (7)$$

The number of packets transmitted per clock is given by np_{ck}. The np_{ck} is no. of transmitted packets per cycle, as $packet_{flit}$ is packet size of 16 bits, $flit_{zise}$ is size of flit of 16 bits, $nsim_{pkt}$ is total latency of every packet transmission and T is total cycle period.

The hardware platform for implementing NoC with considerably different factors in the proposed system is the Xilinx Design Suite, which has already been used by the prior systems. Table 3 compares the summary of previous work with the proposed work. The relative plots and thorough analysis of NoC with and without the application of power optimization technique are shown in Figures 7 and 8.

As a result, when compared to the existing work, the suggested system performs better in all parameters, whether or not LS is used to store and then transmit ECG signals, as shown in Figure 9. The proposed LS-NoC is extended from 3×3 to 8×8 to analyse latency and routing paths that having totally 64 routers, the simulated results of 3×3 is shown in Figure 10 for the source 3 and destination node 6.

Table 2. Source and destination router

Parameter	Source router/value of delay/TP/latency	Destination node			
Node	R00 TO				
Delay (D) in ns	10, 16 (as per Figure 6)	R03	R06		
Delay (D) in ns	10, 16, 40 (as per Figure 7)	R03	R04	R06	
Delay (D) in ns	10, 16, 35,40 (as per Figure 8)	R03	R04	R05	R06
Throughput (TP) in MHz	6, 4, 2.2, 5.9	R03	R06		
TP in MHz	3.1, 3.2, 3.1, 3.4	R03	R04	R06	
TP in MHz	4.1, 4.6, 4.9	R03	R04	R05	R06
Latency (L) in pico seconds	0.9, 0.4, 2.1, 4.9	R03	R06		
L in pico seconds	4, 3, 3.1, 5.9	R03	R04	R06	
L in pico seconds	3.1, 3.3, 3.5, 3.5	R03	R04	R05	R06

Table 3. Summary of the existing work with proposed work

Parameter	Without ECG signals		With ECG signals	
	Existing work [26], [27]	Proposed work	Existing work	Proposed work
Slice registers	3930	2399	3634	2864
Slice LUT's	4671	2590	4090	2972
Slice FF's	2495	1855	3635	2843
Delay in ns	5	3.024	19.285	14.4
PowermW	43.06	8.2	214	82
Area	16854	3694	12353	2529
Frequency (MHz)	312.69	391.850	75.27	450.85
Throughput (Gbps)	80	260.8	---	170.8

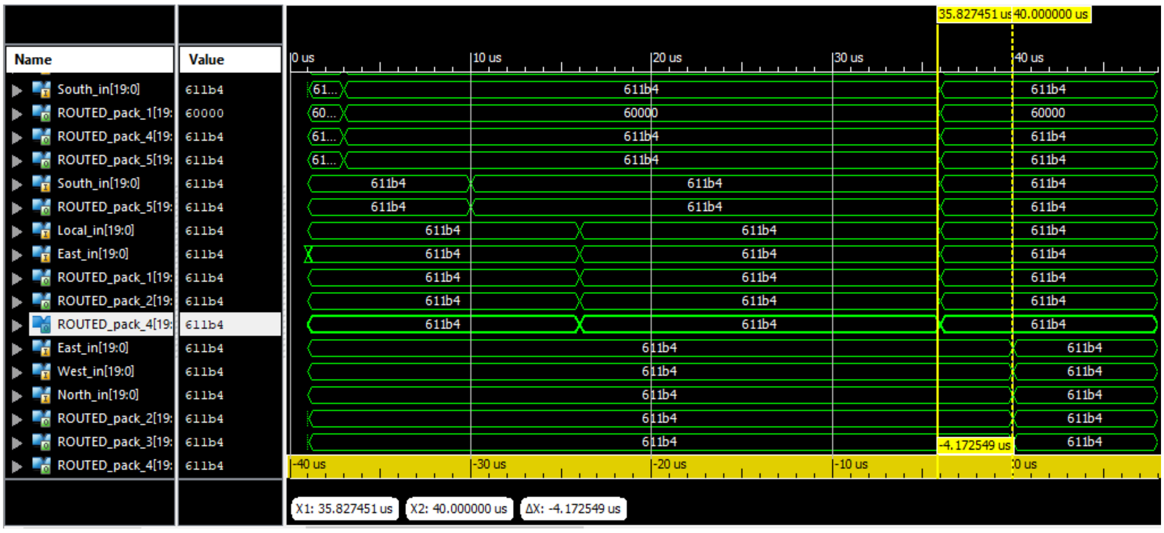


Figure 7. Delay calculation between source and destination of 3×3 NoC, intermediate nodes are 3 and 6, node 3 is received data at 10 ns, node 4 is received the data at 40 ns and node 6 received at 16 ns

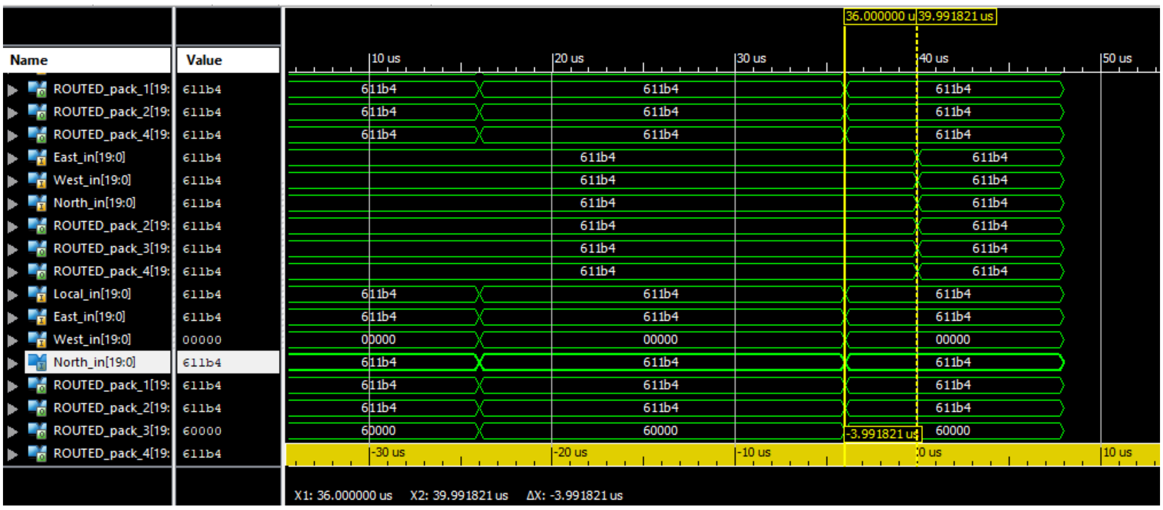


Figure 8. Delay calculation between source and destination of 3×3 NoC, intermediate nodes are 3, 4, 5 and 6, node 5 is received data at 35 ns, node 6 is received the data at 40 ns

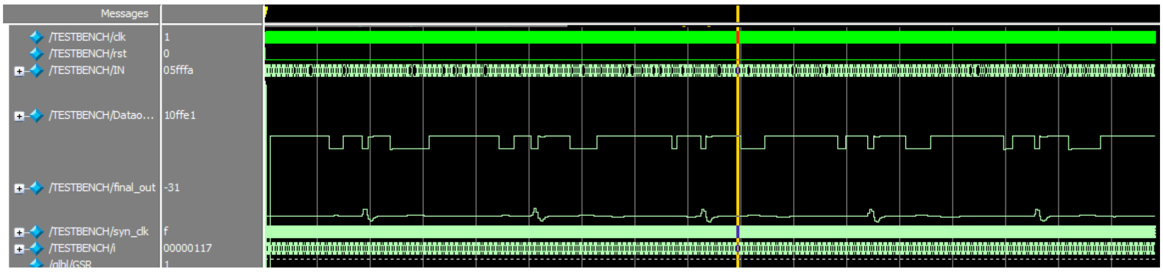


Figure 9. Simulated results of LS-NoC for ECG signals transmitted from source to destination nodes

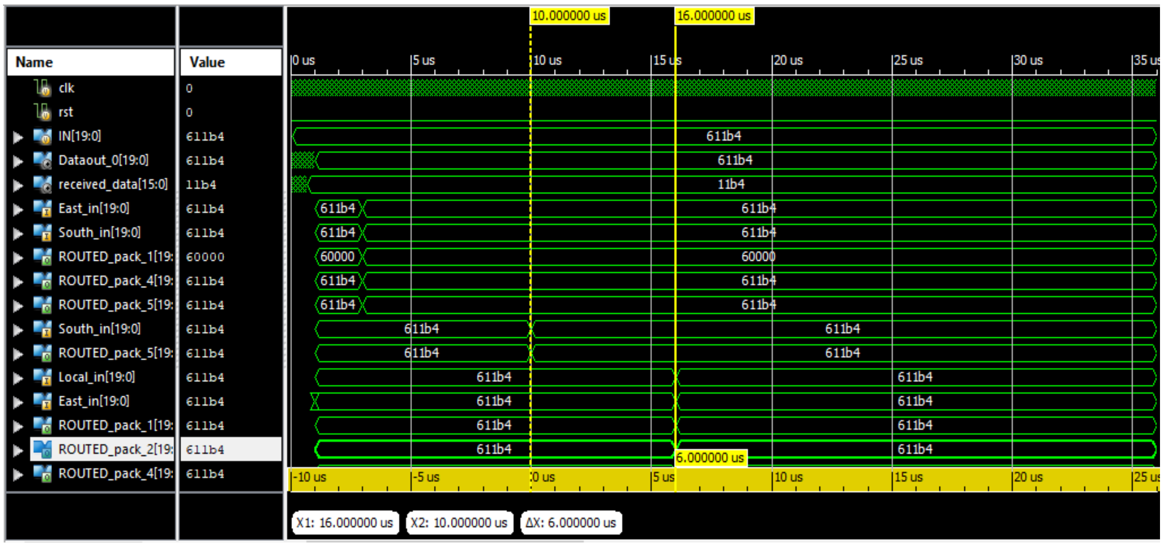
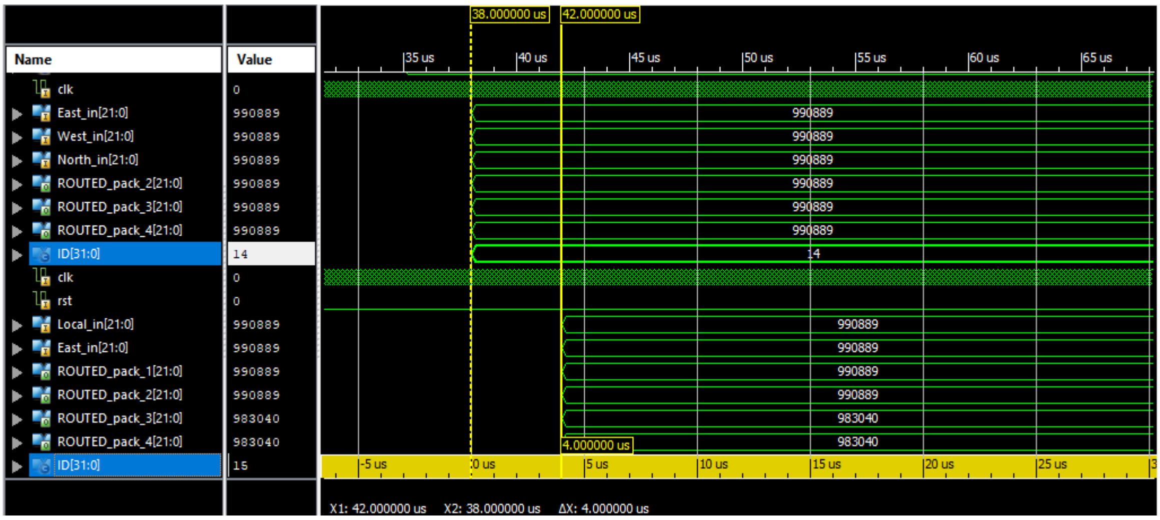


Figure 10. Delay calculation between source and destination of 3×3 NoC, intermediate nodes are 3 and 6, node 3 is received data at 10 ns and node 6 received at 16 ns

To analyse details latency, we have considered source id is 01 and destination id is 15 as shown in Figure 11, it is found that 1 ns delay for router 01 to 03 and 3 ns delay for router 04 to 15. The packet is transmitted from source 01 through 02, 06, 08, 09, 10, 11, 12, 13, 14 and to 15 as shown in Table 2. Due to congestion and contention because of conflict, packet is not travelled through 03, 04, 05, and 07, this is clearly observed in simulated results as shown in Figure 12.



Name	Value
clk	0
East_in[21:0]	990889
West_in[21:0]	990889
North_in[21:0]	990889
ROUTED_pack_2[21:0]	990889
ROUTED_pack_3[21:0]	990889
ROUTED_pack_4[21:0]	990889
ID[31:0]	14
clk	0
rst	0
Local_in[21:0]	990889
East_in[21:0]	990889
ROUTED_pack_1[21:0]	990889
ROUTED_pack_2[21:0]	990889
ROUTED_pack_3[21:0]	983040
ROUTED_pack_4[21:0]	983040
ID[31:0]	15

X1: 42.000000 us X2: 38.000000 us ΔX: 4.000000 us

Figure 11. Simulated results of 8x8 LS-NoC for source:01 and destination: 15 and their latency

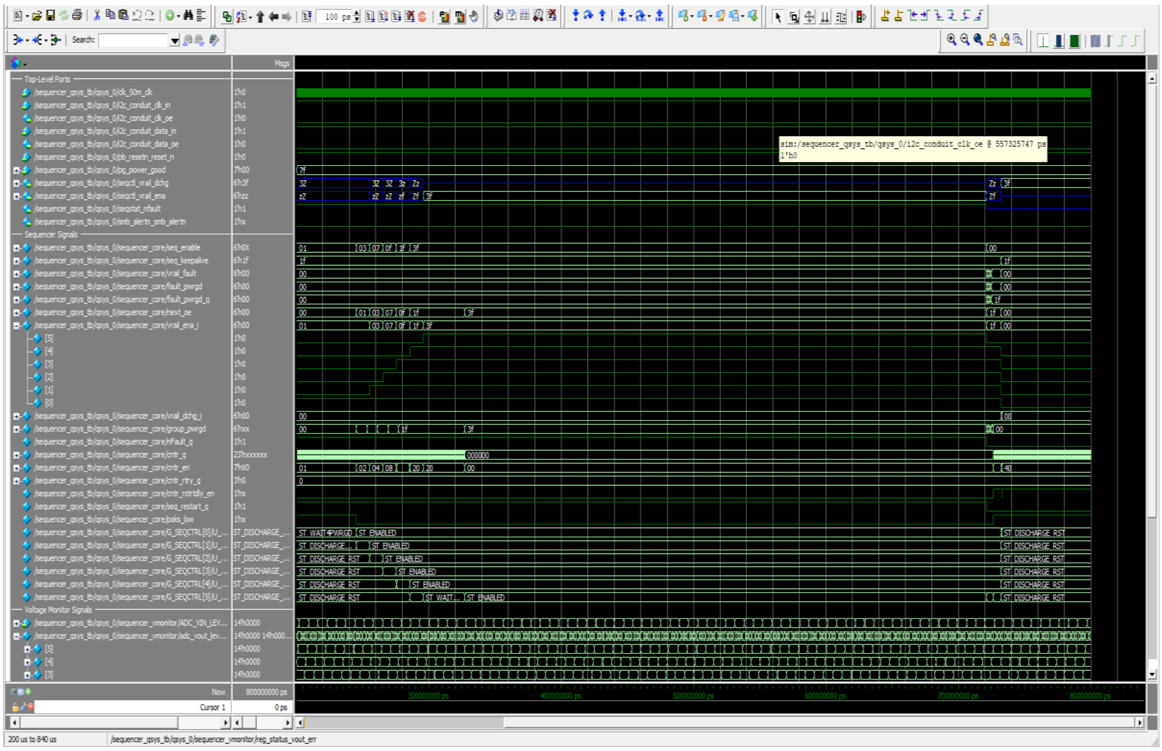


Figure 12. Simulated results of LS-NoC with LEDR encoding technique

The test bench of top-level design includes both LS-NoC and LEDR and its implementation includes 6-rail voltage with full functionality and inputs for the LEDR come from text files, in which the voltage levels are specified are continuously looped through during simulation as shown in Figure 10 Because of this, there is no effect from the VRAIL_EN signal on the simulated analog input (Voltage). The analog input will not rise when VRAIL_EN is asserted, nor will it fall with VRAIL_EN is de-asserted as shown in Figure 11.

5. CONCLUSION

Dynamically Reconfigurable network on chip (DRNoC) NoC uses mesh topology and XY-routing with deadlock freedom to minimize latency. The streaming data are converted into packet which consists of source, destination id’s and flit bits. These packets are encoded by adding two additional request signals like handshake signals. The proposed design has asynchronous clocks which are synchronizer is used to manage synchronization. The router has 1302 LUTs as well as 530 latches in its region, with delay elements using 12% of the LUTs. The router's overall output measured was found to be 46 MHz. Three CPUs as well as three external units make up the prototype, which is connected via a 3×2 mesh. The power, as well as the area used by router buffers in NoC, seem to be a major issue in the deep submicron domain which elimination of buffers. When compared to another conventional bufferless routing algorithm, the computational results demonstrate that the designed routing algorithm optimizes average latency by 22%, power consumption by 21%, as well as area overhead by 44%. An 8×8 switch router with a suitable shortest path detector, such as a minimal spanning tree, is utilized to design the suggested network architecture for effective run-time routing. Therefore, Verilog hardware description language (VHDL) is been chosen for executing in VIVADO Xilinx 2018-1 software and is implemented on Nexys DDR-4 Artix-7 FPGA family with a part number XCA7CGS100t, which has 324 pins, with improved accuracy as well as 35% latency and when compared to the conventional router, the proposed router increases the efficiency by 40% and this technique outperforms the traditional one in terms of delay, area as well resource allocation.

REFERENCES

[1] A. Viswanathan, N. Feldman, Z. Wang, and R. Callon, “Evolution of multiprotocol label switching,” *IEEE Communications Magazine*, vol. 36, no. 5, pp. 165–173, May 1998, doi: 10.1109/35.668287.

[2] D. Boulegane *et al.*, “Real-time machine learning competition on data streams at the IEEE big data 2019,” in *2019 IEEE International Conference on Big Data (Big Data)*, Dec. 2019, pp. 3493–3497. doi: 10.1109/BigData47090.2019.9006357.

[3] A. Mansour, A. Elnaggar, B. Alabassy, M. Khamis, and A. Shalaby, “A 4-PAM interconnect in network-on-chip for high-throughput and latency-sensitive applications,” in *2018 19th International Symposium on Quality Electronic Design (ISQED)*, Mar. 2018, pp. 112–118. doi: 10.1109/ISQED.2018.8357274.

[4] A. M. Sllame and N. Salama, “An MPLS-based fat tree network-on-chip systems,” in *2016 International Conference on Advances in Computing and Communication Engineering (ICACCE)*, Nov. 2016, pp. 29–36. doi: 10.1109/ICACCE.2016.8073719.

[5] T. M. VanEtten, A. C. Williams, J. Deng, F. Wang, and L. Gao, “SoC-based implementation of a lightweight label switching router,” in *2017 29th International Teletraffic Congress (ITC 29)*, Sep. 2017, pp. 126–129. doi: 10.23919/ITC.2017.8064347.

[6] S. K. Mandal, A. Krishnakumar, and U. Y. Ogras, “Energy-efficient networks-on-chip architectures: design and run-time optimization,” in *Network-on-Chip Security and Privacy*, Cham: Springer International Publishing, 2021, pp. 55–75. doi: 10.1007/978-3-030-69131-8_3.

[7] J. Fang, S. Liu, S. Liu, Y. Cheng, and L. Yu, “Hybrid network-on-chip: an application-aware framework for big data,” *Complexity*, vol. 2018, pp. 1–11, Jul. 2018, doi: 10.1155/2018/1040869.

[8] S. Xiao *et al.*, “Neuronlink: An efficient chip-to-chip interconnect for large-scale neural network accelerators,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 9, pp. 1966–1978, Sep. 2020, doi: 10.1109/TVLSI.2020.3008185.

[9] R. Yao and Y. Ye, “Toward a high-performance and low-loss Clos–Benes-based optical network-on-chip architecture,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 12, pp. 4695–4706, Dec. 2020, doi: 10.1109/TCAD.2020.2971529.

[10] Y. Chen and A. Louri, “Learning-based quality management for approximate communication in network-on-chips,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 11, pp. 3724–3735, Nov. 2020, doi: 10.1109/TCAD.2020.3012235.

[11] X. Li, Y. Miura, S. Kang, and Y. Sakamoto, “A scrambling technique embedding soundtracks into videos for streaming media,” in *2018 International Workshop on Advanced Image Technology (IWAIT)*, 2018, pp. 1–4. doi: 10.1109/IWAIT.2018.8369752.

[12] B. Amen and A. Grigoris, “A theoretical study of anomaly detection in big data distributed static and stream analytics,” in *2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, Jun. 2018, pp. 1177–1182. doi: 10.1109/HPCC/SmartCity/DSS.2018.00198.

[13] L. Bamberg, J. M. Joseph, R. Schmidt, T. Pionteck, and A. Garcia-Ortiz, “Coding-aware link energy estimation for 2D and 3D networks-on-chip with virtual channels,” in *2018 28th International Symposium on Power and Timing Modeling, Optimization and Simulation (PATMOS)*, Jul. 2018, pp. 222–228. doi: 10.1109/PATMOS.2018.8464171.

[14] Z. Jiang, K. Yang, N. Fisher, I. Gray, N. Audsley, and Z. Dong, “AXI-IC^{RT}: towards a real-time AXI-interconnect for highly integrated SoCs,” *IEEE Transactions on Computers*, pp. 1–1, 2022, doi: 10.1109/TC.2022.3179227.

[15] Y. R. Gonzalez and G. Nelissen, “HopliteRT: real-time NoC for FPGA,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 11, pp. 3650–3661, Nov. 2020, doi: 10.1109/TCAD.2020.3012748.

[16] P. V. Bhanu, R. Govindan, P. Kattamuri, J. Soumya, and L. R. Cenkeramaddi, “Flexible spare core placement in torus topology based NoCs and its validation on an FPGA,” *IEEE Access*, vol. 9, pp. 45935–45954, 2021, doi: 10.1109/ACCESS.2021.3066537.

[17] A. Kumar and V. K. Reddy, “Advanced FIFO structure for router in Bi-NoC,” in *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*, May 2021, pp. 1219–1224. doi: 10.1109/ICICCS51141.2021.9432353.

[18] M. A. Al-Shareeda, M. A. Saare, S. Manickam, and S. Karuppayah, “Bluetooth low energy for internet of things: review, challenges, and open issues,” *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 31, no. 2, pp. 1182–1189, 2023, doi: 10.11591/ijeecs.v31.i2.pp1182-1189.

[19] N. D. Majeed, S. Q. Mahdi, and M. A. Kadhim, “Implementation of 4×4 2D mesh NoC architecture using FPGA,” in *2021 1st Babylon International Conference on Information Technology and Science (BICITS)*, Apr. 2021, pp. 133–137. doi: 10.1109/BICITS51482.2021.9509904.

[20] N. Jindal, S. Gupta, D. P. Ravipati, P. R. Panda, and S. R. Sarangi, “Enhancing network-on-chip performance by reusing trace buffers,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 4, pp. 922–935, Apr. 2020, doi: 10.1109/TCAD.2019.2907909.

[21] B. Talwar and B. Amrutur, “Traffic engineered NoC for streaming applications,” *Microprocessors and Microsystems*, vol. 37, no. 3, pp. 333–344, May 2013, doi: 10.1016/j.micpro.2013.02.003.

[22] Y. Chen, H. Cui, and Z. Wang, “An efficient reconfigurable encoder for the IEEE 1901 standard,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 30, no. 9, pp. 1368–1372, Sep. 2022, doi: 10.1109/TVLSI.2022.3177239.

[23] D. Alex, V. C. Gogineni, S. Mula, and S. Werner, “Novel VLSI architecture for fractional-order correntropy adaptive filtering algorithm,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 30, no. 7, pp. 893–904, Jul. 2022, doi: 10.1109/TVLSI.2022.3169010.

[24] F. Restuccia, A. Meza, R. Kastner, and J. Oberg, “A framework for design, verification, and management of SoC access control systems,” *IEEE Transactions on Computers*, vol. 72, no. 2, pp. 386–400, Feb. 2023, doi: 10.1109/TC.2022.3209923.

[25] A. P. D. Nath, K. Raj, S. Bhunia, and S. Ray, “SoCCom: automated synthesis of system-on-chip architectures,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 30, no. 4, pp. 449–462, Apr. 2022, doi: 10.1109/TVLSI.2022.3141326.

[26] M. A. Al-Shareeda and S. Manickam, “A systematic literature review on security of vehicular Ad-Hoc network (VANET) based on VEINS framework,” *IEEE Access*, vol. 11, pp. 46218–46228, 2023, doi: 10.1109/ACCESS.2023.3274774.

[27] S. S. R. Abidi and S. Manickam, “Leveraging XML-based electronic medical records to extract experiential clinical knowledge: An automated approach to generate cases for medical case-based reasoning systems,” *International Journal of Medical Informatics*, vol. 68, no. 1–3, pp. 187–203, Dec. 2002, doi: 10.1016/S1386-5056(02)00076-X.

BIOGRAPHIES OF AUTHORS

Trupti Patil completed her B.E. from VTU, Belagavi in the year 2011 and M.Tech. in embedded system from JNTU, Hyderabad in 2013. Currently doing research design from VTU (Belagavi) India on efficient NoC router designs. Published 3 papers in international and national journals, 2 papers in international and national conferences and published a book titled “microcontroller and microprocessor”. Her main research interests and activities include design of asynchronous NoC. She can be contacted at email: patiltrupti11@gmail.com.

Dr. Anuradha M. Sandi completed her B.E., M.Tech. and Ph.D. degrees from Gulbarga University, India in the year 1996, 2005 and 2014, respectively. She is currently working as an associate professor in GNDEC, Bidar. She has more than 21 years of experience in teaching field. Published more than 9 papers in international and national journals and 10 papers in international and national conferences from research. Her main research interests and activities include CMOS VLSI technology, embedded system design and network security. She is a member of the Life Member of Institute of Electronics and Telecommunication Engineering (LMIETE) Life Member of Indian Society for Technical Education (LMISTE). She can be contacted at email: anu29975@gmail.com.