International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Design and Performance Analysis of FDOR with Label Switching Buffered and Bufferless Noc for Intelligent Low Power Communication Systems

[1]Sujata S. B, [2]Anuradha M. Sandi

Submitted: 06/09/2023 **Revised**: 22/10/2023 **Accepted**: 07/11/2023

Abstract: The existing network systems used to communicate on-chip networks have buffers to share and store packets to avoid contention and for memory optimization. To address these two issues, the deflection-based bufferless routing algorithms have been proposed recently as an alternative design for the current system, but power and area utilizations are more. A better solution for optimizing buffer-less routing systems has been presented in this work, and it guarantees power and area optimization with high throughput. When comparing these algorithms to the realistic design, the buffer less gave a significantly reduced performance. The proposed FDOR and Label switching (LS) are two different network topologies for routing and switching, effectively controlling the packet and successfully delivery of a packet, this lead to a high packet delivery ratio. While implementing the buffer-less algorithm, it is to be significantly less complex, and a comparison is made on the frequency, area, performance, and power conservations to the buffered counterparts. The flexible direction order routing algorithm (FDOR) inherits the advantages like deadlock, freedom, and simplicity, which uses a logic-based implementation of flexible direction order routing. The Dynamic reconfiguration NoC and FDOR have been designed and synthesized in the Xilinx software tool and implemented on Artix-7 FGPA. The proposed NoC has an 8x8 design with 64 routers incorporated with power reduction techniques before transmitting and after receiving packets. Based on synthesis results obtained with a buffer, less routing can lead to a reduction in the area by 27%., the power consumption is reduced by 31%, and the router cycle time is also reduced by 11%. Finally, FDOR with LS is tested and validated for streaming data transmission on FPGA under higher throughput and injection rate.

Keywords: *FDOR, Dynamic reconfiguration NoC along with FDOR (Dr-FDOR-NoC) Label switching algorithm, Buffered & Bufferless NoC TED-based Power reduction.*

1. Introduction

The Multi-chip cores in [1], are now having a new era of communication between the cores to memory and core to cache in the form of On-Chip Communication; this is the backbone of the communication in the chip. A few networks are connected from each node to another, or from the slice of the cache or memory, or it might be a combination of them [2]. The increased number of nodes also helps to increase the number of nodes for communication; these two are proportional to each other. The resilience of the communication on the chip is the efficient routing, area, power, and performance; these properties should be well maintained to keep the communication well [3]. There are a few crucial assumptions that help the current on-chip communication is used and implemented, this is mentioned in [4]. The main idea was the packets should be buffered before the switching is done; this is coined as the scheduling or the process of the routing. The ultimate goal of the buffering is to increase the efficiency of the bandwidth at the time of congestion. Considering a scenario where two packets must reach a common destination, the first packet is routed to the goal. The second is buffered until the path is cleared for communication. Still [5], we have several choices when the buffers are used; they are additional power consumption, occupancy of the on-chip network area, around 75% of the TRIPS chip prototype, and the complexity of the design need to be reallocated for the management of the allocation and deallocation. Shortly days, bufferless algorithms are popular to overcome the disadvantages mentioned [6], in the buffered system [7].

The primitive idea of a bufferless algorithm is to remove the input and the output buffers in the routes of the routing algorithm. If the incoming packets have the same destination, then one packet is i) deflected or misrouted or sent to an undesirable route ii) dropped off the route and asked for retransmission; these are the two methods used instead of the buffer [8]. If the network is congestion-free, the real-time working applications work on the on-chip web [9]. Choosing bufferless will hinder the algorithm's traditional method of the oldest-first arbitration [10]. This will additionally cause the deflections in the route and the information in the header to be transmitted for the flit; this will increase the link traversals, for both can lead to increased energy consumption in the network. Few of the research papers also evaluate the excellent terms and the benefits of the back-of-the calculation of the envelope [11]. It is only possible to implement the three main

[1]*Research Scholar, Department of Electronics and Communication, GNDEC, Bidar, VTU Belgaum Karnataka, reddy20sujata@gmail.com*
[2]*Professor, Department of Electronics and Communication, GNDEC, Bidar Karnataka, anu29975@gmail.com*

aspects of the bufferless system by analyzing and implementing with real-time implementation. The FDOR (coined as the Flexible Dimension Order Routing) algorithm [12], is the best alternative for removing the deadlocks in routing packets. This will also reduce the complexities in the algorithm when compared to other algorithms [13]. The switching schemes in the FDOR will not include additional Virtual Channels (VCs). The deadlocks are accessible in the channel in the irregularly-shaped topologies for processing the modules in the network [14].

2. Research Method

The main goal of this paper is to evaluate how far the bufferless system is implementable, the benefits of the bufferless system, and the disadvantages of using them; a comparison of the algorithms and current and future implementation of the developed network are also part of this paper [15]. The area, frequency, and power performances are particularly calculated for the result and analysis of commissions [16]. The logic-level gate optimization must be analyzed and investigated to eliminate the buffers [17]. This may improve the solutions that can stop the challenges encountered in the bufferless routing algorithm. There are many implementations of the DRNoCs presented in the [18], and several discussions on the routing algorithm are mentioned to release the deadlocks of the system for accessible communication. The complexity increases when the area for the chip increases, the power consumption rises abruptly, and the routing algorithms' reconfiguration is used to route the packets [18].

Bufferless NoC: The reconfiguration of the DRNoC is also done on the system, which is based on the logic gate implementation of the FDOR algorithm, which is implemented in the 2-D torus [19]. This helps to automate the adaptability of the routing algorithm for the current and the obtained positioned topologies [20]. This helps improve the network's processing modules after the reconfiguration of the partial structure. DyAFNoC (for Dynamic Automatic FDOR-based NoC) is the name coined for the network architecture, this network is clearly described using VHDL, and the simulation is done using the DCS method (Dynamic Circuit Switching) [21], this is used for the partial reconfiguration of the system [22]. The routers used in the network use the switching technique called packet-switched networks; they communicate from one node to another in the form of packets and flit control (FC) which can control the transmission of a packet in multiple directions from one router to another without conflict as shown in Fig.1.

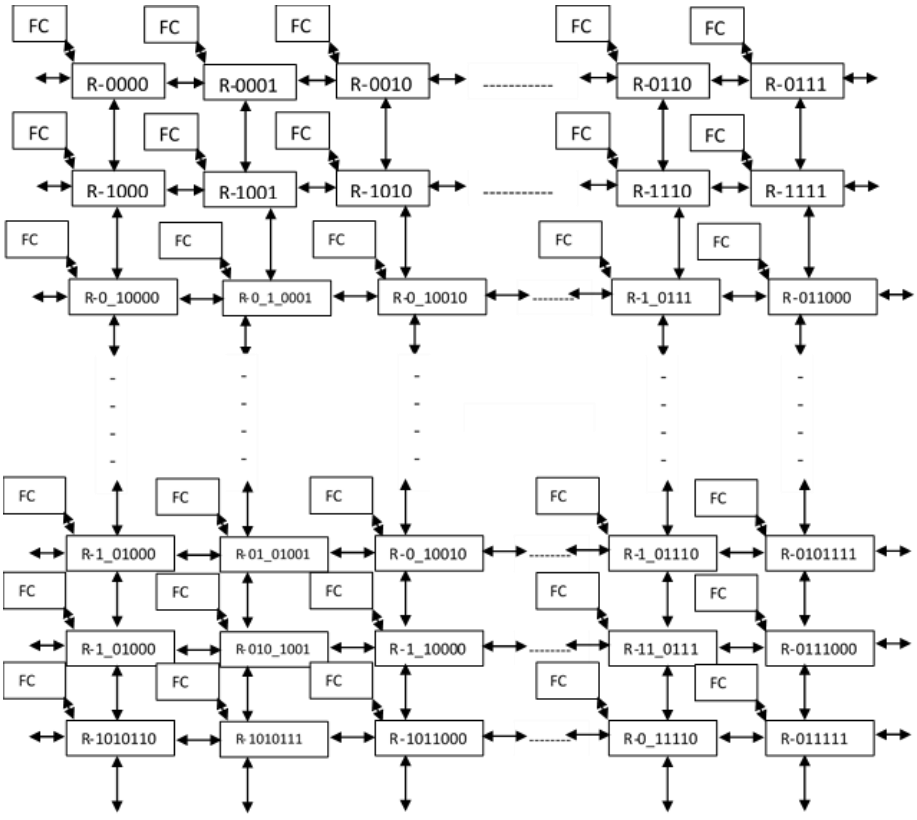

Fig. 1: General 8x8 Torus Topology example

To complete this operation, the router uses the buffering for the buffers for the flits whenever the multiple incoming flits need to reach the same destination [23], [24]. The proposed [25], NoC with torus topology for

Software-Defined Networking (SDN) applications focused on optimizing the chip area in a silicon chip, i.e., FPGA. The NOC with complexity management and associated issues in SoCs and NoCs using different topologies and routing algorithms and comparisons have been made regarding power, area, and delay optimizations. [26], presented the SMART NoC for multiple cluster formation to manage more extensive data transmission through a more significant number of source and destination nodes. This SMART NoC is concentrated on the minimization of contentions and collusions in bufferless, which is reconfigurable NoC routers. [27], proposed ASIC-based NoC using Time Division Multiplexing (TDM) techniques to manage the scheduling of packet delivery through bufferless NoC. The complete design is based on synchronous and dynamic allocations of port connectivity and bandwidth adjustment. In larger dimensions NoC, the significant concerns are faults detections, run time simulation, and reduction in contentions; these issues are addressed in [28]. In [29], is mainly focused on fault identifications and successful delivery of packets with minimal delay. This work uses bit shuffling techniques for fault detection and corrections and exploits in run time and title of fault position and accordingly changes the order of bits using the flit concept, which is usually Most Significant Bits (MSB) [29].

Aim: The main of this work is to design 3x3, 4x4, and 8x8 NoC with novel routing and switching algorithms for optimization parameters using torus topology for buffer and bufferless networks, as shown in Fig.1. These three designs are mainly concentrated on the optimization of power, increasing of a lifetime, low cost and bandwidth. Each router of Fig.1 is connected to flit control and labeled as router-id for easy monitoring of packet transmission among 64 routers.

Motivation: This work mainly focuses on three parameters: Scalability, Reuse of usual network technics, and Predictability. The scalability depends on performance, complexity, and design interconnect. The 8x8 NoC is mainly on interconnecting among 64 routers and how it affected performance in terms of hardware resources utilizations. The NOC can provide managers with dynamic allocation and adjustment of bandwidth and how increasing the count of utilizations will affect the complexity of the NoC chip in real time.

3. Proposed 4x4 AND 8x8 Dr-FDOR-Noc With High Througput and Low Latency

Based on the literature survey, it is concluded that power consumption management, optimization of area, delay, and increasing of throughput are significant concerns and challenges highlighted in the introduction sections, which are addressed in the proposed methodology section. So this research mainly concentrated on those areas, developed dynamic bandwidth allocation within routers, and automatically switched from one router to another. The realistic model of the DBBR algorithm (Deflection Based Bufferless routing) is proposed in the paper; the model is compared with other current networks and algorithms. Primarily the buffered algorithms are compared on the same platform of FPGA. The FPGA used is 65nm technology. Bufferless routing is an efficient implementation for torus-based on-chip networks. This resulted in a reduction of 38% in area and 30% in power consumption and the shown as the best-buffered router implementation in the 65nm for the FPGA design [36]. The area is reduced on the BLESS by 24%, and power is reduced by 18%.

3.1 Buffered Routing technique

As it is mentioned earlier, the on-chip and the off-chip networks always used the buffers for the input and the output ports, or sometimes for both cases, as shown in Fig.2. Whenever the router gets a flit, it is buffered into it, the buffer behaves as a FIFO with the corresponding ports of the input. The header is designed differently based on the destination in all the flits. The header holds the address of the output arbitrates for that of the output port. The round-robin method prioritizes the input over the others; this method is used in different cycles. To optimize the routers that are already existing in the system, there are several methods proposed. A simple routing algorithm is Wormhole routing [8]. This simplified packet performs computation only for the flit's header. As it is known, the head flit is made of the header information. The head flit acts as the guide for the rest of the flits in the channel; this is very similar to a worm; wherever the head moves, the rest part of the worm moves in the same direction. Other flits do not have any independence in choosing the output ports [26].

3.2 Bufferless system in routing system

The BLESS routing mechanism is used, particularly the FLIT-BLESS [30], method for the bufferless routing algorithm. The BLESS routing mechanism eliminates the input and output buffers, but the pipeline latches are not destroyed. This reduces the on-chip network's area, complexity, and power consumption. The flits are immediately routed toward the output port as soon as they arrive at the nodes. The misrouted flit that is undesirable in the output port is deflected in the misrouted unpleasant for the output port. This will always keep the distance between the destination and the flit. The primitive idea of BLESS is to change the path of the flit or to mislead the route of the flit; anyhow, the goal remains the same as the flit consists of an unchanged destination address. The flaw is that the path could be more optimal [28]. FLIT-BLESS algorithm performs the route for the independent flit and

is a simple method of route allocation. Header information is present in each flit that flits from the same input packets are routed to different output ports based on the routing availability.

3.3 Label switching topology for effective direction finding

All the topologies accept the buffered routing methods because only a few of the topologies accept the bufferless topologies; the network should be capable enough of routing every flit to the destination node immediately. The essential requirement is to have the number of outgoing links greater or equal to the number of incoming links. To implement the basic structure, the 2D torus topology is chosen. BLESS can also be implemented using the Taurus topology. There are several alternative implementations of the Taurus topologies. BLESS was never considered with the torus topology. Evaluation of Microarchitecture router: The Microarchitecture of the routers is defined in detail, using the buffered design, bufferless design, and the designed based on the deflection-based buffered prototype.

4. Microarchitecture Based on the Baseline Buffered Router

The latest baseline buffered router microarchitecture is depicted in Fig. 2. There are a few other parameters that are designed for easy configurable and to enable the additional features of the credit-based flow control of the architecture. The virtual channels and wormhole routing are the few additional features. The NOC manager mainly updates the bandwidth availability with each port and pipes setup when all ports are routed, and it is derived as per the equation (1).

$$Clock\ time_{conf} = Time_{routing} + T_n\ cycles - - - - - (1)$$

Where $T_n = S_n x deg^2\ cycles$, where S represents size of network which is routing table and deg is 2-D network size.

The module of Route Computation (RC): The route packet is calculated using this module. The routing method is wormhole routing; only the head flit is used to compute the destination address. As mentioned earlier, XY routing for n cube Network is the logic used. The network on the chip commonly uses XY routing.

4.1 Working of a buffer router

Four stages take the buffer to operate. These stages can further be reduced to minimize the critical path of the router. The four steps can be reduced to 2 or three, respectively. This phenomenon is a part of this prototype.

Idle State: The virtual channel that is at the initial stage is assumed to be in the Ideal state. This ideal state is otherwise termed cycle 0. If the input controller encounters a new flit of a new packet, the address of the destination and the virtual channel field are Decoded and extracted. The routing logic receives the destination address at the start of the first cycle, as shown in Fig. 2. The FIFA buffer saves them. Virtual channel field value of the flit at the same time. The input controller changes the GRS of the virtual channel, and it is changed from idle to routing.

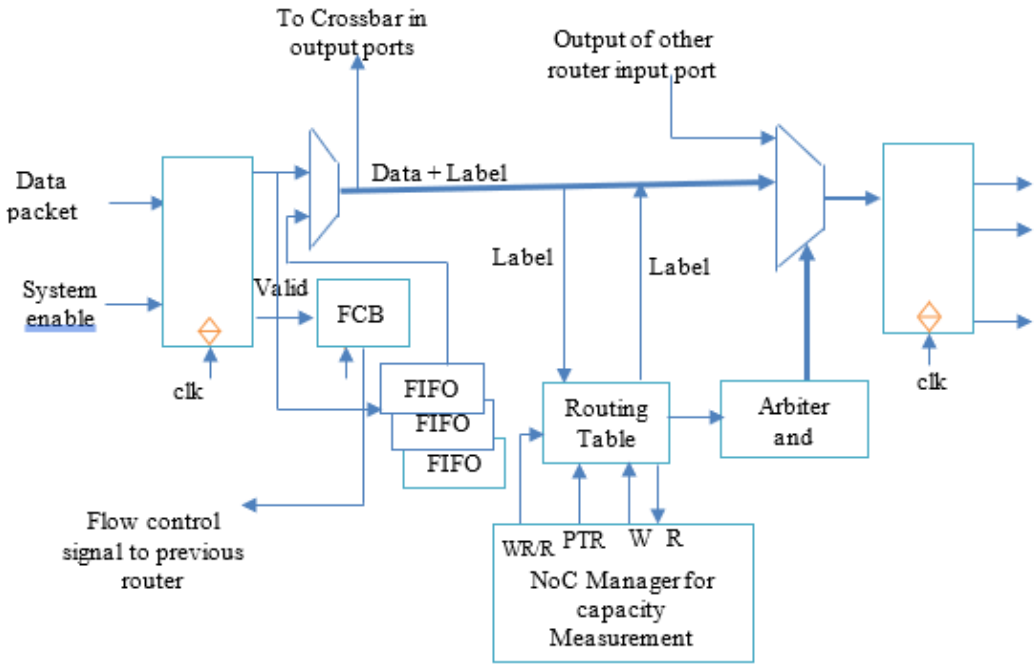

Fig. 2 Proposed Label based switched micro architecture router with single-cycle flit traversal and their internal micro blocks.

Route computation stage: The XY routing algorithm is used to decide which output port has been used as the destination for the head flit. This whole thing is done in cycle 1. At the end of cycle one, the destination output port will be ready to receive the packet.

Virtual-channel **allocation stage:** The phase from routing to virtual allocation is changed in the register at the beginning of cycle 2. The state is registered from RC to the root register. The VC allocator receives a virtual channel allocation request from the input controller at the same time as cycle 2. The proposal is sent using the packets of the physical output port number assigned by RC. The arbitration is done on the allocator in the next stage from the VC input controller.

4.2 The buffer less router micro architecture

This is also coined as BLESS. The FLIT-BLESS algorithm is used for the implementation of BLESS routers. There is an in-detail explanation in the [11]. The algorithm is designed in such a way that it prioritizes elder flits based on their age. The fits are sorted according to age; the ranking is allocated to the older first and the youngest later. Based on the order of priority, the output ports are assigned to the flit. Always the youngest flit gets the last output port allocation, and the oldest flit will get the output port allocated first. Based on the direction of the port project, the flit can handle arbiters by picking one approach over another. Considering this algorithm, the BLESS router's microarchitecture will be designed into three stages. This is depicted in Fig. 2.

The phenomena of route computation: This is a stage where each flit is processed in parallel with the RC module. It is processed to determine the shortest path that can be taken between the X or Y dimensions to reach its destination. This phenomenon will also give us a signal notifying the resource whether the flit can be injected or not to the shared network. The RC module now starts determining how many valid incoming flits are present. Based on the number of incoming flits, the signal is generated. The RC module will make sure either one of the four flits has reached its destination before generating the next signal

The priorities Arbiter using LUTs: An arbiter is the first job for the priority comparison sub-module. Ranking the incoming flits depending on the ages by a series of comparisons will decide the rank priority, as mentioned in Fig.2. The oldest flit will get the highest priority, and the youngest flit will be the least priority. The priority list is bubbled down if the input port has no valid flits during the cycle. The focus of the resource port will not remain the lowest always. The size of the age priority field will be determined based on the maximum time packets present in the network.

The Switch Allocations (SA): The method of switch allocation: Fig.3 shows the buffered NoC is the first Arbiter for BLESS architecture. An effective output port list and a priority list have been produced once the first stage of the process is completed. The Arbiter chooses X dimension as a priority to allocate the available output port. If it doesn't succeed in allocation, it will give the productive port present in the Y dimension. If the Arbiter fails to allocate both directions, the flit will be deflected and willed. Or it will choose the arbiter technique of fixed order allocation. In fixed order allocation, the ports will be allocated from North, South East, and West. Once the Arbiter completes the allocation of all five ports, a configuration table is generated for the behavior of crossbar switches in the next cycle. As shown in Fig.4, the oldest incoming flit is P0 which gets the output port allocated first in the complete cycle. Based on the priority calculation logic, the total delay of the orbiter is being decided; if the serial arbiter cells are considered, the following equation tells the delay in the arbiter.

DelayArbiter=DelayRank +DelayArbiterCellStage.

To calculate the delay in the priority rank, the delay is proportional to log 2 delays of the comparators. If the number of ports is multiplied by the uncertainty of each arbiter cell, then this has resulted in a delay in the port assignment. There is a linear proportionality between the number of ports and the uncertainty of each arbiter cell. In other words, this can be termed a delay of the output. The port assignment is very much proportional to the square of the number of incoming ports. If arbitration is assumed to be an atomic operation, it will affect the BLESS to have the lowest to clock frequency, however, less than one VC router. The critical path can be seriously reduced by using the oldest first arbitration.

5. Dyafnoc Architecture for High Throughput

In modern communications, the significant concerns are the successful delivery of packets without losses between source and destination nodes, low power, and low memory utilization. An advanced and high speed with a high throughput communication system is required to achieve them. Therefore, this proposed 8x8 NoC consists of hybrid techniques for routing, switching, and managing bandwidth for increasing throughput and speed. The proposed 4x4 and 8x8 NoC can handle streaming data transmission with more Packet Delivery Ratio (PDR) and automatic dynamic bandwidth allocation using NoC Manager, which is part of the design. Considering the HERMES network as a baseline, the DyAFNoC architecture was developed [9]. The structure of DyAFNoC architecture is very similar to a 2-D torus. It consists of 5 ports routers: north, South East, West, and local port. The ports operate in wormhole switching mode.

Each port is designed to be bidirectional to manage the wormhole switching mode. Each port consists of two registers that can store incoming flits of packets. A processing element (PE) connects each router to the local port. During the runtime of DyAFNoC Architecture, the processing modules (PM) will allocate and position over the network to form a different context. By keen observation, the basic configuration of this architecture will start with one PE per switch shown in Fig.3.

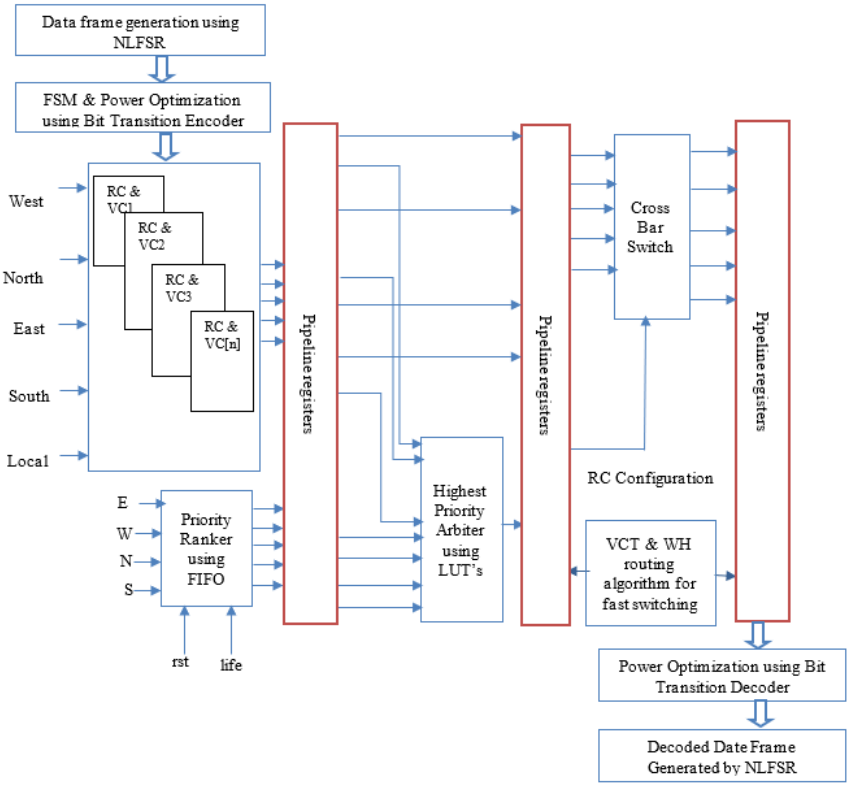

Fig.3: Proposed micro architecture of buffered router

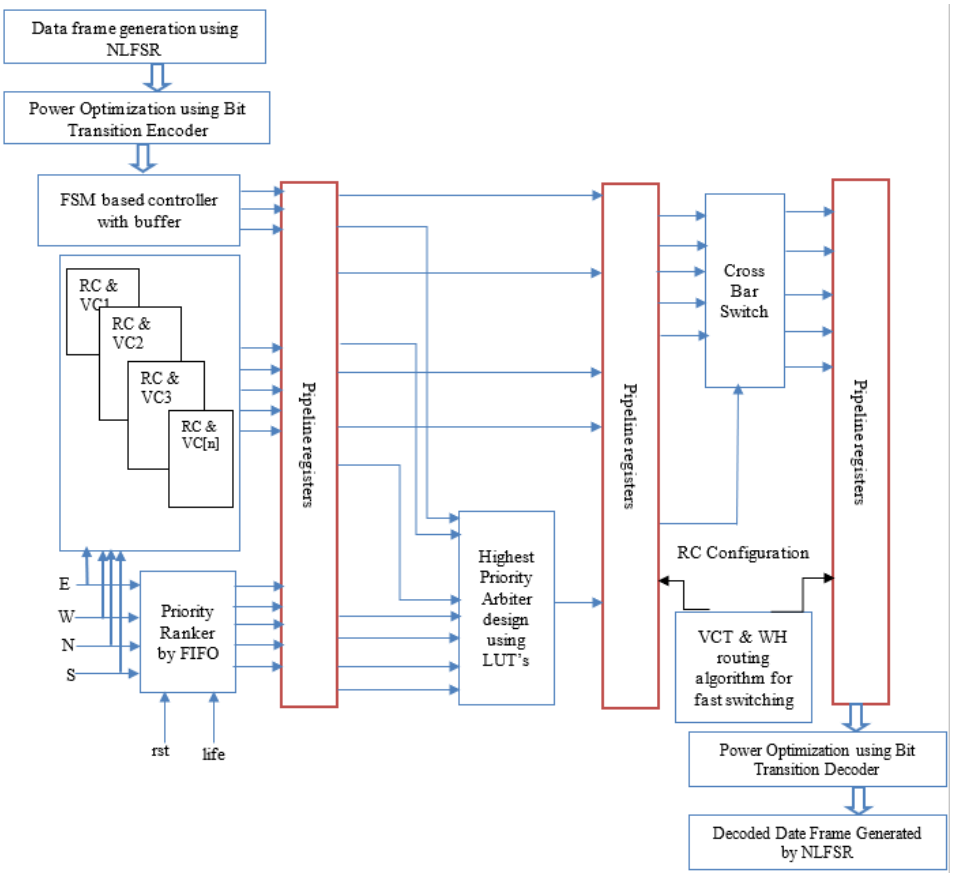

Fig.4: Proposed microarchitecture of a Bufferless with power optimization technique.

The partial dynamic reconfiguration in a dark green arrow in Fig.5(a) is data transmission between the source and destination routers. Its routing table is shown in Fig.5(b). The partial bit stream is now sent to the partial reconfiguration area of FPGA devices. The Red circle and the red arrow indicate the task and the processing of the task, respectively. This helps the FDOR configuration to be applied on the torus. These processes are completed using two control systems that are MRCS (Mesh Reconfiguration Control System) and PDRCS (Partially Dynamic Reconfiguration Control System).

5.1 The MRCS for NoC

The reconfiguration of PDRCS is done when the start configuration bit is equal to 1. Now the MRCS will receive a configuration map bit matrix, also called a configuration map of the network. This map mainly identifies the DOR routing (YX or XY). This is an automatically detected DOR algorithm. The network drain operation is started in the 2-D torus controller when all these calculations are completed. Before processing the routing algorithm each router before processing the routing algorithm, each router sends a positive signal to MRCS. This is considered an update to the algorithm stating the switch condition as accessible to the buffer and indicating the router is empty. When there are free routers, Merckx updates the current configuration matrix set up to apply the configuration signal to configure all the rotors present in the network. When the reconfiguration is used, the network should be free from the deadlock. This functioning is exploited by a typically static reconfiguration approach (TSR). This is to avoid the presence of two different algorithms that could lead to an extra dependency and, consequently, to a deadlock.

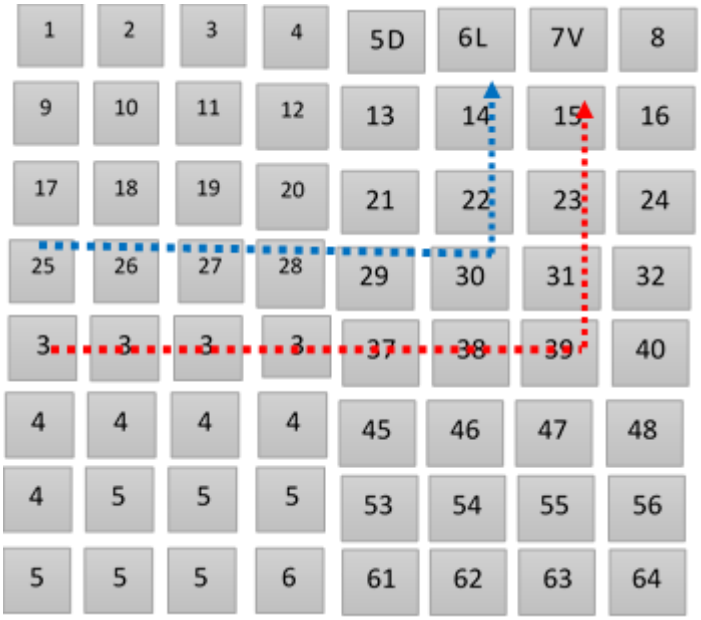

Fig.5 (a) Time diagram of the DyAFNoC and FDOR architectures during data packet transmission

6.2 Partial Dynamic Reconfiguration Control System (PDRCS)

The PDRCS is fabricated using a general arbiter and finite state machines. These finite-state machines are used to control the sequence in the DyAFNoC operation. This is illustrated in Fig.4. The PM and PE send the corresponding signals. Once the task is done successfully, now the MRCS will obtain the 2D mesh configuration map from the sender. These blogs are depicted in Fig.8. Dynamic circuit switching is the technique used to emulate and isolate the partially dynamic reconfigurable bit streams sent for simulation [6]. Now the isolation switches are positioned so that they cross the boundaries of reconfigurable regions (RRs). Fig.8 shows how the PMs are configured and placed in the network. The reconfigurable fabric will be simulated whether the presence or absence of PEs or PMs. The other word for the isolation switch controller is RR. This RR will receive the request from the arbiter to send the signal for the isolation switches. The isolation switches will work to simulate the partial dynamic reconfiguration process. The time of each bitstream configuration is emulated using programmable counters. This is controlled according to the FPGA logical resources. Fig.7 will also explain the five steps involved in the partial dynamic reconfiguration process. Every node element located as the router for data will be working with injection logic; this logic is the same as the working principle of the available cores of the processor and cache with the different traffic patterns in the given network. The current scenario has been analyzed with six various hybrid combinations of the traffic pattern, as

shown in table I. The proper utilization of the routing process statistically to reach the destination address either by Nearest Neighbor (NN) or Uniform Random (UR). To route, the packet to the destination in this research work, the Linear Feedback Shift Register (LFSR) technique is utilized in a Pseudo random pattern of packet delivery. The various ways of traffic are demonstrated in a suitable FPGA implemented on the NoC with packet injections of 100 million. Table I below summarizes the different traffic patterns that can be used for the analysis with the designed NoC on FPGA.

Ei	ni	nj	ui	vj	Rij
25_26	1	6	10	0	{0..64}
27_28	2	8	10	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..64}
39_31	8	1	0	10	{1..64}
31_23	9	3	0	10	{1..64}
23_15	10	5	0	10	{1..64}

Ei	ni	nj	ui	vj	Rij
25_26	1	6	10	0	{0..64}
10_11	2	1	10	0	{0..64}
17_18	3	8	0	10	{1..64}
18_19	4	9	0	10	{1..64}
20_21	5	1	0	10	{1..64}
21_22	6	2	0	10	{1..64}
22_14	7	4	0	10	{0..64}
14_6	8	6	0	10	{1..64}
45_46	9	6	10	0	{0..64}
55_56	10	6	10	0	{0..64}

Fig. 5(b) LS-DyAFNoC and FDOR-based NoC state table's routes from 33 to 15 and 17 to 6 have been established

7.2. Power Estimation /Area / Frequency of Proposed NoC

The power consumption and its optimization in NoC is a primary challenging task and will degrade the performance level. In this work, as shown in Fig.3 and Fig.4, the Bit Transition Encoder technique is applied before the transmission of a packet to the source router and after receiving the packet at the destination router for power optimization. In any on-chip memory or networks, power consumption depends on number transitions such as bit 1 to bit 0 (formally known as Type 1) or bit 0 to bit 1 (officially known as Type 2); there is no bit transition if both bits are same like bit 0 to bit 0 (officially known as Type 3) or bit 1 to bit 1(officially known as Type 4). The power reduction technique will work only on Type 1 and Type 2. The power optimization purely works based on the number of transitions in packet data; if there are more transition bits, then encoding techniques are minimized before sending the packet to the next router. The generalized logical expression for encoding is given in Eq.(2) and Eq.(3)

$$Encoded_i = data_i \oplus FI \; for \; all \; odd \; number \; in \; i - - - - - (2)$$

$$Encoded_i = data_i \oplus FI \oplus HI \; for \; all \; even \; number \; in \; i - - - - - (3)$$

Where FI is full invert, it can be either 1 or 0, and HI is half inverted, its bit is the same as FI, is the present bit in the given packet, and is the last bit in the given packet; between these two bits, the XOR operations are performed to reduce number transitions, for example, let consider the number of bits in the packet is 16 bits (let say: 1010101010101010), the number of transitions is 15. After performing the Bit Transition Encoder on the packet through XOR operation, encoded bits are 1111111111111111, as shown in Fig.5(c), the number of transitions in encoded bits is 0. Therefore the number of transitions is reduced from 15 to 0.

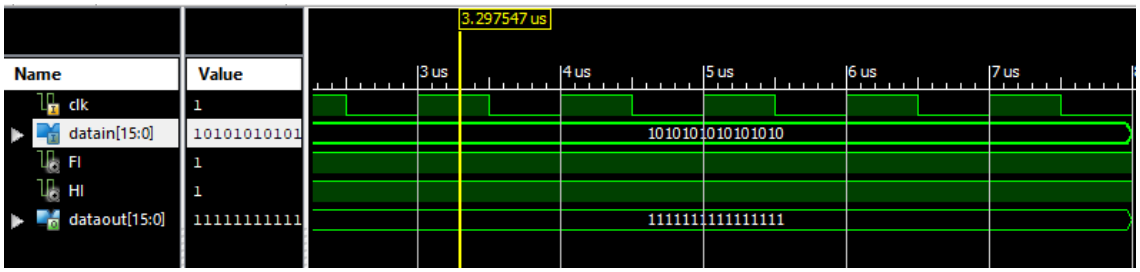

Fig.5(c) Simulated results of power reduction through Bit Transition Encoder technique, here input is 16bits (1010101010101010) and output is 16bits (1111111111111111).

The encoded packet is transmitted from the source router to the destination router, at the destination router; after receiving the packet before decoding, the Bit Transition decoder is applied to decode the original packet. The generalized logical expression for decoding is given in Eq.(3) and Eq.(4).

$$Decode_i = rec_{data_i} \oplus FI \quad for\ odd\ bits\ of\ i - - - - - -(3)$$

$$Decode_i = rec_{data_i} \oplus FI \oplus HI \quad for\ even\ bits\ of\ i - - - - - - -(4)$$

After applying Eq.(3 & 4), the simulated results are shown in Fig.5(d), the decoded packet bits are same as packet bits which is transmitted at source node.

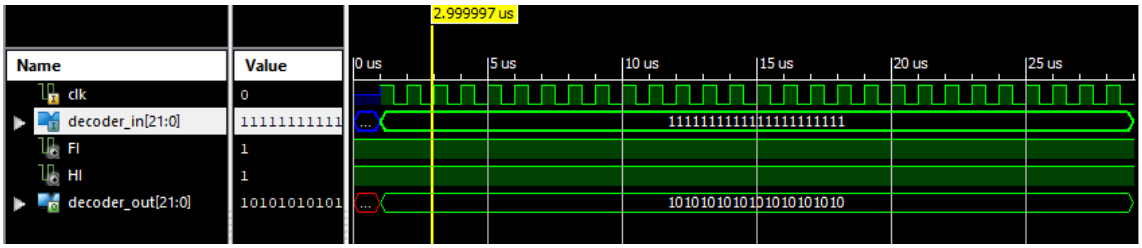

Fig.5(d). Simulated results of power reduction through Bit Transition decoder technique, here input is 16bits (1111111111111111) and output is 16bits (1010101010101010).

6. Results and Discussion

With the obtained results, it can be commented that the significant overhead of the reconfiguration of a partially dynamic system is the network drained even before the proposed FDOR configuration, which is applied to node members as routers. The time consumed for the draining time is dependent on the traffic rate in the network when the request for the draining is treated to the desired system. FDOR algorithm calculations are designed to operate with a single clock cycle as it is a purely gated logic design. It is partially verified with the dynamic reconfiguration, which will not affect DyAFNoC performance as data packets are transmitted to the network, and the different processing modules are simulated are shown in Fig.6. FDOR algorithm is also developed with an advanced simulator, Noxim [10], to evaluate the data transmission for every uncommon topology for torus topology of 8x8 architecture for the representation of sub-regions of FDOR with two-dimensional DOR algorithm as represented in Fig.4.

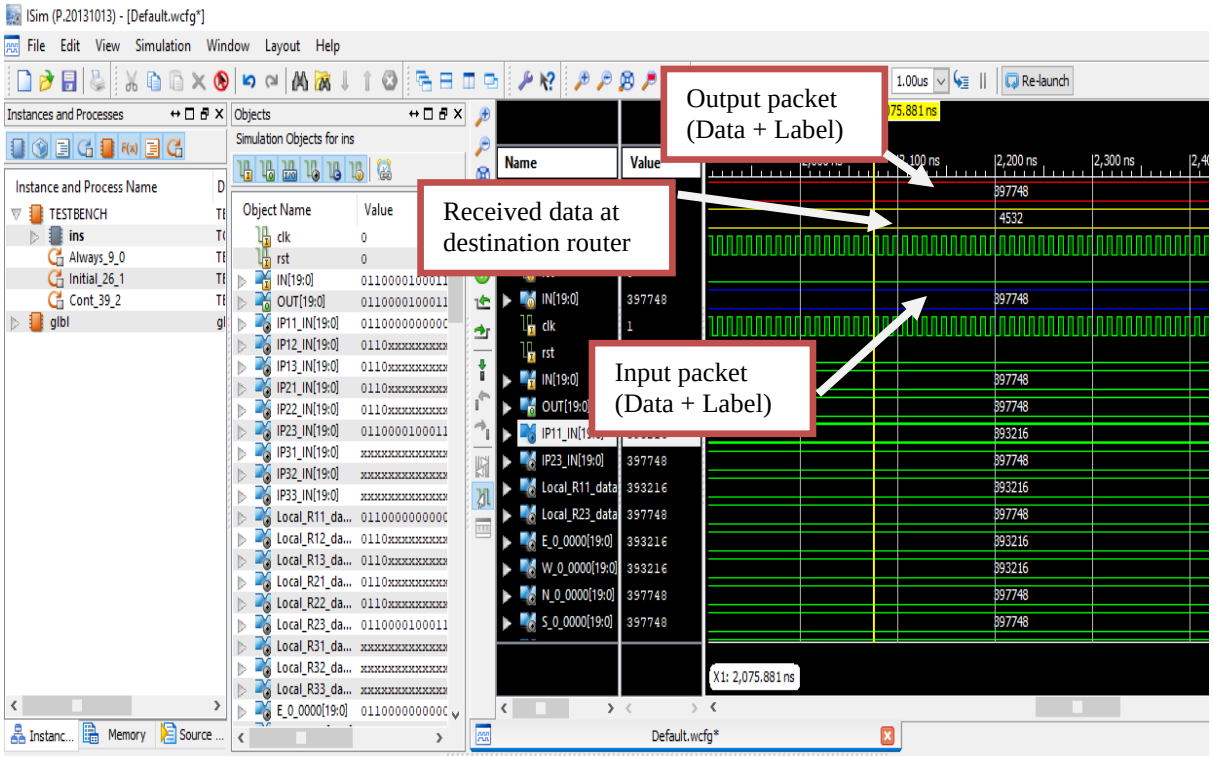

Fig.6. Simulated results of proposed 3x3 NoC and first router is source and 6th router is destination router and input data is 4532.

The obtained results by simulations with different size of torus-type topology for the router architecture is tabulated in Table II, which is composed of 11,640 slices with the usage of the FPGA belonging to Artix-7 family with the square configuration 3x3, 4x4, and 8x8, in which each slice in the FPGA consists of 8 flip-flops and 4 LUTs. Fig.7 shows the comparative analysis of the DyAFNoC blocks and each parameter expressed in terms of the percentage as the scope for future work; it is suggested to put up concave topologies; therefore, a DyAFNoC structure would be only a mini-torus, part of a larger hierarchical network structure. The optimization of slice FF's between buffered and bufferless is shown in Fig.7(a), and their LUTs are shown in Fig.7(b); the utilized slice register is shown in Fig.7(c), and the maximum frequency used is shown in Fig.7(d).

TABLE II. SYNTHESIS RESULTS FOR ARTIX-7 Logic Utilization 3x3, 4x4, 8x8

Parameter	Buffered NoC		Bufferless NoC	
	Existing	Proposed	Existing	Proposed
Slice FF's	4210	3720	4093	2901
LUT's	3610	1830	3582	951
Slice Registers	3571	2491	2980	1609
Max. Freq. (MHz)	142.6	201.46	124.1	138.5

8.1 FPGA Area Consumption

The slice is the basic configurable logic block in an FPGA; it can be programmed for any of the quad input Boolean functions with two single-bit registers, which are configured with the programming of the latches and flip-flops and dual dedicated user-controlled logic for every logical device with dedicated logics on the FPGA hardware, each area of consumption on the FPGA is the number of slices it is utilized for the execution of the particular logic. Table III compares the various regions among the five different routers' architecture. Fig.8 gives the area covered under each router with unique router components, including crossbar, SA, VA, and input controller. With Fig.9 it can

be drawn several observations, which can be summarized as follows:

- BLESS portrays remarkably lesser area than any other kind of router, a 1-VC wormhole router that is embedded with a 4-flit FIFO in every incoming port has made utilized 55.9% more area than BLESS which has no input buffers. Hence, it can be concluded as the significant area efficient router design with BLESS with a very limited buffer.
- The area consumption is relatively constant across designs of BLESS and BLESS with buffers also will not require VC allocation strategy as they eradicate VCs and hence larger area-efficient than combinational VC designs, which demand substantial area to perform allocation of multiple VCs, since the crossbar is mainly very significant in BLESS and buffered designs.

Frequency Analysis: As shown in Table. IV depicts the maximum allowable clock frequency and delays in critical path delay for different routers. It can be listed that the clock frequency is higher by 32% for BLESSrouter'sthanbuffered1-VCrouter. In router design, as the increased VCs will impact the buffer, the critical path of the buffered router increases due to the increased complexity of switch allocation between multiple virtual channels. Due to additional processing on must Schedule bits, BLESS with buffers has an 8% lower clock frequency than a single virtual channel router. The higher advantage rate can even be observed in the BLESS router than the 2-VC buffered router, which provides 2.41 times higher frequency. As the multiple Virtual channels are of more excellent order in BLESS, it results in higher complexity than BLESS as it has incorporated the pair of five sets of I/O ports.

Network Analysis: As shown in Table. V is observed with the condition of area usage of FPGA for the 4x4 and 3x3 torus router. The BLESS network has used 22% slices for 3x3, 6% less for quad input LUTs, and 55% cheaper flip flops usage with highly efficient in terms of area with single VC. Similarly, the 4x4 network for the BLESS network is observed to have 7% lessor quad input LUTs, 55% lessor flip flops, and 24% reduced usage of the slices than the single VC buffered network. The BLESS kind of router design will experience higher order of logic power when compared to a single VC router for the fact of the complexity of new arbitration at the recent place to satisfy the kind of priority for the incoming flits, as it will impact the amount of signal power consumption as it is not employed with the VC allocation for the buffer state organization. It can be concluded that BLESS router architecture leads to considerable network power savings, ranging from 72% to 18% upon networks with buffers with increased arbitration complexity in the router. The power saving rate is higher in BLESS network compared to the dual VC network, which has achieved 72% from 32%. Fig.11 represents the total consumption of network power for various torus networks operating at 25MHz for injection rates of 10 and 5%. In the architecture of 3x3, the BLESS structure has saved 25% and 23% energy for the injection rate of 5% and 10%, respectively, for the single VC network. Similarly, for a 4x4 network, the considered BLESS has gained the advantage in terms of the total power of about 19% and 18% for injection rates of 10% and 5%, respectively, as compared with the single VC network.

8.3. FPGA Implementation Results

For the experimental analysis and to make the tradeoff among the prescribed architectures of the routers with its FPGA design, the same design is implemented with the router's RTL Verilog design and presented. The obtained results with the utilization of the standard cell library of 65nm TSMC with Synopsys compiler in an optimized mode for the Verilog design in terms of area, frequency, and power, as tabulated in table IV. The values obtained for the FPGA results seem different, but the required linear relationships among the routers remain the same. With the comparative results, it is still noted that the BLESS architecture is designed efficiently with 28% of power saving and 41% area optimization, as shown in Fig.11 when compared to the single virtual channel router with buffer; on the other hand, BLESS is 11% accurate in the speed of operation are shown in Fig.10.

Table V: 65nmASICresults for all the routers

Router	Area(μm^2)		Power (mW)		Max Frequency (MHz)		Memory Usage (MB)	
	Existing [12,21]	Proposed	Existing [9,31]	Proposed	Existing [41]	Proposed	Existing [14]	Proposed
BLESS	12353	11030	2.14	1.2	214.6	260.6	4901	4884
BLESS (Buf)	16854	12150	2.74	2.14	204.5	206.8	5381	5019
1-VC	20139	12129	3.03	2.14	197.2	201	3997	3910

2-VC	36152	27122	5.12	4.52	100.4	170.4	4902	4813
4-VC	79900	62050	10.11	9.94	50.3	142.3	5891	5718

7. Conclusion

In the present research article, the performance of the bufferless router has proven the guaranteed performance with the reductions in the total power consumption and complexity with its network area, which is employed on the existing networks of integrated chips. It has attempted to find its experimental results on par with the real-time applications with its comparative results with buffered and bufferless architecture type of networks demonstrated on an FPGA board. With the detailed results based on the various conditions of the analysis with its network topologies with variable size of the network and multiple types of traffic conditions, it is noted that the enhancement in the ease of user-friendly usage when compared with the other existing techniques for the design of the architecture. In general, it can be concluded that bufferless routing has majorly impacted a few design terms, such as the area of consumption, overall power consumption, and the clock speed of the torus topologies for real-time applications. The significant advantages which have boosted the applications of bufferless routing applications are the path diversity in the interconnections will enhance for the better topologies and, in continuation, better response to the higher order frequency as it results in the higher order of virtual channels which turn resulted in the minimized power consumption and provided better efficiency in communication.

References

[1] C. Fallin et al., MinBD: minimally-buffered deflection routing for energy-efficient interconnect, in NOCS-4, Copenhagen, 2012.

[2] Mandal S.K., Krishnakumar A., Ogres U.Y. (2021) Energy-Efficient Networks-on-Chip Architectures: Design and Run-Time Optimization. In: Mishra P., Charles S. (eds) Network-on-Chip Security and Privacy. Springer, Cham. https://doi.org/10.1007/978-3-030-69131-8_3.

[3] Tariq, U.U., Ali, H., Liu, L. *et al.* Energy-efficient scheduling of streaming applications in VFI-NoC-HMPSoC based edge devices. *J Ambient Intell Human Comput* **12,** 9991–10007 (2021). https://doi.org/10.1007/s12652-020-02749-7.

[4] Subodh Charles, Yangdi Lyu, and Prabhat Mishra. 2019. Real-time detection and localization of DoS attacks in NoC-based SoCs. In *Design, Automation & Test in Europe Conference & Exhibition*. IEEE, 1160—1165.

[5] Trupti Patil, Anuradha Sandi, Design and implementation of asynchronous NOC architecture with the buffer-less router, Materials Today: Proceedings, Volume 49, Part 3,2022, Pages 756-763, ISSN 2214-7853,https://doi.org/10.1016/j.matpr.2021.05.282.

[6] S.B., Sujata. and M. Sandi, A. (2021), "Design and analysis of buffer and bufferless routing based NoC for high throughput and low latency communication on FPGA", International Journal of Pervasive Computing and Communications, Vol. ahead-of-print No. ahead-of-print. https://doi.org/10.1108/IJPCC-05-2021-0115

[7] Yu Cai. et.al, "Comparative Evaluation of FPGA and ASIC Implementations of Bufferless and Buffered Routing Algorithms for On-Chip Networks ", 16th Int'l Symposium on Quality Electronic Design, 978-1-4799-7581-5/15,2015 IEEE.

[8] MONIKA KATTA.et.al, "SB-Router: A Swapped Buffer Activated Low Latency Network-on-Chip Router", September 20, VOLUME 9, 2021, 2021, DOI 10.1109/ACCESS.2021.3111294

[9] Juan Fang, Sitong Liu, Shijian Liu, Yanjin Cheng, Lu Yu, "Hybrid Network-on-Chip: An Application-Aware Framework for Big Data", *Complexity*, vol. 2018, Article ID 1040869, 11 pages, 2018. https://doi.org/10.1155/2018/1040869

[10] Basavaraj Talwar. et.al "Traffic engineered NoC for streaming applications", 0141-9331, 2013 Elsevier B.V. http://dx.doi.org/10.1016/j.micpro.2013.02.003, Microprocessors and Microsystems

[11] Karthikeyan, A., Kumar, P.S. GALS implementation of randomly prioritized buffer-less routing architecture for 3D NoC. *Cluster Comput* **21,** 177–187 (2018). https://doi.org/10.1007/s10586-017-0979

[12] TruptiPatil.et.al, "Design and implementation of asynchronous NOC architecture with buffer-less router", doi.org/10.1016/j.matpr.2021.05.282, Materials Today: Proceedings

[13] Karthikeyan, A., Kumar, P.S. GALS implementation of randomly prioritized buffer-less

routing architecture for 3D NoC. *Cluster Comput* **21,** 177–187 (2018). https://doi.org/10.1007/s10586-017-0979-0

[14] C. Effiong, G. Sassatelli, and A. Gamatie, "Scalable and Power-Efficient Implementation of an Asynchronous Router with Buffer Sharing," 2017 Euromicro Conference on Digital System Design (DSD), 2017, pp. 171-178, DOI: 10.1109/DSD.2017.55

[15] G. Nychis et al., On-Chip Networks from a Networking Perspective: Congestion and Scalability in Many-core Interconnects. SIGCOMM, 2012.

[16] C. Bobda and A. Ahmadinia, "Dynamic interconnection of reconfigurable modules on reconfigurable devices," Design Test of Computers, IEEE, vol. 22, no. 5, pp. 443–451, Sept 2005.

[17] E. Wachter and F. Moraes, "Mazenoc: Novel approach for fault-tolerant NoC routing," in SOC Conference (SOCC), 2012 IEEE International, Sept 2012, pp. 364–369.

[18] A Strano, D. Bertozzi, F. Trivino, J. Sanchez, F. Alfaro, and J. Flich, "Osr-lite: Fast and deadlock-free NoC reconfiguration framework," in Embedded Computer Systems (SAMOS), 2012 International Conference on, July 2012, pp. 86–95.

[19] Gomez-Rodriguez, J.R.; Sandoval-Arechiga, R.; Ibarra-Delgado, S.; RodriguezAbdala, V.I.; Vazquez-Avila, J.L.; Parra-Michel, R. A Survey of Software-Defined Networks-on-Chip: Motivations, Challenges and Opportunities. Micromachines 2021, 12, 183. https://doi.org/10.3390/ mi12020183.

[20] H. Chen, P. Chen, J. Zhou, L. H. K. Duong, and W. Liu, "ArSMART: An Improved SMART NoC Design Supporting Arbitrary-Turn Transmission," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, DOI: 10.1109/TCAD.2021.3091961

[21] V. Venkataramani, B. Bodin, A. K. Mohite, T. Mitra and L. -S. Peh, "ASCENT: Communication Scheduling for SDF on Bufferless Software-defined NoC," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, DOI: 10.1109/TCAD.2021.3128445.

[22] R. Mercier, C. Killian, A. Kritikakou, Y. Helen and D. Chillet, "BiSuT: A NoC-Based Bit-Shuffling Technique for Multiple Permanent Faults Mitigation," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2021 DOI: 10.1109/TCAD.2021.3101406.

[23] A. Das, A. Kumar, J. Jose, and M. Palesi, "Opportunistic Caching in NoC: Exploring Ways to Reduce Miss Penalty," in IEEE Transactions on Computers, vol. 70, no. 6, pp. 892-905, 1 June 2021, DOI: 10.1109/TC.2021.3069968.

[24] B. Bhowmik, P. Hazarika, P. Kale, and S. Jain, "AI Technology for NoC Performance Evaluation," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 68, no. 12, pp. 3483-3487, Dec. 2021, DOI: 10.1109/TCSII.2021.3124297.

[25] Sivaganesan, D. "Improvisation of a mesh network with wideband code division multiple access." Journal: IRO Journal on Sustainable Wireless Systems 3 (2019): 198-205.

[26] Kiran, W. S., S. Smys, and V. Bindhu. "Clustering of WSN Based on PSO with Fault Tolerance and Efficient Multidirectional Routing." Wireless Personal Communications 121, no. 1 (2021): 31-47.

[27] Ashokkumar, N., P. Nagarajan, and P. Venkatramana. "3D (dimensional)—Wired and wireless network-on-chip (NoC)." In Inventive Communication and Computational Technologies, pp. 113-119. Springer, Singapore, 2020.

[28] Gangula, R. ., Vutukuru, M. M. ., & Kumar M., R. . (2023). Network Intrusion Detection Method Using Stacked BILSTM Elastic Regression Classifier with Aquila Optimizer Algorithm for Internet of Things (IoT). International Journal on Recent and Innovation Trends in Computing and Communication, 11(2s), 118–131. https://doi.org/10.17762/ijritcc.v11i2s.6035

[29] Prof. Madhuri Zambre. (2016). Analysis and Modeling of Physical Stratum for Power Line Communication. International Journal of New Practices in Management and Engineering, 5(01), 08 - 13. Retrieved from http://ijnpme.org/index.php/IJNPME/article/view/42

[30] Nagendram, S., Singh, A., Harish Babu, G., Joshi, R., Pande, S.D., Ahammad, S.K.H., Dhabliya, D., Bisht, A. Stochastic gradient descent optimisation for convolutional neural network for medical image segmentation (2023) Open Life Sciences, 18 (1), art. no. 20220665, .