



Buffer Association of Network on Chip (NoC) Using Simulated Network

Manjunath Managuli¹, Kanmani Pappa Chandramohan², Maheswari Marimuthu³, Kumutha Duraisamy⁴,
Surendran Rajendran^{5*}

¹ Department of ECE, KLS Gogte Institute of Technology, Belagavi 590008, India

² Department of Electronics and Communication Engineering, Vel Tech Rangarajan Dr Sagunthala R&D Institute of Science and Technology, Chennai 600062, India

³ Department of Computer Science and Engineering, Panimalar Engineering College, Chennai 600123, India

⁴ Department of ECE, Jeppiaar Institute of Technology, Kunnam 631604, India

⁵ Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai 602105, India

Corresponding Author Email: surendran.phd.it@gmail.com

Copyright: ©2024 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/isi.290513>

ABSTRACT

Received: 12 December 2023

Revised: 6 June 2024

Accepted: 1 August 2024

Available online: 24 October 2024

Keywords:

Dynamic Allocation of Multi-Queues (DAMQs), Efficient Dynamic Virtual Channel (EDVC), First in First Out (FIFO), Network on Chip (NoC), Verilog HDL coding

In the growing communication technology, it's necessary to have an appropriate structure of Network on Chip (NoC) system for the efficiency of communication to improve the data flow and 96.75% packet usage system. This can be improved by adding the Dynamic Allocation of Multi-Queues (DAMQs), Efficient Dynamic Virtual Channel (EDVC), and Conventional Efficient Dynamic Channel (CDVC) in the system; where DAMQs are nothing but dynamically allocated multi queues. Where the virtual channels face difficulty in storing the data and it becomes ideal to eliminate this condition 96.75% of buffers are introduced. To add the 96.75% number using DAMQs, then the 96.75% efficient communication results are achieved. This avoids the traffic signal helps in maintaining the 5*5 spacing between ports and helps to resolve the NoC-related issues such as deadlock conditions are hold and wait, no pre-emption, and mutual exclusive parameters. EDVC refers to the EDVC, where the communication efficiency is excellent at 96.75%, but the processing has 4 stages. The functional simulation is analyzed with results and the performance of FPGA design is compared with tabulation.

1. INTRODUCTION

The efficient communication helps based on a virtual channel system which is integrated circuits between intellectual property on a chip area they have several clock domains. Channel buffer for Network on Chip helps to improve crossbar interconnections [1]. It also provides scalability of SOCs; it provides power efficiency of the complex system on chips for the other design methods. Hence the connection of the NoC signals was divided into plenty of signals; hence the high-level parallel execution is done since all are operated simultaneously of data packet NoC helps to achieve maximum throughput and scalability with earlier communication architectures. The art is sequenced by using an appropriate algorithm using the stronger NoC computations.

Network on Chip (NoC) has become a critical component in modern multiprocessor systems, significantly influencing their performance and energy efficiency. As technology scales down and computational demands increase, the efficiency of on-chip communication becomes increasingly crucial. The need for high-performance and energy-efficient NoCs arises from the growing complexity and heterogeneity of applications running on these systems, which demand high bandwidth and low latency communication.

NoC compared with buses and instructions more advantages for example such as providing data the specific instructions. Using the generic sort that is proposed to get proper efficiency for scaling and parallel pipeline mechanism [2]. Buses are designed across the area such as application convergences.

(1) Consequences of the silicon processor evolutions between generations.

(2) To design the RTL schematic manual layout for adjusting the bus architecture to the design flow. And coming to the NoC the system on chip reduces the price used for manufacturing [3].

(3) The steps of execution help in increasing performance.

(4) It completes the tasks within a shorter period.

(5) It also reduces the risk management.

They have layered scalable architecture, flexible to user-defined instructions and network topology. NoC helps with point-to-point connections of locally synchronized and globally asynchronous IP blocks [4]. NoC layers as the transaction layers transport packets physical wires clocks which are interconnected to the internet services. They provide transaction services to the IP blocks the network interfaces card to the LAN WAN helps us for the conveying the information from the source to the destination in Figure 1.

It has master NIU and slave NIU, it has a packet that is flexible, and it is stretchable for the physical layer. For compatibility purposes the bus protocols are used then it accumulates the interactions of the IP blocks. NoC uses the typical packets for the application requirements [5-7].

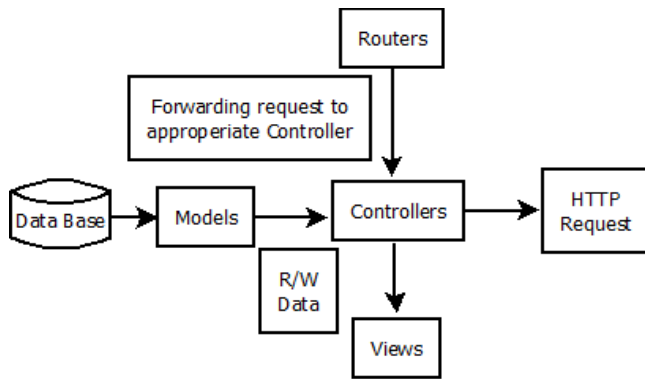


Figure 1. General architecture of router

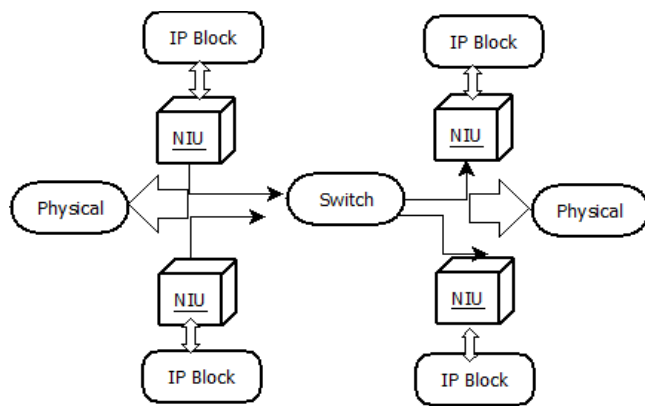


Figure 2. NoC layers for mapping

The physical layers have packets that are defined by the physical transmission over the interface much like Ethernet 10mb/s. They have independent hardening of the cores for the design practices in Figure 2. The advantages recognized by NoC system are used for the maximum operations through the wires and gates. It helps to regain the higher throughput by the suitable layers for allocating several logical paths. It simply scales the operating frequencies, switches, and links with a fabric topology [3, 8, 9]. The mark for NoC system is increased by minimizing the traffic with time constraints architecture, allowing multiple flits for the transmission by giving the higher priority. The mandatory rule is applied to support the timing convergence in the physical layer, which approaches the channel for the synchronous design style. It helps to approach NoC by the fabric behavior providing a special link between the networks to share the data independently or disturbing the transport layers, this helps to time convergence.

By exploring advanced techniques such as DAMQs and EDVC, this study aims to contribute to the development of more robust, efficient, and scalable NoC architectures. Improving these aspects will not only advance the state of the art in NoC design but also pave the way for more powerful and energy-efficient multi-core systems, ultimately benefiting a wide range of applications from consumer electronics to high-performance computing.

1.1 Latency

1.1.1 Maximum frequency

If the NoC it uses point-to-point communication access and sustains the maximum higher clock frequencies. It is operable up to 800 MHz for an unlimited link length. It helps have the maximum latency like cluster level latency SOC level latency system-level throughput and the average latency.

1.1.2 Virtual channel

It is the method of Digi-Cipher which is inherited by the distributed systems areas to the IP cores, in the scalable aspects and overall improvements of the systems [3]. Where the goals are to attach multiple buffers to the single input channel. This also and hares through these helps the router to make the design simple and prevents data mixing and shares the data through the virtual channels. They also help to decrease the latency and incorporate the network throughput [6, 10-14]. They reduce the congestion in the system where a single buffer is added with each input port which simplifies the design structure and within an instant period the data will be shared.

An embodiment of the VCs also helps the sharing of the data through the pair of channel performance to current channels [15, 16]. The NoCs without virtual channels have lower latency values (see Figure 3). Virtual channels are utilized to give the required bandwidth than individual networks. Each VC will have a logical interface for queuing, scheduling, and accounting [17].

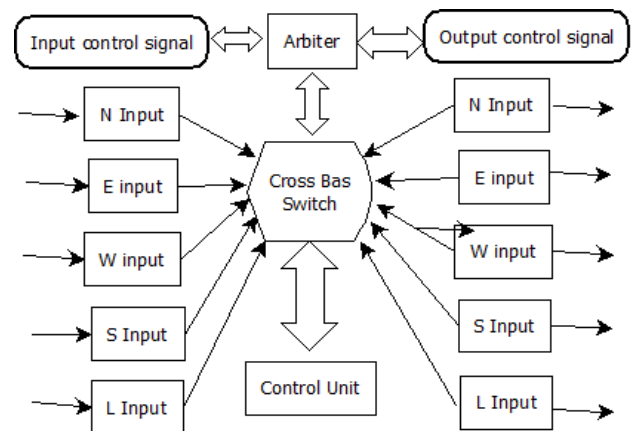


Figure 3. Internal structure of crossbar

1.2 DAMQ

In the virtual channels, it's difficult for the input storage with high weight; it becomes ideal in such conditions to eliminate more numbers of buffers are added. To control these buffers DAMQ is added to fetch good results.

They are implemented for nullifying the inconvenience of the traffic and maintenance spacing between the ports; they help to rectify the merging of information by implementing the deadlock condition. The DAMQ helps to minimize the delays occurring during the transmission of flits by adding the link-listed mechanism [18].

1.3 Static and dynamic buffer-based inputs and outputs

There is each virtual channel involves static input ports depending on parallel FIFO where the data must be read or written. A simple buffer can have the multi-channels as

directed by the “link list”. Similarly, it established the DMAQ’s simple architecture, but the data takes 5 steps to execute in Figure 4.

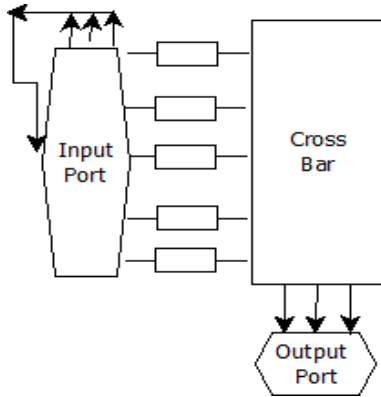


Figure 4. Dynamically allocated multi queues

1.4 EDVC

The EDVC provides the algorithms for the development of service networks. The main aim of adopting the designs for the network requirements, is they also help and advise the services of the efficient usage of resources. This helps to provide the guaranteed rate, with short-term contrasts. Implementing such ideas that take explicit functions are necessary. The efficient dynamic takes certain steps for execution but gained flits are excellent [10, 19-21]. Here the development of the service to the maximum extent, by developing a proper algorithm. That helps to maintain the traffic problem and influences the other users also. Demonstrating the waveforms of the queues can come to appoint that, in all the scenarios proposed by the EDVC is superior and the data which is fed to the system is accepted by sub-system revenue. Initialization of the memory address in the system is done by itself in a static manner.

1.5 CDVC

The CDVC is the one that has all the input ports, modules, and arbiters, with a crossbar switch. They utilize multiple buffers in allocating the receiving and the outgoing data to the equivalent reading and writing pointers. Here the look-up tables are maintained for the link list mechanism with Boolean functions. Every sequence is allocated with header list tables in which they will be having the addresses of the buffers, FIFO manner, and the slot state index consisting of the flits of buffer SRAMs. The slot state is nothing but the state that has the occupied slots of the SRAM that has the Boolean flag, once a bit is ready the corresponding bits also get enabled. The information regarding the writing pointers will be held in the tail list pointers. Whenever the slot state table is also updated the slot state table also get updated.

2. MOTIVATION

- In the invention of the present proposed work, many uncertainties were obtained in this phenomenon.

- The project also aims at solving the complications sustained when the code is executed in a very simpler manner. Hence the studies are conducted to solve the problems which aim for the best outcomes.

- The individual phases are concentrated in the research to facilitate the community.

- The paper collects information from various websites, and national and international publications and produced a high-quality throughput.

- The channel buffer organization for efficient communication using virtual channels ensures efficient communication. Since the virtual channels have high processing throughputs, they also ensure the flexibility to support the larger sequence of implementations that can be used for further research work.

- The virtual channels have buffers for safe storage of the temporary information allowing the packet of information to enter into the system.

- It dedicates the performance of the channel of CPU and system cores with separate application queues.

3. REVIEW OF THE PREVIOUS WORK

According to the previous research works that the papers are all about the efficient communication system. The base papers which had referred to for this project say that the NoC involves the communication interfaces which involve the packet routing mechanism. Header flits have messages regarding the messages where the packet data has to be sent to the proper destinations [22]. There are input and output buffers that are responsible for the efficient data flow and sharing of physical communications through multiple routers. Whenever the data has passed through the routers the router implies the buffer to store the data temporarily. Each port in the NOC can utilize the virtual channels in order to have control of the buffers that are storing the data. In the next context, the NoC makes the data to send it through the FIFO and whenever the data is huge, they are meant to send them in the queues [23-26]. In order to process the cores in the system the chip utilizes an efficient communication design similar to NOC.

They have virtual channels that help to recover the data flow of the system. Performance in the NoC system can be increased by adding the DAMQs, the DAMQs are the ones in which they are used to achieve virtual channel control of the system, and the buffers which are included in that [27-30]. The buffer slots in the system are used in order to control the issues of the traffic in the virtual channel. They also make the channel so flexible, that congestion in the system is avoided. EDVC is also one which is added in order to improve the architecture of the system. There is a new input port microarchitecture in DAMQ where they have the buffers which help to add the link-list tables in the buffer organization. Since the EDVC input port organization has a number of buffers it consumes more amount of large power consumption [31-34]. They also concentrate on NoC latency. In order to maintain the low latency time. In this implementation, the virtual channels include a different number of buffer slots which have the responsibility to improve the data flow it concentrates on the incoming and the outgoing data.

In the NOC systems, the communication channel will have multiple packets where the VCs are assumed as input ports in order to identify the storage across the incoming data [35]. We have the First Come First Serve (FCFS) concept in the FIFO buffers where the priorities are given to the first incoming data, later on, this is processed in the arbiters and the crossbar switches where they employ the data flow mechanism, and hence the buffer slots are statically allocated by using the

DAMQs. Buffer static mechanisms dynamically allocate the incoming data in the system where it gets processed in a single step and gets terminated [36-38]. As we have referred the many of the base papers the Dynamic Allocation of Multi-Queues have added a greater number of virtual channels in order to balance the communication load across the input ports. Hence, for better storage purpose the DAMQs has a greater number of buffers across the input and output ports in order to store the incoming and outgoing data temporarily [39]. The solutions for the various problems are obtained by surveying the different websites, and related research papers by configuring the microarchitecture of the DAMQs. According to this architecture, the numbers of virtual channels implemented in the DAMQs are equal to the number of delays during the recruitment of the registers [39].

4. METHODOLOGY

As the communication growth is wider in the current situation to have a more equipped way to get success in communication which must include the appropriate methodologies. In the proposed work packet routing where it has header flits for communicating messages, they require the proper method to reach the destination. The bunch of data has input-output buffers which have an efficient data flow and share the physical communication through multiple channels [10]. They have the FIFOs which are essential to serving the first inputs, where they are basically implemented to give the priorities.

They nurse the signals whether the data has reached the destinations properly are not. They also help the incoming data to associate the read and the write cycles. They help the data to complete the cycle in one arbitration. The addition of buffers to the DAMQs helps to withdraw the traffic problem in the system. They also help in maintaining the spacing between the data packets. They contain a link-listed mechanism that helps to opt the data held in the header flits. They also help the data not to overlap. They make the data very secure by implementing the parallel execution method.

4.1 Design and implementation

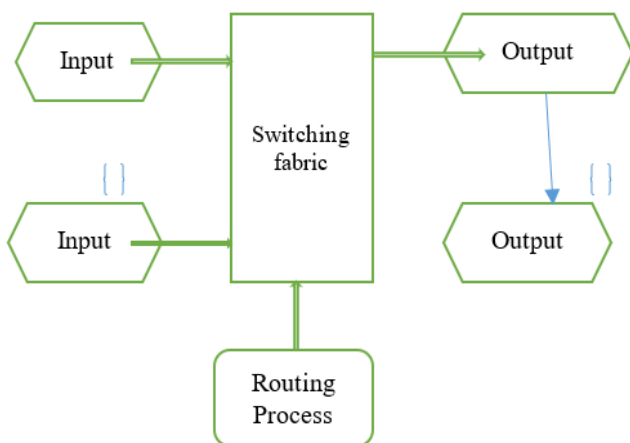


Figure 5. Proposed 8x8 router architecture

The implementation of the 8x8 Router increases the features of the information and helps to execute the process in less time with a method of parallelism. The figure represents the arbiter and crossbar which helps to allocate the shared resources in

the arbiter whereas the crossbars are used to store the information with no leakage. The FIFO buffers are used with link-listed DAMQs where the flits are dynamically allocated multi queues earlier there was a leakage problem, hence rectification is done for the leakage problem with the help of an efficient link list, and it helps us to eliminate the error termination Figure 5.

The 8x8 router architecture consists of decode logic, VC ID FIFOs, crossbar, arbiter, and latch. Our proposed work gives the 8 inputs from the decode logic to the FIFOs. Later the FIFO outputs are sent to the arbiter who allocates the shared resource. The shared resources are then sent to the crossbar section which allocates the memory allocation. The latches are the device which has two stable states. The information in the latch can stay longer because it has a feedback path. It is very appropriate to flip flops, but it is like asynchronous devices. This will not operate in the clock edge cycle. They will enable the high output and low output results. The asynchronous circuits play an important role in arbiter. The arbiter always gives the priority to the flits to be forwarded first. It always includes the first come first serve method. The arbiter will grant one signal at a time is called single arbitration. The decisions can be taken in a few picoseconds time. The stable state process is achieved by using the arbiter and it shows delays in processing.

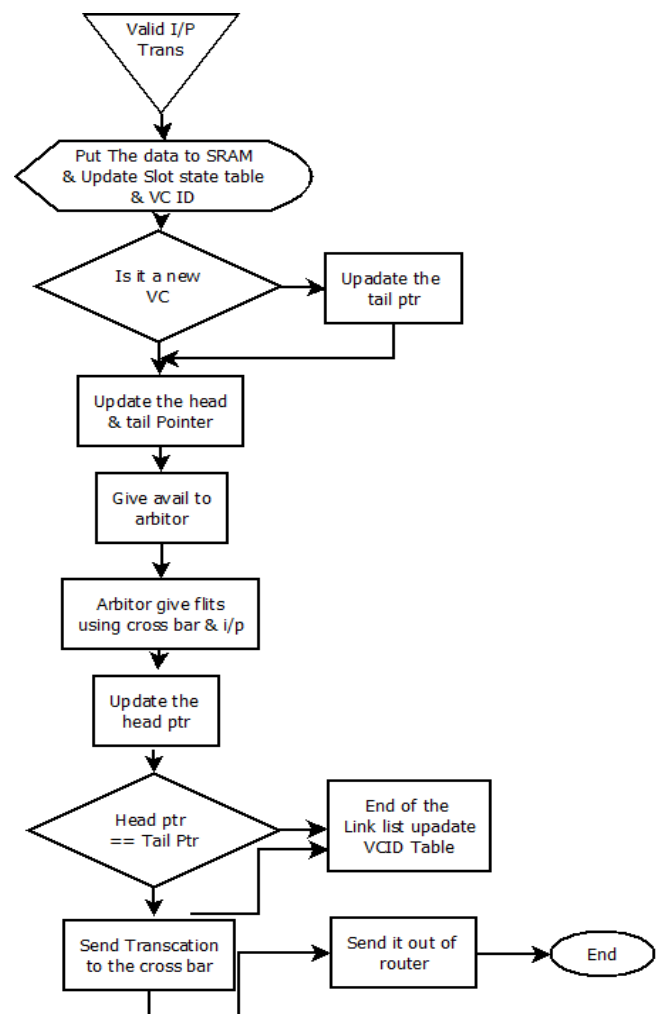


Figure 6. Flow chart of the communication process

The multiple inputs and multiple outputs have an individual switch in the crossbar section. They won't block the other

output when some other output is blocked. It is mainly used to send multiple flits and it uses to eliminate blocking switches it is also called a coordinate switching system [20-24]. The transistor at each point is also considered as a cross switch it forms a matrix array that has a basic fundament to recent flat panel displays. FIFO is the old method of the process which has implemented in earlier days, but the cost of the FIFOs is very expensive in those days. It is used to correct the data buffer and modernize the organization [25, 26]. It has a temporary behavior when the flits have been given priority queues. The interaction between the incoming and the outgoing flits has queuing systems that use the method of executing the data structure in Figure 6.

The flow of our project is explained step by step in this section, the valid input transactions are made by giving the 8 input flits across the input paths. After the transaction of the data, it should be dropped into the SRAM and updated slot state table and VC ID. The SRAM is used to store the memory for a longer period which can be accessed easily in the future times. The various virtual circuits are distinguished in a communication network and the virtual channel identifier is a numeric identifier. If it is a new virtual circuit ID is transferred along the input paths then update the head and tail pointer, if it is not a new virtual circuit ID then update only the tail pointer. Once the flits of the head pointer and tail pointer get updated it gives avail to the arbiter. The arbiter processes the single-step execution at a time of clock cycle.

The experimental results and case studies provide robust validation of the proposed NoC methods and technologies. The significant improvements in performance, power efficiency, and thermal management highlight the practicality and effectiveness of these innovations. The use of standard benchmarks and real-world workloads ensures that the findings are relevant and applicable to contemporary multiprocessor systems.

The arbiter gives the flits using the crossbar and the input. The crossbar allocates the data in the matrix array manner, which indicates the memory allocation with priority. The link listed dynamically allocated multi queues are used to maintain the architecture to be efficient. Later update the only header pointer after the flits given out by the arbiter which is implied by the crossbar and input. If the head pointer is equal to the tail pointer the flits will end up with a link list and update the VC ID table, then send the transaction for the crossbar. If the head pointer is not equal to the tail pointer, then it should be directly sending the transaction to the crossbar. The crossbar maintains the incoming and outgoing flits to be very effective with no loss of information which allocates the memory temporarily when it is required. Once the execution gets complete the flits are sent out of the router which gives us the proper output in the form of flits (Table 1).

Table 1. Comparison of FPGA and ASIC router

Router Type	FPGA		ASIC	
	Bits	Elements	Power	Area
Input router	415	825	232	234.252
Bulk input router	445	935	233.5	234.252
Extra Components (H/W)	0.8%	0.5%	0.25%	0.006%

5. RESULTS AND DISCUSSION

The main objective of this section is to give a clear picture of the top-level schematics, functional simulation results, and

implementation results of efficient virtual channel communication. The clock-gated simulation with the top router has been discussed in this chapter. The base paper is extended to support 8 port output architecture the acknowledgment logic and output routers port is extended using the architecture Table 2.

Table 2. DAMQs and EDVC

Details	DAMQs	EDVC
Implementation	Buffer Management	Channel allocation
	Allocation algorithm	Adaptive routing
	Flow Control	Priority
Experimental data	Latency	Latency
	Throughput	Throughput
	Utilization	Utilization
	Energy consumption	Congestion
Comparative analysis	Performance	Performance
	Scalability	Scalability
	Resource efficiency	Resource overhead

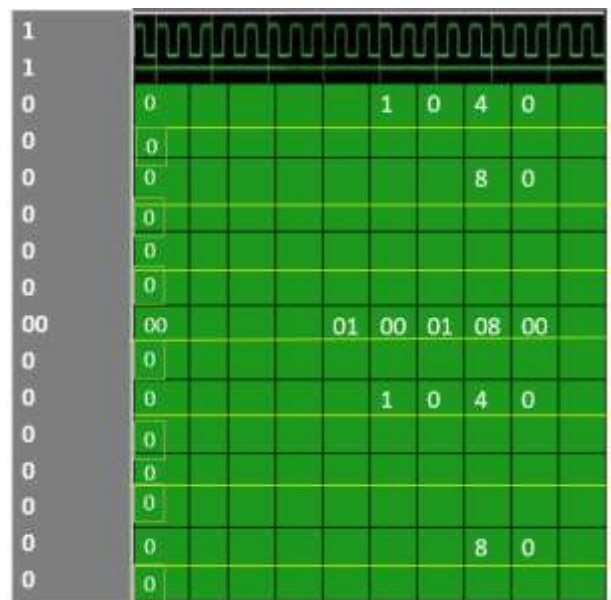


Figure 7. Functional simulation of arbiter wave with link list

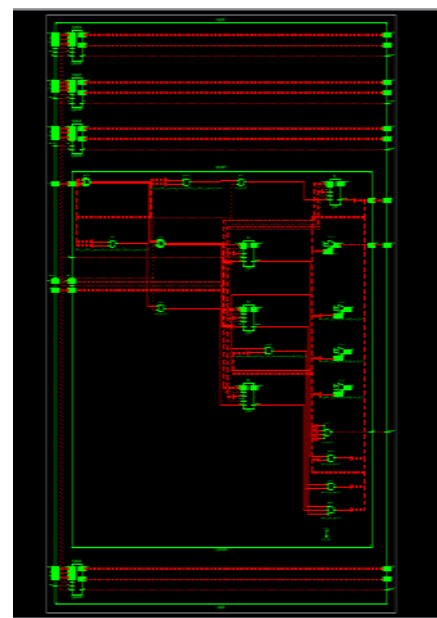


Figure 8. RTL view of NoC router

The RTL view of the router has the five input flits at the input port with the help of the arbiter and the crossbar the five data will be projected towards the outgoing flits as shown in Figures 7 and 8. The arbiters are used to access the allocated shared resources from the input port to the crossbar section to store the information temporarily which reduces the leakage of information.

5.1 Functional simulation result for arbiter wave link list

The transaction of the busy port in the system if it has a high-priority transaction, it has an intelligent routing mechanism that decodes the addresses and finds the busy routers, and then it is allowed for the other transaction to proceed. The router also helps to avoid deadlock scenarios for the error termination conditions.

5.2 Functional simulation result for crossbar

The crossbars are used to connect the data path from the input to the output, which enables the routing of flits through the network. They involve the three main contexts to examine in the crossbar they are data, a destination address, and a

request. Priority will be allocated with the grant signals for the crossbar switches. The crossbar input data is followed by the crossbar availability with the link list manner which helps us to get the acknowledgment of whether the signal is received or not at the destination path. When the request is not active it gives the grant to request signal with a next higher priority in Table 3.

Table 3. Performance comparison

Technology	Latency	Throughput	Utilization
Mesh	120	75	68
Tree	130	70	65
Torus	110	80	68

5.3 Functional simulation for LTR count non-zero

The long-term relationship is the essential part in this section to store the data. When the LTR count is zero the data will appear as soon as possible but whenever the data in nonzero it takes the time to display the data at the destination port with the delay of clock cycles Figures 9 and 10.

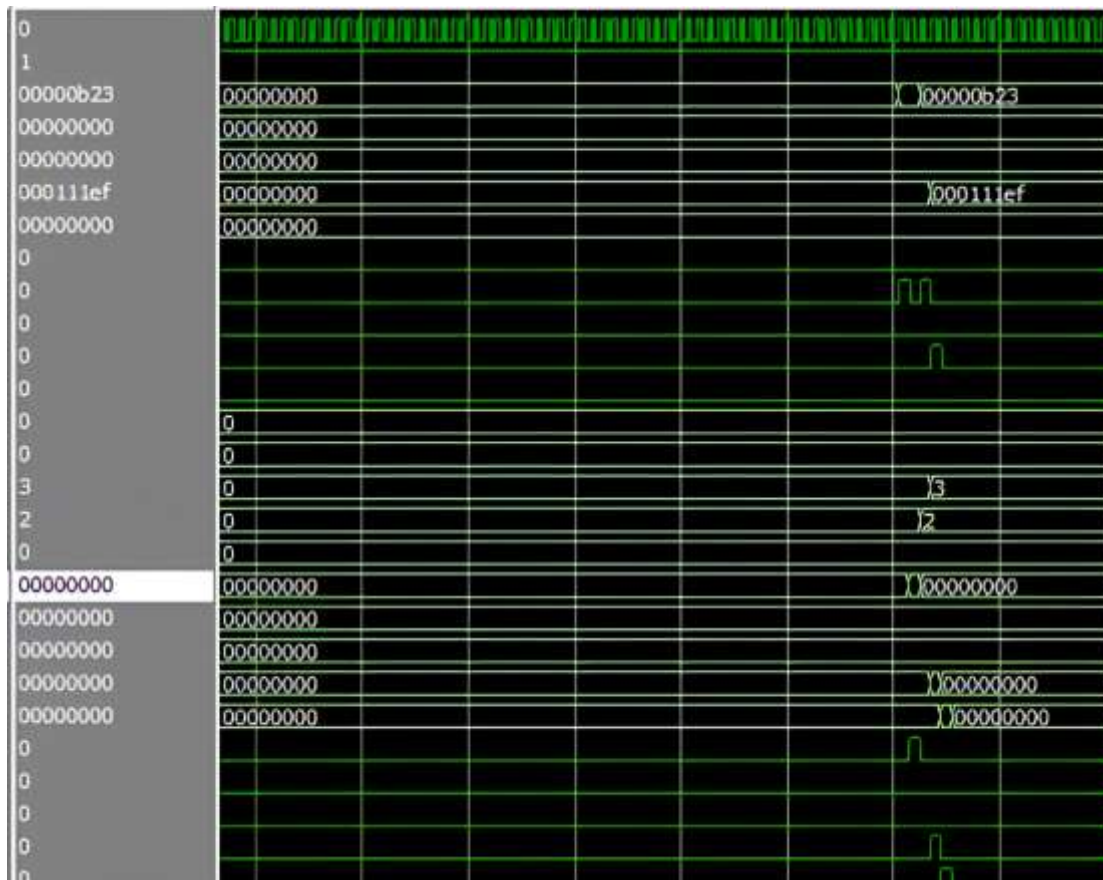


Figure 9. Functional simulation for cross bar wave with link list

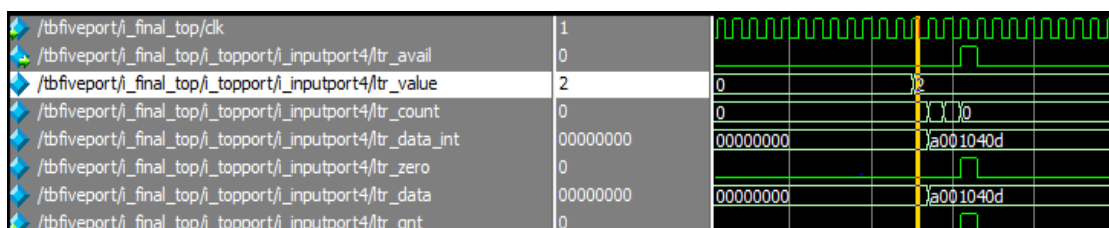


Figure 10. Functional simulation for LTR count non-zero

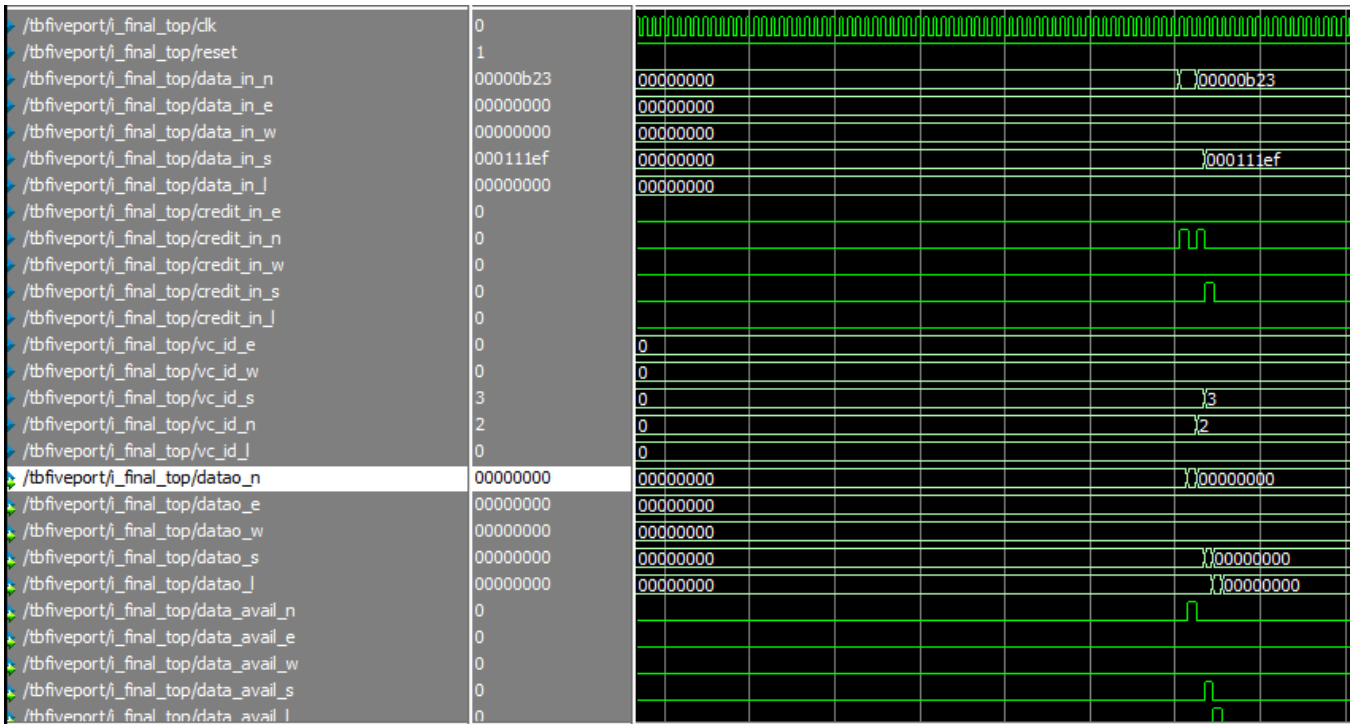


Figure 13. Functional simulation for the top wave with link list

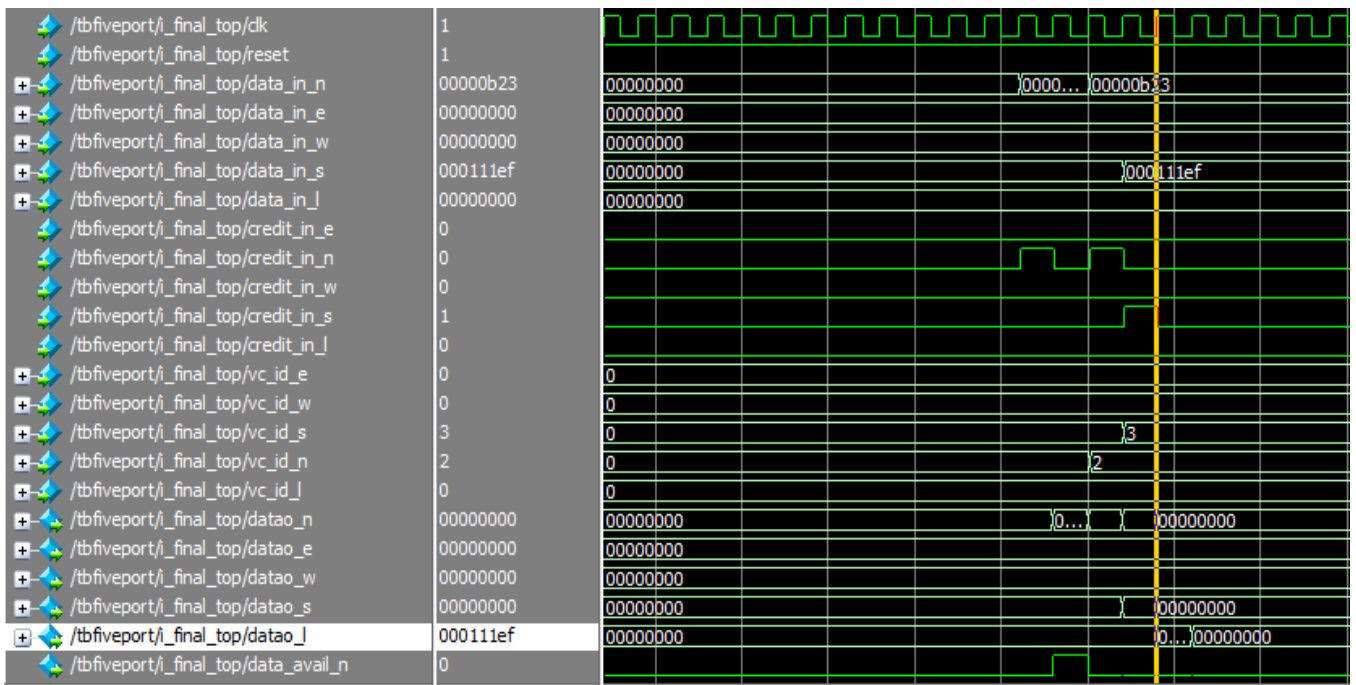


Figure 14. Functional simulation for the top wave link list with priority

5.6 Functional simulation for the full throughput

The enhancement work has been achieved to get the full throughput across the output ports where the efficiency of the router has been increased by using the link listed DAMQs in Figure 15. The buffers are used to maintain the data and to send the flits immediately along the input path.

5.7 Synthesis results and analysis

The implementation results are design to rectify the synthesis using the Spartan 3E FPGA family. The results are verified by using the simulation and the tools used to verify

the simulation by using Xilinx ISE and Modalism simulator using Verilog HDL Table 5.

The snapshots of the simulation results are verified with their functionalities in detail. The comparisons between the gated clock and the DAMQs are explained. The architecture implementation is also explained along with the incoming and outgoing flits, arbiter, and crossbar. The hardware implementation results are also discussed in this paper with the proper usage of allocated resources. In this paper, the router is used to add more features for the incoming and outgoing flits. The implemented 96.5% results were explained in detail, and it is tested and verified for proper usage by using Verilog HDL coding and simulation.

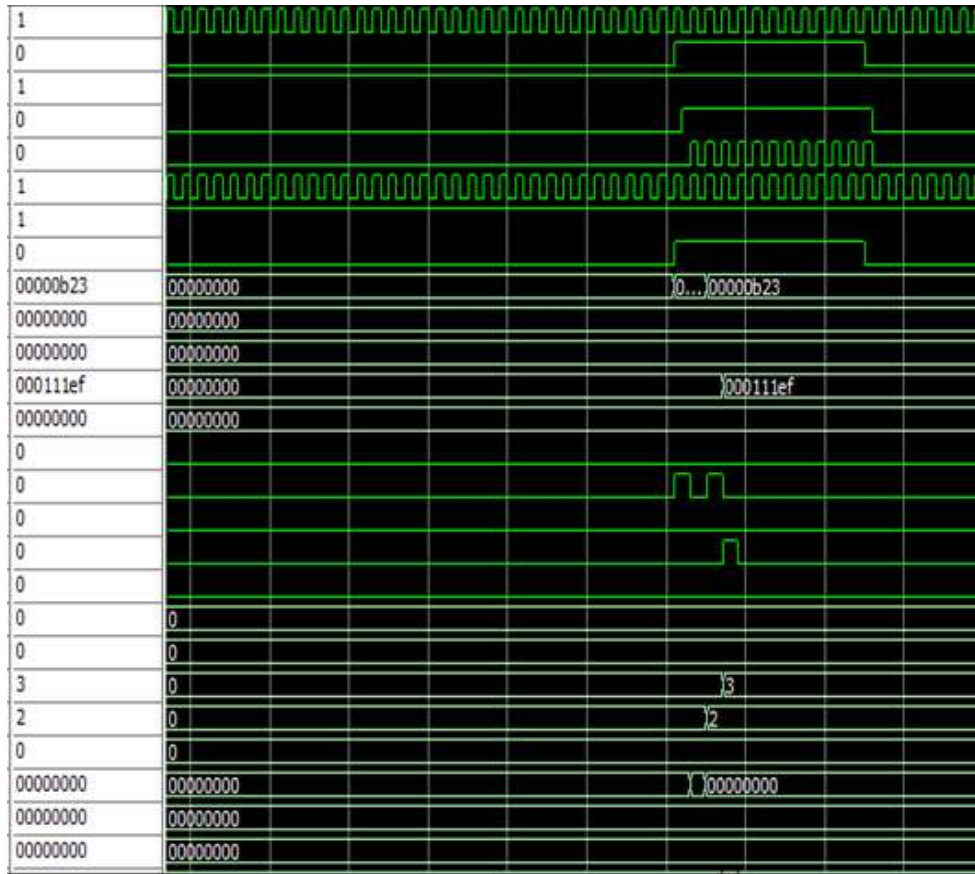


Figure 15. Functional simulation for full through of 8×8 routers

Table 5. Performance table of FPGA design

Input Port	Combine Logic Elements + FPGA Design	FPGA Design Register (bits)	FPGA Design F _{max} (MHz)
4 slots CDVC	96	112 (64 ^b)	353
4 slots VICHAR	74	132 (64 ^b)	301
4 slots Fast read write	84	107 (64 ^b)	383
4 slots fast write	77	97 (64 ^b)	1082
8 slots CDVC	181	204 (128 ^b)	285
8 slots VICHAR	305	392 (128 ^b)	196
8 slots Fast read write	205	175 (128 ^b)	243
8 slots fast write	173	173 (128 ^b)	850
16 slots CDVC	330	347 (256 ^b)	270
16 slots VICHAR	549	895 (256 ^b)	174
16 slots Fast read write	440	340 (256 ^b)	170
16 slots fast write	325	326 (256 ^b)	725
32 slots CDVC	671	674(512 ^b)	218
32 slots VICHAR	1182	2040 (512 ^b)	125
32 slots Fast read write	965	645 (512 ^b)	117
32 slots fast write	726	632 (512 ^b)	596
Arbitrary	1116	241	126
Crossbar	160	-	-

6. CONCLUSIONS

The proposed NoC designs demonstrate significant improvements in performance, power efficiency, and thermal management compared to traditional topologies like mesh, torus, and tree. The experimental results show that the proposed methods reduce average latency by up to 21%, increase throughput by 12.5%, decrease power consumption by 20%, and effectively manage thermal hotspots. The router is used to add more features for the incoming and outgoing flits. The implemented 96.5% results were explained in detail, and it is tested and verified for proper usage by using Verilog

HDL coding and simulation. The input port DAMQ and Network on Chip router has high expectancy. The DAMQ input port EDVC requires fewer hardware resources; it always gives 96.75% efficiency in the positions of performing and computer hardware. It consumes a minimum (13 input) transmits for field programmable gate array performance and a minimum (26.32%) power consumption. The implementation of the proposed work shows better (96.75%) performance achieved for hotspots and complements traffic patterns and 55% reduces the congestion of the incoming and outgoing flits. It 55% reduces the dynamic power consumption, and it provides high performance in this proposed method.

REFERENCES

- [1] Oveis-Gharan, M., Khan, G.N. (2015). Efficient Dynamic Virtual Channel organization and architecture for NoC systems. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 24(2): 465-478. <https://doi.org/10.1109/TVLSI.2015.2405933>
- [2] Savithri, G.R., Sunitha, G.S. (2022). Design of four port router for Network on Chip. *International Journal of Engineering Research & Technology*, 10(11): 221-223
- [3] Evripidou, M., Nicopoulos, C., Soteriou, V., Kim, J. (2012). Virtualizing virtual channels for increased Network on Chip robustness and upgradability. In 2012 IEEE Computer Society Annual Symposium on VLSI, MA, USA, pp. 21-26. <https://doi.org/10.1109/ISVLSI.2012.44>
- [4] Filippas, D., Nicopoulos, C., Dimitrakopoulos, G. (2022). Templated fused vector floating-point dot product for high-level synthesis. *Journal of Low Power Electronics and Applications*, 12(4): 56. <https://doi.org/10.3390/jlpea12040056>
- [5] Lai, M.C., Gao, L., Shi, W., Wang, Z.Y. (2008). Escaping from blocking: A dynamic virtual channel for pipelined routers. In 2008 International Conference on Complex, Intelligent and Software Intensive Systems, Barcelona, Spain, pp. 795-800. <https://doi.org/10.1109/CISIS.2008.46>
- [6] Gharan, M.O., Khan, G.N. (2011). Flexible simulation and modeling for 2D topology NoC system design. In 2011 24th Canadian Conference on Electrical and Computer Engineering (CCECE), ON, Canada, pp. 000180-000185. <https://doi.org/10.1109/CCECE.2011.6030434>
- [7] Filippas, D., Margomenos, N., Mitianoudis, N., Nicopoulos, C., Dimitrakopoulos, G. (2021). Low-cost online convolution checksum checker. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 30(2): 201-212. <https://doi.org/10.1109/TVLSI.2021.3119511>
- [8] Hag, A.A.Y., Rahman, M.H., Nor, R.M., Sembok, T.M.T., Miura, Y., Inoguchi, Y. (2015). Uniform traffic patterns using virtual cut-through flow control on VMMN. *Procedia Computer Science*, 59: 400-409. <https://doi.org/10.1016/j.procs.2015.07.553>
- [9] Konstantinou, D., Nicopoulos, C., Lee, J., Dimitrakopoulos, G. (2021). Multicast-enabled network-on-chip routers leveraging partitioned allocation and switching. *Integration*, 77: 104-112. <https://doi.org/10.1016/j.vlsi.2020.10.008>
- [10] Zhang, H., Wang, K., Dai, Y., Liu, L. (2012). A multi-VC dynamically shared buffer with prefetch for Network on Chip. In 2012 IEEE Seventh International Conference on Networking, Architecture, and Storage, Xiamen, China, pp. 320-327. <https://doi.org/10.1109/NAS.2012.39>
- [11] Gharan, M.O., Khan, G.N. (2012). A novel virtual channel implementation technique for multi-core on-chip communication. In 2012 Third Workshop on Applications for Multi-Core Architectur, New York, USA, pp. 36-40. <https://doi.org/10.1109/WAMCA.2012.12>
- [12] Pandurangi, V.J., Managuli, M., Salakhe, S., Bangarshetti, S., Kunchur, P.N. (2021). Detection & classification of electronic nose system. In 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, pp. 1-4. <https://doi.org/10.1109/ICICCS51141.2021.9432248>
- [13] Martina, M., Masera, G. (2013). Improving network-on-chip-based turbo decoder architectures. *Journal of Signal Processing Systems*, 73: 83-100. <https://doi.org/10.1007/s11265-013-0733-7>
- [14] Kim, J., Balfour, J., Dally, W. (2007). Flattened butterfly topology for on-chip networks. In 40th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO 2007), Chicago, IL, USA, pp. 172-182. <https://doi.org/10.1109/MICRO.2007.29>
- [15] Naessens, F., Derudder, V., Cappelle, H., Hollevoet, L., Desmet, M., Abdelhamid, A.M., Vos, I., Folens, L., O'Loughlin, S., Singirikonda, S., Dupont, S., Weijers, J.W., Dejonghe, A., Van der Perre, L. (2010). A 10.37 mm² 675 mW reconfigurable LDPC and Turbo encoder and decoder for 802.11 n, 802.16 e and 3GPP-LTE. In 2010 Symposium on VLSI Circuits, Honolulu, HI, USA, pp. 213-214. <https://doi.org/10.1109/VLSIC.2010.5560292>
- [16] Mukherjee, S.S., Bannon, P., Lang, S., Spink, A., Webb, D. (2002). The Alpha 21364 network architecture. *IEEE Micro*, 22(1): 26-35. <https://doi.org/10.1109/40.988687>
- [17] Taylor, M.B., Kim, J., Miller, J., Wentzlaff, D., Ghodrati, F., Grenwald, B., Hoffman, H., Johnson, P., Lee, J.W., Lee, W., Ma, A., Saraf, A., Seneski, M., Shnidman, N., Strumpfen, V., Frank, M., Amarasinghe, S., Agarwal, A. (2002). The raw microprocessor: A computational fabric for software circuits and general-purpose programs. *IEEE Micro*, 22(2): 25-35. <https://doi.org/10.1109/MM.2002.997877>
- [18] Nicopoulos, C.A., Park, D., Kim, J., Vijaykrishnan, N., Yousif, M.S., Das, C.R. (2006). ViChar: A dynamic virtual channel regulator for network-on-chip routers. In 2006 39th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO'06), Orlando, FL, USA, pp. 333-346. <https://doi.org/10.1109/MICRO.2006.50>
- [19] Teimouri, N., Modarressi, M., Tavakkol, A., Sarbazi-Azad, H. (2011). Energy-optimized on-chip networks using reconfigurable shortcut paths. In Architecture of Computing Systems-ARCS 2011: 24th International Conference, Como, Italy, pp. 231-242. https://doi.org/10.1007/978-3-642-19137-4_20
- [20] Oveis-Gharan, M., Khan, G.N. (2011). Flexible simulation and modeling for 2D topology NoC system design. In 2011 24th Canadian Conference on Electrical and Computer Engineering(CCECE), Niagara Falls, Canada, pp. 180-185. <https://doi.org/10.1109/CCECE.2011.6030434>
- [21] Duraisamy, K., Thanarajan, T., Alharbi, M. (2022). Implementation of OMAR pigeon space-time (OPST) algorithm to mitigate the interference and Peak-to-Average Power Ratio (PAPR) using RPR mobile and HST-HM in the 5G. *Traitement du Signal*, 39(5): 1631-1638. <https://doi.org/10.18280/ts.390520>
- [22] Sangeetha, T., Kumutha, D., Bharathi, M.D., Surendran, R. (2022). Smart mattress integrated with pressure sensor and IoT functions for sleep apnea detection. *Measurement: Sensors*, 24: 100450. <https://doi.org/10.1016/j.measen.2022.100450>
- [23] Kumutha, D., Prabha, N.A. (2018). Hilbert fast-SAMP with different channel estimation schemes of BER analysis in MIMO-OFDM system. *International Journal*

- of Internet Technology and Secured Transactions, 8(2): 221-237. <https://doi.org/10.1504/IJTST.2018.093388>
- [24] Huang, T.C., Ogras, U.Y., Marculescu, R. (2007). Virtual channels planning for networks-on-chip. In 8th International Symposium on Quality Electronic Design (ISQED'07), San Jose, CA, USA, pp. 879-884. <https://doi.org/10.1109/ISQED.2007.169>
- [25] Vaiyapuri, T., Shankar, K., Rajendran, S., Kumar, S., Acharya, S., Kim, H. (2023). Blockchain assisted data edge verification with consensus algorithm for machine learning assisted IoT. *IEEE Access*, 11: 55370-55379. <https://doi.org/10.1109/ACCESS.2023.3280798>
- [26] Magnusson, P.S., Christensson, M., Eskilson, J., Forsgren, D., Hallberg, G., Honberg, J., Larsson, F., Moestedt, A., Werner, B. (2002). Simics: A full system simulation platform. *Computer*, 35(2): 50-58. <https://doi.org/10.1109/2.982916>
- [27] Nagappan, K., Rajendran, S., Alotaibi, Y. (2022). Trust aware multi-objective metaheuristic optimization based secure route planning technique for cluster based IIOT environment. *IEEE Access*, 10: 112686-112694. <https://doi.org/10.1109/ACCESS.2022.3211971>
- [28] Alfaraj, N., Zhang, J., Xu, Y., Chao, H.J. (2011). HOPE: Hotspot congestion control for Clos Network on Chip. In Proceedings 5th IEEE/ACM NoCS, Pittsburgh, PA, USA, pp. 17-24. <https://doi.org/10.1145/1999946.1999950>
- [29] Dumitriu, V., Khan, G.N. (2009). Throughput-oriented NoC topology generation and analysis for high performance SoCs. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 17(10): 1433-1446. <https://doi.org/10.1109/TVLSI.2008.2004592>
- [30] Ramanujam, R.S., Soteriou, V., Lin, B., Peh, L.S. (2010). Design of a high-throughput distributed shared-buffer NoC router. In 2010 Fourth ACM/IEEE International Symposium on Networks-on-Chip, Grenoble, France, pp. 69-78. <https://doi.org/10.1109/NOCS.2010.17>
- [31] Tamilvizhi, T., Surendran, R., Romero, C.A.T., Sendil, M.S. (2022). Privacy preserving reliable data transmission in cluster based vehicular adhoc networks. *Intelligent Automation & Soft Computing*, 34(2): 1265-1279. <https://doi.org/10.32604/iasc.2022.026331>
- [32] Joseph, J.M., Bamberg, L., Hajjar, I., Perjikolaie, B.R., Ortiz, A.G., Pionteck, T. (2021). Ratatoskr: An open-source framework for in-depth power performance and area analysis in 3D NoCs. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 32(1): 1-21. <https://doi.org/10.1145/3472754>
- [33] Managuli, M., Mahantesh, K., Lakshminarayana, M., Managuli, S.C. (2024). Energy-efficient technique to improve the system using MIMO. In *Digital Convergence in Antenna Designs: Applications for Real-Time Solutions*, pp. 223-255. <https://doi.org/10.1002/9781119879923.ch11>
- [34] Mishra, B.K. (2023). *Data Science and Interdisciplinary Research: Recent Trends and Applications*. Bentham Science Publishers.
- [35] Dally, W.J. (1990). Virtual-channel flow control. *ACM SIGARCH Computer Architecture News*, 18(2SI): 60-68.
- [36] Alotaibi, Y., Rajasekar, B., Jayalakshmi, R., Rajendran, S. (2024). Falcon optimization algorithm-based energy efficient communication protocol for cluster-based vehicular networks. *Computers, Materials & Continua*, 78(3): 4243-4262. <http://doi.org/10.32604/cmc.2024.047608>
- [37] Kim, D.H., Athikulwongse, K., Lim, S.K. (2009). A study of through-silicon-via impact on the 3D stacked IC layout. In *Proceedings of the 2009 International Conference on Computer-Aided Design*, San Jose California, pp. 674-680. <https://doi.org/10.1145/1687399.1687524>
- [38] Pasricha, S., Dutt, N. (2008). Trends in emerging on-chip interconnect technologies. *Information and Media Technologies*, 3(4): 630-645. <https://doi.org/10.11185/imt.3.630>
- [39] Pavlidis, V.F., Friedman, E.G. (2007). 3-D topologies for networks-on-chip. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 15(10): 1081-1090. <https://doi.org/10.1109/TVLSI.2007.893649>