International Journal of Communication Networks and Information Security

ISSN: 2073-607X, 2076-0930

Volume 14 Issue 1s Year 2022 Page 37:53

Design and Performance Analysis of Low Latency Routing Algorithm based NoC for MPSoC

T. Nagalaxmi[1], Dr. E.Sreenivasa Rao[2], Dr.P.Chandrasekhar[3]

[1]Research Scholar, Department of Electronics and Communication Engineering, Osmania University, Hyderabad, Telangana, India.
[2]Professor, Department of Electronics and Communication Engineering, Vasavi College of Engineering, Hyderabad, Telangana, India.
[3]Professor, Department of Electronics and Communication Engineering, Osmania University, Hyderabad, Telangana, India.
tnagalaxmi@stanley.edu.in

Article History

Received: 10 July 2022
Revised: 23 September 2022
Accepted: 16 October 2022

Abstract

The Network on Chip is appropriate where System-on-Chip technology is scalable and adaptable. The Network on Chip is a new communication architecture with a number of benefits, including scalability, flexibility, and reusability, for applications built on Multiprocessor System on a Chip (MPSoC). However, the design of efficient NoC fabric with high performance is critically complex because of its architectural parameters. Identifying a suitable scheduling algorithm to resolve arbitration among ports to obtain high-speed data transfer in the router is one of the most significant phases while designing a Network on chip-based Multiprocessor System on a Chip. Low latency, throughput, space utilization, energy consumption, and reliability for Network on chip fabric are all determined by the router. The performance of the NoC system is hampered by the deadlock issues that plague conventional routing algorithms. This work develops a novel routing algorithm to address the deadlock problem. In this paper, a deterministic shortest path deadlock-free routing method is developed based on the analysis of the Turn Model. In the 2D-mesh structure, the algorithm uses separate routing methods for the odd and even columns. This minimizes the number of paths for a single channel, congestion, and latency. Two test scenarios—one with and one without a load test—were used to evaluate the proposed model. For a zero-load network, three clock cycles are utilized to transfer the packets. For the load network, five clocks are utilized to transfer the packets. The latency is measured for both cases without load and with load test and the corresponding latency is 3ns and 7ns respectively. The proposed method has an

18.57Mbps throughput. The area and power utilization for the proposed method are 69% (IO utilization) and 0.128W respectively. In order to validate the proposed method, the latency is compared with existing work and 50% latency is reduced both with and without congestion load.

Keywords - Low latency, Network on Chip, Zynq Multiprocessor System on Chip, Routing algorithm, and Mesh topology.

CC License
CC-BY-NC-SA 4.0

1. Introduction

The classic System-on-Chip (SoC) communication architecture based on the shared bus mechanism has faced several difficulties in recent years when hundreds or thousands of IP cores are combined on a single chip, primarily in the SoC's scalability [1]. The problem has become prominent, and the problem of limited area resources has increasingly become the bottleneck of expanding the system. In addition, the traditional System-on-Chip cannot truly realize parallel communication, which leads to the problem of communication efficiency. In order to solve these problems, reference [2 to 5] introduces a new Network-on-Chip (NoC) communication architecture that separates communication resources from computing resources. This technology is transplanted into the SoC, which is the problem brought by the traditional bus architecture. NoC not only provides a good area but also provides good parallel communication capability, thereby improving data throughput and network performance. The main research contents of NoC interconnection include topology structure, switching mechanism, routing algorithm, congestion control, and router structure [6 to 8]. At present, routing algorithms and topology are still two main aspects of NoC research. In this paper, the low latency-based routing algorithm is developed with a 2D mesh topology. The main contribution of this work is as follows:

The algorithm proposed in this paper abandons the singularity adopted by the dimensional order routing, so that, it always transmits from the X direction first, which is easy to cause blocking in this direction,

- The distributed deterministic routing mechanism is proposed in different directions for the parity column, thereby reducing the network communication delay. The complete flow of the proposed routing algorithm is explained in section 3.2.
- The proposed algorithm is tested in with and without congestion.

The remaining portions of the paper are structured as follows: in section II, work connected to NoC and the current states of research effort are covered. The well-known NoC routing algorithms and proposed routing algorithm is explained in section IIII. The proposed router architecture in is discussed in section IV. The corresponding simulation results are explained in section V.

2. Related Work

NoC is an on-chip interconnection structure formed by routing nodes and links between them through a certain topology. Common NoC topologies include the mesh, torus ring, fat tree, hypercube, and spidergon. Fig.1 shows the NoC model based on the classic 2D-mesh structure [10, 11]. In addition, some routing nodes and links can be removed from the regular network topology structure to form an irregular topology structure.

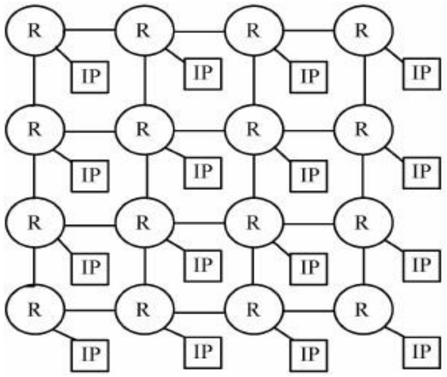


Figure 1. NOC with mesh structure

The topology can be divided into regular network topology and irregular network topology [12]. However, most of the researchers prefer regular mesh networks to it has lower network diameter and average distance, constant node degree, and scalability [13]. This paper is the research on the related routing technology on the regular mesh topology. Based on the different characteristics of the network topology, the routing algorithm will also change. For a certain NoC topology, the data communication between IP cores affects the network performance to a large extent, so an effective routing algorithm is crucial to the performance of a NoC network [14 to 15]. According to different standards, routing algorithms can be divided into the following categories. For instance, it can be separated into deterministic routing and adaptive routing depending on whether the network status is taken into account. Similar to the traditional XY routing algorithm and the e-cube routing method, the prior is a static routing algorithm that chooses a fixed routing path between the source and destination nodes [16]. These algorithms are simpler to implement and require less hardware logic. The adaptive routing algorithm affords multiple paths between the source and destination nodes, which improves the adaptive ability to route so that hot spots and faulty nodes can be bypassed to some extent, but the hardware implementation is more complicated [17]. Shortest path routing and non-shortest path routing techniques can be used to determine whether the number of paths is equal to the Manhattan distance. In order to reduce latency, the shortest path routing is chosen for routing. When there is a faulty node, it is necessary to choose a non-shortest routing strategy, but a live lock may occur because the data packet can never reach the destination node. However, the live lock can be avoided through the shortest route. Deadlock and live lock are necessary conditions for efficient routing algorithms. Deadlock refers to the fact that the data packets apply for other resources while occupying their own resources, which will cause a resource dependency loop, and the data packets are blocked and cannot be routed. As shown in Fig. 2, packet A and packet B occupy the buffer of this node respectively.

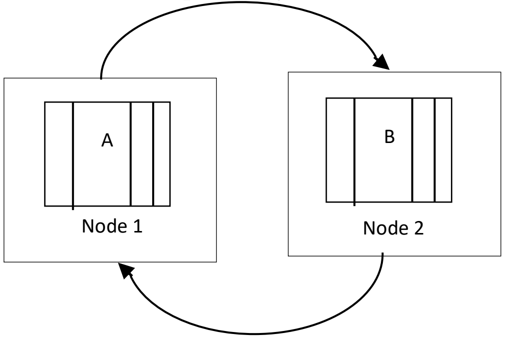


Figure 2. Deadlock Description

When the buffer zone is used, they apply for each other's buffer zone, resulting in a deadlock situation.
At present, there are three main solutions for deadlocks. First, when designing routing algorithms, deadlocks are avoided by restricting the direction of routing; secondly, deadlocks are avoided by adding virtual channels; thirdly, in the case of deadlocks, by forcing data packets to release resources to avoid deadlocks which break the resource dependency cycle. Most researchers mainly focus on the first two methods to solve the deadlock problem, and this paper adopts the best way to solve the deadlock problem.
For 2D mesh, the authors of [18] presented the prime turn model (PRTM) and the first last turn model (FLTM). These two models are built on a deadlock-free, adaptive routing method. The system C programming language was employed by the writers to carry out simulations. XY routing algorithm east last turn model, odd-even turn model, and column portioning turn models are some examples of more traditional methodologies that are used to compare the latency of the proposed PRTM and FLTM systems. In real-world circumstances, this proposed methodology is unable to achieve the maximum throughput.

The high-performance minimum pressure turn model (MPTM) rerouting algorithm proposed in [19]. This MPTM routing algorithm is computed on a 3D mesh network range from 4x4x4 to 6x6x6. This MPTM routing algorithm, provides the deadlock-free without using virtual channels. The MPTM works based on repetitive turn aspects with vertical and planar turn restriction mechanism. The authors claim that this proposed model works cost-effectively inconsideration of virtual channels. The latency of the proposed algorithm is compared with balanced odd-even, 3D odd-even, and repetitive turn model (RPTM) algorithms. But, this 3D model consumes more computational resources in terms of time and area.

In research article [20], [22] the theoretical study is carried out for designing a deadlock-free NoC architecture based on a turn model with a partially adaptive logic-based distributed routing algorithm. In this article, the author explained two properties that are essential to design the proposed routing algorithm. In this approach 16X16 mesh is considered and implemented in SystemC. The proposed method is compared with the odd-even and repetitive turn model in terms of throughput and latency. However, this proposed algorithm is not suitable for real-time applications due to non-concurrent execution of the algorithm will take more computational time.

The majority of current efforts cannot be implemented for real-time applications. This research proposes a novel turn-based routing method that may be appropriate for real-time applications. The suggested routing method is to port to an SoC target, and on the logic analyzer to check the latency.

3. NOC Routing Algorithms

3.1. Turn Model

The wormhole switching mechanism is widely used due to the small node cache and low latency required for routing. However, since the data packets are divided into flits without releasing resource nodes during the data transmission process, it is easy to generate resource dependency loops, which can be referred to as deadlock. The Turn model can solve the deadlock problem very well. Its main idea is to analyze whether the data packet may form a loop, and then when designing the routing algorithm, it can avoid the occurrence of deadlock by prohibiting specific directions. References [18, 19, 20, 21] are based on the Turn model to prohibit specific turns and avoid deadlock.

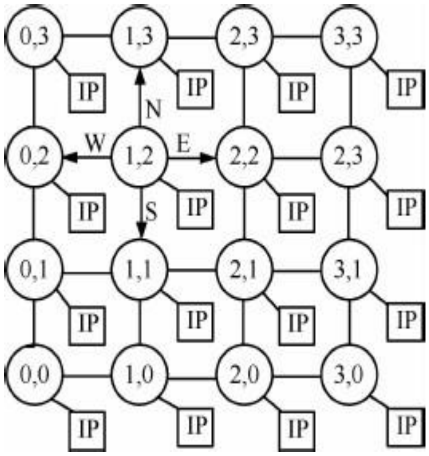


Figure 3. 2D-Mesh Node Channel

As shown in Fig. 3, the Turn Model is to identify the four channels adjacent to each routing node with E (East), W (West), S (South), and N (North). WS represents the steering from West to South, the steering from North to East is represented by NE, and so on, and the remaining turns are: WE, SW, EW, EN, ES and SE, and these 8 turns can constitute two rings, shown in Fig 4.

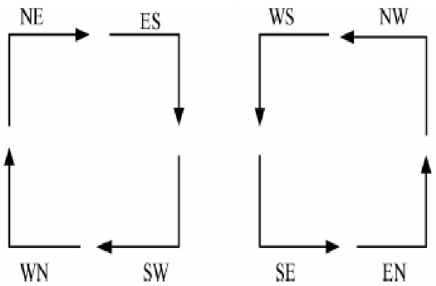


Figure 4. Turn Model

For instance, there won't be deadlock if the Turn model is used to examine the conventional XY routing method. According to the XY routing algorithm, traffic should be routed first in X and then Y directions. Fig. 5 depicts the steering for the XY path. The approved path is represented by the solid line, and the prohibited path is shown by the dotted line. XY routing prevents loops from ever forming, preventing stalemate.

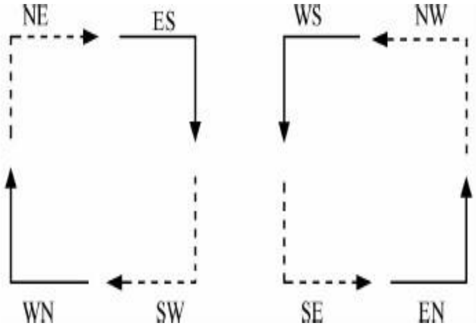


Figure 5. XY Routing Corresponds To The Turn Model

3.2. Proposed Routing algorithm

In XY routing, data packet routes in the X direction first and then in the Y direction to the destination node. Although XY routing is the minimum path and is deadlock-free, it is prone to congestion in one direction. The advantage of the routing method in this paper is that it is not always the routing mode in the X direction and then in the Y direction. Congestion occurs when routing is always performed in a single direction. From Figure 5, it can be seen from the turning model that the route has no dependency loop, so it is deadlock-free.

This paper adopts mesh topology. The algorithm in this paper abandons the singularity adopted by the dimensional order routing, so that, it always transmits from the X direction first, which is easy to cause blocking in this direction, and this paper adopts a distributed deterministic routing mechanism in different directions for the parity column, thereby reducing the network communication delay. The main contribution of algorithm is mainly described step wise as follows:

1. Waiting for the data packets;
2. Receive the data packet, if the destination address of the data packet is equal to the router address, send the data packet to the IP core connected to the router, otherwise, jump to the next step;
3. Send data packets along the x-axis or y-axis direction. If the x-coordinate or y-coordinate of the source and destination addresses are equal, the data packet is directly transmitted in this coordinate direction, otherwise, it jumps to the next step;
4. Determine the size of the x value of the source node and the destination node;
5. If the source node is in an even column and the destination node's x value is greater than the source node's, the Y first then X routing strategy is used; otherwise, the X first then Y routing algorithm is used;
6. If the source node is in an even column and the destination node's x value is smaller than the source node's, the routing algorithm of X followed by Y is used; otherwise, the routing algorithm of Y followed by X is used;
7. Return to step after sending a data packet (1).

Let (sx, sy) be the source router address, (dx, dy) be the destination router address (cx, cy) be the current router address, $\Delta x = dx - sx$, $\Delta y = dy - sy$, $rx = dx - cx$, $ry = dy - cy$. The flow of the algorithm is shown in Fig 6.

4. Proposed Architecture

This section presents the design of the proposed router architecture and provides a detailed description of a network with a 2 X 2 mesh topology. This design is simple for a credit-based flow control system that focuses specifically on latency reduction. A single channel router is incorporated within the design although the provisioning of a packet structure has been done for virtual channel bits. The proposed router is a single pipeline stage router design, unlike multiple pipeline stage routers. It ensures reduced latency, minimized clock cycles, and hardware requirements.

The network topological view shown below in Fig. 7. is a 2 X 2 mesh fashion based on connecting nodes with each router. The router functionality has been tested using this network structure. From sensor nodes, packets are introduced for testing and read at the nodes that could be any type of core processing.

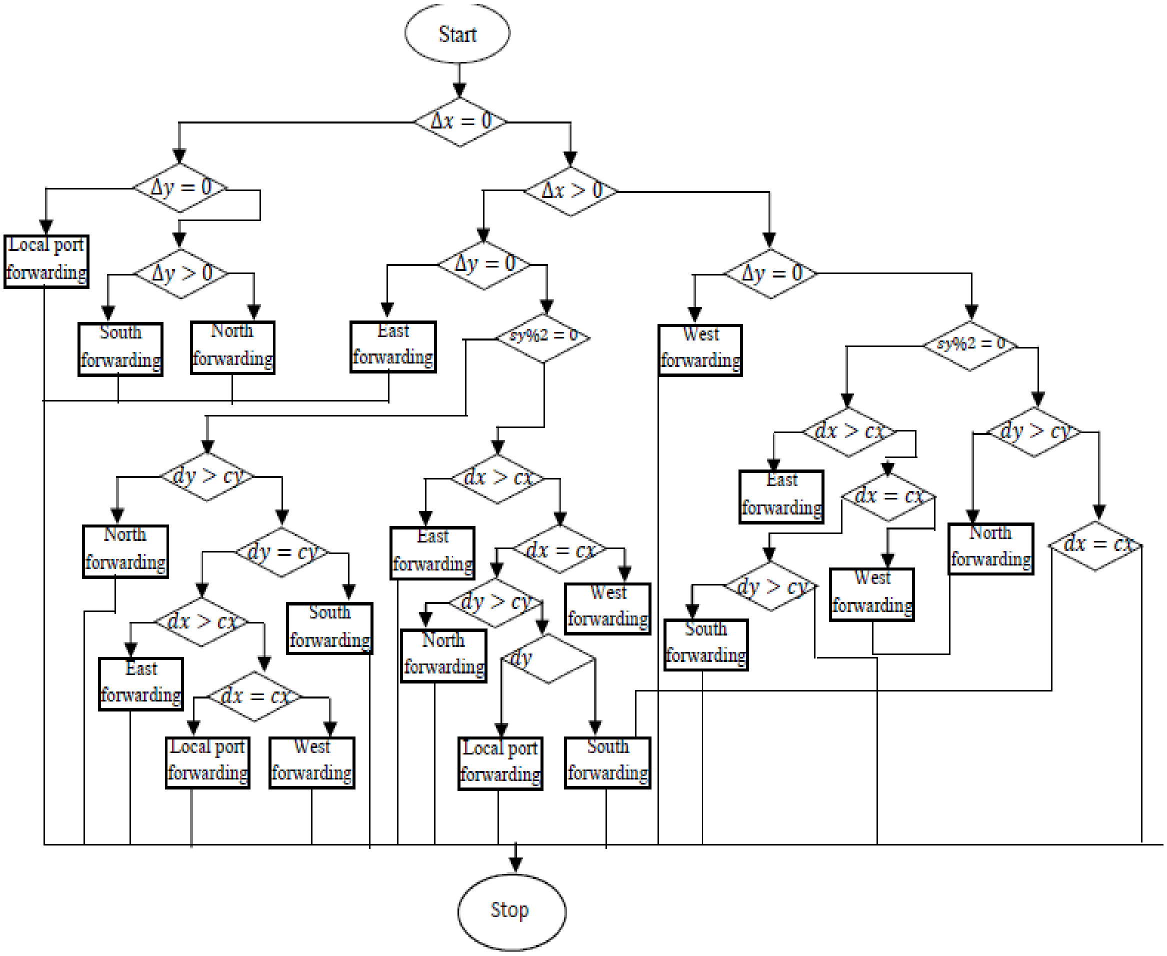


Figure 6. Proposed Routing Algorithm Flow

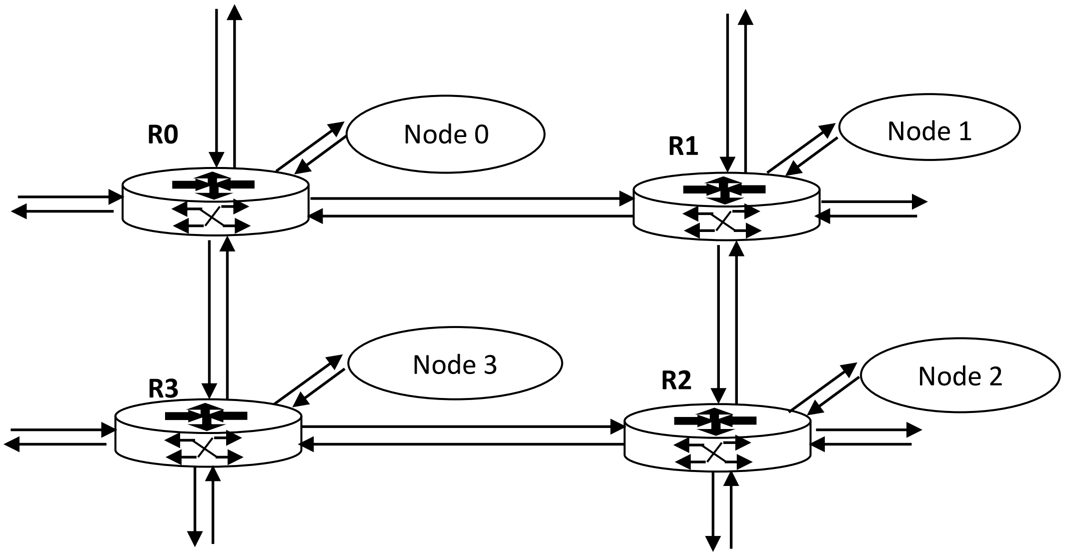


Figure 7. Topological View of the Network

4.1. Proposed Router Design Overview

The communication of a router is processed with cores and the other routers through the I/O ports of a router. As shown in Fig.8, the architectural block diagram, I/O ports include two channels, each of which is used to send and receive data and flow control bits. For advertisement of buffer slot availability, a corresponding signal is travelled in the opposite direction for data transmission leads to overcoming a router with packets bursting beyond the capacity.

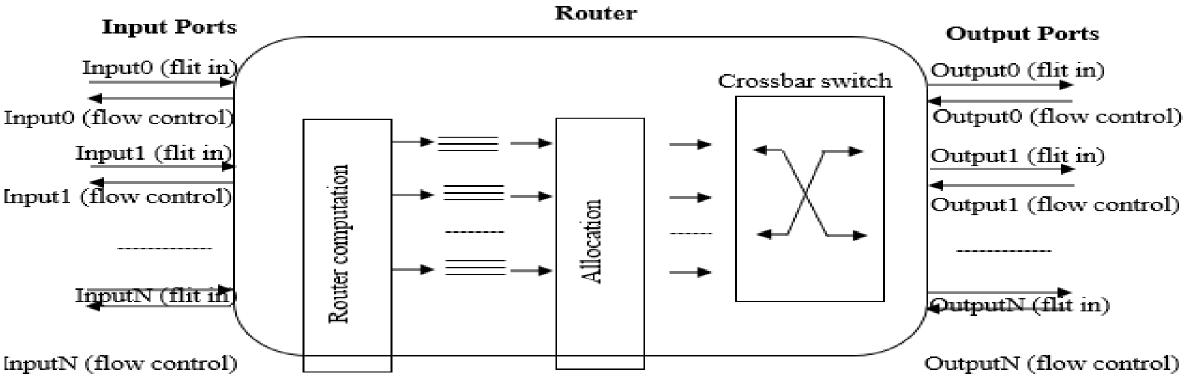


Figure 8. Proposed Router Architecture

For making the flit size smaller and routing simpler, a routing algorithm uses a lookup table based on the distributed deterministic routing. The routing process is computed based on the stored routing table in the hexadecimal format as the data flits inject into the router. According to the obtained destination ID from the flit, the output port number is tagged to the flit. The data flits are kept in the flit buffer until choosing the switch for the appropriate output port. Based on the tagged output port number, the input-first Separable allocator is used to handle the allocation and arbitration. The crossbar switch is connected to the right output port that allows sending data flits after allocation. From the input port, the flits are read in as it is a single pipeline stage router for one clock cycle and they have been stored in the flit buffer. From the router, the earlier stored flits have been read out for the same clock cycle. To inject a flit into a router, two clock cycles are considered.

5. Simulation Experiment Analysis

The experiment is validated using Verilog HDL based on the Xilinx Vivado platform. A 2*2 2D-mesh structure and a mechanism of wormhole switching are used in this experiment. The flit is the measurement of the minimum unit for data packet transmission. The simulation parameters areused as the frequency of 1GHz, router computation allocation with the time interval determines between packets using a data injection rate, and thetransmissionintervalswitchesbetweenflitsis2ns based on input ports output ports router crossbar.

In this experiment, the proposed routing algorithm is compared with the traditional XY routing algorithm. Based on the simulation environment and Zynq MPSoC, the proposed model is validated. The network performance evaluates using the average delay.

The packet delay describes as the difference in time between entering a packet into the network and when leaving the packet in the network. Three different parts are included in the data packet delay of a network:

$P_{delay} = T_{delay} + B_{delay} + L_{delay}$, where L_{delay} refers to the link's propagation delay, T_{delay} is the data packet's transmission delay, and B_{delay} is the buffering delay of the internal queue of a router. The propagation delay of a link is much smaller than the buffering and transmission delays. In the simulation, the link propagation delay ignores, i.e., $L_{delay} = 0$. Based on the averaging and accumulating of each data packet's delays, the average delay is obtained for a network.

The average delay for a network is defined as the accumulation and averaging of obtained delays for each data packet giving latency as:

$$Latency = \sum_{i=1}^{pk\_num} P_{\text{delay}}/pk\_num$$

Where pk_num describes the number of received packets.

5.1. Verilog Modeling

Based on Verilog Hardware Description Language, the router has been designed and simulated using Zynq-7000-XC7Z020 SoC device and Xilinx Vivado software. A bottom-up approach is considered to design a router. Primarily, the design is broken into different smaller components or blocks. In Verilog, each component is written as modules separately and all are integrated to create a router. Another module has been written as Network after a core router is ready. Based on the connecting inputs and outputs of a router, a router core with four instances is created and a 2X2 mesh topology is formed programmatically while introducing a routing table before allocating inputs for each router.

As shown in above Figure 9, the generated results from the Xilinx Vivado tool for a network module that encompasses the network for 2X2 mesh topology with the connection of four routers. The input and output ports of node_0 are 'send_ports_1_putflit_flit_in[12..0]' and 'recv_ports_1_getFlit[12..0]' that connecting to the 'router_1'. Similarly, all other remaining nodes or processor cores connect to the respective routers that have the same number, i.e. node 1,node 2, and node 3 are connected to routers 1, 2, and 3 respectively.

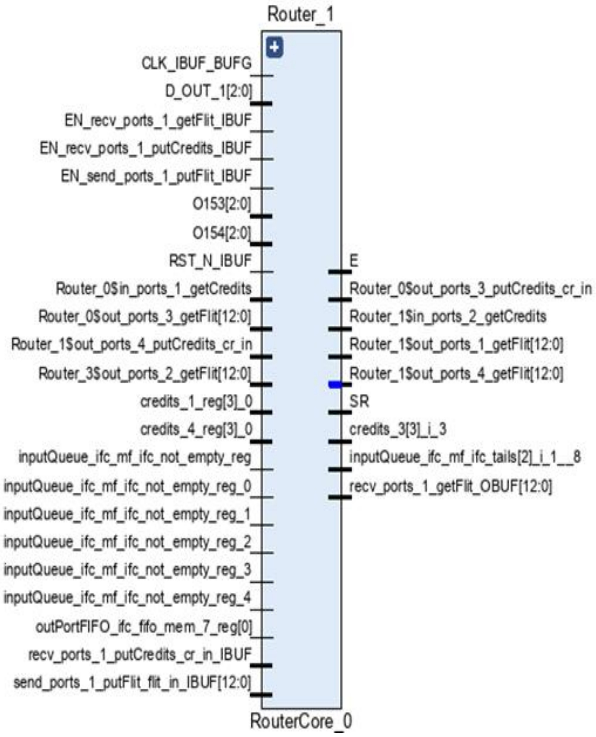

Router_1
CLK_IBUF_BUFG
D_OUT_1[2:0]
EN_recv_ports_1_getFlit_IBUF
EN_recv_ports_1_putCredits_IBUF
EN_send_ports_1_putFlit_IBUF
O153[2:0]
O154[2:0]
RST_N_IBUF
Router_0$in_ports_1_getCredits
Router_0$out_ports_3_getFlit[12:0]
Router_1$out_ports_4_putCredits_cr_in
Router_3$out_ports_2_getFlit[12:0]
credits_1_reg[3]_0
credits_4_reg[3]_0
inputQueue_ifc_mf_ifc_not_empty_reg
inputQueue_ifc_mf_ifc_not_empty_reg_0
inputQueue_ifc_mf_ifc_not_empty_reg_1
inputQueue_ifc_mf_ifc_not_empty_reg_2
inputQueue_ifc_mf_ifc_not_empty_reg_3
inputQueue_ifc_mf_ifc_not_empty_reg_4
outPortFIFO_ifc_fifo_mem_7_reg[0]
recv_ports_1_putCredits_cr_in_IBUF
send_ports_1_putFlit_flit_in_IBUF[12:0]
E
Router_0$out_ports_3_putCredits_cr_in
Router_1$in_ports_2_getCredits
Router_1$out_ports_1_getFlit[12:0]
Router_1$out_ports_4_getFlit[12:0]
SR
credits_3[3]_i_3
inputQueue_ifc_mf_ifc_tails[2]_i_1__8
recv_ports_1_getFlit_OBUF[12:0]
RouterCore_0

Figure 9. RTL Schematic Of Router Core

From Fig.7, the network topology diagram is assisted in understanding how the I/O ports map to the actual topology based on a block diagram. In a network, four flit input and output ports are there for four routers. From these ports, data packets inject from the cores or nodes into the routers when performing the testing. From the right outputs, packets have been expected to be read upon successful traversal which means from the right node or destination router.

5.2. Results and Analysis

A number of tests is performed after verifying the design for analyzing the router delay based on two different scenarios. For evaluating the design latency, two main cases are selected as reference tests that could be categorized into two phases: without load test and with load test. The simulation parameters considered for both with and without load tests are reported in table 1.

Table 1. Simulation Parameters

Router Topology	Mesh
Routing Algorithm	XY distributed deterministic routing Mechanism
Mesh size	Tested with 2X2,4X4 and16X16
Dataflow management	Credit Based-flow control
Latency without load test	3clockcyclesforoneflittransmission
Latency with load test	5clockcyclesforone flit transmission
So Device	Zynq-7000-XC7Z020
Simulation frequency	1GHz
Transmission interval between flits	2ns
Latency given by Vivado tool	3ns(without load test) 7ns(with load test)
LatencygivenbyAgilent1690A Logic Analyzer	2.5ns(without load test)5ns(with load test)

5.2.1. Without- Load Test

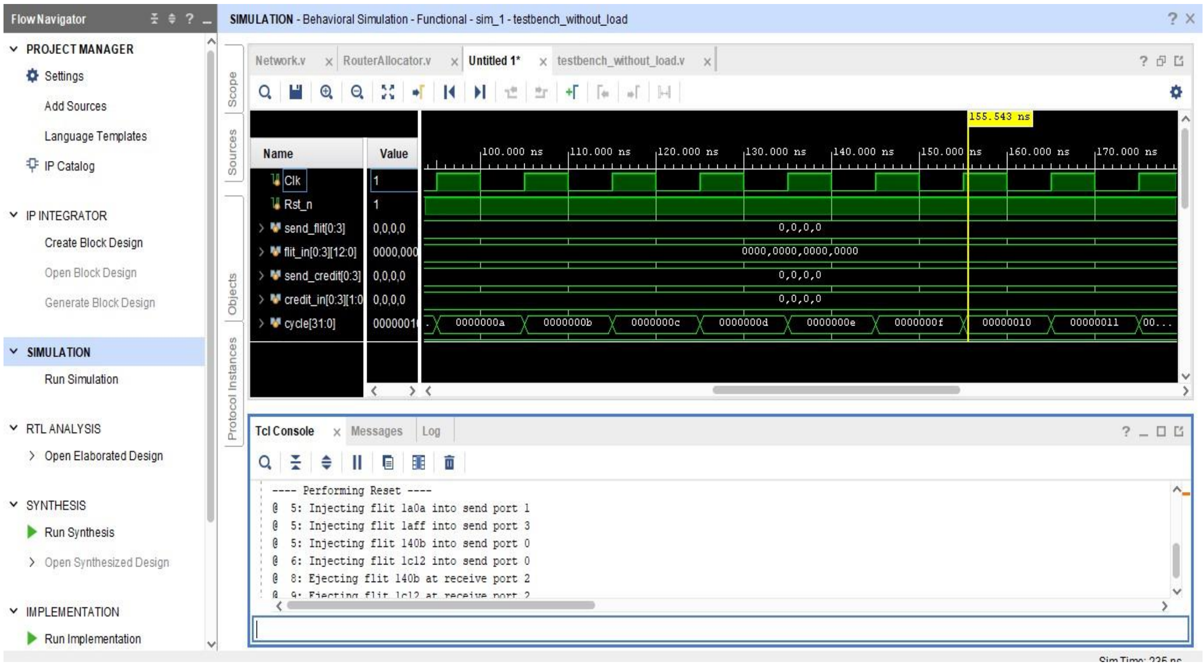
In the without-load test, one router is responsible for sending data packets to another router with no competition from other packets of the same input channel. It is an ideal case, which validates the design and evaluates the latency efficiently. In the zero-load test, one node is only responsible to send packets to another node of a network. All packets have been set as single flit packets for making the simulation as easy to understand and simple. Six clock cycles will be taken by a packet when a flit is taken 2 clock cycles for traversal of data packets from one router to another router as it is operated in the flit level.

Table 2 shows the obtained results based on the tests for zero loads. It indicates that two clock cycles for a node are needed for the transmission of packets to a router directly. One more clock cycle is taken for traversing packets through the intermediate routers when it is there in between the source and destination. The intermediate router is traversed in three clock cycles as opposed to four clock cycles because a first clock cycle flit is injected into the router and stored in the flit buffer. In the second clock cycle, the flit leaves the router and moves in the same clock cycle to the next router. After that, it was kept in the next router's flit buffer. The flit ejects from an intermediate router in the third clock cycle, after which it will arrive at the destination router.

Table 2. Router Delay For Zero Load Network

Flit traversal direction (Source Router Id to Destination Router Id)	Number of Clock cycles to reach the destination router
R0 to R1	2(adjacent router)
R0 to R2	2
R0 to R3	3(one router in between)
R1 to R0	2
R1 to R2	3
R1 to R3	2
R2 to R0	2
R2 to R1	3
R2 to R3	2
R3 to R0	3
R3 to R1	2
R3 to R2	2

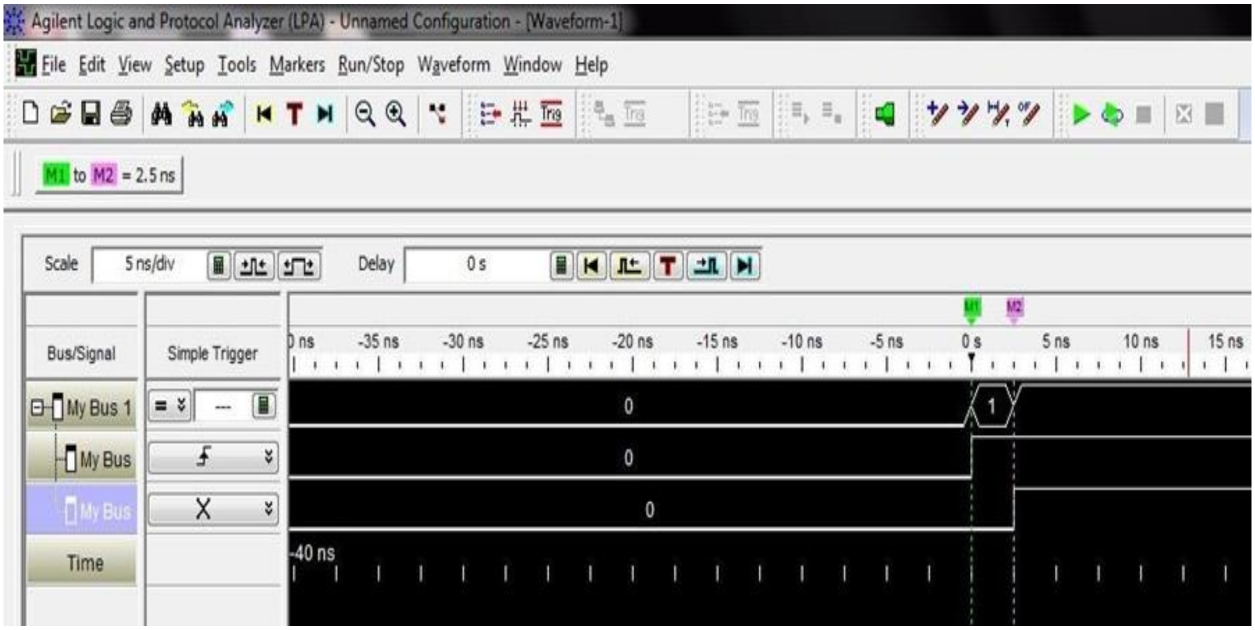
The waveform from the Fig. 10 shows without load transmission of flit in the network.



```
---- Performing Reset ----
@  5: Injecting flit 1a0a into send port 1
@  5: Injecting flit 1aff into send port 3
@  5: Injecting flit 140b into send port 0
@  6: Injecting flit 1c12 into send port 0
@  8: Ejecting flit 140b at receive port 2
@  9: Ejecting flit 1c12 at receive port 2
```

Figure 10. Simulation Results Of Without-Load Test

One intermediate router can be there between two distant routers as the design develops based on a 2X2 mesh topology. For a zero-load network, three clock cycles are the total number of clock cycles in the worst case. Based on the routing algorithm, the maximum number of clock cycles is increased when a network scales with more routers, but the same clock cycles are taken for each router. The least number of clock cycles is taken by the determined path based on the routing lookup table to reach from one to another router in the scaled-up network. The latency test is ported in the ZynqSoC device and interfaced with the Agilent logic analyzer. The corresponding logic analyzer results are depicted in Fig. 11.


Agilent Logic and Protocol Analyzer (LPA) - Unnamed Configuration - [Waveform-1]
File Edit View Setup Tools Markers Run/Stop Waveform Window Help
Bus/Signal
Simple Trigger
My Bus 1
My Bus
My Bus
Time
M1
M2
-40 ns
-35 ns
-30 ns
-25 ns
-20 ns
-15 ns
-10 ns
-5 ns
0 s
5 ns
10 ns
15 ns
0
1
0
0
-40 ns

M1 to M2 = 2.5 ns

Scale 5 ns/div

Delay 0 s

Figure 11. Latency Given By Agilent Logic Analyzer (Without-Load Test)

5.2.2. With-Load Test

The load test includes the transmission of packets to the same output channel at the same time when it is wanted by two or more flit buffers. If multiple channels try to send flits to the same output channel, this test is performed to understand the router's behavior. Based on the selected request for the output port, this priority is considered.

In the case of multiple transmissions, arbitration is performed, which could be occurred if a router sends a packet to the same output port or router from more than one input channel. The arbitration logic decides the priority for the input ports. The output port assigns to the selected one and the remaining requests compete for the output port. A single flit sends by all routers for the same destination router based on the written test benches in this test. At the destination router, the flit is read while recording and comparing the number of clock cycles to reach the destination router for a flit. Two clock cycles are taken for a router if it is connected directly to a zero-load network while it is taken three clock cycles for an intermediate router. The competition is there for getting an output port with a load case when more than one input channel has been trying out for sending the flits via the same output port. In this scenario, the number of clock cycles increases as shown in Table 3.

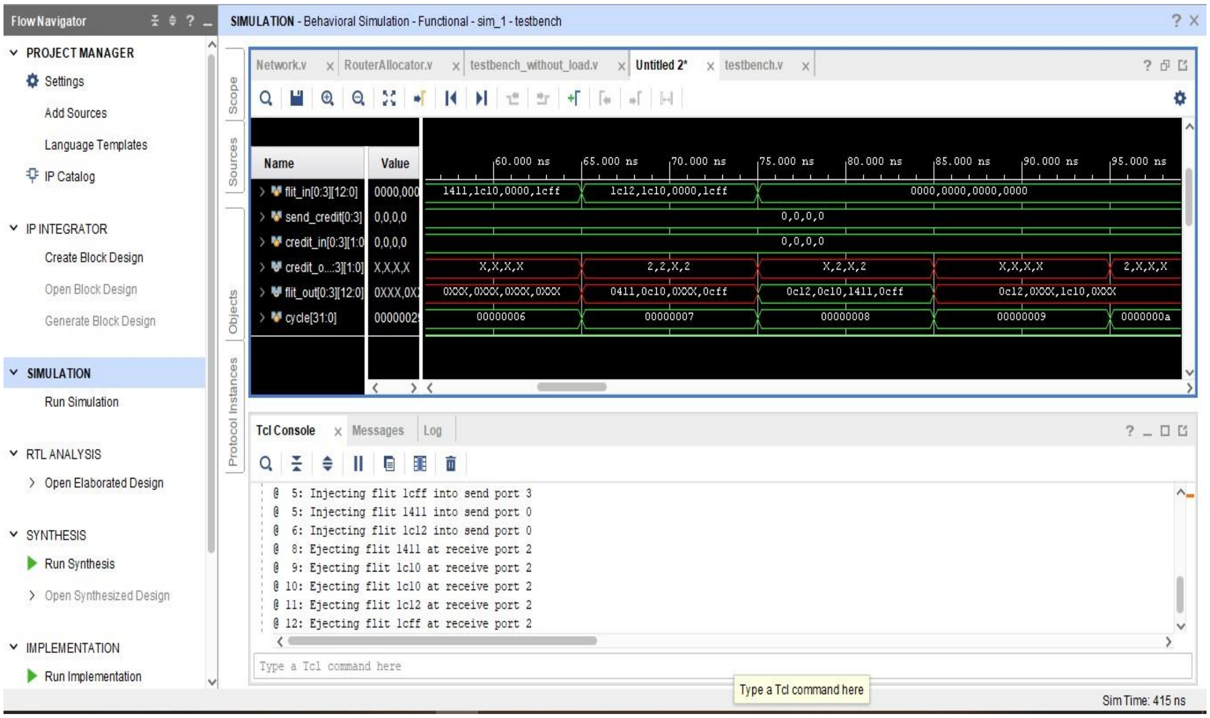
Table 3. Router Delay From All The Inputs With Concurrent Input

Flit traversal direction (Source Router Id to Destination Router Id)	**Number of Clock cycles to reach the destination router**
R0 to R1	2
R2 to R1	4
R3 to R1	3
R1 to R0	4
R2 to R0	2
R3 to R0	3

R0 to R2	2
R1 to R2	3
R3 to R2	4
R0 to R3	3
R1 to R3	2
R2 to R3	4

Due to the fact of not an ideal situation, more than one input channel has been tried to access the output port to go to the destination router. Based on the priority setting and allowing of one transmission only from a particular output port, the collision is avoided in this case.

The clock cycles can increase up to four from the analysis of the above tests. In the multi-flit packets or flit buffer, more than one flit being stored is not handled. Many test cases are there and not possible to perform all tests due to lack of time. The tests are conducted that could be considered as a reference. In case of network disruptions and stored incoming flits in the flit buffer, more delays can occur. Four more flits are already stored previously and the flit needs to wait for another four clock cycles for turning out to be the ejection from a buffer when a flit is stored in the buffer. By comparing with a single flit packet, the packet will consider five times more clock cycles for reaching the destination when a packet has five flits. The remaining packet flits allocate to the channel automatically until receiving the tail flit after a packet's head flit wins the arbitration. Fig.12 illustrates how flits travel a network and arrive at a destination router at various clock cycles by targeting a certain router from other routers.



```
@  5: Injecting flit 1cff into send port 3
@  5: Injecting flit 1411 into send port 0
@  6: Injecting flit 1c12 into send port 0
@  8: Ejecting flit 1411 at receive port 2
@  9: Ejecting flit 1c10 at receive port 2
@ 10: Ejecting flit 1c10 at receive port 2
@ 11: Ejecting flit 1c12 at receive port 2
@ 12: Ejecting flit 1cff at receive port 2
```

Figure 12. Simulation Waveform Of With-Load Test With Subsequent Console Logging

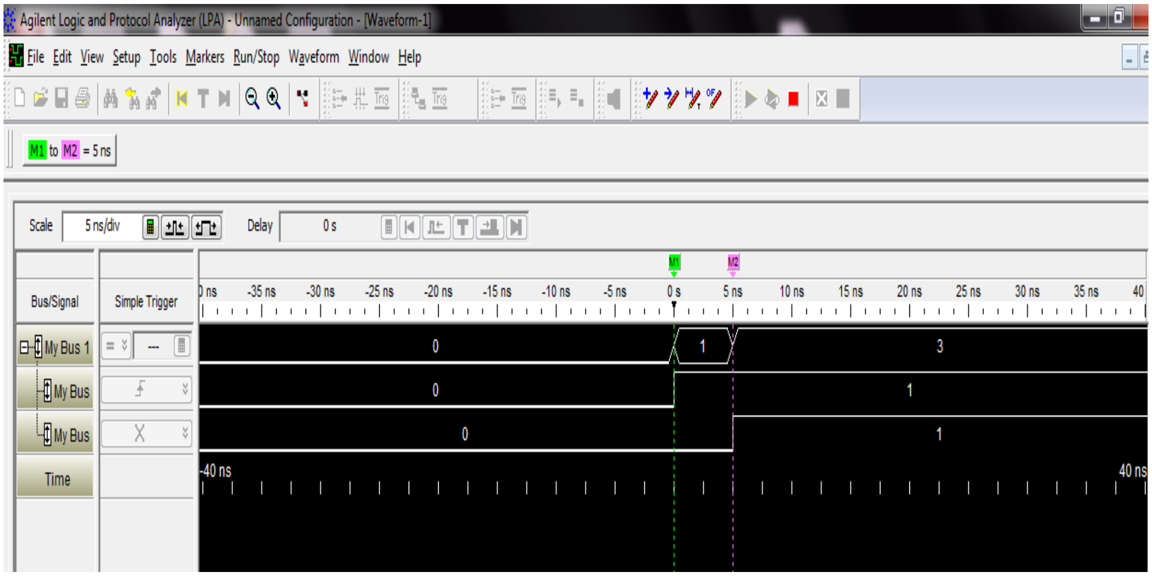

Agilent Logic and Protocol Analyzer (LPA) - Unnamed Configuration - [Waveform-1]
File Edit View Setup Tools Markers Run/Stop Waveform Window Help
M1 to M2 = 5 ns
Scale 5 ns/div
Delay 0 s
Bus/Signal
Simple Trigger
My Bus 1
My Bus
My Bus
Time
M1
M2
0 ns -35 ns -30 ns -25 ns -20 ns -15 ns -10 ns -5 ns 0 s 5 ns 10 ns 15 ns 20 ns 25 ns 30 ns 35 ns 40
0
1
3
0
1
0
1
-40 ns
40 ns

Fig.Ure 13. Latency Given By Agilent Logic Analyzer (With-Load Test)

The latency given by the logic analyzer with load test is shown in Fig.13. Three distinct flits from routers 0, 1, and 3 are headed in three separate directions towards router 2 which is injected at clock cycle 6, according to the waveform and log file shown above. The flits are read using various clock cycles from the same output port and injected at clock cycle 6. The ideal time would be two clock cycles, however, there is competition for access to the output port. Due to rivalry for an output port as all flits travel through the same channel, three and four clock cycles are required to transmit a flit to routers 2 and 3, respectively.

The proposed method is compared with existing work reported in [20], [22]. In this research work, the authors used a 4X4 mesh network is used for the experimental setup. For real-time implementation, the proposed model is ported on a Spartan 6 XC6SLX9 FPGA device. The latency is analyzed by sending several packets having flit size 1 to 4 from the source R1-1 to all other destination nodes. The latency will vary based on the routing path for each test scenario and latency tested with and without congestion. The latency tested in the FPGA device is analyzed in Chip scope pro. The comparison results are reported in table 4.

Table 4. Comparison Results

S.No	**Parameter**	**Results (latency in terms of clock cycles)**		
		Existing [20]	Existing [22]	Proposed
			Without Congestion	
1	R0 to R1	4	5.5	2
	R0 to R2	4	4	2
	R0 to R3	6	5.5	3
2	R1 to R0	4	4	2
	R1 to R2	6	5.5	3
	R1 to R3	4	4	2
3	R2 to R0	4	4	2
	R2 to R1	6	5.5	3
	R2 to R3	4	4	2
4	R3 to R0	6	5.5	3
	R3 to R1	4	4	2
	R3 to R2	4	4	2

		With Congestion		
1	R0 to R1	4	4	2
	R2 to R1	8	12.5	4
	R3 to R1	6	5.5	3
2	R1 to R0	8	12.5	4
	R2 to R0	4	4	2
	R3 to R0	6	5.5	3
3	R0 to R2	4	4	2
	R1 to R2	6	5.5	3
	R3 to R2	8	12.5	4
4	R0 to R3	6	5.5	3
	R1 to R3	4	4	2
	R2 to R3	8	12.5	4

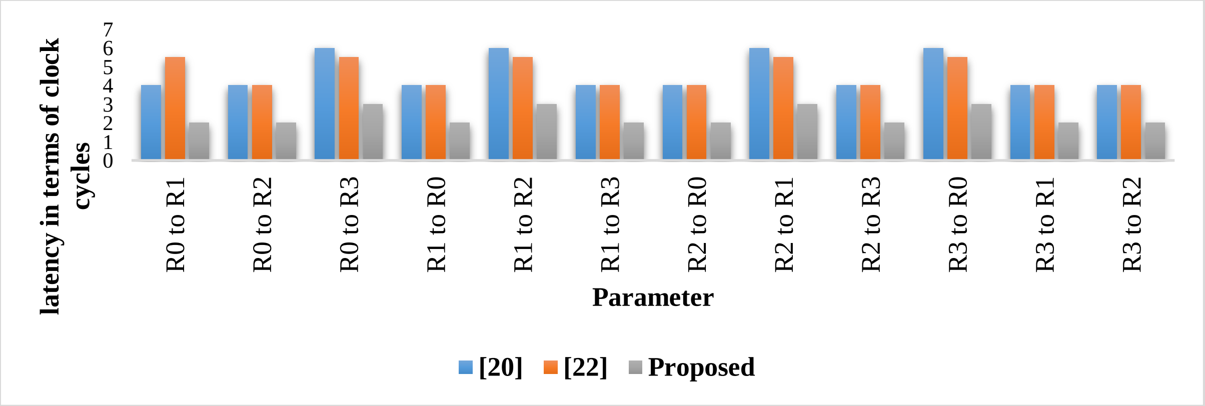


Figure14. Latency Of Proposed and Existing Methods (Without Congestion)

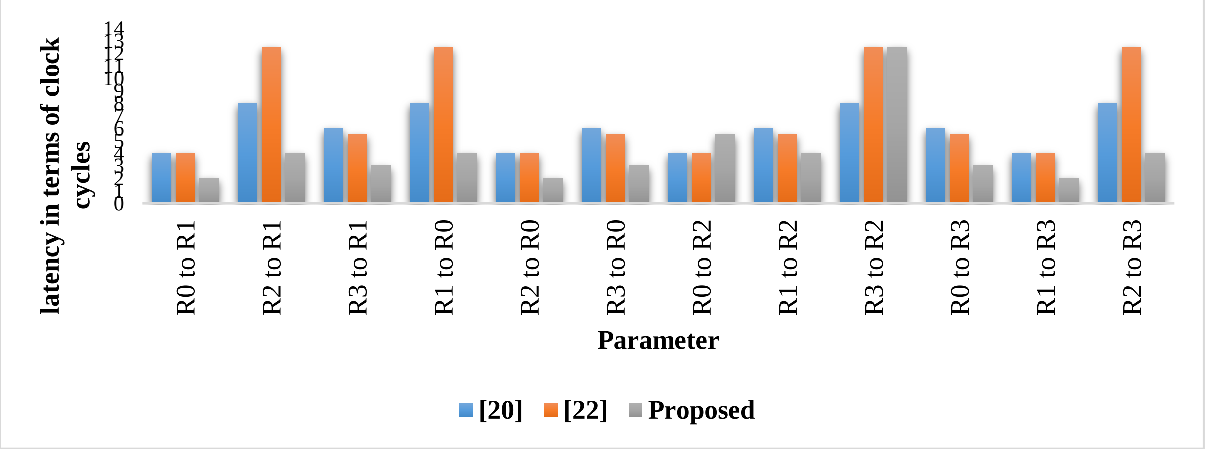


Figure 15. Latency Of Proposed And Existing Methods (With Congestion)

The proposed method is compared with the existing method [20], [22] with and without congestion. In both cases, the proposed method exhibits better performance in terms of latency. The 50% of latency is reduced when compared to existing works. The latency is calculated based on equation (1) and comparison latency results are reported in table 4 and depicted in fig 14-15.

6. Conclusion

This paper proposes a deadlock-free deterministic routing algorithm based on the Turn Model. Adopting different routing algorithms for odd and even columns, reduces the congestion of a single channel, thereby reducing latency by 30%. The proposed method is also compared with existing work and a 50% reduction is achieved in terms of latency. The area and power utilization for the proposed method are 69% (IO utilization) and 0.128W respectively. The simulation results show that compared with the XY routing algorithm, this algorithm obtains less delay in both load tests: zero load test and with load test. Compared with the minimum parity routing algorithm, the implementation of this scheme is simpler, does not require complex hardware logic, and has low hardware overhead.

References

[1] R. Shruthi, H. R. Shashidhara, and M. S. Deepthi, “Comprehensive Survey on Wireless Network on Chips”, In Proceedings of the International Conference on Paradigms of Communication, Computing and Data Sciences, pp. 203-218, Springer, Singapore, 2022.

[2] T. Patil, and A. Sandi, “Design and implementation of asynchronous NOC architecture with buffer-less router”, Materials Today: Proceedings, 49, pp.756-763, 2022.

[3] I. A.Alimi, R. K.Patel, O. Aboderin, A. M. Abdalla, R. A. Gbadamosi, N. J.Muga, and A. L. Teixeira, “Network-on-chip topologies: Potentials, technical challenges, recent advances and research direction, Network-on-Chip-Architecture, Optimization, and Design Explorations, 2021.

[4] R. Ahmed, H. Mostafa, and A. H. Khalil, “Design of a reconfigurable network-on-chip for next generation FPGAs using Dynamic Partial Reconfiguration”, Microelectronics Journal, 108, 2021.

[5] Tsai, Kun-Lin, Feipei Lai, Chien-Yu Pan, Di-Sheng Xiao, Hsiang-Jen Tan, and Hung-Chang Lee, “Design of low latency on-chip communication based on hybrid NoC architecture”, In Proceedings of the 8th IEEE International NEWCAS Conference, pp. 257-260. IEEE, 2010.

[6] L. Daoud, “Secure network-on-chip architectures for MPSoC: overview and challenges”, In 2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS), pp. 542-543, IEEE, 2018.

[7] B.Ahmad, and T.Arslan, “Dynamically Reconfigurable NoC for Reconfigurable MPSoC”, In Proceedings of the IEEE Custom Integrated Circuits Conference, 2005.

[8] Rajeev Kamal, PankajGoyal, and VikasNehra, “Network on Chip: Topologies, Routing, Implementation”, In International Journal of Advances in Science and Technology”, Vol. 4, No.1, 2012.

[9] Hui Zhao, Mahmut Kandemir, and Mary Jane Irwin, “Exploring Performance Power trade-offs in Providing Reliability for NoC Based MPSoC”, In IEEE 12th International Symposium on Quality Electronic Design, USA, 2011.

[10] 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), pp. 133-137. IEEE, 2021.

[11] R. F. Hassan, and R. L. Khaleel, “Hardware Implementation of NoC based MPSoC Prototype using FPGA”, International Journal of Applied Engineering Research, 13(7), 5443-5451, 2018.

[12] Milfont, Ronaldo, Paulo Cortez, Alan Pinheiro, Joao Ferreira, Jarbas Silveira, Rafael Mota, and Cesar Marcon, "Analysis of routing algorithms generation for irregular noc topologies”, In 2017 18th IEEE Latin American Test Symposium (LATS), pp. 1-5. IEEE, 2017.

[13] Alimi, A.Isiaka, K. Romil Patel, M. Oluyomi Aboderin, Abdelgader, A. Abdalla, Ramoni, Gbadamosi, J. Nelson, N. Muga, Armando, Pinto, and L. António Teixeira, “Network-on-chip topologies: Potentials, technical challenges, recent advances and research direction", In Network-on-Chip. IntechOpen, 2021.

[14] Kaleem, Muhammad, and Ismail Fauzi Bin Isnin, “A Survey on Network on Chip Routing Algorithms Criteria", In Advances on Smart and Soft Computing, pp. 455-466. Springer, Singapore, 2021.

[15] Last.Alireza Monemi, Jia Wei Tang, Maurizio Palesi, Muhammad, and N.Marsono, “ProNoC: A Low Latency Network on Chip based many core System on Chip Prototyping platform”. In Microprocessors and Microsystems, 54, pp. 60-74, 2017.

[16] Chawade, D. Shubhangi, A. Mahendra Gaikwad, and M. Rajendra Patrikar. “Review of XY routing algorithm for network-on-chip architecture", International Journal of Computer Applications 43, no. 21, pp.975-8887, 2012.

[17] Wang, Liang, Xiaohang Wang, and Terrence Mak, “Adaptive routing algorithms for lifetime reliability optimization in network-on-chip", IEEE Transactions on Computers 65, no. 9,2896-2902, 2015.

[18] Manzoor, Misbah, and RoohieNaaz Mir, “Prime Turn Model and First Last Turn Model: An Adaptive Deadlock Free Routing for Network-on-Chips”, Microprocessors and Microsystems (2022): 104454.

[19] Cai, Yuan, Dong Xiang, and Xiang Ji, “Deadlock-free adaptive 3D network-on-chips routing algorithm with repetitive turn concept", IET Communications 14, no. 11, 1784-1793, 2020.

[20] Khurshid Ahmad, Muhammad AtharJavedSethi, Rehmat Ullah, Imran Ahmed,Amjad Ullah, Naveed Jan, and Ghulam Mohammad Karami, “Congestion-Aware Routing Algorithm for NoC Using Data Packets”,Hindawi Wireless Communications and Mobile Computing, 2021.

[21] Karpovsky, Mark, Lev Levitin, and Mehmet Mustafa, “Optimal turn prohibition for deadlock prevention in networks with regular topologies”, IEEE Transactions on Control of Network Systems 1, no. 1, pp. 74-85, 2014.

[22] Hung K. Nguyen , Xuan-Tu Tran , A Novel Reconfigurable Router for QoS Guarantees in Real-time NoC-based MPSoCs, Journal of Systems Architecture (2019), doi:https://doi.org/10.1016/j.sysarc.2019.101664.