DESIGN AND ANALYSIS OF LOW-POWER ADAPTIVE ARBITRATION BASED NETWORK-ON-CHIP FOR EFFICIENT SOC COMMUNICATION

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

MASTER OF TECHNOLOGY

in

VLSI & EMBEDDED SYSTEMS DESIGN

by

ANGURU CHANDANA

(2K24/VLS/13)

Under the Supervision of

Prof. J Panda
DTU, Delhi



Department of Electronics & Communication Engineering

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Bawana Road, Delhi-110042, India

MAY, 2026



Department of Electronics and Communication Engineering
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, Anguru Chandana, Roll No. 2K24/VLS/13 hereby certify that the work which is being presented in the thesis entitled '**Design and Analysis of Low-Power Adaptive Arbitration Based Network-on-Chip Router for Efficient SoC Communication**' in partial fulfilment of the requirements for the award of the Degree of Master of Technology (VLSI and Embedded Systems), submitted in the Department of Electronics and Communication Engineering, Delhi Technological University, is an authentic record of my own work carried out during the period from January 2026 to May 2026 under the supervision of **Prof. J Panda.**

I have not submitted the matter presented in the thesis for the award of any other degree of this or any other Institute.

Chandana

Candidate's Signature

Anguru Chandana

(2K24/VLS/13)



Department of Electronics and Communication Engineering
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Bawana Road, Delhi-110042

CERTIFICATE BY THE SUPERVISOR

This is to certify that the work reported in this thesis entitled "**Design and Analysis of Low-Power Adaptive Arbitration Based Network-on-Chip Router for Efficient SoC Communication**" has been carried out by Ms. Anguru Chandana (2K24/VLS/13) under my supervision.

Supervisor's Signature

Prof. J Panda

ECE, DTU, Delhi



Department of Electronics and Communication Engineering
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Bawana Road, Delhi-110042

PLAGIARISM VERIFIFCATION

Title of the Thesis: "**Design and Analysis of Low-Power Adaptive Arbitration Based Network-on-Chip Router for Efficient SoC Communication**"

Total Pages: 51

Name of the scholar: Anguru Chandana

Supervisor: Prof. J Panda

Department of Electronics and Communication Engineering

This is to report that the above thesis was scanned for similarity detection. The process and outcome is given below:

Software used: **Turnitin** Similarity Index: 13% Total Word Count: 9691

Date: 27-05-2026

Chandana

Candidate's Signature

Signature of Supervisor

(Prof J. Panda)

ACKNOWLEDGEMENT

The successful completion of any task is incomplete and meaningless without giving due credit to the people who made it possible, without whom the project would not have been successful and would have existed only in theory. First and foremost, I am grateful to Prof. Neeta Pandey, H.O.D., Department of Electronics and Communication Engineering, Delhi Technological University, and all other faculty members of our department for their constant guidance, motivation, sincere support, and encouragement throughout this project work. I owe a lot of thanks to our supervisor, Prof. J Panda, Department of Electronics and Communication Engineering, Delhi Technological University, for constantly motivating and guiding us in carrying out this project successfully. His valuable guidance, support, and encouragement helped us undertake this challenging work effectively, and he was always available whenever assistance or guidance was required.

I would also like to take this opportunity to express my sincere thanks and gratitude to everyone who directly or indirectly have given us their hand in this challenging task. I feel happy and joyful and content in expressing our vote of thanks to all those who have helped and guided me in presenting this project work for my Major project. Last, but never least, I thank our well-wishers and parents for always being with us, in every sense and constantly supporting me in every possible sense whenever possible.

Date: 29.05.2026

Place: Delhi

Anguru Chandana

2K24/VLS/13

ABSTRACT

Modern System-on-Chip (SoC) architectures integrate multiple processing elements such as processors, memory modules and hardware accelerators on a single chip. As the number of cores increases, efficient on-chip communication becomes essential for maintaining overall system performance. Traditional bus-based architectures suffer from bandwidth limitations, increased latency, and poor scalability, leading to the adoption of Network-on-Chip (NoC) architectures that enable scalable packet-based communication through interconnected routers and links. A critical component of a NoC router is the arbitration unit, which resolves contention when multiple input ports request access to the same output channel. Conventional arbitration techniques such as fixed priority and round robin provide limited adaptability to dynamic traffic conditions.

This work presents low power adaptive arbitration-based Network-on-Chip router architecture for improving communication efficiency in multi-core SoC systems. The design implements and evaluates Fixed Priority, Round Robin, Weighted Round Robin, and Adaptive Weighted Round Robin arbitration techniques. The adaptive scheme dynamically adjusts weights based on traffic conditions to ensure balanced bandwidth allocation and reduced congestion. The system integrates modules such as traffic pattern generator, FIFO buffers, request generation logic, arbitration unit, and crossbar switching fabric. In addition, clock-gating based power optimization techniques are incorporated to reduce switching activity and dynamic power consumption.

The architecture is implemented using Verilog and functionally verified using Vivado simulation. Experimental results demonstrate improved traffic handling efficiency, adaptive resource allocation, and significant reduction in dynamic power consumption while maintaining correct packet routing functionality. The proposed router architecture provides an efficient and scalable communication framework for modern multi-core SoC systems.

CONTENTS

CHAPTER 1 INTRODUCTION 11

1.1 MOTIVATION AND BACKGROUND 11

1.1.1 Evolution of System-on-Chip Communication Architectures 11

1.1.2 Emergence of Network-on-Chip Architectures 12

1.1.3 Role of Arbitration in NoC Routers 13

1.1.4 Need for Adaptive Arbitration Mechanisms 14

1.1.5 Need for Low Power NoC Design 14

1.2 PROBLEM STATEMENT 15

1.3 RESEARCH FOCOUS : Low-POWER ADAPTIVE ARBITRATION BASED NoC ROUTER 16

1.4 OBJECTIVES AND CONTRIBUTIONS 16

1.5 STRUCTURE OF THE THESIS 17

CHAPTER 2 LITERATURE REVIEW 19

CHAPTER 3 NETWORK-ON-CHIP ARCHITECTURE 22

3.1 INTRODUCTION TO NETWORK-ON-CHIP 22

3.2 FIVE-PORT NETWORK-ON-CHIP ROUTER ARCHITECTURE 22

3.3 INTERNAL ROUTER COMPONENTS 24

3.4 ARBITRATION IN NETWORK-ON-CHIP ROUTERS 25

CHAPTER 4 DESIGN METHODOLOGY 27

4.1 OVERVIEW OF THE PROPOSED SYSTEM 27

4.2 PROPOSED SYSTEM ARCHITECTURE 27

4.3 TRAFFIC PATTERN GENERATOR 29

4.4 FIFO BUFFER 30

4.5 REQUEST MATRIX GENERATION 30

4.6 ARBITRATION UNIT 31

4.7 ADAPTIVE WEIGHTED ROUND ROBIN ARBITER 32

4.8 CROSSBAR SWITCH DESIGN 33

4.9 FIVE-PORT NOC ROUTER OPERATION 34

4.10 CLOCK GATING BASED POWER OPTIMIZATION 34

CHAPTER 5 RESULTS AND ANALYSIS 36

5.1 SIMULATION ENVIRONMENT 36

5.2 FUNCTIONAL VERIFICATION OF SYSTEM MODULES 36

5.2.1 Traffic Pattern Generator (TPG) 37

5.2.2 FIFO Buffer Operation 37

5.2.3 Request Matrix Generation 38
5.3 EVALUATION OF ARBITRATION MECHANISMS 39
5.3.1 Fixed Priority Arbitration 39
5.3.2 Round Robin Arbitration 40
5.3.4 Adaptive Weighted Round Robin Arbitration 41
5.4 CROSSBAR SWITCH 42
5.5 NETWORK-ON-CHIP ROUTER OPERATION 43
5.6 POWER ANALYSIS 45
5.7 TIMING ANALYSIS AND RESOURCE UTILIZATION 46
5.8 RNOC IMPLEMENTATIONS. 47
CHAPTER 6 CONCLUSION AND FUTURE WORK 48
REFERENCES 50

LIST OF FIGURES

Figure No	Figure Description	Page No
1.1	Communication Architecture in System on Chip	9
1.2	NoC based communication Architecture	9
3.1	Typical Five-Port Network-on-Chip Router Architecture	19
3.2	Internal Architecture of a Network-on-Chip Router	20
4.1	Proposed Architecture	23
4.2	Packet structure from TPG	24
5.1	Simulated waveform of Traffic Pattern Generator	31
5.2	Simulated waveform of FIFO buffer	32
5.3	Simulated waveform of Request Matrix	32
5.4	Simulated waveform of fixed priority arbitrer	33
5.5	Simulated waveform of round robin arbitrer	34
5.6	Simulated waveform of Static Weighted Round Robin Arbitration	34
5.7	Simulated waveform of Adaptive Weighted Round Robin Arbitration	35
5.8	Simulated waveform of Crossbar switch	36
5.9	Vivado schematic of Network on Chip Router	37
5.10	Simulated waveform of Network-on-Chip Router Operation	37
5.11	Power report of Network-on-Chip Router without Optimization	38
5.12	Power report of Network-on-Chip Router with Optimization	38
5.13	Vivado Timing Analysis Report of Network-on-Chip Router	39
5.14	Vivado Resource Utilization Report of Network-on-Chip Router	40

LIST OF TABLES

TABLE NO.	TABLE DESCRIPTION	PAGE NO.
5.1	Power Comparison of Proposed Router Architecture	39
5.2	Timing Analysis of Proposed Router Architecture	39

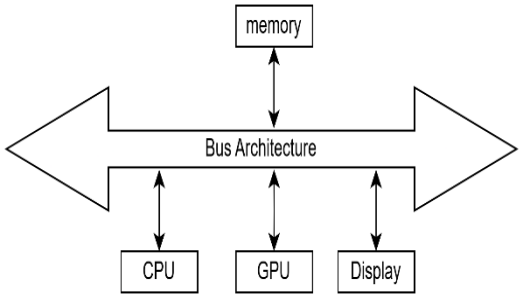
CHAPTER 1
INTRODUCTION

1.1 Motivation and Background

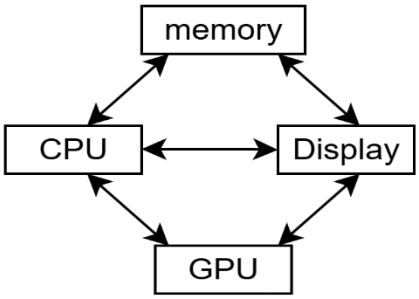
1.1.1 Evolution of System-on-Chip Communication Architectures

The continuous advancement of semiconductor technology has enabled the integration of a large number of functional components within a single System-on-Chip (SoC). Modern SoC platforms typically incorporate multiple processing elements including processors, memory controllers, digital signal processors, graphics units, and specialized hardware accelerators. These components must exchange large volumes of data efficiently in order to maintain high system performance.

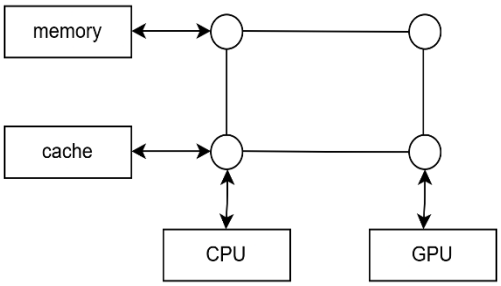
In earlier SoC designs, communication between modules was typically implemented using shared bus architectures. Bus-based interconnects were simple and effective for systems with a limited number of modules. However, as the complexity of integrated circuits continues to increase, bus architectures face significant scalability challenges. Since only one module can access the shared bus at a time, contention increases rapidly when multiple components attempt to communicate simultaneously. This results in higher communication latency and reduced bandwidth utilization. performance for modern multi-core SoC systems.



a) Bus Architecture



b) point to point



c)Switch Based

Fig.1.1 Communication Architecture in System on Chip

1.1.2 Emergence of Network-on-Chip Architectures

Network-on-Chip (NoC) architecture has emerged as an efficient communication framework for modern multi-core systems. Instead of relying on a single shared communication bus, NoC adopts a packet-switched communication model in which routers and communication links form an interconnected network across the chip. In a typical NoC system, each processing element is connected to a router that acts as an interface between the computing node and the communication network. Data is transmitted across the network in the form of packets, which travel from source nodes to destination nodes through intermediate routers. These routers forward packets based on routing decisions and network protocols

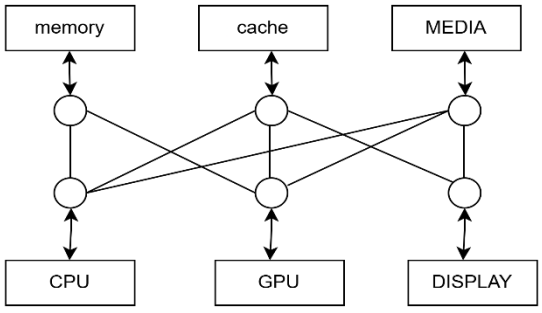


Fig.1.2 NoC based communication Architecture

The architecture of a NoC system can be characterized by several important design parameters including network topology, routing strategy, switching mechanism, flow control policy, buffering strategy, and arbitration techniques.

Structured topologies such as mesh, torus, or tree networks are commonly used to organize routers in large-scale NoC systems. A typical NoC router comprises of numerous functional modules such as input buffers, routing logic, arbitration units, crossbar switches and output ports. Incoming packets are held in input buffers until they are processed by the router. The routing algorithm selects the correct output path for each packet, depending on its destination address. The crossbar switch simply connects the desired input ports to the output ports so that the packet may be sent out across the network. When many packets are competing for the same output channel, the arbitration unit resolves the conflict.

Compared to traditional bus architectures, NoC provides several advantages including improved scalability, higher communication bandwidth, and better resource utilization. Multiple data transfers can occur simultaneously across different communication paths, significantly improving overall system throughput.

1.1.3 Role of Arbitration in NoC Routers

Within a NoC router, Contention happens when numerous input ports try to access the same output channel concurrently. The output channel is a shared resource and so the router needs to decide which packet should be transmitted first. It is the arbitration unit that does this decision making procedure.

To achieve efficient packet scheduling and fair resource allocation in the router, arbitration mechanisms are needed. The performance of the arbitration unit directly affects important network factors such as latency, throughput, fairness and bandwidth utilization.

Several arbitration techniques have been proposed for NoC routers. Fixed priority arbitration assigns fixed priority levels to input ports and serves requests from high priority ports first. This strategy is easy to implement but it may lead to starving of the lower priority ports. Round robin arbitration increases fairness by sequentially rotating the access to the ports. This idea is further extended to the weighted round robin arbitration where ports are assigned various weights for differentiated bandwidth distribution.

While these techniques provide varying levels of fairness and efficiency. However, typical arbitration schemes adopt static scheduling policies that are not adaptive to different traffic conditions in dynamic network environments.

1.1.4 Need for Adaptive Arbitration Mechanisms

Modern multi-core systems exhibit extremely dynamic communication patterns based on application burden and processing activity. Static arbitration techniques may not efficiently allocate network resources when traffic demands vary among different ports.

Adaptive arbitration techniques aim to address this problem by dynamically altering scheduling decisions based on the network traffic conditions. By adjusting arbitration parameters such as weights or priority, adaptive mechanisms can reallocate communication bandwidth more efficiently and enhance overall network performance. Motivated by these issues, this work focuses on the design and assessment of an adaptive arbitration based Network-on-Chip router architecture. The proposed system includes several arbitration approaches such as Fixed Priority, Round Robin, Weighted Round Robin and Adaptive Weighted Round Robin arbitration in a unified framework. The suggested approach compares these mechanisms in different traffic scenarios to improve the fairness, bandwidth utilization and overall communication efficiency in multi-core SoC systems.

1.1.5 Need for Low Power NoC Design

As the complexity of multi-core SoC systems continues to increase, power consumption has become an important challenge in Network-on-Chip (NoC) design. Continuous switching activity within router components such as FIFO buffers, arbitration units, routing logic, and crossbar switches contributes significantly to dynamic power consumption. During low traffic or idle conditions, unnecessary clock-driven switching further reduces energy efficiency and increases thermal complexity in large-scale NoC systems. Low-power optimization techniques such as clock gating can reduce unnecessary switching activity by disabling inactive modules during idle periods. In this work, clock-gating based optimization is integrated with adaptive arbitration techniques to improve communication efficiency while reducing dynamic

power consumption in the NoC router architecture.

1.2 Problem Statement

As the number of processing elements integrated within modern SoC platforms continues to increase, efficient on-chip communication becomes a critical design challenge. Network-on-Chip architectures provide a scalable communication framework. However, contention within routers remains a major issue affecting system performance.

When multiple packets arrive at a router and request the same output channel at the same time, the router has to decide which request should be granted access first. The conventional arbitration systems, such as fixed priority and round robin arbitration, provide easy scheduling algorithms, but may result in poor bandwidth usage and unfair resource distribution under dynamic traffic conditions.

Furthermore, many traditional arbitration techniques use static setups that cannot adapt to the changing communication demands of multiple input ports. This limits their efficacy in modern multi-core systems because traffic patterns vary dynamically based on application workloads. Thus, a need exists for adaptive arbitration methods that may dynamically allocate router resources depending on network traffic behavior while ensuring fairness and enhancing overall communication efficiency.

In addition to communication efficiency, power consumption has also become a major concern in NoC-based architectures. The dynamic power consumption is mostly due to the continuous switching activity in the router components such as FIFO buffers, arbitration units, routing logic and crossbar switches. Clock-driven activity under low-traffic or idle activity significantly decreases the overall energy efficiency and increases the thermal complexity for large-scale NoC systems.

Therefore, there is a need for a low-power adaptive arbitration mechanism capable of dynamically allocating router resources based on varying traffic conditions while reducing unnecessary switching activity and maintaining fair bandwidth distribution within the NoC router architecture.

1.3 Research Focus: Low-Power Adaptive Arbitration Based NoC Router

This research focuses on the design and analysis of a low-power adaptive arbitration based Network-on-Chip (NoC) router architecture for improving communication efficiency in multi-core SoC systems. The work explores various arbitration systems such as Fixed Priority Arbitration, Round Robin Arbitration, Static Weighted Round Robin Arbitration and Adaptive Weighted Round Robin Arbitration to analyse their behaviour under diverse traffic scenarios. Out all these strategies, adaptive arbitration is investigated to be an efficient method to dynamically allocate bandwidth and improve fairness in contention among numerous input ports.

In addition to communication efficiency, the research also aims to minimize dynamic power usage. The continuous switching activity of router components such as FIFO buffers, arbitration units and switching fabric adds significantly to the power dissipation. under order to overcome this problem, clock-gating based optimization strategies are introduced to reduce wasteful switching activity under low traffic and idle state.

The proposed router architecture integrates modules such as traffic pattern generator, FIFO buffers, request generation logic, arbitration units, and crossbar switching fabric within a unified framework. The complete design is implemented using Verilog HDL and functionally verified using Vivado simulation tools to evaluate communication performance, fairness, bandwidth allocation, and power optimization under different traffic scenarios.

1.4 Objectives and Contributions

The primary objective of this project is to design and analyse a low-power adaptive arbitration based Network-on-Chip (NoC) router architecture for improving communication efficiency in multi-core SoC systems. The work mainly focuses on studying different arbitration mechanisms and analysing their performance under varying traffic conditions. The specific objectives and contributions of this work are as follows:

- Designing a modular NoC router architecture capable of supporting multiple arbitration mechanisms.
- Implementing Multiple arbitration methods including Fixed Priority Arbitration, Round Robin Arbitration, Static Weighted Round Robin Arbitration and Adaptive Weighted Round Robin Arbitration.
- Developing modules like traffic pattern generator, FIFO buffers, request generation logic, arbitration unit, and crossbar switching fabric for packet routing operations.
- Investigating the effectiveness of adaptive arbitration techniques in improving fairness, bandwidth utilization, and overall network performance
- Using clock-gating based optimization strategies to cut down on superfluous switching activity and dynamic power consumption..
- Implementing and functionally verifying the complete router architecture using Verilog HDL and Vivado simulation tools.
- Evaluating and comparing the performance of different arbitration mechanisms in terms of fairness, resource allocation efficiency, communication performance, and power consumption.
- Analyzing the impact of low-power optimization techniques on dynamic power reduction while maintaining correct packet routing functionality.

1.5 Structure of the Thesis

The thesis is structured into six chapters:

• **Chapter 1:** Introduction

• **Chapter 2:** Literature Review

• **Chapter 3:** Network-on-Chip Architecture

• **Chapter 4:** Design Methodology

• **Chapter 5:** Results and Analysis

• **Chapter 6:** Conclusion and Future Work

CHAPTER 2
LITERATURE REVIEW

Bjerregaard and Mahadevan [1] presented a detailed survey on Network-on-Chip (NoC) architectures and discussed the limitations of conventional bus-based communication systems in multi-core System-on-Chip (SoC) designs. The study explained that as the number of processing elements increases, traditional shared-bus architectures suffer from congestion, higher latency, and limited scalability. To overcome these problems, NoC architectures were introduced to provide scalable packet-based communication through interconnected routers and communication links. The survey also analysed different routing algorithms, switching techniques, buffering methods, and arbitration schemes that contribute to efficient on-chip communication.

Similarly, Agarwal [2] presented a survey on Network-on-Chip architectures and presented a comprehensive survey of Network-on-Chip architectures and their applications in modern multi-core System-on-Chip designs. The work investigates various router topologies, routing algorithms, arbitration methods and switching approaches adopted in NoC systems. The survey underscores the significance of efficient router design and adaptive arbitration mechanisms in enhancing latency, throughput, and overall communication efficiency in large-scale multi-core computers.

Based on these foundational studies, researchers focused on improving the architecture of routers and packet scheduling methods to better the performance of communication. Swapna et al. [3] proposed a five-port router architecture for NoC communication systems. The router architecture contained FIFO buffers, registers, demultiplexers, and a scheduling unit that controlled the packet transfer between neighbouring nodes. A round-robin arbitration system was implemented to fairly share output channels among various input requests. The proposed design improved packet routing efficiency and ensured balanced resource allocation while preventing starvation during communication. Although the architecture achieved efficient packet transfer, the

arbitration process followed a fixed scheduling mechanism that could not dynamically adapt to varying traffic conditions inside the network.

To analyse communication behaviour and router performance under different traffic scenarios, Jiang et al. [4] introduced a detailed and flexible cycle-accurate NoC router microarchitecture for many-core systems. In their work, they implemented buffering mechanisms, arbitration procedures, and crossbar switching structures to analyse packet movement, network congestion, and throughput characteristics of NoC designs. The proposed approach enables precise investigation of latency and scalability in large scale processing systems. The study established an effective platform for NoC performance evaluation. However, the study was focused mostly on the communication analysis rather than the adaptive traffic management in the router arbitration units.

As the demand for high speed embedded applications increase, lowering communication latency has become an essential research goal in NoC router design. Pratap and Charan [5] suggested a low-latency five-port NoC router architecture for embedded devices. The router had FIFO buffers, routing logic, arbitration units, and crossbar switching structures for efficient packet forwarding between ports. The architecture used deterministic XY routing with look-ahead routing techniques for faster path computation and packet transfer to reduce the routing delay. Experimental results indicated improved packet transmission latency and improved communication efficiency in mesh based NoC designs. However, the router still uses traditional arbitration procedures with low adaptation to dynamic traffic conditions.

In addition to latency reduction, synchronization overhead also emerged as a critical barrier in complicated multi-core SoC systems. Traditional synchronous NoC systems rely on a global clock, which increases the complexity of clock distribution and impacts scalability as the system size expands. To overcome this problem, Saranya and Rathnamala Rao [6] suggested an asynchronous NoC router design for Globally Asynchronous Locally Synchronous (GALS) systems. The suggested system does not require a global clock, instead it uses handshake-based communication between router

modules for packet transport. The design relies on dependable communication, synchronization management and effective buffering mechanisms to provide steady communication between asynchronous components. This solution reduced clock distribution overhead and increased scalability in large SoC systems.

Moreover, communication efficiency enhancements were achieved by incorporation of common communication protocols into NoC systems. Malladhi and Attimarad [7] suggested a unique NoC architecture based on AXI connectivity for SoC applications. The design was combined with the AXI communication protocols in the NoC architecture to lower the latency of communication and increase the throughput between processors and memory subsystems. The proposed design was largely targeted to optimize routing efficiency and communication bandwidth while being compatible with conventional AXI-based communication interfaces typically adopted in recent SoC platforms. Experimental evaluation showed higher data transfer efficiency and better latency performance than conventional connection topologies.

Debbarma et al. [8] suggested an energy-efficient NoC router architecture with a power-gated round-robin arbiter for reducing the power consumption of router activities. The design applied fine-grained power gating techniques to control switching activities and reduce leakage power during packet transmissions. The proposed architecture aimed to enhance the power efficiency while keeping consistent communication performance and equitable resource allocation among competing input demands. Experimental results showed that the proposed NoC systems have lower power consumption with little influence on communication delay.

From the existing literature, it is observed that most NoC router architectures mainly focus on routing optimization, latency reduction, synchronization mechanisms, and throughput improvement. Several works employ conventional arbitration techniques such as fixed priority and round-robin allocation for resolving packet contention between input ports. However, these arbitration schemes provide limited adaptability under varying network traffic conditions, which may result in congestion, inefficient bandwidth utilization, and communication delay in heavily loaded networks. In

addition, many existing architectures do not simultaneously address adaptive traffic management and power optimization within the arbitration process. Therefore, there is a need for an adaptive arbitration-based NoC router architecture capable of dynamically managing traffic allocation while reducing switching activity and improving communication efficiency in modern multi-core SoC systems.

CHAPTER 3

NETWORK-ON-CHIP ARCHITECTURE

3.1 Introduction to Network-on-Chip

Network-on-Chip (NoC) is an advanced communication architecture designed to improve data transfer efficiency in modern System-on-Chip (SoC) platforms. As the number of processing elements integrated within a chip increases, traditional communication mechanisms such as shared buses become ineffective due to bandwidth restrictions and increased communication delay. To overcome these restrictions, Network-on-Chip (NoC) proposes packet-based communication between separate components using routers and communication lines.

In a conventional NoC system, processing elements communicate with each other through interconnected routers arranged in a specific topology such as mesh, torus, or ring. Each router receives packets from neighboring nodes, determines the appropriate output direction using routing logic. The packet is then forwarded to the next router until it reaches the destination node. This distributed communication scheme augments the scalability, minimizes congestion and enables concurrent transmission of data between multiple nodes.

Among several NoC topologies, mesh-based architectures are frequently used due to their regular structure and scalability. For such systems, network components such as buffers, arbitration units, and switching fabrics present in routers are essential factors that determine the communication efficiency, latency and power consumption of the network.

3.2 Five-Port Network-on-Chip Router Architecture

The Network-on-Chip router is the fundamental component responsible for transferring packets between nodes in the communication network. A typical router consists of multiple input and output ports that connect to neighboring routers and local processing elements. In most mesh-based NoC architectures, routers are designed with five ports corresponding to four directional connections and one local connection.

The five ports of the router include the **North, South, East, and West ports**, which connect the router to neighbouring routers through network links. The fifth port, known as the **Local port**, connects the router to the local processing element or core. These ports enable packet communication between processing elements located at different positions within the chip.

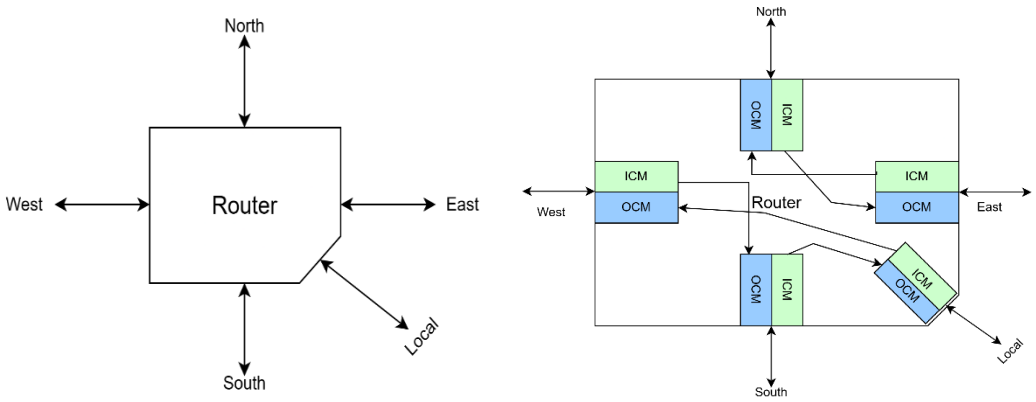


Fig 3.1 Typical Five-Port Network-on-Chip Router Architecture

Each port in the router contains functional modules responsible for managing packet transmission. The **Input Channel Module (ICM)** receives incoming packets from neighbouring routers through the network links. The ICM temporarily stores incoming packets and prepares them for routing decisions within the router. The **Output Channel Module (OCM)** forwards the packets from the router to the next router or destination node on the matching output connection. When a packet arrives at a router, the router first determines the correct output direction according to the packet's destination address. The router has an inbuilt switching structure to create a connection between the input port and the associated output port once the desired input port is selected. This link allows the packet to be routed to the next router on the network path. The internal control logic controls the switching process to enable efficient packet forwarding and to avoid conflicts when numerous packets request the same output port.

The five-port router architecture enables the packets to be routed in numerous directions concurrently, which helps in effective communication in mesh based

Network-on-Chip systems. This distributed routing feature provides a major improvement in network scalability and communication bandwidth over traditional bus-based designs.

3.3 Internal Router Components

The internal design of a NoC router comprises of various functional units that operate together to process incoming packets and to pass them to the correct output port. Such components are input buffers, arbitration control units, crossbar switches, and control signal logic.

The **input buffers** are responsible for temporarily storing packets arriving at the router from different input ports. These buffers store the incoming packets until the router decides on the appropriate output channel. Input buffering also assists in controlling network congestion by not losing packets when many packets arrive at the same time.

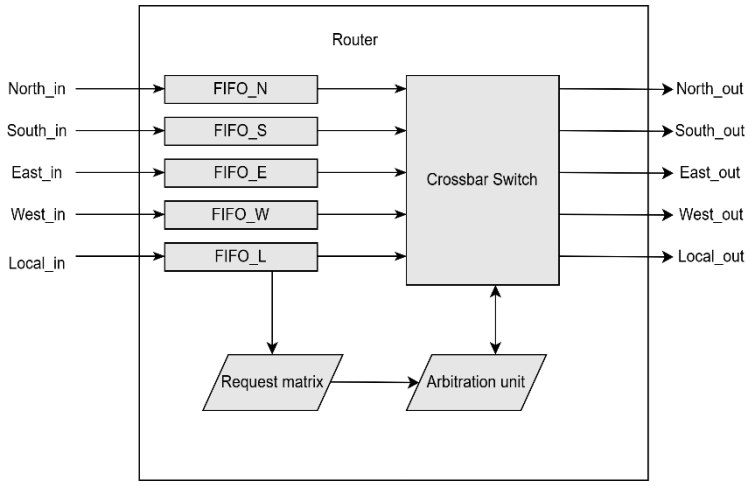


Fig 3.2 Internal Architecture of a Network-on-Chip Router

The **arbitration control unit** is responsible for resolving conflicts when multiple input ports request access to the same output port. Since several packets may attempt to use the same output channel at the same time, the arbiter determines which packet will be granted access based on a specific arbitration algorithm. Arbitration mechanisms such as fixed priority, round robin, weighted round robin, and adaptive arbitration are commonly used to manage resource allocation within the router.

The **crossbar switch** acts as the internal switching fabric of the router. It makes connections between the input and output ports according to the decision of the arbitration control unit. When an arbiter picks a request, the crossbar switch will build a direct connection between the selected input buffer and the appropriate output port so that the packet can be forwarded to the next router or processing element. The router also employs control signals to synchronize communication between different internal modules. These control signals regulate packet forwarding operations, control arbitration decisions, and configure the switching paths in the crossbar switch. The synchronization of these signals guarantees the effective transmission of packets without the conflict between several data flows.

The **request matrix** is responsible for generating request signals for the arbitration unit based on the destination information of incoming packets. It analyzes the packets stored in the input buffers and finds out the required output port for packet transmission. According to the routing requirement, the request matrix sends the matching requests to the arbitration unit. The arbitration unit solves the contention when several packets compete for the same output channel at the same time. This cooperation between the input buffers, request matrix and arbitration unit allows efficient packet routing inside the Network-on-Chip router architecture.

In some router architectures, a small temporary storage component such as a **cache or buffer** may be used to store packet headers or control information during routing operations. This temporary storage improves the router performance by eliminating delays due to congestion and packet blockage. The router combines the input buffers, the arbitration logic and the crossbar switching structures so that it can efficiently handle numerous packets simultaneously. These components together provide the dependable packet forwarding and the high communication throughput in the Network-on-Chip architecture.

3.4 Arbitration in Network-on-Chip Routers

Arbitration is an important mechanism used in Network-on-Chip routers t o prevent contention when numerous packets compete for the same output channel. In case of many input ports requesting access to a single output port at the same time, the router

has to decide which request must be granted. The arbitration unit performs the decision making process by using established scheduling methods.

Common arbitration techniques used in NoC routers include **Fixed Priority Arbitration**, **Round Robin Arbitration**, and **Weighted Round Robin Arbitration**. Fixed priority arbitration assigns priority levels to various input ports, allowing higher priority requests to be processed first. Round robin arbitration ensures fair scheduling by visiting the input ports in a cyclic order. Weighted round robin arbitration adds weights to each input port and allows some ports to be granted greater bandwidth according to their assigned weight values. These arbitration systems guarantee fairness and resource allocation but may not adapt efficiently to the dynamic traffic situations within the network. Therefore, adaptive arbitration mechanisms have been proposed to dynamically adjust scheduling parameters based on network traffic patterns. Adaptive arbitration improves bandwidth utilization, reduces communication latency, and enhances overall network performance.

CHAPTER 4

DESIGN METHODOLOGY

4.1 Overview of the Proposed System

The proposed work implements a low-power adaptive arbitration based NoC router architecture for efficient packet communication in multi-core SoC platforms. The design focuses on improving bandwidth allocation and communication fairness while reducing dynamic power consumption within the router. This router architecture consists of a Traffic Pattern Generator (TPG), FIFO buffers, request matrix, arbitration unit, crossbar switching structure, and output routing paths. Multiple arbitration mechanisms including Fixed Priority Arbitration, Round Robin Arbitration, Weighted Round Robin Arbitration, and Adaptive Weighted Round Robin Arbitration are implemented and analyzed under varying traffic conditions.

Among these techniques, Adaptive Weighted Round Robin (AWRR) arbitration is used as the primary scheduling mechanism because of its ability to dynamically allocate bandwidth according to traffic activity within the network. In addition to adaptive arbitration, clock-gating based optimization techniques are incorporated within selected router modules to reduce unnecessary switching activity and dynamic power consumption.

The overall operation of the router begins with packet generation through the Traffic Pattern Generator. Generated packets are temporarily stored in FIFO buffers before routing decisions are performed. The request matrix analyzes packet destination information and generates request signals for arbitration. The arbitration unit resolves contention between multiple requests, and the crossbar switch establishes communication paths between selected input and output ports for packet transmission within the network.

4.2 Proposed System Architecture

The proposed router architecture is based on a five-port mesh Network-on-Chip router consisting of North, South, East, West, and Local ports. Each input port is connected to an independent FIFO buffer for temporary packet storage before routing and

arbitration operations are performed.

The Traffic Pattern Generator produces pseudo-random packets for all router input ports. Generated packets are stored in FIFO buffers connected to the corresponding ports. The request matrix analyzes packet destination information and generates request signals for the arbitration unit.

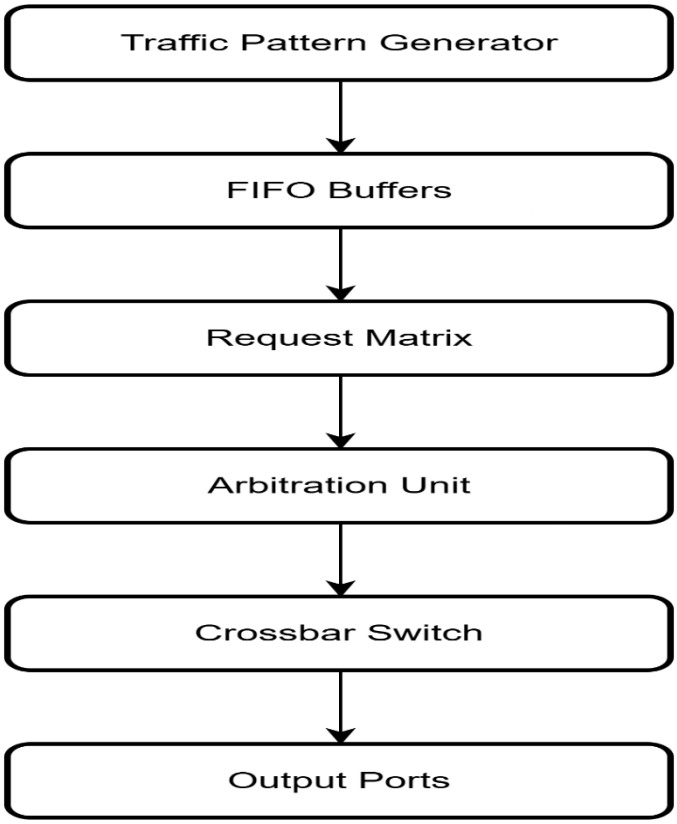


Fig.4.1 Proposed Architecture

When multiple packets request access to the same output port simultaneously, the arbitration unit resolves contention using the selected arbitration mechanism. Grant signals generated by the arbiter are forwarded to the crossbar switch, which establishes communication paths between selected input and output ports.

Clock gating logic is incorporated within the Traffic Pattern Generator, FIFO buffers, and arbitration unit to reduce unnecessary clock activity during idle conditions. This optimization minimizes the switching activity and enhances overall power efficiency of the router architecture.

4.3 Traffic Pattern Generator

The Traffic Pattern Generator (TPG) is responsible for generating packet data for different router input ports. The generated packets simulate communication traffic between nodes in the Network-on-Chip architecture and allow to evaluate the arbitration mechanisms under different traffic scenarios.

The suggested design generates packets utilizing Linear Feedback Shift Register (LFSR) based logic. The LFSR produces pseudo-random sequences which are utilized to generate different packet patterns throughout the router operation.

Each generated packet contains destination information and payload data. The packet format used in the proposed design is represented as:

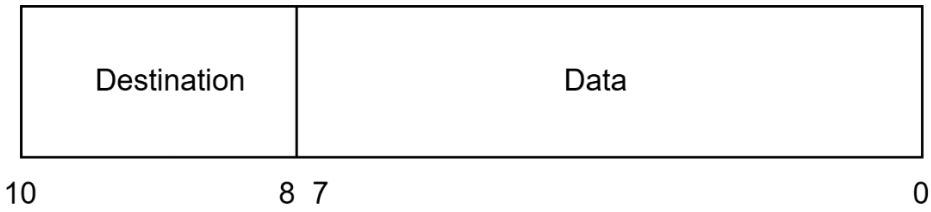
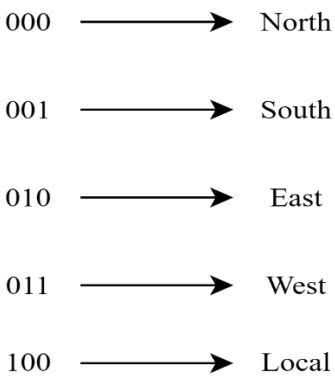


Fig.4.2 Packet structure from TPG

The destination bits generated by the Traffic Pattern Generator determine the required output direction during routing operations. Different destination values correspond to different router output ports within the NoC architecture.

The destination encoding used in the proposed design is given below:



During router operation, the request matrix decodes these destination bits and generates corresponding request signals for the arbitration unit. This enables packets generated by the Traffic Pattern Generator to be routed toward the appropriate output channel within the router architecture. The generated packets are forwarded to the FIFO buffers connected to the North, South, East, West, and Local input ports. Since the generated traffic patterns continuously vary during operation, the router can be evaluated under different network conditions. By generating dynamic traffic patterns, the system can evaluate the performance of different arbitration algorithms under varying network conditions.

4.4 FIFO Buffer

The FIFO (First-In First-Out) buffer is used to temporarily hold incoming packet data until it's processed by the router. FIFO buffers send packets in the order they are received. It is also vital for addressing network congestion and packet loss when there are many packets arriving at once. The proposed router architecture has distinct FIFO buffers for North, South, East, West, and Local ports. Each FIFO stores packet data until the required output channel becomes available. The FIFO operation is controlled using write enable and read enable signals. When a request is generated by the traffic generator, the corresponding packet data is stored in the FIFO buffer until the arbitration unit grants access to the output channel. Once the request is granted, the packet is forwarded to the next stage of the router pipeline.

4.5 Request Matrix Generation

The request matrix is responsible for generating request signals for the arbitration unit according to the destination information of incoming packets. It acts as an intermediate control block between the FIFO buffers and the arbitration logic. The request matrix analyzes packets stored within the FIFO buffers and identifies the required output port using the destination bits present in the packet header. Based on the routing requirement, the request matrix generates corresponding request signals for the arbitration unit.

For example, if a packet stored in the North FIFO requires transmission toward the

East output port, the request matrix generates a North-to-East request signal. Similarly, request signals are generated for all valid routing combinations between input and output ports.The generated request signals are forwarded to the arbitration unit, where contention between multiple requests is resolved. The request matrix therefore enables efficient communication between routing logic and arbitration control during packet transmission.

4.6 Arbitration Unit

The arbitration unit is responsible for resolving contention between multiple input ports requesting access to the same output channel. When several packets request the same output port simultaneously, the arbiter determines which request should be granted access. In this work, several arbitration methods are implemented and analysed including:

- Fixed Priority Arbitration
- Round Robin Arbitration
- Weighted Round Robin Arbitration
- Adaptive Weighted Round Robin Arbitration

Fixed Priority arbitration Input ports have pre-defined priority levels. This is easy to implement, but lower priority requests may starve under heavy demand. Round Robin arbitration enhances fairness by cyclically distributing output channels among competing requests in round-robin fashion. Each input port gets a chance to access the output channel during arbitration. Weighted Round Robin arbitration significantly optimizes the bandwidth distribution by assigning various weight values to the individual input ports. Communication needs can be used to provide ports with more access chances based on their weight values.

The arbitration unit generates grant signals corresponding to the selected request. These grant signals are forwarded to the crossbar switch for packet transmission. Each arbitration technique uses a different scheduling strategy to allocate router resources

among competing requests. The performance of these algorithms is evaluated based on fairness, bandwidth utilization, and communication efficiency.

4.7 Adaptive Weighted Round Robin Arbiter

The Adaptive Weighted Round Robin (AWRR) arbiter is the primary contribution of the proposed system. Unlike classic arbitration techniques, the adaptive arbiter dynamically adjusts weight levels according to network traffic conditions. When a request is identified at the designated port, the arbiter gives access to this port and decreases the credit value connected with it. When the credit approaches zero, the arbiter reloads the credit counter with the provided weight value and advances the pointer to the next port. The weight values are updated according to the activity of each input port via the adaptive method. If a port keeps issuing requests, its weight value is increased up to a predetermined maximum limit. And for port inactivity, its weight value would be steadily reduced to the minimum limit. This adaptive functionality enables the arbitration unit to allocate bandwidth more efficiently based on traffic demand.

The request matrix generates request signals and the arbiter assesses the request for each output port. The arbiter has a pointer that points to the current port being considered for packet transmission. A weight value is assigned to each port which indicates the priority of the port. The arbiter also maintains a credit counter that counts the number of times a particular port can be given access, based on its allocated weight.

If the credit counter is not 0, the same port can continue to send packets to the specified weight. When the credit counter becomes zero, the arbiter reloads the counter with the supplied weight value and rotates the pointer to the next input port.
The adaptive nature of arbiter dynamically changes the weight values according to network traffic activity:

- Ports with continuous constant requests are given higher weight values within the restrictions.
- Ports remaining inactive gradually receive reduced weight values.
- Weight values are maintained within minimum and maximum threshold limits.

To improve fairness and prevent starvation, the proposed arbitration mechanism also incorporates age-based priority handling. Each request is associated with an age value that increases while the packet remains unserved within the router. If a request waits long enough its effective priority is steadily increased regardless of its weight value. This approach ensures that lower priority packets are finally allowed to use the output channel even when there is a heavy traffic load.

During arbitration, the arbiter assesses the availability of requests, the position of pointers, the credit information, the adaptive weight values, and the packet age before picking the suitable request for transmission. Once the arbitration is done, the arbiter generates grant signals matching to the selected input request. The crossbar switch then uses these grant signals to construct communication routes between input and output ports.

4.8 Crossbar Switch Design

The crossbar switch provides the internal switching fabric of the router design. It sets up communication channels between selected input ports and output ports based on arbitration judgments.

The crossbar receives grant signals generated by the arbitration unit and forwards packet data from the selected FIFO buffer toward the required output channel. Each output port independently selects packet data according to the corresponding grant signal.

The proposed router architecture implements a five-port crossbar structure supporting communication between the North, South, East, West, and Local ports. The crossbar enables simultaneous packet transmission through different communication paths, thereby improving throughput and communication efficiency within the NoC

architecture.

4.9 Five-Port NoC Router Operation

The five-port router receives packet requests from multiple input ports through LFSR-based Traffic Pattern Generator. The generated packets contain destination information and data required for routing operations.

Generated packets are forwarded to the corresponding FIFO buffers connected to the North, South, East, West, and Local input ports. The FIFO buffers temporarily store incoming packets until the required output channel becomes available. The request matrix decodes the packet destination information and issues request signals to the relevant output ports. Those request signals are forwarded to the Adaptive Weighted Round Robin arbitration unit.

If numerous packets contend for the same output port, the arbitration unit resolves the congestion by applying adaptive scheduling logic. So, the arbiter checks the availability of the requests, the weight values, the credit counters and the pointer positions and then selects the suitable request.

After arbitration is complete, grant signals are created and sent to the crossbar switch. The crossbar makes the communication connections between some input and output ports and the packets can be sent to the next router or the destination processing equipment.

4.10 Clock Gating Based Power Optimization

Dynamic power consumption in Network-on-Chip routers is primarily caused by continuous switching activity within sequential logic elements and clock distribution networks. During idle network conditions clock signals are still toggling internal registers and control circuitry leading to excessive power dissipation.

The suggested router architecture includes clock gating based optimization approaches to reduce dynamic power consumption. Clock gating stops the clock propagation to unused modules while keeping the right functionality in the active packets being transmitted.

The system clock is combined with module enable signals to form the gated clock. Whenever module activity is absent, the gated clock remains disabled, thereby

preventing unnecessary switching transitions within sequential logic elements.Clock gating is incorporated within the Traffic Pattern Generator, FIFO Buffers, Adaptive Weighted Round Robin Arbiter

In Traffic Pattern Generator to reduce unnecessary switching activity during inactive conditions the gated clock remains active only during packet generation operations. During inactive conditions, the clock signal remains disabled, reducing switching activity within the LFSR and packet generation logic.

The FIFO clock remains active only during read or write operations. During idle conditions, the gated clock disables unnecessary switching activity within FIFO registers and control logic. The FIFO enable condition is generated using the read and write control signals. Whenever both operations remain inactive, the gated clock remains disabled, thereby reducing unnecessary signal transitions within the FIFO module.This optimization significantly reduces switching activity during low traffic conditions while maintaining correct packet storage and transmission functionality.

Similarly, in the Adaptive Weighted Round Robin arbiter, the gated clock remains active only when request signals are present. During idle conditions, switching activity within counters, pointer registers, and arbitration control logic is reduced. Dynamic power consumption is directly related to switching activity within digital circuits. By reducing unnecessary clock transitions in inactive modules, clock gating significantly reduces dynamic power consumption within the proposed router architecture.

CHAPTER 5
RESULTS AND ANALYSIS

5.1 Simulation Environment

The proposed low-power adaptive arbitration-based Network-on-Chip router architecture was implemented using Verilog Hardware Description Language. The design modules including LFSR-based Traffic Pattern Generator, FIFO buffers, Request Matrix, Adaptive Weighted Round Robin arbitration unit, and crossbar switch was developed and simulated using the vivado design environment. Simulation was carried out using a AMD vivado to verify the functional correctness of the router and arbitration mechanisms.

Functional verification was performed using simulation test benches developed for different router modules and arbitration mechanisms. Different arbitration techniques including Fixed Priority Arbitration, Round Robin Arbitration, Weighted Round Robin Arbitration, and Adaptive Weighted Round Robin Arbitration were evaluated under varying traffic conditions. Clock gating based optimization techniques were incorporated within selected router modules including the Traffic Pattern Generator, FIFO buffers, and arbitration unit to reduce unnecessary switching activity and dynamic power consumption. The simulation results were analyzed to evaluate packet transmission behaviour, arbitration efficiency, fairness, bandwidth allocation, and power optimization within the proposed router architecture.

5.2 Functional Verification of System Modules

Before evaluating the arbitration mechanisms, the individual modules used in the Network-on-Chip router architecture were verified through simulation. These modules include the Traffic Pattern Generator, FIFO buffers, Request Matrix, Adaptive Weighted Round Robin arbitration unit, and crossbar switch. Each module performs a specific function in the packet transmission pipeline.

5.2.1 Traffic Pattern Generator (TPG)

The simulation waveform shows the operation of the LFSR-based Traffic Pattern Generator used in the proposed router architecture. The signals pkt_n [10:0], pkt_s [10:0], pkt_e[10:0], pkt_w [10:0], and pkt_l [10:0] represent packets generated for the North, South, East, West, and Local ports respectively.

The packet values vary continuously during successive clock cycles, indicating pseudo-random packet generation using LFSR logic. Each packet contains destination bits and payload data required for routing operations within the router.The waveform shows that the generator continuously produces varying request patterns and data values, enabling the evaluation of arbitration algorithms under dynamic traffic conditions.

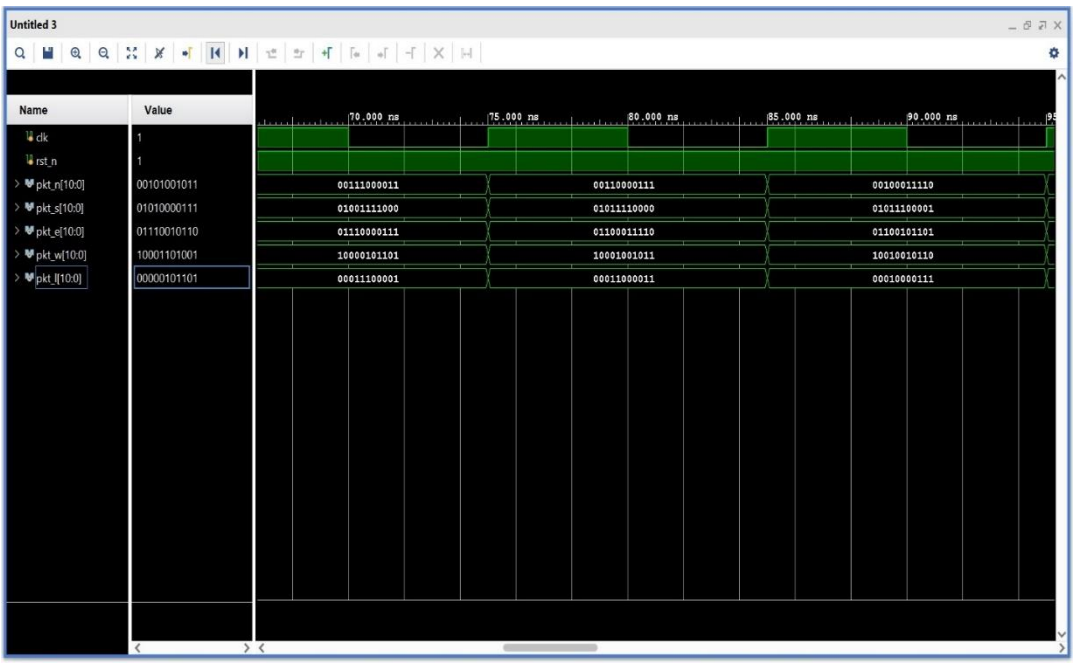


Fig.5.1 Simulated waveform of Traffic Pattern Generator

5.2.2 FIFO Buffer Operation

The below waveform shows the FIFO operation with signals including write_en, read_en, data_in, and data_out. Packet data is written into the FIFO when the write enable signal becomes active. During read operation, the stored packet data is sequentially forwarded to the output through the data_out signal. The waveform also

shows the operation of the full and empty status signals used for FIFO control. Initially, packet data is stored in the buffer during write cycles, and later retrieved during read cycles without altering the order of transmission. The simulation confirms correct packet storage and retrieval operation within the FIFO buffer.

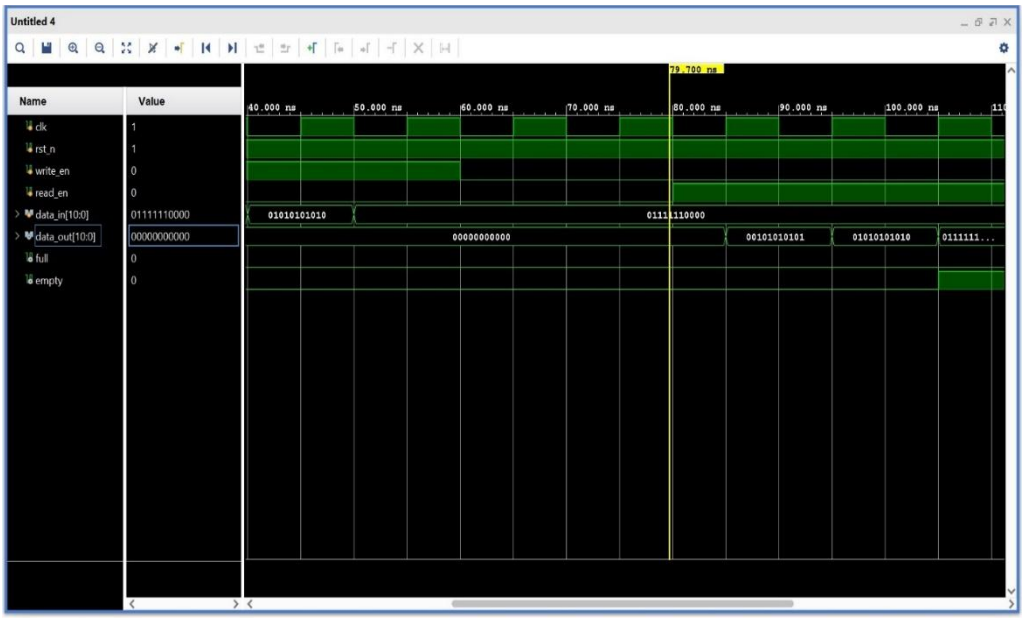


Fig.5.2 Simulated waveform of FIFO buffer

5.2.3 Request Matrix Generation

The waveform illustrates the operation of the Request Matrix used in the proposed router architecture. The signals fifo_n_out[10:0], fifo_s_out[10:0], fifo_e_out[10:0], fifo_w_out[10:0], and fifo_l_out[10:0] represent packet data received from the corresponding FIFO buffers. Based on the destination bits present in the packet header, the Request Matrix generates request signals such as req_n_n, req_l_n, and req_n_s for different output ports. The waveform shows that request signals become active according to the decoded destination information of incoming packets.The simulation confirms correct destination decoding and request generation for routing operations within the proposed NoC router architecture.

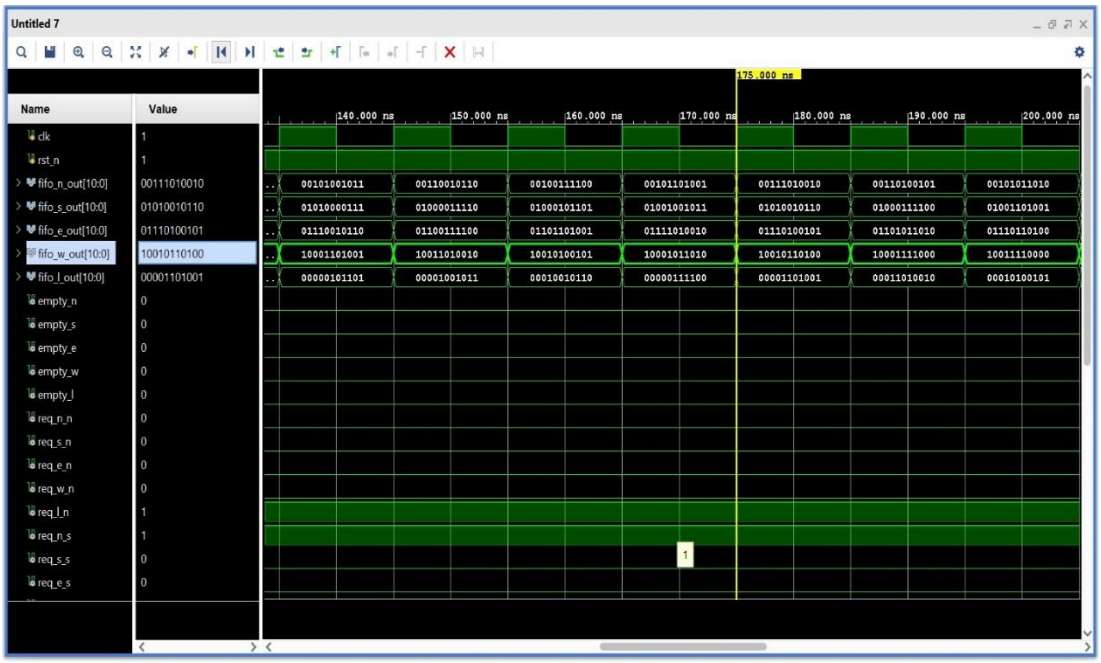


Fig.5.3 Simulated waveform of Request Matrix

5.3 Evaluation of Arbitration Mechanisms

The arbitration unit plays a critical role in resolving conflicts when multiple input ports simultaneously request access to the same output channel. To analyse the effectiveness of different arbitration strategies, several arbitration algorithms were implemented and simulated.

5.3.1 Fixed Priority Arbitration

The waveform shows that when multiple requests are active, the arbiter consistently selects the highest priority input port. Lower priority ports receive access only when higher priority requests are inactive. Although this method is simple and efficient, it may cause starvation of lower priority ports when high priority traffic is continuous.

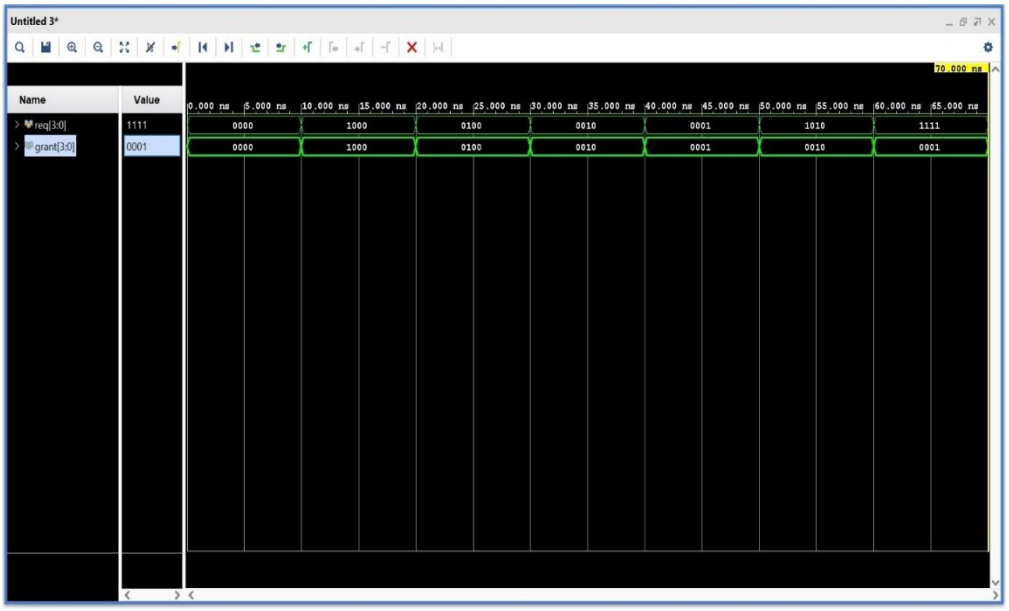


Fig.5.4 Simulated waveform of fixed priority arbiter

5.3.2 Round Robin Arbitration

The waveform illustrates the cyclic movement of grant signals across different input ports. Each port is served sequentially according to the rotating pointer mechanism. This approach eliminates starvation and ensures fair bandwidth allocation among all input ports.

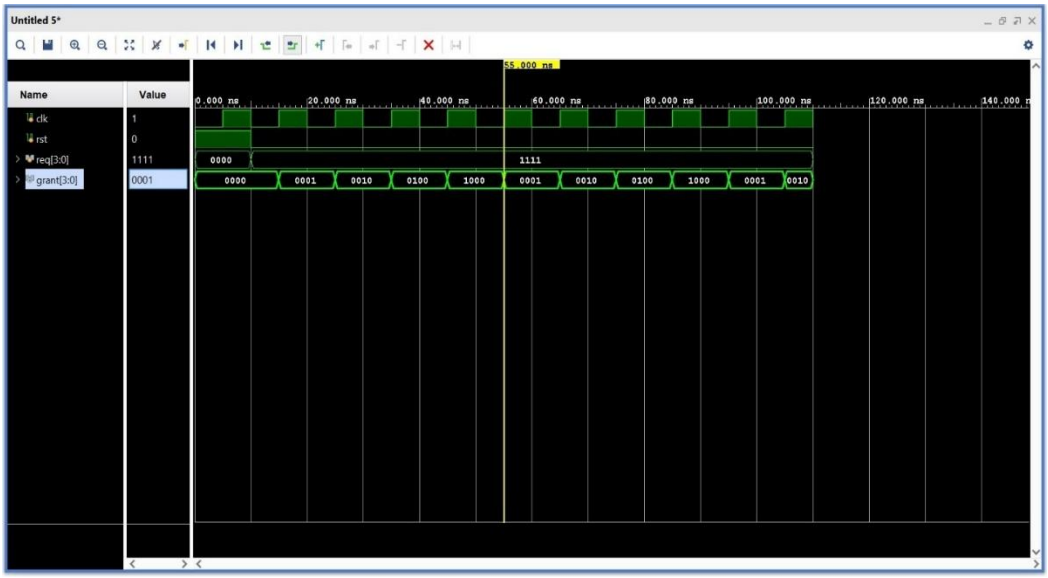


Fig.5.5 Simulated waveform of round robin arbiter

5.3.3 Static Weighted Round Robin Arbitration

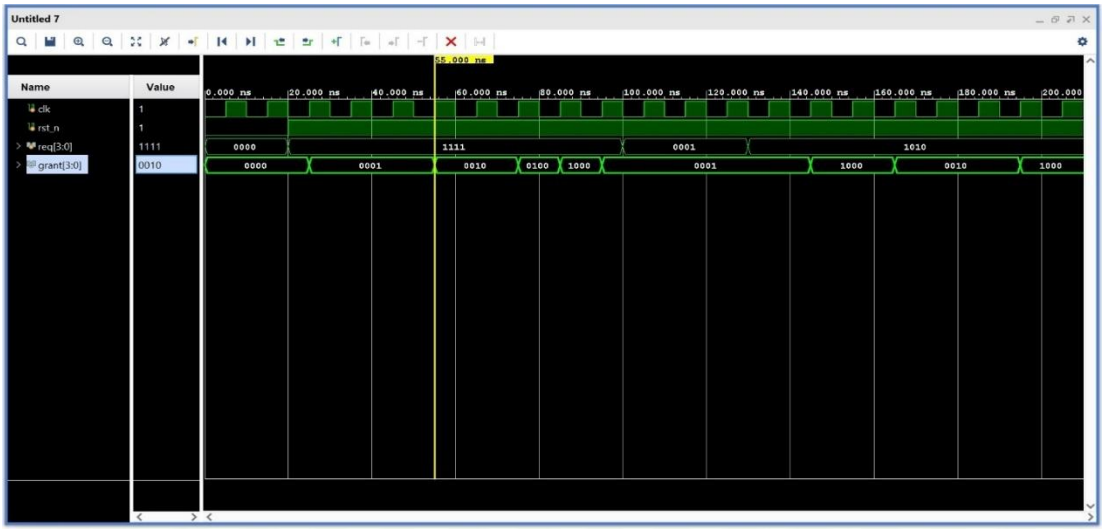


Fig.5.6 Simulated waveform of Static Weighted Round Robin Arbitration

The waveform demonstrates that input ports with higher weights receive multiple grant signals before the arbiter moves to the next port. This behaviour confirms that bandwidth allocation is influenced by the assigned weights. However, since the weights remain fixed, the arbitration mechanism cannot adapt to changing network traffic conditions

5.3.4 Adaptive Weighted Round Robin Arbitration

The waveform illustrates the operation of the proposed Adaptive Weighted Round Robin (AWRR) arbitration mechanism. The signals req[4:0] and grant[4:0] represent the incoming requests and corresponding grant outputs generated by the arbiter. The waveform also shows the variation of weight values, age counters, pointer position, and priority information during arbitration.The arbiter dynamically updates weight values according to request activity, allowing active ports to receive higher bandwidth allocation during packet transmission. The waveform also shows the operation of the pointer mechanism used for cyclic scheduling between competing requests.

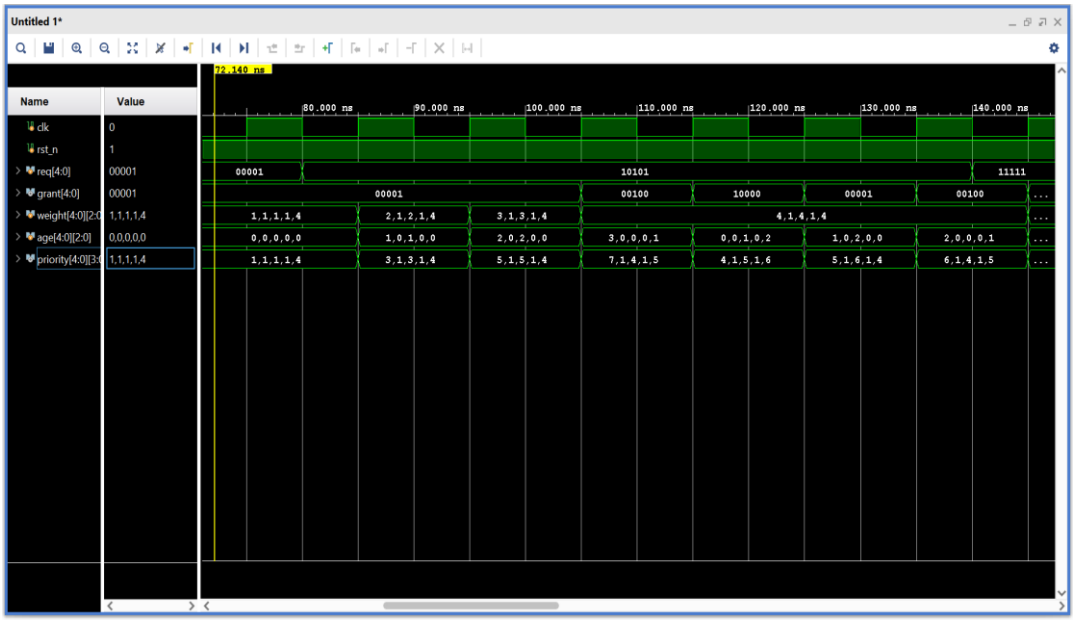


Fig.5.7 Simulated waveform of **Adaptive Weighted Round Robin Arbitration**

The age[4:0] signals increase for requests waiting for transmission over multiple arbitration cycles. As the age values increase, the corresponding priority values are updated to prevent starvation of lower priority requests. The waveform shows the intended behaviour of the proposed Adaptive Weighted Round Robin arbitration mechanism, where active request ports dynamically receive increased weight and priority values, while waiting requests accumulate age values to prevent starvation during arbitration.

5.4 Crossbar Switch

The waveform shows the operation of the crossbar switch during packet transmission within the proposed router architecture. Based on the active grant signals, packets are forwarded to the corresponding output ports through different routing paths. The output signals shows correct packet switching and successful data transmission between the selected input and output ports during successive clock cycles.

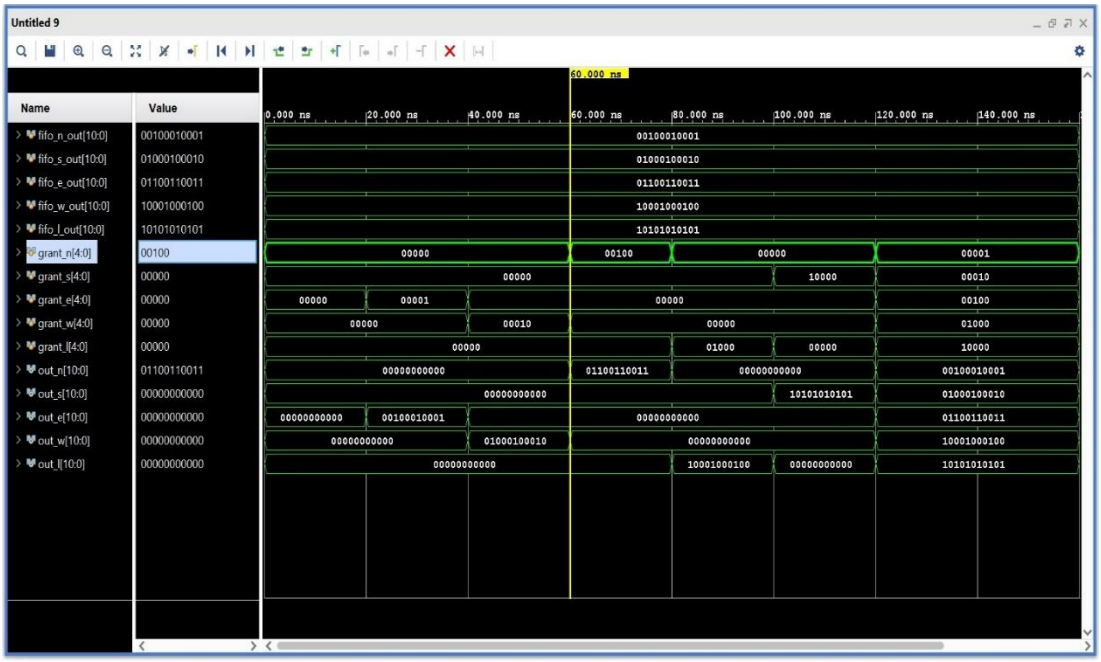


Fig.5.8 Simulated waveform of Crossbar Switch

5.5 Network-on-Chip Router Operation

The waveform illustrates the interaction between different modules within the proposed router architecture. Packet data available at different FIFO outputs is processed by the arbitration unit, which generates corresponding grant signals for packet transmission. The arbiter selects one of the competing input ports and generates the corresponding grant signal. Based on the generated grants, packets are forwarded through the corresponding router output ports during successive clock cycles. The waveform confirms correct packet routing, arbitration, and data transmission within the proposed NoC router architecture. These packets are temporarily stored in FIFO buffers before being processed by the arbitration unit. Packet data available at different FIFO outputs is processed by the arbitration unit, which generates corresponding grant signals for packet transmission.

The arbiter selects one of the competing input ports and generates the corresponding grant signal. Once the request is granted, the packet is forwarded through the router and transmitted to the appropriate output port.Signals north_out, south_out, east_out, west_out, and local_out represent packet transmission in different directions of the

Network-on-Chip router.The waveform confirms that packets are successfully transmitted through the router without conflicts, demonstrating the correct integration of all modules in the proposed architecture.

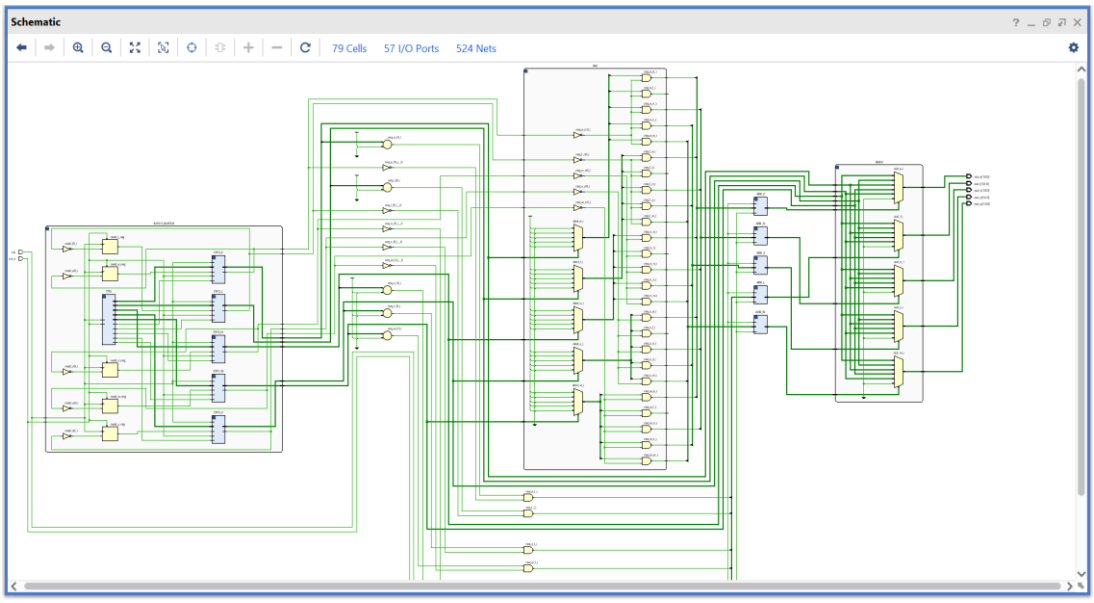


Fig.5.9 Vivado Schematic of Network-on-Chip Router Operation

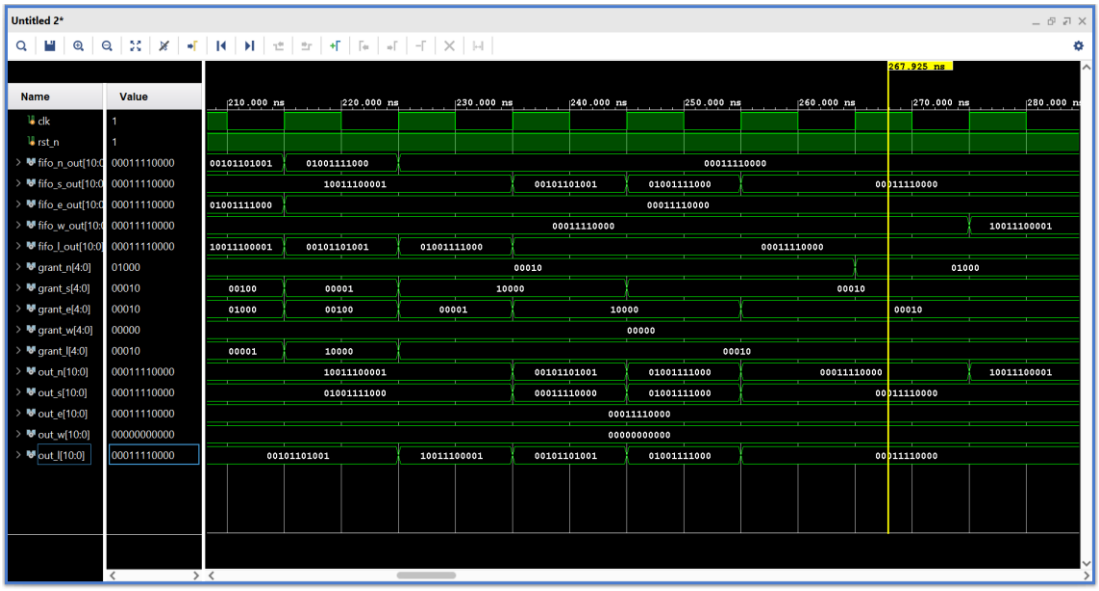


Fig.5.10 Simulated waveform of Network-on-Chip Router Operation

5.6 Power Analysis

Power analysis of the proposed NoC router architecture was carried out using the Vivado power analysis after implementation. Clock gating techniques were incorporated within the Traffic Pattern Generator, FIFO buffers, and arbitration unit to reduce unnecessary switching activity during idle conditions.

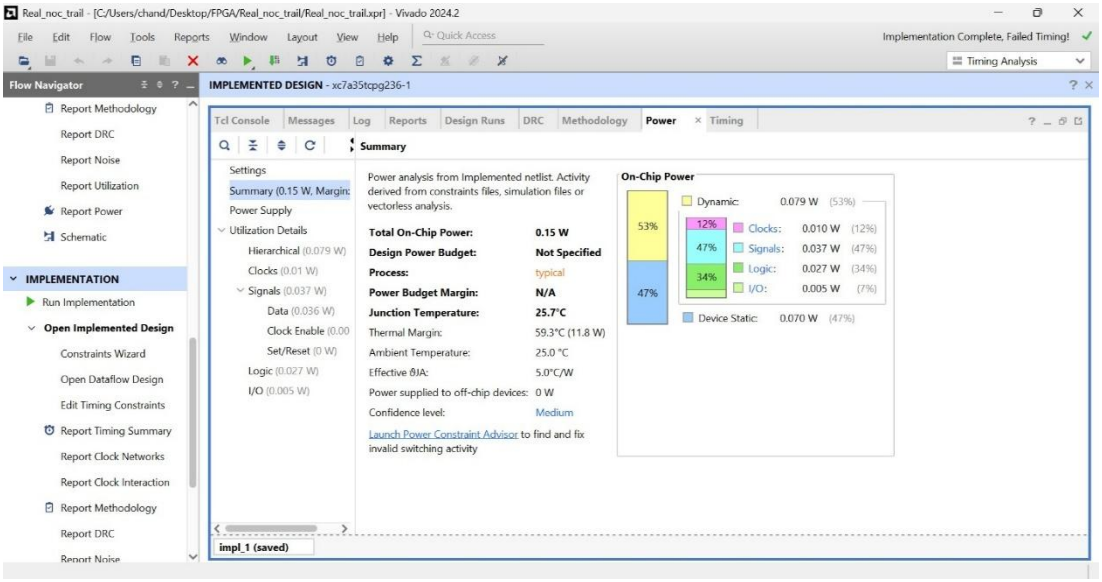


Fig.5.11 Power report of Network-on-Chip Router without Optimization

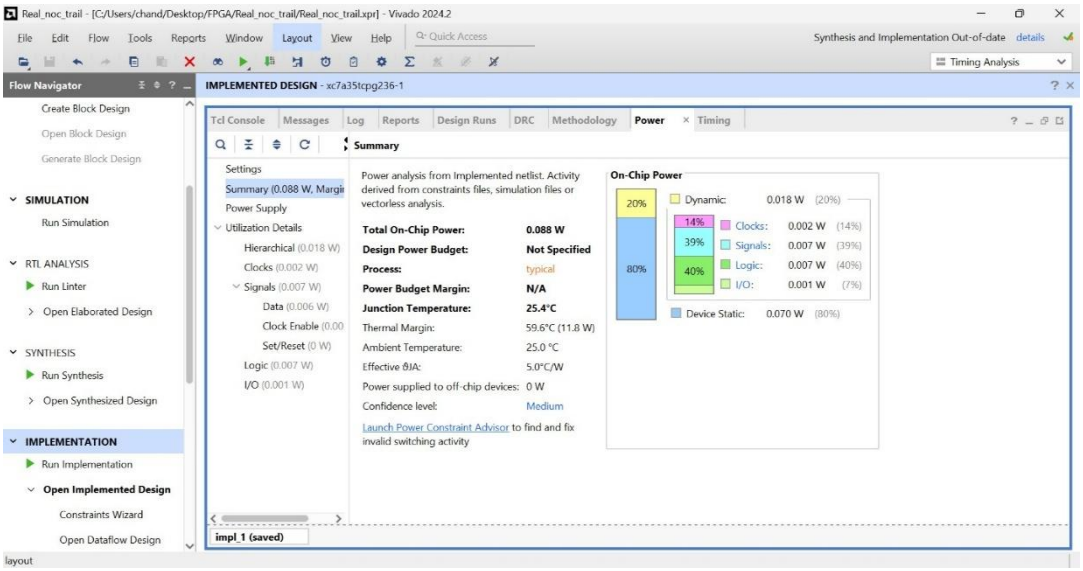


Fig.5.12 Power report of Network-on-Chip Router with Optimization

Table 5.1 Power Comparison of Proposed Router Architecture

Parameter	Without Clock Gating	With Clock Gating
Dynamic Power	0.079 W	0.018 W
Static Power	0.070 W	0.070 W
Total On-Chip Power	0.150 W	0.088 W

The obtained results indicate a reduction of approximately 77.2% in dynamic power consumption and 41.3% in total on-chip power after incorporating clock gating within the proposed architecture. The reduction is achieved by disabling clock activity within inactive modules and minimizing unnecessary switching transitions during router operation.

5.7 Timing Analysis and Resource Utilization

Timing analysis of the proposed NoC router architecture was carried out using the Vivado timing analysis tool after implementation. The timing summary shows that all timing constraints are satisfied without any setup or hold violations.

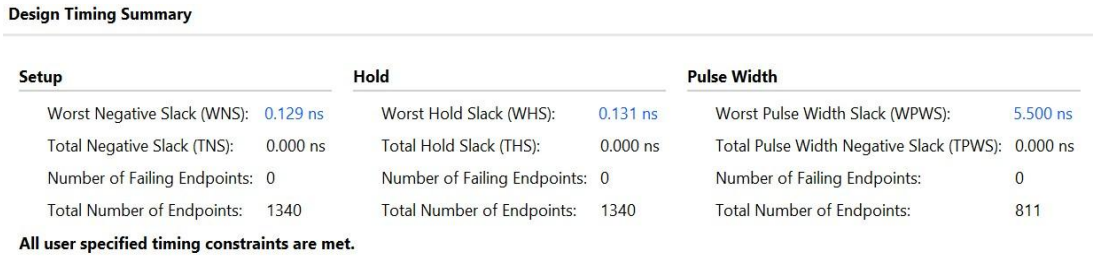


Fig.5.13 Vivado Timing Analysis Report of Network-on-Chip Router

The design operates with a clock period of 14 ns, corresponding to a maximum operating frequency of approximately 71.4 MHz Positive setup slack and hold slack values indicate stable operation of the router architecture under the specified timing constraints.

Table 5.2 Timing Analysis of Proposed Router Architecture

Parameter	Value
Clock Period	14 ns
Maximum Operating Frequency	71.4 MHz
Worst Negative Slack (WNS)	0.129 ns
Worst Hold Slack (WHS)	0.131 ns
Total Negative Slack (TNS)	0.000 ns
Timing Status	Timing Constraints Met

Resource utilization analysis of the proposed NoC router architecture was carried out after implementation using the Vivado after synthesis and implementation. The obtained results indicate that the proposed architecture utilizes a moderate amount of FPGA resources while maintaining efficient router operation and low-power functionality. The utilization report shows that the proposed design occupies limited logic and sequential resources, making it suitable for FPGA-based NoC implementations.

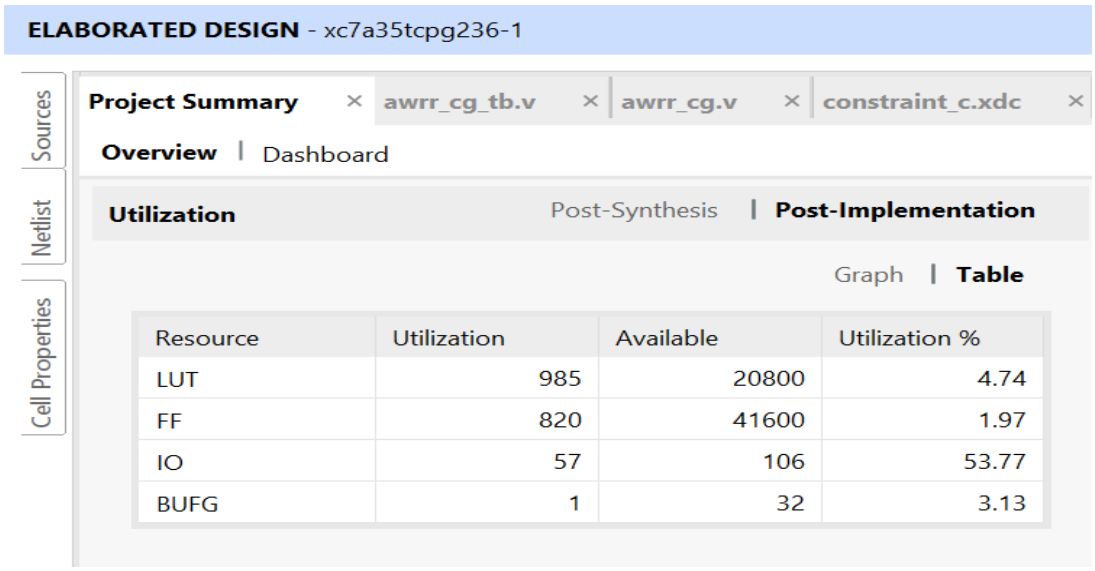


Fig.5.14 Vivado Resource Utilization Report of Network-on-Chip Router

CHAPTER 6
CONCLUSION AND FUTURE WORK

In this thesis, a low-power adaptive arbitration-based Network-on-Chip (NoC) router architecture was designed and implemented for packet-based communication in multi-core System-on-Chip (SoC) systems. The proposed architecture integrates major functional modules including the Traffic Pattern Generator, FIFO buffers, request matrix, arbitration unit, crossbar switch, and five-port NoC router to support efficient packet transmission between network nodes.

Different arbitration techniques such as Fixed Priority, Round Robin, Weighted Round Robin, and Adaptive Weighted Round Robin were implemented and analyzed under varying traffic conditions. Among these techniques, the Adaptive Weighted Round Robin arbitration mechanism demonstrated improved fairness and efficient bandwidth allocation by dynamically updating weight and priority values according to network activity. The inclusion of age-based priority handling also helped in reducing starvation of waiting requests during arbitration.

Furthermore, to improve the efficiency of the router architecture, clock gating techniques were incorporated within major router modules to reduce unnecessary switching activity during idle conditions. Since clock networks contribute significantly to switching power in synchronous systems, the use of clock gating helped in minimizing dynamic power consumption within the proposed design. Power analysis results obtained using the Vivado design environment showed significant reduction in dynamic and total on-chip power after incorporating clock gating into the architecture.

Simulation and implementation results verified the correct operation of packet generation, arbitration, switching, and routing processes within the proposed NoC router architecture. FPGA implementation results also confirmed stable timing performance and efficient resource utilization under the specified timing constraints.

Moreover, this work demonstrates that the proposed low-power adaptive arbitration-based NoC router architecture provides efficient packet communication with improved arbitration fairness and reduced power consumption, making it suitable for modern NoC-based SoC applications.

Future work can focus on extending the proposed router architecture toward large-scale mesh-based multi-router NoC systems for complex multi-core applications. Advanced congestion-aware routing algorithms and adaptive scheduling mechanisms can also be incorporated to further improve network performance under heavy traffic conditions. Additional low-power optimization techniques such as power gating and dynamic voltage scaling may also be explored to further reduce overall power consumption. The proposed architecture can additionally be extended by incorporating Quality-of-Service based arbitration and detailed performance evaluation metrics such as latency, throughput, and bandwidth utilization for comprehensive NoC performance analysis.

REFERENCES

[1] Bjerregaard, Tobias, and Shankar Mahadevan. "A survey of research and practices of network-on-chip." *ACM Computing Surveys (CSUR)* 38.1 (2006): 1-es.

[2] Agarwal, Ankur. "Survey of network on chip (noc) architectures & contributions." (2009).

[3] Swapna, S., Ayas Kanta Swain, and Kamala Kanta Mahapatra. "Design and analysis of five port router for network on chip." *2012 Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics*. IEEE, 2012.

[4] Jiang, Nan, et al. "A detailed and flexible cycle-accurate network-on-chip simulator." *2013 IEEE international symposium on performance analysis of systems and software (ISPASS)*. IEEE, 2013.

[5] Pratap, Surya, and Chhagan Charan. "Design and Implementation of Low-Latency Five-Port NoC Router for Embedded Systems." *2025 Seventh International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*. IEEE, 2025.

[6] Saranya, M. N., and Rathnamala Rao. "Design and verification of an asynchronous noc router architecture for gals systems." *Journal of Electronic Testing* 40.1 (2024): 61-74.

[7] Malladhi, Nagarjuna, and Girish V Attimarad. "Design and development of novel AXI interconnect based noc architecture for SoC with reduced latency and improved throughput." *International Journal of Computing and Digital Systems* 17.1 (2024): 1-14.

[8] Debbarma, Jeetu, Rita Banik, and Dalia Debbarma. "Energy-Efficient Network-On-Chip Routers Using A Power Gated Round Robin Arbiter Ensuring Minimal Impact On Latency." *Results in Engineering* (2025): 108148.

[9] Rashmi Samanth, Subramanya G. Nayak, “Design and Analysis of Four Port Router for Network-On-Chip Applications”, Journal of Physics: Conference Series, 2017.

[10] Kerakalamatti, B.S. and Nagaraj, P., 2015. Design and implementation of NOC based parallel AES computation. *International Journal of Computer Applications*, *975*, p.8887.

[11] De Micheli, G. and Benini, L., 2017. Networks on chips: 15 years later. *Computer*, *50*(5), pp.10-11.

[12] Menaka, M., Aishwarya, K. and Kumar, T.D., 2023, August. Asynchronous circular buffers based on FIFO for network on chips. In *2023 International Conference on Circuit Power and Computing Technologies (ICCPCT)* (pp. 1356-1361). IEEE.

[13] Chaturvedi, S., Saranya, M.N. and Rao, R., 2022, October. Design of Asynchronous Circular FIFO Buffer for Asynchronous Network on Chips. In *2022 International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)* (pp. 66-71). IEEE.

[14] Kundu, S. and Chattopadhyay, S., 2018. *Network-on-chip: the next generation of system-on-chip integration*. CRC press.

[15] Tsai, W.C., Lan, Y.C., Hu, Y.H. and Chen, S.J., 2012. Networks on chips: structure and design methodologies. *Journal of Electrical and Computer Engineering*, *2012*(1), p.509465.