International Journal of
Information Technology & Computer Engineering

ISSN 2347–3657

Volume 14, Issue 2, 2026

1*3 Router Design

Ms B Eleena[1], Vanam Pravalika[2], Chittepu Shreya[3], Molagasi Saraswati[4]
[1]Assistant Professor; Department Of Electronics And Communications Engineering Bhoj Reddy Engineering College For Women Hyderabad India
[2,3,4]B.Tech Student's; Department Of Electronics And Communications Engineering Bhoj Reddy Engineering College For Women Hyderabad India
Mail Id's; shreyareddychittepu@gmail.com,pravalikavanam22@gmail.com,molagasisaraswati@gmail.com

Abstract

The increasing complexity of modern System-on-Chip (SoC) architectures demands fast, scalable, and reliable communication mechanisms between integrated modules. Conventional shared-bus communication methods face significant challenges such as bandwidth limitations, congestion, latency, and poor scalability when the number of connected components increases. To address these issues, packet-switched routing architectures have emerged as an effective solution for high-speed digital communication systems.This paper presents the design and implementation of a Verilog HDL-based 1×3 packet router capable of transmitting data packets from a single input channel to one of three output ports according to destination address information embedded in the packet header. The proposed router architecture consists of a Finite State Machine (FSM) controller, synchronizer, register unit, FIFO buffers, and parity checker modules. The FSM supervises packet reception, routing, buffering, synchronization, and transmission processes to ensure smooth data flow. FIFO memories are incorporated to prevent packet loss and manage temporary storage during varying transmission rates.To enhance communication reliability, parity-based error detection is implemented for identifying corrupted packets during transmission. The complete router design is modeled using Verilog HDL and functionally verified through simulation using standard FPGA/ASIC design tools. Simulation outcomes confirm accurate packet routing, synchronization, and error handling operations.The proposed architecture offers improved throughput, reduced transmission delay, and efficient packet management compared with conventional bus-oriented communication systems. Owing to its modular and scalable structure, the router can be extended for Network-on-Chip (NoC) architectures and advanced multi-core communication applications.

Keywords
1×3 Router, Verilog HDL, System-on-Chip (SoC), Packet Routing, Finite State Machine (FSM), FIFO Buffer, Network-on-Chip (NoC), Digital Communication, Packet Switching, Error Detection.

Introduction

The rapid advancement of semiconductor technology and digital system design has significantly increased the demand for high-speed and reliable communication architectures within integrated circuits. In modern System-on-Chip (SoC) and Network-on-Chip (NoC) environments, multiple functional modules such as processors, memory units, and peripheral devices operate simultaneously and require efficient data exchange mechanisms. Traditional bus-based communication systems face several limitations, including congestion, increased latency, reduced bandwidth, and poor scalability when the number of interconnected modules increases. To overcome these challenges, packet-based routing architectures have emerged as an effective solution for improving communication efficiency in digital systems.A 1×3 router is a fundamental packet-routing component used in digital communication systems to transfer data from a single input channel to one of three output channels based on destination information. The router receives data in the form of packets, where each packet generally consists of a header, payload, and parity or control bits. The header field contains the destination address, which determines the appropriate output port for packet transmission. By analyzing the destination address, the router forwards the packet to the required output channel while preventing conflicts and ensuring organized data flow. This packet-switching mechanism improves communication reliability and supports efficient utilization of system resources.The architecture of a 1×3 router typically includes functional modules such as FIFO buffers, synchronizers, registers, control logic, and a Finite State Machine (FSM). FIFO memories are used to temporarily store packets during transmission and prevent data loss caused by speed mismatches or congestion. The FSM controller manages routing operations, packet synchronization, buffering, and control signal generation to ensure proper communication between modules. Synchronization circuits maintain timing coordination among different components, while registers store intermediate packet information during processing. In addition, parity-checking mechanisms can be implemented to detect transmission errors and improve communication reliability.Verilog Hardware Description Language (HDL) is widely used for designing and implementing digital routing systems because of its flexibility, modularity, and compatibility with FPGA and ASIC development platforms. The implementation of a 1×3 router using Verilog HDL provides an efficient and scalable solution for packet-based communication systems. Such routers are extensively used in FPGA-based applications, embedded systems, and NoC architectures due to their ability to support high-speed data transfer with reduced delay and improved throughput. Therefore, the 1×3 router serves as an essential building block in the development of modern digital communication and on-chip networking systems.

Literature Survey

The continuous growth of digital communication systems and System-on-Chip (SoC) technology has created a strong demand for efficient and reliable data transfer mechanisms. In modern integrated systems, multiple modules such as processors, memory units, and peripheral devices communicate simultaneously, requiring high-speed and organized communication architectures. Conventional communication approaches, particularly shared bus architectures, are widely used in earlier digital systems because of their simple implementation. However, these methods suffer from several disadvantages, including limited bandwidth, high communication delay, congestion, and reduced scalability when the number of connected modules increases. As the complexity of Very Large Scale Integration (VLSI) systems continues to grow, traditional bus-based communication structures are no longer capable of meeting performance requirements.To address these limitations, researchers introduced packet-based communication techniques inspired by computer networking concepts. In packet switching, data is divided into smaller units called packets, where each packet contains destination information, payload data, and control bits. This technique enables efficient bandwidth utilization and reliable data transfer between modules. Packet-based communication later became the foundation for the development of Network-on-Chip (NoC) architectures, where routers act as key communication elements responsible for directing packets between different functional units within a chip.Several research studies have focused on the design and optimization of router architectures for digital communication systems. Early communication systems mainly depended on circuit-switching techniques in which a dedicated communication path was established before data transmission. Although circuit switching provided stable communication, it lacked flexibility and efficient resource utilization. Packet-switching mechanisms improved overall communication efficiency by dynamically routing packets based on destination addresses. Modern routers analyze packet headers and forward packets to the required destination through suitable output channels. This approach significantly improves scalability, throughput, and communication reliability in digital systems.With the advancement of NoC technology, routers became essential components for enabling parallel communication among multiple processing elements. Research on NoC architectures highlights advantages such as reduced communication delay, improved scalability, parallel data transfer capability, and lower power consumption compared with traditional communication methods. Small-scale router models such as 1×2, 1×3, and 2×2 routers are commonly implemented in research studies to analyze routing behavior and communication efficiency in packet-based systems. These routers serve as fundamental building blocks for larger NoC architectures.FIFO-based router architectures have also gained significant attention in the literature because of their ability to improve packet flow control and prevent data loss. FIFO (First-In-First-Out) buffers temporarily store incoming packets before forwarding them to the destination output ports. Researchers observed that FIFO memories effectively handle congestion conditions, synchronize differences between input and output data rates, and ensure reliable packet transmission. In 1×3 router architectures, dedicated FIFO buffers are typically assigned to each output port to support independent packet storage and smooth communication.Another important area discussed in previous studies is synchronization and flow control in packet routing systems. Efficient synchronization between transmitting and receiving modules is essential for maintaining reliable communication. Techniques such as handshaking signals, busy indicators, valid signals, and flow control mechanisms are widely used to avoid packet corruption, timing mismatches, and data collisions. These synchronization methods improve communication stability and ensure proper coordination between different router modules.Researchers have also explored the implementation of Finite State Machine (FSM)-based control logic for packet routers. FSM controllers manage packet reception, routing operations, FIFO read/write processes, parity checking, and synchronization tasks. The use of FSMs improves control efficiency and enables systematic packet handling within the router architecture. Error detection techniques such as parity checking are commonly implemented to ensure data integrity and identify transmission errors during communication.The literature survey indicates that packet-based router architectures provide a more efficient communication solution compared with conventional bus-oriented systems. Features such as FIFO buffering, FSM-based control, synchronization mechanisms, and parity-based error detection contribute to reliable and scalable communication performance. Therefore, the implementation of a Verilog HDL-based 1×3 router represents an effective approach for supporting high-speed data transfer in modern SoC and NoC applications.

Existing and Proposed Methods

Existing Methods

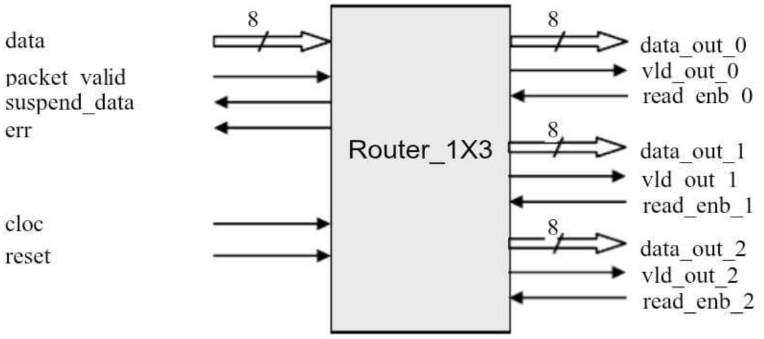


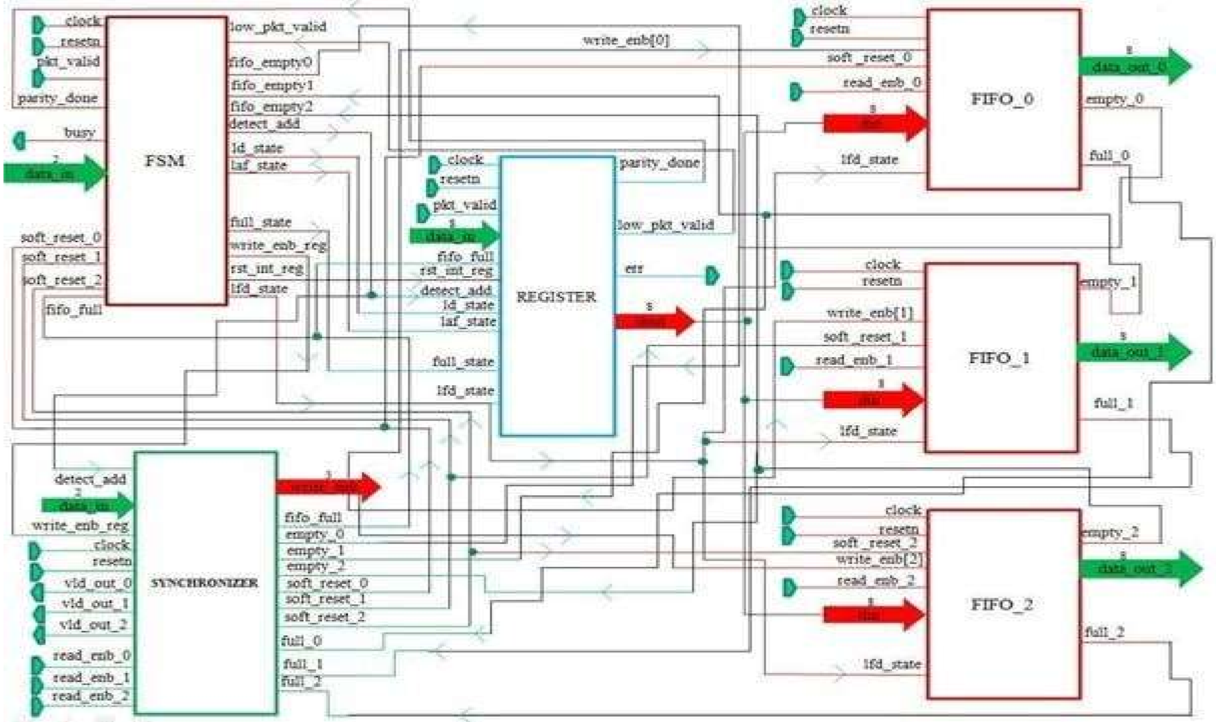
Figure - Block Diagram of Router_1X3

Earlier digital systems mainly depended on centralized communication structures for transferring data between multiple modules. One of the most commonly used techniques was bus-based communication, where all modules shared a common communication channel. In this method, devices requested access to the shared bus, and an arbiter granted permission for data transfer. Although bus architectures were simple and cost-effective for small systems, they suffered from major disadvantages such as increased communication delay during heavy traffic, limited bandwidth sharing, and possible data collision conditions. Furthermore, bus-based systems were not suitable for modern parallel processing applications because all communication operations occurred sequentially.Crossbar switching was later introduced as an alternative communication method. Crossbar switches provided dedicated paths between input and output ports, enabling simultaneous data transfer between multiple devices. This approach improved communication speed and supported parallel processing operations. However, the hardware complexity and implementation cost of crossbar switches increased significantly with system size, making them less practical for large-scale integrated systems. Due to these limitations, packet-based routing architectures became the preferred solution for modern digital communication systems.The existing 1×3 router architecture operates as a packet-switching communication system that transfers data from a single input port to one of three output ports based on the destination address contained in the packet header. The router receives packets synchronously using clock and reset signals and forwards them to the selected output channel through control logic and buffering mechanisms. The architecture typically contains one input interface, three output interfaces, FIFO buffers, synchronization modules, and routing control units. Incoming packets are analyzed, and the destination address field determines the corresponding output port for packet transmission.

Proposed Method

The proposed method focuses on the design and implementation of a Verilog HDL-based 1×3 packet router capable of efficiently routing incoming packets from one input channel to three independent output channels. The router architecture is designed to improve communication efficiency, minimize transmission delay, and ensure reliable packet delivery within digital systems.The proposed router consists of several functional modules including the Input Interface, Router Register, FIFO Buffers, Synchronizer, FSM Controller, and Output Interface. Data packets received at the input port contain destination address information, payload data, and parity bits. The control logic continuously monitors the destination address and activates the corresponding output channel. FIFO buffers are incorporated to temporarily store packets and prevent data loss during congestion or busy conditions. This buffering mechanism ensures smooth packet flow and supports synchronization between transmitting and receiving modules.The router operates synchronously using a clock signal to coordinate internal operations. Error detection and packet validation mechanisms are implemented using parity checking techniques to ensure communication reliability. Once the destination address is identified, the packet is transferred to the selected output port while other output channels remain inactive. The modular structure of the router improves scalability and simplifies debugging and verification processes.

Router Architecture



The 1×3 router architecture is designed as a modular packet-routing system that transfers packets from a single input channel to three separate output ports. The architecture consists of several interconnected modules responsible for packet reception, storage, synchronization, routing, and output transmission.The input interface receives 8-bit packet data from the source along with a packet-valid signal that indicates the presence of valid packet information. The first byte of the packet represents the header, which contains the destination address and payload length. The Router Register temporarily stores header, payload, and parity bytes while generating status signals related to parity completion and error detection.The FSM acts as the central control unit of the router. It manages packet loading, FIFO writing operations, synchronization, parity checking, and state transitions. Based on status signals received from FIFO buffers and the register module, the FSM generates control signals required for packet transfer operations.The Synchronizer module decodes the destination address from the header byte and activates the corresponding FIFO write-enable signal. It also monitors FIFO full and empty conditions to prevent overflow and underflow situations. Separate FIFO memories are used for each output port to temporarily store packet data before transmission. These FIFO buffers support smooth communication by handling variations in transmission speeds between sender and receiver modules.The output interface consists of three independent output ports and corresponding valid signals. Packet data is transferred to the selected output port when the receiver activates the read-enable signal. If FIFO buffers become full, the FSM temporarily suspends writing operations until sufficient space becomes available.

Packet Format

The packet format used in the proposed router consists of three major sections: Header, Payload, and Parity. The header byte contains destination address information and payload length details. The destination address identifies the required output channel, while the payload length specifies the number of payload bytes present in the packet.The payload section contains the actual data to be transmitted. Payload data may represent sensor values, control information, processor instructions, or other communication data depending on the application. The final byte of the packet is the parity byte, which is used for error detection. The router calculates internal parity values and compares them with the received parity byte to identify transmission errors. If parity values do not match, an error signal is generated.

FIFO, Synchronization, and FSM Operation

The router uses three FIFO memories, each having a depth of 16 bytes and width of 9 bits. The additional bit is used to identify header bytes during packet processing. FIFO memories support simultaneous read and write operations and generate full and empty status signals for flow control. During timeout conditions, FIFO memories are internally reset to maintain reliable operation.The Synchronizer module ensures proper coordination between FSM and FIFO modules. It generates write-enable signals for the selected FIFO based on the destination address and monitors timeout conditions using valid and read-enable signals. Internal soft-reset signals are generated if data is not read within a specified number of clock cycles.

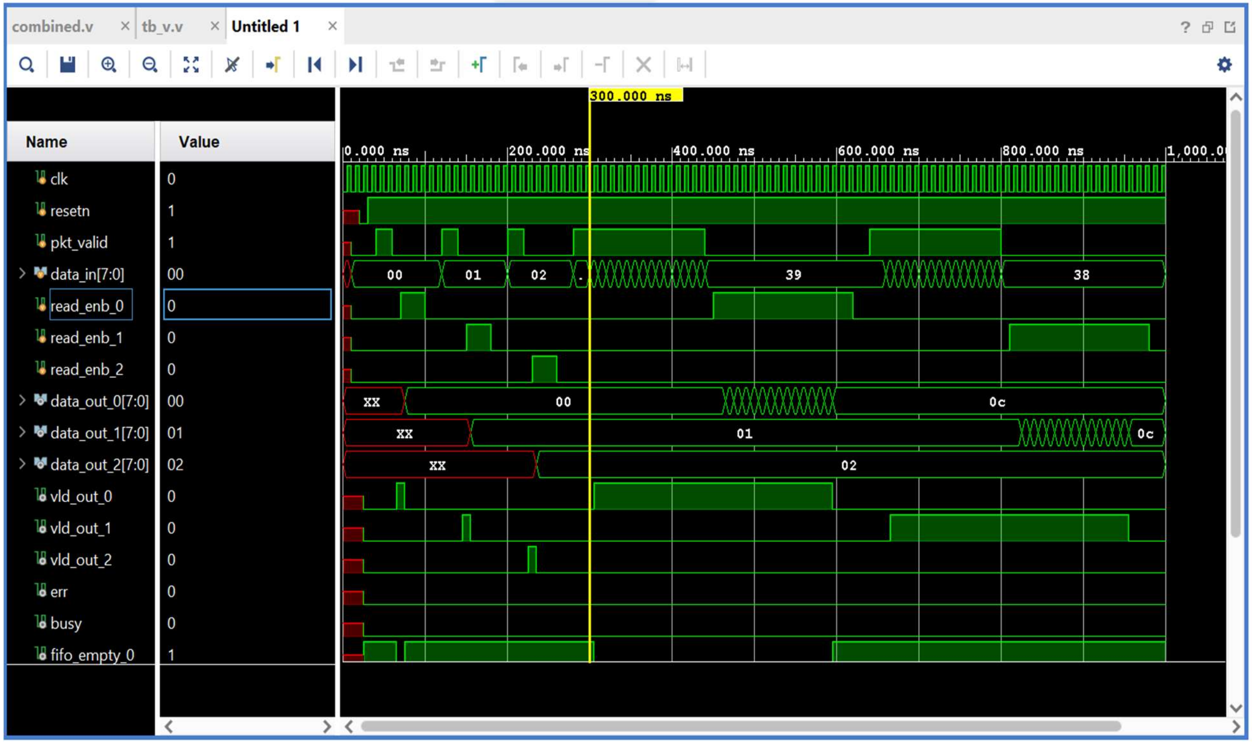
The FSM controls all router operations using multiple states such as Decode Address, Load First Data, Load Data, Load Parity, FIFO Full State, Load After Full, Wait Till Empty, and Check Parity Error. Each state performs specific operations related to packet loading, synchronization, FIFO management, and error checking. The FSM ensures systematic packet transfer and reliable router functionality.

Software Tools and Simulation Methodology

The proposed router is implemented using Verilog HDL and verified through functional simulation. Xilinx Vivado, Icarus Verilog (Iverilog), and GTKWave are used as the primary software tools for coding, compilation, simulation, waveform generation, and debugging. RTL-level design methodology is adopted to model data flow between registers and logical components.A dedicated Verilog testbench is developed to verify router functionality under different operating conditions. The testbench generates clock and reset signals, applies packet data, controls read-enable signals, and monitors router outputs. Simulation waveforms are analyzed to verify correct packet routing, busy conditions, FIFO operations, valid output generation, and parity error detection.Several test cases are applied during simulation, including valid packet routing to different output ports, FIFO full conditions, busy-state testing, parity-error testing, and reset-condition verification. Simulation results confirm the correct functionality and reliability of the proposed 1×3 router architecture.

Results and Discussion

Simulation Results



The proposed 1×3 packet router was implemented and verified using Verilog HDL simulation tools. Functional simulation was performed to analyze the behavior of each internal module as well as the complete router architecture. The simulation waveforms confirm that the router operates according to the specified design requirements under different packet transmission conditions.The input packet data is applied through the data_in[7:0] bus along with the pkt_valid control signal. The router accepts packet data only when the packet-valid signal is active. The system clock controls synchronization between all internal modules, while the active-low reset signal initializes the router during startup conditions. The incoming packet is initially processed by the Register module, where the header, payload, and parity information are temporarily stored and analyzed.The header byte contains the destination address used to determine the required output channel. Based on the destination address, the Synchronizer module generates write-enable signals for the corresponding FIFO buffer. Separate FIFO memories are used for each output port to ensure smooth packet transfer and avoid data loss during congestion conditions. The simulation waveforms confirm that packet data is correctly written into the selected FIFO memory and later transferred to the appropriate output channel when the corresponding read-enable signal is asserted.The FSM controller manages all routing operations by controlling packet loading, synchronization, FIFO writing, parity checking, and state transitions. Simulation results show proper FSM transitions through states such as Decode Address, Load First Data, Load Data, Load Parity, FIFO Full State, and Check Parity Error. The busy signal becomes active whenever the router is processing packet data or handling FIFO full conditions, preventing invalid data transfer during operation.The output waveforms demonstrate that data packets are successfully routed to the correct output ports through data_out_0, data_out_1, and data_out_2. The valid output signals vld_out_0, vld_out_1, and vld_out_2 indicate the availability of valid packet data at the respective output channels. The router transfers packets only when the destination module activates the corresponding read-enable signal. The FIFO empty and full status signals properly indicate buffer conditions and assist in maintaining reliable communication between input and output modules.Parity checking functionality is also verified through simulation. The Router Register calculates internal parity values for the header and payload data and compares them with the received parity byte. When parity mismatches occur, the error signal becomes active, confirming successful error detection capability within the router architecture. The waveform analysis demonstrates that corrupted packets are identified correctly, ensuring reliable packet communication.The simulation results further confirm that the router maintains synchronization between transmitting and receiving modules under different operating conditions. FIFO buffering effectively handles temporary congestion and prevents packet loss. The integrated router design successfully performs packet routing, synchronization, buffering, and error detection operations while maintaining stable communication performance.

Discussion

The simulation waveforms and functional verification results indicate that the proposed 1×3 packet router operates efficiently and reliably. The modular design approach simplifies implementation, debugging, and verification while improving scalability for future enhancements. The FSM-based control mechanism effectively coordinates routing operations and ensures systematic packet handling throughout the communication process.

The FIFO buffering mechanism plays an important role in maintaining smooth packet flow by temporarily storing incoming packets during busy or congestion conditions. This improves communication reliability and prevents packet loss due to speed mismatches between transmitting and receiving modules. Synchronization logic further enhances router stability by coordinating data transfer between different functional blocks.

The simulation results also demonstrate that the packet-based routing architecture provides significant advantages over conventional bus-based communication methods. The router supports organized data transfer, reduced communication delay, improved scalability, and reliable packet delivery. The parity-checking mechanism increases communication integrity by detecting transmission errors before packets reach the destination modules.

Conclusion

This work presented the design and implementation of a Verilog HDL-based 1×3 packet router for efficient packet-based communication in digital systems. The router architecture was developed using Register Transfer Level (RTL) design methodology and divided into several functional modules including FIFO buffers, Register block, Synchronizer, FSM Controller, and Output Interface. The modular design approach improved implementation simplicity, debugging capability, and scalability of the overall system.The proposed router receives packet data from a single input channel and forwards it to one of three output channels based on the destination address specified in the packet header. The FSM controller effectively manages packet loading, routing operations, synchronization, FIFO handling, and parity verification processes. FIFO memories are used to temporarily store packets and handle communication speed mismatches between input and output ports, thereby preventing packet loss during transmission. The Synchronizer module ensures proper coordination between internal modules and maintains stable communication performance throughout router operation.Functional simulation and waveform analysis confirm the correct operation of the router under different packet transmission conditions. The simulation results demonstrate accurate packet routing, proper generation of control signals, reliable FIFO operations, successful parity error detection, and correct handling of busy and valid conditions. The router maintains reliable communication while ensuring organized data transfer and efficient packet handling.The implemented 1×3 router architecture offers several advantages such as improved communication efficiency, reduced congestion, reliable packet transfer, modular scalability, and enhanced synchronization. These characteristics make the proposed design suitable for modern System-on-Chip (SoC), Network-on-Chip (NoC), FPGA-based communication, and embedded digital applications. Therefore, the developed router architecture provides an effective and scalable solution for high-speed packet-based communication systems.

Future Scope

Although the proposed 1×3 packet router achieves reliable and efficient communication, several enhancements can be incorporated in future research to improve performance and scalability further. The current architecture can be extended to larger multi-port routers such as 2×2, 4×4, or N×N routing structures to support complex Network-on-Chip communication systems. Such extensions would enable communication among a larger number of processing elements and improve parallel data transfer capability.Advanced routing algorithms may also be implemented to reduce communication latency and increase throughput under heavy traffic conditions. Adaptive and dynamic routing techniques can improve overall network efficiency by selecting optimal routing paths during packet transmission. Power optimization strategies may further be applied to reduce energy consumption, making the router suitable for low-power embedded and portable devices.Future enhancements may include the integration of advanced error detection and correction techniques such as Cyclic Redundancy Check (CRC), Hamming codes, or forward error correction methods to improve communication reliability. Hardware implementation using modern FPGA and ASIC technologies can further increase operating speed and optimize hardware resource utilization.The router can also be integrated with real-time embedded processors, multi-core systems, and high-performance communication platforms for advanced SoC applications. Additional research may focus on improving Quality of Service (QoS), fault tolerance, congestion management, and security mechanisms for next-generation communication systems. Thus, the proposed work establishes a strong foundation for future development of scalable, reliable, and high-performance router architectures in modern digital communication environments.

References

1. J. Bhasker, *A Verilog HDL Primer*, 3rd Edition, BS Publications, 2003.
2. Verilog HDL: A Guide to Digital Design and Synthesis, 2nd Edition, Pearson Education, 2003.
3. Digital Design, 5th Edition, Pearson Education, 2013.
4. Modern VLSI Design: System-on-Chip Design, Pearson Education, 2008.
5. CMOS VLSI Design: A Circuits and Systems Perspective, 4th Edition, Pearson, 2011.
6. Fundamentals of Digital Logic with Verilog Design, McGraw-Hill Education, 2014.
7. IEEE Standards Association, *IEEE Standard for Verilog Hardware Description Language*, IEEE Std 1364-2005.
8. Data and Computer Communications, 10th Edition, Pearson, 2013.
9. Data Communications and Networking, McGraw-Hill, 2012.
10. Router 1×3 RTL Design and Verification Technical Document, Scribd Online Resource.
11. IEEE Conference Papers and Journals on Network-on-Chip Router Design Techniques.
12. FPGA Prototyping by Verilog Examples, Wiley-Interscience, 2008.
13. Advanced Digital Design with the Verilog HDL, Prentice Hall, 2011.
14. Computer Organization and Design, Morgan Kaufmann Publishers, 2014.
15. IEEE Xplore Digital Library, Research Papers on Network-on-Chip Router Architectures.