

Efficient Data Transfer in VLSI Systems using Smart Buffering and Routing Techniques

Malashree K S¹, Deepak T S², Gopichand R³, Kiran C K⁴, Yashwanth D⁵

¹Professor, ²Final year Student, ³Final year Student, ⁴Final year Student, ⁵Final year Student

Department of Electronics and Communication Engineering, P E S institute of technology and management, Shimoga

Abstract - The increasing use of Network-on-Chip (NoC) interconnect schemes is driven by their versatility and ability to scale efficiently. In NoCs, routers are critical components that significantly influence both performance and cost. To overcome design challenges and improve NoC router efficiency, a variety of new techniques have been integrated. We present a novel NoC router design with multiple local ports, created using Verilog models. The main objectives of this design are to minimize the router's physical size and enhance the speed of data transmission. The architecture leverages XY routing and incorporates optimized buffering, Credit-Based Flow Control, and a Deterministic Clock Approach. The proposed design is evaluated in terms of area usage and operating frequencies. Using distributed control mechanisms, the routers achieve autonomous operation without the need for complex handshaking processes, resulting in improved efficiency and scalability. This Multi-Local Router design is capable of handling multiple independent data requests concurrently, making it ideal for managing high data traffic in advanced Field Programmable System-on-Chips (FPSoCs). The design's strengths in Power, Performance, and Area (PPA) optimization make it particularly suited for computationally demanding applications. To validate its effectiveness, the router was implemented and synthesized on a Xilinx Virtex 4 FPGA (4vsx25ff668-12), demonstrating its practical viability. This breakthrough paves the way for more efficient NoC implementations in future FPSoC designs.

Key Words: Network-on-Chip (NoC), Router design, Verilog models, XY routing, Optimized buffering, Credit-Based Flow Control, Deterministic Clock Approach, Distributed control mechanisms, Field Programmable System-on-Chips (FPSoCs), Power, Performance, and Area (PPA) optimization, FPGA implementation.

1. INTRODUCTION

The rapid development of System-on-Chip (SoC) technology has enabled the integration of various types of cores, from simple memory units to complex Digital Signal Processors (DSPs), onto a single chip. This growing complexity has led to increased challenges in managing communication between the different on-chip components, as the number of processing elements continues to rise. A promising solution to these communication challenges is the adoption of Network-on-Chip (NoC) architectures within SoCs, which provide an efficient way to manage data transmission across the chip

while meeting critical design requirements such as Power, Performance, and Area (PPA). NoC achieves this by establishing a network of routers and interconnects that link the processing components, effectively overcoming the limitations of conventional bus-based communication systems.

The performance of a NoC largely depends on the router, which is responsible for managing data flow between network nodes. As the scale of on-chip networks increases, the design of the router becomes increasingly important. Router performance is influenced by several factors, including the NoC architecture (whether Synchronous, Asynchronous, or Globally Asynchronous Locally Synchronous, GALS), buffer design, arbiter structure, network topology, and routing strategies. By optimizing these elements, it is possible to enhance overall system performance and ensure efficient communication among the cores.

In Synchronous NoCs, routers function according to a global clock, which can lead to higher power consumption. Although these designs are often fast and compact, they face challenges when operating at higher frequencies, potentially leading to issues such as Electromagnetic Interference (EMI). To address the limitations of global clock distribution in Synchronous designs, intermediate approaches like GALS have been introduced. GALS systems divide the NoC into smaller synchronous regions, allowing each region to operate independently without a global clock. Hybrid NoCs, which combine both Synchronous and Asynchronous communication, have also been explored, with some designs focusing on energy efficiency and reduced latency by utilizing sub-routers dedicated to Synchronous control and Asynchronous data transfer.

On the other hand, Asynchronous NoCs, which operate without a global clock, offer greater power efficiency, although they tend to be slower than their Synchronous counterparts. These designs are ideal for real-time applications that require low power and the transmission of smaller data packets. Conversely, Synchronous NoCs are better suited to applications with large data packets and continuous transmission requirements, such as multimedia processing. Several Asynchronous NoC designs have been proposed, including those using

Bundled Data Logic, which delivers high throughput with simple hardware but is sensitive to timing variations. Additionally, a two-phase clocked Mesh NoC using bonded Bundled Data Logic has been proposed to further enhance latency performance.

The choice of routing protocol is another significant factor influencing NoC performance. More complex routing algorithms can lead to increased router complexity, which in turn raises power consumption and chip area. Conversely, simpler routing protocols may be more energy-efficient and cost-effective but could result in suboptimal traffic management across the network. Another key aspect of NoC design is buffer size, which helps store data packets and prevent packet loss or misrouting. However, larger buffers increase both power consumption and the area required on the chip. For instance, input buffers in some designs can occupy a significant portion of the network area, as seen in cases where they account for up to 75% of the total area.

The proposed router designs in this work have been thoroughly evaluated in terms of their area usage and operating frequency. By using distributed control mechanisms, these routers can function autonomously without the need for complex handshake protocols, which enhances both efficiency and scalability. The architecture of the proposed routers, consisting of Input Channels, a Crossbar Switch, and Output Channels, plays a key role in enabling effective data routing and communication within the NoC, ensuring smooth interaction between the different cores and routers in the system.

2. Body of Paper

The proposed Network-on-Chip (NoC) architecture consists of two primary components: a low-power, area-efficient router design and a Network Interface (NI) accompanied by a traffic generator.

- **Router Design:** In NoC systems, routers are essential switching elements that ensure the efficient transmission of data packets from the source core to the destination core. In a mesh-based NoC, each router is equipped with four directional ports—North, East, West, and South—which enable communication between adjacent routers. Additionally, each router includes a local port that connects to the core it serves. When a data packet is generated by a source core, it is routed through the network towards the router associated with the destination core.
- **IP Core of Router Design:** In NoC architectures, the router or switch serves as a central element, managing data flow and ensuring efficient communication between different cores

or processing elements. The proposed router architecture, known as the Multi-Lane Parallel Router (MLPR), is illustrated in Fig. 1. The MLPR is designed with multiple ports, each with a specific function. Four of these ports are directional—North (N), East (E), South (S), and West (W)—allowing the router to direct outgoing data packets based on the location of the destination core. For example, if a packet is destined for a core located to the east, the router will send the packet via the East port.

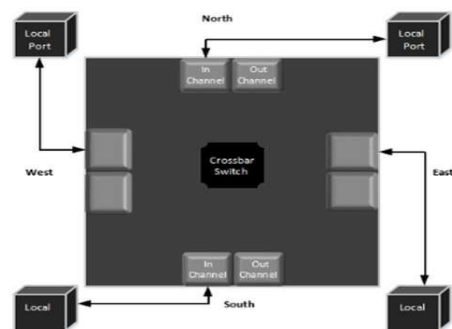


Fig. 1. Architecture of Multi Local Port Router (MLPR)

Implementation of MLPR

As illustrated in Fig. 2, the architecture of the proposed Multi-Local Port Router (MLPR) consists of three main components: the Input Channel, Cross Switch Matrix, and Output Channels. Each component plays a vital role in facilitating data routing and transfer within the router.

- **Input Channel:** The Input Channel is critical to the MLPR, acting as the entry point for incoming data packets from connected processing elements or design cores. Its primary function is to receive data packets and prepare them for processing, which includes tasks such as packet segmentation and header extraction to obtain routing information. Within the MLPR framework, the Input Channel is essential for managing the flow of incoming data packets through the router. Fig. 3 depicts the block diagram of the Input Channel, which comprises a Buffer, Control Logic, and XY Routing.
- **Buffer:** Each port in the Input Channel is equipped with a dedicated buffer that temporarily holds incoming data as it arrives at the router. These buffers are designed with a First-In-First-Out (FIFO) structure, featuring a depth of 16 bits and a width of 8 bits. When data arrives, it is stored in the FIFO buffer until it is processed and sent to the appropriate output channel.
- **Control Logic:** The Input Channel includes a distinct area of control through its specialized Control Logic, implemented as a

Finite State Machine (FSM) Controller. This Control Logic effectively oversees various data transfer operations within the Input Channel. It manages read and write operations for the FIFO buffer and coordinates request and grant signals to ensure smooth data transfer between the input and output channels. Additionally, the Control Logic handles acknowledgment signals from neighboring routers or processing elements, ensuring efficient communication throughout the network.

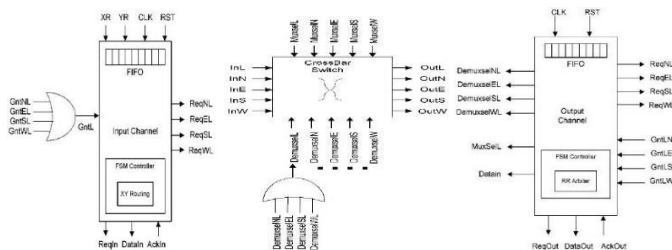


Fig. 2. MLPR Input Channel, Crossbar Switch, and Output Channel

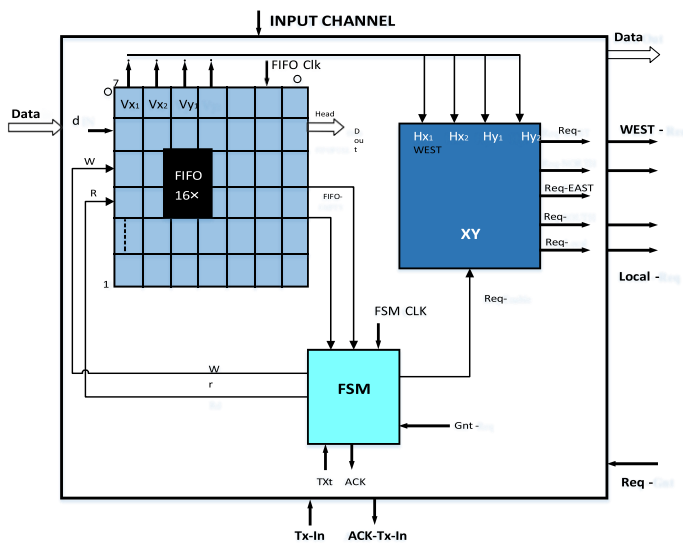


Fig. 3. Input channel block diagram

Data Transfer Procedure in the Output Channel

The data transfer process within the Output Channel can be summarized as follows:

- **8-Bit FIFO:** Each Output Channel features a dedicated 8-bit FIFO with a depth of 16, which temporarily holds data packets before they are sent to neighboring routers or processing elements. When multiple requests from various input channels arrive, a Round Robin Arbiter (RRA) effectively manages the arbitration process, selecting the most suitable request for processing and storing it in the FIFO.
- **Control Logic (FSM):** Within the Output Channel, an advanced Control Logic, implemented as a Finite State Machine (FSM), plays a key role in making important arbitration

decisions amid numerous incoming requests from different input channels. Utilizing the Round Robin Arbiter, the FSM ensures fair data transfer among all input channels, promoting a balanced system. Once the RRA approves a request, the FSM activates the control bit lines of the Crossbar Switch to establish the necessary connection for seamless data transfer.

- **Round Robin Arbiter (RRA):** The RRA acts as a neutral arbitrator, selecting and prioritizing data requests from the various input channels. By ensuring that all input channels have an equal chance to participate in data exchange with the Output Channel, the RRA promotes fairness in the routing process.
- **Handshake Mechanism:** After a data packet is received and stored in the FIFO, the FSM initiates the transmission process to the neighboring router through a handshake mechanism. This ensures reliable data transfer and proper synchronization between the Output Channel and the neighboring router.

- **Crossbar Switch Control:** The Output Channel controls the configuration of the control bit lines within the Crossbar Switch, creating the necessary connections for data transfer from the input channel to the Output Channel. This crucial function guarantees that data is routed correctly and efficiently to its intended destination.

The Output Channel is vital for managing data transfers from the router to neighboring routers or processing elements. Its Control Logic and Round Robin Arbiter work together to ensure fair and efficient data routing within the MLPR router and the Network-on-Chip architecture.

XY Routing Process in the Input Channel

The XY Routing mechanism is essential for the MLPR router, directing data packets based on their destination coordinates. The routing process involves several key stages:

- **Horizontal Displacement:** When the Input Channel's FIFO is full, it compares the X-coordinate of the target router (Hx) with the local X-coordinate. If Hx is greater, the packet is routed to the East port; if it's smaller, the packet goes to the West port.
- **Vertical Displacement:** If Hx matches the local X-coordinate, the packet is ready for vertical movement. The Y-coordinate of the destination router (Hy) is then compared with the local Y-coordinate. If Hy exceeds Y , the packet moves to the North port; if it's less, it proceeds to the South port.
- **Final Destination:** If Hy matches the router's Y-coordinate, the packet has reached its

destination and is sent to the local port of the router, completing the routing process.

This XY Routing strategy optimizes resource usage in the router design. By directing packets horizontally until they reach the correct column and then vertically to the target router, there's no need for the North or South input ports to access the East or West output ports. This simplification allows the FSMs of the East and West output channels to be streamlined, reducing area utilization and minimizing the number of clock cycles required to process requests. As a result, the Multi-Local Port Router achieves significant performance benefits with minimal overhead, making it an efficient solution for data routing tasks.

Traffic Generator

The Traffic Generator (TG) simulates data flows originating from various sources to the communication architecture. A deterministic TG provides a structured model of the communications emitted by the IP blocks connected to the Network-on-Chip (NoC), based on traces left by these blocks. This type of TG can generate precise transactions over time, packet size, and idle periods that reflect the behavior of the connected IPs. It is specifically designed for a complete system configuration (including the type and number of nodes) and for particular applications. The key advantages of deterministic TGs include high accuracy and increased efficiency for emulation compared to simulating all traffic.

In this work, deterministic traffic generators are employed to evaluate the performance of the NoC in spectral applications. Two packet formats are used: one for data in an FPGA (for implementations on a single FPGA architecture) and another for multi-FPGA architectures. Each packet in our emulation platform is structured into two primary sections: a header and a data portion. These sections contain essential information for proper functionality. The header includes the destination node address (Dest), the source address (Source), and the Initiator Clock (Clk_init), which is crucial for latency measurement. The Clk_init data corresponds to the clock cycle when the packet is dispatched. Additionally, the packet size (Sz_pkt), the Ext_cpt (applicable only for multi-FPGA setups, indicating the number of cycles for inter-FPGA transfers), and the total number of transmitted packets (Nb_pkt) are also included.

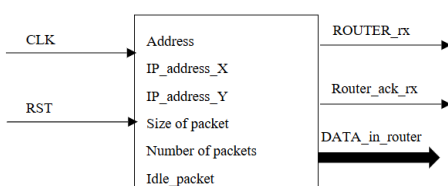


Fig. 4. Signals and Parameters for Generic Traffic Generators

As shown in Fig. 4, the signals and parameters governing the Generic Traffic Generators illustrate the integration of the TG within our system. The TG generates control signals, such as router_rx and router_ack_rx, while producing data packets at the data_in output. The packet size is synchronized with the bus size to optimize data transfer efficiency. The various packet quantities and formats are determined by the specific traffic scenarios detailed in the Data_transfer package discussed later in this paper. The TG is implemented in generic VHDL and is strategically incorporated into the TG and TR library within the flow.

3. CONCLUSIONS

In conclusion, we explore the potential of Field-Programmable Systems-on-Chip (FPSoCs) as reliable and efficient digital systems, designed for modern applications with high computational demands and compact form factors. To address the limitations of traditional bus-based and point-to-point on-chip communications in System-on-Chip (SoC) designs, the Network-on-Chip (NoC) has emerged as the preferred interconnect solution. However, extensive research is necessary to thoroughly investigate the design possibilities of FPGA-based NoCs and develop more effective solutions to current NoC challenges.

Our research makes a significant contribution to the field of FPGA-based NoCs by proposing efficient, area-optimized designs for NoC routers. Rigorous implementation and evaluation of our proposed router design on Xilinx Spartan 3 FPGAs have clearly demonstrated its feasibility and potential for real-world applications in FPSoCs. By continuously exploring the design opportunities of FPGA-based NoCs and introducing more streamlined solutions, we can anticipate further enhancements in the performance and capabilities of FPSoCs, especially for computationally intensive applications.

ACKNOWLEDGEMENT

We would like to express our heartfelt gratitude to Mrs. Malashree K S, for their exceptional guidance, support, and expertise throughout this research project. We are deeply indebted to P E S institute of technology and management and its faculty members for providing us with the necessary resources and facilities that enabled us to conduct our study. We also appreciate the insightful discussions and feedback from our colleagues and peers, which significantly contributed to the advancement of this research. Lastly, we extend our sincere appreciation to our families and friends for their unwavering support and encouragement throughout this endeavor. Their love and motivation played a vital role in our success.

REFERENCES

1. Jain, A., Dwivedi, R. K., Alshazly, H., Kumar, A., Bourouis, S., & Kaur, M. (2022). Design and simulation of ring network-on-chip for different configured nodes. *Computers, Materials & Continua*, 71(2), 4085-4100.
2. Kumar, N. A., Priyan, S. V., Venkatramana, P., & Nandan, D. (2022). Routing Strategy: Network-on- Chip Architectures. *VLSI Architecture for Signal, Speech, and Image Processing*, 167-197.
3. Naqvi, M. R. (2021). Low power network on chip architectures: A survey. *Computer Science and Information Technologies*, 2(3), 158-168.
4. Yazdanpanah, F. (2023). A two-level network-on-chip architecture with multicast support. *Journal of Parallel and Distributed Computing*, 172, 114-130.