Area-Efficient and Low-Latency On-Chip Communication via a Hybrid Circuit-Packet Switched NoC Router

K. Venugopal Rao[1], **Venkata Reddy Adama**[2]*, **G. Koteswar Rao**[3]

[1]Professor & Head, Department of Electronics and Communication Engineering, Vaageswari College of Engineering, Karimnagar, Telangana, India
[2]Department of Electronics and Communication Engineering, Jyothishmathi Institute of Technology and Science, Thimmapur, Karimnagar, 505527, Telangana India
[3]Associate Professor, Department of Electronics and Communication Engineering, Vaagdevi College of Engineering, Warangal, Telangana, India
*Corresponding: Venkata Reddy Adama (venkat7641@gmail.com)

Abstract

Efficient Network-on-Chip (NoC) router design is critical in modern many-core System-on-Chip (SoC) architectures, where scalable communication directly dictates overall system performance. Conventional NoC architectures rely on either circuit switching or packet switching. While circuit switching provides predictable paths, it suffers from high setup latency and poor resource utilization. Conversely, packet switching offers flexibility but introduces congestion, heavy buffering requirements, and elevated dynamic power consumption. To overcome these limitations, this work proposes a hybrid NoC router architecture that integrates Virtual Circuit Switching (VCS) with traditional circuit and packet switching mechanisms. The proposed design enables multiple virtual circuit-switched connections to multiplex over a single physical channel, drastically improving channel utilization and reducing idle bandwidth. By dynamically allocating virtual circuits based on real-time traffic demands, the router minimizes setup delays while maintaining a highly fluid data flow. Physical implementation results demonstrate that this hybrid strategy optimizes buffer usage and control logic complexity, yielding superior area efficiency. Compared to conventional implementations, the design achieves lower latency, reduced power consumption, and enhanced scalability, providing a highly efficient, energy-aware communication backbone for next-generation, high-performance SoCs.

Keywords: Network-on-Chip, Hybrid Router Architecture, Virtual Circuit Switching, Packet Switching, Many-Core SoC, Channel Utilization, Area-Efficient Hardware.

1. Introduction

NoC is a key component in chip multiprocessors (CMPs) as it supports communication between many cores. However, NoCs for modern chips contribute up to 30% of the chips overall power budget. Also, as chips continue to shrink, the contribution of NoC power is projected to increase. Routers. NoC router comprises a number of input ports, a number of output ports, a switching matrix connecting the input ports to the output ports, and a local port to access the IP core connected to this router. The router also contains a logic block that implements the flow control policies.

- The first and most important ones are the links that physically connect the nodes and implement the communication.
- The second block is the router, which implements the communication protocol.
- The last building block is the network adapter (NA) or network interface (NI). This block makes the logical connection between the IP cores and the network

On-chip routers are a crucial component in System-on-Chip (SoC) designs, facilitating communication between various components on the chip. Improving latency and throughput in on-chip networks is essential for enhancing the overall performance of integrated circuits. Some strategies and considerations include:

Topology Design: Choosing an efficient network topology is critical. Mesh, torus, or hierarchical topologies are commonly used, and the selection depends on the specific requirements of the application.

Routing Algorithms: Implementing efficient routing algorithms can significantly impact latency and throughput. Adaptive or dynamic routing algorithms can adapt to changing traffic patterns and optimize communication paths.

Quality of Service (QoS): Implementing QoS mechanisms can prioritize certain types of traffic, ensuring that critical data receives preferential treatment in terms of routing and resources.

The paper presents an innovative approach to designing NoC routers that prioritize area efficiency while facilitating seamless D2D communication. Existing NoC architectures typically employ either circuit switching or packet switching techniques, each with its own limitations. Circuit switching can lead to high latency due to setup time, while packet switching may suffer from increased power consumption and congestion. This work introduces a hybrid scheme for NoC, aiming to significantly reduce latency and power consumption.

2. LITERATURE SURVEY

The rapid advancement of many-core Systems-on-Chip (SoCs) has prompted a critical re-evaluation of physical on-chip communication frameworks, architectural reliability, and hardware acceleration paradigms. To optimize network utilization and throughput, various high-performance Network-on-Chip (NoC) architectures have been introduced. Arulananth et al. [1] and Prasad et al. [10] presented a High-Speed Virtual Logic Network-on-Chip (HSVLN) and associated routing algorithm (HSRA) that utilizes diagonal paths to reduce power dissipation by up to 22% and lower communication latency compared to standard 3D topologies. Baseline multi-bit transmission systems and five-port configurations using standard XY or round-robin scheduling algorithms have been rigorously modeled in VHDL and verified on FPGAs to guarantee structural stability [2], [3]. To mitigate localized channel contention and packet congestion, dynamic bandwidth allocation protocols using real-time injection monitoring have been deployed [5], alongside hybrid buffering techniques like *IP Den 2.0* to minimize packet deflection penalties [8]. Furthermore, comprehensive parameter reviews highlight that minimizing latency and preventing deadlock states remain the core objectives of modern routing development [9].

Simultaneously, mitigating hardware faults induced by aggressive nanometer-scale fabrication has driven deep research into architectural resilience and data security. Shafique et al. [4] introduced *NoCGuard*, leveraging stage-level resource borrowing and structural bypass lines to achieve a 5.53-fold improvement in Mean Time to Failure (MTTF). This emphasis on stage-level component paring is shared by Hussain et al. [5], whose fault-tolerant allocation mapping yielded a peak Silicon Protection Factor (SPF) of 24.8. Moving toward fully decentralized recovery, Khalil et al. [6] implemented autonomous self-healing circuits within port buffers to recover from runtime circuit faults without external host intervention. In tandem with reliability, architectural security has surfaced as a critical parameter; Sarihi et al. [7] proposed an anonymous source-routing algorithm operating on encrypted address headers, successfully neutralizing side-channel snooping threats with minimal chip-wide power and area overheads.

Proposed System

Network on chip router is a key component in chip multiprocessors. As it supports communication between many cores. This plays a crucial role in facilitating efficient data transmission and data exchange among various components. It offers a structured and scalable approach to interconnect many processing elements. Figure 1 has two different architectures for interconnection of processing elements (PEs) in a system with a small number of processor cores. The architectures are based on traditional interconnection methods: a shared bus (Figure 1a) and a fully connected crossbar (Figure 1b).

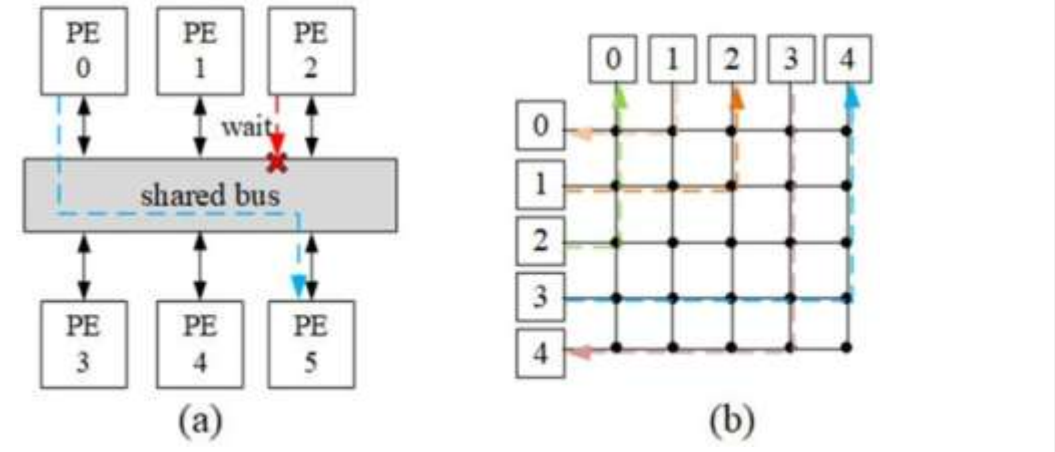

Figure 1. Shared bus and a fully connected crossbar.

Figure 1 shows the process of shared bus architecture. In the shared bus architecture, all processing elements (PEs) share a common transmission medium or bus. The shared bus is one of the NoC. topologies, and it involves a single communication bus that is shared among multiple processing elements (PEs) or IP cores on the chip. PEs communicate with each other by sending data over the shared bus. However, as the number of PEs connected to the bus increases, the bus can quickly reach a saturation state. The shared bus has poor scalability because when more PEs are connected, the bus traffic increases, making it difficult to achieve higher bandwidth.

The shared bus has relatively low transmission efficiency since it does not support parallel data transmission. In the illustrated scenario, PE2 with low priority has to wait when the bus is occupied, leading to potential delays. In the fully connected crossbar architecture, all processing elements are connected to each other with multiplexers. A crossbar connection refers to a type of network topology where multiple input lines can be connected directly to multiple output lines. In the context of on-chip communication or NoC designs, a crossbar switch is often used to establish direct and simultaneous connections between different processing elements or IP cores.

The crossbar provides direct connections between every pair of PEs, allowing for parallel data transmission and avoiding the issues faced by the shared bus. The advantage of supporting parallelism, the crossbar architecture has poor scalability. As the number of PEs increases, there is a significant area overhead and power consumption associated with implementing a fully connected crossbar. The complexity and resource requirements of a fully connected crossbar make it less suitable for systems with a large number of processing elements.

Packet switching

Packet Switching in computer networks is a method of transferring data to a network in the form of packets. In order to transfer the file fast and efficiently manner over the network and decrease the latency, the data is broken into small pieces of variable length, called Packet. At the destination, all these small parts have to be reassembled, belonging to the same file. A packet is formed with its original information from the small packets.

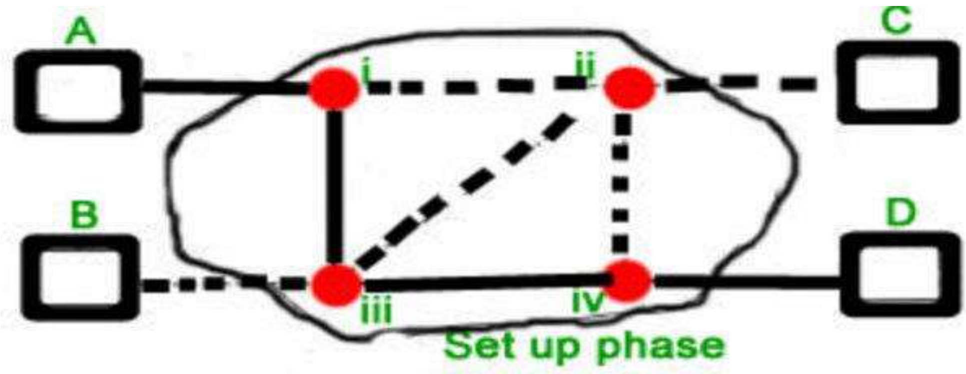

Figure 2. Connection-oriented Packet Switching (Virtual Circuit).

Figure 2 shows the process of Packet Switching uses the Store and Forward technique while switching the packets; they are forwarding the packet each time when it moves, first stores that packet then forwards. This technique is very beneficial because packets may get discarded at any hop for some reason. More than one path is possible between a pair of sources and destinations they independently travel through the network. In other words, packets belonging to the same file may or may not travel through the same path. If there is congestion at some path, packets are allowed to choose different paths possible over an existing network. In packet switching the data is divided into small packets which allow faster movement of data. Each packet contains two parts that is Header and Payload, the header on each packet conation information and the data which is send by the sender is called payload.

- More efficient in terms of bandwidth, since the concept of reserving a circuit is not there.
- Minimal transmission latency.
- More reliable as a destination can detect the missing packet.
- More fault tolerant because packets may follow a different path in case any link is down, Unlike Circuit Switching.
- Cost-effective and comparatively cheaper to implement.

Route Computation Using SSA

In NoC architectures, the Static Switch Allocator (SSA) serves as a vital component for efficiently routing data packets between input and output ports. Route computation using SSA involves a systematic process aimed at establishing static paths to facilitate reliable and predictable packet transmission within the network.

Let's explore each step-in detail:

Identification of Input and Output Ports: The route computation journey with SSA commences with the meticulous examination of the input and output addresses embedded within incoming data packets. These addresses serve as the digital coordinates guiding the data flow, akin to the addresses on envelopes guiding mail delivery. SSA diligently deciphers these addresses to identify the source (input port) and destination (output port) of the data packets. This initial step lays the foundation for subsequent routing decisions, as SSA determines the ingress and egress points within the NoC architecture.

Evaluation of Available Paths in the Crossbar Switch: Once the input and output ports are identified, SSA embarks on a quest to explore the labyrinthine pathways within the Crossbar Switch—the central switching element in the NoC. This step involves a comprehensive assessment of the network's topology and channel availability, akin to charting a course through a complex maze. SSA meticulously evaluates various pathways within the Crossbar Switch, ensuring that the selected route can accommodate the data flow without encountering bottlenecks or contention issues. This evaluation is essential for establishing an efficient communication channel between the input and output ports.

Validation of Path Integrity and Avoidance of Overlap: With potential paths identified, SSA transitions into a phase of rigorous validation to verify the integrity and reliability of the selected route. This validation process is akin to conducting a thorough inspection to ensure the structural integrity of a bridge before allowing vehicles to cross. SSA scrutinizes the chosen path, meticulously verifying its compatibility with existing data flows and pre-empting any potential overlap or collision scenarios. By rigorously vetting

the selected route, SSA fortifies the network's defences against data corruption, loss, or misrouting, ensuring seamless packet transmission.

Finalization of the Primary Static Route via Packet Switching: Upon completing the validation phase, SSA proceeds to crystallize the primary static route using sophisticated packet switching techniques. Packet switching involves encapsulating data into discrete packets, each containing routing information to guide its traversal through the network. SSA leverages packet switching to establish a fixed pathway between the input and output ports, akin to laying down a digital highway for data transmission. By adhering to the predefined static routes, SSA ensures efficient and reliable data transfer within the NoC architecture.

Finally, route computation using SSA in NoC systems involves a series of meticulous steps, including port identification, path evaluation, validation, and route finalization. By allocating static paths based on predetermined rules and priorities, SSA ensures reliable and efficient data transmission within the network, contributing to the overall performance and scalability of NoC architectures. Through its systematic approach, SSA facilitates the seamless routing of data packets, enabling the efficient communication of various components within complex computing systems.

Parallel Virtual Switch Allocator

In the event of a damaged path between an input unit and an output unit within a NoC architecture, the risk of data loss due to packet loss becomes a critical concern. To mitigate this risk and maintain continuous data transmission, a temporary virtual path can be created using a Virtual Switch Allocator (VSA), which combines virtual switching and packet switching techniques. The step wise operation of VSA given as follows:

Step1: Identification of Path Damage: The journey of the Virtual Switch Allocator begins with the detection of path damage between input and output units within the NoC. This could occur due to various factors such as physical faults, congestion, or interference. Like a vigilant guardian, the VSA continuously monitors the network for signs of trouble, swiftly identifying any disruptions that may impede the flow of data. Once path damage is detected, the VSA initiates proactive measures to mitigate its impact and prevent data loss.

Step 2: Understanding the Consequences of Packet Loss: The ramifications of path damage extend beyond mere inconvenience; they can lead to data loss, jeopardizing the integrity of critical information traversing the network. Data packets may become stranded or corrupted, resulting in gaps or inconsistencies in the transmission process. This can have far-reaching implications, particularly in mission-critical applications where data accuracy and timeliness are paramount. The Virtual Switch Allocator recognizes the gravity of packet loss and endeavours to address it promptly and effectively.

Step 3: Creation of Temporary Virtual Paths: To circumvent the adverse effects of path damage and mitigate the risk of data loss, the Virtual Switch Allocator employs a proactive approach by creating temporary virtual paths. These virtual pathways act as alternative routes, bypassing the damaged segments of the network and ensuring uninterrupted data flow between input and output units. Think of it as paving a temporary road detour to bypass a blocked highway, allowing traffic to continue moving smoothly until the main route is restored. By dynamically allocating resources and rerouting data packets, the VSA minimizes disruptions and maintains the continuity of communication within the NoC.

Step 4: Integration of Virtual Switching and Packet Switching: At the heart of the Virtual Switch Allocator lies the seamless integration of virtual switching and packet switching techniques. Virtual switching enables the creation of temporary virtual paths, establishing dynamic connections between input and output units to circumvent path damage. This flexibility allows the VSA to adapt swiftly to changing network conditions and mitigate the impact of disruptions. Meanwhile, packet switching facilitates the efficient transmission of data packets along these virtual paths, ensuring timely delivery and optimal utilization of network resources. By combining the strengths of both virtual and packet switching, the VSA optimizes data transmission within the NoC, enhancing performance and reliability.

Overall NoC Architecture

The step wise operation of overall NoC given as follows:

Step1: -In accordance with the illustration provided, the input data comprises four distinct entities: input A, input B, input C, and input D. Each of these inputs represents a unique stream of data that is directed towards the input unit for processing. Whether it be input A, input B, input C, or input D, each serves as a source of information that contributes to the overall dataset being managed by the system.

Step 2: -The input unit serves as a repository for incoming data, housing a memory bank comprised of multiple registers. Each piece of input data is allocated a dedicated register within this memory bank, ensuring that no data is lost during processing. By utilizing these registers, the input unit effectively manages and stores the required input data without any risk of loss or corruption. This systematic organization allows for efficient data handling, as each input is securely stored in its designated register, ready to be accessed and utilized as needed.

Integral to the functionality of the input unit is its implementation of a clock synchronization mechanism. This mechanism ensures seamless coordination between the various input ports, synchronizing the timing of data transfer across the unit. With clock synchronization in place, the input unit can efficiently manage the flow of data from one input to multiple input ports, maintaining consistency and reliability throughout the process. By synchronizing the clocks of different inputs, the unit ensures that data is processed and transferred in a controlled and synchronized manner, optimizing overall system performance.

Each individual input within the unit is equipped with its own clock synchronization mechanism, further enhancing the unit's ability to manage multiple streams of data simultaneously. This decentralized approach to clock synchronization ensures robustness and flexibility, allowing for independent control and coordination of data flow within the input unit. Through the harmonized operation of dedicated registers and synchronized clocks, the input unit effectively fulfils its role as a reliable and efficient component of data processing systems, facilitating seamless communication and data management across various input channels.

Step 3: -In the computational journey facilitated by the SSA, a meticulous process unfolds to navigate data from its source to its intended destination. With the presence of four input ports, decisions must be made regarding the most efficient route from the router to the desired output. Whether it's steering data from input A to input B or any other combination, the SSA undertakes the task of path selection, conducting internal assessments to evaluate available options.

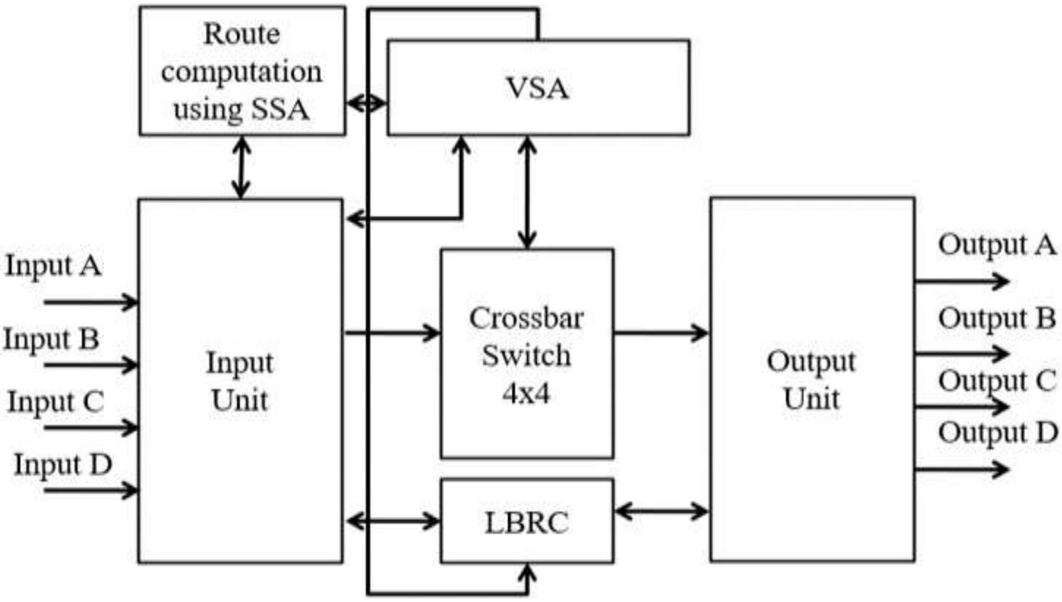

Figure 3. Architecture of the router.

Internally, within the SSA framework, a thorough examination takes place to scrutinize the input and output ports involved in the data transfer process. This scrutiny extends to identifying any fixed or common paths that may streamline the routing process. Through this internal analysis, the SSA methodically assesses potential pathways, strategically allocating resources to optimize data transmission efficiency. This strategic allocation of pathways is a hallmark of the Static Switch Allocator, guiding the data along the most suitable routes within the system. As the route computation progresses, the SSA ventures into the realm of the Crossbar Switch, exploring the available pathways from input A to input B and beyond. This exploration involves scrutinizing each potential route, identifying any instances of path invalidity, congestion, or overlap. Despite the careful deliberation, it's essential to acknowledge that occasional data overlap or loss may occur along the selected pathway. While the SSA remains steadfast in its function, it's important to recognize that any data loss incurred does not fall within the realm of the SSA's responsibility, as its primary role is to ensure efficient route allocation and management within the system.

Step 4: -In the realm of data management, a critical aspect lies in ensuring the seamless flow of information across various basic pathways. This task is effectively addressed through the utilization of the VSA method. Operating as a virtual switch allocator, the VSA method plays a pivotal role in overseeing data flow dynamics within the system. In instances where pathways encounter damage or disruption, the VSA swiftly responds by initiating the creation of virtual or temporary paths, thereby safeguarding the continuity of data transmission.

The operation of the VSA method hinges on its ability to swiftly adapt to changing circumstances within the data network. In the event of path damage, the VSA method employs a strategic combination of virtual switching and packet switching techniques to rectify the issue. By creating temporary virtual paths, the VSA method effectively circumvents potential data loss that may arise due to packet loss along damaged pathways. This proactive approach ensures that data continues to flow uninterrupted, maintaining the integrity and reliability of the system's communication infrastructure.

It is crucial to recognize that the absence of swift intervention in addressing damaged pathways can result in detrimental consequences, including data loss and system inefficiencies. Hence, the implementation of the VSA method stands as a proactive measure to mitigate such risks. By promptly creating virtual pathways as needed, the VSA method exemplifies its role as a guardian of data integrity, preserving the smooth operation of data flow within the system. Thus, through the strategic interplay of virtual and packet switching techniques, the VSA method effectively fulfils its mandate of ensuring uninterrupted data transmission in the face of pathway disruptions.

Step 5: -In navigating the intricate network of data exchange, the Crossbar switch serves as a critical tool, facilitating the creation of efficient routes for transferring information among various components. As a multi-level packet switching network, the Crossbar switch embodies adaptability and versatility, offering a dynamic platform for establishing pathways. Within its architecture, data transfer paths are meticulously crafted, capable of taking on various forms, whether temporary or virtual, to accommodate the evolving demands of data transmission within the system.

The essence of the Crossbar switch lies in its ability to orchestrate seamless communication between different components by dynamically configuring pathways. These pathways are essential conduits for the smooth flow of data, ensuring that information reaches its destination swiftly and reliably. Whether forging temporary connections to address immediate needs or establishing virtual links for future scalability, the Crossbar switch plays a pivotal role in optimizing data transfer efficiency and enabling seamless communication across the network.

Step 6: -When Input A seeks to transmit data to Output B and C simultaneously, it entails a complex data transfer scenario involving one-to-many and many-to-many connections. However, amidst this process, there's a risk of data loss, even with the implementation of virtual switching mechanisms, particularly when dealing with temporary connections across multiple devices. To mitigate such losses effectively, the utilization of LBRC operation becomes imperative. LBRC operation is adept at managing multiple input and output connections concurrently, thereby addressing the intricacies of data transfer within the system.

In the realm of LBRC operation, the focus lies on ensuring robust data transmission without succumbing to potential losses, especially in cases where junction issues may arise. LBRC takes proactive measures to safeguard data integrity by employing a lookahead mechanism. This mechanism allows LBRC to pre-emptively identify any impending data loss due to invalid paths or overlapping virtual connections. In such instances, LBRC promptly alerts the Virtual Switch Allocator (VSA) to rectify the situation by generating alternative pathways for communication among multiple devices.

The collaboration between LBRC and VSA proves instrumental in orchestrating efficient data transfer processes within the system. By seamlessly coordinating input and output connections, LBRC and VSA work in tandem to ensure data integrity and reliability. Through the implementation of bypass connections, wherein 99% of data remains unscathed, LBRC establishes itself as a pivotal component in the system's architecture, effectively mitigating the risks associated with data loss during complex data transfer operations.

Step 7: -Similar to the input unit, the output unit serves as a crucial component in the data processing system, albeit with a distinct function. While both units handle data transfer, the output unit differs in its approach by exclusively storing output data in memory and registers. Instead of directly processing the data, the output unit focuses on efficiently storing the transmitted information and subsequently transmitting it to output devices for further processing or display. This distinction underscores the specialized role of the output unit in managing data flow within the system, ensuring that output data is securely stored and ready for subsequent processing or presentation.

4. Result and Discussion

Table 1 offers a comprehensive comparison between an existing method and a newly proposed alternative, both tailored for a system with N=8. These metrics provide invaluable insights into the efficacy and potential benefits of adopting the proposed method over the established approach. Initially, the table reveals a striking discrepancy in the utilization of Look-Up Tables (LUTs) between the two methods. While the existing method relies heavily on 492 LUTs, the proposed method operates with a significantly leaner count of just 39. This substantial reduction underscores the proposed method's ability to achieve comparable functionality with far fewer resources, hinting at its potential for optimizing hardware utilization and design efficiency.

Table 1. Performance comparison of existing and proposed method for N=8

Metric	Existing Method	Proposed Method
LUT	492	39
I/O	35	69
Total Power	9.503	0.903
Static Power	0.130	0.130
Dynamic Power	9.374	0.791
Logic Power	4.914	0.076
Signal Power	4.395	0.586
Net Delay	19.407	11.301
Logic Delay	14.661	4.428
Total delay	34.057	15.709

In addition to resource allocation, the comparison delves into Input/Output (I/O) requirements, where the proposed method demonstrates a substantial improvement by nearly doubling the count from 35 to 69. This enhancement suggests a broader scope of connectivity and data handling capabilities, likely expanding the applicability and versatility of the proposed method in real-world scenarios. Moreover, the evaluation extends to power consumption, a critical consideration in modern hardware design. The proposed method emerges as a clear winner in this aspect, boasting a drastic reduction in total power consumption from 9.503 to a mere 0.903. This reduction is particularly pronounced in dynamic power (from 9.374 to 0.791) and logic power (from 4.914 to 0.076), indicating significant strides towards energy efficiency and reduced operational costs.

Furthermore, the proposed method exhibits notable improvements in signal power management, as evidenced by the decrease from 4.395 to 0.586. This reduction underscores the efficacy of the proposed method in optimizing signal routing and transmission, further contributing to overall power efficiency. Beyond power considerations, the comparison also sheds light on delay characteristics, crucial for system responsiveness and performance. Here, the proposed method excels, showcasing substantial reductions in both net delay (from 19.407 to 11.301) and logic delay (from 14.661 to 4.428). The cumulative effect is reflected in the total delay, which experiences a significant decrease from 34.057 to 15.709, indicative of improved data propagation and processing efficiency. Finally, the findings presented in Table 6.1 underscore the compelling advantages of the proposed method across various critical metrics, encompassing resource utilization, power consumption, and delay characteristics.

These results collectively advocate for the adoption of the proposed method as a promising avenue for achieving enhanced hardware performance and efficiency in systems designed for N=8.

Table 2. Performance comparison of Existing and Proposed method for N=16

Metric	Existing Method	Proposed Method
LUT	982	70
I/O	65	133
Total Power	19.383	1.929
Static Power	0.167	0.114
Dynamic Power	19.216	1.816
Logic Power	9.784	0.145
Signal Power	9.316	1.426
Net Delay	22.563	15.078
Logic Delay	14.157	4.089
Total delay	36.400	19.118

Table 2 provides a detailed performance comparison between the existing method and a proposed alternative for systems with *N*=16. The proposed method significantly outperforms the existing one across various metrics. Currently, the existing method requires 982 Look-Up Tables (LUTs), whereas the proposed method utilizes a much leaner count of only 70 LUTs. This reduction streamlines resource allocation, indicating improved hardware efficiency. In terms of Input/Output (I/O) efficiency, the existing method handles 65 inputs/outputs, whereas the proposed method doubles this count to 133. This enhancement suggests improved connectivity and data handling capabilities, crucial for complex system operations. Regarding power consumption, the existing method consumes a total power of 19.383, with a significant portion attributed to dynamic power (19.216). In contrast, the proposed method exhibits substantially lower total power consumption at 1.929, with dynamic power reduced to 1.816. This reduction underscores the proposed method's energy efficiency. Additionally, the proposed method shows significant improvements in delay characteristics. For instance, the net delay is reduced from 22.563 to 15.078, and the total delay decreases from 36.400 to 19.118. These improvements signify faster data propagation and processing, highlighting the proposed method's potential to enhance system responsiveness and overall performance.

Table 3. Performance comparison of Existing and Proposed method for N=32

Metric	Existing Method	Proposed Method
LUT	1964	135
I/O	125	261
Total Power	41.478	3.902
Static Power	0.388	0.117
Dynamic Power	41.090	3.785

Logic	19.591	0.290
Signal	21.283	0.475
Net Delay	26.865	14.438
Logic Delay	14.907	4.091
Total Delay	41.611	18.492

Table 3 illustrates a comprehensive comparison between an existing method and a proposed alternative tailored for systems with N=32. Currently, the existing method utilizes 1964 Look-Up Tables (LUTs), while the proposed method operates with a more streamlined count of only 135 LUTs. This reduction suggests a more efficient resource allocation strategy, potentially leading to more compact and cost-effective hardware designs. Moreover, the proposed method significantly enhances Input/Output (I/O) efficiency, with 261 I/O operations compared to the existing method's 125. This improvement indicates enhanced connectivity and data handling capabilities crucial for accommodating the diverse input and output requirements of larger systems. Regarding power consumption, the existing method exhibits a total power consumption of 41.478, with dynamic power accounting for a significant portion at 41.090. In contrast, the proposed method demonstrates substantially lower total power consumption at 3.902, with dynamic power reduced to 3.785. This reduction highlights the proposed method's energy efficiency and potential for reducing operational costs. Additionally, the proposed method exhibits significant improvements in delay characteristics. For instance, the net delay decreases from 26.865 to 14.438, and the total delay decreases from 41.611 to 18.492. These improvements signify faster data propagation and processing, indicating the proposed method's ability to enhance system responsiveness and overall performance in N=32 configurations.

5. Conclusion

In conclusion, the hybrid connected NoC router demonstrates a commendable advancement in addressing latency and throughput challenges within integrated circuits. By amalgamating various routing strategies, this innovative approach seeks to strike a balance between efficient data delivery and reduced communication delays. The integration of both deterministic and adaptive routing mechanisms allows the router to adapt dynamically to varying traffic conditions, optimizing latency in diverse scenarios. The significance of packet switching emerges as a focal point in this, highlighting its role in breaking down data into smaller packets for rapid transmission and subsequent reassembly at the destination. The advantages of packet switching over circuit switching, including improved bandwidth, reduced latency, enhanced reliability, fault tolerance, and cost-effectiveness, further emphasize its pivotal role in modern NoC designs. The distinction between the SSA and the VSA within NoC systems adds depth to the discussion, showcasing the meticulous planning involved in crafting static paths for dependable packet transmission and the dynamic responsiveness of virtual paths in the face of potential disruptions. This seamless blend of virtual and packet switching techniques demonstrates the adaptability and resilience required in contemporary NoC architectures. However, it's essential to recognize that the performance gains achieved by the hybrid connected NoC router may vary depending on the specific workload, application characteristics, and system architecture. Real-world implementations and rigorous testing across diverse use cases are crucial to validate its effectiveness comprehensively. As technology advances and more sophisticated applications emerge, ongoing research and refinement of the hybrid connected NoC router will be crucial to harness its full potential in improving latency and throughput across a broad spectrum of computing scenarios.

References

[1]. Arulananth, T. S., M. S. M. U. S. Baskar, Udhaya Sankar SM, R. Thiagarajan, Pasupuleti Raja Rajeshwari, Aruru Sai Kumar, and A. Suresh. "Evaluation of low power consumption network on chip routing architecture." *Microprocessors and Microsystems* 82 (2021): 103809.

[2]. Swapna, S., Ayas Kanta Swain, and Kamala Kanta Mahapatra. "Design and analysis of five port routers for network on chip." In *2012 Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics*, pp. 51-55. IEEE, 2012.

[3]. Jain, Arpit, Adesh Kumar, Rakesh Dwivedi, and Sanjeev Sharma. "Network on chip router for 2D mesh design." *International Journal of Computer Science and Information Security* 14, no. 9 (2016):

[4]. Shafique, Muhammad Akmal, Naveed Khan Baloch, Muhammad Iram Baig, Fawad Hussain, Yousaf Bin Zikria, and Sung Won Kim. "NoCGuard: A reliable network-on-chip router architecture." *Electronics* 9, no. 2 (2020): 342.

[5]. Hussain, Ayaz, Muhammad Irfan, Naveed Khan Baloch, Umar Draz, Tariq Ali, Adam Glowacz, Larisa Dunai, and Jose Antonino-Daviu. "Savior: A reliable fault resilient router architecture for network-on-chip." *Electronics* 9, no. 11 (2020): 1783.

[6]. Khalil, Kasem, Omar Eldash, Ashok Kumar, and Magdy Bayoumi. "Self-healing router approach for high-performance network-on-chip." *IEEE Open Journal of Circuits and Systems* 2 (2021): 485-496.

[7]. Sarihi, Amin, Ahmad Patooghy, Mahdi Hasanzadeh, Mostafa Abdelrehim, and Abdel-Hameed A. Badawy. "Securing network-on-chips via novel anonymous routing."

[8]. In *Proceedings of the 15th IEEE/ACM International Symposium on Networks-on-Chip*, pp. 29-34. 2021.

[9]. Ahmad, Khurshid, and Muhammad Sethi. "Review of network on chip routing algorithms." *EAI Endorsed Transactions on Context-aware Systems and Applications* 7, no. 22 (2020).

[10]. Meghabber, Mohammed Amine, Abdelkader Aroui, LakhdarLoukil, Kamel Benhaoua, and Toufik Djeradi. "A flexible network on-chip router for data-flow monitoring." In *2017 5th International Conference on Electrical Engineering-Boumerdes (ICEE-B)*, pp. 1-6. IEEE, 2017.