International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

Heterogeneous Network-on-Chip Design with Adaptive Key Rotation Aware Obfuscation

Papa Mamatha[1] , kanne Naveen[1*]
[1]Department of Electronics & Communication Engineering, Vaagdevi Engineering College, Warangal, 506005, Telangana, India.
*Correspondence: Kanne Naveen (kanne.naveen@gmail.com)

Abstract
The global semiconductor industry surpassed USD 790 billion in revenue during 2025, while modern multicore and AI processors integrate hundreds of processing elements that generate massive on-chip communication traffic. Industry studies indicate that communication latency and interconnect overhead account for a significant portion of overall System-on-Chip (SoC) performance degradation, making efficient and secure Network-on-Chip (NoC) communication a critical design requirement. NoC architectures are extensively utilized in artificial intelligence accelerators, cloud data-center processors, autonomous vehicles, high-performance computing systems, edge devices, and advanced mobile SoCs, where reliable and secure packet transmission is essential for maintaining computational efficiency. Existing NoC security architectures based on centralized logic obfuscation and distributed interconnect obfuscation provide basic protection against unauthorized access and hardware attacks; however, they suffer from limitations such as static key management, increased routing complexity, higher latency, congestion sensitivity, scalability constraints, excessive Look-Up Table (LUT) utilization, increased power consumption, and limited adaptability to dynamic attack scenarios. To address these challenges, this work proposes a Heterogeneous Network-on-Chip (HNoC) with Adaptive Key Rotation Aware Obfuscation (AKRAO) architecture. The proposed framework integrates Input Switching Allocation with Buffer Management, Forwarding Buffer, Dynamic Look-Ahead Bypass Route Computation, Shortest Path Computation, Parallel Virtual Channel and Switch Allocation, Adaptive Key Rotation, and Crossbar Switching with Obfuscation within a unified routing architecture. The adaptive key rotation mechanism continuously updates security credentials to enhance resistance against key compromise, traffic analysis, and side-channel attacks, while intelligent routing and forwarding strategies reduce congestion and communication delays. Consequently, the proposed AKRAO-enabled HNoC improves throughput, routing efficiency, security robustness, scalability, resource utilization, and energy efficiency, making it highly suitable for next-generation heterogeneous multicore and manycore VLSI systems.

Keywords: Adaptive Key Rotation, Buffer Management, Crossbar Switching, Heterogeneous Network-on-Chip (HNoC), Network-on-Chip (NoC), Obfuscation.

1. Introduction

The semiconductor industry has experienced unprecedented growth due to the widespread adoption of artificial intelligence, machine learning, cloud infrastructure, 5G communication, autonomous vehicles, and edge computing technologies. Global semiconductor revenue has reached record levels, with market analyses predicting sustained expansion over the next decade. The increasing demand for high-performance processors, AI accelerators, and specialized computing platforms has intensified the need for efficient on-chip communication mechanisms. Modern SoCs integrate numerous processing cores, memory modules, accelerators, and peripheral units on a single silicon die, resulting in complex communication requirements that cannot be efficiently handled by traditional bus-based architectures. Network-on-Chip technology has therefore become a fundamental solution for enabling scalable communication among interconnected processing elements. As

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

multicore processors continue to evolve, communication efficiency has become equally important as computational capability. Current market reports suggest that the multicore processor industry is expected to grow significantly over the next decade, driven by increasing demand for parallel computing applications. NoC architectures provide structured packet-based communication that enables efficient data movement among processing units while supporting scalability and modular design methodologies. These architectures are extensively employed in high-performance computing systems, AI accelerators, consumer electronics, and data-center processors where massive amounts of information must be exchanged with minimal latency. The emergence of chiplet-based integration and heterogeneous computing further emphasizes the importance of intelligent on-chip networking solutions capable of handling diverse workloads and communication patterns.

2. Literature Survey

Ahmad et al. [1] proposed a comprehensive review of AI-driven application mapping and scheduling techniques for Network-on-Chip systems. The authors systematically analyzed task mapping approaches used in multicore NoC architectures. They investigated machine learning, heuristic, and metaheuristic scheduling algorithms for workload allocation. The study compared static and dynamic mapping strategies under varying traffic conditions. Multiple performance metrics including latency, throughput, and energy consumption were evaluated. The review highlighted the effectiveness of AI-based optimization for improving resource utilization and communication efficiency. The work focuses on comparative analysis and lacks a dedicated hardware architecture for secure NoC implementation.

Kavitha [2] proposed a novel Network-on-Chip architecture targeting high throughput and energy-efficient VLSI systems. The methodology employed optimized packet routing mechanisms to reduce communication delays. Efficient resource allocation techniques were integrated to improve data transfer performance. The architecture focused on balancing communication load across interconnected processing elements. Performance evaluation was carried out using throughput and energy efficiency metrics. The architecture does not incorporate advanced security mechanisms for protecting communication channels. Bhargavi et al. [3] proposed a scalable Network-on-Chip design specifically optimized for FPGA implementation. The methodology utilized modular router structures to support scalability across multiple processing nodes. Resource-aware routing strategies were employed to minimize hardware overhead. FPGA-based implementation and validation were conducted to evaluate practical feasibility. The design focused on reducing area utilization while maintaining communication performance. The proposed design primarily emphasizes scalability and does not address communication security challenges.

Manikandan and Karthikumar [4] proposed an AI-driven clock tree synthesis methodology for modern VLSI systems. The framework combined traditional clock distribution techniques with machine learning optimization algorithms. Clock skew, insertion delay, and power consumption were considered during optimization. AI models were employed to predict optimal clock routing configurations. The methodology enhanced timing closure and reduced clock distribution overhead. E The study is limited to clock optimization and does not address NoC communication or routing security. Fasiku et al. [5] proposed a hybrid wireless Network-on-Chip architecture with a load-balanced congestion-aware routing algorithm. The methodology integrated wired and wireless communication links to improve data transfer efficiency. Congestion monitoring mechanisms were incorporated for dynamic

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

route selection. Load balancing strategies distributed traffic across available communication resources. Routing decisions were adapted according to network conditions. The routing framework lacks dynamic security and key management mechanisms for secure communication.

Shree et al. [6] proposed an intelligent DVFS-enabled VLSI framework for ultra-low-latency IoT systems. The methodology integrated Dynamic Voltage and Frequency Scaling to optimize power-performance tradeoffs. Intelligent workload analysis was employed to adjust operating parameters dynamically. Resource utilization and latency metrics were continuously monitored. Energy-efficient scheduling mechanisms were incorporated for IoT workloads. The framework focuses on power optimization and does not provide secure NoC communication capabilities. Rahaman et al. [7] proposed a performance-centric topology for hybrid wireless Network-on-Chip systems. The methodology introduced optimized topological arrangements to enhance communication performance. Wireless communication links were strategically integrated within conventional NoC structures. Traffic distribution mechanisms were employed to improve data flow efficiency. Network performance was evaluated under various workload scenarios. The architecture does not include obfuscation or adaptive security mechanisms against hardware attacks.

Kumar [8] proposed thermal-reliable VLSI architectures for power-constrained applications. The methodology focused on thermal-aware design strategies to improve system reliability. Temperature-sensitive optimization techniques were employed during architectural design. Power dissipation and thermal distribution were continuously analyzed. Reliability enhancement mechanisms were incorporated to prevent thermal hotspots. The work does not address communication routing efficiency or NoC security concerns.

Knag et al. [9] proposed a hybrid-bonded 56-core DNN processor incorporating a high-bandwidth 3D Network-on-Chip. The methodology utilized hybrid bonding technology to achieve dense interconnect integration. A high-speed NoC infrastructure was developed to support massive data movement. The architecture optimized communication bandwidth among processing cores. Advanced packaging techniques were integrated for performance enhancement. The architecture prioritizes performance and bandwidth while providing limited discussion on communication security.

Khedersolh et al. [10] proposed a thermal-aware routing framework for three-dimensional NoCs using PCA and ANFIS prediction models. The methodology employed principal component analysis for temperature feature extraction. Adaptive neuro-fuzzy inference systems predicted thermal behavior across the network. Routing decisions were dynamically adjusted according to predicted temperatures. Thermal balancing mechanisms minimized hotspot formation. The routing mechanism focuses primarily on thermal optimization rather than communication security. Benhaoues et al. [11] proposed a modified XYZ routing algorithm for thermal management in 3D NoC architectures. The methodology incorporated temperature-awareness into conventional XYZ routing. Thermal conditions were continuously monitored during packet transmission. Routing paths were modified to avoid overheated regions. Adaptive route selection improved thermal balance across the network. The approach lacks security-aware routing and protection against malicious attacks.

Waddoups et al. [12] proposed a probabilistic verification methodology for modular Network-on-Chip systems. The framework utilized formal verification techniques to analyze NoC correctness. Probabilistic models were developed to capture system uncertainties. Verification procedures evaluated communication reliability and protocol compliance. Modular analysis improved

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

scalability for complex NoC systems. The work focuses on verification rather than improving routing performance or security. Aruna et al. [13] proposed a power-aware VLSI synthesis methodology using static and dynamic clock gating techniques. The methodology minimized unnecessary switching activity during operation. Clock gating strategies were incorporated at multiple design levels. Power consumption analysis guided optimization decisions. Dynamic control mechanisms adapted clock distribution according to workload demands. The methodology addresses power optimization only and does not enhance NoC communication security.

Khaidukov and Alekseev [14] proposed efficiency improvement approaches for mesh networks in distributed memory SoC. The methodology analyzed mesh topology communication bottlenecks. Optimization strategies were developed to improve packet routing efficiency. Resource management mechanisms enhanced communication performance. Network traffic behavior was evaluated under different workload conditions. The proposed mesh optimization framework does not incorporate security-aware communication mechanisms. Senthilkumar et al. [15] proposed the D2C-GMH-DSTN framework for VLSI partitioning and floorplanning. The methodology employed dilated causal convolution for feature extraction. Multi-head decision transformers were utilized for optimization-driven partitioning. Floorplanning decisions were guided by learned spatial relationships. The framework aimed to improve layout quality and design efficiency. The work focuses on physical design automation and does not address NoC routing architectures.

3. Proposed System

The proposed HNoC Router with AKRAO is designed as shown in Figure 1 to provide secure and efficient packet communication between heterogeneous processing elements within a SoC. The architecture consists of an Input Switching Allocator and Buffer, a HNoC Router integrated with AKRAO, an Output Switching Allocator and Buffer, and a Forwarding Buffer. The forwarding buffer provides an alternative high-speed forwarding path for packet transmission, while the AKRAO-enabled router enhances security by dynamically rotating cryptographic keys and obfuscating packet information to protect against traffic analysis, side-channel attacks, and unauthorized data access, The proposed architecture aims to improve routing efficiency, communication security, throughput, and network reliability while maintaining low latency in heterogeneous computing environments.

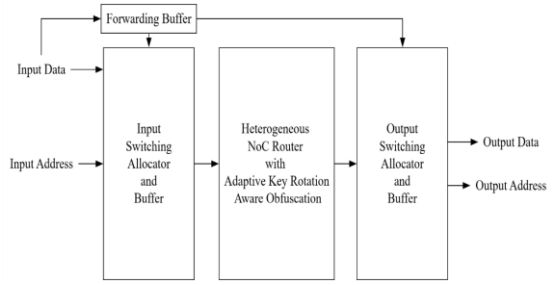


Figure. 1: Proposed system architecture on adaptive key rotation aware obfuscation.

Input Data and Address Reception: The operation begins when the Input Data and its corresponding Input Address arrive at the network interface. The input data contains the payload information to be transmitted, while the input address specifies the destination node within the heterogeneous NoC architecture. Both inputs are simultaneously fed into the Input Switching Allocator and Buffer, where the incoming packets are temporarily stored and prepared for routing decisions. This stage ensures proper synchronization between packet contents and destination information before forwarding them into the network.

Input Switching Allocation and Buffer Management: The Input Switching Allocator and Buffer is responsible for managing incoming traffic and allocating network resources. It performs packet buffering to

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

prevent congestion during high traffic conditions and examines the destination address to determine routing requirements. The allocator prioritizes packets based on scheduling policies and allocates the appropriate virtual channels and switch resources. By maintaining temporary storage and arbitration mechanisms, this module ensures efficient packet organization and minimizes packet loss during transmission.

Forwarding Buffer Operation: In parallel with the normal routing process, incoming packets can also access the Forwarding Buffer. This buffer acts as a temporary holding and bypass mechanism for packets requiring rapid forwarding or congestion avoidance. The forwarding buffer stores packets and provides an alternative transmission path directly toward the output side when network conditions permit. This mechanism reduces router contention, improves packet delivery speed, and enhances overall throughput. The forwarding buffer therefore serves as a latency reduction and traffic balancing component within the architecture.

Heterogeneous NoC Router with AKRAO: After allocation, packets are forwarded to the Heterogeneous NoC Router integrated AKRAO. This is the core component of the proposed architecture. The router performs route computation and determines the optimal communication path based on destination information. Simultaneously, the AKRAO module dynamically rotates encryption keys at predefined intervals or according to network conditions. The packet headers and selected payload information are obfuscated using the current security key, preventing attackers from identifying routing patterns or extracting sensitive information. The adaptive key rotation mechanism continuously updates security credentials, making cryptographic attacks significantly more difficult while maintaining secure communication among heterogeneous processing nodes.

Secure Routing and Packet Transmission: Within the router, routing algorithms analyze network topology, congestion levels, and communication priorities to select the most efficient path. The obfuscated packets are then transmitted through internal routing channels toward the designated output port. The integration of adaptive key rotation ensures that even if a key becomes compromised, subsequent packet transmissions remain protected through newly generated keys. This stage enhances confidentiality, integrity, and resilience against side-channel attacks and traffic monitoring.

Output Switching Allocation and Buffering: Packets emerging from the router are received by the Output Switching Allocator and Buffer. This module performs output arbitration and manages outgoing packet queues. It temporarily stores routed packets and schedules their transmission according to output port availability. The allocator resolves output port conflicts, ensures fair resource distribution, and maintains proper packet sequencing. Packets arriving through either the router path or forwarding buffer path are merged and managed efficiently before final delivery.

Output Data Generation: After successful allocation and buffering, the processed packets are transmitted as Output Data. The output data represents the securely routed information that has passed through the AKRAO-enabled routing process. The transmitted data maintains confidentiality and integrity due to the obfuscation and adaptive key rotation mechanisms employed within the router. This stage completes the payload delivery process to the destination node.

Output Address Generation: Simultaneously, the corresponding Output Address information is generated and delivered alongside the output data. The output address ensures that the destination processing element correctly interprets and receives the incoming packet. Proper address management guarantees accurate packet delivery across the

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

heterogeneous network and supports reliable communication among multiple computational units.

3.1 Forwarding Buffer

The Forwarding Buffer as shown in figure 2 serves as an intermediate packet storage and fast-forwarding mechanism within the proposed HNoC with AKRAO architecture. Its primary function is to temporarily hold incoming packets and provide an alternative transmission path that bypasses unnecessary routing delays. By maintaining a dedicated forwarding space, the buffer reduces network congestion, minimizes packet waiting time, and improves overall communication efficiency. Additionally, the forwarding buffer supports secure packet movement by cooperating with the AKRAO-enabled router, ensuring that packet transfers remain protected while sustaining high throughput and low latency across heterogeneous processing elements.

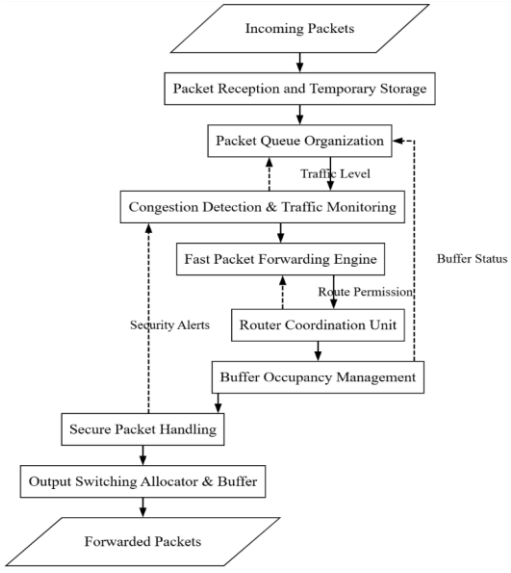


Figure. 2: Proposed forwarding buffer block diagram.

Packet Reception and Temporary Storage: The forwarding buffer begins its operation by receiving packets from the incoming communication stream. As packets arrive from the input interface, they are temporarily stored within the buffer memory. This temporary storage prevents packet loss during periods of heavy traffic and ensures that incoming packets remain available for subsequent forwarding decisions. The buffer continuously monitors available storage locations and manages packet placement efficiently to accommodate varying traffic loads.

Packet Queue Organization: After reception, the forwarding buffer organizes the stored packets into queues according to predefined scheduling policies. The queue management mechanism arranges packets based on factors such as arrival time, destination requirements, traffic priority, and network status. This organized structure enables efficient packet retrieval and prevents disorderly transmission. By maintaining structured packet queues, the forwarding buffer enhances traffic management and supports smooth data flow within the NoC environment.

Congestion Detection and Traffic Monitoring: The forwarding buffer continuously observes network conditions to identify congestion levels and communication bottlenecks. It evaluates buffer occupancy, router utilization, channel availability, and packet waiting times. When congestion is detected in the primary routing path, the forwarding buffer takes corrective action by temporarily retaining packets or redirecting them through alternative forwarding opportunities. This monitoring capability helps prevent packet accumulation and improves overall network stability.

Fast Packet Forwarding Mechanism: Once suitable transmission resources become available, the forwarding buffer initiates the forwarding process. Instead of forcing every packet through multiple routing stages, selected packets can be rapidly forwarded toward the output side using the dedicated forwarding path. This mechanism significantly reduces transmission latency and minimizes router contention. The fast-forwarding capability is particularly beneficial during high traffic scenarios where conventional routing resources may become temporarily overloaded.

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

Coordination with Routing Decisions: The forwarding buffer works collaboratively with the HNoC router to ensure correct packet delivery. Before forwarding a packet, the buffer verifies destination information and routing permissions. The forwarding operation is synchronized with router control signals to avoid transmission conflicts and resource contention. This coordination guarantees that forwarded packets maintain proper routing integrity while benefiting from accelerated delivery.

Buffer Occupancy Management: To maintain efficient operation, the forwarding buffer dynamically manages its storage resources. Occupancy levels are continuously monitored, and packets are released as soon as forwarding opportunities arise. If buffer utilization exceeds predefined thresholds, control mechanisms regulate incoming traffic and prioritize packet transmission. This adaptive management strategy prevents buffer overflow and ensures consistent communication performance under varying network workloads.

Secure Packet Handling: As part of the AKRAO-enabled architecture, the forwarding buffer handles packets that may already contain obfuscated routing and payload information. The buffer preserves packet confidentiality by preventing unauthorized modifications during storage and forwarding operations. Since the packet content remains protected through adaptive key rotation and obfuscation techniques, the forwarding process does not compromise communication security. This secure handling mechanism strengthens resistance against packet interception and traffic analysis attacks.

Packet Delivery to Output Switching Allocator: After successful forwarding, the packet is transmitted to the Output Switching Allocator and Buffer. The forwarded packet joins the normal routing stream and undergoes output arbitration before final delivery. This stage completes the forwarding buffer operation, ensuring that packets reach their intended destination with reduced latency and improved transmission efficiency. The seamless integration between the forwarding buffer and output allocator contributes significantly to enhanced network throughput and communication reliability.

3.2 Input Switching Allocation with Buffer Management

The Input Switching Allocation with Buffer Management module as shown in Figure 3 serves as the primary entry point for packet processing in the proposed HNoC with AKRAO architecture. It is responsible for receiving incoming data packets and destination addresses, temporarily storing packets in dedicated input buffers, allocating switching resources, and managing traffic flow toward the routing engine. The module performs arbitration, congestion control, packet scheduling, and resource allocation to ensure efficient utilization of network resources. By intelligently managing incoming traffic and buffering operations, the ISABM module reduces packet loss, minimizes contention, improves throughput, and enables seamless communication between heterogeneous processing elements while maintaining compatibility with the security mechanisms implemented in the AKRAO-enabled router.

Packet and Address Reception: The operation of the ISABM module begins with the arrival of input data packets and their corresponding destination addresses from source processing elements. The incoming packet stream contains both payload information and routing information necessary for communication within the NoC. Upon arrival, the module captures the packet data and address information simultaneously and prepares them for temporary storage and further processing. This stage establishes the foundation for reliable packet transmission by ensuring that all incoming information is properly registered before entering the network.

Input Buffer Storage: After packet reception, the incoming packets are stored in dedicated

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

input buffers. These buffers act as temporary holding locations that accommodate fluctuations in network traffic and prevent packet loss during periods of congestion. The buffer management mechanism continuously monitors available memory locations and allocates storage resources efficiently. Temporary buffering allows packets to wait for routing resources without disrupting incoming traffic, thereby improving communication reliability and network stability.

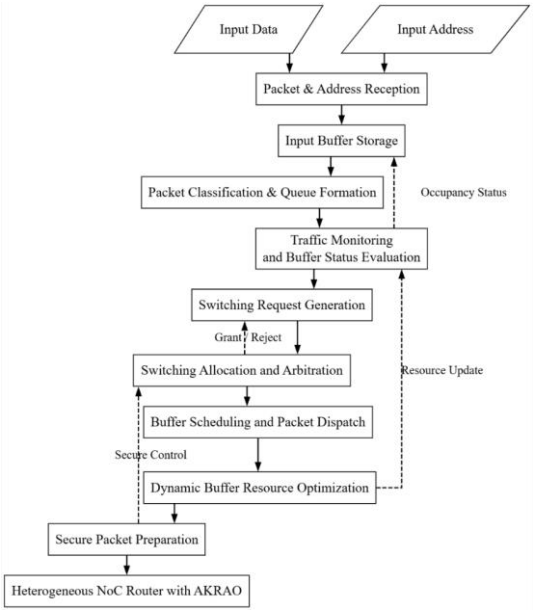


Figure. 3: Proposed input switching allocation with buffer management.

Packet Classification and Queue Formation: Once packets are stored, the ISABM module classifies them according to their destination addresses, traffic classes, priority levels, and service requirements. Packets with similar routing characteristics are organized into separate queues. This queue formation process enables structured packet handling and simplifies subsequent scheduling decisions. Proper packet classification ensures that high-priority traffic receives appropriate attention while maintaining fairness among different communication flows.

Traffic Monitoring and Buffer Status Evaluation: The module continuously evaluates buffer occupancy levels and incoming traffic conditions. Parameters such as queue length, packet arrival rate, waiting time, and channel availability are monitored in real time. This information provides an accurate assessment of current network conditions and helps identify potential congestion points. Through continuous traffic observation, the module can proactively adjust resource allocation strategies to maintain optimal communication performance.

Switching Request Generation: After packet classification, each buffered packet generates a switching request indicating its desired output direction. The switching requests are forwarded to the switching allocator for arbitration. Multiple packets may simultaneously request access to the same routing resource or output channel. Therefore, the switching request generation stage serves as a critical interface between packet storage and resource allocation, ensuring that routing requirements are accurately communicated to the allocator.

Switching Allocation and Arbitration: The switching allocator examines all pending requests and determines which packets will receive access to the switching fabric. Arbitration algorithms evaluate packet priorities, waiting times, fairness requirements, and resource availability before granting access. When multiple packets compete for the same communication resource, the allocator resolves conflicts efficiently and selects the most suitable transmission candidate. This allocation process minimizes contention and maximizes utilization of available routing resources.

Buffer Scheduling and Packet Dispatch: Once switching resources are assigned, the scheduler selects packets from the input queues and prepares them for transmission. The scheduling mechanism determines the order in which packets leave the buffer and enter the routing stage. By carefully controlling packet dispatch operations, the module prevents congestion propagation and ensures smooth traffic flow throughout the network. Efficient scheduling contributes significantly to reduced

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

latency and improved packet delivery performance.

Transfer to Heterogeneous NoC Router: After successful scheduling and allocation, packets are forwarded from the input buffers to the HNoC Router with AKRAO. The routing engine receives packets along with their destination information and proceeds with route computation, security processing, and packet forwarding. This stage marks the completion of input-side traffic management and initiates secure routing operations within the NoC architecture.

Dynamic Buffer Resource Optimization: During continuous operation, the buffer module dynamically adjusts buffer utilization according to changing network conditions. Buffer occupancy thresholds, traffic patterns, and packet priorities are analyzed to optimize resource allocation. This adaptive optimization mechanism improves memory efficiency, prevents buffer overflow, and maintains stable performance under varying workload conditions. Dynamic buffer management ensures that the system remains responsive even during high traffic scenarios.

Support for Secure Communication: This module supports the security objectives of the AKRAO-enabled architecture by preserving packet integrity before routing. Buffered packets remain protected from unauthorized modification, and packet metadata required for secure routing is maintained accurately. By providing reliable packet handling and organized traffic management, the module establishes a secure and efficient foundation for subsequent key rotation and obfuscation operations within the router.

4. Results and Discussions

Figure 4 presents the functional simulation results of the proposed Heterogeneous Network-on-Chip with Adaptive Key Rotation Aware Obfuscation (AKRAO). The waveform verifies the successful transmission of packet data among multiple ports during the simulation interval of 0 ns to 1000 ns. The input ports contain hexadecimal packet values such as 00000019, 00000023, 0000002D, and 00000038, while the routing control signals In_add[1:0] and out_add[1:0] indicate routing decisions with values 2 and 1, respectively. The output ports successfully receive the corresponding packet values, where port_A[7:0] = 23, port_B[7:0] = 2D, and port_C[7:0] = 2D, confirming correct packet forwarding and routing functionality. The simulation demonstrates that the proposed architecture correctly processes packet transfers, address selection, and output generation without functional errors, validating the effectiveness of the routing and obfuscation framework.

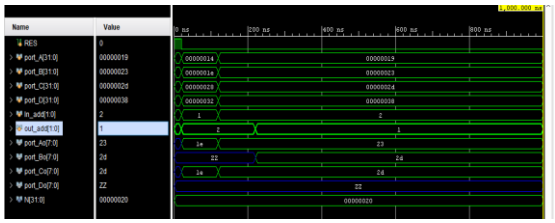


Figure. 4: Proposed simulation outcome.

Figure 5 illustrates the FPGA resource utilization of the proposed architecture. The implementation requires only 135 LUTs out of the available 134,600 LUTs, resulting in an extremely low utilization of 0.10%. The design utilizes 261 I/O pins from the available 500 I/O pins, corresponding to an I/O utilization of 52.20%. Compared with the existing architecture that utilized 1,976 LUTs, the proposed design achieves a substantial reduction in logic resource consumption. The significant decrease in LUT utilization indicates that the proposed architecture is highly area-efficient and requires minimal programmable logic resources for implementing adaptive routing, key rotation, and obfuscation mechanisms. Such reduced hardware complexity allows the architecture to support larger NoC deployments while preserving FPGA resources for additional processing and communication modules.

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

Resource	Estimation	Available	Utilization...
LUT	135	134600	0.10
IO	261	500	52.20

Figure. 5: Proposed area outcome.

Figure 6 presents the power consumption analysis of the proposed architecture. The total dynamic power consumption is 3.759 W, accounting for 97% of the total power usage, while static power contributes only 0.117 W (3%). Within the dynamic power category, signal activity consumes 2.995 W (80%), logic resources consume 0.289 W (8%), and I/O resources consume 0.475 W (12%). The static power is entirely represented by PL Static Power of 0.117 W (100%). Compared with the existing architecture, which consumed 42.556 W of dynamic power, the proposed design achieves a remarkable reduction in power consumption. The lower logic power and overall dynamic power demonstrate the effectiveness of the optimized routing structure, adaptive communication management, and efficient hardware implementation, making the architecture highly suitable for energy-efficient VLSI applications.

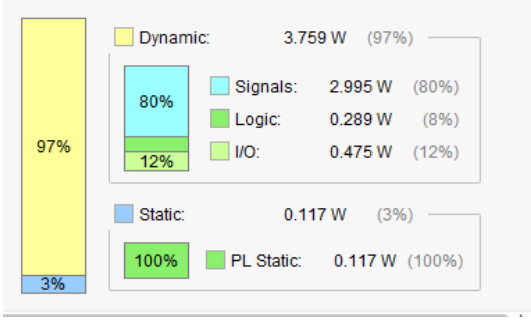


Figure. 6: Proposed power summary.

Figure 7 shows the setup timing analysis results of the proposed architecture. The critical setup paths exhibit total delays ranging from 17.802 ns to 18.275 ns, which is significantly lower than the setup delays observed in the existing design. The maximum setup delay of 18.275 ns occurs between port_D[24] and port_B0[24], while other critical paths report delays of 18.157 ns, 18.041 ns, 18.020 ns, 17.979 ns, 17.943 ns, 17.915 ns, 17.879 ns, 17.829 ns, and 17.802 ns. The corresponding logic delays vary from 3.822 ns to 4.214 ns, whereas net delays range between 13.666 ns and 14.204 ns. The design contains only 3–4 logic levels, 2–3 routing levels, and fanout values between 4 and 32, indicating a simplified routing structure. The substantial reduction in setup delay demonstrates that the proposed architecture effectively minimizes communication latency and improves timing performance through optimized routing and resource allocation mechanisms.

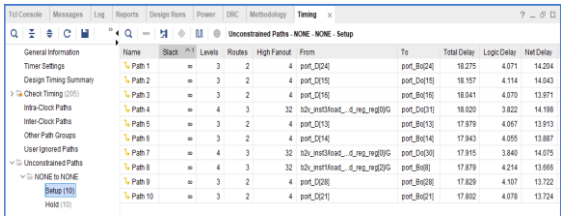
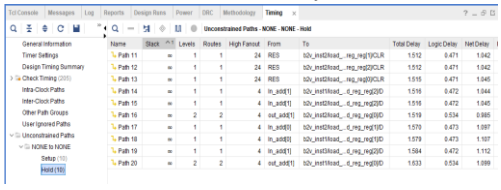


Figure. 7: Proposed setup delay outcome.

Figure 8 presents the hold timing analysis of the proposed architecture. The reported hold path delays range from 1.512 ns to 1.633 ns, ensuring stable data transfer and synchronization throughout the routing system. The minimum hold delay of 1.512 ns is observed in paths associated with b2v_inst2/load_..._reg_reg[1]/CLR and b2v_inst2/load_..._reg_reg[2]/CLR, while the maximum hold delay reaches 1.633 ns between out_add[1] and b2v_inst1/load_..._reg_reg[0]/D. The corresponding logic delays vary from 0.471 ns to 0.534 ns, while net delays range between 0.985 ns and 1.112 ns. The timing report further indicates low routing complexity with only 1–2 logic levels, 1–2 routing levels, and fanout values between 4 and 24. These results confirm that the proposed architecture maintains reliable timing behavior and stable packet forwarding operations while supporting significantly improved setup performance and reduced communication delay.



International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

Figure. 8: Proposed hold delay outcome.

4.1 Comparative Analysis

Table 1 compares the hardware area utilization of the existing architecture and the proposed AKRAO-HNoC architecture. The existing design requires 1,976 LUTs, whereas the proposed architecture utilizes only 135 LUTs, resulting in a substantial 93.17% reduction in LUT utilization. Similarly, the percentage of LUT resource consumption decreases from 1.47% in the existing architecture to only 0.10% in the proposed design, corresponding to a 93.20% improvement in area efficiency. This significant reduction demonstrates that the proposed AKRAO-HNoC architecture achieves a highly optimized hardware implementation while maintaining routing and security functionality. The lower LUT requirement indicates reduced logic complexity, simplified routing structures, and efficient resource allocation, allowing additional modules to be integrated into the FPGA without imposing excessive hardware overhead. Consequently, the proposed architecture provides a more compact and scalable solution for Network-on-Chip implementations compared with the existing approach.

Table 1. Area comparison between existing and proposed architectures

Resource Metric	Existing Architecture	Proposed AKRAO-HNoC	Improvement (%)
LUT Utilization	1,976	135	93.17
LUT Utilization (%)	1.47	0.10	93.20

Table 2. Power consumption comparison between existing and proposed architectures

Power Metric	Existing Architecture (uW)	Proposed AKRAO-HNoC (uW)	Reduction (%)
Dynamic Power	42.556	3.759	91.17
Signal Power	22.361	2.995	86.61
Logic Power	19.979	0.289	98.55
Static Power	0.413	0.117	71.67
Total Power	42.969	3.876	90.98

Table 2 presents the power consumption comparison between the existing architecture and the proposed AKRAO-HNoC architecture. The existing system consumes 42.556 µW of dynamic power, while the proposed design requires only 3.759 µW, achieving a significant 91.17% reduction. Signal power consumption decreases from 22.361 µW to 2.995 µW, representing an 86.61% reduction, while logic power is dramatically reduced from 19.979 µW to 0.289 µW, corresponding to a remarkable 98.55% improvement. Similarly, static power consumption decreases from 0.413 µW to 0.117 µW, providing a 71.67% reduction. As a result, the overall total power consumption is reduced from 42.969 µW in the existing architecture to 3.876 µW in the proposed architecture, yielding an overall 90.98% power reduction. These results clearly indicate that the proposed AKRAO-HNoC architecture significantly minimizes switching activity, logic utilization, and static resource consumption, making it highly suitable for low-power VLSI and energy-constrained computing environments.

Table 3 compares the timing performance of the existing and proposed architectures using setup, logic, and net delay metrics. The maximum setup delay is reduced from 42.999 ns in the existing architecture to 18.275 ns in the proposed AKRAO-HNoC, resulting in a 57.50% improvement. Similarly, the minimum setup delay decreases from 40.244 ns to 17.802

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

ns, achieving a 55.76% reduction, while the average setup delay is lowered from 42.066 ns to 17.984 ns, corresponding to a 57.25% improvement. The maximum logic delay is significantly reduced from 15.103 ns to 4.214 ns, providing a substantial 72.10% improvement, indicating that the proposed architecture greatly simplifies the logic processing path. Furthermore, the maximum net delay decreases from 28.132 ns to 14.204 ns, yielding a 49.51% reduction in routing delay. These improvements demonstrate that the proposed AKRAO-HNoC architecture effectively minimizes communication latency, reduces routing complexity, and accelerates packet transmission. Consequently, the architecture achieves faster timing performance and improved operational efficiency compared with the existing NoC implementation.

Table 3. Delay Comparison Between Existing and Proposed Architectures

Timing Metric	Existing Architecture (ns)	Proposed AKRAO-HNoC (ns)	Improvement (%)
Maximum Setup Delay	42.999	18.275	57.50
Minimum Setup Delay	40.244	17.802	55.76
Average Setup Delay	42.066	17.984	57.25
Maximum Logic Delay	15.103	4.214	72.10
Maximum Net Delay	28.132	14.204	49.51

5. Conclusion

This work presented a HNoC architecture integrated with AKRAO to address the challenges of communication efficiency, hardware security, resource utilization, power consumption, and routing latency in modern multicore VLSI systems. The proposed architecture combines Input Switching Allocation with Buffer Management, Forwarding Buffer, Dynamic Look-Ahead Bypass Route Computation, Shortest Path Computation, Parallel Virtual Channel and Switch Allocation, Adaptive Key Rotation, and Crossbar Switching with Obfuscation to establish a secure and efficient communication framework. Experimental FPGA implementation results demonstrated significant improvements over the existing architecture, achieving a reduction in LUT utilization from 1,976 to 135 (93.17% improvement), total power consumption from 42.969 µW to 3.876 µW (90.98% reduction), and average setup delay from 42.066 ns to 17.984 ns (57.25% improvement). Furthermore, logic delay and net delay were reduced by 72.10% and 49.51%, respectively, indicating substantial enhancement in timing performance and routing efficiency. The adaptive key rotation mechanism further strengthens protection against key compromise, traffic analysis, and hardware-based attacks while maintaining reliable packet transmission. Overall, the proposed AKRAO-enabled HNoC architecture successfully achieves a balanced combination of high security, low power consumption, reduced hardware overhead, and improved communication performance, making it a promising solution for next-generation heterogeneous multicore and manycore System-on-Chip applications.

References

[1]. Ahmad, Naveed, Muhammad Kaleem, Mourad Elloumi, Muhammad Azhar Mushtaq, Ahlem Fatnassi, Mohd Fazil, Anas Bilal, and Abdulbasit A. Darem. "A Comprehensive Literature Review of AI-Driven Application Mapping and Scheduling Techniques for Network-on-

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

Chip Systems." Computer Modeling in Engineering & Sciences 146, no. 1 (2026).

[2]. Kavitha, M. (2026). A Novel Network-on-Chip Architecture for High-Throughput and Energy-Efficient VLSI Systems. National Journal of Advanced VLSI Design and Systems, 73-82.

[3]. Bhargavi, Mekala Bindu, Sai Siddharth Rokkam, Vattipelli Srinath, Sri Parameswaran, and J. Soumya. "Scalable Network-on-Chip Design for FPGA Implementation." IEEE Access 14 (2026): 5746-5763.

[4]. Manikandan, B., and S. Karthikumar. "Clock Tree Synthesis in Modern VLSI: From Foundational Algorithms to AI-Driven Optimization." Integration (2026): 102665.

[5]. Fasiku, Ayodeji Ireti, Jeevan Sirkunan, Muhammad Nadzir Marsono, Mohd Shahrizal Rusli, Ab Al-Hadi Ab Rahman, Mohammed Sultan Mohammed, and Jumeidi Dirwan Alexander. "Hybrid wireless network-on-chip architectures with load-balanced congestion-aware routing algorithm." Journal of Telecommunications and the Digital Economy 14, no. 1 (2026): 1-17.

[6]. Shree, S. Kirthika, T. Aravind, and R. Saravanakumar. "An Intelligent DVFS-Enabled VLSI Framework for Ultra-Low-Latency IoT Systems." In 2026 9th International Conference on Inventive Computation Technologies (ICICT), pp. 314-320. IEEE, 2026.

[7]. Rahaman, Munshi Mostafijur, Prasun Ghosal, and Chandan Giri. "A performance-centric topology for hybrid wireless-network-on-chip." Circuits, Systems, and Signal Processing 45, no. 1 (2026): 574-595.

[8]. Kumar, P. Sathish. "Thermal-Reliable VLSI Architectures for Power-Constrained Applications." Annals of Energy-Efficient VLSI Architectures (2026): 20-27.

[9]. Knag, Phil C., Gregory K. Chen, Shanshan Xie, Satish Yada, Wei Wu, Yu-Shiang Lin, Alexander Kashirin et al. "10.6 A Hybrid-Bonded 12.1 Tops/mm 2 5 6-Core DNN Processor with 2.5 Tb/s/mm 2 3D Network on Chip." In 2026 IEEE International Solid-State Circuits Conference (ISSCC), vol. 69, pp. 178-180. IEEE, 2026.

[10]. Khedersolh, Erfan, Majid Nezarat, HadiShahriar Shahhoseini, and Mohammad Reza Mosavi. "Thermal-aware routing in three-dimensional network on chips with temperature prediction using principal component analysis and adaptive neuro-fuzzy inference system." Engineering Applications of Artificial Intelligence 164 (2026): 113334.

[11]. Benhaoues, Atef, Abdelhalim Rabehi, El-Bay Bourennane, Abdelaziz Rabehi, Abdelmalek Douara, and Mohamed Benghanem. "Thermal Management in 3D Network on Chip Using Modified XYZ Routing Algorithm." International Journal of Numerical Modelling: Electronic Networks, Devices and Fields 39, no. 1 (2026): e70138.

[12]. Waddoups, Nick, Jonah Boe, Arnd Hartmanns, Prabal Basu, Sanghamitra Roy, Koushik Chakraborty, and Zhen Zhang. "Probabilistic Verification for Modular Network-on-Chip Systems." In International Conference on Verification, Model Checking, and Abstract Interpretation, pp. 383-407. Cham: Springer Nature Switzerland, 2026.

[13]. Aruna, M., Archana Nair, B. U. Vaishnavi, Shilpa Suresh Jagadal, and Yashaswini Yadav. "Power-Aware VLSI Synthesis Using Static and Dynamic Clock Gating Techniques." In 2026 9th International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech), vol. 9, pp. 1-6. IEEE, 2026.

[14]. Khaidukov, Danila, and Aleksandr Alekseev. "Efficiency Improvement

Approaches for a Mesh Network in a Distributed Memory System on Chip." International Journal of Open Information Technologies 14, no. 2 (2026): 74-86.

[15]. Senthilkumar, K. K., V. Magesh, S. S. Saravana Kumar, and Badiganchela Shiva Kumar. "D2C-GMH-DSTN: A high-precision partitioning and floor planning framework for VLSI circuits using dilated causal convolution and multi-head decision transformers." Analog Integrated Circuits and Signal Processing 126, no. 2 (2026): 28.