International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online) Volume 13, Issue 12, December 2024)

Efficient Routing in Network-on-Chip: Topology, Congestion, and Fault Tolerance

[1]Saurabh Kumar Singh [2]Pr. Sunny Jain
[1]Research Scholar, [2]Assistant Professor
[1&2]Department of Electronics and Communication Engineering
[1&2]Lakshmi Narain College of Technology, Bhopal, India

Abstract:Network-on-Chip (NoC) is a crucial communication framework for multiprocessor systems-on-chip (MPSoCs). NoC allows the components within an MPSoC to communicate efficiently through a network-based architecture. The performance of NoC is primarily influenced by factors such as topology, routing algorithms, and switching techniques. This paper reviews various NoC routing algorithms based on essential NoC architecture parameters. The study also highlights key considerations for designing efficient routing algorithms, including congestion-awareness, fault tolerance, deadlock, and livelock avoidance, which help reduce latency and enhance throughput.

Keywords: System-on-Chip, Network-on-Chip, Routing Algorithm

1. Introduction

System on Chip (SoC) has become a key component in modern computing, playing a pivotal role in mobile computing, embedded systems, and even extending to personal computing devices such as laptops and tablets. SoC design is a widely used methodology among VLSI (Very-Large-Scale Integration) designers. The traditional interconnection system in SoCs is based on either shared or dedicated bus architectures. However, one significant limitation of the bus system is that it allows only one communication at a time, leading to performance bottlenecks [1].

With advancements in technology, SoCs used in embedded systems are becoming increasingly large and complex [2]. As illustrated in **Figure 1**, the interconnection system in SoCs involves connecting various devices through a shared bus. This bus architecture presents challenges in terms of area utilization, single clock synchronization, propagation delay, latency, throughput, and power consumption [3]. To address these limitations, **Network on Chip (NoC)** has emerged as a more efficient solution for on-chip communication in SoCs [1]. NoC can be described as “a communication network designed for on-chip communication,” and it has proven to be highly effective in managing communication needs within SoC architectures [4-6].

2. Current Challenges in NoC Design

As the number of **IP blocks** integrated into a chip increases, the number of routes between cores grows at a squared rate, exacerbating the problem of congestion. Congestion elimination has become one of the primary challenges in **NoC** design [3]. As congestion increases, it results in a larger die size, the need for additional metal mask layers, and the creation of unpredictable paths, all of which complicate timing closure. These factors pose significant challenges to achieving low-cost and high-performance chips [3].

Moreover, as technology nodes shrink, the driving strength of transistors decreases, while the signal propagation time along the interconnect wires increases, negatively impacting the overall speed of the chip. The growing number of wires also adds to the cost of the chip. NoC designers must contend with a range of issues, including the complexity of handling multi-variable problems, navigating a large design space, balancing performance, power, and area trade-offs, and managing buffer sizing and pipeline depth [2].

Additionally, interconnect resources often become bottlenecks in performance, especially under high-traffic conditions. To address this, NoC designs must support network-level congestion control while tackling issues such as buffering and channel width. Modern NoC circuits need to be designed to mitigate these challenges, aiming to reduce latency, increase bandwidth, and ensure overall efficiency [2].

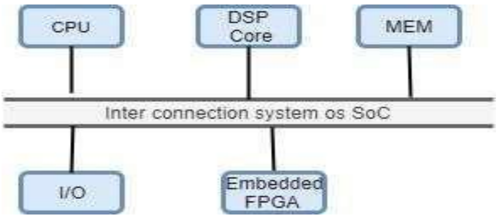


Fig (1). Interconnection System of SoC

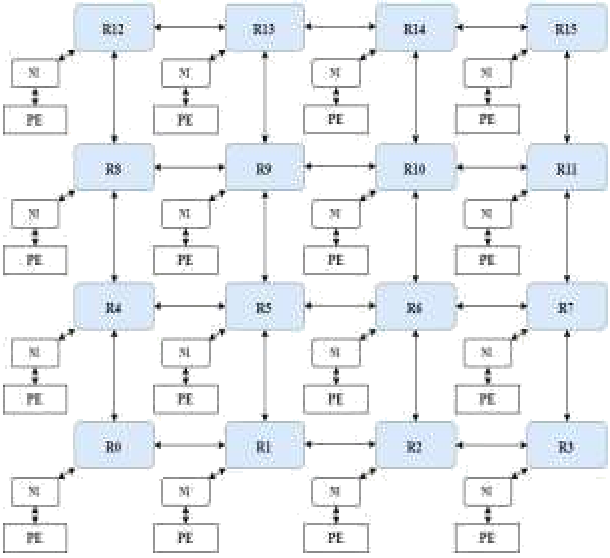


Fig (2). 4*4 NoC

Fig. (2) illustrates a 4x4 mesh topology of a Network-on-Chip (NoC), where processing elements (PEs) are connected to local routers through a network interface (NI). The routers are interconnected via point-to-point links. The NI is responsible for transforming messages into packets, which are forwarded by the routers to neighboring routers. The packets continue to travel through the network until they reach their destination [10, 11]. The overall performance of NoC depends on four key factors: topology, routing algorithm, flow control, and switching technique [4].

Topology refers to how nodes are connected in the network. Various topologies include mesh, torus, tree, ring, star, spidergon, and irregular topologies [10]. The routing algorithm determines the path a packet takes from the source to the destination. Examples of routing algorithms include XY, IX/Y, and XYX [4]. Switching techniques in NoC are primarily of two types: circuit switching and packet switching. In circuit switching, a physical or virtual link is established between the source and destination, while in packet switching, messages are divided into packets that are routed individually. The routing algorithm plays a crucial role in determining the route for each packet [3].

A deep understanding of routing algorithms is critical for effective NoC design, as they are a key factor influencing communication performance. This review compares various routing algorithms based on fundamental parameters of NoC architecture. These parameters include topology, routing type, switching technique, packet and flit size, power dissipation, latency, throughput, and the simulator used for implementation. The comparative analysis of these routing algorithms is summarized in Table 1. We believe this review will aid the research community in addressing routing challenges in future NoC architectures.

3. Description of Network-On-Chip and Routing Architectures

Different types of routing algorithms have been developed for designing Networks-on-Chip (NoC). These algorithms are classified based on three keycharacteristics: routing decision, path definition, and path length [1, 12].

1. Routing Decision:

Source Routing: The routing path is determined entirely by the source router.

Distributed Routing: Each router along the path makes independent decisions to determine the next hop for the packet [1].

Path Definition (Adaptivity):

Deterministic Routing: The path from source to destination is predetermined and fixed.

Adaptive Routing: The path can change dynamically based on network conditions, such as congestion or faults. A sub-type, partially adaptive routing, restricts certain directions while allowing some flexibility [13].

2Path Length:

Minimal Routing: This type selects the shortest possible path between the source and destination.

Non-minimal Routing: This type allows for longer, possibly more complex paths [1, 13].

Examples of routing algorithms include a range of adaptive algorithms like GOAL, GAL, DyXY, BARP, ADBR, MaS, Fault-tolerant, FAFT, FT-DyXY, Free-rider, Novel Adaptive, Traffic Allocator, MCAR, Efficient Deadlock-Free, ESPDA, and Adaptive Multipath [9, 14-28]. Partial adaptive algorithms include OE and 3DEP [29, 30]. Adaptive and deterministic algorithms include DyAD and FA-DyAD [31, 32], while deterministic routing algorithms include FTXY and ZigZig [33].

Further classifications, such as congestion-aware algorithms and fault-tolerant routing algorithms, will be discussed in sections 6 and 7.

4. Topologies

Topology in NoC refers to the organization of routers and channels, essentially defining the roadmap for communication between elements (PEs) [3]. The topology directly influences performance, scalability, and fault tolerance in NoC designs. Different topologies exhibit different properties in terms of communication efficiency, scalability, and power consumption.

Topologies can be classified into two main categories: regular and irregular [10].

Regular Topology: Nodes are connected in a fixed, predictable pattern. Common examples include:

Mesh: Routers are arranged in an MxN grid, where each intersection represents a router connected to its neighbors.

Torus: Similar to mesh, but the end routers are connected to form a continuous loop along the rows and columns.

Star: All routers are connected to a central router.

Ring: Routers are connected in a circular manner, where each router is linked to two neighbors.

Tree: Routers are arranged hierarchically, where each child router is connected to a parent router. Figures 3-6 depict the structures of torus, star, ring, and tree topologies, respectively.

Irregular Topology: Nodes are connected in a non-uniform, non-patterned manner, providing flexibility in how they are linked. **Figure 7** illustrates an irregular topology.

Fig (3). Torus topology

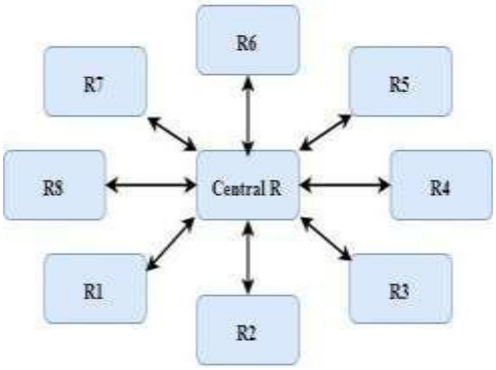


Fig (4). Star topology

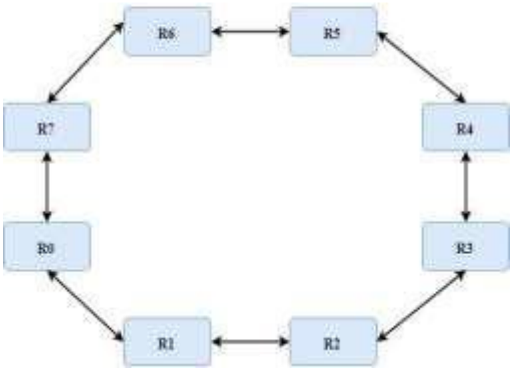


Fig (5). Ring topology

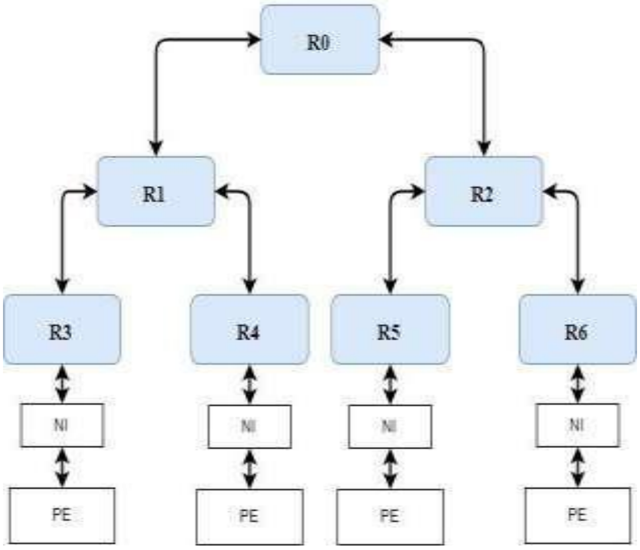
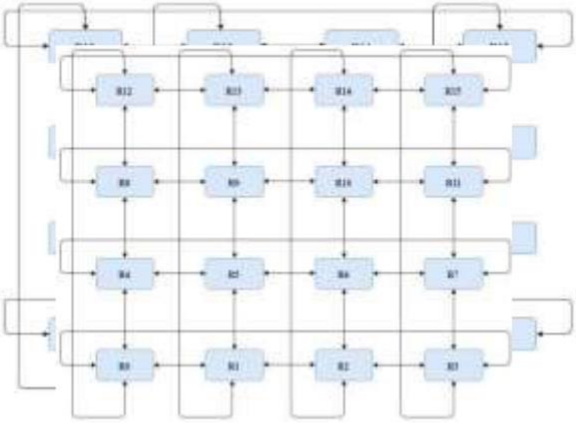


Fig (6). Tree topology

Fig (7). Irregular topology



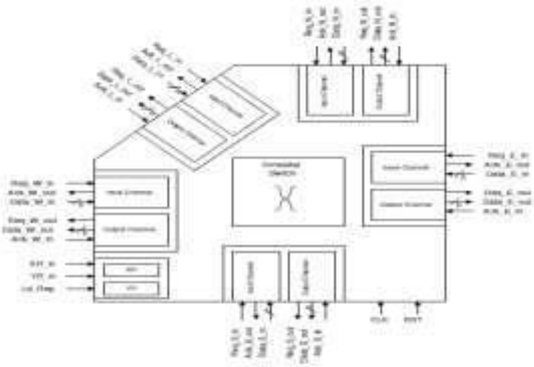


Figure 8 Router Architecture

2. Results and Conclusion

We utilize the Xilinx ISE 9.2i, specifically targeting the XC2V3000 FPGA [15], to functionally verify a 3x3 mesh-based router and the overall NoC system. The synthesis process is performed using Xilinx ISE 9.2i [15], while ModelSim SE 6.3f [13] is employed to simulate the model and generate activity data from the Placed-And-Routed (PAR) model. The XPower tool within Xilinx ISE 9.2i is used to estimate the power consumption of the designs. The router core is implemented in VHDL in a modular manner. Key parameters, such as data width and FIFO depth, are configurable. In this work, the data width is fixed at 8 bits (flit size). The coordinates of each router are provided as inputs via the primary I/O interface, making it necessary to initialize the routers with their coordinate values at the start of the simulation. Alternatively, the coordinates can be hardcoded, but the first approach offers greater flexibility, especially in dynamic reconfiguration environments.

We use the Synchronous FIFO v4.0 from Xilinx LogiCORE for buffering purposes. The parameters of the FIFO are customizable, allowing it to be adapted to meet specific system requirements. The FIFO can be implemented using either Block RAM (BRAM) or Distributed RAM (DRAM).

The flow control is handshake-based, with minimal decoding logic. To reduce bottlenecks in the store-and-forward buffering scheme, both input and output channels are buffered. We implement XY routing, and the corresponding finite state machines (FSMs) and decoding logic have been optimized accordingly.

The arbitration scheme is dynamic, featuring a round-robin arbiter with a dynamic priority mechanism. This ensures fair resource allocation under varying traffic conditions.

Initially, we test a single router by feeding random inputs in such a way that no blocking occurs on any of the output channels. The router is capable of establishing five simultaneous connections in parallel. In Figure 4, five simultaneous requests are serviced by the router. Each of the five input channels requests a different output, allowing the router to process and transmit all five requests concurrently through its output channels.

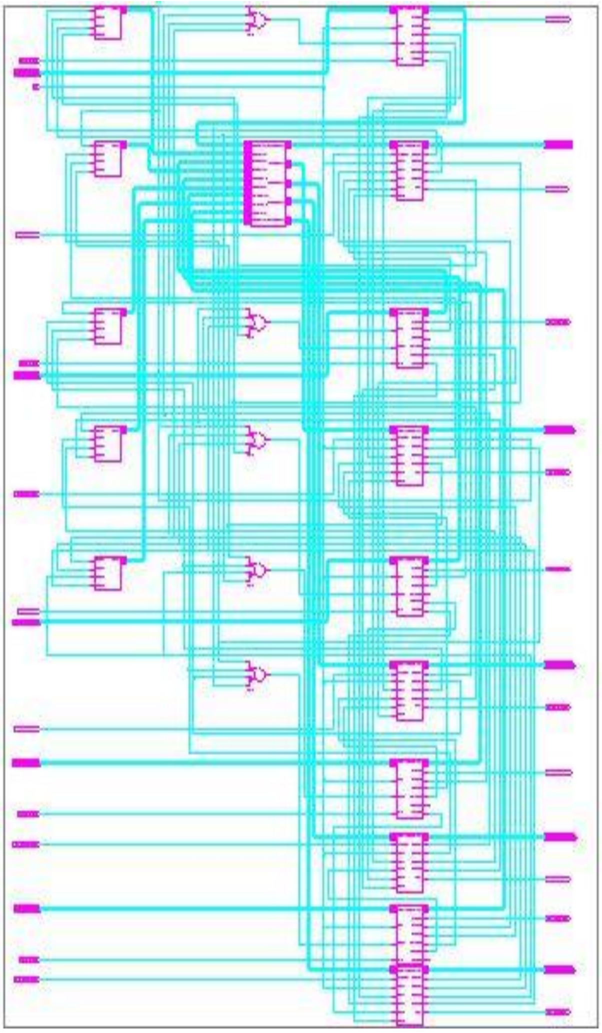


Figure.9: RTL View of NoC Route

Figure 10. waveform of input Channel

Figure 11: waveform of North Output Channel

8. Conclusion

This paper various **NoC routing algorithms** based on key parameters such as latency, throughput, power consumption, congestion awareness, and fault tolerance. From our analysis, it is evident that most researchers focus on developing routing algorithms that aim to minimize latency, maximize throughput, reduce power consumption, and address congestion and fault tolerance. **Table 1** highlights that significant progress has been made in these areas, but there are still unresolved challenges that require further exploration. In our view, this review is a valuable resource for the research community, providing insight into the existing gaps and encouraging future efforts to tackle these challenges.

It is also important to note that while designing routing algorithms, researchers often face trade-offs between different performance metrics. As a result, designers must balance competing factors, such as latency and power consumption, or throughput and fault tolerance, depending on the specific application requirements.

9. References

[1]. Ruirui Shen, Yanjun Wang, Xiang Liu, and Qian Xu, "Reinforcement Learning Assisted Routing Algorithm for 3D Network-on-Chip," in *2023 IEEE 9th International Conference on Computer and Communications (ICCC)*, 2023, pp. 507-511. DOI: 10.1109/ICCC58281.2023.00106.

[2]. Z. Cao, Y. Wang, J. Gao, and H. Li, "An Efficient Congestion-Aware Routing Algorithm for 2D Mesh-based Network-on-Chip," *IEEE Transactions on Computers*, vol. 72, no. 5, pp. 920-933, May 2023. DOI: 10.1109/TC.2022.3175371.

[3]. A. V. de Mello, L. C. Ost, F. G. Moraes, and N. L. V. Calazans, "Evaluation of routing algorithms on mesh-based NoCs," PUCRS, Av. Ipiranga, p. 22, 2004.

[4]. Mejia, M. Palesi, J. Flich, S. Kumar, P. López, R. Holsmark, et al., "Region-based routing: a mechanism to support efficient routing algorithms in NoCs," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 17, pp. 356-369, 2009.

[5]. G. Adamu, M. P. Chejara, and A. B. Garko, "Review of Deterministic Routing Algorithm for Network-On-Chip," in *2nd International Conference on Science, Technology and Management*, 2015, pp. 741-745.

[6]. S. D. Chawade, M. A. Gaikwad, and R. M. Patrikar, "Review of XY routing algorithm for network-on-chip architecture," *International Journal of Computer Applications*, vol. 43, pp. 975-8887, 2012.

[7]. P. Guerrier and A. Greiner, "A generic architecture for on-chip packet-switched interconnections," in *Proceedings of the Conference on Design, Automation and Test in Europe*, 2000, pp. 250-256.

[8]. S. Kumar, A. Jantsch, J.-P. Soininen, M. Forsell, M. Millberg, J. Oberg, et al., "A network on chip architecture and design methodology," in *Proceedings IEEE Computer Society Annual Symposium on VLSI. New Paradigms for VLSI Systems Design. ISVLSI 2002*, 2002, pp. 117-124.

[9]. D. Atienza, F. Angiolini, S. Murali, A. Pullini, L. Benini, and G. De Micheli, "Network-on-chip design and synthesis outlook," *Integration*, vol. 41, pp. 340-359, 2008.

[10]. P. Liu, B. Xia, C. Xiang, X. Wang, W. Wang, and Q. Yao, "A networks-on-chip architecture design space exploration—the LIB," *Computers & Electrical Engineering*, vol. 35, pp. 817-836, 2009.

[11]. F. Safaei and M. ValadBeigi, "An efficient routing methodology to tolerate static and dynamic faults in 2-D mesh networks-on-chip," *Microprocessors and Microsystems*, vol. 36, pp. 531-542, 2012.

[12]. M. A. Javed Sethi, F. A. Hussin, and N. H. Hamid, "Review of network on chip architectures," *Recent Advances in Electrical & Electronic Engineering (Formerly Recent Patents on Electrical & Electronic Engineering)*, vol. 10, pp. 4-29, 2017.

[14]. Wang, W.-H. Hu, S. E. Lee, and N. Bagherzadeh, "Area and power-efficient innovative congestion-aware network-on-chip architecture," *Journal of Systems Architecture*, vol. 57, pp. 24-38, 2011.

[15]. G. Adamu, P. Chejara, and A. B. Garko, "Review of Deterministic Routing Algorithm For Network-On-Chip," in *2nd International Conference on Science, Technology and Management*, 2015, pp. 741-745.

[16]. P. Parandkar, J. Dalal, and S. Katival, "Performance comparison of XY, OE and DyAD routing algorithm by load variation analysis of 2-Dimensional Mesh Topology Based Network-on-Chip," *BIJIT Journal*, vol. 4, pp. 391-396, 2012.

[17]. A. Singh, W. J. Dally, A. K. Gupta, and B. Towles, "GOAL: a load-balanced adaptive routing algorithm for torus networks," in *30th Annual International Symposium on Computer Architecture*, 2003. Proceedings., 2003, pp. 194-205.

[18]. Singh, W. J. Dally, B. Towles, and A. K. Gupta, "Globally adaptive load-balanced routing on tori," *IEEE Computer Architecture Letters*, vol. 3, pp. 2-2, 2004.

[19]. M. Li, Q.-A. Zeng, and W.-B. Jone, "DyXY: a proximity congestion-aware deadlock-free dynamic routing method for network on chip," in *Proceedings of the 43rd Annual Design Automation Conference*, 2006, pp. 849-852.

[20]. X. Canwen, Z. Minxuan, D. Yong, and Z. Zhitong, "Dimensional bubble flow control and fully adaptive routing in the 2-D mesh network on chip," in *2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*, 2008, pp. 353-358.