



2D Mesh Topology in Clusters for Network-on-Chip Architecture at Large Scales

M. Baboli^{*a}, H. Asgharpour-Alamdari^b

^a Department of Electrical Engineering, ARYAN institute of science and Technology, Babol, Iran

^b Department of Electrical Engineering, National University of Skill (NUS), Tehran, Iran

PAPER INFO

Paper history:

Received 21 March 2025

Received in revised form 18 April 2025

Accepted 09 May 2018

Keywords:

Network Topology

Clustered 2-D Mesh Topology

Network on Chip Architecture

RaMesh Topology

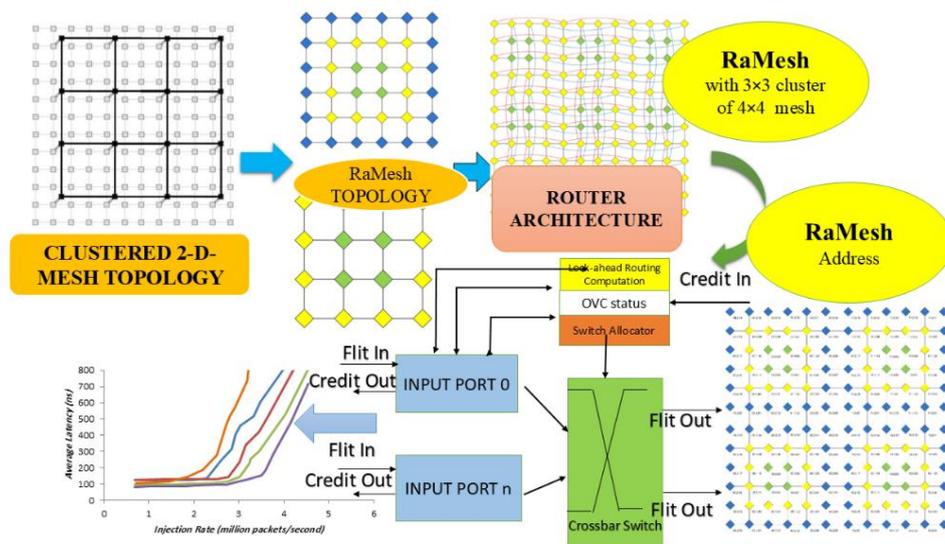
Altera ModelSim

ABSTRACT

It is anticipated that the number of processing cores in chip multiprocessors and systems-on-chip will increase significantly shortly, soon due to Moore's law's continual growth. The task of efficiently and scalably connecting the various parts of a multiprocessor device has grown more difficult. The network-on-chip (NoC) topologies that are now in use are suitable for small networks but not well-suited for big networks. Due to the longer routes needed to get to their destinations, sent packets inside a big NoC result in higher performance metrics like latency and power consumption. Therefore, creating a new topology suitable for large-size NoCs is required. In this study, we suggested an affordable network design that enhances end-to-end latency performance for large-scale NoCs. The RaMesh topology is made up of mesh network clusters. Additionally, a routing technique appropriate for this architecture was suggested. Altera ModelSim was used to simulate the RaMesh architecture for Verilog hardware models, along with mesh, torus, and clustered 2D mesh. Various network traffic scenarios and network sizes were used in the simulations. According to experimental data, RaMesh outperformed torus, clustered 2D mesh, and similar 2D mesh topologies. Additionally, RaMesh topology was compared to another clustered mesh topology for benchmarking. In comparison to mesh, torus, and clustered 2D mesh, the suggested topology had an average hop count that was at least 31% lower. Additionally, the average latency was reduced by at least 25% when compared to mesh, torus, and clustered 2D mesh.

doi: 10.5829/ije.2026.39.02b.07

Graphical Abstract



*Corresponding Author Email: m.baboli@aryan.ac.ir (M. Baboli)

Please cite this article as: Baboli M, Asgharpour-Alamdari H. 2D Mesh Topology in Clusters for Network-on-Chip Architecture at Large Scales. International Journal of Engineering, Transactions B: Applications. 2026;39(02):361-72.

1. INTRODUCTION

For high-performance computation, a multi-processor system-on-chip (MPSoC) may support a large number of processing resources (1-3). On-chip communication is MPSoC's primary bottleneck. The efficiency and scalability of connecting several cores on a single device are not possible with traditional bus-based on-chip interconnects (4, 5). To address the issues of on-chip connectivity, network-on-chip (NoC) has been proposed (6-8). NoCs are made up of networked routers that combine memory, processing power, or intellectual property (IP) components according to a certain topology (such as a mesh) (9). Unlike bus-based interconnects, which use circuit switching (10), NoC-based systems use packet transfer via routers to facilitate communication between IPs (11). Today, MPSoCs with hundreds or thousands of cores are probably standard (12). A communication network with high bandwidth and scalability is necessary as the number of on-chip cores increases (13, 14). NoCs were introduced to meet these needs, and they have rapidly become the dominant interconnection fabric (15, 16).

Topologies are progressively turning into the bottleneck restricting NoC performance (17, 18). In fact, a large-scale NoC's topology significantly affects both the network's cost and performance (19, 20). It is in charge of between 60% and 75% of the delay (21). The two-dimensional mesh is the traditional NoC topology (22, 23). Because of its straightforward implementation and extremely regular overall architecture, it is favored over alternative topologies (24, 25). Nevertheless, despite its benefits, the two-dimensional mesh topology has drawbacks for large-scale NoCs, including high hop count, congestion, and communication delay (26). In fact, the mesh topology's wide communication radius, which results in a lengthy packet delivery path, is a major drawback (27). Mesh topology has been shown to be effective for networks with up to 64 nodes (28, 29). However, mesh topology performance drastically deteriorates for large-scale NoC networks (30, 31). Mesh topology's performance is not scalable as networks get bigger (32). Another popular topology for NoCs is the torus (33). The torus architecture has a lot of long-range links that might cause issues with cost and performance (34). It takes longer for a packet using a long-range link to get to the next hop than a packet using a regular link. Every lengthy link also has a minimum delay and might cause problems (35). Long-distance networks, on the other hand, may enhance performance by cutting down on hops. Given the aforementioned drawbacks, a topology with low hop count and network latency must be developed (36). Combining mesh and ring topologies may help large-scale networks avoid congestion and deal with latency and hop count (15, 29).

Current network-on-chip (NoC) topologies are adequate for small-size networks, but they are not suitable for large-scale networks. Thus, it is necessary to design and develop a new topology which is appropriate for large-size NoCs. The proposed topology consists of clusters of mesh networks, and is called RaMesh. A routing algorithm suitable for this topology is also proposed. As a result, the RaMesh NoC topology is proposed in this study. Because of its cluster-based design, RaMesh works well with big NoCs with more than 100 IP cores. A mesh topology is used for each cluster. Nonetheless, the cluster's internal IP core-to-core communication follows the ring topology rule. Large-scale NoCs with the suggested architecture have fewer switches, hops, and latency. Avoiding congestion and having a low average network latency and hop count are among the goal performance metrics.

In this paper, a 2-D mesh topology in clusters is evaluated for large-scale network-on-chip architecture. In section 1, several articles were evaluated and the results were presented. In the next section, a topology is proposed based on the torus topology and concerning hybrid ring/mesh topologies. Furthermore, in Section 3, a suitable routing technique for this architecture was proposed. In Section 4, ModelSim was used to simulate the RaMesh architecture in Verilog hardware models, along with mesh, torus, clustered 2D meshes and proposed topology. Different network traffic scenarios and network sizes were used in the simulations. The experimental work was carried out in four parts and its results were analyzed. Finally, conclusions are drawn from the discussions.

2. RELATED WORKS

This study proposes a NoC topology based on torus, ring road mesh, clustered 2D mesh, and hybrid ring/mesh topologies. The usage of hybrid ring/mesh and clustered 2D mesh topologies is justified by their demonstrated ability to improve network performance by lowering average latency and hop count.

The proposed topology overcomes the shortcomings of clustered 2D mesh (Figure 1) by omitting the backbone switches and using bypass links to link the clusters. The bypass links connect the clusters both vertically and horizontally. These bypass links in the proposed topology are performed with switches that are attached to IP, reducing hardware by removing switches used solely for connecting clusters with no IP attachment, as was done in clustered 2D mesh proposed by Owens (37).

In RaMesh topology, the internal architecture of each cluster is also optimized. RaMesh topology uses a mesh cluster that adopts a ring road technique that avoids the congestion at the center of mesh topology. The following are the primary characteristics of the suggested topology:

- A) While each cluster architecturally adheres to the mesh topology, internal communication between IP cores within the cluster is governed by the ring topology rule. Every cluster's central mesh topology may be made less congested by using the ring road strategy. A 4x4 or 6x6 mesh network makes up each cluster.
- B) In addition to regular connections, there are certain long-distance links that serve as bypass links for communication between distant source and destination pairs.

The following sections describe the design and development of RaMesh architecture. The base architecture is illustrated in Figure 2(a), which is a 6x6 mesh network that consists of three layers or rings. The 6x6 mesh cluster consists of three layers: the internal layer (green layer in Figure 2(a)), the interval layer (in yellow), and the external layer (blue layer). The 4x4 mesh cluster consists of two layers: the internal layer (green layer in Figure 2(b)) and the external layer (yellow layer).

3. RAMESH TOPOLOGY

In this section, the RaMesh topology and its switches will be introduced. Figure 3 shows RaMesh architecture with a 2x2 cluster, where each cluster is a 6x6 network (accommodating 144 nodes) and Figure 4 shows RaMesh architecture with a 3x3 cluster, where each cluster is a

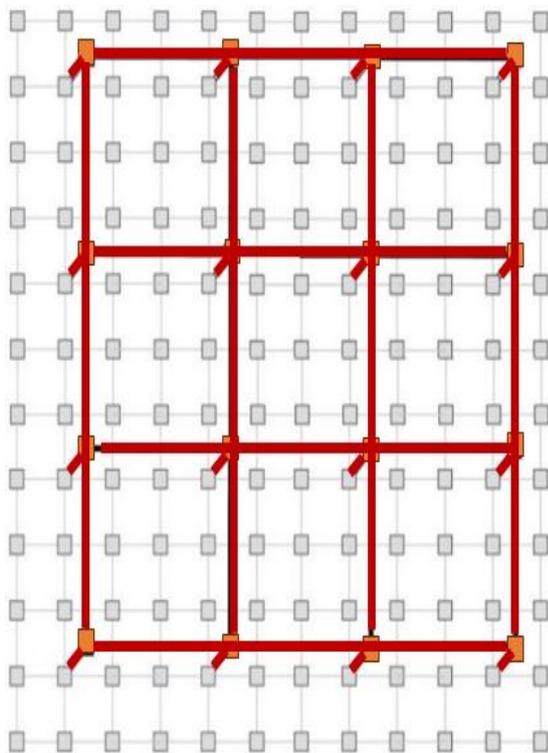


Figure 1. Clustered 2D mesh topology

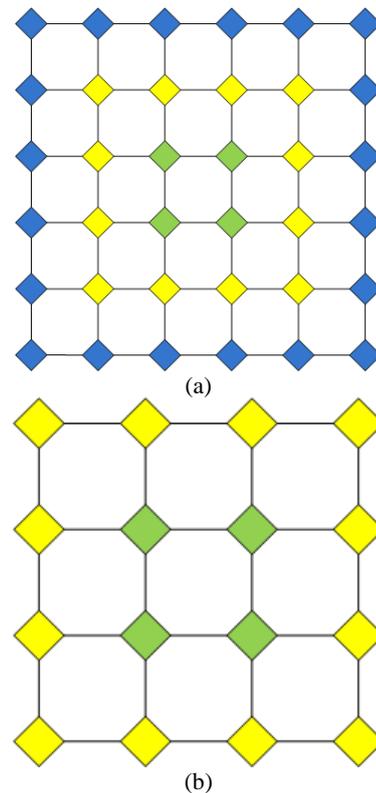


Figure 2. A mesh cluster in RaMesh topology

4x4 network (also accommodating 144 nodes). As can be seen in Figures 3 and 4, bypass links are used to connect two neighbor clusters in RaMesh. RaMesh has only two kinds of switches. The switch for the internal and interval layers of a cluster is called a local switch (LS). Figure 5(a) illustrates the five ports that each LS possesses. To connect to an IP core, one port is set aside. The other four ports called UPLY, DNLY, FRWD, and RVRS are used to connect to neighbor switches. UPLY and DNLY ports are used for sending packets between layers, FRWD port is used for sending packets to own layer in the forward direction, and RVRS port is used for sending packets to own layer in the reverse direction. In RaMesh architecture, forward direction refers to the movement of packets in the same layer in the counter-clockwise direction, whereas the reverse direction is in the clockwise direction. The switch for the external layer has seven ports. Two of the ports are utilized to link to neighbor clusters, four are connected to four neighbor switches, and the fifth is connected to the IP core. This switch is called ES (External Switch), as shown in Figure 5(b).

The four ports used to connect to neighbor switches are called UPLY, DNLY, FRWD, and RVRS. The long-range links to connect to neighbor clusters use ports the BPSF and BPSR. BPSF port sends packets to the next cluster, while the BPSR port sends packets to the previous cluster.

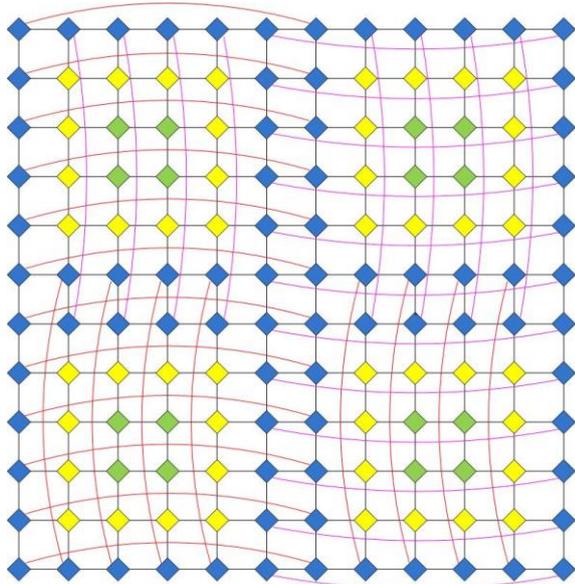


Figure 3. RaMesh with 2x2 cluster of 6x6 mesh

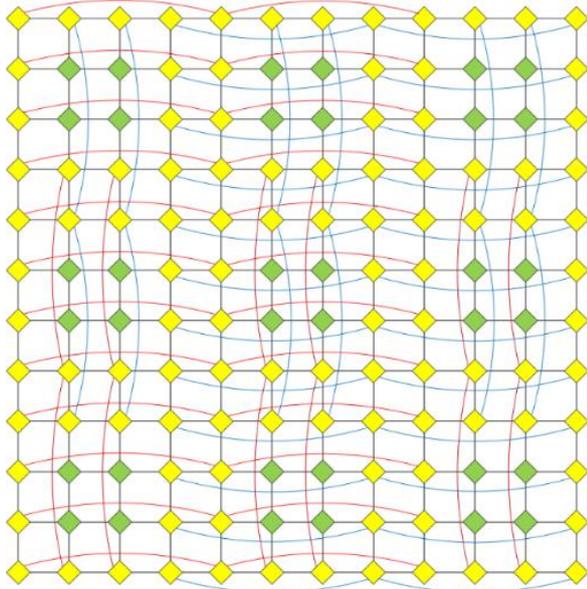


Figure 4. RaMesh with 3x3 cluster of 4x4 mesh

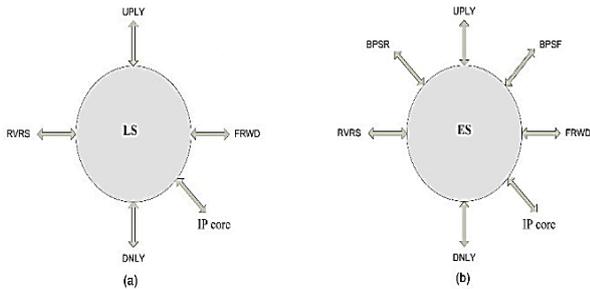


Figure 5. (a) LS switch for internal and interval layer, (b) ES switch for external layer

Equations 1, 2, and 3 determine the number of IP cores (C), switches (N), and channels between switches (Ch) in RaMesh; where n is the number of switches in each row of a cluster (n is six in the example in Figure 3), $x = 0, 1, 2, \dots$ is the number of clusters in the horizontal direction (x is one in the example in Figure 3), and $y = 0, 1, 2, \dots$ is the number of clusters vertically (y is one in the example). Thus, N , C , and Ch for RaMesh4 are 144 switches, 144 IP cores, and 304 channels respectively. Where, x is the width axis and y is the length axis of the node.

$$C = n^2(x + 1)(y + 1) \tag{1}$$

$$N = n^2(x + 1)(y + 1) \tag{2}$$

$$Ch = n(y + 1)(n(x + 1) - 1) + n(x + 1)(n(y + 1) - 1) + 2nx(y + 1) + 2ny(n - 2)(x + 1) \tag{3}$$

4. ADDRESSING IN RAMESH

For packets to be sent within NoC, each accessible destination has to have a unique address. Figure 6 shows RaMesh topology to connect 144 IP cores using 6x6 mesh clusters. The IP address in RaMesh is 16 bits, partitioned into three parts, as shown in Table 1. The first part of the address (X and Y) is reserved to indicate the cluster address, Part Two (L) refers to the layer of the cluster, and the third part (P) refers to the IP number, which is uniquely assigned for each cluster layer. The first part of the address needs eight bits, Part Two needs three bits, and the third one is allocated five bits.

The first part consisting of eight bits, is divided into two 4-bit parts. The first nibble (the X address field in Table 1) indicates cluster location in the horizontal direction, and the second nibble (the Y address field in Table 1) indicates cluster location in the vertical direction.

4. 1. Routing Algorithm for RaMesh There are two steps in RaMesh's routing algorithm for transmitting packets from source to destination. The first stage, which is related to the first part of the address is applied to find the destination cluster. The second stage is used for finding the destination IP inside the cluster. The first step involves comparing the first part of the current address with the first part of the destination address (the XY field). The X address field of the destination and the X address field of the current node will be compared first. If they are not equal, then packets should be moved either right or left on the X -axis, until they are equal. Then the Y address field of the destination and the Y address field of the current location will be compared next. If they are not equal, then the packet should be moved either up or down on the Y -axis, until they are equal. In this first stage, the destination cluster will be found.

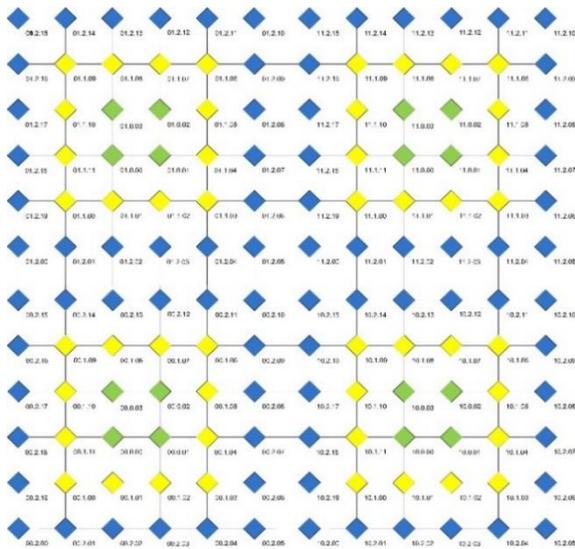


Figure 6. Address for RaMesh

TABLE 1. Address for RaMesh

Address of cluster		Layer number	IP identity for each switch
XY			
X	Y	L	P
4 bits	4 bits	3 bits	5 bits

In the second stage of routing, the L address field of the destination and the current node will be compared to each other. Finally, the P address field of the destination and current node will be compared together. This process will be repeated until the packet reaches the destination.

5. EXPERIMENTAL WORK

The topologies of RaMesh, torus, mesh, and clustered 2D mesh (see Figure 1), together with the routing method that goes with them were modeled in Verilog HDL and then simulated using ModelSim to obtain average latency, average hop count, hardware cost, and maximum operating frequency. We used a reference design router with low latency micro-architecture (37). The router architecture is illustrated in Figure 7. The experimental work had four parts, described as follows.

- I. In the first part, average latency and average hop count for topologies with 36 to 3,600 cores for random traffic were obtained.
- II. Part Two was similar to the first part but was done for different traffic models for different network sizes to obtain average latency.
- III. The third part was an experiment to find hardware cost or area consumption.
- IV. The final part was timing analysis to obtain average latency in ns.

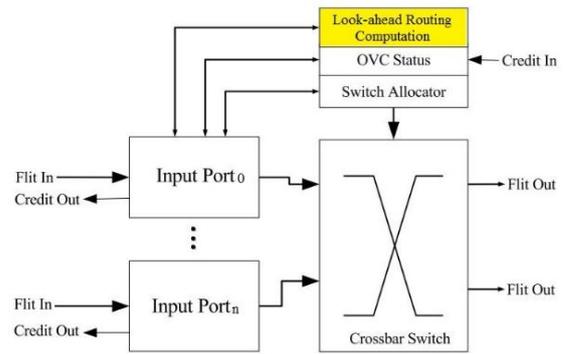


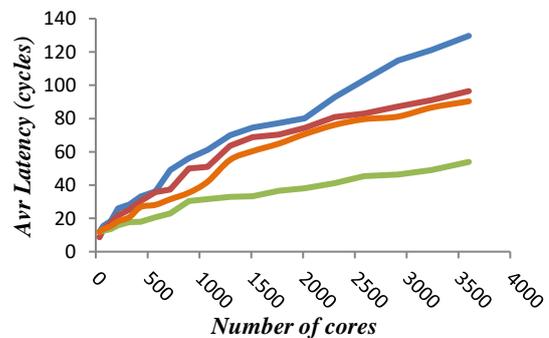
Figure 7. Router architecture

5. 1. Part One In the first part of this experimental work, torus, mesh, clustered 2D mesh, and RaMesh topologies were modeled in Verilog and were simulated using ModelSim to obtain average latency and average hop count.

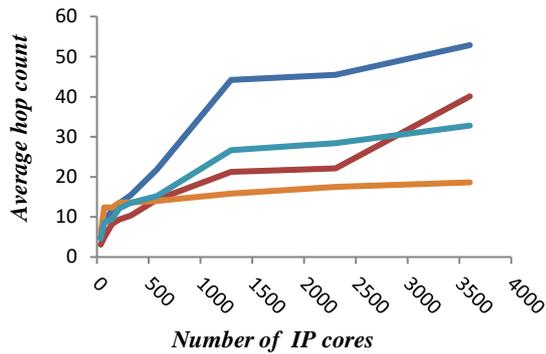
This experiment was done under random traffic, the constant injection rate (0.005 packets/cycle/node), and to send 1,000-100,000 packets to the destination for 36 to 3,600 cores. The results of this comparison are a preliminary result to understand which topology has better performance than others. Each simulation was run for a specific packet size, which is 1,000, 5,000, 10,000, 50,000, and 100,000 packets. For each packet size, the simulation was run twice. In total, 10 simulations were run for each a NoC size before average latency and average hop count were taken.

The result from this experiment is shown in Figure 8. The result shows that RaMesh has the lowest average latency and average hop count. On average, RaMesh is 49.5%, 41.5%, and 31.1% better than mesh, torus, and clustered 2D mesh topologies respectively in terms of average latency for NoC size of 3,600.

5. 2. Part Two The Part Two was similar to the first part but with more robust input conditions. The experiment was done for random, hotspot, transpose, and uniform distribution traffic instead of only random



(a)



(b)

Figure 8. (a) Average latency and (b) Comparison of the average hop count for various NoC sizes across torus, RaMesh, mesh and clustered 2D mesh topologies

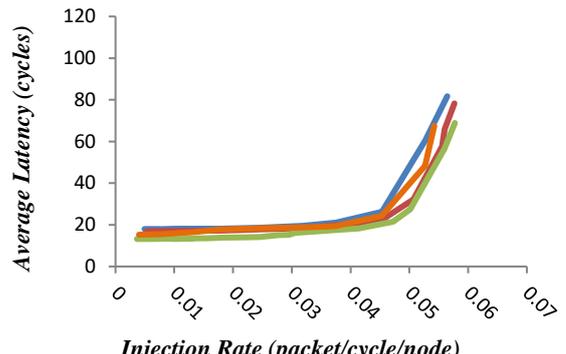
traffic. It also was done under different injection rates of traffic to send 10,000 packets to destinations. We evaluated these topologies for different network sizes with 144, 324, 576, and 900 IP cores.

5. 2. 1. Result of Part Two with Different Injection Rates of Random Traffic

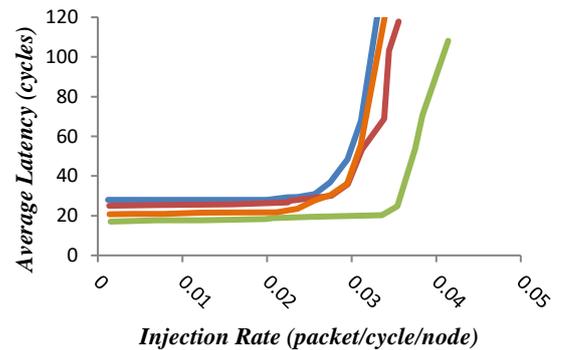
In random traffic, every router sends packets to another router where the destination node was chosen randomly. Figure 9 shows the average latency under different random traffic injection rates for the networks with 144, 324, 576, and 900 IP cores. As can be seen in Figure 9, RaMesh topology on average is 39.2, 46.91, and 40 percent better in terms of average latency performance than clustered 2D mesh, mesh and torus topologies respectively under random traffic for NoC with 900 IPs at an injection rate of 0.02 packets/cycle/node.

5. 2. 2. Result of Part Two with Different Injection Rates of Transpose Traffic

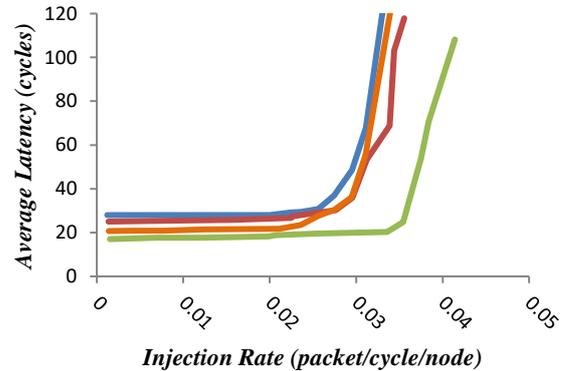
In transpose traffic, every source router sends packets to a destination node which is at the transpose location in the network. For a source node (i, j) the destination node is (N-1-i, N-1-j) for a network with dimension N×N. For example, node (1, 0) only sends packets to node (4, 5) for a 6×6 mesh network. Figure 10 shows the average latency transposed traffic for the networks at varying injection rates with 144, 324, 576, and 900 IP cores. In transpose traffic, a packet is mostly transferred through the nodes at the edges of the topology. Torus topology has long links that connect the edges together. Therefore, most of the network traffic in torus topology with transpose traffic passes through these long links. The congestion in the long links results in queuing delay and hence latency increases. Thus, the torus topology under transpose traffic gets saturated earlier than mesh. As can be seen in Figure 10, under transpose traffic, the performance of RaMesh on average is 43.7, 49.53, and 55.73 percentage better than torus, mesh, and clustered 2D mesh topologies respectively.



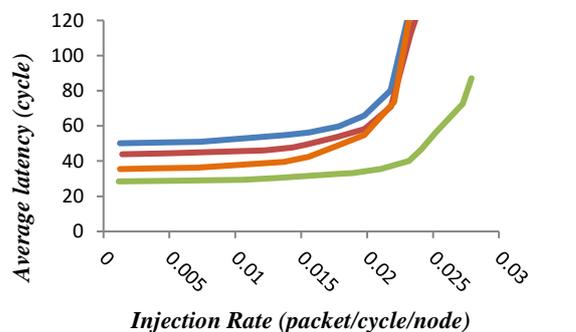
(a) 144 IP cores



(b) 324 IP cores



(c) 576 IP cores



(d) 900 IP cores

- Mesh - Torus - RaMesh - Winter

Figure 9. Torus, mesh, clustered 2D mesh and RaMesh average Latency (cycles) for varying topology sizes under varying random traffic injection rates

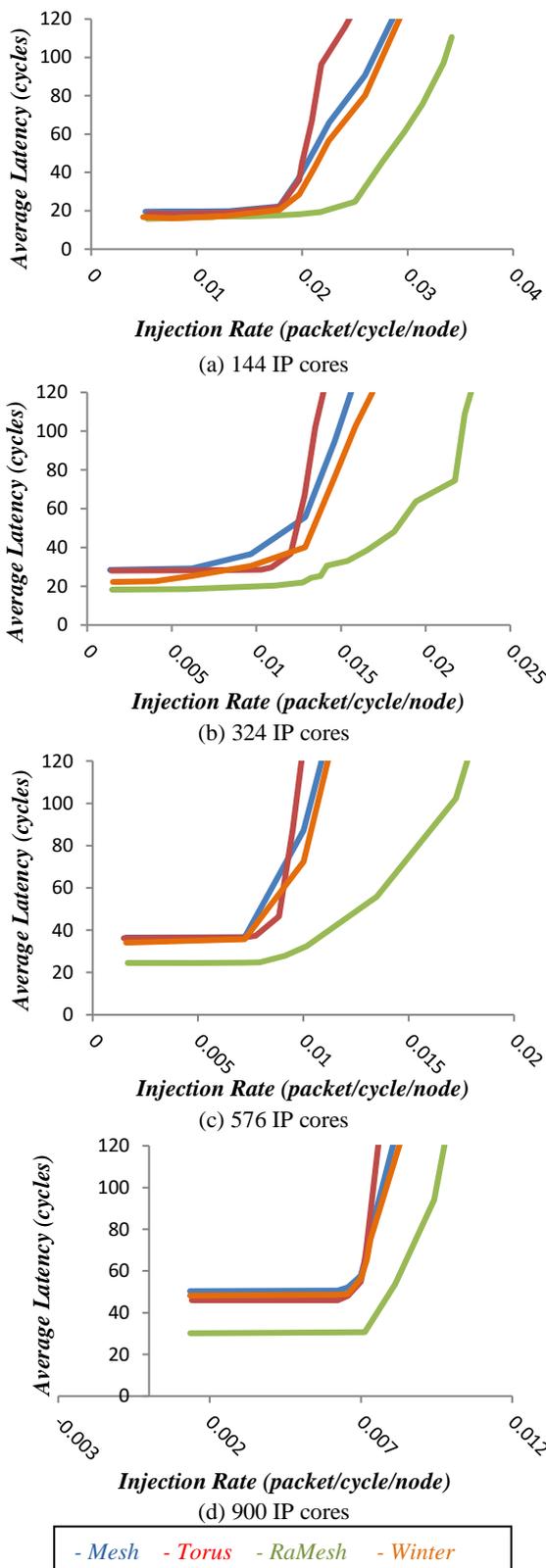


Figure 10. Torus, mesh, clustered 2D mesh and RaMesh average Latency (cycles) for varying topology sizes under varying transpose traffic injection rates

5. 2. 3. Result of Part Two with Different Injection Rates of Hotspot Traffic

In hotspot traffic, the packets are routed towards only a few fixed nodes. Hotspot traffic conditions can check the effectiveness of a routing algorithm and how the congestion across the node is balanced. Under the hotspot traffic several routers are chosen as hotspots, receiving an extra portion of traffics in addition to the regular uniform traffic. In this experiment, we have selected five nodes as hotspot nodes, and they received 5% of the total traffic. Figure 11 shows the average latency with varying hotspot traffic injection rates. The performance of RaMesh topology on average is 61.325, 59.28 and 49.165 percent better than clustered 2D mesh, mesh and torus respectively under hotspot traffic. As shown in Figure 11, clustered 2D mesh topology has the worst results. Once a node is selected as a hotspot node, the other nodes should send greater traffic to the hotspot node. The extra traffic sent from the other nodes in clustered 2D mesh can access to hotspot node through one individual link. This leads to an increase in congestion resulting in a delay time to rise. Therefore, the clustered 2D mesh topology gets saturated earlier than the other topologies.

5. 2. 4. Result of Part Two with Different Injection Rates of Uniform Traffic

In this simulation, each source router sends data packets to all other routers using a uniform distribution traffic. Figure 12 shows average traffic injection rates for varying network sizes. Under uniform distribution traffic, RaMesh topology on the average is 48.25, 57.82, and 47.18 percent better than clustered 2D mesh, mesh and torus topologies respectively.

5. 3. Part Three

In this section, NoC implementation on FPGA is discussed. The analysis is based on implementation using Altera Stratix V 5SEEBF45I4 device on FPGA development board. The hardware evaluation was done for 144 IP cores utilizing Quartus II software, to obtain the number of adaptive logic modules (ALMs) and frequency estimation. Network size is limited to 144 because the Stratix device cannot support bigger networks. The logic elements (LEs) for recent Stratix V devices have a core reconfigurable fabric called logic array block (LAB) composed of ALM, which can be configured to implement logic, arithmetic, and memory functions. Half of the available LABs may be used as memory LAB (MLAB). The NoC code was written in Verilog HDL and compiled using Quartus II version 13 software. The code was verified using the Modelsim Altera version. In this FPGA, there are 359,200 ALMs, 2,640 memory blocks, and 14,374 registers. The hardware simulation model was verified through Verilog testbenches.

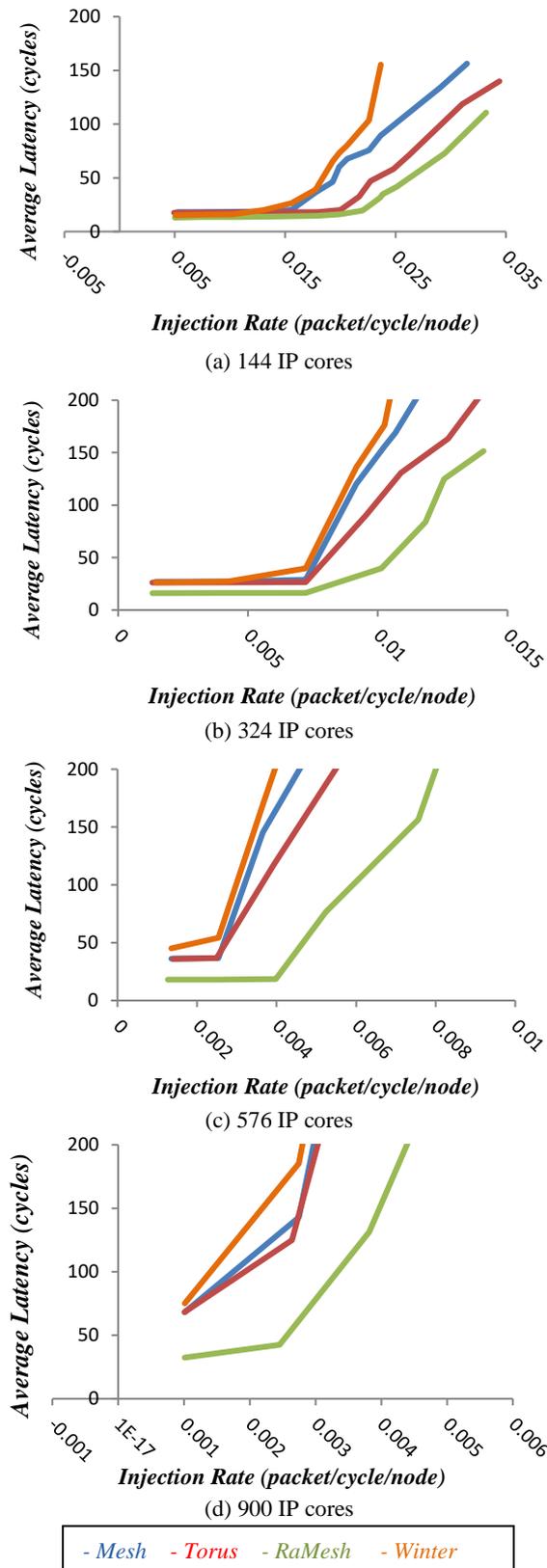


Figure 11. Torus, mesh, clustered 2D mesh and RaMesh average Latency (cycles) for varying topology sizes under varying hotspot traffic injection rates

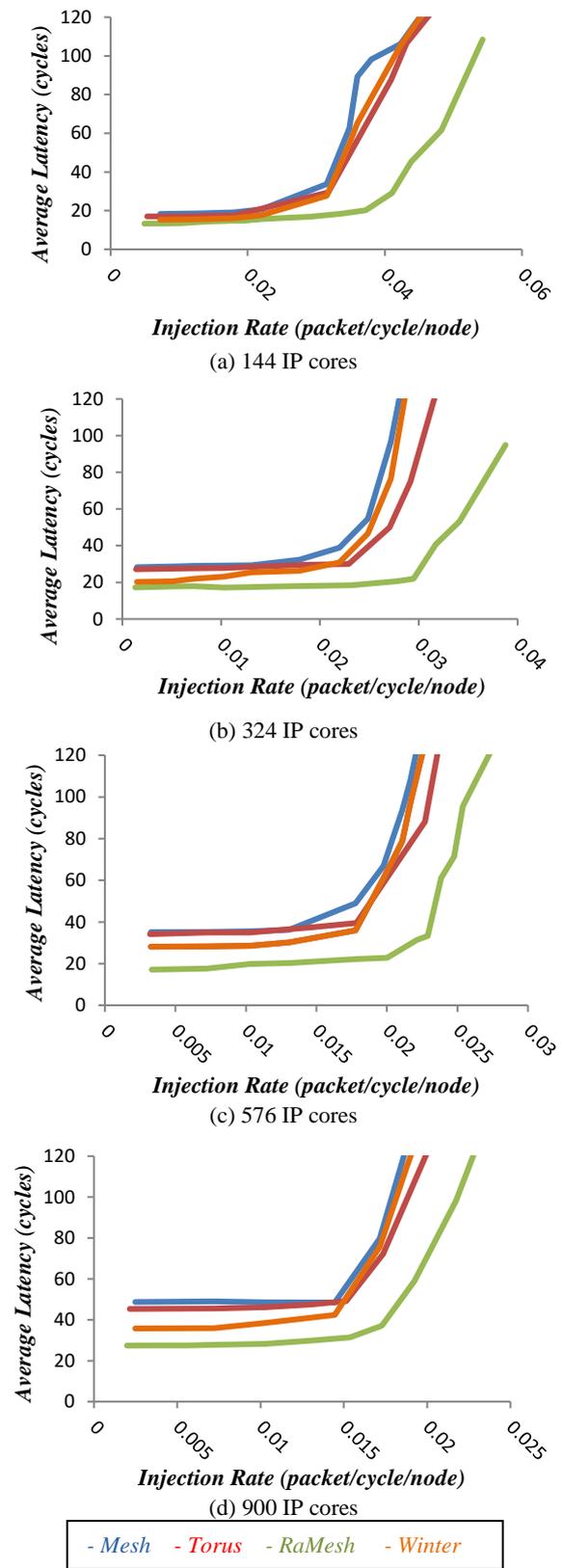


Figure 12. Torus, mesh, clustered 2D mesh and RaMesh average Latency (cycles) for varying topology sizes under varying uniform traffic injection rates

In this experiment, mesh, torus, clustered 2D mesh (see Figure 6), RaMesh_{6×6} (see Figure 2), and RaMesh_{4×4} (see Figure 3) for 144 IP cores that worked under wormhole routing were implemented using Verilog HDL. We used a reference design router with low latency micro-architecture (38). The router architecture is illustrated in Figure 7. Hence, we prototyped a NoC router using four VCs with a buffer size of four flits per VC and a flit size of 32 bits. In these experiments, the IP core is replaced by a packet generator. The result is summarized in Table 2, which shows the number of used ALMs, maximum operating frequency, critical path delay, and logic utilization.

5.3. Part Four Besides hardware cost and maximum operating frequency, the NoC topologies were also evaluated to find average latency in nanoseconds. The simulations were done under different injection ratios of traffic for hotspot, transpose, uniform, and random distribution traffic models. In this experimental work, the simulation result for average latency in nanoseconds is presented in Figure 13. In this figure, the Y axis is the average latency in ns, and the X axis is the packet injection rate (MP) in units of million packets per second, which is obtained through Equations 4 and 5. The average latency in nanoseconds is obtained using Equation 6.

TABLE 2. Overview of hardware usage for Stratix V Altera FPGA NoC implementation with 144 IP cores

Topologies	Mesh	Torus	Clustered 2D-mesh	RaMesh _{6×6}	RaMesh _{4×4}
ALMs	197264	209143	214658	215142	215454
Maximum operating frequency	142.8 MHz	138.3 MHz	140.1 MHz	136.3 MHz	135.8 MHz
Critical path delay	6.8ns	7.2ns	7.1ns	7.3ns	7.3ns
Memory blocks	672	716	752	752	824
Logic utilization (in ALMs)	55%	58.2%	59.8%	58.9%	60%

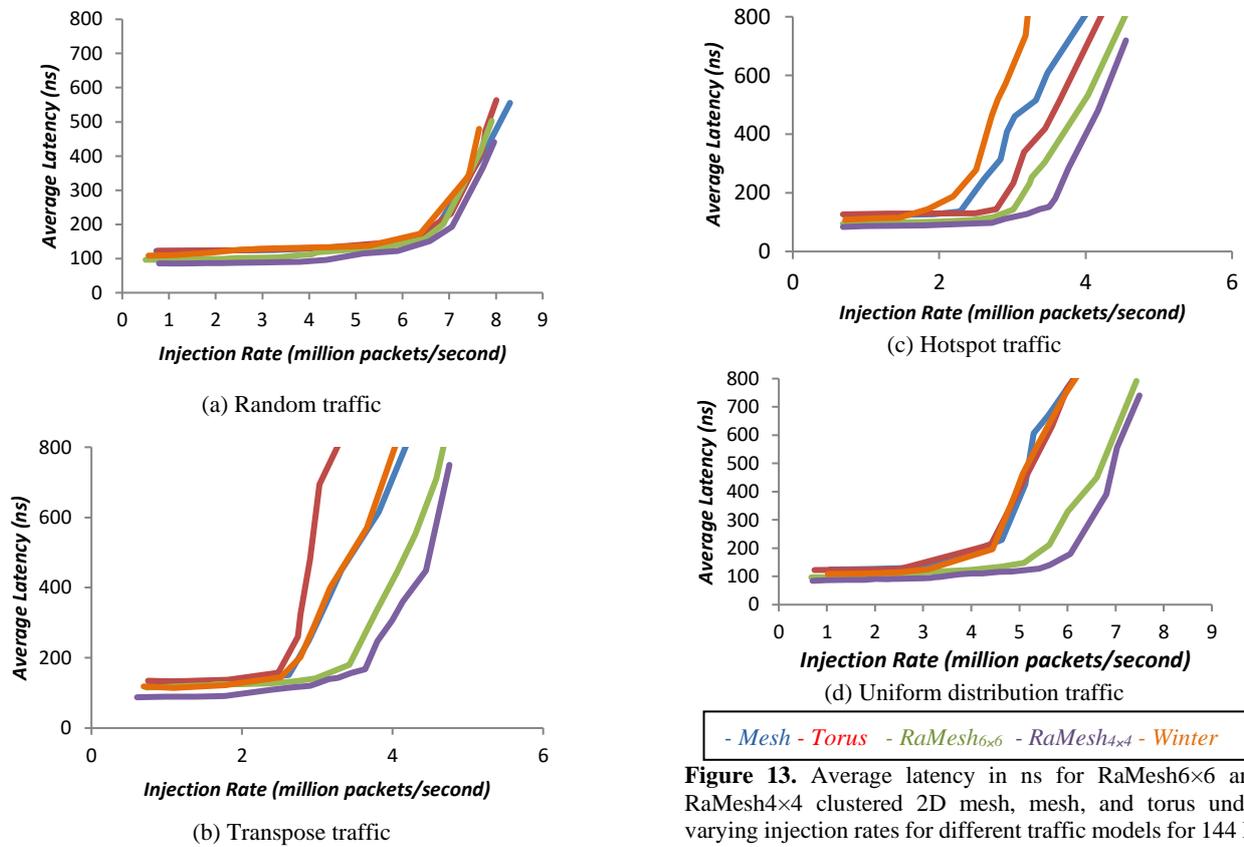


Figure 13. Average latency in ns for RaMesh_{6×6} and RaMesh_{4×4} clustered 2D mesh, mesh, and torus under varying injection rates for different traffic models for 144 IP cores is displayed in the timing study result

$$T = 1 / f_{max} \quad (4)$$

$$MP = p^{ir} / T \quad (5)$$

$$Avr\ lat(ns) = Avr\ lat \times T \quad (6)$$

For the equations, T is time for each period, MP is million packets per second, per is packet injection rate, and Avrlat is average latency in terms of clock cycles. Referring to Figure 13, it is seen that the proposed topology has better performance compared to other topologies.

Figures 13a, 13b, 13c, and 13d show average latency for random, transpose, hotspot, and uniform distribution traffic respectively. The performance of RaMesh is at least 24.61, 41, 41, and 50 percent better than other topologies under random, transpose, hotspot, and uniform distribution traffic for 144 IP cores respectively.

6. CONCLUSION

Given the importance of NoC design for large-scale topology, this paper analyzed several topologies and routing algorithms for large-scale topologies. Afterward, the problems of some existing topologies were demonstrated, and the reasons for the requirement of a new topology for large-scale NoC were discussed.

In this study, we presented an affordable network design that enhances latency performance for large-scale NoCs. Mesh network clusters make up the RaMesh topology. The proposed topology is an improved architecture for large-scale NoC. The Proposed topology together with its routing algorithm reduces average latency and average hop count. Compared to other topologies like mesh, torus, and clustered 2D mesh, the suggested topology is far more appropriate for the implementation of large-scale NoC because of its reduced average latency. Comparing the suggested topology to the mesh, torus, and clustered 2D mesh, the average hop count was at least 31% lower. The average latency was also reduced by at least 25% when compared to clustered 2D mesh, mesh, and torus.

7. REFERENCES

- Zhang X, Chen H, Wu Y. Evaluation and validation of efficient hierarchical interconnection in multi-processor system on chip. *Cyber-Physical Systems*. 2024;1-26. <https://doi.org/10.1080/23335777.2024.2326931>
- Seiler L, Carmean D, Sprangle E, Forsyth T, Dubey P, Junkins S, et al. Larrabee: A many-core x86 architecture for visual computing. *IEEE micro*. 2009;29(1):10-21. <https://doi.org/10.1109/MM.2009.9>
- Borkar S, editor *Thousand core chips: a technology perspective*. Proceedings of the 44th annual design automation conference; 2007. <https://doi.org/10.1145/1278480.1278667>
- Weerasena H, Mishra P. Security of Electrical, Optical, and Wireless On-chip Interconnects: A Survey. *ACM Transactions on Design Automation of Electronic Systems*. 2024;29(2):1-41. <https://doi.org/10.1145/3631117>
- Pouralizadeh Moghaddam F, Gholizade Narm H. Modelling and Compensation of Uncertain Time-delays in Networked Control Systems with Plant Uncertainty Using an Improved Robust Model Predictive Control Method. *International Journal of Engineering*. 2020;33(6):1134-41. <https://doi.org/10.5829/ije.2020.33.06c.11>
- Valinataj M. Reliability and performance evaluation of fault-aware routing methods for network-on-chip architectures. *International Journal of Engineering*. 2014;27(4):509-16. <https://doi.org/10.5829/idosi.ije.2014.27.04a.01>
- Lunnikivi H, Hämäläinen R, Hämäläinen TD. Keelhaul: Processor-Driven Chip Connectivity and Memory Map Metadata Validator for Large Systems-on-Chip. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*. 2024. <https://doi.org/10.1109/TVLSI.2024.3454431>
- Jalali G, Tavakkoli-Moghaddam R, Ghomi-Avili M, Jabbarzadeh A. A network design model for a resilient closed-loop supply chain with lateral transshipment. *International Journal of Engineering*. 2017;30(3):374-83. <https://doi.org/10.5829/idosi.ije.2017.30.03c.07>
- Hutchings D, Taylor A, Goeders J. Toward FPGA Intellectual Property Encryption from Netlist to Bitstream. *ACM Transactions on Reconfigurable Technology and Systems*. 2024;18(1):1-27. <https://doi.org/10.1145/3656644>
- Huynh PK, Mustafazade I, Cattoor F, Kandasamy N, Das A. A Scalable Dynamic Segmented Bus Interconnect for Neuromorphic Architectures. *IEEE embedded systems letters*. 2024. <https://doi.org/10.1109/LES.2024.3452551>
- Lee AY-P, Wang MI-C, Hung C-H, Wen CH-P. PS-IPS: Deploying Intrusion Prevention System with machine learning on programmable switch. *Future Generation Computer Systems*. 2024;152:333-42. <https://doi.org/10.1016/j.future.2023.11.011>
- Gonzalez-Martinez G, Sandoval-Arechiga R, Solis-Sanchez LO, Garcia-Luciano L, Ibarra-Delgado S, Solis-Escobedo JR, et al. A Survey of MPSoC Management toward Self-Awareness. *Micromachines*. 2024;15(5):577. <https://doi.org/10.3390/mi15050577>
- Dally WJ, Towles B, editors. *Route packets, not wires: on-chip interconnection networks*. Proceedings of the 38th annual design automation conference; 2001. <https://doi.org/10.1145/378239.379048>
- Liu Y, Liu P, Jiang Y, Yang M, Wu K, Wang W, et al. Building a multi-FPGA-based emulation framework to support networks-on-chip design and verification. *International Journal of Electronics*. 2010;97(10):1241-62. <https://doi.org/10.1080/00207217.2010.512017>
- Benini L, De Micheli G. Networks on chips: A new SoC paradigm. *computer*. 2002;35(1):70-8. <https://doi.org/10.1109/2.976921>
- Manzoor M, Mir RN. A Review of Design Approaches for Enhancing the Performance of NoCs at Communication Centric Level. *Scalable Computing: Practice and Experience*. 2021;22(3):347-64. <https://doi.org/10.12694/scpe.v22i3.1896>
- Salminen E, Kulmala A, Hamalainen TD. Survey of network-on-chip proposals. white paper, OCP-IP. 2008;1:13. https://www.academia.edu/download/49663645/Survey_of_Network-on-chip_Proposals.pdf

18. Sanchez D, Michelogiannakis G, Kozyrakis C. An analysis of on-chip interconnection networks for large-scale chip multiprocessors. *ACM Transactions on Architecture and Code Optimization (TACO)*. 2010;7(1):1-28. <https://doi.org/10.1145/1736065.1736069>
19. Bourduas S, Zilic Z, editors. A hybrid ring/mesh interconnect for network-on-chip using hierarchical rings for global routing. *First International Symposium on Networks-on-Chip (NOCS'07)*; 2007: IEEE. <https://doi.org/10.1109/NOCS.2007.3>
20. Fu W, Chen T, Liu L. Energy-efficient hybrid optical-electronic network-on-chip for future many-core processors. *Elektronika ir Elektrotechnika*. 2014;20(3):83-6. <https://doi.org/10.5755/j01.eee.20.3.6682>
21. Bahmani M, Reshadi M, Khademzadeh A, Reza A, editors. *Corona: Ring-based interconnected topology for on-chip network*. 2008 3rd International Design and Test Workshop; 2008: IEEE. <https://doi.org/10.1109/IDT.2008.4802497>
22. Gulzari UA, Salcic Z, Farooq W, Anjum S, Khan S, Sajid M, et al. Comparative analysis of 2D mesh topologies with additional communication links for on-chip networks. *Computer Networks*. 2024;241:110193. <https://doi.org/10.1016/j.comnet.2024.110193>
23. Lis M, Ren P, Cho MH, Shim KS, Fletcher CW, Khan O, et al., editors. Scalable, accurate multicore simulation in the 1000-core era. (IEEE ISPASS) *IEEE International Symposium on Performance Analysis of Systems and Software*; 2011: IEEE. <https://doi.org/10.1109/ISPASS.2011.5762734>
24. Taylor MB, Kim J, Miller J, Wentzlaff D, Ghodrati F, Greenwald B, et al. The raw microprocessor: A computational fabric for software circuits and general-purpose programs. *IEEE micro*. 2002;22(2):25-35. <https://doi.org/10.1109/MM.2002.997877>
25. Baboli M, Husin NS, Marsono MN, editors. A comprehensive evaluation of direct and indirect network-on-chip topologies. *Proceedings of the 2014 international conference on industrial engineering and operations management*; 2014. https://www.researchgate.net/profile/Mehdi-Baboli/publication/261364225_A_COMPREHENSIVE_EVALUATION_OF_DIRECT_AND_INDIRECT_NETWORK-ON-CHIP_TOPOLOGIES/links/00b495340f3ac3efbe000000/A-COMPREHENSIVE-EVALUATION-OF-DIRECT-AND-INDIRECT-NETWORK-ON-CHIP-TOPOLOGIES.pdf
26. Nychis GP, Fallin C, Moscibroda T, Mutlu O, Seshan S. On-chip networks from a networking perspective: Congestion and scalability in many-core interconnects. *ACM SIGCOMM computer communication review*. 2012;42(4):407-18. <https://doi.org/10.1145/2377677.2377757>
27. Qian Z, Bogdan P, Wei G, Tsui C-Y, Marculescu R, editors. A traffic-aware adaptive routing algorithm on a highly reconfigurable network-on-chip architecture. *Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis*; 2012. <https://doi.org/10.1145/2380445.2380475>
28. Bjerregaard T, Mahadevan S. A survey of research and practices of network-on-chip. *ACM Computing Surveys (CSUR)*. 2006;38(1):1-es. <https://doi.org/10.1145/1132952.1132953>
29. Johari S, Kumar A, Sehgal VK, editors. *Heterogeneous and Hybrid Clustered Topology for Networks-on-Chip*. 2015 7th International Conference on Computational Intelligence, Communication Systems and Networks; 2015: IEEE. <https://doi.org/10.1109/CICSyN.2015.41>
30. Chiu G-M. The odd-even turn model for adaptive routing. *IEEE Transactions on parallel and distributed systems*. 2000;11(7):729-38. <https://doi.org/10.1109/71.877831>
31. Baboli M, Husin NS. An Empirical Evaluation of Topologies for Large Scale NoC. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2014;12(12):8133-9. <http://dx.doi.org/10.11591/telkomnika.v12i12.6840>
32. Ezhumalai P, Chilambuchelvan A, Arun C. Novel NoC Topology Construction for High-Performance Communications. *Journal of Computer Networks and Communications*. 2011;2011(1):405697. <https://doi.org/10.1155/2011/405697>
33. Grecu C, Pande PP, Ivanov A, Saleh R. Timing analysis of network on chip architectures for MP-SoC platforms. *Microelectronics Journal*. 2005;36(9):833-45. <https://doi.org/10.1016/j.mejo.2005.03.006>
34. Sharma H, Pfromm L, Topaloglu RO, Doppa JR, Ogras UY, Kalyanraman A, et al. Florets for Chiplets: Data Flow-aware High-Performance and Energy-efficient Network-on-Interposer for CNN Inference Tasks. *ACM Transactions on Embedded Computing Systems*. 2023;22(5s):1-21. <https://doi.org/10.1145/3608098>
35. Howard J, Dighe S, Hoskote Y, Vangal S, Finan D, Ruhl G, et al., editors. A 48-core IA-32 message-passing processor with DVFS in 45nm CMOS. 2010 *IEEE International Solid-State Circuits Conference-(ISSCC)*; 2010: IEEE. <https://doi.org/10.1109/ISSCC.2010.5434077>
36. Owens JD, Dally WJ, Ho R, Jayasimha D, Keckler SW, Peh L-S. Research challenges for on-chip interconnection networks. *IEEE micro*. 2007;27(5):96-108. <https://doi.org/10.1109/MM.2007.4378787>
37. Winter M, Prusseit S, Gerhard PF, editors. *Hierarchical routing architectures in clustered 2D-mesh networks-on-chip*. 2010 *International SoC Design Conference*; 2010: IEEE. <https://doi.org/10.1109/SOCD.2010.5682890>
38. Monemi A, Ooi CY, Marsono MN. Low Latency Network-on-Chip Router Microarchitecture Using Request Masking Technique. *International Journal of Reconfigurable Computing*. 2015;2015(1):570836. <https://doi.org/10.1155/2015/570836>

COPYRIGHTS

©2026 The author(s). This is an open access article distributed under the terms of the Creative Commons Attribution (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, as long as the original authors and source are cited. No permission is required from the authors or the publishers.

**Persian Abstract****چکیده**

انتظار می‌رود تعداد هسته‌های پردازشی در پردازنده‌های چند تراشه‌ای و سیستم‌های روی تراشه‌ای به دلیل رشد مستمر قانون مور در آینده‌ای نزدیک به میزان قابل توجهی افزایش یابد. کار اتصال مؤثر و مقیاس‌پذیر بخش‌های مختلف یک دستگاه چند پردازنده دشوارتر شده است. توپولوژی‌های شبکه‌ای روی تراشه‌ای (NoC) که در حال حاضر مورد استفاده قرار می‌گیرند برای شبکه‌های کوچک مناسب هستند اما برای شبکه‌های بزرگ مناسب نیستند. با توجه به مسیرهای طولانی‌تر مورد نیاز برای رسیدن به مقصد، بسته‌های ارسال شده در یک NoC بزرگ منجر به معیارهای عملکرد بالاتری مانند تأخیر و مصرف انرژی می‌شوند. بنابراین، توسعه‌ی یک توپولوژی جدید مناسب برای NoC‌های با اندازه‌ی بزرگ مورد نیاز است. در این مطالعه، ما یک طراحی شبکه‌ی مقرون به صرفه را پیشنهاد کردیم که عملکرد کلی تأخیر را برای NoC‌های مقیاس بزرگ افزایش می‌دهد. توپولوژی RaMesh از خوشه‌های شبکه‌ی مش تشکیل شده است. علاوه بر این، یک تکنیک مسیریابی مناسب برای این معماری پیشنهاد شد. Altera ModelSim برای شبیه‌سازی معماری RaMesh برای مدل‌های سخت‌افزار Verilog، همراه با مش، چنبره و مش دوبعدی خوشه‌ای استفاده شد. سناریوهای مختلف ترافیک شبکه و اندازه‌ی شبکه در شبیه‌سازی‌ها استفاده شد. با توجه به داده‌های تجربی، RaMesh بهتر از توپولوژی‌های چنبره، مش دو بعدی خوشه‌ای، و توپولوژی‌های مش دوبعدی مشابه عمل کرد. علاوه بر این، توپولوژی RaMesh با سایر توپولوژی‌های مش خوشه‌بندی شده برای معیار مقایسه شد. در مقایسه با مش، چنبره و مش دوبعدی خوشه‌ای، توپولوژی پیشنهادی دارای تعداد پرش متوسطی بود که حداقل ۳۱ درصد کمتر بود. علاوه بر این، میانگین تأخیر حداقل ۲۵ درصد در مقایسه با مش، چنبره و مش دو بعدی خوشه‌ای کاهش یافت.