An Analytical Model of Many-Core System Using N-Conjugate Shuffle Cluster (NCSC)

Mahmoud Obaid
Department of Computer System Engineering, Arab American University
Palestine
Mahmoud.obaid@aaup.edu

Allam Abumwais
Department of Computer System Engineering, Arab American University
Palestine
Allam.Abumwais@aaup.edu

Suhail Odeh
Department of Software Engineering
Bethlehem University
Palestine
sodeh@bethlehem.edu

Mahmoud Aldababsa
Department of Electrical and Electronics Engineering
Nisantasi University, Turkey
mahmoud.aldababsa@nisantasi.edu.tr

Rami Hodrob
Department of Computer System Engineering
Arab American University, Palestine
rami.hodrob@aaup.edu

Abstract: *The previous study introduced the N-Conjugate Shuffle Cluster (NCSC), a router-less Interconnection Network (IN) designed to link multiple multi-core clusters, enabling the construction of a scalable many-core system. To ensure efficient intra-cluster communication, the system utilizes a Multi-Port Content-Addressable Memory (MPCAM) architecture. This paper presents a mathematical analysis of NCSC, demonstrating a linear relationship between bandwidth and the number of cores. Numerical results show that NCSC achieves a bandwidth of up to 2n Probability (PR) per cluster, where PR=r+p-rp, outperforming Grid crossbar and multi-cluster crossbar networks by 20-35% across various core counts (e.g., N<1024N). Static performance metrics further highlight NCSC's advantages: it maintains a constant diameter of 2, a degree of 4, and a bisection width of $K^2/2$, ensuring low latency, high scalability, and strong reliability. Comparative analysis with mesh, hypercube, and tree topologies confirms NCSC's superior scalability and cost-effectiveness, particularly as a router-less solution.*

Keywords: *Many-core, interconnection network, n-conjugate shuffle, content addressable memory, scalability, reliability.*

Received April 7, 2025; accepted September 10, 2025
https://doi.org/10.34028/iajit/23/1/1

1. Introduction

In all Interconnection Networks (INs), the primary concern has always been to improve scalability and performance. Performance is typically measured in terms of latency, probability of acceptance, and bandwidth. A design is said to be more scalable if it allows the system to be expanded in size and thus achieve higher performance. A system is scalable if it can grow and achieve higher performance without having to re-design its functional elements. The interconnection network is at the heart of multi-core and many-core systems, as it enables fundamental access to all shared resources, primarily shared memory. In these systems, congestion between cores and competition for access to shared memory are the most pressing issues for most interconnection net-works. These issues can lead to decreased performance of the execution program. In previous papers, the authors have presented a new many-core architecture that ad-dresses these issues. This architecture is based on a novel interconnection network called the N-Conjugate Shuffle Cluster (NCSC). NCSC is a router-less network that achieves high bandwidth and scalability by using a recursive partitioning of the network into smaller clusters. NCSC also uses a Multi-Port Content Addressable Memory (MPCAM) organization to improve communication within a cluster. As they shown that NCSC can significantly improve the performance of many-core systems by reducing congestion and competition for shared memory. NCSC has also been shown to be scalable to large systems with thousands of cores [2, 3, 4, 7].

In many-core interconnection architectures, whether each cluster is managed by a dedicated server or the cores are allowed direct access to the inter-cluster communication medium, the available options are largely limited to conventional topologies. In the first approach, cores compete to access the server through a local network topology, while the servers of different clusters compete for the global network topology [8, 16, 17]. In the second approach, all cores in the system compete directly for access to the global network topology, as seen in most conventional topologies such as tree, ring, mesh, torus, mesh-of-trees, and hypercube [1, 5, 6, 10, 13]. In both approaches, the ultimate destination is typically the shared memory of other cores. Clearly, both approaches complicate communication within the system and increase the number of router devices. Since this work proposes a hybrid IN that combines these two approaches, it is useful to briefly compare some of the topologies that have been implemented to enhance performance and scalability.

In the previous work [7], the NCSC-a hybrid architecture combining both approaches-was introduced. NCSC connects N multi-core clusters, each containing n cores. In this study, we assume that the number of clusters is equal to the number of cores in each cluster, resulting in a total of N^2 cores. This architecture consists of two main components:

1. A MPCAM organization for efficient intra-cluster communication.
2. A novel conjugate shuffle interconnection for inter-cluster communication.

To evaluate the performance of the NCSC topology, two approaches are employed:

- Analytical modeling, used to evaluate static performance metrics (bandwidth, delay, size, diameter, degree, connectivity, cost, and bisection width) through mathematical formulations.
- Implementation in a real many-core system, to validate the results.

This paper presents preliminary simulation results for static performance metrics derived from the mathematical model.

The remainder of this paper is organized as follows. Section 2 briefly lists some of the related work in this area. Section 3, explains the mathematical analysis of the main components of the NCSC. A comparative study with other famous INs is conducted in section 5. Section 6, explain simulation study and discussion. Section 7 give us use case and application scenario. Section 8 briefly explain the hardware feasibility and cost analysis. Finally, section 9 draws the paper's conclusion.

2. Related Work

First, due to the issues with router-based Network on a Chip (NoC), researchers first make a valiant effort to cut back on routers and later reach router-less NoC. Awal *et al*. [6] suggested a combination of 2-D mesh on several layers of NoC based on multilayer chips. The main goal of this architecture is to decrease the number of routers required in the network. This 2-D mesh network was compared various performance metrics, such as network complicity, cost, diameter, and network degree to other INs mathematically. It has an average access latency, fault-tolerant and complexity. Even though this IN offers a number of desirable characteristics, scaling the network is difficult and is therefore not possible with current technology. Furthermore, communication between the cores and shared memory is not considered in this architecture.

In addition to earlier works, recent advances (2020-2024) in router-less and hybrid NoC architectures have demonstrated improvements in scalability and power efficiency. These include hierarchical NoCs combining router-less intra-cluster designs with router-based inter-cluster communication, as well as novel approaches integrating multi-port memories and reconfigurable interconnect fabrics. These advancements reinforce the motivation for NCSC and provide further context to its design principles.

SecureNoC is a learning-based framework that improves NoC security against Hardware Trojan (HT) attacks while enhancing performance and power efficiency. It uses neural network-based HT detection, multi-function bypass channels to isolate threats, and adaptive lightweight encryption. A deep-Q-learning controller optimizes security, latency, and energy. Simulations show SecureNoC outperforms existing methods in detection accuracy and reduces latency and power consumption significantly [19].

Many-core systems require efficient interconnection networks to reduce power, area overhead, and high access latency caused by simultaneous core accesses. The NCSC architecture addresses these challenges by eliminating routers, distributing shared caches, and using MPCAMs, achieving low intra- and inter-cluster access latencies [7].

Machine learning has become a key tool in computer architecture, enabling scalable design, control, and simulation that surpass traditional methods. This research reviews current applications, introduces a deep reinforcement learning framework for complex design spaces like routerless NoCs, and develops machine-learning-based dynamic resource allocation frameworks for improved workload management in cloud environments [15].

Machine learning offers promising solutions for complex design space exploration in domain-specific architectures but faces challenges in selecting suitable algorithms and fairly comparing them. To address this, ArchGym, an open-source framework, connects diverse search algorithms with architecture simulators, enabling extensive evaluation across various architectures and workloads. ArchGym facilitates efficient research by simplifying data collection and supports fast, accurate proxy modeling, significantly reducing simulation time [11].

Wang *et al*. [18] presented a mathematical model using Semi-Markov Process (SMP) for modified NoCs to study the performance against other INs. It is the first work that applies the Semi-Markov process to analyze the performance communication analytical model of NoC. The experimental results show that the SMP model can be used to obtain NoC performance and it performs better than the state-of-art models.

The Chained-Cubic Tree (CCT) IN was proposed by Abdullah *et al*. [1] and a comparison of CCT and other contemporary network topologies was then conducted based on the mathematical probabilities model. CCT tries to solve hypercubes, mash and tree network drawbacks. It enhances some performance metrics such as diameter, degree, bandwidth and connectivity. However, it has its own drawbacks such as the high-cost constraints to implement it, undetermined access latency, and also doesn't discuss the shared memory bottleneck issue between processors.

A multi-cluster connectivity network named Nesting Ring NoC (NRO) was suggested by Li *et al*. [12]. NRO consists of a number of clusters, each with a fixed size of four to eight cores. NRO provides desirable performance and latency characteristics, but it has a clear scalability issue, especially as the number of cores approaches or exceeds hundreds. This will expand the network diameter, resulting in longer delays and more network traffic on inter-cluster routes.

Hamid *et al.* [9] proposed a new Multi-Core Multi-Cluster Architecture (MCMCA) network. MCMCA consists of clusters each with a set of nodes. Each node has multiple processors containing multiple cores. The communication process in MCMCA was divided into two types. First, the internal cluster communication consists of three kinds the Intra-Chip communication network (AC); the Inter-Chip communication network (EC) and the Intra-Cluster Network (ACN). The second is the Inter-Cluster Network (ECN) and the Multi-Cluster Network (MCN). The main goal of this architecture is to avoid congestion in the IN to guarantee faster communication. Hamid *et al*. [8] presented a mathematical model for MCMCA IN. The analytical model has been established with various numbers of processors. The analytical result shows that as the number of cores rises, the IN's performance can be improved and achieved lower access latency.

The IN in a many-core architecture is one of the study topics addressed by this body of literature. The main goal of this article is to present a mathematical model for the previously suggested effective IN, which is described in a scalable interconnection scheme in many-core systems [7].

3. The Mathematical Analysis of the Main Components of the NCSC

Multi-core and many-core INs are inherently well-suited for mathematical modeling and analysis, as opposed to direct parallel computing evaluation. This is primarily due to the deterministic nature of request handling, where a request can be issued, arbitrated, and either granted or denied within a single operational cycle. Mathematical analysis of such INs offers valuable insights into the system's structural properties and performance behavior, serving as a reliable indicator of scalability, latency, and bandwidth characteristics. In this context, the NCSC architecture is subjected to a comprehensive mathematical analysis to evaluate its static performance metrics. To facilitate this evaluation, the following assumptions are established:

1. (n) is the number of processors or cores.
2. Each cluster is built using nxn DPCAM modules, each of which can connect (n) OF core to (n) SB destinations.
3. The number of system clusters utilized to implement the system is (K), with k having a range of up to n+1.
4. (N) is the total number of cores in the system, with K*n.
5. (r) represents the probability that a core will send a request to the DPCAM network at some point in a cycle, i.e., the probability to access the internal cluster.
6. (p) denotes the probability that it received a request from another cluster in a previous cycle, i.e., the probability to access inter-cluster.

3.1. The MPCAM Organization

The MPCAM IN and the MPCAM-based multi-core system were proposed in previous works [2, 3]. As shown in Figure 1, the MPCAM is organized as a 2-D array of DPCAMs distributed across the IN. The most crucial setting in any multicore system network (measured as the total number of requests handled by the system) is Band-Width (BW). These networks' mathematical models are derived from theories of probability.

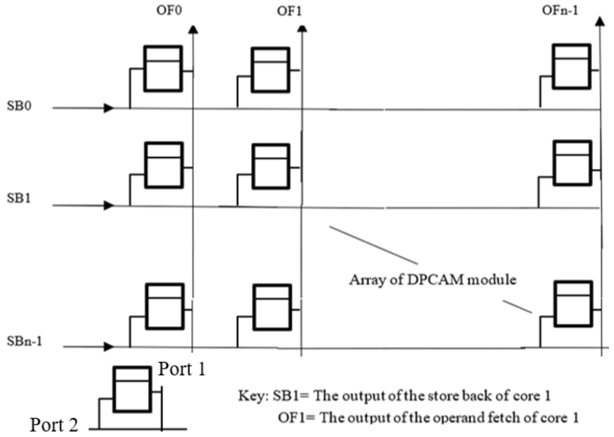

OF0
OF1
OFn-1
SB0
SB1
SBn-1
Array of DPCAM module
Port 1
Port 2
Key: SB1= The output of the store back of core 1
OF1= The output of the operand fetch of core 1

Figure 1. The internal cluster MPCAM network.

In MPCAM, n cores SB units are connected to n row buses, whereas n cores OF units of the same n cores are connected to the column buses. All *SB*s and *OF*s in the *MPCAM* organization can read and write at the same time. Therefore, 2*n* cores are expected to present requests to the *MPCAM* IN, and 2*nr* requests are expected during the cycle.

Therefore, the *BW* of *MPCAM* equals BW_{SB}=2*n* plus BW_{OF}=2*n*. Equation (1) provides the MPCAM's BW information:

$$BWmpcam = BW_{SB} + BW_{OF} = 2(nr) \tag{1}$$

In MPCAM, we are concerned with the probability that more than one request is made for a memory module as, in such cases; the multiple requests can be serviced. Let us assume that all processor makes a request for some memory module during each bus cycle. Taking a small numerical example, with two processors and four memories. Table 1 lists the possible requests. The average bandwidth is given by the average number of requests that can be accepted. Thirty-two requests can be accepted and the average bandwidth is given as 32/16=2.

Table 1. List the possible requests with two processors and four memories.

Memory requests processors		Number of requests accepted	Memory contention
P1	P2		
1	1	2	No
1	2	2	No
1	3	2	No
1	4	2	No
2	1	2	No
2	2	2	No
2	3	2	No
2	4	2	No
3	1	2	No
3	2	2	No
3	3	2	No
3	4	2	No
4	1	2	No
4	2	2	No
4	3	2	No
4	4	2	No

3.2. The NCSC Organization Band-Width Analysis

The NCSC is a scalable many-core IN was proposed by Eleyat and Abumwais [7]. As shown in Figure 2 the NCSC connects the MPCAM clusters using N-conjugate shuffle combinations to connect between cores of the K system clusters.

According to the above presumptions, each cluster bandwidth in the i^{th} cycle is determined by Equation (1). Assuming we have k clusters, then it is expected that $\frac{1}{K}$*BWmpcam*(n, n) requests be accepted for intra-cluster, and $\frac{K-1}{K}$*BWmpcam*(n, n) requests are accepted for inter-cluster via NCSC.

In the (j+1) clock cycle, the probability that requests from other clusters have reached is p, and the probability for intra-cluster requests is r. So, probability that a request happened from any core in the system is then determined by the:

$$PR = r + p - rp \tag{2}$$

So, the total *BW* for each cluster is determined by the:

$$BW2(n, n) = 2n(PR) \tag{3}$$

So, the total *BW* in both cycle 1 and cycle (j+1) is:

$$BW1{,}2(n, n) = \frac{1}{K}\,[2nr] + [2n(PR)] \tag{4}$$

Substitute the value of *PR* from Equation (2), and the p with $\frac{K-1}{Kn}(2nr)$ i.e.,

$$p = \frac{\mathrm{K}-1}{K}(2r) \tag{5}$$

The total number of cores in the NCSC system equal N, were N=nk. In this work, we assume that n=K, i.e., N=n^2 cores are included in the many-core system. Therefore, multiplying Equation (4) by K gives the *BW* of all clusters for the both cycles:

$$BW1{,}2(n, n, k) = 2\mathrm{k}\,[r + nr + np - nrp] \tag{6}$$

On the other hand, to determine the average *BW* of all clusters per cycle is given by:

$$BWavg = \frac{BW1{,}2(n, n, k)}{2} \tag{7}$$

Substitute the value of p from Equation (5). The $BWavg$ can be reduced to:

$$BW_{NCSC} = \mathrm{k}\,[r + nr + np - nrp] \tag{8}$$

In order to make a comparison, the BW of Grid crossbar many cores [14] and many-core depends on the multi-cluster-crossbar network are listed in the following equations:

The BW of the *Grid* crossbar is given by:

$$BW_{Grid} = N - N\left(1 - \frac{r}{N}\right)^{N}] \tag{9}$$

The BW of the multi-cluster-crossbar network is given by:

$$BW_{multi-cluster-crossbar} = \frac{n}{2}\left[1 + \left(\frac{p}{k-1}\right) - \left(1 - \frac{(r + p - rp)}{n}\right)^{n}\right] \tag{10}$$

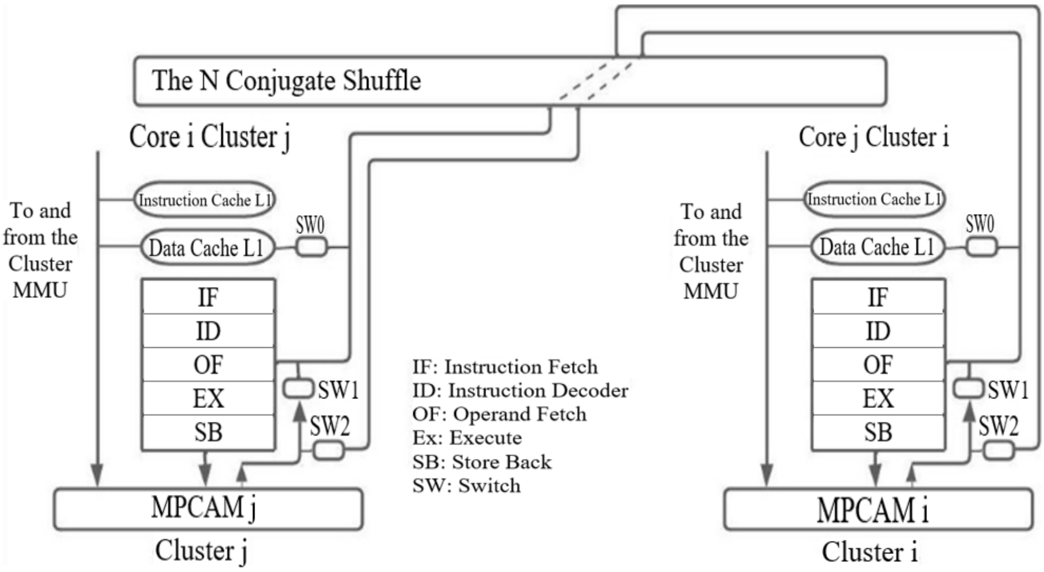


Figure 2. NCSC IN.

4. Comparison Results of Bandwidth Mathematical Analysis

The bandwidth of NCSC, NXN Grid crossbar and multi-cluster crossbar are obtained from the equations above over various values of *K* and *r*. The results are dis-played in Figures 3 to 6.

Figure 3 shows the bandwidth of NCSC as compared to Grid crossbar and multi-cluster crossbar networks. The figure shows that NCSC has a higher bandwidth than other networks for K<32 with r=1 with N<1024. It also has a better bandwidth for K>32 with r=1than other networks as shown in Figure 4. Grid crossbar and multi-cluster crossbar are based-router INs that use 2×2 or more switches for a block, it can be said that NCSC betters them in having higher bandwidth, being it is a routerless network and easier to scale. Therefore, this demonstrated why the NCSC is the most cost-effective, as will be discussed in the section 5.1.

The bandwidth of NCSC is compared to the bandwidth of other networks with K>32 and r=0.5 in Figures 5 and 6. These figures demonstrate that NCSC's bandwidth position with respect to other networks will not be affected by reducing r or increasing K.

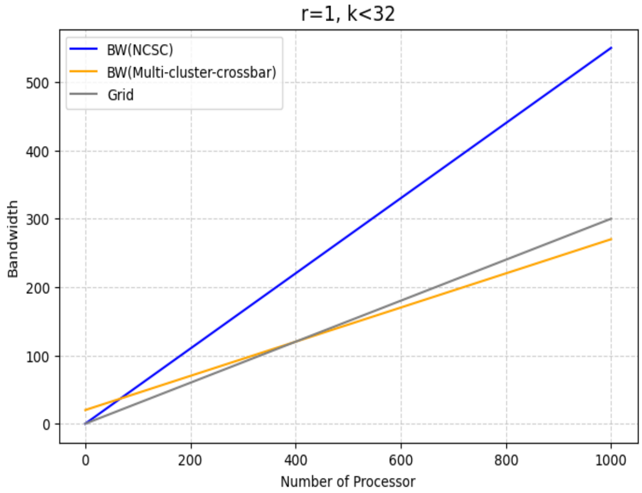


Figure 3. BW of NCSC compared to other networks at K<32 and r=1.

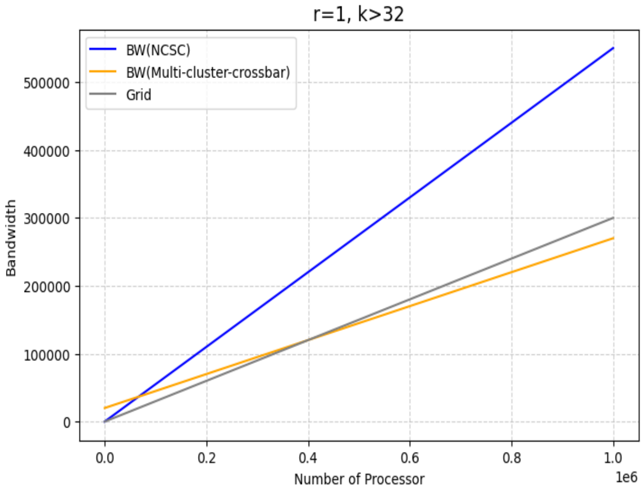


Figure 4. BW of NCSC compared to other networks at K>32 and r=1.

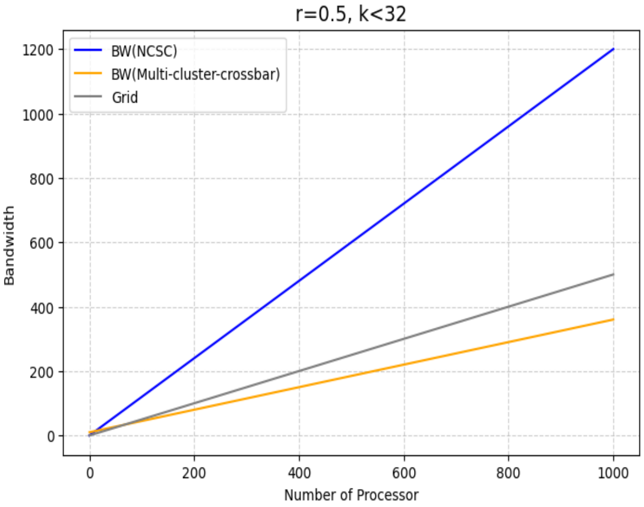


Figure 5. BW of NCSC compared to other networks at K<32 and r=0.5.

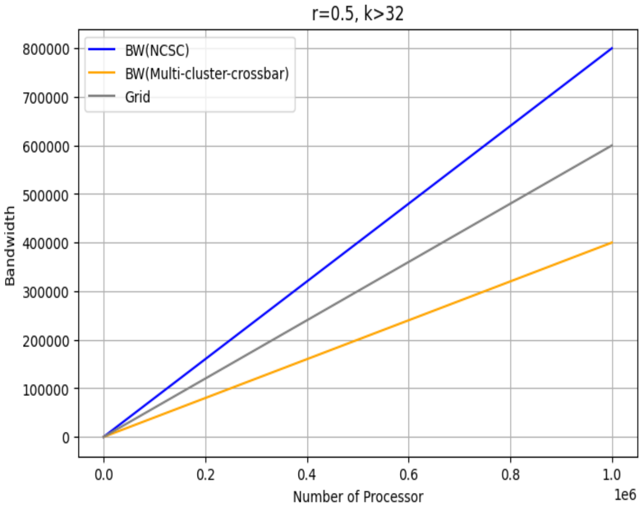


Figure 6. BW of NCSC compared to other networks at K>32 and r=0.5.

5. Performance Analysis of NCSC

The static performance metrics are the most popular way to evaluate the effectiveness of any IN such as (size, diameter, degree, connectivity, cost and bisection width) by deriving a mathematical equation for each metric. For comparison purposes, the most well-known network topologies, such as mesh, ring, tree, mesh of tree, hypercube, and any other customized topology, should have known static network performance metrics. This is crucial since the dynamic metrics of any IN before deployment are typically reflected by these metrics. For instance, a small diameter reduces network latency and power consumption. Therefore, a smaller diameter is appropriate. Several researchers conducted a comparison study to assess the static performance of various topologies in recent years [1, 5, 6, 9, 12, 13].

5.1. Static Performance Analysis

The NCSC topology's static performance metrics have been analyzed and compared with some of the topologies most frequently used in many-core systems, including mesh, ring, tree, mesh of tree and hypercube. To analyze the static performance of any network topology, various metrics are employed. The NCSC (*K*, *n*) IN is studied using the following metrics. For all metrics analyses, consider *n*=*K*.

1. Size: is the total number of nodes (cores) in the network; a larger number can make the network more scalable. Equation (11) displays the size of the NCSC.

$$Size(NCSC(K,n)) = K^2 \quad (11)$$

The scale of this topology can be expressed in terms of the number of cores (*n*) in each cluster multiplied by the number of clusters (*K*) in a system.

2. Diameter: is a key parameter that influences the IN, it refers to the maximum distance between any two cores through the network. Better latency and power consumption for the IN are associated with a small and constant diameter Awal *et al*. [6]. Hence, a smaller diameter is desirable. Equation (12) displays the diameter of the NCSC.

$$Diameter(NCSC\ (K,n)) = 2 \quad (12)$$

Regardless of system size. The NCSC diameter is calculated by adding the diameter of MPCAM within the cluster to the inter-cluster diameter. In MPCAM, each core is directly connected to every other core through a broadcast bus which means that the diameter equals 1. Furthermore, because the conjugate shuffle interconnection is connected by a bidirectional bus, the diameter between the clusters is equal to 1. The diameter is 2 as a result.

3. Degree: this parameter determines the largest number of links in a topology that are directly connected to any core. A constant node degree is preferable for INs. It is easy to scale a network with a constant degree. Equation (13) illustrates degree of the NCSC.

$$Degree(NCSC\ (K,n)) = 4 \quad (13)$$

Each core is connected to other cores within the same cluster using two links on OF and SB; on the other hand, the conjugate core is connected to other clusters using two links. Therefore, the maximum number of links connected to any core is 4 for conjugate cores.

4. Connectivity: is a metric for counting the number of paths that lead from one core to another. For the NCSC, connectivity is represented by the following equation:

$$Connectivity(NCSC\ (K,n)) = 2 \quad (14)$$

There are two paths between any two cores within the same cluster or across different clusters.

5. Cost: though hardware devices determine the total cost of the *IN* that contains the number of routers and links. For comparative purposes, cost mostly refers to the total number of links needed to build the network.

   The cost of NCSC is shown in the following equation:

$$Cost(NCSC\ (K,n)) = 3K^2 - K \quad (15)$$

The cost depends on the number of cores in each cluster (*n*) and the number of clusters in the system (*K*). The number of links in each cluster=2*n* and so, the number of total links within all clusters=2*nK*.

Number of total links between different clusters depends on *n* and *K* and is equal to the conjugate shuffle interconnection. Assuming that *n*=*K*, series analysis for *n*= (2, 3, ⋯, ∞) and referring to Figure 2 connection, inter-cluster links is equal to *K*(*K*-1). Therefore, the total links in the scheme are equal to $2k^2$+*K*(*K*-1)=$3K^2$-*K*.

6. Bisection width: the term “bisection width” refers to the smallest number of links that had to be changed in order to divide the entire network into two equal halves. Low bandwidth between two portions is produced by small bisection width. A very large bisection width, on the other hand, requires a large number of wires for designing. As a result, large bisection width is preferred for all INs. The bisection width of the proposed system is shown by the following equation:

$$Bisection(NCSC\ (K,n)) = \frac{K^2}{2} \quad (16)$$

Dividing this scheme into two equal parts means having an equal number of clusters in each part; on the other hand, we should maintain the connection links in each part to be an independent part. It can be noticed that the bisection width can be calculated as the total number of links between different clusters is *K(K-1)*, calculated in point 5, subtracted from them the number of links of the two new parts which is equal $(\frac{K}{2}\left(\frac{K}{2}-1\right)$. For example, in a system with 10 clusters, the result after dividing the system into two parts is five clusters in each part. Therefore, by logic the bisection width is equal to the number of links in *N*=10 minus the links of two new parts each with *K*=5. So, Bisection width of *NCSC* (*K*, *n*)=$K(K\text{-}1)\text{-}2(\frac{K}{2}(\frac{K}{2}\text{-}1))=\frac{K^2}{2}$.

5.2. The Comparison Results and Discussion of Static Performance Metrics

In order to assess the effectiveness of the topological properties of NCSC, a comparison study between the suggested scheme NCSC (K, n) and other well-known topologies utilized in parallel systems is carried out in this section. In addition, research is also conducted to determine whether the intended improvements are achieved.

We chose the following topologies: a 2-D Mesh M(n, n), a Tree T(h) with high h, a Mesh of Trees (Mot) (n, h) of n by n mesh and a tree of height h, and a hypercube Q(d) of dimension d. These topologies have been chosen because of their good properties and because they are used in most of many-core and parallel systems. Table 2 displays the topological properties of these networks. In order to best evaluate the NCSC (K, n), all of the mentioned properties of the proposed topology need to be computed with respect to different systems sizes and then compared with others topologies. This allows determining NCSC location relative to other INs. After that, we can judge if NCSC has some improvements over other networks. Figures 7 to 11 show the comparative results.

Table 2. Properties of some INs.

Parameter/Properties					
Topology	M(n,n)	T(h)	Mot(n,n)	Q(d)	NCSC(N,n)
Size	n^2	2^{h+1}-1	$3n^2$-2n	2^d	n^2
Diameter	2n-2	2h	4log n	D	2
Degree	4	3	3	D	4
Connectivity	2	1	2	d	4
Cost	$2n^2$-2n	2^{h+1}-2	$4n^2$-4n	$d2^{d-1}$	$3n^2$-n
Bisection width	n	1	n	2^{d-1}	$\frac{n^2}{2}$

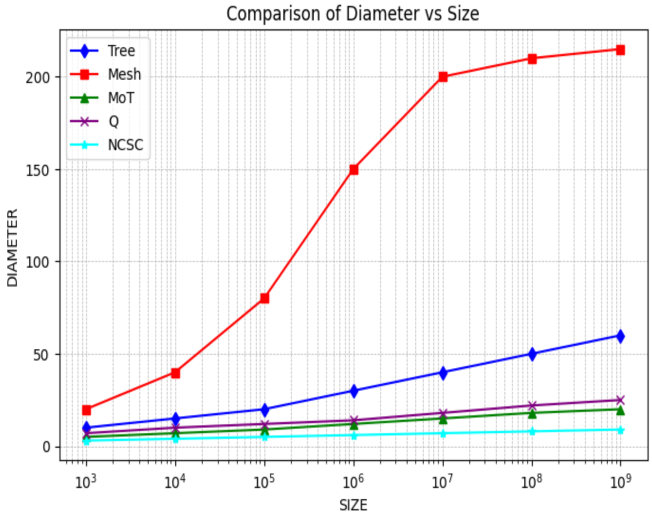


Figure 7. Diameter of the networks for different sizes.

Figure 7 depicts the diameter of the five studied topologies as their sizes increase. It is clear that NCSC has a constant diameter that is independent of its size and so it scores best while hypercube topology has the second rank and mesh topology is the worst in this regard. Therefore, the NCSC has the highest speed due to its small diameter. Because the latency and power consumption of an IN is influenced by a number of factors, among them diameter. As a result, a network with a reduced diameter has the properties of a power-efficient network. Figure 8 depicts a comparison of the degree parameter between the five topologies. NCSC, mesh, tree and Mot topologies have a contiguous and constant degree. This means that they have the best capability for scaling the system to any number of cores without changing the old cores. In addition, there is no need in NCSC to rebuild the clusters or change the algorithms used by the compiler to schedule the tasks on the system. On the other hand, the hypercube topology has the worst degree which increases the cost and complexity when scaling up the system.

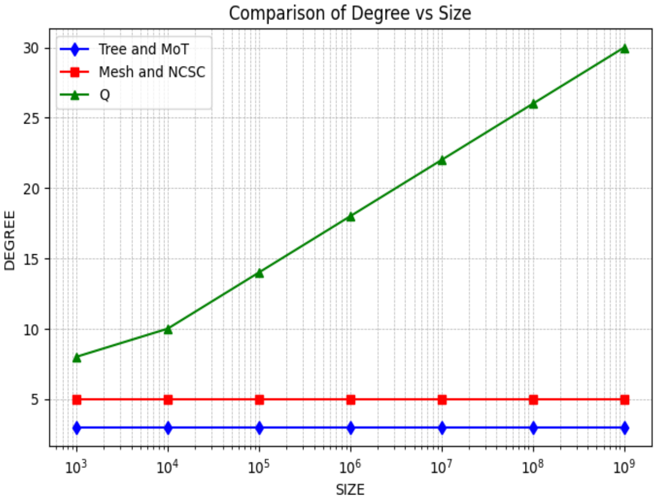


Figure 8. Degree of the networks for different sizes.

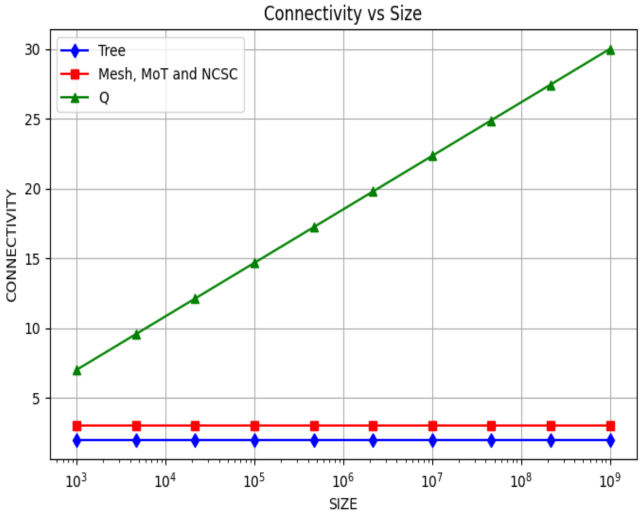
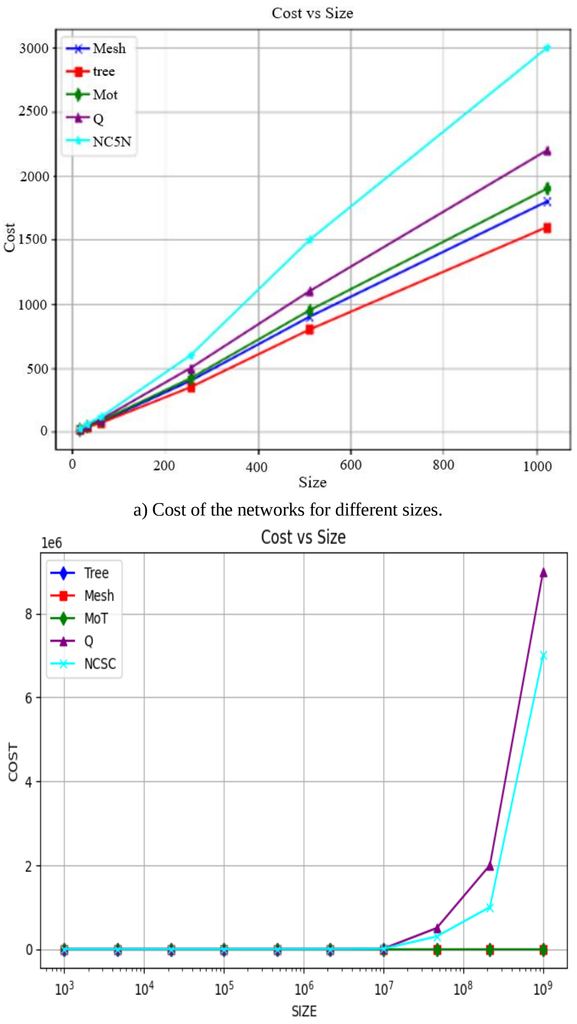


Figure 9. Connectivity of the networks for different sizes.

Figure 9 shows a comparison of the connectivity parameter between several INs. It shows that the NCSC has the same connectivity as mesh and Mot networks with constant connectivity of 2 between any two cores. Whereas, the hypercube has the highest connectivity which increases the cost as will be explained later. On the other hand, as the connectivity increases, the robustness of the network increases also. Both Figure 10-a) and (b) show the cost of the five studied topologies; however, Figure 10-a) shows the cost when using a smaller number of cores. The hypercube has the highest cost while the tree topology has the cheapest cost relative to other topologies. In addition, the NCSC can be considered to have a mild cost that is often close to that of mesh. However, this parameter does not reflect the total cost in many-core systems because it depends on cost related only to the number of nodes and links without considering other factors such as power and area costs, which are very important issues to be considered. Anyway, nodes and links can be useful in another parallel system where the main goal is to produce high-performance systems bargained with cost. Figure 11 displays the bisection widths of the above topologies. Both the NCSC and hypercube topologies have the highest bisection width. On the other hand, Tree has the lowest bisection width that equals to one in all situations. In addition, it can be noted that Mesh and Mot have low values of bisection widths. The bisection width is quite important in INs because the reliability of a topology increases with it. Therefore, NCSC and hypercube have the best reliability.



a) Cost of the networks for different sizes.

b) Cost of the networks for different sizes.

Figure 10. Network cost as a function of network size.

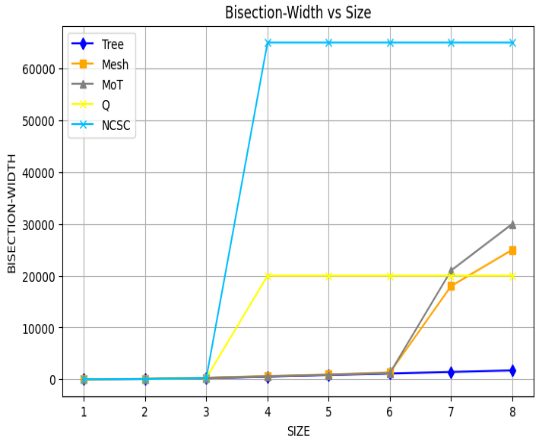


Figure 11. Bisection-width of the networks for different sizes.

As a result, it is obvious that NCSC has encouraging characteristics when compared with other well-known topologies, except for the links cost, which has a moderate cost, as we have mentioned earlier. Moreover, the cost of NoCs is mostly dependent on other parameters like the area and power consumption, which are related to the complex structure of network and usage of routers [10]. In contrast, the NCSC has no routers and arbiters needed in other topologies with free blocking or contention. So, it can be expected that the cost will be reduced if these parameters are taken into account.

6. Simulation Study Discussion

NCSC and MPCAM have been successfully implemented, compiled, and verified within a many-core system utilizing Quartus Prime 20.1, which encompasses the Intel-supported ModelSim package and the Nios II Embedded Design Suite (EDS) for design and simulation. The system was developed using Verilog Hardware Description Language (Verilog HDL) code in the Cyclone IV-E Field Programmable Gate Array (FPGA) device family.

The timing analyzer tool is utilized to assess the read/write latency for the MPCAM and NCSC.

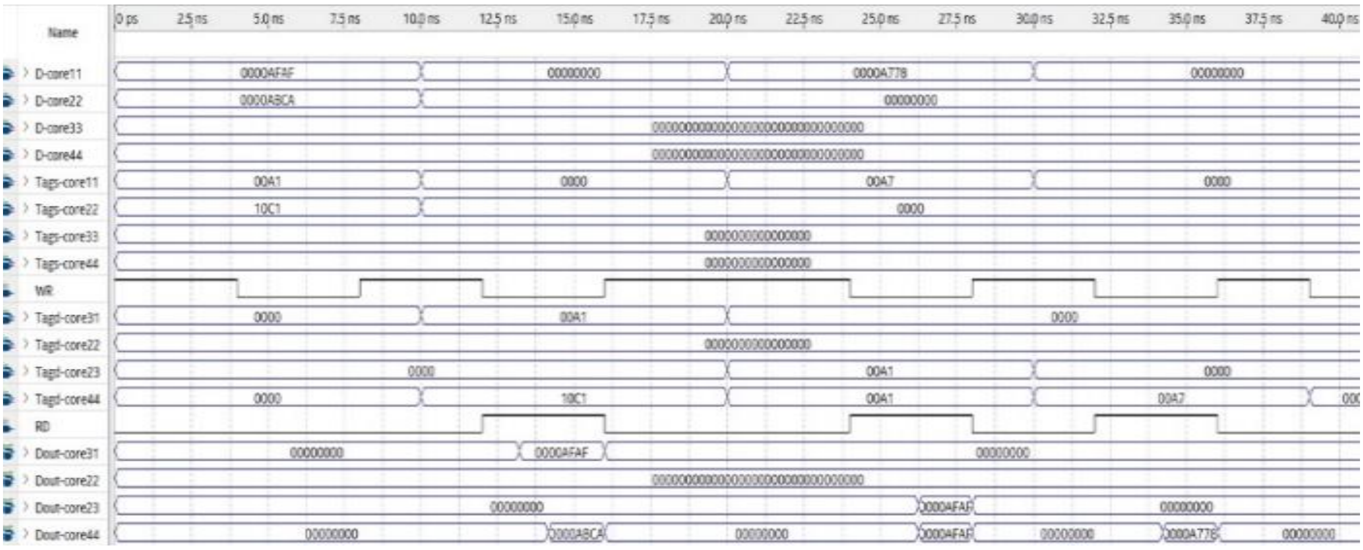


Figure 12. NCSC organization timing simulation for a read operation.

Figure 12 shows the timing simulation of the proposed NCSC with several intervals. It was noticed that the delay for read access between cores in different clusters is around 1.92738±0.139588 ns, which is nearly equal to the latency of read operation in (Dout-core 44). The third Interval (30 to 40 ns) shows the same behavior as the second interval l (20 to 30 ns).

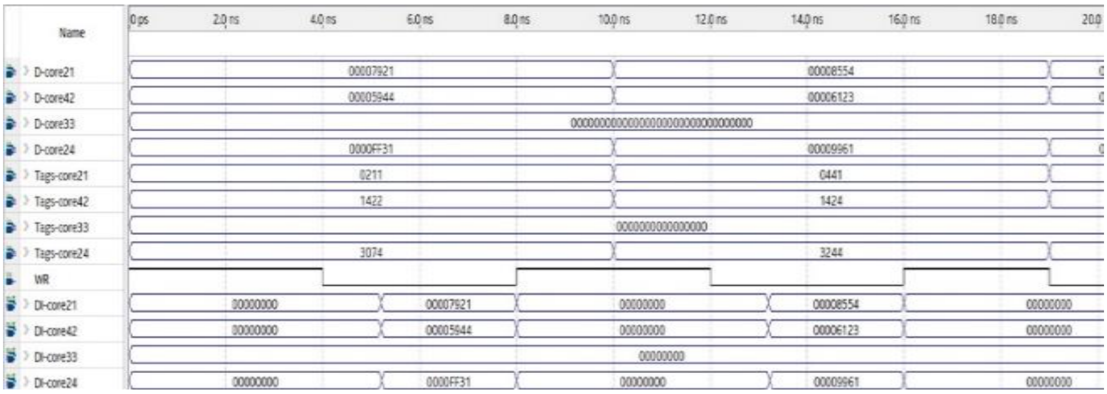


Figure 13. NCSC organization timing simulation for a write operation.

Figure 13 shows an image of two intervals to assess the write latency over NCSC. In the first interval (0 to 10 ns), core 21, core 42, and core 24 write their shared data with their tags (0211), (1422), and (3074) respectively, each to its MPCAM. It can be observed that the written data are stored in DI-core21, DI-core42, and DI-core24 pins after a delay time with an average of 1.14785±0.04532 which is nearly equal to the latency of the write operation in the MPCAM organization. In the second interval (10 to 20 ns), core21, core42, and core 24 write the shared data to their MPCAM simultaneously with an average delay of 1.15235±0.06132 which is identical to the latency in the previous interval.

7. Use Case and Application Scenario

The NCSC topology is designed to support scalable many-core systems with low latency and high bandwidth communication. This makes it particularly suitable for integration into modern System-on-Chip (SoC) designs aimed at High-Performance Computing (HPC), data center processors, and workloads that demand efficient intra- and inter-cluster communication. By enabling a router-less interconnection with predictable and low-diameter communication paths, NCSC can reduce contention and improve throughput for parallel applications requiring frequent data sharing.

In practical SoC implementations, NCSC can be deployed as the backbone interconnection network linking multiple multi-core clusters, each responsible for specific application domains or computation tasks. For instance, in HPC applications such as scientific simulations, machine learning, and real-time data analytics, the need for frequent data exchange between cores demands a network with low contention and predictable latency characteristics inherently provided by NCSC’s routerless, low-diameter design.

Moreover, the scalable nature of NCSC allows SoC designers to flexibly expand system size without redesigning the interconnect fabric or communication protocols. This enables rapid adaptation to evolving workload demands, such as increasing core counts for higher throughput or integrating heterogeneous accelerators within clusters.

NCSC is especially beneficial for workloads exhibiting a mix of intensive intra-cluster communication efficiently handled by the multi-port CAM organization and moderate inter-cluster communication, where the conjugate shuffle topology maintains low-latency links. Examples include graph processing, parallel database queries, and large-scale neural network inference.

Additionally, by eliminating routers and arbiter logic typical in conventional NoCs, NCSC reduces silicon area and power overhead, critical factors in energy-sensitive applications such as mobile and embedded systems. The deterministic communication paths also simplify real-time scheduling and guarantee Quality of Service (QoS) levels required in mission-critical applications.

In summary, the NCSC topology’s unique combination of scalability, bandwidth efficiency, and low power consumption makes it a promising candidate for future SoC designs targeting diverse domains ranging from HPC and AI accelerators to embedded real-time systems.

8. Hardware Feasibility and Cost Analysis

A critical aspect of the NCSC design is the use of MPCAMs to support efficient intra-cluster communication. While MPCAMs provide significant bandwidth advantages, their scalability in terms of silicon area and power consumption must be considered. Recent studies indicate that multi-port CAM implementations can be optimized to minimize area overhead and power usage through architectural enhancements and technology scaling. The estimated silicon area overhead for MPCAM-based clusters remains competitive compared to traditional router-based interconnects, particularly when accounting for the elimination of routing logic and arbiter circuitry. Power consumption is also expected to be lower in NCSC due to the reduction in active router components and reduced communication latency, leading to energy savings in practical implementations.

The MPCAM is a central component enabling the efficient intra-cluster communication in the NCSC topology. While MPCAMs deliver substantial improvements in bandwidth and contention-free access, their practical implementation raises concerns related to silicon area, power consumption, and circuit complexity, which must be carefully evaluated for large-scale integration.

Recent advancements in CAM design techniques, including circuit-level optimizations such as low-power sensing amplifiers, segmented search lines, and hierarchical CAM architectures, have significantly reduced both dynamic and leakage power. These improvements allow multi-port CAM arrays to achieve scalability while minimizing the energy per search operation.

From an area perspective, MPCAM modules inherently require more transistor count than conventional memory structures due to parallel search and comparison logic for each port. However, the absence of complex router logic and arbiters, which are typically required in traditional NoC designs, offsets this overhead. Specifically, eliminating routers reduces wiring congestion and routing resources, which often dominate interconnect area and power in many-core chips.

Silicon area estimations based on recent Complementary Metal-Oxide-Semiconductor (CMOS) technology nodes (e.g., 7nm and below) indicate that multi-port CAM arrays can be designed to fit within feasible footprints for clusters containing dozens of cores. Techniques such as port sharing, time-multiplexed access, and clock gating further optimize resource utilization without significantly degrading performance.

Power consumption analysis shows that the NCSC's architecture benefits from reduced active switching in routing logic and decreased communication latency, leading to lower energy per bit transmitted. Furthermore, the reduction in packet buffering and arbitration stages contributes to power efficiency gains, especially under moderate to high traffic conditions.

While exact power and area figures depend on the specific implementation details and process technology, preliminary synthesis and layout results from similar MPCAM-based systems confirm that the trade-offs are favorable. The NCSC's router-less approach coupled with optimized MPCAM design provides a cost-effective and energy-efficient alternative to conventional NoC solutions, especially for applications demanding high throughput and low latency.

In conclusion, the hardware feasibility of NCSC's MPCAM-based clusters is supported by emerging CAM design innovations and system-level power-area trade-offs. These factors, combined with the scalability and performance benefits of the NCSC topology, justify its potential for practical deployment in future many-core SoCs.

9. Conclusions

This in this work, authors have mathematically analyzed the behavior of NCSC in connecting multi-clusters of many-core systems. This proposed network is a type of new-era NoC called router-less IN.

In on-chip many-core systems, the mesh and MoT topologies are considered the best; however, they have some disadvantages like big diameter, low bisection width and high cost, which is mostly due to the usage of router-based structures. Therefore, many researchers try to find alternatives for these topologies based on router-less interconnection. NCSC maintains the linearity between the performance and core count over any number of clusters. This paper mathematically proved that NCSC has desirable properties in terms of bandwidth, diameter, degree, connectivity, cost, latency, scalability and reliability.

Finally, in addition to the features presented in this paper, we can claim that this system is scalable and can be expanded to have N clusters without the need to redesign the cluster or change the connectivity program and the compiler. All that is needed to change the value of the "number of clusters" in the program. For example, in the case of N=n=32 cores, a system of 2, 3, 4,..., N clusters can be produced as far as there is a silicone space on the chip. Furthermore, system latencies from 2 to N clusters, in both intra-cluster and inter-cluster, are constant and independent of the number of clusters because each cluster is connected only to its conjugate. This applies to any feasible value of N. However, the results can be extended to any homogeneous multi-core multi-cluster architecture.

References

[1] Abdullah M., Abuelrub E., and Mahafzah B., "The Chained-Cubic Tree Interconnection Network," *The International Arab Journal of Information Technology*, vol. 8, no. 3, pp. 334-343, 2011. https://C:/Users/acit2k/Downloads/The_chained-cubic_tree_interconnection_network%20(1).pdf

[2] Abumwais A. and Ayyad A., "The MPCAM Based Multi-Core Processor Architecture: A Contention Free Architecture," *WSEAS Transactions on Electronics*, vol. 9, pp. 105-111, 2018. https://wseas.com/journals/articles.php?id=2706

[3] Abumwais A. and Obaid M., "Shared Cache Based on Content Addressable Memory in a Multi-Core Architecture," *Computers, Materials and Continuavol*, vol. 74, pp. 4951-4963, 2023. https://doi.org/10.32604/cmc.2023.032822

[4] Abumwais A., Amirjanov A., Uyar K., and Eleyat M., "Dual-Port Content Addressable Memory for Cache Memory Applications," *Computers, Materials and Continua*, vol. 70, pp. 4583-4597, 2022. https://doi.org/10.32604/cmc.2022.020529

[5] Alam M. and Varshney A., "A Comparative Study of Interconnection Network," *International Journal of Computer Applications*, vol. 127, pp. 37-43, 2015. DOI: 10.5120/ijca2015906378

[6] Awal R., Rahman H., Nor R., Sembok T., and Akhand M., "Architecture and Network-on-Chip Implementation of a New Hierarchical Interconnection Network," *Journal of Circuits, Systems and Computers*, vol. 24, pp. 1540006, 2015. https://doi.org/10.1142/S021812661540006X

[7] Eleyat M. and Abumwais A., "A Scalable Interconnection Scheme in Many-Core Systems," *Computers, Materials and Continua*, vol. 77, pp. 615-632, 2023. https://doi.org/10.32604/cmc.2023.038810

[8] Hamid N., Walters R., and Wills G., "An Analytical Model of Multi-Core Multi-Cluster Architecture," *Open Journal of Cloud Computing*, vol. 2, pp. 4-15, 2023. https://www.ronpub.com/OJCC_2015v2i1n02_Hamid.pdf

[9] Hamid N., Walters R., and Wills G., "An Architecture for Measuring Network Performance in Multi-Core Multi-Cluster Architecture," *International Journal of Computer Theory and Engineering*, vol. 7, no. 1, pp. 57-61, 2015. https://www.ijcte.org/vol7/930-AC0023.pdf

[10] Hoskote Y., Vangal S., Singh A., Borkar N., and Borkar S., "A 5-GHz Mesh Interconnect for a Teraflops Processor," *IEEE Micro*, vol. 27, pp. 51-61, 2007. https://doi.org/10.1109/MM.2007.4378783

[11] Krishnan S., Yazdanbakhsh A., Prakash S., Jabbour J., and et al., "Archgym: An Open-Source Gymnasium for Machine Learning Assisted Architecture Design," *in Proceedings of the 50th Annual International Symposium on Computer Architecture*, Orlando, pp. 1-16, 2023. https://doi.org/10.1145/3579371.3589049

[12] Li W., Guo B., Li X., Yin S., and et al., "Nesting Ring Architecture of Multichip Optical Network on Chip for Many-Core Processor Systems," *Optical Engineering*, vol. 56, no. 3, pp. 1-10, 2017. https://doi.org/10.1117/1.OE.56.3.035106

[13] Ou Y., Agwa S., and Batten C., "Implementing Low-Diameter On-Chip Networks for Manycore Processors Using a Tiled Physical Design Methodology," *in Proceedings of the 14th IEEE/ACM International Symposium on Networks-on-Chip*, Hamburg, pp. 1-8, 2020. https://doi.org/10.1109/NOCS50636.2020.9241710

[14] Patterson D. and Hennessy J., *Computer Organization and Design ARM Edition: The Hardware Software Interface*, The Morgan Kaufmann Series in Computer Architecture and Design, 2016. http://home.ustc.edu.cn/~louwenqi/reference_books_tools/Computer%20Organization%20and%20Design%20ARM%20edition.pdf

[15] Penney D., Machine Learning for Computer Architecture Design and Optimization, Ph.D. Thesis, Oregon State University, 2023. https://ir.library.oregonstate.edu/concern/graduate_thesis_or_dissertations/70795h33q

[16] Ruaro M., Velloso N., Jantsch A., and Moraes F., "Distributed SDN Architecture for NoC-based Many-Core SoCs," *in Proceedings of the 13th IEEE/ACM International Symposium on Networks-on-Chip*, New York, pp, 1-8, 2019. https://doi.org/10.1145/3313231.3352361

[17] Udipi A., Muralimanohar N., and Balasubramonian R., "Towards Scalable, Energy-Efficient, Bus-based On-Chip Networks," *in Proceedings of the HPCA-16 the 6th International Symposium on High-Performance Computer Architecture*, Bangalore, pp. 1-12, 2010. https://doi.org/10.1109/HPCA.2010.5416639

[18] Wang J., Li Y., and Peng Q., "A Novel Analytical Model for Network-on-Chip Using Semi-Markov Process," *Advances in Electrical and Computer Engineering*, vol. 11, no. 1, pp. 111-118, 2011. https://www.aece.ro/abstractplus.php?year=2011&number=1&article=18

[19] Wang K., Zheng H., Li Y., and Louri A., "SecureNoC: A Learning-Enabled, High-Performance, Energy-Efficient, and Secure On-Chip Communication Framework Design," *IEEE Transactions on Sustainable Computing*, vol. 7, pp. 709-723, 2021. https://doi.org/10.1109/TSUSC.2021.3138279

Mahmoud Obaid is Dean of the Faculty of Engineering and an Assistant Professor in the Computer System Engineering Department at the Arab American University. He earned his B.Sc. (2009) and M.Sc. (2013) in Electronic and Computer Engineering from Al-Quds University, and his Ph.D. in Computer Engineering from Eastern Mediterranean University (Turkey) in 2019. His professional experience includes serving as a Research Assistant at Al-Quds University (2009-2013), Network Engineer at the Palestine Broadcasting Corporation (2010), and Instructor at Modern University College (2011-2016). He joined the Arab American University in 2017. His research interests include Computer Networks, Computer Architecture, Blockchain, Mobile Commerce and Payments (including security), Wireless Communication, and MIMO.

Allam Abumwais is an Assistant Professor in the Computer System Engineering Department at the Arab American University. He received his B.Sc. and M.Sc. degrees in Electronic and Computer Engineering from Al-Quds University in 2010 and 2014, respectively. He earned his Ph.D. in Computer Engineering from Near East University in Turkey in 2022. From 2010 to 2013, he worked as a Research Assistant in the Computer Engineering Department at Al-Quds University. After that, in 2014, he joined the Computer System Engineering Department at the Arab American University Palestine. His research interests include Computer Architecture and Organization, Parallel Systems, Computer Networks, and AI Applications.

Suhail Odeh is an Associate Professor in the Software Engineering Department at Bethlehem University, where he has taught and conducted research since 2006. A native of Bethlehem, he earned his B.Sc. in Physics and Electronic Technology (1996) and M.Sc. in Physics (2001) from Al-Quds University, followed by a Ph.D. in Computer Engineering from the University of Granada, Spain (2006). He completed a postdoc at the University of L’Aquila, Italy (2016), and held visiting positions at the Universities of Cyprus, Granada, and Salamanca. His research focuses on Artificial Intelligence, Pattern Recognition, Intelligent Systems, Brain-Computer Interfaces, and Multi-Agent Systems, with numerous publications in international conferences.

Mahmoud Aldababsa (Senior Member, IEEE) received the B.Sc. degree in electrical engineering from An-Najah National University, Palestine, in 2010, the M.Sc. degree in Electronics and Communication Engineering from Al-Quds University, Palestine, in 2013, and the Ph.D. degree in Electronics Engineering from Gebze Technical University, Turkey. He was a Research and Teaching Assistant at Al-Quds University from 2010 to 2013. He then held the position of Post-Doctoral Researcher at the Communications Research and Innovation Laboratory, Koc University, Turkey. Subsequently, he was an Assistant Professor in electrical and electronics engineering at Istanbul Gelisim University, Turkey. He is currently an Associate Professor in the Department of Electrical and Electronics Engineering at Nisantasi University, Istanbul, Turkey. His current research interests include Non-Orthogonal Multiple Access and Reconfigurable Intelligent Surfaces in 5G and 6G Wireless Systems.

Rami Hodrob is Dean of Admission and Registration and an Assistant Professor in the Computer System Engineering Department at the Arab American University (AAUP), Palestine. He has been a Lecturer in AAUP’s Faculty of Engineering and IT since 2003 and headed the Computer Technology Information Department during two terms (2014-2015 and 2018-2019). He holds a Ph.D. from the Czech University of Life Sciences (2016), an M.Sc. in Computing from Birzeit University (focusing on graphic notations in ontology engineering), an MBA from An-Najah University, and a B.Sc. in Electronic Engineering from Yarmouk University (1995). His research interests include Knowledge Engineering, Ontologies, Digital Economics, the Semantic Web, and Augmented Reality. He also contributed to the Erasmus+CBHE project TESLA.