A Review of Optimization Techniques in Network-on-Chip (NoC) Architecture

M. Sudais Khan
Electronics Engineering Department
University of Engineering and Technology
Taxila, Pakistan
22-ms-enc-05@students.uettaxila.edu.pk

Yaseer A. Durrani
Electronics Engineering Department
University of Engineering and Technology
Taxila, Pakistan
yaseer.durrani@uettaxila.edu.pk

2024 International Conference on Engineering & Computing Technologies (ICECT) | 979-8-3503-4971-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICECT61618.2024.10581134

***Abstract*—This paper reviews the recent research on network-on-chip (NoC) optimization techniques, which address the challenges of fault tolerance and congestion management in multi-core systems. It reviews the innovative strategies that improve the resilience and efficiency of NoC systems, such as fault-tolerant routing algorithms, wireless NoC congestion control mechanisms, dynamic voltage scaling techniques for NoCs, and design methodologies for heterogeneous architectures. It also evaluates the effectiveness of these strategies in terms of network latency, energy efficiency, and system reliability. The paper provides a comprehensive overview of the current state of the art and the future directions in NoC optimization, and offers valuable insights for researchers and practitioners in this field.**

***Index Terms*—Network-on-chip (NoC), Optimization techniques, learning algorithm**

I. Introduction

NoC, or Network-on-Chip, is a communication architecture employed in complex integrated circuits, such as multi-core processors and Systems-on-Chip (SoCs). Instead of relying on traditional bus-based communication, NoC utilizes a network-like structure to facilitate data exchange between different components and cores within a chip. This approach enhances scalability, reduces communication bottlenecks, and improves overall system performance by providing a more efficient and flexible means of interconnecting various processing elements. However, as the size and complexity of integrated circuits increase, so do the difficulties in designing and implementing NoC architectures. Some of the main challenges that NoC designers face are ensuring fault tolerance, managing congestion, optimizing energy consumption, and enhancing performance [1]. These challenges require novel solutions that can cope with the intricacy and diversity of current computing systems. In this paper, recent research is being reviewed that creatively tackles these issues and proposes new methods that aim to transform the NoC design field. The study explores the following aspects of NoC architectures and the recent studies that address them: Fault tolerance: the methods for fault detection, isolation, and recovery in the NoC components or links, and the impact of faults on the system functionality and reliability. Congestion management: the strategies for congestion avoidance or reduction in the NoC network, and the consequences of congestion on latency, throughput, power consumption, and load balancing. Energy optimization: the techniques for energy consumption reduction in the NoC system, which consumes a large fraction of the total power budget of multi-core architectures. Performance enhancement: the factors that affect the performance of the NoC system, such as topology, routing, arbitration, flow control, and buffer management, and the optimization of the NoC parameters for various application scenarios and requirements. Heterogeneous architectures: the benefits of exploiting the heterogeneity of the NoC components, such as processing elements, memory

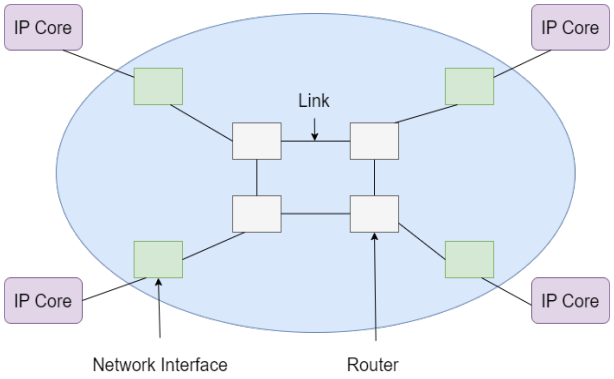

Fig. 1. NoC Components

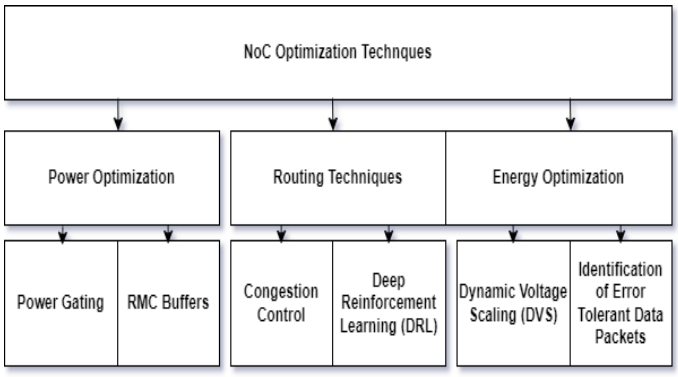

Fig. 2. Classification of NoC Optimization Techniques

979-8-3503-4971-9/24/$31.00 ©2024 IEEE

Authorized licensed use limited to: TRINITY COLLEGE LIBRARY DUBLIN. Downloaded on September 30,2024 at 15:21:18 UTC from IEEE Xplore. Restrictions apply.

elements, and accelerators, and the integration of different types of NoC networks, such as electrical, optical, and wireless, to achieve higher performance, lower power, and greater flexibility [2].

By analyzing these studies, the research aims to provide a comprehensive overview of the current state of the art in NoC research and to highlight the potential directions for future work in this area.

II. High Performance NoC using Reinforcement Learning

NoCs are increasingly becoming vulnerable to faults. These faults are classified into two categories which are permanent faults that occur due to the hardware aging and transient errors, which are caused by heated hardware, delays within transistors and due to runtime variations. Different techniques are used to compensate for these errors like SHIELD and Vicis but these techniques use a massive portion of the overall chip surface and are also inefficient. Some other strategies incur significant power usage and losses. Energy Efficiency is defined as:

$$EnergyEfficiency = [(P_{static} + P_{dynamic}) \times T_{exec}]^{-1} \quad (1)$$

Where P_{static} and $P_{dynamic}$ are static and dynamic power consumption respectively. The design incorporates eversible multi-function adaptive channel (RMC) buffers to improve the inter-router channel design. The reversibility of RMC reduces the overall power consumption, provides flexibility that improves reliability at link level and also enhances the performance by providing an extra layer for increased traffic. The CURE microarchitecture design also includes a self-diagnostics hardware that detects error at link level. The study also presents a learning model based on deep reinforcement learning (DRL) to overcome permanent and transient faults and also to lower the overall power consumption [3].

A. Advantages

RMC buffers and self-diagnostics hardware improve NoC reliability by detecting and correcting errors. RMC reversibility reduces power consumption, making NoC more energy-efficient than other techniques. RMC layer also boosts performance by handling more traffic and adapting to dynamic conditions. DRL learning model helps NoC cope with permanent and transient faults, increasing fault tolerance and system robustness. CURE NoC design reduces network latency by 39%, enhances energy efficiency by 92%, and improves reliability by 7.7 times compared to the baseline NoC.

B. Disadvantages

RMC buffers, self-diagnostics hardware, and deep reinforcement learning increase NoC complexity, making implementation and debugging harder. SHIELD and Vicis use much chip surface, affecting resource utilization. Deep reinforcement learning needs much training overhead, affecting NoC setup. CURE NoC performance depends on experimental conditions and workloads. CURE NoC needs trade-offs between reliability, power efficiency, and complexity based on application needs [4].

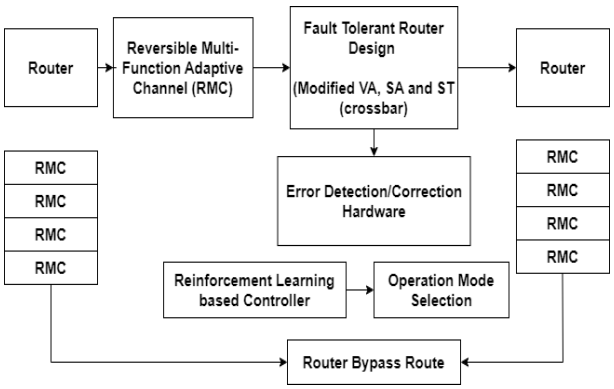

Fig. 3. CURE Architecture [3]

III. Fault Tolerant Routing Algorithm based NoC Design

The communication in NoC infrastructure is done via packets and smaller chunks of packets called flits. To determine the path of destination, routing algorithms are used. The deterministic category of routing algorithms is straightforward but not robust. However, the partly adaptive category of routing algorithms makes use of alternate paths for sending data therefore optimizing the network to some extent. And lastly the fully adaptive technique changes the path on the go based on the circumstances of the network. NoC architecture is sensitive to temperature and heat variations. Due to these sensitivities, there can be faults in the data path which may result in permanent and transient errors. To overcome these difficulties, the research presents a network adaptive fault-tolerant routing (NAFTR) algorithm. This algorithm makes use of the efficient dynamic and adaptive routing (EDAR) algorithm. To assess the enhanced reliability of the proposed designs concerning timing errors, we initially establish a transient error injection model. This model is designed to realistically generate a probability of timing errors for each link, providing a quantitative measure of the improvement in reliability. The probability of flit delivery (n-bit) is calculated as:

$$P_{fault} = 1 - (1 - R_e)^n \quad (2)$$

Where P_{fault} is probability of error in flit while probability of timing errors is given by Re. Aging factor is calculated as follows:

$$\triangle V_{\text{th}} = \triangle V_{\text{th}_{\text{NBTI}}} + \triangle V_{\text{th}_{\text{HCI}}} \quad (3)$$

Where $\triangle V_{th}$ is the shift in the threshold voltage, while $\triangle V_{th_{NBTI}}$ and $\triangle V_{th_{HCI}}$ are the threshold voltage of Negative Bias Temperature Instability (NBTI) and the threshold voltage of Hot Carrier Injection (HCI) drift-related aging effects in current technologies respectively.

$$Aging = 1 + \frac{\triangle V_{th}}{V_{th0}} \times 100\% \quad (4)$$

Where $Aging$ is degradation of circuit timing.

The NAFTR algorithm decreases the latency, ensures load balance in partially adaptive technique, and increases the flit

Authorized licensed use limited to: TRINITY COLLEGE LIBRARY DUBLIN. Downloaded on September 30,2024 at 15:21:18 UTC from IEEE Xplore. Restrictions apply.

delivery ratio. The algorithm is based on 2D mesh topology of NoC (Fig. 2). A local port connects each router to the processing element which has an ID that defines X and Y coordinates. The equation for finding the X and Y coordinates is as follows:

$$X = mod(ID, no.ofcolumns) \tag{5}$$

$$Y = [ID/no.ofcolumns] \tag{6}$$

The generic network is broken down into 8 sub networks. This allows the algorithm to select different paths based on circumstances. The emulating biologically inspired architecture in hardware (EMBRACE) router architecture is also modified by adding additional OR and AND gate to resolve conflicts regarding busy, faulty and congested channels more efficiently. The NAFTR algorithm increases the flit delivery ration by 18% in case of multiple faults [5].

A. Advantages

NAFTR uses EDAR to adapt to network changes and improve fault tolerance. NAFTR lowers latency, speeds up data transmission, and balances network traffic. NAFTR boosts flit delivery ratio by 18%, showing reliable data delivery with multiple faults. NAFTR divides network into 8 subnetworks, choosing different paths based on conditions. NAFTR handles temperature and heat faults, making NoC robust. NAFTR adds OR and AND gates to EMBRACE, solving channel conflicts better.

B. Disadvantages

Dynamic and adaptive routing algorithms and sub-network segmentation increase NoC complexity, affecting design, testing, and maintenance. EMBRACE router with more gates uses more resources, questioning hardware efficiency and scalability. Algorithm needs accurate path selection, but may struggle in some scenarios. Algorithm adds computational overhead, possibly lowering NoC efficiency. It shows better flit delivery with faults, but needs more details on fault types and algorithm applicability [6].

A recent study introduces an Integer Linear Programming (ILP) approach for designing deadlock-free routing algorithms in NoC architectures, suitable for various network topologies. It focuses on overcoming the deadlock issue in wormhole routing by analyzing and constructing deadlock-free routes for mesh and torus topologies. The study integrates application mapping with deadlock-free routing into a unified ILP framework, which is tested against benchmark applications. The results show that the ILP method outperforms heuristic approaches in achieving optimal solutions efficiently. Additionally, the study incorporates fault tolerance into the ILP method, exemplified by a 1-link-fault-tolerant deadlock-free routing for an MP3 application on a mesh network, highlighting the approach's adaptability and effectiveness in NoC design [7].

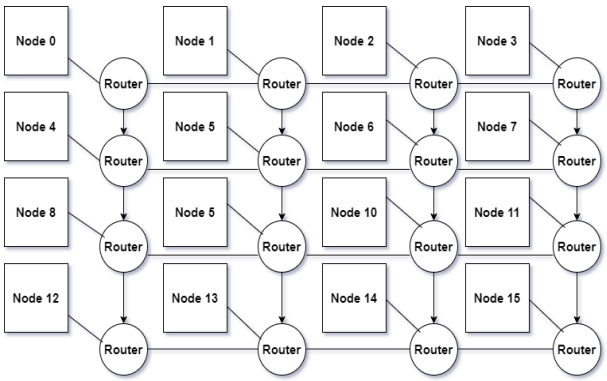

Fig. 4. 2D Mesh NoC Topology

IV. Congestion Control Mechanism in WiNoCs

Growing number of cores in wired NoC architectures have reduced the performance threshold in them. This problem is solved by the adaption of wireless network on chip architectures especially in the RF domain where communication latency is important. Throughput is given by:

$$Throughput = \frac{Time\ for\ transmission}{No.\ of\ bits\ transmitted} \tag{7}$$

However, congestion is one of the main problems related to WiNoC architectures. The bandwidth in WiNoC have a finite capacity and the number of wireless routers (WRs) is also low. To overcome these challenges, a survey of the congestion control mechanisms in WiNoC is presented. In hardware resources-based congestion control, router-based algorithms are implemented to reduce the latency. A hybrid architecture of both wired and wireless networks is discussed, which is useful in case one of the paths (baseline router, wireless router) has high traffic. Statistics based method is also discussed which makes use of the historical patterns of WRs and decide the communication path of network based on the information. Rate based congestion control method reduces the packets inflow to the network thus minimizing the overall congestion. The mechanism makes use of the flow control algorithms to adjust the packet size and transmission rate thus optimizing the network to its full efficiency. Network congestion ratio is given by:

$$Congestion\ Ratio = \frac{Utilizaed\ Network\ Resources}{Total\ Available\ Resources} \tag{8}$$

A congestion ratio close to 1 indicates high congestion, while a ratio close to 0 signifies low congestion. The design of medium access control (MAC) network also has a significant impact on the congestion control. Fixed channel-based MAC techniques offer a high throughput but do not work efficiently under high traffic environments. Whereas in random access-based MAC techniques, the network performance is low, but the design is flexible and scalable [8].

A. Advantages

Hardware resources-based congestion control lowers latency and improves performance in WiNoC. Hybrid architecture

Authorized licensed use limited to: TRINITY COLLEGE LIBRARY DUBLIN. Downloaded on September 30,2024 at 15:21:18 UTC from IEEE Xplore. Restrictions apply.

adapts to traffic and optimizes paths. Statistics-based methods use WRs patterns for adaptive path selection. Rate-based methods control packet size and rate, prevent congestion, and reduce inflow. MAC network design affects congestion control and resource utilization. Random access-based MAC techniques are flexible and scalable.

B. Disadvantages

Bandwidth limits in WiNoC cause congestion with low WRs, lowering throughput. Hybrid architecture adds design and management complexity, raising integration and maintenance issues. Statistics-based methods depend on WRs patterns, and may fail in changing or unknown conditions. Rate-based methods need flow control algorithms, adding computational overhead. Fixed channel-based MAC techniques trade off efficiency for throughput in high traffic.

V. Dynamic Voltage Scaling Technique

One of the major problems with NoC architecture is that the power consumption is a lot higher especially with the 3D NoCs. Although, 3D NoCs are better than 2D NoCs in terms of performance and scalability, but the consumption of power is where they lag. The main factor behind the power expense is due to the 3D routers used and Through Silicon Via (TSV) cost. To overcome these factors, a method is discussed which is known as Dynamic Voltage Scaling (DVS). In this technique, the resources of 3D NoCs such as router and links are optimized by switching between two levels of voltage. Flags are used to identify error tolerant data packet and the level of voltage is lowered when the approximal packet data is mapped. Results show an improvement of 27% and 31% in energy consumption when the approximal packets are 50% and 75% respectively of total [9].

A. Advantages

DVS lowers power consumption in 3D NoC, solving a major problem of high power usage. Voltage scaling optimizes resources, improving efficiency and performance. Flags mark error-tolerant packets for lower voltage, avoiding errors. DVS boosts energy efficiency by 27% and 31% for 50% and 75% approximal packets.

B. Disadvantages

Energy improvement depends on approximal packets. Benefits change with packet proximity. DVS saves energy but needs more techniques for better performance. DVS has limits and needs mapping and load balancing. These add design complexity and need algorithms. Power reduction and performance have trade-offs. High voltage scaling lowers performance. DVS helps but does not solve power issues in 3D NoC.

VI. Power Optimization using Power Gating

Power gating, clock gating and multi-level voltage supply are some of the techniques used to lower the energy consumption in an NoC. In power gating technique, NMOS and PMOS switches are used. The NMOS switch is connected between the power gated block and Vss. Whereas, the PMOS switch is connected between Vdd and the power gated block. The study describes several routers connected in mesh topology with power gated block attached. The power gating block controls the active and sleep state of the router thus minimizing the energy usage. The destination path of the packet will decide the corresponding router's state. In order to avoid deadlocks and cater for latency, some neighbouring routers are kept on. The router is notified 5 cycles earlier in order to receive the data and avoid missing any packet [9]. Total power is given by the equation:

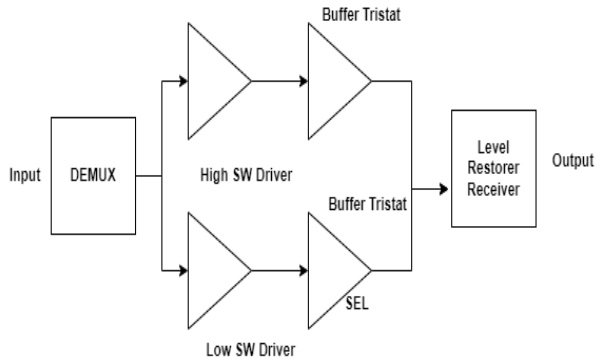

Fig. 5. Link Architecture [9]

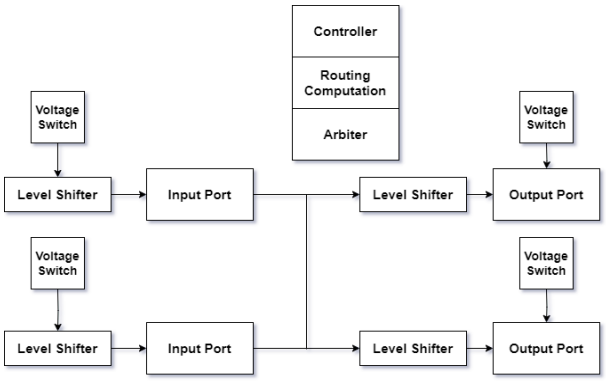

Fig. 6. Router Architecture [9]

$$Power(t) = Power(sw) + Power(st) + Power(leak) \quad (9)$$

Where $Power(t)$ is the total power, $Power(sw)$ is the switching power, $Power(st)$ is the power when short circuit, and $Power(leak)$ is the leakage power. In similar context, D. Belkebir, Lati and M. Belkebir provide a comprehensive classification of power management strategies for embedded systems [10]. These strategies based on the operational states—run-time, activation, and idle—of NoC components are categorized and emphasis is shown on the importance of power management by discussing key parameters and outlining the advantages and limitations of each technique. Additionally, it offers a classification based on NoC abstraction levels, aiming to inform researchers and developers about power management approaches that can enhance the efficiency and performance of future embedded systems.

Authorized licensed use limited to: TRINITY COLLEGE LIBRARY DUBLIN. Downloaded on September 30,2024 at 15:21:18 UTC from IEEE Xplore. Restrictions apply.

TABLE I
COMPARISON OF POWER WITH GATING AND WITHOUT GATING [9]

Router	Total Power (µW)	
	Without Power Gating	With Power Gating
2*2	47.705	41.03
2*3	107.3	93.91
4*4	217.09	175.498
8*8	878.34	701.66

A. Advantages

Power gating, clock gating, and multi-level voltage supply techniques reduce energy consumption in NoC architectures. Power gating controls the power states of individual routers based on the packet path, saving energy. Power gating blocks in a mesh topology optimize the network efficiency by adjusting the power states of routers according to packet destinations. Some routers stay active to prevent deadlocks and ensure packet flow. The routers are notified 5 cycles before data arrival, managing latency and avoiding packet loss. The total power equation considers switching, short-circuit, and leakage components, offering a complete way to optimize power consumption in NoC.

B. Disadvantages

Power gating, clock gating, and multi-level voltage supply in NoC architectures may increase design and management complexity, needing optimization and testing. Power gating may affect NoC performance and responsiveness. The notification system for routers needs precise timing, and any errors may impact latency. These power management techniques may add computational overhead for monitoring, control, and coordination. These techniques depend on design, topology, and traffic factors, making the implementation variable.

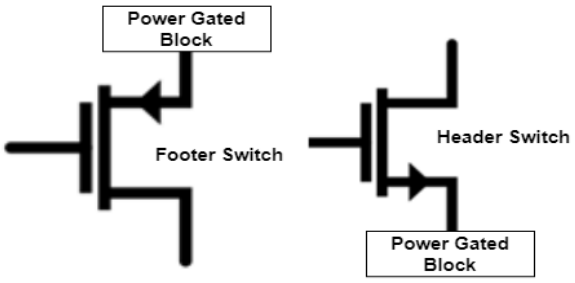

Fig. 7. Power Gating Technique [9]

VII. DESIGN FOR HETEROGENOUS METHODOLOGIES

Architectures that combine CPU and GPU are known as heterogenous architectures. While combining architectures have several benefits, they also have some challenges such as optimizing power and performance. This study presents a model based on fusion of CPU and GPU architecture while also improving their performance and power. The presented research introduces a Genetic Algorithm (GA) based approach for optimizing NoC configurations. GA is a random search technique that operates on a population of chromosomes, each representing a potential solution to the problem. The objective is to minimize packet latency. The algorithm employs various operators such as selection, crossover, and mutation that are guided by the survival of the fittest theory. A queueing-theory-based model is utilized to estimate average packet latency. Number of virtual channels and the buffer size are bound by the maximum limit:

$$\sum_{\forall S,D} P^{S\to D} L^{S\to D} \tag{10}$$

Where P is the set of processing elements to be placed on the NoC, S is the set of source nodes that generate traffic in the network and D is the set of destination nodes that receive traffic in the network.

Whereas buffer size of router and port is less than and equal to the original buffer size and number of virtual channels for router and port are less than and equal to the orignal number of virtual channels. The GA utilizes a chromosome representation consisting of 2D mesh routers with each router having an index representing its position in the NoC. Selection is performed through k-Tournament selection and two types of crossover operators, one-point, and partially mapped crossover, are applied to change the buffer size and placement of PEs respectively [11].

The proposed GA model is a systematic approach to optimize NoC configurations, determine PE placement, and specify buffer sizes and virtual channels. The methodology is useful for mechanisms governing the evolution of solutions in the context of large-scale design spaces.

A. Advantages

GA optimizes NoC configurations, improving performance of CPU-GPU architectures by lowering packet latency. GA is a flexible optimization technique that can explore and adapt to different configurations, scenarios, and constraints. GA uses a population of chromosomes and operators like selection, crossover, and mutation to search for optimal NoC configurations. GA solves the challenges of optimizing NoC configurations, PE placement, buffer sizes, and virtual channels. GA handles large-scale design spaces, offering a structured evolution of solutions.

B. Disadvantages

GA adds computational overhead, affecting optimization time. GA needs parameter tuning, and wrong settings may lower solution quality. GA may find local optima, missing better regions. 2D mesh routers in chromosome complicate design representation, needing careful mapping. GA performance changes with design space and scenarios. GA optimizes performance, but needs trade-offs with power.

Authorized licensed use limited to: TRINITY COLLEGE LIBRARY DUBLIN. Downloaded on September 30,2024 at 15:21:18 UTC from IEEE Xplore. Restrictions apply.

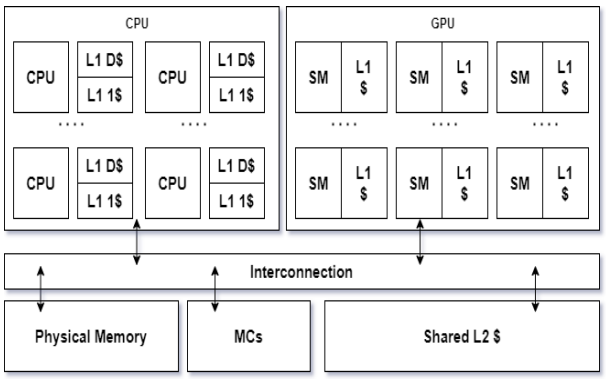

Fig. 8. CPU-GPU Fused Architecture [11]

VIII. Conclusion

This paper provides a comprehensive overview of the recent research on NoC architectures, which play a vital role in enabling efficient communication among numerous processing elements in multi-core systems. This study has explored how the recent studies tackle the key challenges that NoC designers face, such as ensuring fault tolerance, managing congestion, optimizing energy consumption, enhancing performance, and exploiting heterogeneity and integration. It has also discussed how the recent studies propose novel solutions that improve the resilience, efficiency, and adaptability of NoC systems, and how they compare with the existing approaches in terms of various metrics and benchmarks. Furthermore, this survey has identified the potential directions for future work in this area, and provided a useful reference for researchers and practitioners who are interested in advancing the NoC design domain. This review hopes to inspire further innovation and collaboration in this field, and contribute to the development of interconnected computing systems that can meet the growing computational demands of modern applications.

IX. Future Trends and Development

NoC architectures are constantly evolving to meet the challenges and opportunities of emerging technology, architecture, and application trends.

Photonics is an exciting and potentially disruptive technology for on-chip networks. In 2015, researchers at Berkeley demonstrated a microprocessor chip with on-chip photonic devices for the modulation of an external laser light source and on-chip silicon waveguides as the transmission medium [12].

Heterogeneous and hybrid NoCs exploit the diversity of the NoC components and networks to achieve higher performance, lower power, and greater flexibility. Heterogeneous NoCs can leverage the different types of processing elements, memory elements, and accelerators that are integrated on the same chip, such as CPUs, GPUs, FPGAs, and neuromorphic cores. However, heterogeneous and hybrid NoCs also pose new challenges and opportunities for NoC design, such as the optimization of NoC configurations [13].

Artificial intelligence and machine learning are revolutionizing the field of computing [14]. A notable study has applied AI, particularly neural networks, to NoC platforms, optimizing task distribution across processing cores. By mapping neurons to tasks and utilizing optimization algorithms, the study aims to reduce computation time. Simulations conducted in Octave/Google Colaboratory, varying the neural network's hidden layer complexity, have yielded improvements in energy efficiency, on-chip communication, and processing time, showcasing the symbiotic relationship between AI and NoC architectures [15][16].

These are some of the future trends and development in NoC architectures. There are many other aspects and dimensions that can be explored and improved in NoC design, such as security and privacy, testing and verification, modeling and simulation, and standardization and benchmarking.

References

[1] Mishra, Prabhat, and Subodha Charles, eds. Network-on-chip security and privacy. Springer International Publishing, 2021.
[2] Reza, Md Farhadur. "High-Performance Application Mapping in Network-on-Chip based Multicore Systems." (2024).
[3] Wang, Ke, and Ahmed Louri. "Cure: A high-performance, low-power, and reliable network-on-chip design using reinforcement learning." IEEE Transactions on Parallel and Distributed Systems 31.9 (2020): 2125-2138.
[4] Jagadheesh, Samala, and P. Veda Bhanu. "Noc application mapping optimization using reinforcement learning." ACM Transactions on Design Automation of Electronic Systems (TODAES) 27.6 (2022): 1-16.
[5] Nain, Zulqar, et al. "A network adaptive fault-tolerant routing algorithm for demanding latency and throughput applications of network-on-a-chip designs." Electronics 9.7 (2020): 1076.
[6] Narayanasamy, Poornima, and Seetharaman Gopalakrishnan. "Novel fault tolerance topology using Corvus seek algorithm for application specific NoC." Integration 89 (2023): 146-154.
[7] Liu, Shuang, and Martin Radetzki. "Systematic Construction of Deadlock-Free Routing for NoC Using Integer Linear Programming." 2023 IEEE 16th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC). IEEE, 2023.
[8] Rad, Farhad, Midia Reshadi, and Ahmad Khademzadeh. "A survey and taxonomy of congestion control mechanisms in wireless network on chip." Journal of Systems Architecture 108 (2020): 101807.
[9] Bijapur, Abhinav, Sumeet Siddappa Shirahatti, and R. Jayagowri. "Power Optimization Techniques for NOC."
[10] Belkebir, Djalila, Abdelhai Lati, and Malak Belkebir. "Physical and System Level Power Reduction Technique for NoC Architectures: Overview of State of the Art." 2023 International Conference on Decision Aid Sciences and Applications (DASA). IEEE, 2023.
[11] Alhubail, Lulwah, Masoomeh Jasemi, and Nader Bagherzadeh. "Noc design methodologies for heterogeneous architecture." 2020 28th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP). IEEE, 2020.
[12] Narayan, Aditya, et al. "PROWAVES: Proactive runtime wavelength selection for energy-efficient photonic NoCs." IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 40.10 (2020): 2156-2169.
[13] Fang, Juan, et al. "TB-TBP: A Task-Based Adaptive Routing Algorithm for Network-On-Chip in Heterogenous CPU-GPU Architectures." (2023).
[14] Lin, Ting-Ru, et al. "A deep reinforcement learning framework for architectural exploration: A routerless NoC case study." 2020 IEEE International Symposium on High Performance Computer Architecture (HPCA). IEEE, 2020.
[15] Suleman, Tayyaba, and Zeeshan Ali Khan. "An Efficient Algorithm for Mapping Deep Learning Applications on the NoC Architecture." 2023 25th International Multitopic Conference (INMIC). IEEE, 2023.
[16] Khan, Afshan Amin, and Roohie Naaz Mir. "Scheduling Strategies and Future Directions for NoC: A Systematic Literature Review." Automatic Control and Computer Sciences 57.4 (2023): 413-421.

Authorized licensed use limited to: TRINITY COLLEGE LIBRARY DUBLIN. Downloaded on September 30,2024 at 15:21:18 UTC from IEEE Xplore. Restrictions apply.