



## TOPICAL REVIEW

## OPEN ACCESS

RECEIVED  
15 July 2021

REVISED  
31 December 2021

ACCEPTED FOR PUBLICATION  
21 January 2022

PUBLISHED  
18 February 2022

Original content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](#).

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.



# Roadmap for machine learning based network-on-chip (M/L NoC) technology and its analysis for researchers

K Balamurugan<sup>1</sup> , S Umamaheswaran<sup>2</sup> , Tadele Mamo<sup>3</sup> and Lakshmana Rao Namamula<sup>4</sup>

<sup>1</sup> Department of Electronics and Communication Engineering, Swarnandhra College of Engineering and Technology, Andhra Pradesh, India

<sup>2</sup> Department of Computer Science and Engineering, MVJ College of Engineering, Bangalore, India

<sup>3</sup> Department of Mechanical Engineering, College of Engineering and Technology, Mettu University, Ethiopia

<sup>4</sup> Department of Information Technology, University of Sierra Leone, Sierra Leone

E-mail: [s.nagarajan@meu.edu.et](mailto:s.nagarajan@meu.edu.et)

**Keywords:** NOC, machine learning NoC, 3D NoC, 3.5D NoC, 2.5D NoC, NoC security

## Abstract

A few decades ago, communication inside the chip is done by transferring signals between the cores. This conventional method is not worthy because of the increase in latency and power consumption. To rectify this issue Network-on-Chip (NoC) technology has emerged. NoC technology is invented to transfer data packets instead of signals. Machine Learning NoC (M/LNoC) is a very fast-growing technology in today's Integrated Circuit world for the communication between Intellectual property (IP) cores. The machine learning algorithms are used in the existing and emerging novel NoCs. In this paper, various evolving NoC technologies to decrease the transfer latency, power consumption of the IC is addressed for the implementation of the machine learning algorithm. The NoCs working with machine learning algorithms are called M/L NoC. We also provided the security issues to be focused-on in the M/L NoC. Also, we have provided the available NoC tools for the NoC researchers.

## Contents

1.	Introduction	3
2.	NoC Technologies for the implementation of machine Learning	4
3.	Emerging research in NoC	8
4.	NoC tools	10
5.	Conclusion and future scope	12

## 1. Introduction

Multicore Integrated Circuits are popular nowadays due to its high performance. Interconnection inside a chip is very important for proper communication between the Intellectual property (IP) cores. In multicore chips, the delay is mainly due to lengthy wires that affects the overall performance of the chip. The wire temperature also increases during the signal transfer between two distant IP cores. To eliminate those disadvantages in the conventional method, the NoC technology has emerged. This technology transmits digital packets instead of analog signals. The architecture of NoC contains (i) Intellectual Property (IP) core (ii) NoC Router (iii) Network Interface (iv) Topology (v) Channel.

IP core is a chip that provides the Intellectual property of the vendors. In a multicore processor, many IPs are incorporated in a single chip as illustrated in figure 1. These IPs inside the chip can be homogeneous or heterogeneous. Homogenous SoC contains IP cores with the same characteristics and Heterogeneous contains IP cores with different characteristics [1].

The router in the NoC architecture is to route the packets to the exact destinations. Most of the NoCs follow the multicast routing technique to achieve high speed. Multicast routing is the method to send packets from a single source to different destinations. NoC router needs an algorithm to route the packets via shortest path. The NoC algorithm plays a significant role in the transmission and reception of packets. Designing a perfect NoC router may reduce the area of multicore chip as illustrated by the authors in the work [2]. Researchers in this NoC area concentrate in developing NoC algorithms for achieving high-speed communication. The NoC routers use either circuit switching or packet switching technology for data transfer. Circuit switching and packet switching are useful for less complex and high complex switching paths respectively. The speed of the circuit switching depends on the distance between the source and the end node. The speed of the packet switching depends on the type of router used. Packet switching has more advantage than circuit switching in terms of traffic, data transfer speed etc [3]. Most of the NoC architectures are designed to use packet switching instead of circuit switching. Compare to the other packet switching, wormhole packet switching is accurate and fast. Hence, researchers are giving priority to wormhole packet switching. In this switching, messages travelling from a core to another core are divided into several data packets. This process is called Packetization.

The router communicates with the IP core through Network Interface (NI). The NI is connected with IP core in one side and the router in the other side. NI divides the data packet into flits (Flow Control Units) to speed up the data transfer. This process is called the Flitization process. The flitization process occurs at the source node and at the destination node these flits are recombined to form packets. Each flit is made up of one or more physical unit called as Phits. For Example, a message is divided into 'p' number of packets; Each packet is splitted into 'm' number of flits; Each flit is again splitted into 'n' number of phits [2]. Each flit contains a header, body, and tail flit. The header flit consist of the routing information such as the address details of the source, destination, and the path details from the source to destination. The other flits like body flit and tail flit followed the header flit. The NI assembles the order of flits and inserts the header information.

The topology used in the NoC plays an important role to reduce the latency and the power consumption [4]. The important parameter in the design of algorithm is the NoC topology. Each node of the NoC topology contains the IP core and the router. The IP core and the router is connected via NI. Some of the topologies used by the researchers [5] in their work are Ring, Star, Mesh, Torus, Binary-Tree, Fat Binary Tree, Butterfly, and SPIN. The authors of [6] stated various factors to be considered in topology research are node degree, diameter, link complexity and the bisection width.

Channel is a medium of NoC through which the packets are transferred from the source node to the destination node. There are two types of medium available in NoCs. They are (1) Wired medium (2) Wireless medium. The components of NoC architecture explained above are obligatory to design any type of NoC. NoCs are designed with different routing algorithms, modes of data transfer and additional features. Recently, machine learning algorithm is used to develop the routing algorithm of NoC to improve its performance.

In this paper, section 2 illustrates the various NoC technologies available for the implementation of Machine Learning. Section 3 focuses on the emerging research area in NoC. Section 4 illustrates various tools available for NoC research. Chapter 5 illustrates the conclusion.

## 2. NoC Technologies for the implementation of machine Learning

As the conventional communication consumes more power because of the lengthy interconnection wires, express channels (also called low power express channels) are introduced on the wire. As these express channels are also basically metal wires, it also consumes more power [7]. It leads to the emerging of other NoC technologies such as Wireless/RF/Millimeter-wave Network-on-Chip, Hybrid Wireless Network-on-Chip, 3D NoC, Photonic Network-on-Chip, etc are proposed to reduce these problems. Recently Machine Learning (M/L) techniques is implemented on the above NoCs to achieve high performance.

### 2.1. Wired NoC

Wired NoC is the conventional NoC communication inside the chip [8]. This method can use any of the available topologies according to the usage of the chip. In Wired NoC, metal wire is used to connect two nodes. The data transfer process can be done only through this metal wire. A lot of researches were already done in this area to improve its performance parameters. In this structure, each node contains an IP core. The IP core is attached to the NoC wired router through the Network Interface Unit. As communication is through the metal wire, the speed of the data transfer is considerably higher than a few of the other NoCs [9]. Many algorithms are developed to reduce the latency and the power consumption of this NoC. This wired NoC can be designed with 2D and 3D architectures.

### 2.2. Wireless/RF/millimeter-waveNoC

Wireless NoC is the emerging NoC technology [10]. The wired routers at the nodes are replaced with wireless routers. The router in a node is communicating with the other routers through wireless technology. The same procedure is followed for both neighbour nodes as well as the long-distance node. This requires a lot of power for the transmission. The latency is also not considerably reduced in this technology. By using this method, the area of the chip will be considerably reduced, as it does not use the metal wire for the connection between the nodes [11]. The physical faults due to the metal interconnection will be avoided in this method. Wireless NoC does not achieve 100% throughput during communication due to the packet loss. The packet loss of wireless NoC is higher than wired NoC.

### 2.3. Hybrid network on chip

Hybrid NoC or Hybrid wireless NoC performs both wired communication and wireless communication for its operation visualized in figure 2. This NoC uses wired communication for short distance communication and wireless communication for long-distance communication. Compare to the wireless NoC, this hybrid NoC shows improved performance results [12]. In some research, the wired NoC router is replaced with wireless

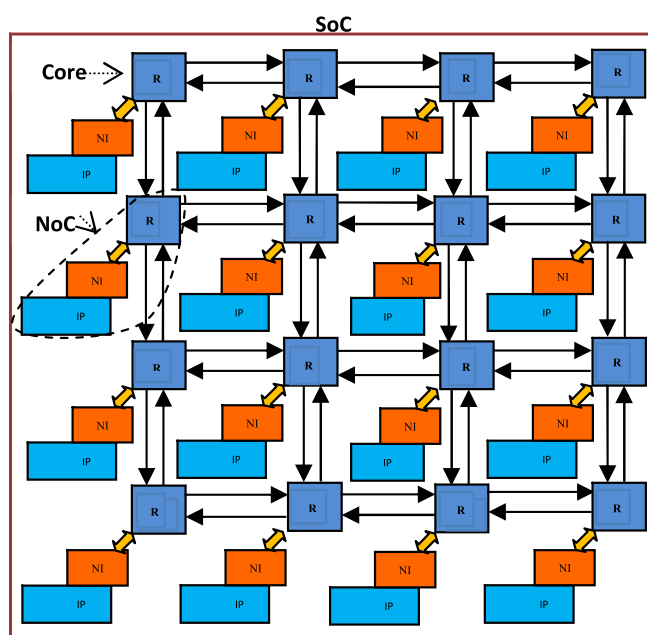


Figure 1. Hybrid NoC structure.

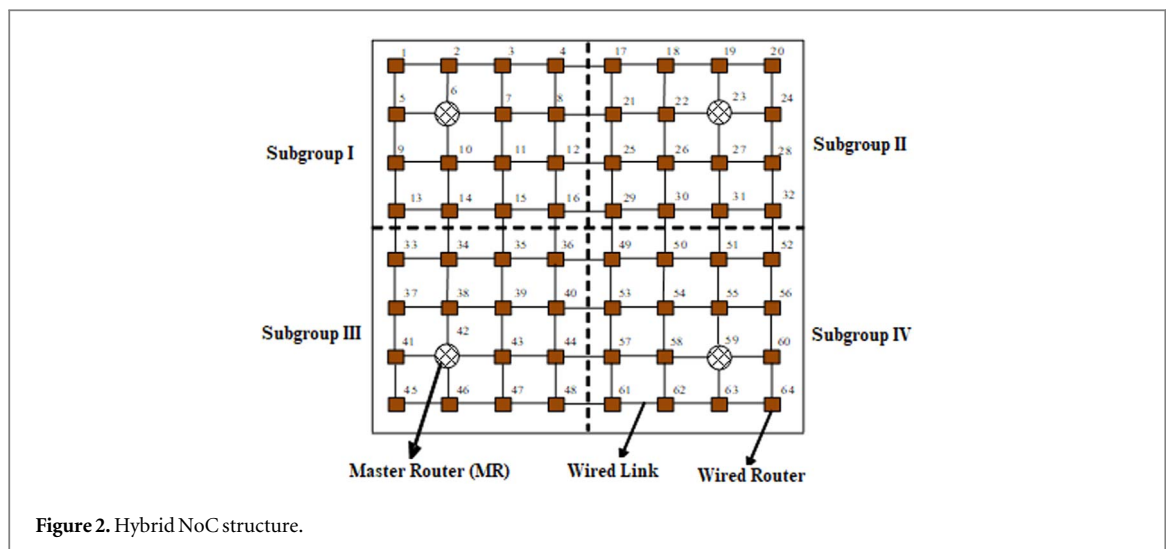


Figure 2. Hybrid NoC structure.

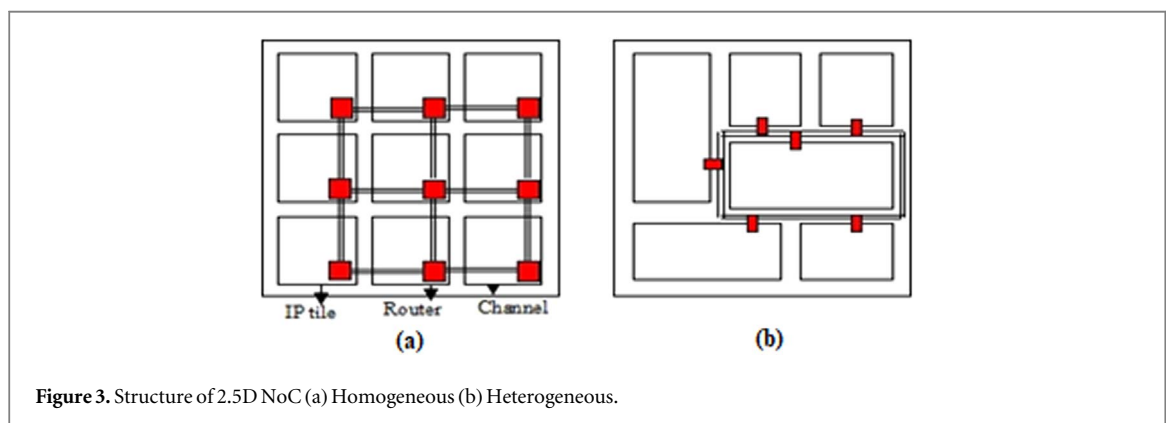
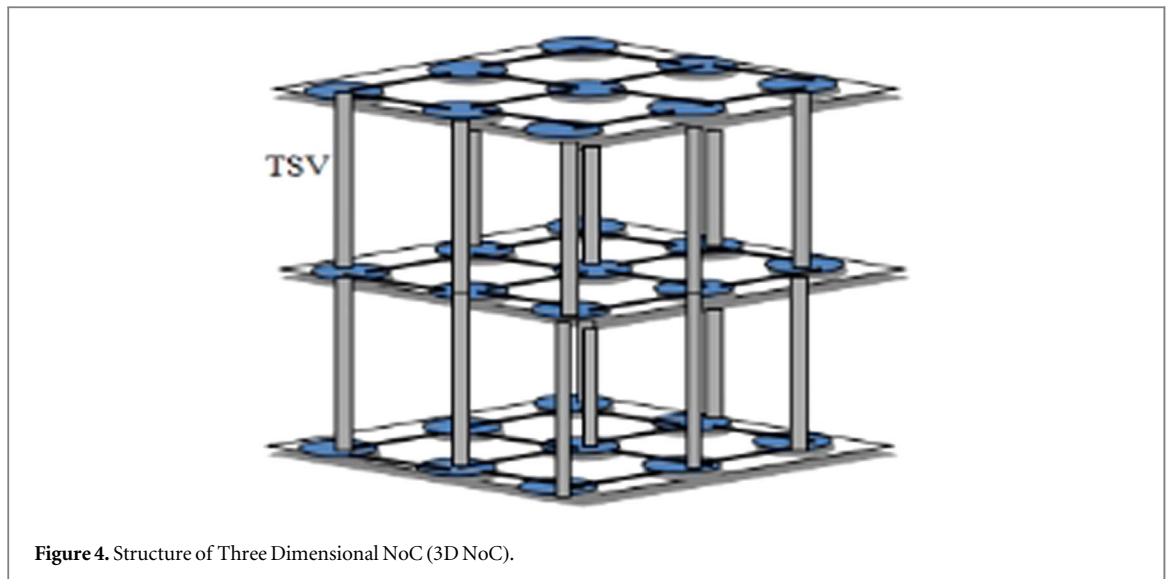


Figure 3. Structure of 2.5D NoC (a) Homogeneous (b) Heterogeneous.

router but in some cases, additional wireless routers will be added along with the required number of wired routers.

#### 2.4. Two-point five dimensional (2.5 D) NoC

The 2.5D NoC architecture has two silicon substrates placed on the top of the parent silicon substrate. The authors of [13] stated that out of the two silicon layers, one is homogeneous and another one is heterogeneous. In the homogeneous layer, the IP cores are arranged uniformly by perfect floor planning techniques [14] as shown in figure 3(a). In the heterogeneous silicon layer, the IP cores are not arranged uniformly but randomly by floorplanning techniques as shown in figure 3(b). Layers in SoC provides improved performance by using appropriate buffer size, mesh size, optimized virtual channels. The authors in the work [14] specified the advantages of using a heterogeneous compact layer by providing optimal floor planning and wiring. Their experimental results shows considerable amount of performance improvement compared to 2D NoC. The authors of [15] proposed another 2.5D integrated multicore network-on-chip with heterogeneous components in each of the two silicon die. The heterogeneous components in the silicon die are 8 MIPS microprocessor core, memory, and accelerator. The designed 2.5D NoC architecture is verified for many digital applications. They designed a 65 nm technology chip operated at 0.5 GHz with 1.08 W power dissipation for a 1.2 V supply. To avoid the thermal challenges, the authors proposed 2.5D NoC which uses an interposer. The interposer in the chip is a normal silicon chip in which metal layers are facing the upward direction. The interposer is designed with small input-output pads, clock generation, and all other types of signalling, etc. Many IP cores [15] such as CPU, GPU, DRAM, NVRAM, are placed on the top of the silicon interposer. These IP cores are mounted horizontally on the interposer. The IP cores are communicated through metal contact and the interposer. The metal connectors are used for the core to core communication and TSVs (Through Silicon Via) are used for vertical communication. In this 2.5 D technology, complete wireless technology is not yet used so far.



**Figure 4.** Structure of Three Dimensional NoC (3D NoC).

### 2.5. Three dimensional (3D) NoC

The two dimensional (2D) NoC performs an efficient operation for many applications. As the functions of electronic devices such as computers, laptops, mobile phones, notebooks, etc are getting more and more complex, the chip designer looks for other options in the NoC area. This leads to the invention of 3D NoCs. Three-dimensional (3D) NoCs are a very popular and fast-growing technology in the IC era to achieve high-speed data transfer in SoCs. The performance parameters such as the size of the chip, speed, power consumption are considered during the design of 3D NoC. In 3D NoC, three number of 2D NoC layers are arranged one above the other using vertical interconnects (TSV) as illustrated in figure 4. In 3D NoC, hundreds or thousands of chips are integrated on a single IC. The authors of [16] stated that the new two layers of the silicon substrate are grown on the top of the parent substrate in a sequence to build 3D NoC. The growth of silicon over the top of the parent silicon substrate is done by the fabrication technologies such as silicon epitaxial growth, recrystallization process, and solid-phase crystallization. There are two integration techniques [17] to build the 3D IC architecture (1) 3D Through Silicon-Vias (TSV) and (2) 3D monolithic integration. The advantages of 3D NoCs are (1) Improved Latency (2) Global interconnects are shorter (3) Power consumption is low (4) High Density. Even though many advantages are there in 3D NoC, there are a few disadvantages that are also available. They are (1) As the density of the 3D IC is more, the power density also increases which leads to thermal heating. Sometimes the thermal power will not dissipate fully from the second layer of the three-layer 3D NoC. This causes a serious problem in the 3D ICs. (2) The size of the 3D IC reduces horizontally but it increases vertically. The TSV [18] used to connect the layers vertically takes more area compared to the conventional connecting wires. TSVs are like a waveguide that requires  $5\mu \times 5\mu$  space and it needs more area to fix it on the substrate or layer. (3) Increase in cost: TSV uses copper for its waveguide, so this increases the manufacturing cost of the chip. The Monolithic 3D ICs are having a base silicon substrate and at the top of the base layer, additional metalized layers, and circuitry is formed. The purpose of the Via is to connect the layers of the IC vertically. Network-on-Chip (NoC) circuitry is made above the silicon substrate [19]. This is known as 3D monolithic NoC. Monolithic 3D NoC.

### 2.6. Three-point five dimensional (3.5 D) NoC

Three-point five (3.5D) NoC is the hybrid technology of 3D NoC. It uses monolithic integration technology and TSV vertical integration technology. Hence this technology is named 3.5D Technology. A case study [20] is done with 288 core Multiprocessor System on Chip and predicted that the cost of manufacturing and testing. It is proved that this 3.5 D technology decreased by 20% of the manufacturing cost and 30% of the test cost compared to other 3D Integration technology. This work is done by connecting all the cores of the SoC. Also, it is proved that the overall performance of the chip is increased by an average of 11.5% and the latency is improved by 25.4% compared to the 3D Integration technology. The researchers can use 3.5D technology according to their applications for N number of cores.

### 2.7. Photonic network-on-chip

Another method to improve the performance parameters of NoC is Photonic or Optical NoC. The Photonic NoC has a laser source to send light energy through optical waveguides (optical fiber) with different wavelengths.

As the data travels with the speed of light ( $3 \times 10^8 \text{ m s}^{-1}$ ), these photonic NoC chips are emerging nowadays. The optical coupler couples the optical fiber and the optical waveguide. The waveguide can allow a different range of wavelengths, so parallel communication can be done on the same waveguide. Another component named MR (Microring Resonator) is used for the modulation technique and segregate the wavelength of the parallel signals [21]. The MR can also be used to encode the received data. The signal sent by the sender is received at the receiver side by the photo-detectors. The photo-detectors detect and also convert photons into electrons for a wide spectrum of signals. As per the statement of [22], there are two types of buses available for Optical NoC. They are (1) Single-Writer-Multiple-Reader (SWMR) bus (2) Multiple-Writer-Single-Reader (MWSR) bus. In MWSR, each sender sends its modulated signals with separate wavelengths to the single receiver. Hence the receiver receives multiple signals with different wavelengths from many senders [22]. The receiver receives the signal and the data is decoded. The router transfers the data to the Intellectual Property (IP) core via Network Interface (NI) for further process [23].

As different wavelength is used for the signals there is no possibility for any interference or collision of signals. In SWMR, many receivers will receive the same signal sent by a single sender. It is concluded that the Photonic NoC will have low crosstalk, less power consumption, and less chip area.

Authors of [24] have provided an idea to alter the source of Photonic NoC by Silicon avalanche light-emitting devices (SiAvLEDs). They also proved that the light emission point of SiAvLED is about one micron just below the silicon-silicon oxide interface. They have proved that the SiAvLED optical source achieved high performance compared to other Optical sources.

Authors of [25] used Si field-effect LED (SiFELED) as the optical source and obtained its performance in Optical network. They compared SiAvLED with SiFELED and proved that SiFELED is better than SiFELED in terms of its performance parameters such as speed, throughput etc.



### 3. Emerging research in NoC

Although many NoC architectures are there for the communication inside the chip, every day new technologies are emerging. As the world is moving towards the miniaturization of electronic devices, the size of the chip also decreases. This in turn affects the NoC architecture and this leads to the innovation of new NoCs.

#### 3.1. Machine learning NoC

The Machine Learning techniques can be adopted for the NoC Technologies such as wired NoC, Wireless NoC, Hybrid NoC, 3D NoC, 3.5D NoC, 2.5D NoC, Photonic NoC. Machine Learning NoC (M/L NoC) depends on the characteristics of the NoCs mentioned above. As monitoring is needed for the NoC operation, most of the M/L NoCs are using supervised learning algorithms such as classification and regression. The routers in the NoCs will be trained by the M/L algorithm to transfer the packets from the source router to the destination routers. These supervised learning-based M/L algorithms are very efficient in terms of speed and power consumption. Some of the popular M/L algorithms used by the researchers [26] are linear regression algorithm, decision tree algorithm, random forest algorithm, K-nearest neighbour algorithm. Hence M/L based NoCs may be the choice of many upcoming researchers.

The data set for the N core NoC is developed by considering every node as the source node and the possible paths for the data packet. The data set is divided into 80% and 20% for training and testing NoC respectively. Any one of the M/L algorithms can be used. Once the required performance is achieved by the NoC model, then it is considered to use in the real-time scenario. The M/L algorithm can also be used to predict the accurate delay before sending the packet from source to destination. Figure 5 illustrates the implementation of a machine learning algorithm in NoC in the process of sending the data from source to destination [27]. The input details such as source address, a destination address, data length are given to the M/L algorithm. With the aid of the data set generated, the M/L algorithm determines the efficient path to reach the destination. If any congestion due to traffic exists, then the M/L algorithm will determine the alternate path to reach the destination.

The parameters are to be considered during the architecture design of M/LNoC are (1) Grid features (2) Topological features, (3) Selection of routing, (4) Traffic Features, (5) Quality of service, (6) Size of the buffer and (7) Deadlock, livelock avoidance. Grid features contain the size of the buffer, number of I/O ports, the total number of agents, total power, total voltage, clock-frequency, and NoC protocols [28]. Traffic Features provides the end-to-end connection between agents, types of interfaces/channels, and bandwidth, latency requirements for each agent, QoS, and real-time requirements. Topological features provide the type and size of the topology of the chip, algorithm for routing, size of agents, area for NoC routing in SoC, and the thickness of the routing wire. The authors applied the M/Lbased NoC technique in various SoCs and compared M/L NoC with other conventional NoCs. It is proved that M/L-based NoC is superior to others in terms of performance.

By using thousands of SoC designs, the authors of [29] have generated training data with many characteristics. Next, the predictive models are generated using algorithms such as random forest, neural network, multi-variant linear regression, vector machines, or pattern matching. After the training, the generated model is used to determine the steps to design an optimal interconnect system using linear regression. Similarly, after the completion of training and model generation, the new specification is applied to the preferred Machine Learning predictor. Based on the values provided, the predictor develops different strategies. This M/L technique allows the routers in a core of the chip to decide itself to communicate with the other core if the fault exists. The authors used 30 numbers of real-time SoC for their evaluation of M/L NoC. Also, three predictive models such as balanced, performance, and area are used for the experimental analysis.

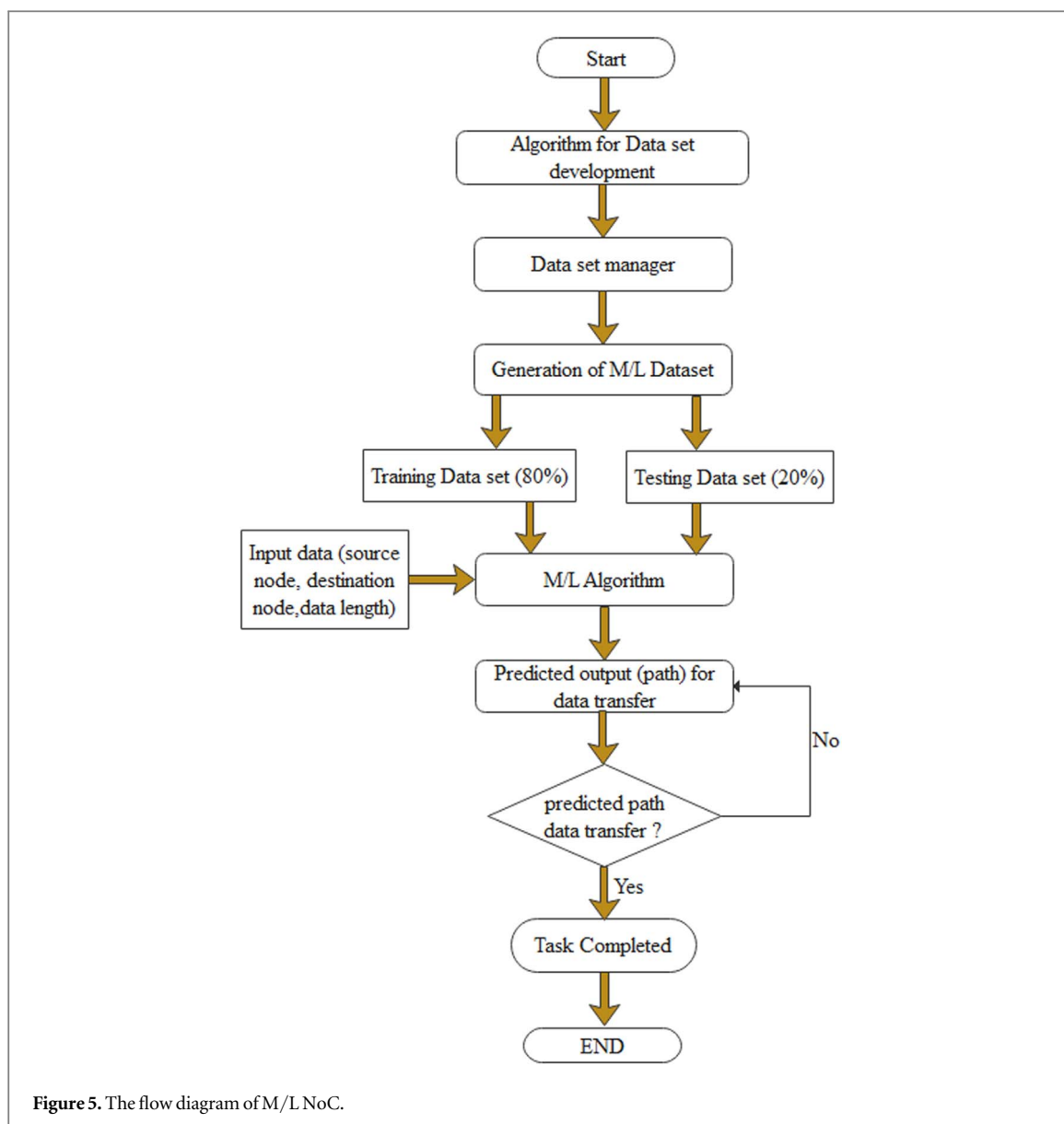
The performance model deals with the performance of the chip. In the Area Model, the MLNoC is looking for possible ways to minimize the area of the chip. The authors of proposed the Regression-based M/L model in the training phase and the testing phase. During the testing phase, the machine learning algorithm of NoC predicted the results.

Similarly, the Machine Learning Technique is used to develop the framework of NoC that is helpful to evaluate the performance of NoC and it can be used to improve the performance. The authors of [29] developed an NoC framework and compared it with the cycle-accurate simulator-Books. It is proved that the machine learning based framework is more efficient than Booksim.

#### 3.2. Machine learning based NoC security

Another important feature in M/L NoC is the security issues. Security of NoC is very much important nowadays to protect the IP cores from hackers. Nowadays, Hackers are extracting the data from the IP core itself. Hackers can able to malfunction the IC by using anonymous threats. This kind of anomaly affects the hardware part of the NoC such as routers, network interfaces, IP cores which in turn affects the electronics devices such as mobile phones, laptops, tablets, etc [30]. Nowadays the Internet of Things (IoT) is also emerging drastically. This IoT





device uses ICs which work with the Internet. This paves the way for hackers to send cyber anomalies through the internet. They can alter the data packet in the NoC channel to collapse the system. This NoC security is the emerging research area for the researchers.

The authors of [31] stated that security issues are mainly because of third-party IP companies. As most of the IC design companies are using IP cores from third parties, it leads to the insecurity of NoC. The detection and mitigation of cyber threats can be done by many techniques. The researchers used firewalls to secure the NoC. The authors of [32] created their theft detection mechanism and they avoided the theft done by the hackers. To experiment, they inserted some malicious data to rob the data. The algorithm developed by the authors detected the possibility of theft and it protects the data by encryption. The authors of [33] stated that cyber threats affect the Look-up table of the routers. Because of these security issues in NoC, energy consumption also increases. So, to reduce the energy consumption of NoC the authors used Cyclic Redundancy Check (CRC) and AES encryption system to avoid faults or attacks. Machine Learning-based technique is a very efficient technique to identify and suppress anomalies or threats. The authors explain the Machine Learning techniques used in the security of NoC. They stated that many techniques are there to predict the malfunction. The prediction is based on any of the regression such as Linear regression (LR), Multiple Linear regression (MLR), Decision Tree regression (DTR), Random Forest Regression (RFR), Artificial neural networks, Bayesian learning, computational learning, instance-based learning, genetic algorithms, and reinforcement learning.

## 4. NoC tools

One more difficulty in carry-out the research in NoC is the NoC Tools. By using the simulation tool, the performance parameters of NoC can be obtained. Table 1 denotes some of the effective NoC simulation tools, the operating system, languages or software used, and the organization that developed those tools. This helps the researchers to select the appropriate tool for their research work.

NS2, NS3 are network simulators [34] useful for any network to determine the performance of the network. NS2 and NS3 can be used for NoC to determine its performance such as speed and power consumption. This simulator is used to design and analyze various components of NoC and its routing. Noxim [35] is the simulator for NoC to determine its parameters. By using this simulator, one can customize the structure of NoC, routing algorithm, data transmission and reception control, Synthetic traffic generation and distribution etc DARSIM [36] is a simulator for designing the architecture of NoC. It uses wormhole router in the NoC structure. To speed up the data transmission and reception, it uses a table-based routing methodology. This simulator also uses Virtual Channel (VC) allocation scheme for routing inside the NoC. Visualsim [37] is a new simulator for NoC research. It is a user-friendly simulator helpful to obtain the delay and power consumption of NoC. ATLAS [38] simulator environment is useful for building the NoC architecture. The NoC can be simulated without traffic and with traffic. The simulator itself creates the traffic for the data flow. By using the ATLAS simulator the performance such as power estimation, delay etc can be determined. ATLAS is a graphical-based simulator that can generate performance tables and output reports. ATLAS uses the VLSI tool Modelsim for simulation. It also uses the GNU plot tool for generating the output graphs. HORNET [39] is a simulator helpful for the multicore NoC simulation. It has the capability of generating the algorithms for the Virtual channel allocation and the routing for the NoC. The performance evaluation of the NoC can be obtained both for traffic scenario and no traffic. X-pipes compiler [40] is a simulator useful for simulating NoC of the heterogeneous System on chip (SoC). This simulator automatically optimizes the building blocks of NoC. It is useful to get optimized results in terms of latency and power consumption. Mspider [41] is a CAD-based simulation tool for NoC. It generates the VHDL code for the NoC design. The generated VHDL code can be synthesizable on FPGA platforms. It uses PHP for generating the output graphs. Mspider is very much useful for getting the performance parameters of NoC such as delay, power consumption and area.

NoCGEN [42] is a simulator used to generate the mesh topology based networks in NoC. Using NoCGEN one can design the components of NoC and its routing. The user can vary the number of cores, routers, data length and buffer size. Wormhole based routing can be used with this simulator. The simulator generates the VHDL code for the NoC. FlexNoC [43] is a commercial simulator designed by With Arteris. FlexNoC is one of the best simulators for NoC interconnection.

FlexNoC concentrates on the congestion during data transfer, delay, area and the power consumption of the NoC. FlexNoC provides the details of data throughput also. The users can achieve high throughput by adjusting the parameters of the NoC. INSEE [44] is a simulation environment that can generate the traffic for the NoC simulation. INSEE is a lightweight tool that can inject data packets in the designed NoC architecture and the performance of the network can be determined. OMNeT++ [45] is a framework for designing the Network simulator. This simulator is useful for the on-chip wired and wireless simulation. It is a GUI based simulator with eclipse based IDE. The components of NoC are designed using C++. PIRATE [46] simulator is used to design the NoC architecture with Intellectual Property (IP) core. This Tool is also used to design the routers and other interconnection components such as channels etc. This tool can be used in any of the network topologies of NoC. By adjusting the load of the network, high-performance NoC in terms of power and delay can be designed. OCCN [47] is the short form of On-Chip Communication Network. It is a framework developed for NoC research. It can develop the communication drivers for the NoC using a universal Application Programming Interface (API). It concentrates on the area of the NoC design and the performance issues of NoC. AcENoCs [48] is the FPGA emulation platform for NoC architectures. It is a flexible tool that provides the Hardware-Software framework for NoC research. The tool has traffic generators, reconfigurable NoC components and analysis modules. This tool has additional features to reconfigure the system. AdapNoC [49] and DucNoC [50] are also the configurable, flexible NoC research tools. Both AdapNoC and DucNoC uses the table based NoC algorithm for increasing the speed of NoC. It has both Traffic Generators and Traffic Receptors. It can generate synthetic traffic and also it can work with real-time traffics.

BookSim is a standard cycle-accurate interconnection network simulator. Booksim is used to customize the router of the NoC. By customizing the NoC router the performance parameters of NoCs can be improved. Booksim can use the topologies such as mesh, torus and flattened butterfly networks. It also provides routing algorithms with numerous options. NETMAKER [51] is also a general-purpose efficient simulator for Network simulation. Netmaker can be used for developing the topologies of NoC and the simulation results can be taken. NIRGAM [52] is a tool for NoC research. This tool works for the 2D mesh and torus network. It uses the wormhole based switching mechanism. The tool can generate the routing algorithms such as deterministic XY,

**Table 1.** List of various NoC simulation tools.

Name of the simulation tool	Developed by	OS	Language/Software
Network Simulator [34]	Lawrence Berkeley National Lab	Linux, FreeBSD, macOS	C++, Python
Noxim [35]	University of Catania	Linux	SystemC, C++
Darsim [36]	Massachusetts IT	Linux	C++, Python
Visualsim [37]	Mirabilis Design	Windows, Linux	System C
Atlas [38]	GAPH research group	Windows	Java, VHDL
<b>HORNET</b> [39]	Massachusetts IT	Linux	C++, Python
Xpipes compiler [40]	Bologne—Stanford Univ.	Linux	SystemC
$\mu$ spider [41]	LAB-STICC	Linux	VHDL
NoCGEN [42]	LaNoC-UFC	Windows, Linux	VHDL/Verilog
FlexNoC [43]	Arteris	Windows/Linux	SystemC
Insee [44]	Manchester University	Linux	ANSI C
OMNeT++ [45]	OpenSim Ltd	Windows/Linux	C++
Pirate [46]	Politecnico di Milano	Linux	SystemC
OCCN [47]	STM, AST Grenoble Lab	Unix/Linux	SystemC, C++
AcENoC [48]	Texas A&M University	Windows/Linux	Xilinx Microblaze
AdapNoC [49]	Sharif University of Tech.	Windows/Linux	Xilinx Microblaze
DuCNoC [50]	Sharif University of Tech.	Windows/Linux	C++, Verilog HDL
BookSim [24]	Stanford University	Windows/Linux	C++
Netmaker [51]	University of Cambridge	Linux	SystemVerilog
NIRGAM [52]	Soton university	Linux	System C
ARTEMIS (3D) [53]	Colorado State University	Linux/Windows	MATLAB
Gem5 [54]	University of Michigan	Linux	System C

Adaptive odd–even(OE), Source Routing. ARTEMIS [53] is the tool for analyzing the NoC system. This tool is useful for determining the efficiency of the NoC components such as routers and cores. This tool can also be used for analyzing 2D and 3D NoC systems. The gem5 [54] is an open-source simulator used for the NoC research, system-level research and processor research. The tool has caches, crossbar switches, filters, DRAM controllers etc to make the NoC unique and fast. Using this tool, homogeneous and heterogeneous multi-core architectures can be designed.

Most of the NoC tools are working in Linux operating system. Few Tools are working in both Linux, windows and MAC operating system. Researchers with C++ knowledge can use many NoC tools in the list provided. Less number of Tools are working with RTL languages such as VHDL, Verilog. AcENoC [48] and AdapNoC [49] uses Xilinx-Microblaze. As Microblaze is a very user friendly software most of the researchers may use it.

## 5. Conclusion and future scope

This paper summarizes the research work carried out by the researchers of Network-on-Chip (NoC). This paper explains the various NoC technologies such as wired NoC, wireless/RF NoC, hybrid NoC, photonic NoC, 2.5D NoC, 3D NoC, 3.5 D NoC which are helpful for the emerging Machine Learning based research work. The security issues of NoC is also addressed. This paper also highlights the current research in Machine Learning-based NoC security. This paper provides various tools available to carry out the research in the NoC area which are popular and efficient nowadays.

### **Data availability statement**

No new data were created or analysed in this study.

## ORCID iDs

K Balamurugan  <https://orcid.org/0000-0001-5226-1665>

S Umamaheswaran  <https://orcid.org/0000-0003-1941-9605>

S Nagarajan  <https://orcid.org/0000-0003-4453-0866>

Lakshmana Rao Namamula  <https://orcid.org/0000-0001-5617-3339>

## References

- [1] Md Yunus N A, Othman M, Mohd Hanapi Z and Lun K Y 2016 Reliability Review of Interconnection Networks *IETE Tech. Rev. (Institution Electron. Telecommun. Eng. India)* **33** 596–606
- [2] Poluri P and Louri A 2016 Shield: a reliable network-on-chip router architecture for chip multiprocessors *IEEE Trans. on Parallel and Distributed Systems* **27** 3058–70
- [3] Owens J D, Dally W J, Ho R, Jayashima D N, Keckler S W and Peh L S 2007 Research challenges for on-chip interconnection networks *IEEE Micro* **1** 96–108
- [4] Wang C, Hu W H and Bagherzadeh N 2013 Scalable load balancing congestion-aware network-on-chip router architecture *J. Comput. Syst. Sci.* **79** 421–39
- [5] Benini L and Bertozzi D 2005 Network-on-chip architectures and design methods *Comput. Digit. Tech. IEE Proc.* **152** 261–72
- [6] Wu Y, Lu C and Chen Y 2016 A survey of routing algorithm for mesh Network-on-Chip *Front. Comput. Sci.* **10** 591–601
- [7] D'Souza S, Soumya J and Chattopadhyay S 2015 Integrated mapping and synthesis techniques for network-on-chip topologies with express channels *ACM Trans. Archit. Code Optim.* **12** 1–26
- [8] Tsai W C, Chu K C, Hu Y H and Chen S J 2013 Non-minimal, turn-model based NoC routing *Microprocess. Microsyst.* **37** 899–914
- [9] Lee H G, Chang N, Ogras U Y and Marculescu R 2007 On-chip communication architecture exploration: a quantitative evaluation of point-to-point, bus, and network-on-chip approaches *ACM Trans. Des. Autom. Electron. Syst.* **12** 1–20
- [10] Bahrami B, Jabraeli Jamali M A and Saeidi S 2018 A novel hierarchical architecture for wireless network-on-chip *J. Parallel Distrib. Comput.* **120** 307–21
- [11] Duraisamy K, Xue Y, Bogdan P and Pande P P 2017 Multicast-aware high-performance wireless network-on-chip architectures *IEEE Trans. Very Large Scale Integr. Syst.* **25** 1126–39
- [12] Rezaei A, Daneshmand M, Safaei F and Zhao D 2015 Hierarchical approach for hybrid wireless network-on-chip in many-core era *Comput. Electr. Eng.* **51** 225–34
- [13] Li C, Ma S, Wang L, Wang Z, Zhao X and Guo Y 2016 DLL: a dynamic latency-aware load-balancing strategy in 2.5D NoC architecture *Proc. of the 34th IEEE Int. Conf. on Computer Design (2016/ICCD)* (<https://doi.org/10.1109/ICCD.2016.7753352>)
- [14] De Paulo V and Ababei C 2009 A framework for 2.5D NoC exploration using homogeneous networks over heterogeneous floorplans *ReConFig'09 - 2009 Int. Conf. on ReConfigurable Computing and FPGAs* (<https://doi.org/10.1109/ReConFig.2009.14>)
- [15] Jerger N E, Kannan A, Li Z and Loh G H 2014 NoC architectures for silicon interposer systems *Microarchitecture (MICRO), 2014 47th Annu. IEEE/ACM Int. Symp.* 458–70
- [16] Ben Ahmed A and Ben Abdallah A 2014 Graceful deadlock-free fault-tolerant routing algorithm for 3D network-on-chip architectures *J. Parallel Distrib. Comput.* **74** 2229–40
- [17] 64 Eghbal A, Yaghini P M, Bagherzadeh N and Khayambashi M 2015 Analytical fault tolerance assessment and metrics for TSV-based 3D network-on-chip *IEEE Trans. Comput.* **64** 3591–604
- [18] Khor C Y and Abdullah M Z 2012 Modelling and analysis of the effect of stacking chips with TSVs in 3D IC package encapsulation process *Maejo Int. J. Sci. Technol.* **6** 159–85
- [19] Kumar M P, Murali S and Veezhinathan K 2012 Network-on-chips on 3-D ICs: Past, present, and future *IETE Tech. Rev. (Institution Electron. Telecommun. Eng. India)* **29**, 318–35
- [20] Bobba S, Gaillardon P E, Seiculescu C, Pavlidis V F and De Micheli G 2013 3.5D integration: a case study *Proc. - IEEE Int. Symp. on Circuits and Systems* (<https://doi.org/10.1109/ISCAS.2013.6572285>)
- [21] Zhao Y, Jia H, Ding J, Zhang L, Fu X and Yang L 2016 Five-port silicon optical router based on Mach–Zehnder optical switches for photonic networks-on-chip *J. Semicond.* **37** 47–53
- [22] Cao R, He Y, Zhu Q, Li J, An S, Zhang Y and Su Y 2019 Multi-channel 28-GHz millimeter-wave signal generation on a silicon photonic chip with automated polarization control *J. Semicond.* **40** 33–9
- [23] Sun C, Chen C H O, Kurian G, Wei L, Miller J, Agarwal A, Peh L S, Stojanovic V et al 2012 DSENT—a tool connecting emerging photonics with electronics for opto-electronic networks-on-chip modeling *6th IEEE/ACM Int. Symp. Networks-on-Chip (2012/NoCS)* 201–10
- [24] Xu K, Chen Y, Okhai T A and Snyman L W 2019 Micro optical sensors based on avalanching silicon light-emitting devices monolithically integrated on chips *Opt. Mater. Express* **9** 3985–97
- [25] Xu K 2021 Silicon electro-optic micro-modulator fabricated in standard CMOS technology as components for all silicon monolithic integrated optoelectronic systems *J. Micromech. Microeng.* **31** 054001
- [26] Ditomaso D, Sikder A, Kodi A and Louri A 2017 Machine learning enabled power-aware network-on-chip design *Proc. of the 2017 Design, Automation and Test in Europe*, (<https://doi.org/10.23919/DAT.2017.7927203>)
- [27] Yin J, Sethumurugan S, Eckert Y, Patel C, Smith A, Morton E, Oskin M, Enright Jerger N and Loh G H 2020 Experiences with ML-driven design: a NoC case study *Proc. - 2020 IEEE Int. Symp. on High Performance Computer Architecture, HPCA 2020* (<https://doi.org/10.1109/HPCA47549.2020.00058>)
- [28] 52 Moradi S and Manohar R 2019 The impact of on-chip communication on memory technologies for neuromorphic systems *J. Phys. D: Appl. Phys.* **52** 11–8
- [29] Kumar A and Talawar B 2018 Machine learning based framework to predict performance evaluation of on-chip networks *2018 11th Int. Conf. Contemp. Comput. IC3 2018* 1–6
- [30] Boraten T and Kodi A 2018 Mitigation of hardware trojan based denial-of-service attack for secure NoCs *J. Parallel Distrib. Comput.* **111** 24–38
- [31] Goertzel K M 2013 Integrated circuit security threats and hardware assurance countermeasures *CrossTalk* **26** 33–8
- [32] Raparti V Y and Pasricha S 2019 Lightweight mitigation of hardware Trojan attacks in NoC-based manycore computing *Proc. - Design Automation Conf.* (<https://doi.org/10.1145/3316781.3317851>)

- [33] Hussain M and Guo H 2017 Packet leak detection on hardware-trojan infected NoCs for MPSoC systems *ACM Int. Conf. Proc. Series* (<https://doi.org/10.1145/3058060.3058061>)
- [34] Moussa N, Nasri F and Tourki R 2014 NOC architecture comparison with network simulator NS2 *Int. J. Eng. Trends Technol.* **13** 340–6
- [35] Catania V, Mineo A, Monteleone S, Palesi M and Patti D 2015 Noxim: an open, extensible and cycle-accurate network on chip simulator *Proc. of the Int. Conf. on Application-Specific Systems, Architectures and Processors* (<https://doi.org/10.1145/3058060.3058061>)
- [36] Lis M, Shim K and Cho M 2010 DARSIM: a parallel cycle-level NoC simulator *Sixth Annu. Work. Model. Benchmarking Simul.*
- [37] Chidella K K, Asaduzzaman A and Almohaimeed A 2019 Impact of non-uniform subnets on the performance of wireless network-on-chip architectures 2019 *IEEE 9th Annual Computing and Communication Workshop and Conf., CCWC 2019* (<https://doi.org/10.1109/CCWC.2019.8666466>)
- [38] ATLAS NOC Simulator (<https://corfu.pucrs.br/redmine/projects/atlas>)
- [39] Chen C, Asoni D E, Barrera D, Danezis G and Perrig A 2015 HORNET: high-speed onion routing at the network layer *Proc. ACM Conf. Comput. Commun. Secur. 2015-October* 1441–1454
- [40] Bertozzi D and Benini L 2004 Xpipes: a network-on-chip architecture for gigascale systems-on-chip *IEEE Circuits Syst. Mag.* **4** 18–31
- [41] Evain S, Diguët J P and Houzet D 2004  $\mu$ Spider: a CAD tool for efficient NoC design *Proc. - Norchip* (<https://doi.org/10.1109/NORCHIP.2004.1423862>)
- [42] Chan J and Parameswaran S 2004 NoCGEN: a template based reuse methodology for networks on chip architecture 17th *International Conference on VLSI Design (Mumbai, India, 9th, January 2004)* (Picastaway, NJ: IEEE) (<https://doi.org/10.1109/ICVD.2004.1261011>)
- [43] Flexnoc (<https://www.artemis.com/flexnoc>)
- [44] Perez F J R and Miguel-Alonso J 2005 INSEE: an interconnection network simulation and evaluation environment *European Conference on Parallel Processing* 1014–23
- [45] Mansour A and Gotze J 2012 An OMNeT++ based network-on-chip simulator for embedded systems *IEEE Asia-Pacific Conf. Circuits Syst. Proc., APCCAS* 364–67
- [46] Palermo G and Silvano C 2004 *PIRATE: A Framework for Power/Performance Exploration of Network-On-Chip Architectures* 1–10
- [47] McKeown M, Lavrov A, Shahrad M, Jackson P J, Fu Y, Balkind J, Nguyen T M, Lim K, Zhou Y, Wentzlaff D et al 2018 Power and energy characterization of an open source 25-core manycore processor *Proc. - Int. Symp. High-Performance Comput. Archit. 2018-Febru* 762–775 (<https://doi.org/10.1109/HPCA.2018.00070>)
- [48] Lotlikar S, Pai V and Gratz P V 2011 AcENoCs: a configurable HW/SW platform for FPGA accelerated NoC emulation 24th *International Conference on VLSI Design* 147–52
- [49] Kamali H M and Hessabi S 2016 AdapNoC: a fast and flexible FPGA-based NoC simulator *FPL 2016 - 26th Int. Conf. Field-Programmable Log. Appl.* (<https://doi.org/10.1109/FPL.2016.7577377>)
- [50] Kamali H M, Azar K Z and Hessabi S 2018 DuCNoC: a high-throughput FPGA-based NoC simulator using dual-clock lightweight router micro-architecture *IEEE Trans. Comput.* 208–221
- [51] Romanov O and Lysenko O 2012 The comparative analysis of the efficiency of regular and pseudo-optimal topologies of networks-on-chip based on Netmaker 2012 *Mediterr. Conf. Embed. Comput. MECO 2012 (Bar, Montenegro)* (Picastway, NJ: IEEE) pp 13–6
- [52] Jain L, Al-Hashimi B and Gaur M 2007 NIRGAM: a simulator for NoC interconnect routing and application modeling *Des. Autom. Test Eur* **4** 16–20
- [53] Raparti V Y, Kapadia N and Pasricha S 2017 ARTEMIS: an aging-aware runtime application mapping framework for 3D NoC-based chip multiprocessors *IEEE Trans. Multi-Scale Comput. Syst.* **3** 72–85
- [54] Lowe-Power J et al 2020 The gem5 Simulator: Version 20.0+\* A new era for the open-source computer architecture simulator arXiv:2007.03152