Computational Methods for Differential Equations
http://cmde.tabrizu.ac.ir
Vol. 7, No. 4 (Special Issue), 2019, pp. 511-520

A fuzzy-based routing scheme for network-on-chip with honeycomb topology

Mohammad Alaei*
Computer Engineering Department, Faculty of Engineering,
Vali-e-Asr University, Rafsanjan, Iran.
E-mail: alaeim@vru.ac.ir

Fahimeh Yazdanpanah
Computer Engineering Department, Faculty of Engineering,
Vali-e-Asr University, Rafsanjan, Iran.
E-mail: yazdanpanahf@vru.ac.ir

Abstract Network-on-chip (NoC) paradigm, which is based on a modular packet-switched mechanism, effectively addresses many of the on-chip communication challenges such as wiring complexity, communication latency, and bandwidth of many-core systems. In designing an efficient NoC, topology and routing algorithm are the most important challenging issues that have significant impact on area, latency and power consumption. The goal of this paper is designing a fuzzy-based routing algorithm for a NoC architecture with honeycomb topology. The proposed algorithm is a livelock and deadlock free routing algorithm based on fuzzy logic for hexagonal zones with flat triple coordinate system. The analysis of simulation results demonstrates that the proposed algorithm, provides higher performance in terms of latency, power consumption, throughput and area than a traditional fuzzy-based routing algorithm for mesh-based NoC architectures. Comparing to a non-fuzzy routing algorithm for honeycomb NoCs, the proposed scheme performs faster with higher throughput with a negligible area and power consumption overhead.

Keywords. Network-on-chip, Honeycomb topology, Fuzzy-based controller.

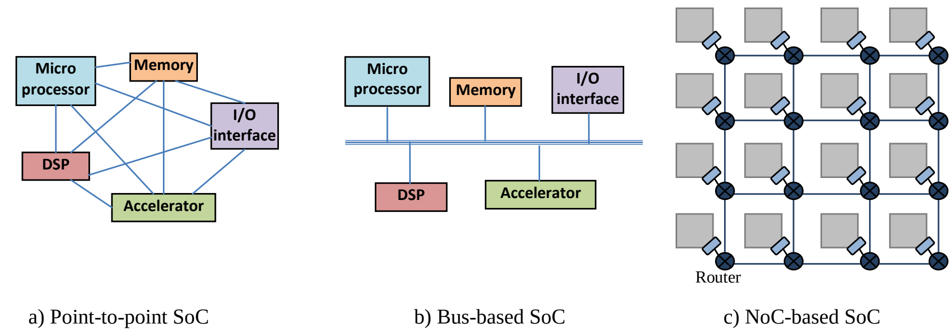
2010 Mathematics Subject Classification. 68M10, 68M14.

1. Introduction

Traditionally, the interconnection structure of a system-on-chip (SoC) was based on dedicated point-to-point wires or shared buses. The point-to-point interconnection (Figure 1-a) is only applicable for systems with a small number of cores; because as the number of cores grows, the number of connecting wires dramatically increases. Shared buses (Figure 1-b) are more scalable, but using bus interconnection, only one communication is allowed at one time. Network-on-chip architectures (NoCs) [1,2], on the other hand, provide a higher level of communication parallelism and even higher scalability, for communicating between processing elements using routers, network interfaces and links (Figure 1-c).

* Corresponding author.

FIGURE 1. Different interconnections for SoC architectures.



Topology and routing algorithms are the most important challenging issues that have significant impact on NoC performance. Topology defines how NoC nodes are placed and connected and greatly affects the speed, energy efficiency, and circuit area of many-core processor arrays. Due to its simplicity scalability and regularity, 2D mesh is mostly used for existing on-chip networks. However, efficiently mapping applications can be a challenge for the cases that require communication between processors that are not adjacent on the 2D mesh [3]. On the other side, the researchers of [3-8] have shown that a wired NoC with honeycomb topology provides better performance characteristics, in terms of energy consumption, area, delay and throughput, compared to a wired mesh NoC. With high regularity, symmetry and scalability, honeycomb-based NoCs provide high performance and energy-efficient communication structures.

In NoCs, routing algorithms are used to determine the path of a packet from the source to the destination. Implementation complexity and performance requirements are the main parameters affecting the choice of routing strategy. Routing protocols have significant impacts on the latency and power consumption of NoC-based systems [9-12]. These algorithms are classified as deterministic and adaptive. The implementation of deterministic routing algorithms is simple but they are not able to balance the load across the links for various packet distributions of real applications. Adaptive routing algorithms are proposed to address these limitations and improve the network performance by better distributing load across links according to network traffic.

Fuzzy controllers are widely used in many different fields nowadays, ranging from control applications, robotics, image and speech processing to biological and medical systems as well as in Ad Hoc, wireless and interconnection networks. The fuzzy system is employed to estimate the latency of each candidate direction [13-15]. Specifically, fuzzy logic control is used to build a simple, generic, and efficient nonlinear control law that dynamically calculates the link cost. Fuzzy systems are commonly used to improve performance or to resolve ambiguities in complex problems that are difficult to tackle mathematically. Since control problems in communication systems become increasingly complex (due to their characteristics of having multiple performance criteria), the use of fuzzy and adaptive algorithms is indeed well suited to increase performance [13].

The aim of this paper is designing a fuzzy-based routing algorithm for NoCs with honeycomb topology. The proposed algorithm is an adaptive routing algorithm based on fuzzy logic for hexagonal zones with flat triple coordinate system. The evaluations demonstrate that the proposed scheme provides high performance.

The remainder of this paper is organized as follows. The proposed routing scheme, a fuzzy-based routing algorithm for honeycomb-based NoCs, is expounded by section 2. Then, section 3 presents the methodology and performance evaluation of the proposed scheme including results, comparisons and discussions. Finally, the paper is concluded by section 4.

2. The proposed routing algorithm

The proposed routing algorithm includes three steps: 1) constructing a honeycomb-based structure, 2) applying turn restrictions in hexagonal zones, 3) congestion control using fuzzy system. These steps are explained in the following sections.

2.1. Constructing a honeycomb-based structure.

The proposed routing algorithm is based on a planar 3-axes coordinate system [3,5]. The X, Y and Z axes start from the center of the network and divide the topology into three regions, hence, we have (x,y,z) coordinate system with positive and negative values. The first step of the proposed routing algorithm is to construct a honeycomb-based structure and determine triple coordinates to each node. In this structure, each zone consists of six nodes.

Using the approach presented in [3], each node is placed to form a honeycomb topology, considering following conditions:

c1) $-t+1 \leq x, y, z \ \leq t$

c2) $\mathrm{x+y+z} = 2 \ \ or \ \ \mathrm{x+y+z} = 1$

c3) There is a link between two nodes $(\mathrm{u}_1, \mathrm{v}_1, \mathrm{w}_1)$ and $(\mathrm{u}_2, \mathrm{v}_2, \mathrm{w}_2)$ if

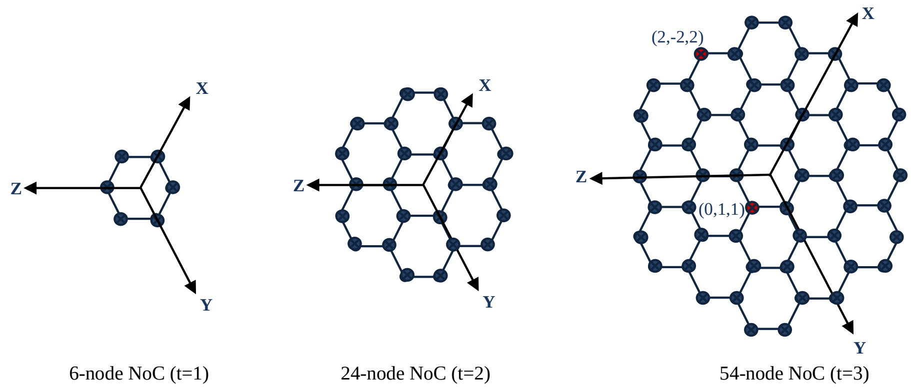
$$| u_1 - u_2| + | v_1 - v_2| + | w_1 - w_2| = 1$$

As Figure 2 illustrates, a honeycomb-based NoC consists of 4-port routers. Each router is connected to its corresponding core and at most three adjacent routers.

2.2. Turn restrictions in hexagonal zones.

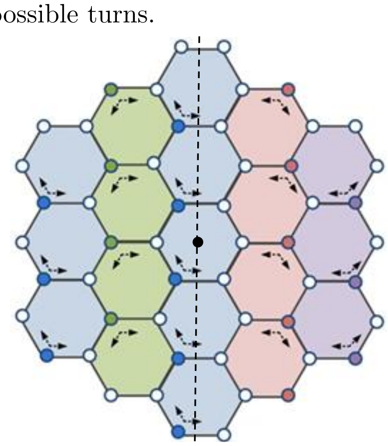
The second step is to applying turn restrictions to each zone because the proposed routing is based on restricted turns in order to prevent livelocks according to the method of [5]. It allows us to define bidirectional turn restrictions locally within a hexagonal zone. As segments are independent, we are free to place turn restrictions within a zone independently from other zones. As Figure 3 shows, one out of six possible turns is disabled in clockwise or non-clockwise turn within a zone. In the

FIGURE 2. The output of first step of the proposed algorithm: constructing honeycomb topology [3].



proposed routing algorithm, we start from the hexagon at the center of topology; one of the six turns is restricted in this zone. Then, considering a vertical line crossing the center point of the topology (the dashed line in Figure 3), all the zones along this line have the same turn restriction. The restricted turns at right side and left side are symmetric to this line, and all the zones in a column, have the same turn restriction. Employing this strategy in the proposed algorithm effectively ignores the livelocks. In addition, in order to prevent the deadlocks, two virtual channels are adapted for each input port. These techniques decrease flit drops and increase the number of received flits per cycles (i.e., the network throughput).

FIGURE 3. The second step of the proposed algorithm: avoidance one out of six possible turns.



2.3. **Congestion control using fuzzy system.**

The third step of the proposed routing algorithm is congestion control using fuzzy system. The fuzzy inference system (FIS) is employed to determine the cost of next directions. At each router, the output direction with the lowest cost is chosen to deliver the flit. The use of fuzzy-logic algorithms in the path decision making leads to a systematic comparison among the candidates of output ports. The routing unit of the router computes the routing algorithm. When there are two possible paths, the routing unit obtains the congestion information of the two routers. Then, FIS determines a cost for each possible link. In the proposed scheme, the cost is determined based on two metrics: 1) the number of stored flits in the buffer of corresponding input port of the next router (input port congestion), and 2) the number of stored flits in the buffers of the next router (router congestion). Figure 4 illustrates the fuzzy unit of the proposed algorithm.

In the fuzzy inference system, there are four units:

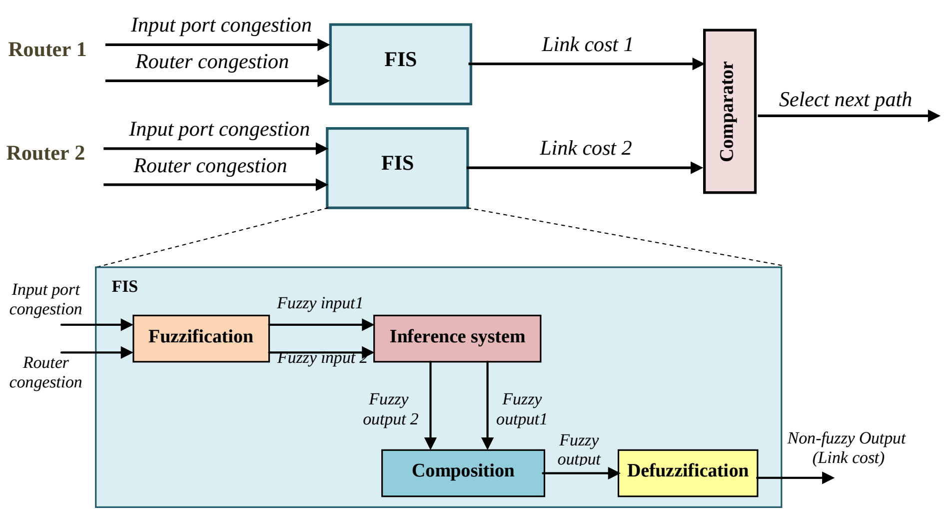
a) *Fuzzification* is the process of converting non-fuzzy input values to fuzzy values.

b) *Inference system* uses the collection of linguistic rules to convert the fuzzy inputs into fuzzy outputs.

c) *Composition unit* combines the fuzzy outputs of all rules together to obtain a single fuzzy output.

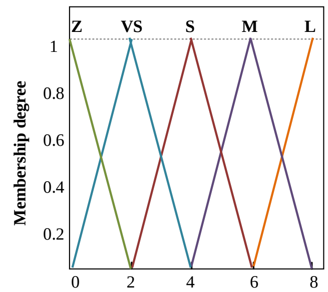
d) *Defuzzification* converts the fuzzy output into non-fuzzy output value.

Figure 4. The third step of the proposed algorithm: the fuzzy system.

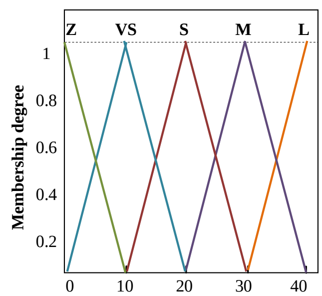


Fuzzification and inference system produce output fuzzy values using membership functions, as it is illustrated in Figure 5. We select the fuzzy set with five states

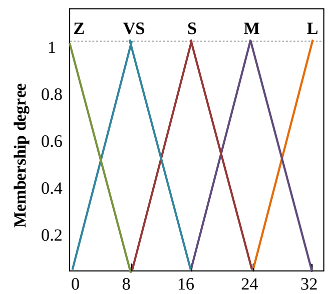
FIGURE 5. Membership functions.



a) Number of flits in each input port



b) Number of flits in a mesh-based router



c) Number of flits in a honeycomb-based router

as zero (Z), very small (VS), small (S), medium (M), and large (L). The triangular membership function defines how each point in the input space is mapped to a membership degree between 0 and 1, with eight flits per buffer (Figure 5-a). Figure 5-b depicts the membership function of congestion for a mesh-based router, which has four global ports and one local port. The membership function of congestion of a router in honeycomb topology is shown in Figure 5-c. In this case the membership function is between 0 and 32; it means for each router the number of stored flits in the all buffers of the router is between 0 and 32.

Fuzzy roles with two fuzzy inputs and one fuzzy output for the number of stores flits are presented in Table 1. The table provides various ranges of the output for different ranges of inputs. Filling a data table with fuzzy attributes is subjective. The table is filled based on the basic knowledge on the impact of each metric in the overall performance of the network.

3. Results and evaluation of the proposed algorithm

In this section, performance of the proposed routing algorithm is evaluated and is compared with three schemes: a non-fuzzy routing algorithm for mesh-based NoCs [13], a fuzzy routing algorithm for mesh-based NoCs [13], a non-fuzzy routing algorithm for honeycomb-based NoCs. This scheme is almost similar to the proposed algorithm without the third step (fuzzy-based congestion control). Table 2 shows

Table 1. Fuzzy roles for the number of stored flits

Input Port \ Router	Z	V	S	M	L
Zero (Z)	Z	Z	V	S	M
Very small (V)	Z	V	V	S	M
Small (S)	V	V	S	M	M
Medium (M)	S	S	M	L	L
Large (L)	M	M	L	L	L

the simulation parameters. For simulating the intended schemes and obtaining the results, we have used Noxim [16], as a cycle accurate NoC simulation framework, and also, Orion, as a power consumption analyzer. The synthetic traffic patterns are *Uniform* and *Hotspot*, and the real traffic patterns are *VoPD (Video Object Plane Decoder)* and *MPEG (Motion Picture Experts Group)* applications.

Table 2. Simulation parameters

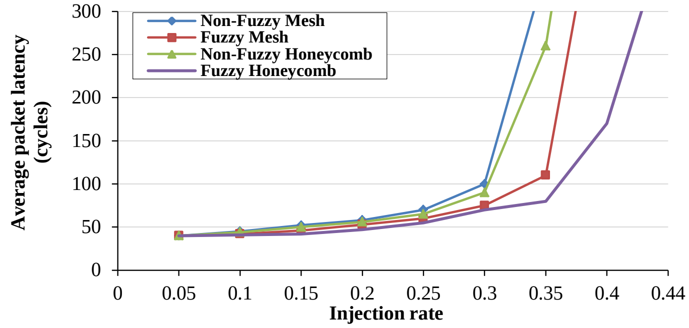
Parameter	**Value**
Number of cores	256
Buffer depth	8 flits
Packet size	1 flit
Flit size	64 bits
Frequency	1GHz
Switching method	Wormhole
Warm-up time	1000 cycles
Number of executions	40000 cycles

Figure 6 shows the average packet latency for different injection rates under synthetic traffic patterns. As it can be seen, fuzzy-based NoC routing performs faster than non-fuzzy routing. Also, the NoCs with honeycomb topology have less latency than the NoCs with mesh topology. Therefore, the proposed routing algorithm for NoCs with honeycomb topology has the least latency.

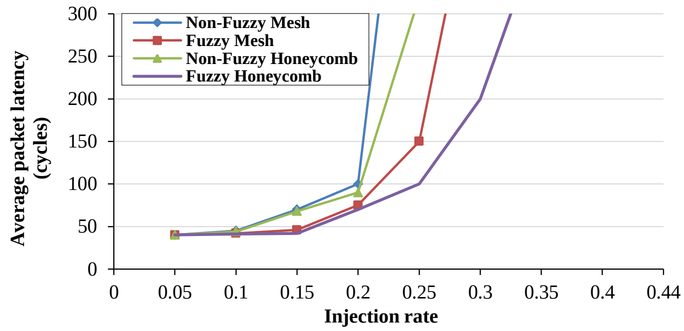
Figure 7 shows average packet latency for real traffic patterns. As the figure illustrates, similar to the results of the synthetic packet distribution, the proposed routing algorithm has the least latency among the intended schemes.

Figure 8 demonstrates throughput, latency, network area and power consumption of the intended schemes in normalized form. These results are obtained as the average for the mentioned traffic patterns with injection rate of 0.2 packets per cycle. The baseline for calculating the normalized numbers is non-fuzzy routing algorithm for the mesh architecture. As Figure 8 shows, the proposed routing algorithm offers higher performance in term of speed and throughput than the other schemes, because of controlling the congestions and balancing the network load by fuzzy system. Moreover, employing the proposed algorithm, we have smaller NoC with less power consumption comparing to the non-fuzzy and fuzzy-based routing algorithms on mesh topology. Off course, the proposed routing method has a negligible penalty in cost and area comparing to the non-fuzzy routing algorithm for honeycomb-based NoC architectures.

Figure 6. Latency evaluation with synthetic traffic pattern.

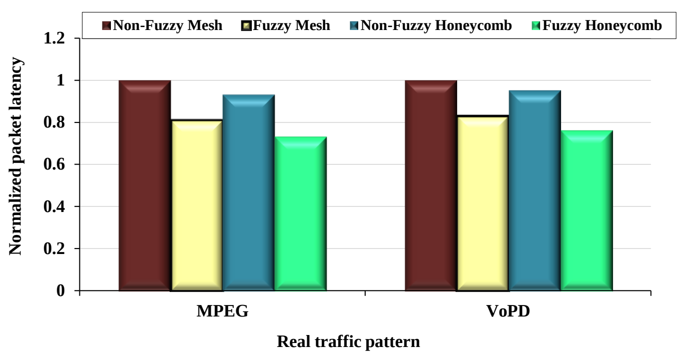


a) Uniform traffic pattern



b) Hotspot traffic pattern

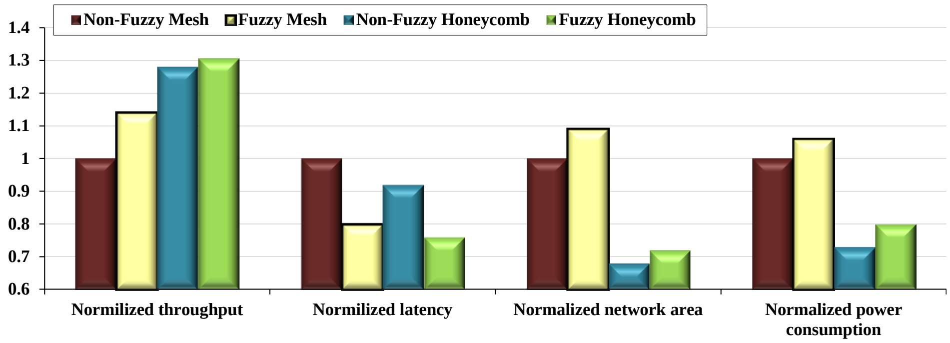
Figure 7. Latency evaluation with real traffic pattern.



4. Conclusion

In this paper, a fuzzy-based adaptive routing algorithm for NoC architectures with honeycomb topology was proposed. In the proposed scheme, congestion and network

FIGURE 8. Performance evaluation.



traffic is effectively controlled by fuzzy system. Moreover, the proposed routing algorithm is livelock-free because one out of six possible turns is disabled within a zone. Analysis the simulation results demonstrated that the proposed algorithm is more efficient than a fuzzy-based routing algorithm for mesh-based NoC architectures in terms of throughput, latency, power consumption and area. Moreover, the proposed algorithm outperforms a non-fuzzy algorithm for honeycomb-based NoCs in throughput and latency parameters. The fuzzy-based control unit for routing is at the expense of insignificant power consumption and area.

References

[1] S. Kundu and S. Chattopadhyay, *Network-on-chip: the next generation of system-on-chip integration*, 1st Edition, CRC Press, 2017.

[2] S. Ma, L. Huang, M. Lai, and W. Shi, *Networks-on-chip, from implementations to programming paradigms*, 1st Edition, Morgan Kaufmann, 2014.

[3] A. W. Yin, T. C. Xu, P. Liljeberg, and H. Tenhunen, *Explorations of honeycomb topologies for network-on-chip*, Sixth IFIP International Conference on Network and Parallel Computing, Gold Coast, QLD, (2009), 73–79.

[4] Y. Alexander, C. Nan, L. Pasi, and T. Hannu, *Comparison of mesh and honeycomb network-on-chip architectures*, Proceedings of ICIEA, (2012), 1716–1720.

[5] F. Yazdanpanah, R. AfsharMazayejani, M. Alaei, A. Rezaei, and M. Daneshtalab, *An energy-efficient partition-based XYZ-planar routing algorithm for a wireless network-on-chip*, Journal of Supercomputing, *75*(2) (2019), 837–861.

[6] R. Afsharmazayejani, F. Yazdanpanah, A. Rezaei, M. Alaei, and M. Daneshtalab, *HoneyWiN: novel honeycomb-based wireless NoC architecture in many-core era*, Proceedings of ARC, *10824* (2018), 304–316.

[7] Z. Xiao and B. Baas, *A hexagonal processor and interconnect topology for many-core architecture with dense on-chip networks*, In: Burg A., Cokun A., Guthaus M., Katkoori S., Reis R. (eds) VLSI-SoC: From Algorithms to Circuits and System-on-Chip Design. VLSI-SoC 2012. IFIP Advances in Information and Communication Technology, Springer, Berlin, Heidelberg, (2012).

[8] M. Alaei and F. Yazdanpanah, *H2WNoC: A honeycomb hardware-efficient wireless network-on-chip architecture*, Nano Communication Networks, *19* (2019), 119–133.

[9] M. Daneshtalab, M. Ebrahimi, P. Liljeberg, J. Plosila, and H. Tenhunen, *Memory efficient on-chip network with adaptive interfaces*, IEEE T. Comput. Aid. D., *31*(1) (2012), 146–159.

[10] M. Palesi and M. Daneshtalab (Eds), *Routing algorithm in networks-on-chip*, Springer-Verlag, New York, 2014.

[11] Z. Ghaderi, A. Alqahtani, and N. Bagherzadeh, *AROMa: aging-aware deadlock-free adaptive routing algorithm and online monitoring in 3D NoCs*, IEEE T.Parall. Distr., *29*(4) (2018), 772–788.

[12] E. Chang, H. Hsin, C. Chao, S. Lin, and A. Wu, *Regional ACO-based cascaded adaptive routing for traffic balancing in mesh-based network-on-chip systems*, IEEE T. Comput., *64*(3) (2015), 868–875.

[13] M. Ebrahimi, H. Tenhunen, and M. Dehyadegari, *Fuzzy-based adaptive routing algorithm for networks-on-chip*, J. Syst. Architect., *59*(7) (2013), 516–527.

[14] K. Tatas and C. Chrysostomou, *Hardware implementation of dynamic fuzzy logic based routing in network-on-chip*, Microprocess. Microsy., *52* (2017), 80–88.

[15] H. Phan and X. Tran, *Fuzzy-logic based low power solution for network-on-chip architectures*, Proceedings of ATC, (2016), 334–338.

[16] V. Catania, A. Mineo, S. Monteleone, M. Palesi, and D. Patti, *Cycle-accurate network on chip simulation with noxim*, ACM T. Model. Comput. S., *27*(1) (2016), 1–25.