

Affine-NoC: Multi-ring NoCs exploiting long physical links

Enrique Vallejo
 Computer Science Dept.
 Universidad de Cantabria, Spain
 Santander, Spain
 enrique.vallejo@unican.es

Evangelos Mageiropoulos, Nikolaos Chrysos, Manolis G.H. Katevenis
 Computer Architecture and VLSI Systems (CARV) Laboratory,
 Foundation for Research and Technology – Hellas (FORTH)
 Heraklion, Greece
 {emageir, nchrysos, kateveni}@ics.forth.gr

Abstract—Routerless multi-ring proposals are low-cost NoCs that employ multiple independent rings, avoiding any crossbars or arbitration mechanisms. Their complexity lies on the selection of the set of rings to connect all the nodes. Previous proposals result in unbalanced designs with high hop counts.

This work introduces Affine-NoC, a balanced routerless NoC based on a novel arrangement of rings derived from the Affine Plane. Affine-NoC exploits express channels to connect distant processing elements, allowing for a completely balanced layout of the set of rings and a reduction in average distance and diameter without sacrificing bisection bandwidth.

Analysis shows that Affine-NoC presents a balanced design with a low number of rings per node and reduced complexity, similar cost to previous proposals in terms of aggregated link length, while it simplifies the multiplex units and reduces the hop count. Simulation results show that Affine-NoC reduces hop count and average latency by 76% and 20.5% respectively compared to previous designs, it reduces deflections by 27% and avoids unfairness, making it a feasible alternative for multi-ring routerless NoCs.

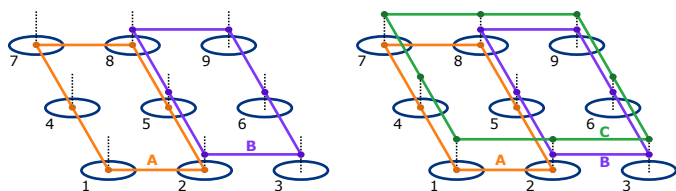
Index Terms—Affine-NoC, multi-ring, routerless.

I. INTRODUCTION

Routerless multi-ring networks [1], [2] are interconnect architectures that rely on a collection of partially overlapping independent rings, as depicted in Fig. 1. Because of the simplicity of the ring implementations, these architectures present compelling low-area and low-power characteristics. These rings are selected such that any pair of nodes is connected by at least one ring in the NoC, and packets never leave the selected ring until being consumed.

The complexity of the design lies on selecting a competitive set of rings, while bounding the number of rings per node. Previous designs, algorithmic [2], [3] or machine generated [1], [4], [5], consider rings as consecutive segments always connecting adjacent nodes in X or Y. However, as we identify in this paper, this approach is unnecessarily limiting. Indeed,

This work is supported by RED-SEA, a project that receives funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 955776. The JU receives support from the European Union’s Horizon 2020 research and innovation programme of France, Greece, Germany, Spain (PCI2021-121934 and PCI2021-121976), Italy, Switzerland. This work is also supported by grants TED2021-131176B-I00 and PID2022-136454NB-C21 funded by MICIU/AEI/ 10.13039/501100011033 and by ERDF/EU. Work developed while E. Vallejo was a visitor in FORTH.



(a) NoC with 2 independent rings, not completely connected: no ring connects nodes 1 and 9 (b) NoC with 3 independent rings, completely connected.

Fig. 1: Several multi-ring examples.

with such approach neighbor nodes share multiple rings (e.g. both rings A and C connect nodes 1 and 2 in Fig 1b), it generates short and long rings (leading to load imbalance and unfairness) and it restricts the potential bandwidth and distance properties achieved by using multiple rings.

Connecting ring segments to distant (not adjacent) nodes can reduce the number of unnecessary connections between neighbors and reducing the implementation overhead, in particular the output multiplexor unit. However, a design with such long channels (denoted *express* [6] or *ruche* [7] channels) is not trivial, since it must deal with two issues: first, the total wire cost must remain bounded; second, the design space (which is already huge [1], [4], [5]) grows exponentially, making the ring-selection process even more difficult.

This paper introduces Affine-NoC, which exploits long physical links for the systematic design of balanced multi-ring NoCs. Affine-NoC combines multi-ring routerless NoCs, express channels and concentration to build a completely balanced configuration of rings based on the geometrical construction of the Affine plane.

In particular, the main contributions of the paper are:

- Affine-NoC, a multi-ring NoC that exploits long physical channels to produce a balanced design and reduce implementation costs.
- A topological evaluation of Affine-NoC, which shows that it removes imbalance, has lower hop count and less hardware requirements than previous multi-ring approaches, for a similar aggregate path length.
- A performance evaluation of Affine-NoC, which shows latency reductions of 20.7%, reduced deflection count and

improved fairness.

II. BACKGROUND AND MOTIVATION

A. Wiring availability, long channels and NoC topologies

Current VLSI technology provides a large number of metalization layers, for example 16 metal layers in Intel 4 node [8]. Considering the pitch size, this yields a huge number of wires available for the chip interconnect.

To exploit this ample wiring, many topologies employ long physical links to connect distant parts of the network, both increasing bandwidth and reducing average distance and latency. Express channels [6] connect distant nodes in the same row or column. Increasing the radix moderately, they have been proven to be a competitive alternative for NoC topologies [9] and have been proposed for large NoCs [7]. Similarly, SMART [10] introduces a multi-hop bypass mechanisms that connects distant routers (in a mesh) in a single cycle.

B. Multi-ring NoC organization

Traditional register-insertion rings implement an insertion buffer. Injection occurs when a free slot reaches the node. The buffer transiently stores data received in the ring during a multiflit packet injection. Afterwards, the buffer progressively drains when no traffic is received. These rings require low power and area: they do not employ crossbars or buffers, and there is almost no control logic since they do not have flow control. However, they do not scale to large node counts.

Routerless multi-ring NoCs [1], [2] exploit the wiring availability by employing multiple independent register-insertion rings per node (SoC tile). They employ source routing: the source injects into one of the rings that connect to the destination, and the packet never leaves such ring. The design in [2] employs few insertion buffers, shared between all the rings in a node, denoted *extension buffers*, EXB.

The design complexity of routerless multi-ring NoCs lies on the selection of the set of rings, which needs to be as small as possible, connect any pair of nodes at least once, and be as balanced as possible. This set of rings can be seen as the topology of the routerless NoC. Systematic constructions have been presented before, for example in REC [2] and Onion [3]. However, these proposals fail to produce balanced designs, relying on very short and very long rings. This causes traffic imbalance and, therefore, the maximum throughput is limited.

All these ring proposals do not implement flow control. Thus, they rely on *deflection*: if the destination node is not available, the packet is deflected and makes another turn.

C. Limitations of routerless multi-ring NoCs

This section lists the main limitations of previous multi-ring designs, with a particular focus on systematic constructions of the topology [2], [3]. We consider a 16×16 REC [2] or Onion [3] design as an example:

Multiple connections between nodes: Each ring partially overlaps with many other ones, meaning that neighbor nodes have multiple unnecessary connections; up to $N = 16$.

Large ring count per node and HW complexity: $N = 16$ rings cross each node, making its architecture quite complex. In particular, the $N:1$ (de)multiplexers employed for injection/ejection grow in complexity and delay with the node count, limiting the scalability of the design.

Long rings: Required to connect distant nodes; the largest ring lays in the NoC perimeter, with 28 hops ($4 \times (N - 1)$).

Imbalance: The proposals require both short (only 4 hops) and very long rings, resulting in significant length imbalance and, therefore, reduced ring utilization.

Buffer requirements: With shared EXBs, an empty EXB is required to inject multiflit packets; if data are received during injection, flits are stored in the EXB. An EXB must completely drain before it can be reused. Therefore, in practice several EXBs per injector are required for continuous transmission.

D. The Affine Plane

Affine-NoC design is based on the Finite Affine Plane. A Finite Affine Plane is defined by a set n^2 points (nodes) and $n^2 + n$ lines, each line containing n points and each point belonging to $n+1$ lines. Any two lines are *incident* or *parallel*. *Incident* lines share a single point. *Parallel* lines have no points in common. The $n^2 + n$ lines are arranged into $n + 1$ sets of n *parallel* lines, each set with a different *slope*, including horizontal (slope=0), vertical (slope=inf) and diagonal lines.

Lines wrap-around the borders of the $n \times n$ plane. For example, with $n = 5$ the Affine Plane contains 5 horizontal and 5 vertical lines forming a 5×5 orthogonal grid, plus other 4 sets of diagonal lines, each set with 5 lines of 5 points. Each point belongs to 6 different lines, with different slopes.

A Finite Affine Plane of order n exists when the order n is a prime power (2, 3, 4, 5, 7, 8, 9, 11, ...). Affine Planes are well known, can be build algorithmically, and are implemented in available free mathematical software [11].

III. AFFINE-NOC

A. Overview

Affine-NoC employs multiple rings derived from the Affine Plane (AP): Each AP point is mapped to one NoC node and each AP line to one ring, defined by the set of points it connects. With the AP design only $\lambda = 1$ ring connects any pair of nodes, minimizing overheads, and all rings have a similar number of nodes.

When the desired NoC size does not match an existing Affine Plane, slightly unbalanced designs are derived from a larger Affine Plane, removing certain nodes and rings. The set of rings built in this manner defines a *logical* topology.

This logical topology is then mapped to the physical layout generating a physical pattern. Express links in this physical topology connects distant nodes, avoiding unnecessary connections to neighbor nodes. The resulting network presents an unnecessarily large Bisection BandWidth (BBW) and requires a large amount of wiring compared to previous designs. Therefore, for a competitive alternative we introduce designs with concentration, this is, connecting multiple processing tiles

to the same NoC node. In particular, our design employs concentration $c = 4$, based on the measured BBW.

B. Construction of the base logical topology

We consider a NoC with $X \times Y$ nodes (where each node services $c = 4$ injectors and ejectors), with $X \geq Y$. We generate the logical topology in two steps:

1. Generate the Affine plane: We employ an Affine Plane of order $n \geq X$, with n a prime power as low as possible. This Affine plane comprises n^2 points and $n + 1$ sets of n parallel lines, of which n lines are horizontal and n are vertical. The selected Affine Plane is completely balanced (all lines contain exactly n elements), but when $n > X$ or $n > Y$ it contains more rows or columns than the desired configuration. The points (x, y) in each line are defined by $y = m \cdot x + b$, where m represents the slope ($m = 0$ and $m = \text{inf}$ for horizontal/vertical lines; $1 \leq m < n$ for diagonal lines) and where operations are modulo n ¹. We employ the implementation in SAGE [11] (an open-source mathematical tool) to generate the Affine Plane.

2. Remove rows and columns: to obtain the target *logical configuration*, we remove $n - X$ vertical lines (columns) and $n - Y$ horizontal lines (rows). Removing a line consists on consecutively removing all nodes from the line. Removing a node shrinks the $n + 1$ lines that include it. When any lines are removed, the resulting configuration is no longer completely balanced, but the imbalance ratio is small.

For example, a rectangular Affine-NoC with 10×9 nodes (servicing $90 \times 4 = 360$ processing tiles) is built from an Affine Plane of order $n = 11$, since $10 = 5 \cdot 2$ is not a prime power. This AP has 121 nodes; 11 horizontal, 11 vertical and 10 sets of 11 diagonal lines. Next, we remove $11 - 10 = 1$ vertical line (column) and $11 - 9 = 2$ horizontal lines (rows), shortening the length (node count) of the affected lines by one unit in each step. The resulting arrangement contains 10 vertical lines of size 9 (9 nodes), 9 horizontal lines of size 10, and 110 diagonal lines, 20 of size 9 and 90 of size 8. The max/min imbalance ratio is only $10/8=1.25$. Note that for square designs with side equal to a prime power the design is completely balanced (ratio 1).

C. Physical topology mapping

Step 3. ring mapping maps each of the *lines* to a NoC *ring*, by determining the specific sequence of nodes to travel in each ring, i.e. the connection order. The goal is to minimize the accumulated physical length of the links of the resulting ring. Since ring nodes are not necessarily adjacent, some express channels will connect nodes in different rows and columns (diagonal connections); those are physically implemented using horizontal and vertical links (Manhattan distance).

For a ring that spans x columns and y rows, its minimum accumulated length is $2 \cdot ((x - 1) + (y - 1))$. For each ring, we employ three consecutive strategies to obtain a layout with

¹The Finite Affine Plane is actually defined over a Finite Field \mathbb{F} , so wrap-around diagonal calculation is slightly more complex than a modulo operation when n is not a prime; it is omitted for simplicity.

TABLE I: Buffer utilization with EXBs in the baseline and with insertion buffers in Affine-NoC.

NoC size (tiles)	EXBs/tile (multi-ring baseline)	Buffers/node (Affine $c=4$)	Buffers/tile (Affine $c=4$)
64	2-4	5	5/4=1.25
256	2-4	9	9/4=2.25
1024	2-4	17	17/4=4.25

the minimum accumulated link length, selecting the first one that generates the optimal link length:

- A folded-ring layout with links of physical length 2. It is generated by selecting odd nodes first in ascending order and even nodes next in descending order. It is always optimal for horizontal and vertical rings.
- A heuristic folding, which traverses the first half of the line nodes in ascending order, followed by the second half of the line in descending order.
- A Mixed Integer Linear Programming (MILP) solver for the Travelling salesman problem in the points of the line (available in [11]). While this is NP-complete, it is solved in a few seconds for rings up to 20 nodes, and in feasible time for ring sizes up to 30 nodes.

Fig. 2 presents the final layout of a 5×5 Affine-NoC, with the rings that pass through each one of the lower-row nodes. Rings with the same color have the same slope and no nodes in common. The figure depicts all the rings in the NoC, except for the horizontal rings in the four upper rows.

D. Affine-NoC node architecture

This section presents the relevant details of the architecture of each node.

Concentration: The topology generated by the AP contains $n^2 + n$ rings for a NoC with n^2 nodes (if no nodes are removed in step 2). The topological analysis in Section IV-B2 shows that this configuration is over-dimensioned. Therefore, in Affine-NoC each node services four different NoC tiles, i.e., we connect $c = 4$ different injectors/ejectors to each node.

This concentration reduces the node count by a factor of 4 with respect to the NoC tile count, significantly reducing the number of rings per node. This also drastically reduces the size of the (de)multiplexers required to inject/eject traffic to/from the appropriate ring.

Insertion buffers: Section II-C discusses the problem of using few EXBs. The concentration in Affine-NoC drastically reduces the number of rings per node, making a design based on per-ring private buffers affordable. Indeed, Table I shows how an Affine-NoC design with private buffers requires similar or less buffer area than an EXB-based approach for designs up to 1024 cores. Note that Affine-NoC employs one buffer per ring in the node, but there are $c = 4$ tiles/node.

Connection of multiple injectors and ejectors: Affine-NoC introduces the problem of how to connect these four tiles to each other. In our model, each injection demultiplexer is extended with additional outputs that connects to each other ejection multiplexers. Packets are sent directly to available

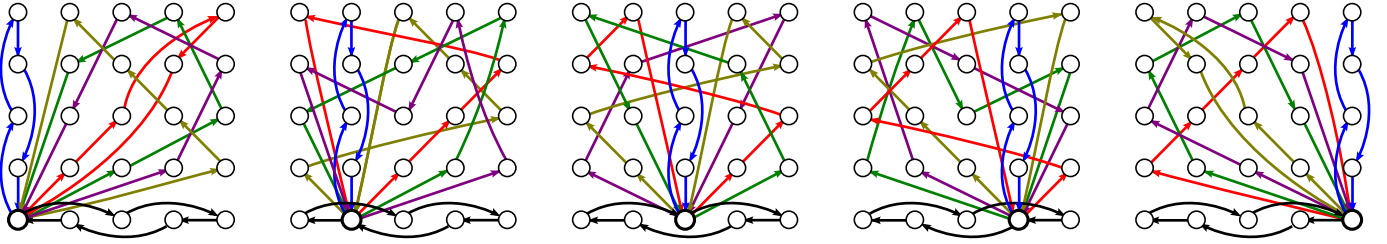


Fig. 2: Rings passing through each of the nodes in the lowest row in a 5×5 Affine-NoC NoC. Rings with the same color are generated from parallel lines in the Affine plane. Note that logically diagonal lines are physically implemented as consecutive horizontal and vertical links.

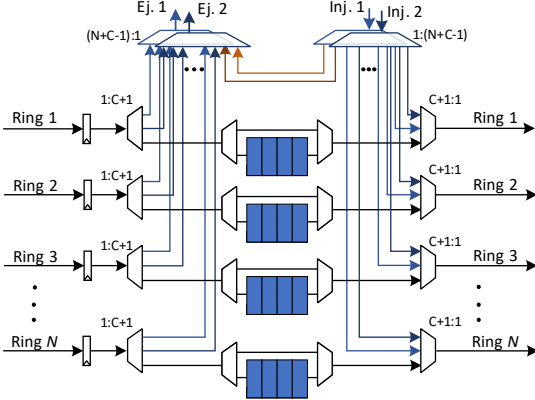


Fig. 3: Affine-NoC architecture, depicted with concentration $c=2$ (instead of $c=4$) for clarity.

outputs. This minimizes delay, but requires $c-1 = 3$ additional ports per MUX/DEMUX. Fig. 3 shows the node organization.

IV. EVALUATION

A. Evaluation Methodology

1) *Evaluated models*: We employ the following routerless configurations in our evaluation:

- *Affine* and *Affine-C4*: The Affine-NoC mechanism introduced in Sect. III, without and with concentration $c = 4$.
- *REC* and *REC-C2*: The configuration in [2], also considering concentration $c = 2$ based on the BBW results.
- *Onion*: The Onion configuration from [3].

We also compare to traditional router-based NoCs:

- *Mesh*: A traditional mesh with 5 ports per router.
- *Mesh-opt*: A mesh with express channels in both X and Y, increasing port count to 9. The express channel length is selected to provide full BBW, as analyzed in Fig. 4a.

2) *Evaluation infrastructure*: Topological analysis results are obtained using custom scripts. Performance results are obtained using BST-Booksim [12]. We extend the simulator with a routerless multi-ring model, which accepts custom topology description models with or without express channels. We run the simulations for 50 000 cycles after a similar warmup. We employ synthetic traffic and packets with size 1 or 5. In the multi-ring models, we consider a single cycle per hop delay and buffers equal to the maximum-size packet.

The HPC_{Max} parameter (from [10]) determines how far traffic travels per cycle. It is used to determine the cycles required to traverse express channels in Affine-NoC: We calculate the length L of each link using the Manhattan distance (in terms of tile size and taking into account the impact of concentration). When $L > HPC_{Max}$, link traversal requires multiple cycles, employing intermediate registers.

B. Topological results

1) *Complexity of the Affine-NoC layout calculation*: We employ a python implementation to generate an $n \times n$ square Affine-NoC, using SAGE for Affine Plane generation and for solving the Travelling Salesman Problem in the ring mapping phase. Small designs are generated in negligible time: a 16×16 configuration (which accommodates 1024 cores with $c = 4$) is generated in 8.7 seconds and a 24×24 in 125.7 seconds. The computation time for large designs grows significantly: a 32×32 configuration is generated in 47 minutes. The most time-consuming operation is clearly the layout generation, since the number of rings is quadratic on n and the TSP is NP-complete.

Comparatively, the evolutionary proposal in IMR [1] does not adapt to networks sized 16×16 or larger, the DRL approach [5] requires hours for a 10×10 network (compared to 0.6 seconds required for Affine-NoC) and the ILP formulation in [4] simply does not scale further than 6×6 .

2) *Bisection bandwidth (BBW)*: We compare the BBW of our proposal to other routerless multi-ring topologies and meshes. We consider two mesh references. Traditional meshes have small router radix and regular layout, but their BBW decreases with the network size. For this reason, we also consider either replicated designs (*Meshx2*) or *ruche* (express) links (*rucheX*, with X the length of the express link hop). Fig. 4a shows the *normalized BBW* of each configuration, i.e. bisection links divided by $N^2/4$, such that 1 represents a full-BBW configuration. We consider square arrangements to avoid imbalance. For each network size, we select the lowest-cost configuration (implying less total wiring length) that provides full BBW. This is highlighted in the Fig. as *MeshOpt*.

Fig. 4b presents the normalized BBW results for the selected baselines (*Mesh* and *Mesh-opt*) and different configuration of routerless networks. The base *REC* proposal is over-dimensioned by a factor of 2, whereas the base Affine-NoC by

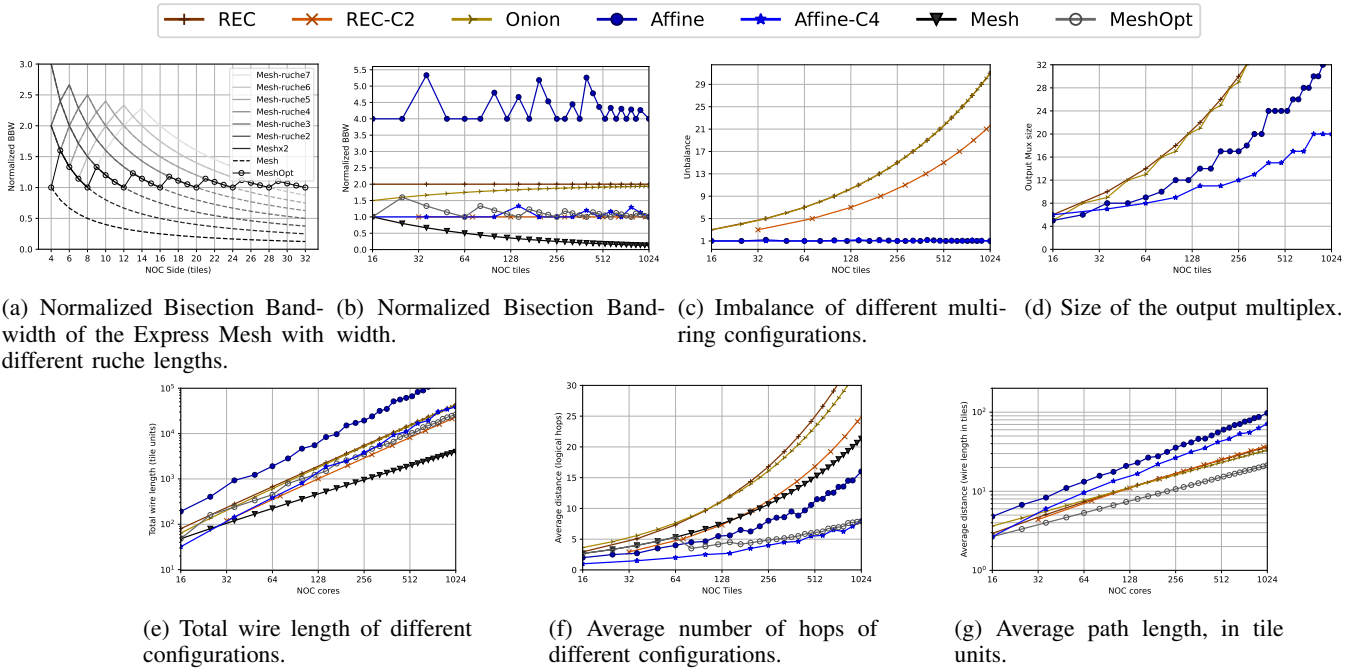


Fig. 4: Topological properties of Affine-NoC and different baselines.

a factor of 4 thanks to exploiting more available wiring. These overdimensioned designs motivate the use of *concentrated* designs in which 2 or 4 NoC Tiles are connected to each node. With concentration $c = 2$ for *REC* and $c = 4$ for Affine-NoC, all the configurations (except the base mesh) have full BBW.

3) *Balance and complexity of the design*: We define imbalance as the number of nodes in the largest ring divided by the number in the shortest ring. It is relevant since deflections start to occur in most-loaded rings and performance is reduced. Fig. 4c compares the imbalance of *REC*, *Onion* and Affine-NoC for $n \times n$ designs. The imbalance of the baselines *REC* and *Onion* grows proportionally to the network side, since they employ the smallest square rings (4 nodes) and rings that go over the periphery ($4 \cdot (n - 1)$ nodes). By contrast, Affine-NoC designs are almost perfectly balanced. A slight imbalance is observed when n is not a power of a prime, which corresponds to the impact of the *ring removal* mechanism. The design complexity relies mainly on the size of the output mux; Fig. 4d presents its size for the previous configurations, identifying the low requirements of Affine-NoC.

4) *Wiring requirements*: Affine-NoC employs long wires to connect consecutive nodes in the rings, which is a concern; total wiring in a suitable design should be not larger than other full-BBW proposals. Fig. 4e presents the accumulated wire length for the baseline router-based networks and the selected multi-ring approaches. It is calculated as the sum of the length (measured in “tiles”, so that a link that connects two adjacent tiles has length 1) of all the links in the NoC; the effect of concentration has been taken into account, by doubling the length of the links in one ($c = 2$) or both ($c = 4$) dimensions.

Affine-NoC without concentration requires the most wiring, but this design is also largely overdimensioned as confirmed by the BBW analysis. Affine-NoC with concentration $c = 4$ has full-BBW and its total wiring length is similar to MeshOpt and the other multi-ring configurations *REC* and *Onion*.

5) *Average logical distance*: Fig. 4f presents the average number of hops of different networks. It is calculated averaging the required number of hops for all possible network pairs, without considering the length of each hop. The balanced arrangement in Affine-NoC minimizes the number of nodes per ring, and this results in significantly lower number of hops than the other multi-ring approaches even without considering the benefit of concentration. With concentration, the average distance of Affine-NoC-C4 is similar or lower than *Mesh-opt*.

6) *Average physical distance*: Fig. 4g presents the average physical distance of the paths in each network. The Mesh (with or without express channels) provides the optimal value, since it directly maps to the underlying grid. Multi-ring approaches always increase this metric, since the rings do not minimally connect to any possible destination.

The average physical distance in Affine-NoC with concentration is slightly larger than in the references *REC* and *Onion*, because of the express channels. However, this length does not directly translate to average path delay, since it depends on the channel length traversed per cycle (defined by HPC_{Max}).

C. Performance results

1) *Latency*: Fig. 5a presents average latency results of Affine-NoC (using DirectConnect and $c=4$) vs *REC*, in a NoC connecting 256 tiles and using single-flit packets. The HPC_{Max} parameter that determines the length of the track

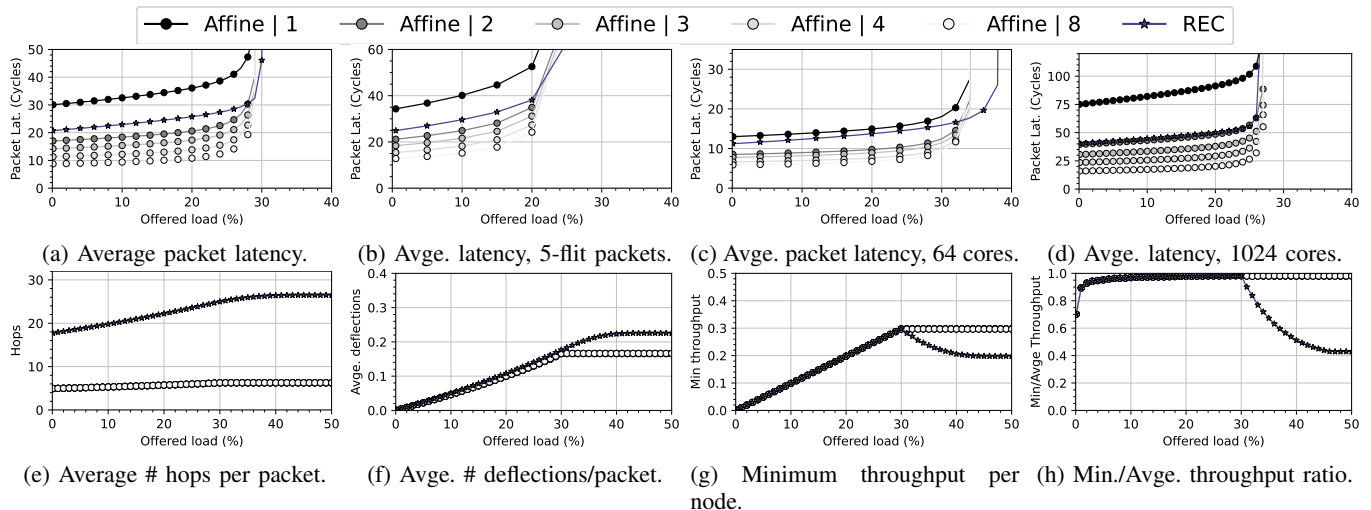


Fig. 5: Performance results of a NoC using Affine-NoC (DirectConnect with $c = 4$ and different HPC_{Max} parameters) and REC, under random uniform traffic. Unless otherwise noted, packets are 1-flit and the NoC employs 256 cores.

(measured in tile units) that is traversed per cycle is presented in the legend. Affine-NoC exploits long channels, and it outperforms the reference REC for $HPC_{Max} \geq 2$. Higher HPC_{Max} values, as expected, improve latency. Such configuration does not apply to REC, since its rings connect adjacent tiles. At 20% load, Affine-NoC with $HPC_{Max} = 2$ reduces latency by 20.5%, and this improvement rises to 60.0% with $HPC_{Max} = 8$. Figs. 5b, 5c and 5d confirm that the same trend is observed for larger packet size (5 flits), or different network sizes (64 and 1024 cores respectively).

2) *Hops and deflections*: Figs. 5e and 5f depict the average number of hops and deflections per packet. In this case the HPC_{Max} parameter has negligible impact. As expected, Affine-NoC has much lower hop count than REC thanks to its express channels. Interestingly, its balanced design also implies that all rings receive similar load, reducing congestion and deflections. At saturation, average hop count and deflections are reduced by 76% and 27%.

3) *Fairness*: Fig. 5g shows minimum node throughput, and Fig. 5h its ratio to average throughput. After saturation, Affine-NoC maintains steady throughput and fair results (ratio close to 1). By contrast, the REC model degrades after this point.

V. RELATED WORK

The paper has considered previous proposals that generate systematic constructions for the set of rings in routerless NoCs. Several alternatives employ machine-generated layouts, including the use of genetic algorithms [1], integer linear programming [4] or deep-reinforcement learning [5]. While these mechanisms can generate beneficial layouts, they are relatively complex, are slow to converge to a solution and may not converge at all for medium or large networks. Additionally, they cannot be easily replicated, since results are not available and experiments depend on many internal variables.

VI. CONCLUSIONS

This paper presents Affine-NoC, a multi-ring interconnect that relies on the Affine Plane to build a balanced NoC design that leverages express channels. The topology is built systematically and requires limited computation resources, mostly for the physical mapping of the rings. By avoiding crossbars and flow control logic, the node architecture requires limited resources. Results show that, with a small HPC_{Max} value, Affine-NoC outperforms previous multi-ring designs.

REFERENCES

- [1] S. Liu, T. Chen, L. Li, X. Feng, Z. Xu, H. Chen, F. Chong, and Y. Chen, "IMR: High-performance low-cost multi-ring noCs," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, pp. 1700–1712, June 2016.
- [2] F. Alazemi, A. AziziMazreah, B. Bose, and L. Chen, "Routerless network-on-chip," in *International Symposium on High Performance Computer Architecture (HPCA)*, pp. 492–503, Feb 2018.
- [3] J. Xiao and K. L. Yeung, "Onion: An efficient heuristic for designing routerless network-on-chip," in *LCN Symposium*, pp. 125–132, 2019.
- [4] J. Xiao, K. L. Yeung, and S. Jamin, "ILP formulation for designing rings in routerless network-on-chip," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pp. 1–6, 2019.
- [5] T.-R. Lin, D. Penney, M. Pedram, and L. Chen, "A deep reinforcement learning framework for architectural exploration: A routerless NoC case study," in *Intl. Symp. High Performance Computer Architecture*, 2020.
- [6] W. Dally, "Express cubes: improving the performance of k-ary n-cube interconnection networks," *IEEE Transactions on Computers*, vol. 40, no. 9, pp. 1016–1023, 1991.
- [7] D. C. Jung, S. Davidson, C. Zhao, D. Richmond, and M. B. Taylor, "Ruche networks: Wire-maximal, no-fuss NoCs : Special session paper," in *International Symposium on Networks-on-Chip*, 2020.
- [8] B. Sell et al, "Intel 4 CMOS technology featuring advanced FinFET transistors optimized for high density and high-performance computing," in *IEEE Symposium on VLSI Technology and Circuits*, 2022.
- [9] A. Psathakis, V. Papaefstathiou, N. Chrysos, F. Chaix, E. Vasilakis, D. Pnevmatikatos, and M. Katevenis, "A systematic evaluation of emerging mesh-like CMP NoCs," in *Symposium on Architectures for Networking and Communications Systems (ANCS)*, 2015.
- [10] T. Krishna, C.-H. O. Chen, W. C. Kwon, and L.-S. Peh, "Breaking the on-chip latency barrier using SMART," in *Symposium on High Performance Computer Architecture (HPCA)*, 2013.

- [11] W. Stein, *Sage: Creating a Viable Free Open Source Alternative to Magma, Maple, Mathematica, and MATLAB*, p. 230–238. London Mathematical Society Lecture Notes, Cambridge University Press, 2012.
- [12] I. Pérez, E. Vallejo, M. Moretó, and R. Beivide, “BST: A booksim-based toolset to simulate NoCs with single- and multi-hop bypass,” in *Intl Symp. on Performance Analysis of Systems and Software (ISPASS)*, 2020.