The Journal of Supercomputing (2022) 78:6664–6695
https://doi.org/10.1007/s11227-021-04098-4

Power efficient network selector placement in control plane of multiple networks-on-chip

Sonal Yadav[1,2] · **Ritu Raj**[3]

Accepted: 20 September 2021 / Published online: 25 October 2021

Abstract

Multiple networks-on-chip is a popular on-chip interconnect. This parallel communication infrastructure uses more than one NoCs to facilitate customized traffic distribution. Parallel architectures improve performance, however, at the cost of huge power dissipation. We propose power efficient customized placement of network selector hardware unit in the control plane at router. A network selector hardware unit is essentially used to distribute traffic between NoCs. Conventionally, this unit is placed in the data plane at network interface. We place network selector at switch allocator and at the routing unit of the router. The placement at switch allocator is more efficient than placement at routing unit or network interface. It improves 21% static power, 29% dynamic power, and 33% critical path delay of the circuit over network interface placement.

Keywords Multiple networks-on-chip · Placement · Network selector · Router · Network interface · Routing unit · Switch allocator

✉ Sonal Yadav
syadav.cse@nitrr.ac.in; sonaldv4@gmail.com

Ritu Raj
rituraj.ece@iiitkota.ac.in

[1] Department of Computer Science and Engineering, National Institute of Technology, Raipur, India

[2] Department of Computer Science and Engineering, Indian Institute of Information Technology, Kota, India

[3] Department of Electronics and Communication Engineering, Indian Institute of Information Technology, Kota, India

1 Introduction

Modern applications such as healthcare, production, sales, web, and organizations generate huge volume of data [1]. These applications heavily use computer vision, pattern recognition, image processing, etc., that are computation and communication intensive requiring manycore processors. The benefits of manycore processors can be increased by applying parallel processing of data. Parallel processing enables faster execution of these applications. A networks-on-chip (NoC) [2] architecture is used to support parallel processing of instructions and data by providing efficient interconnect to multi/many cores processors. These applications belong to interdisciplinary field dealing with the extraction of high-dimensional data from the real world such as computational biology, biometrics, biomedical imaging, artificial intelligence, robotics, security, self-driving cars, big data, and knowledge engineering to produce useful information. The consolidation of data-intensive applications is putting unprecedented pressure on NoC interconnection fabric. On the one hand, one of the processors' computation limiting factor is communication speed of on-chip network. On the other hand, the power consumption of NoC approaches 42% among on-chip components due to increasing network size in manycore processors [3]. So, the efficiency of NoC architecture is one of the major concerns for parallel computer architects.

The NoC efficiently scales bandwidth to enhance performance of manycore processors over dedicated point-to-point signal wires and shared buses. Parallel applications utilize NoC potential via operating data packets on different data links simultaneously. In tiled architecture [4] of manycore processors, each tile comprises core, caches, and router. Each core comprises a tiny network interface (NI) to split the large cache messages into smaller flits. These flits are transmitted to destination core through NoC. In manycore processors, the conventional NoC was using single network to interconnect on-chip routers and router to/from network interface. These networks are more than one in multiple networks-on-chip (Multi-NoC). The Multi-NoC emerges as faster communication infrastructure for parallel applications. These architectures are known for providing flexibility to devise a better power performance trade-off [5]. These circuits can address the communication bottleneck raised by emerging data-intensive applications. They facilitate simple, independent, and parallel data flows through more than one NoC interconnects. So Multi-NoC has been adopted in silicon prototypes, i.e., OpenPiton [6], Xeon Phi [7], SCORPIO [8], Tile [9], TRIPS [10], and RAW [11] to isolate different message classes for deadlock-freeness and quality of service.

Parallelism is considered as one of the primary ways to increase performance consistently. Since Moore's law slows down performance gain as there is insufficient power to keep all the cores active. Parallelism was initially explored for applications at the software level, then the focus shifted on hardware parallelism. Hardware-based solutions are faster; however, they are power expensive, hence less preferred for actual implementation. The power-efficient customized architectures are popular in this direction [12].

Current Multi-NoCs are always designed with network selector (i.e., traffic distribution hardware unit); however, its placement and circuit design are not analyzed. In this paper, we explore the answers to the following questions:

- Why do we need network selector?
- Where should network selector be placed?
- What should be the hardware circuit design of network selector?
- Does the placement of network selector in data plane or control plane affect the complexity of the network selector circuit as well as overall Multi-NoC design?
- What consequences can be observed in Multi-NoC design on changing the placement of network selector?

In this paper, we achieve a customized Multi-NoC hardware circuit design by changing the placement of network selector hardware unit from conventional network interface to router of the Multi-NoC. Network selector functions as the digital demultiplexer that takes the information from one input and transmits over one of several outputs. In Multi-NoC, a flit is demultiplexed to any of the networks through network selection hardware unit (named as Net-Demux1). A network selector is placed in the data plane of the network interface in Multi-NoC. We propose a new hardware implementation in the control plane of the router. This placement changes the network selector's circuit complexity, area, power, and performance. The premise is that changing the placement of network selector can efficiently customize the Multi-NoC. Our design includes two important hardware features. First, the network selector circuit is customized by changing its placement from network interface to router. Second, the consequence of these changes results in customization of the Multi-NoC. We utilize this opportunity to reduce the hardware complexity of Multi-NoC. Extensive experimental results have demonstrated the effectiveness of the proposed customization in Multi-NoC design to boost the NoC performance.

We propose an improved placement of Net-Demux at switch allocator of the router as an alternative to the network interface and the routing unit. Net-Demux placement changes the average number of signal transitions in a single cycle of the circuit, and hence it varies the switching activity of the circuit. Power dissipation at Net-Demux is related to the difference in input and output switching activity. The network selector in proposed Multi-NoC architectures is placed at the network interface. Hence, power efficiency is achieved. The key contributions of this paper are as follows:

1. Exploring the possibilities of placements of network selector.
2. Analysis of network selector architecture with different placements.
3. Comparison of different placements with the traditional one.

The idea is implemented and experimented with Gem5[1] full system simulator that is extensively used open-source simulator for evaluating performance of manycore

[1] Gem5 simulator is an integration of M5 and GEMS simulators [13, 14]. M5 supports CPU models, Instruction Set Architecture (ISAs), input/output devices, infrastructure, whereas GEMS supports interconnect models including cache coherence protocols.

processor architectures. Gem5 is integrated with Garnet that is a detailed interconnection network model. Garnet simulates a detailed router micro-architecture model and requisite components of on-chip networks [15]. Hardware parameters of these components can be configured with different values in Garnet during simulations. For placement of network selector, we need to make significant changes in the Garnet code. Gem5 is also integrated with ORION 2.0 simulator [16] to estimate NoC power and area that helps in early stage design space exploration for multicore and manycore processors.

We have integrated Gem5 with PARSEC benchmark [17] for real-time analysis of our proposals. The full system simulation runs the parallel section of benchmarks, and it is most important for performance assessment. The full system simulation is performed in three phases. The first phase is warm up phase where empty caches are initially filled with data. The second phase, i.e., region of interest (RoI) is relatively steady state. After benchmark execution, the third phase is the clean-up phase where the operating system does garbage collection.

The rest of this paper is organized as follows. Section 2 discusses the related work on Multi-NoC architectures. Section 3 introduces the placement impact on digital circuit design techniques. Section 4 discusses Net-Demux placement at the data and the control plane of the NoC. Hardware synthesis and benchmark results are presented in Sect. 5. Finally, Sect. 6 concludes the results.

2 Related work

In this section, we discuss various research works related to placement of hardware components on-chip and customizations in Multi-NoC architectures.

2.1 Placement of hardware components on-chip

In this subsection, we discuss the latest work on placement of Net-Demux and other hardware components. The different approaches, contribution, benefits, and their limitations are given in Table 1. Yadav et al. [18, 19] initiated the discussions on the placement of Net-Demux in Multi-NoCs. They also proposed the idea of Net-Demux placement at the control plane of the router.

The placement of hardware components is explored by a number of researchers. Abts et al. [20] has explored optimal placement of memory controllers for different topologies (mesh and torus), routing, and workloads as memory controllers are less than the number of cores. Efficient placement of memory controller can reduce contention, hot spots, and lower the latency variance for memory-intensive applications. Zhao et al. [21] proposed fast evaluation of memory controller placement using path-load assessment. The complexity of path load counting algorithm is $O(M \times M \times K)$ for $M \times M$ mesh with K memory controllers. These explorations are limited to mesh and torus topologies. Likewise, Hung et al. [22] proposed optimized IP placement using a genetic algorithm to minimize thermal hot spots. Hu and

Table 1 Literature summary on placement of hardware components on-chip

Domain	Authors, Year	Contribution	Benefits (+) and Overheads (–)	Gap/limitation
Net-Demux placement	Yadav et al. 2019	Proposed the placement of Net-Demux in the control plane at switch allocator of the router. Evaluate Net-Demux hardware benefits on control plane placement over data plane placement [18]	Net-Demux saves + static power by 78% + area by 87% + dynamic power by 28% + critical path delay of the circuit by 33%	Brief analysis on hardware design metric of Net-Demux with proposing the idea of Net-Demux placement at switch allocator
	Yadav et al. 2020	Discussed the idea of Net-Demux placement at routing unit and switch allocator of the router [19]	Energy efficiency over single-NoC is achieved + 46% by placement at switch allocator + 40% by placement at routing unit + 30% by placement at network interface Energy efficiency over network interface placement is achieved + 33% by placement at switch allocator + 26% by placement at routing unit + 33% improvement in critical path delay	Briefly discussed the idea of Net-Demux placement at routing unit and switch allocator. No hardware architecture was discussed. Energy efficiency improvement discussed without experiments

Table 1 (continued)

Domain	Authors, Year	Contribution	Benefits (+) and Overheads (–)	Gap/limitation
Network controller	Robaei et al. 2019	A hybrid wired–wireless NoC architecture uses network controller to distribute the traffic between wireless antennas and wired NoC network. Each router needs to be equipped with a network controller to manage handover between wired and wireless networks [25]	+ reduction in traffic congestion near hotspots + instruction per cycle is improved by 80% + the mm-wave & Sub-THz design achieve 33% and 22% reduction in average energy per message – the mm-wave & Sub-THz design have area overhead 2.87× and 2.38×, respectively, relative to the baseline mesh	Placement of network controller at wireless antenna or wired network was not explored
Memory controller placement	Abts et al. 2009	Optimal placement of memory controllers was explored for mesh and torus topologies, routings, and workloads [20]. Simulation showed that the diamond placement performed best using dimension-ordered routing, as it can spread traffic across all rows and columns	+ The diamond placement has 33% less link contention compared to the baseline placement + the diamond placement reduces interconnect latency by an average of 10% for real workloads	Memory controller placement is explored only for mesh and torus topologies
	Zhao et al. 2021	Fast placement of memory controllers using path load evaluation [21]. They have compared six types of memory controller placements, i.e., S1-a to S1-f, in 4×4 mesh with eight memory controllers. The path load value rate of S1-a:S1-b:S1-c:S1-d:S1-e:S1-f is about 1.92:1.21:1.1:1:1:1.	+ S1-f placement reduced average latency by 64.57%, 24.63%, and 9.69% over S1-a, S1-b, S1-c, respectively + S1-f placement improves throughput by 64.73%, 7.50%, and 3.09% over S1-a, S1-b, S1-c, respectively	The complexity of path load counting algorithm is $O(M \times M \times K)$ for $M \times M$ mesh with K memory controllers. Placement of memory controller was explored only for mesh

Table 1 (continued)

Domain	Authors, Year	Contribution	Benefits (+) and Overheads (–)	Gap/limitation
Intellectual property (IP) placement	Hung et al. 2004	IP virtualization and placement using genetic algorithm to minimize thermal hotspots [22]	+ Simultaneous virtualization and placement is established as it reduces both peak and average temperature by 2–3 °C as compared to non-virtualized approach	Complexity of genetic algorithm. The runtime for the experiments is about 9 ∼ 13 minutes for 5000 generations
IPs/cores mapping	Hu and Marculescu 2003	Presented an algorithm which automatically maps the IPs/cores onto a generic regular NoC architecture to minimize energy consumption [23]	Categories I, II, III and IV contain ten applications with 9, 16, 25 and 36 IPs + On average, the energy consumption of the solution generated by proposed algorithm is only 3%, 6% and 10% for category II, III and IV, respectively	Solving time of simulated annealing is affordable only for 5×5, the run time of simulated annealing increases dramatically as the system size scales up
Optimal static routes	Srinivasan and Chatha 2005	Proposed mapping and routing on NoC architecture (MOCA) approach by bi-partitioning based slicing tree generation technique to map cores on to the different routers of the mesh. In the second phase, MOCA invokes a hierarchical router that generates routes for all the communication traces. Optimal static routes between cores to achieve bandwidth and latency requirements [24]	+ Resulting solutions were within 14% optimum + the energy consumption of MOCA was within 22% + MOCA generated results within 0.01 s	Algorithm complexity to trade off energy minimization (obtained by routing high bandwidth traces in minimum hops), with the objective of obtaining legal solutions (by routing tight latency traces in minimum hops) to obtain a pareto optimal point

Table 1 (continued)

Domain	Authors, Year	Contribution	Benefits (+) and Overheads (–)	Gap/limitation
Application mapping	Kullu et al. 2020	Maps the given application-specific irregular topologies to the reconfigurable mesh topology [26]	+ Average hop count is 18.16% + energy consumption reduces up to 16.11% over approach [27]	Entire evaluation is based on the assumption that energy consumption of a router is 5 times higher than a switch. Extra $O(n^2)$ time complexity for extracting paths
Power gating	Baharloo et al. 2020	Power gating in Multi-NoC is a promising solution to minimize the performance penalty faced in traditional NoC [28]. Packets change the subnet in Multi-NoC to avoid encountering switched-off routers	+ Network latency decreases by 10.5% + execution time improves by 4.5% + static power consumption improves by 17.6% —imposing 1.9% hardware overhead compared to traditional Multi-NoC design	Due to power-gating technique, waking-up a chain of routers in a switched-off subnet incurs performance penalty. Power gating logic implementation overhead
	Ali et al. 2018	Proposed a system generating discrete frequencies via a series of microring resonators to generate frequency channels with 200 MHz full width at half maximum and 2.5 GHz free spectral range for wireless-assisted Multi-NoC application [29]	+ Average network power is 55.4% and 11.1% lower than traditional single-NoC and power-gating Multi-NoC designs, respectively + 10.4% improvement in network latency due to wireless links	The experiment was limited to four network 32-bit physical channel (4N-32b-P). Applying power-gating mechanism in 4N-32b-P decreases the overall network power, on the other hand, causes an increase in network latency
Switch allocator customization	Ahsen et al. 2018	Switch allocator design that makes two sets of decisions for the two halves of a cycle to transfer two flits in a single cycle [30]	+ Low-voltage DDRNoC implementation reduces power consumption by 40%-latency increases by 26–40%	Two flits transferred through data path in a single cycle

Table 1 (continued)

Domain	Authors, Year	Contribution	Benefits (+) and Overheads (–)	Gap/limitation
Crossbar switch	Rad et al. 2021	An arbitration mechanism was proposed for crossbar switch to fairly allocate the port priorities based on the current traffic load and the wireless channel bandwidth [31]	+ Network throughput was increased by 12% for random traffic patterns	The increase in demand for using shared wireless links and the presence of few numbers of these channels on a chip leads to port contention in WiNoCs
Adding bypass channels	Ma et al. 2021	Adding explicit physical channels inside a router to bypass intermediate nodes could achieve lower latency. All bypass requests are combined with corresponding flits [32]	+ Transmission latency of the proposed NoC is 5.24 cycles that is 63.54% less than 1-cycle NoC + 30% latency reduction —additional wires	Bypass setup request in previous designs requires additional pipeline stages and a lot of wire resources. Use of unicast mechanism instead of broadcasting

Marculescu [23] proposed an energy-aware mapping of tasks to cores. Srinivasan and Chatha [24] proposed optimal static routes between cores to achieve bandwidth and latency requirements. However, all such placement aspects have been explored with single-NoC.

In this paper, we present a detailed analysis on the Net-Demux placement. Yadav et al. [18, 19] discussed this idea briefly. Likewise, network controller [25] is implemented for traffic distribution between hybrid wired–wireless architecture. However, they do not provide analysis on network controller architecture and its placement. The placement of other on-chip hardware components have different challenges than Net-Demux placement. In memory controller placement [20, 21], the number of memory controllers are less than the number of routers. This requires to devise a suitable placement of memory controllers on-chip, whereas Net-Demux units are in proportion to the number of routers. We look for Net-Demux placement within the tile. IP virtualization and placement [22] minimizes thermal hotspot in NoC. Likewise, IP/Core maps [23] on to NoC to minimize energy consumption according to their proposed algorithm. In our approach, Net-Demux placement minimizes energy consumption by reducing static power due to hardware customizations. The optimal route selection [24] is also proposed to find minimal path to save energy. However, algorithm complexity introduces some hardware overheads. In contrary, proposed placement simplifies the hardware complexity.

Net-Demux placement improves static power significantly without any performance penalty, while power gating techniques [28] improve static power at the cost of performance penalty [33]. To mitigate the performance penalty, the power gating is applied to wireless multiple NoC [29]. However, the benefits of power saving is mitigated due to overhead of antenna power. Switch allocator [30] is customized to transfer two flits in a single cycle. However, the latency increases by 26–40% with this approach. We have also customized switch allocator by integrating Net-Demux. Since our hardware is simple, we do not have any latency overhead.

2.2 Multi-NoC customizations

In this subsection, we discuss traditional Multi-NoC architecture, their customizations, and traffic communication approaches. Multi-NoC is adopted over single-NoC since several customization flexibility further improves NoC efficiency significantly. Balfour and Dally [34] duplicated NoC topologies, such as Mesh and CMesh, to improve system performance. Carara et al. [35] replicated the physical networks by taking advantage of the abundance of wires between routers for enhancing efficiency. Since replication doubles the power of the NoC, Yoon et al. [36] partitioned NoC into a number of sub-networks while keeping the overall amount of wires and buffering resources constant to avoid power overheads.

The wires can be customized in Multi-NoC in different ways. Grot et al. [37] proposed partitioning of two parallel networks named as multidrop express channels (MECS-X2). Kumar et al. [38] proposed channel slicing to improve the poor channel

utilization of concentrated channels. Gómez et al. [39] followed a slightly different approach for network partitioning. They divided the wires into several parallel links, while the router remains single to improve the network throughput, which reduced the consumption of area and power.

These architectures are explored for replicating internal components as well. Teimouri et al. [40] divided the n-bit wide network resources in a router, such as links, buffers, and crossbar switch, into two parallel n/2-bit sub-networks to introduce reconfigurable shortcut paths. Noh et al. [41] proposed parallel crossbars to increase the flit transfer rate between input and output ports. The resulting router has a simpler design that performs better than a single-plane router. Gilabert et al. [42] proposed a multi-switch architecture and compared it to a multiplane NoC. The multi-switch approach provides better performance than an equivalent multi-network with only a small area overhead.

The traffic communication can also be customized using Multi-NoC [43, 44]. Volos et al. [45] proposed a specialized cache–coherence network-on-chip (CCNoC). This architecture combines asymmetric multiplane and virtual channels for efficient cache coherence communication.

3 Placement impact on circuit design techniques

In this section, we have discussed design metrics of digital circuits that are affected by the placement of the hardware logic. These design metrics include timing delay, static and dynamic power. Typical placement objectives include the improvement in these metrics for the overall gain in power-performance efficiency. The timing delay is directly affected by the placement of Net-Demux. An inferior placement might degrade the overall efficiency. The dynamic power of a logic gate can be reduced by minimizing the physical capacitance[2] and the switching activity. At all levels of the design abstraction, the switching activity can be minimized. However, we discuss the front end digital circuit characteristics that are affected by Net-Demux placement.

3.1 Logic restructuring

The topology of a circuit affects the overall power dissipation. The implementation of logic network, as shown in Fig. 1, $O = X \cdot Y \cdot Z \cdot W$ is possible in two alternate ways. Let's assume that all primary inputs (X, Y, Z, W) are uniformly distributed (i.e., $P_{1\,(X,Y,Z,W)} = 0.5$), i.e., $P(X = 1) = 0.5$ and this holds true for all other inputs as well. For an AND gate with input X and Y, the probability that the output is 1, i.e., $P_{1\,(X,Y)} = P_{1\,(X)} \times P_{1\,(Y)}$. The probability that the output is 0, i.e., $P_{0\,(X,Y)} = 1 - P_{1\,(X,Y)}$, and the transition probability is

[2] The physical capacitance can be minimized in a number of ways, including circuit style selection, transistor sizing, placement and routing, and architectural optimizations [46]. These being a part of the fabrication, hence beyond the scope of this paper.

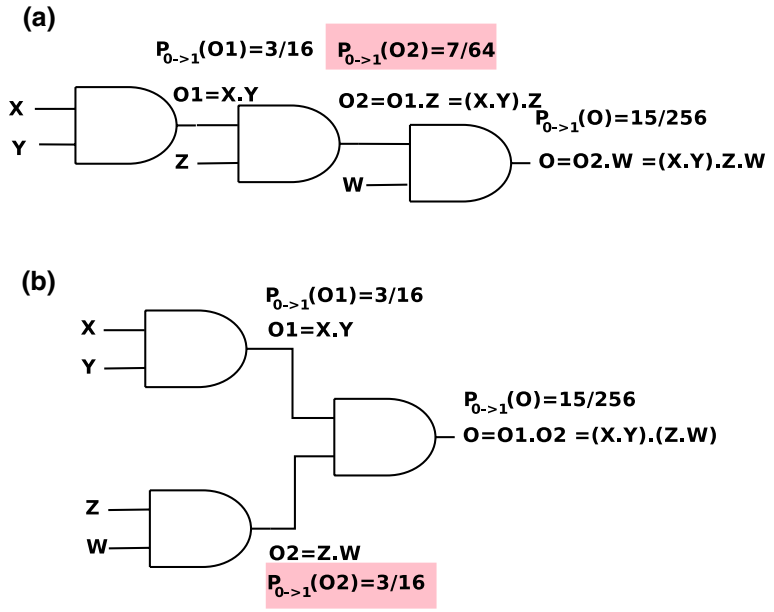

Fig. 1 Switching activity variation due to circuit topology as illustrated through **a** chain structure, and **b** tree structure. The output transition probability is uniform, i.e., ($P_{1(X,Y,Z,W)} = 0.5$) for all the inputs

$$P_{(0\to1)(X,Y)} = P_{0(X,Y)} \times P_{1(X,Y)} \tag{1}$$

Likewise, the output signal transition probabilities $P_{0\to1(O1)}$, $P_{0\to1(O2)}$, $P_{0\to1(O)}$ is calculated in chain structure as shown in Fig. 1a. The output signal transition probability $P_{0\to1(O2)}$ for output $O2$ changes in a tree structure is demonstrated in Fig. 1b. On comparing chain and tree topology, the results indicate that the chain implementation has lower switching activity in intermediate outputs than the tree implementation as observed with random inputs. The lower switching activity during intermediate outputs is important as these intermediate signals may become the input of other circuits. Thus, switching activity propagates in the circuit. It may vary significantly with placement as Net-Demux integrates with different hardware modules of the NoC circuit.

3.2 Input signal ordering

The input signal ordering is another parameter that affects the switching activity of the circuit. The impact of reordering of the input signals on switching activity is shown in Fig. 2. The circuits are identical in topology, but the input signal X is swapped with Z between the circuits of Figs. 2a, b. The reordering of the input signals affects the output switching activity of the circuit. Let the probabilities of input

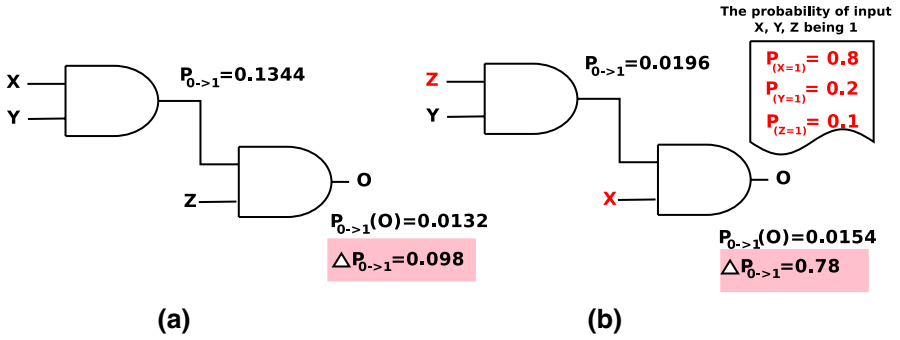

Fig. 2 Reordering of input signals affects the output switching activity of the circuit **a** inputs are ordered as *X*, *Y*, *Z* **b** input order changes as *Z*, *Y*, *X*. The $\triangle P_{0\to1}$ shows the reduction in output switching activity over intermediate output switching activity. (Here $\triangle P_{0\to1} = \triangle P_{0\to1}(O)/P_{0\to1}$. The lower value of switching activity is better)

signal being 1 be $P_{(X=1)} = 0.8$, $P_{(Y=1)} = 0.2$, $P_{(Z=1)} = 0.1$. Then from Eq 1, the output switching activity is calculated as follows.

1. Circuit in Fig. 2a: the output signal transition probability is $(1 - 0.8 \times 0.2)\,(0.8 \times 0.2) = 0.1344$. The final output transition is $(1 - 0.1344 \times 0.1)\,(0.1344 \times 0.1) = 0.0132$.
2. Circuit in Fig. 2b: the probability of a $0 \to 1$ transition is $(1 - 0.2 \times 0.1)\,(0.2 \times 0.1) = 0.0196$. The final output transition is $(1 - 0.0196 \times 0.8)\,(0.0196 \times 0.8) = 0.0154$.

We observe a substantial reduction in final switching activity over intermediate switching activity in the circuit shown in Fig. 2b, i.e., 0.78 compared to the circuit shown in Fig. 2a, i.e., 0.098. It is beneficial to postpone the introduction of signals with a high transition rate [46]. Thus a simple reordering of the input signals, significantly reduces the signal transition rate (switching activity).

3.3 Time delay of signal paths

The *timing delay* refers to the time required by a signal to reach from input to output pin of the circuit. It is the sum of interconnect and gate delays that constitute the path. A signal delivers from input pads to gate outputs and proceeds from the output pad to gate input by following the circuit path [47]. The longest path that introduces maximum timing delay is known as *critical path* of the circuit.

Delay along critical path decides the time period of the clock or, alternately, the maximum operating frequency of the circuit. For reliable operation of the circuit, the difference between two clock arrivals should be more than the critical path delay. Net-Demux placement at various locations varies the signal path length, the number of signal paths, and the critical path delay.

Placement of Net-Demux can be modeled mathematically using timing graphs [48]. Let the total number of signal paths in Multi-NoC circuit be ρ_n and in network selector be q_m, where 'n' and 'm' are finite integers greater than equal to

1. Lets assume each signal path comprises 'X' number of logic gates. The network selector is integrated at, lets say, i^{th} position of any one signal path ρ_j, where $1 \leq \mathrm{i} \leq X$ and $1 \leq \mathrm{j} \leq n$.

1. The total number of signal paths is $s_o = \rho_n + q_m$, where 'o' is an finite integer, after integration of network selector.
2. One of the signal path (s_{iz}) out of the total number of signal paths (s_o) is starting from logic gate x_i and ending at logic gate x_z, i.e., $s_{iz} \rightarrow (x_i \rightsquigarrow x_z) \exists\, i$ and $z \geq 1 \ni s_i = D_{x_i}$ and $s_z = D_{x_z}, \forall\, 1 \leq i, z < X$, where D_{x_i} and D_{x_z} are delay of logic gate x_i and x_z. So the timing delay of logic gate x_i is dependent on x_z. After integration of Net-Demux, the timing delay of the circuit would be $D_{x_i} + D_{x_z}$.
3. Lets say ϕ is a set of signal paths' delays without integration of network selector, $\phi = \{D_{\rho_1}, D_{\rho_2}, \ldots, D_{\rho_n}\}$, where D_{ρ_1}, D_{ρ_2} and D_{ρ_n} are the total delay of all logic gates in ρ_1, ρ_2 and ρ_n signal paths, respectively. There exists a signal path (ρ_j) having maximum delay max(ϕ) which is the critical path of the circuit without the integration of the Net-Demux.
4. Lets say ψ is a set of signal paths' delays, $\psi = \{D_{s_1}, D_{s_2}, \ldots, D_{s_{n+m}}\}$, where D_{s_1}, D_{s_2} and $D_{s_{n+m}}$ are the total delay of all logic gates in s_1, s_2 and s_{n+m} signal paths, respectively. There exists a signal path (s_o) having maximum delay max(ψ) which is the critical path of the circuit.
5. When max(ψ) > max(ϕ), then critical path delay of Multi-NoC circuit is increased. Otherwise, there is no impact on critical path of the circuit since the critical path is unchanged even after integration of the Net-Demux.

We infer that the integration of Net-Demux can either increase the critical path or may not affect the critical path of the circuit. This depends on where the Net-Demux is integrated in Multi-NoC. Therefore, Net-Demux should be integrated with a non-critical signal path. It is noted that besides the path delay, the input signal ordering also impacts circuit timing characteristics. We analyze all these properties at the abstraction level of the circuits. In the next section, we discuss the placement impact on Net-Demux and NoC architecture.

4 Architectural implications of net-demux placement

The Net-Demux can be placed in either control plane[3] or data plane. Control planes refer to the part of the hardware responsible for generating control signals, i.e., coordinating the movement of packets through the data plane. Likewise, the data plane or data path handles the storage and movement of packets. It comprises a set of input and output buffers located in the network interface (NI) as well as routers/switches. The data and control plane have data and control input/output (I/O), respectively.

[3] We synonymously use the term control plane or control path.

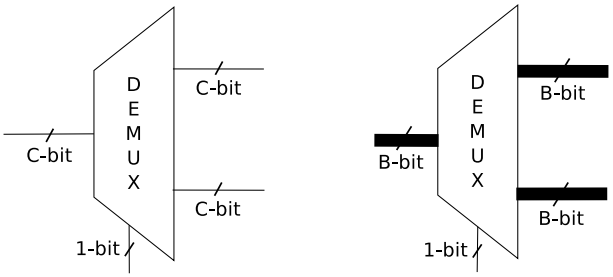

Fig. 3 Placement of demultiplexer

In Fig. 3, we compare control versus data plane placement hardware implementation overhead of a single demultiplexer. A small[4] bus-width C-bit (let's say) I/O is sufficient to find the NoC network as compared to a large[5] bus-width B-bit (let's say) I/O on data plane demultiplexer. The placement affects static power and area as follows.

1. The control plane placement is independent of the network link width.[6]
2. The benefits in hardware overhead with control plane are $(I_{NI} \times B)/(I_R \times C)$[7] times more than data plane placement. The placement in the control plane is B/C times better than data plane placement because link-width $B >> C$ and $I_{NI} = I_R = 4$.[8]
3. The control plane placement is more scalable than data plane placement with increasing number of NoC networks. Because the width of the select line and the number of output lines have a minor hardware implementation overhead due to a lower input link width, i.e., C-bit.

Thus, control plane placement is beneficial over data plane placement. It modifies not only the NoC micro-architecture but also bring variations in circuit paths. Since the circuit's physical design significantly impacts power dissipation, area, and timing delays, an incompatible placement degrades the overall efficiency, which cannot be improved even with intelligent routings. Multi-NoC is made via partitioning of single-NoC having four network links and four NI.

In Fig. 4, we compare the placement impact in the data plane (at NI) vs control plane (on the router) of dual-NoC. In conventional Multi-NoC, the Net-Demux is

[4] 3 and 4-bits, for our proposed architecture.

[5] 128/256/512-bits (128-bits for our proposed architecture).

[6] Recent NoC architectures are using higher bandwidth networks (for example, a link width of 512 bits is required to sustain modern per-core bandwidth [49]) So the placement in data path significantly increases hardware overheads.

[7] Here, I_{NI} is the number of NI links, which are the inputs to the router, and I_R is the number of inputs from other routers.

[8] We place Net-Demux only for one NoC as another NoC already has separate traffic.

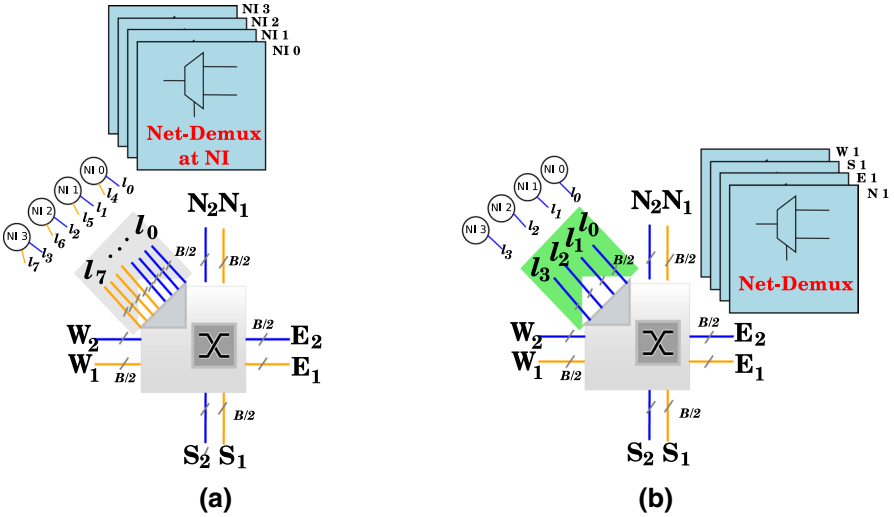

Fig. 4 Placement of Net-Demux at **a** network interface (NI 0, NI 1, NI 2, NI 3), **b** router's network links (N1, E1, S1, W1)

placed at NI, as shown in Fig. 4a. The links of both NoC networks are connected with different NIs as *NI* 0 (l_0, l_4), *NI* 1 (l_1, l_5), *NI* 2 (l_2, l_6), and *NI* 3 (l_3, l_7). Every NI has a choice of traffic distribution between any one of NoC networks, so four hardware units of Net-Demux is placed corresponding to each NI. We named this conventional Multi-NoC as dual-network-NoC. In Fig. 4b, the placement of Net-Demux at router customizes dual-network-NoC architecture. The 2-networks can be formed through routers. The single links are sufficient to connect different NIs, i.e., *NI* 0 (l_0), *NI* 1 (l_1), *NI* 2 (l_2), and *NI* 3 (l_3), with a single router. The four hardware units of Net-Demux is placed corresponding to N1, E1, S1, and W1 network links of the router. We named this architecture as 2-network-NoC.

The placement changes the architecture of the Net-Demux according to the function of the hardware unit. It also customizes NoC architecture. In the subsequent subsections, we discuss in detail the conventional placement at the NI and compare it with our proposed routing unit and switch allocator placements.

4.1 Net-demux at network interface

Dual-network-NoC places Net-Demux at NI between the core and router. Core messages enter into NI, and then into the router through NI links, as shown in Fig. 5. A router consists of eight core links and eight network links for dual-network-NoC. The odd-numbered links constitute the NoC_1 network, whereas the even-numbered links represent the NoC_2 network.

Messages are the logical unit of communication and may be arbitrarily long. These messages are divided into packets that are further segmented into

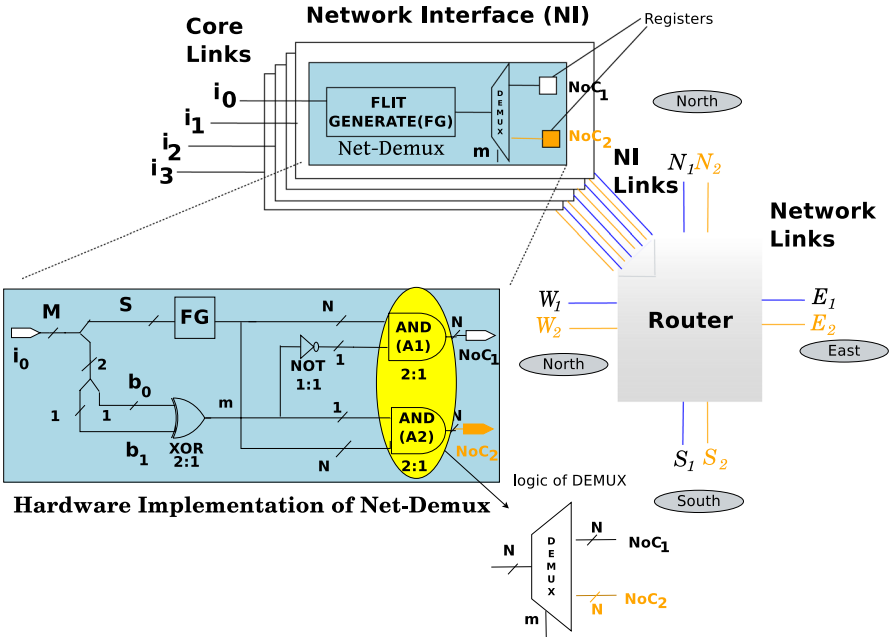

Fig. 5 Network interface with Net-Demux hardware implementation

fixed-length flits.[9] If the number of generated packets is k, then single packet size is $S = (M-2)/k$, and the flit size is $N = (M-2)/k \times f$ where f is the number of flits per packet. On receiving a message at NI, the flits are checked for the control or data packets since both types of flits are generated separately.

A control packet is composed of one control flit. It consists of a coherence command and the memory address. Contrary, data packets are made up of five[10] flits. It consists of a head flit containing the destination address, three body flits, and a tail flit. The end of a packet is indicated by tail flit. On arrival of this flit, a signal is triggered for generating the next flit.

The control packet is a single flit though it keeps all the required information of destination. Only control flit and head flit keep the control and routing information. The data and tail flits follow the head flit status to reach their respective NoC networks. These flits consist of payload. However, the last data flit is padded with zeros (if required).

Every message is prepended with two bits to identify the message class as shown in Table 2. The encoding is specified for flag m according to different message classes. The flag m is computed from bits b_0 and b_1. Its value decides NoC selection for message forwarding. These bits are passed to inputs $i_1[0]$ and $i_2[0]$ of the Net-Demux circuit as shown in Fig. 5. These two bits are XORed. Output m of the

[9] FLow control unITs.

[10] Five flits for our NoC architecture.

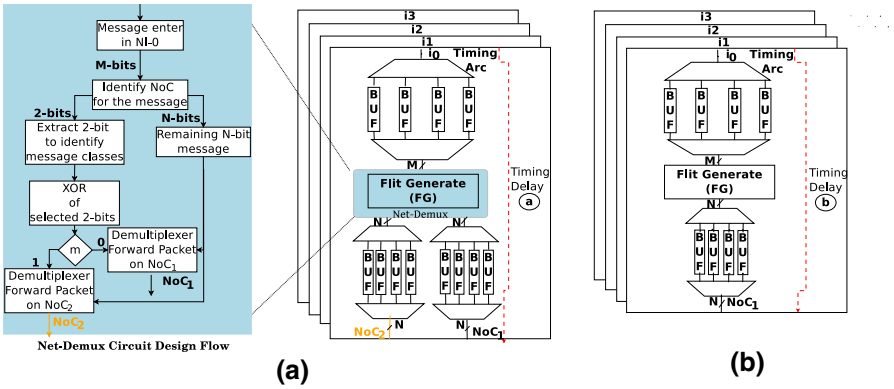

Fig. 6 Comparison of network interface **a** with Net-Demux placement in dual-network-NoC, **b** without placement in single or 2-network-NoC. The circuit design flow reflects the complexity of placement over NI on comparing timing delay of both circuits that is $a > b$

Table 2 NoC selection for cache messages via bit encoding

Type	Message classes	2-bit Encoding[a]	m[b]	NoC network
1	Control	$b_0 = 0, b_1 = 1$	1	NoC_2
2	Writeback control	$b_0 = 1, b_1 = 0$		
3	Request control	$b_0 = 0, b_1 = 0$	0	NoC_1
4	Response control			
5	Writeback data			
6	Response data			

[a] 2-bit codes assigned

[b] m-flag indicates the message type

XOR gate is connected to two AND gates $A1$ and $A2$. The other input to these gates is a message signal. So whenever the message belongs to Control or Writeback Control, the value of flag m is set to 1. Then, AND gate $A2$ passes message bits. This ensures the selection of NoC_2 for these messages. For other message classes, NoC_1 is selected.

The Net-Demux is integrated with flit generation (FG), as shown in Fig. 6a, which is compared to single-NoC (in Fig. 6b). Once the credit signal indicates the availability of an empty virtual channel, the FG checks the counter. If it is zero, the first flit is checked for signal m. If m is true, the output port address of NoC_2 is assigned to flit, else NoC_1 address is allocated. The complete Net-Demux circuit design flow, from the incoming message in NI to a selection of NoC network for the packet, is shown in Fig. 6a.

Placement of Net-Demux at NI increases the critical path delay as it is on the critical path. As a result, the timing delay 'a' with Net-Demux placement is larger than the timing delay 'b' without Net-Demux placement at NI. Since $a > b$, the placement of Net-Demux increases the critical path of the circuit. There is overhead in power, area, and timing delay on placing Net-Demux at NI.

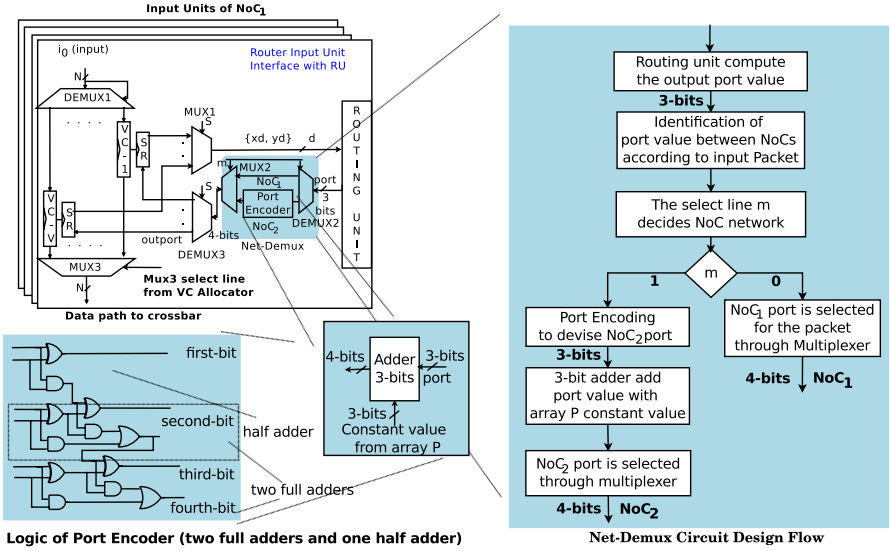

Fig. 7 Placement of Net-Demux between the input unit and the routing unit of the router. One half-adder and two full-adders require to implement the logic of port encoder. Flowchart demonstrates the hardware implementation of Net-Demux placement after the route computation of RU for selecting either of NoC networks

4.2 Net-Demux at router

Conventional Net-Demux placement is at NI that is a part of the data plane. On the router, placement can be done in control or data plane. In the data path, the placement at the input/output of the crossbar is feasible. However, the area and power consumption are higher than the control plane. The required Net-Demux hardware units are in the order of the number of inputs to NoC_1 from the router.

The area and static power are proportional to the data path width. Since the hardware cost increases in proportion to data path width, we have not explored the Net-Demux placement at the crossbar. We propose placement at the routing unit and switch allocator of the router due to lower hardware costs in these control planes.

Routing Unit (RU). It selects the output port for flit traversal. Flits are divided into head, body, and tail flits. The least significant three bits of head flit has destination information that is utilized during route computation. The port encoder of Net-Demux converts this information into four-bit output port. It is made of one half-adder and two full-adders. The computed output port is updated in Status Registers (SR).

Routing unit reads the status register to get the routing information for head flit. As shown in Fig. 7, two control lines read the destination address x_d and y_d from the SR and feed this information to the input of Net-Demux that checks the message class of head flit and assign its NoC network. The selection of NoC network is dependent on signal m. If the signal m holds the value zero, the output port of NoC_1 is selected by Net-Demux, else the output port of NoC_2 network. The circuit design

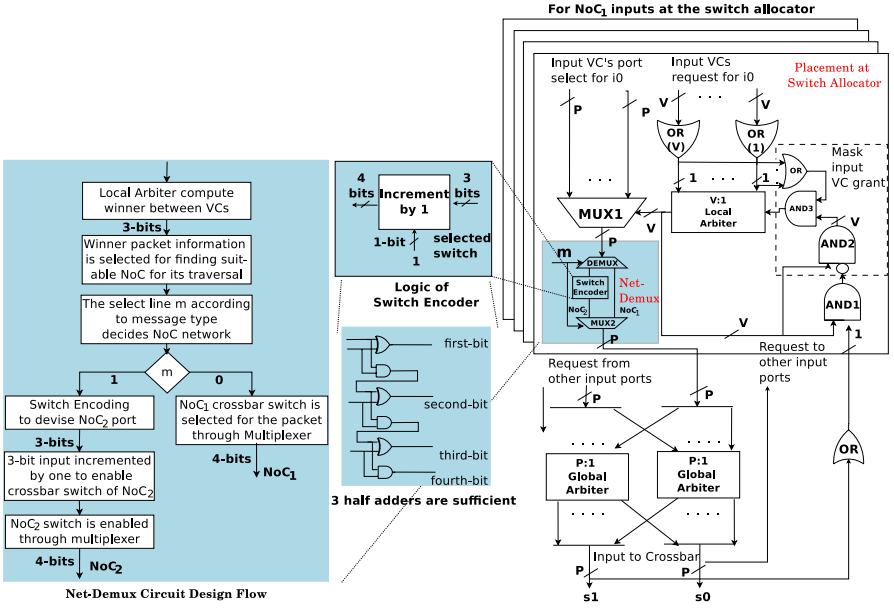

Fig. 8 Net-Demux placement at switch allocator of the router. Only three half adders are sufficient to implement switch encoder logic. The flowchart demonstrates the selection of either of NoC networks with the placement of Net-Demux

flow illustrates the control flow of the routing unit. The integration of Net-Demux increases the length of each path for the routing unit.

Switch Allocator (SA). In each cycle, the crossbar connections are established between input and output ports for flit traversal. SA schedules these connections. In Fig. 8, the timing arc (red dash-line) exhibits a critical path between three submodules: local arbiter, global arbiter, and masking logic. The local arbiter selects a winner among the competing Virtual Channels (VCs) at each input port. The output of local arbiter becomes the select line of MUX1 that selects only one wire consisting of output port information for the flit of winner VC.

The Net-Demux is placed in between local and global arbiters. A 1-bit select line *m* makes the selection of NoC network according to a type of message class (refer to Table 2). If *m* is '0,' it proceeds to NoC_1, else it proceeds to NoC_2 network. The competition between different inputs requesting for the same output physical channel is resolved by Global Round-Robin (G-RR) arbiter. Each input bit processed through G-RR arbiters corresponds to each output channel. The grants generated by global arbiter are used to set up the crossbar control registers.

The output of the global arbiter also feeds into the input of masking logic. Another input of masking logic is the V-bit input line from the local arbiter. The masking logic activates the next VC request in lieu of the recent VC. In the next section, we compare hardware implementation overhead of Net-Demux across different placements.

Table 3 Comparative changes in hardware component's architectures with different Net-Demux placements

Hardware components	NI	RU	SA	Comments
Input signals	N-bit input	K-bits input	K-bits input	N >> K
Demultiplexer	Single unit with N-bits	Single unit with K-bits	Single unit with K-bits	Total required four units
Encoder	None	Port encoder	Crossbar switch encoder	Negligible overhead
Adder	None	5 half adder (2 full & 1 half)	3 half adder	SA adder is more efficient
Multiplexer	None	Single unit with K-bits	Single unit with k-bits	Negligible overhead

Table 4 Gain in hardware metrics with different Net-Demux placements

Hardware metric	NI	RU	SA	Comments
Critical path delay	Placement increases path length	No impact	No impact	Router's critical path decided by crossbar so no impact of placement on other components
Number of Net-Demux units required	4	4	4	No change in the number of units
Power overheads	$4 \times N$	$4 \times K$	$4 \times K$	NI overhead is much higher
Area overheads	$4 \times N$	$4 \times K$	$4 \times K$	NI overhead is much higher

4.3 Comparative analysis between different placements

Initially, we compare the network interface versus router placement. In Table 3, we examine changes in the hardware component's architecture with different Net-Demux placement. Subsequently, the impact of placement on hardware metrics is compared in Table 4. Then, we compare placement within the router, i.e., RU versus SA.

Network Interface (NI) versus Router Placement. Net-Demux is placed in the data plane at the NI, whereas Net-Demux is placed on the control plane at the router. As $K << N$[11], the hardware component overhead is less for control plane placement as compared in Table 3. So the placement at the router's control plane is beneficial over placement at NI. The data plane placement of NI has more area and power overhead for the implementation of Net-Demux, as given in Table 4.

Placements at Routing Unit (RU) versus Switch Allocator (SA). The Net-Demux can be placed on the router either at the RU or SA. The architecture of Net-Demux varies with placement. Since each functional unit of the router performs a specific task, Net-Demux inputs and outputs vary accordingly, as compared in Table 3. For example, the RU computes the output port for the incoming header flit, whereas SA resolves the contention between channels and enables the switch to forward the flit across the crossbar. The encoder is different for both the placements, and therefore, the minimum number of required half adders is different. The architecture of Net-Demux is less complicated for placement at SA as compared to placement at RU. Though the input signals are of the same width for placements at RU and SA, the architecture of Net-Demux is less complicated for placement at SA.

[11] Where K and N are the number of input bits.

Table 5 Parameter configuration for hardware synthesis

Parameters	Configuration
Synthesis tool	Synopsis design compiler
Hardware language	System verilog
Hardware design	Register transfer level (RTL) synthesis
Cell technology	Low voltage threshold (LVT)
NoC networks	Single-NoC, dual-network-NoC and 2-network-NoC
Technology	32 nm
Clock frequency	2 GHz

Impact on Circuit Characteristics. Though, we have not explored the back-end impact of placement in details, we have presumed implications as follows:

1. *Input Signal Ordering:* This characteristic recommends postponing the introduction of input signals with a high transition rate.[12] The signal passes through the routing algorithm, then becomes the input of Net-Demux. The input signal transition rate for the RU is higher than the SA. Since the SA signal passes through one multiplexer and becomes the input for Net-Demux, the placement of Net-Demux at the SA is more efficient than placement at RU because a low transition rate signal enters early to the Net-Demux.
2. *Timing Delay:* The critical path is an important characteristic for timing delay. The critical path delay is affected by placement at NI. The router micro-architecture is a pipelined architecture. The crossbar of the router dominates the critical path delay. The Net-Demux placement at the RU and SA do not affect the router's critical path delay. Since the frequency need not change on placement, there is no impact on data arrival in each router cycle.
3. *Circuit Topology:* The optimization algorithms used by the fabrication tools decide the topology of the circuit. Therefore, the topology of the circuit changes with placement[13] and hence switching activity.[14] These are the primary factors that affect the dynamic power.

Impact on Scalability. The current technology breakthrough makes NoC sensitive to faults, severe congestion, traffic hotspot, and thermal issues. So routings of these processors are more complex. The complex routing circuits have high switching activity because of the number of digital units in every path. It is better to place Net-Demux at SA of the router.

[12] Refer to Sect. 3.2.

[13] refer to Sect. 3.1.

[14] Switching activity has two components (1) a static component—the function of the logic's topology, and (2) a dynamic component—the function of the timing behavior (glitching). We have not discussed the dynamic part in detail as it is out of the scope of the paper.

Table 6 Path slack variations in different signal paths across Net-Demux placements

Placement of Net-Demux at					
NI		RU		SA	
# Paths	ΔSlack[a] (10^{-3})	# Paths	ΔSlack[a] (10^{-2})	# Paths	ΔSlack[a] (10^{-2})
32	0–0.1	8	0.36–0.4	32	0.5–0.6
4	0.32–0.35	8	0.45–0.56	8	0.6–0.7
14	0.78–0.88	20	0.62–0.70	10	0.11–0.14
		14	0.72–0.78		

[a] The unit of slack measurement is nanoseconds

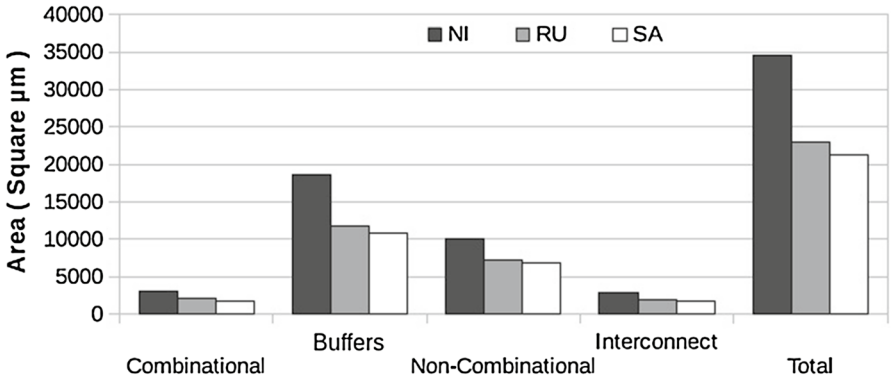

Fig. 9 Area comparison

5 Experimental results

In this section, we first discuss the hardware synthesis results to estimate the placement impact on power, area, and critical path delay of the circuit. Then, for performance evaluation, we run benchmarks in full system simulation mode.

5.1 Hardware synthesis

The parameter configuration for hardware synthesis is listed in Table 5. We have performed Register Transfer Level (RTL) synthesis using System Verilog to evaluate the hardware area, power, and critical path delay of the Routing Unit (RU) and Switch Allocator (SA) placement of Net-Demux. This is compared with conventional Network Interface (NI) placement. We synthesize using Synopsys Design Compiler with 32nm standard-cell libraries and LVT[15] cell technology. The target clock frequency is set to 2 GHz.

[15] Low Voltage Threshold [50].

Table 7 Area, power, and critical path delay comparison relative to single-NoC for different placements (the area, power, and critical path delay of single-NoC is normalized to one unit time)

Hardware metric	Single	Placement of Net-Demux					
		NI		RU		SA	
		NoC					
		Dual-net[a]	2-net[b]	Dual-net	2-net	Dual-net	2-net
Area	1	0.52	0.47	0.68	0.55	0.80	0.64
Total power	1	0.51	0.48	0.66	0.56	0.89	0.68
Static power	1	0.50	0.47	0.61	0.52	0.75	0.59
Dynamic power	1	0.51	0.48	1.13	0.96	1.46	1.04
Critical path delay	1	1.08	1	1.13	0.96	1	0.67

[a] Dual-network-NoC

[b] 2-network-NoC

We compare the slack values of a number of signal paths between NI, RU, and SA in Table 6. A slack shows the permissible delay of a cell activity without delaying overall circuit output. So, path slack is the difference between time of data arrival, and data requirement. If the data arrival coincides with the time of data requirement, slack is zero ideally. However, the paths that are close to zero slack values are critical. If a higher number of paths in a circuit are close to zero, a slight variation in the slack may render the circuit unreliable. The time when a signal arrives can vary due to variation in input data, temperature, voltage, and manufacturing defects. Hence, the design should ensure that all signals will arrive neither too early nor too late despite these variations.

The positive slack implies the arrival time of a cell may be postponed by slack value without affecting the overall delay of the circuit. It shows that data arrives before being required by the circuit. So the researchers emphasize on the restructuring of the schematic for fewer equally critical paths. Instead of looking at the slack for just the most critical path, the design should consider the entire distribution of slacks [51]. The NI's path slack value is in the order of 10^{-3} ns, which is very close to zero slack as compared to RU and SA, i.e., order of 10^{-2} ns. The rest of the positive slack values reduces the risk of violating the timing delay of a circuit [52].

Area. The area of different components of a circuit for various Net-Demux placements are compared in Fig. 9. The total area is the sum of four factors: Combinational, Noncombinational, Net Interconnect area, and Buffers. The combinational logic gates, like ANDs, ORs, etc., cover area due to logic cells. The noncombinational factors are registers. The net interconnect connects all cells. This net area is computed by the library of wire load models. The buffers are the primary contributor to the area as compared to other NoC components. The total gain[16] in the area is 39% with placement at SA and 36% with placement at RU over placement at NI, respectively.

[16] The actual area advantage would be more for SA over NI. We get these area results when we have skipped the virtual channels and virtual networks in the circuit.

Table 8 Improvement in area, power and critical path delay for different placements

Hardware metric	% Improvement in					
	Dual-net		2-net		2-net	
	Net-Demux placement at					
	NI		RU		SA	
	Gain over		Gain over		Gain over	
	Single	2-Net	Single	Dual-net	Single	Dual-net
Area	+ 48	– 10	+ 45	+ 19	+ 36	+ 20
Total power	+ 49	– 6	+ 44	+ 15	+ 32	+ 24
Static power	+ 50	– 6	+ 48	+ 14	+ 41	+ 21
Switching power	+ 49	– 6	+ 4	+ 15	– 4	+ 29
Critical path delay	– 8	– 8	+ 4	+ 15	+ 33	+ 33

Power and Critical Path Delay. We compare Net-Demux placements at NI in dual-network-NoC with placement at RU and SA in 2-network-NoC. We examine the area, power, and critical path delay of these placements with single-NoC.

The power dissipation is the sum of total static/leakage power and dynamic/switching power. The leakage power is dependent on the static current flow from voltage supply to ground in the absence of switching activity. It consists of (dis) charging of the capacitor and short circuit power. Thus, leakage power varies according to the switching activity of the circuit. Another important design metric is critical path delay because it determines the NoC frequency by estimating the worst path taken by the data.

In Table 7, we normalize the results of placements with respect to single-NoC. The values of design metrics are considered as '1' for single-NoC. We estimate the gain in dual-network-NoC and 2-network-NoC over single-NoC.

Table 8 is drawn from Table 7 to show the percentage improvement in the area, power, and delay for different placements. Here, we estimate the gain of metrics for different NoCs. Since dual-network-NoC already exists, the gains are compared with single-NoC and proposed 2-network-NoC. For example, the gain in area of dual-network-NoC over single-NoC is $(1 - 0.52)/1 = 0.48$ and over 2-network-NoC is $(0.47 - 0.52)/0.47 = -0.106$. Likewise, other metrics are computed. This computation is done for placement of Net-Demux at NI. Similarly, for placement of Net-Demux at RU, the gain of 2-network-NoC is compared with single-NoC and existing dual-network-NoC. For example, the gain in area of 2-network-NoC over single-NoC is $(1 - 0.55)/1 = 0.45$ and over dual-network-NoC is $(0.68 - 0.55)/0.68 = 0.191$. The gain of other metrics is computed in the same way for RU as well as SA. In Table 8, the values are represented in percentage wherein values after two decimal places are truncated for simpler representation. The '+' sign indicates the advantage, and '–' shows the drawback in the table. The area and static power are improved significantly for all three placements over single-NoC. Since the flit-width in dual-network-NoC and 2-network-NoC is half

Table 9 Simulation configuration

Parameters	Configuration
Simulator	Gem5 (Full system simulation)
NoC simulator	GARNET[a]
Cores	64 cores
ISA[b]	ALPHA
Topology	8×8 Mesh
NoC networks	Single-NoC, dual-network-NoC and 2-network-NoC
Routing	Dimension order XY
Caches	Three level of cache hierarchy
Cache coherence	Directory-based MESI[c] protocol
Power	ORION[d] 2.0
Benchmark	PARSEC[e] (medium workload, 16 threads)

[a] GARNET—a cycle accuRate iNtErconnection neTwork simulator

[b] ISA—Instruction set architecture

[c] Modified exclusive shared invalidation

[d] ORION—pOwer-peRformance InterconnectiOn Network simulator

[e] PARSEC—Princeton Application Repository for Shared mEmory Computers

of the single-NoC, the buffer size is determined in the proportion of flit-width. As buffers occupy major area and consume more static power [53], a significant gain is observed in these metrics.

However, Net-Demux placement at NI in dual-network-NoC has more consumption of area and static power as compared to 2-network-NoC. The number of NI links in dual-network-NoC is eight, whereas it is only four in 2-network-NoC. The less number of links implies less number of buffers and hence less area and static power for 2-network-NoC. The overall gain in the area and static power is observed with Net-Demux placement at SA in 2-network-NoC over dual-network-NoC.

The Net-Demux placement at NI also increases critical path delay, as shown in Table 8. The data arrival is delayed by 8% over single-NoC and 2-network-NoC. However, the increase in critical path delay at RU and SA does not impact the overall router frequency as the router is a pipelined[17] architecture.

The switching power varies across different placements because of the variation in input and output switching activity. As these results are calculated with the default wire load model having low traffic, we further evaluate the performance of different placements with PARSEC benchmark workload.

[17] The crossbar of the router dominates the critical path of the router.

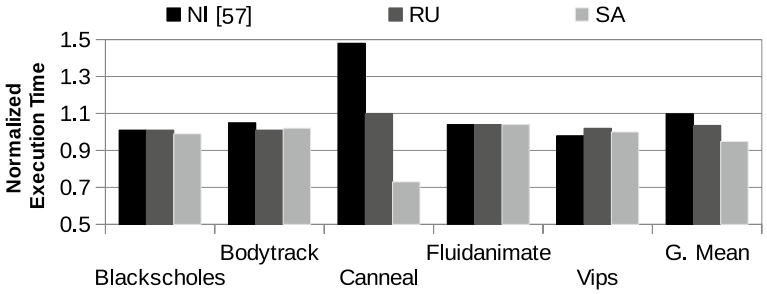

Fig. 10 Normalized execution time comparison among Net-Demux placements at Routing Unit (RU), Switch Allocator (SA), and Network Interface (NI) [57] (the execution time of single-NoC is normalized to one unit time)

5.2 Benchmark results

We have experimented on PARSEC benchmarks using Gem5 simulator. The configuration details of the simulation are listed in Table 9. We compare the execution time of Net-Demux placement at router with placement at NI using the PARSEC benchmark. For full system simulation, we have used Gem5 [13, 14, 54] integrated with the GARNET [15, 55] NoC simulator. A Linux 2.6.27 kernel image is booted with ALPHA instruction set architecture. For power results, GARNET is integrated with ORION [16, 56].

The volume of messages varies across message classes of different PARSEC applications [17]. The PARSEC comprises the data-parallel benchmarks, i.e., *blackscholes, bodytrack, fluidanimate*, and *vips*, whereas the *canneal* benchmark is unstructured. These benchmarks have different granularity in data like *fluidanimate*, and *canneal* are fine-grained, whereas *blackscholes* is coarse-grain. The *bodytrack* and *vips* are moderate. So these benchmarks show different impacts on performance.

We compare the execution time of Net-Demux placement at RU and SA with conventional placement at NI [57] in Fig. 10. Minor degradation of 10% and 4% is observed in execution time for placements at NI and RU, respectively. However, 6% improvement is achieved with placement at SA over single-NoC. In comparison with placement at NI, placements at RU and SA are, respectively, 6% and 14% more efficient.

6 Conclusions

We propose the placement of the network selector hardware unit in the control plane of the router to improve the power efficiency of Multi-NoC instead of conventional network interface placement. Net-Demux architecture is modified according to the circuit where it is placed. In the control plane, wires' bus-width and registers' size are quite small as compared to the data plane. Net-Demux placement at switch allocator has the lowest hardware implementation overhead in Net-Demux architecture compared to the placement at routing unit and network interface. Placement on the

control plane does not impact critical path delay as the router is a pipelined architecture. Contrary, Net-Demux placement at network interface increases critical path delay up to the critical path length. On comparing synthesis results, we observe that Net-Demux placement at switch allocator improves 20% area, 21% static power, 29% dynamic power, and 33% critical path delay of the circuit over conventional placement. With the PARSEC benchmark, the execution time with Net-Demux placement at switch allocator improves by 6% over single-NoC. Subsequently, the Net-Demux placement at routing unit and switch allocator improves execution time by 6% and 14%, respectively, over placement at network interface.

Presently, our work is limited to two networks NoC. In future, we shall implement the proposed idea of placement at router for more than two NoC networks. The benefits could increase with increasing number of NoC networks, since the number of outputs for Net-Demux shall be in proportion to the number of NoC networks. Another limitation is that we have explored the proposed placement only for five-stage pipelined router architecture. This can be further explored for one-/two-/three-/four-stage pipelined architecture. It will also be interesting to explore the placement impact on critical path delay of different pipelined router architectures. However, on reducing router pipeline stages, the bypassing and speculation techniques are used to transfer the flit. These techniques are effective to improve network throughput in low network traffic. Since bypassing or speculation failure results in a longer critical path delay, the exploration of placement with reduced pipelined router architecture is suitable only for low traffic workloads. This approach can also be implemented for wireless Multi-NoC as our approach reduces power and it is a big concern in wireless NoC [58]. The multi-chip processor architectures also require traffic distribution unit between multiple chips. So our proposed idea of placement can be repurposed for multiple chips [59].

References

1. Reddy GT, Reddy MPK, Lakshmanna K, Kaluri R, Rajput DS, Srivastava G, Baker T (2020) Analysis of dimensionality reduction techniques on big data. IEEE Access 8(2020):54776–54788
2. Jain A, Laxmi V, Tripathi M et al (2019) S2DIO: an extended scalable 2D mesh network-on-chip routing reconfiguration for efficient bypass of link failures. J Supercomput 75:6855–6881
3. Zhan J, Xie Y, Sun G (2014) NoC-sprinting: interconnect for fine-grained sprinting in the dark silicon era. In: Proceedings of the 51st Annual Design Automation Conference, pp 1–6
4. Flores A, Aragón JL, Acacio ME (2008) An energy consumption characterization of on-chip interconnection networks for tiled CMP architectures. J Supercomput 45:341–364
5. Xiang X, Sigdel P, Tzeng N-F (2020) Bufferless network-on-chips with bridged multiple subnetworks for deflection reduction and energy savings. IEEE Trans Comput 69(4):577–590
6. McKeown M, Fu Y, Nguyen T, Zhou Y, Balkind J, Lavrov A, Shahrad M, Payne S, Wentzlaff D (2017) Piton: a manycore processor for multitenant clouds. IEEE Micro 37(2):70–80
7. Sodani A, Gramunt R, Corbal J, Kim H-S, Vinod K, Chinthamani S, Hutsell S, Agarwal R, Liu Y-C (2016) Knights landing: second-generation intel Xeon Phi product. IEEE Micro 36(2):34–46
8. Daya BK, Chen C-HO, Subramanian S, Kwon W-C, Park S, Krishna T, Holt J, Chandrakasan AP, Peh L-S (2014) SCORPIO: a 36-core research chip demonstrating snoopy coherence on a scalable mesh NoC with in-network ordering. In: Proceedings of 41st international symposium on computer architecture (ISCA), pp 25–36

9. Wentzlaff D, Griffin P, Hoffmann H, Bao L, Edwards B, Ramey C, Mattina M, Miao C-C, Brown JF III, Agarwal A (2007) On-chip interconnection architecture of the tile processor. IEEE Micro 27(5):15–31
10. Gratz P, Kim C, Sankaralingam K, Hanson H, Shivakumar P, Keckler S, Burger D (2007) On-chip interconnection networks of the TRIPS chip. IEEE Micro 27(5):41–50
11. Taylor MB, Kim J, Miller J, Wentzlaff D, Ghodrat F, Greenwald B, Hoffman H, Johnson P, Lee J-W, Lee W, Ma A, Saraf A, Seneski M, Shnidman N, Strumpen V, Frank M, Amarasinghe S, Agarwal A (2002) The raw microprocessor: a computational fabric for software circuits and general-purpose programs. IEEE Micro 22(2):25–35
12. Wang Z, Ma S, Huang L, Lai M, Shi W (2015) Network-on-chip customizations for message passing interface primitives. Morgan Kaufmann, Networks-On-Chip, Burlington, pp 285–315
13. Semakin AN (2021) Simulation of a multi-core computer system in the gem5 simulator. In: AIP Conference Proceedings. 2318, 1. AIP Publishing LLC
14. France L, Bruguier F, Mushtaq M, Novo D, Benoit P (2021) Implementation of Rowhammer effect in gem5. In: 15ème Colloque National du GDR SoC^2
15. Das A, Kumar A, Jose J, Palesi M (2021) Opportunistic caching in NoC: exploring ways to reduce miss penalty. IEEE Trans Comput 70(06):892–905
16. Dahir N, Karkar A, Palesi M, Mak T, Yakovlev A (2021) Power density aware application mapping in mesh-based network-on-chip architecture: an evolutionary multi-objective approach. Integration 81(2021):342–353
17. Bienia C, Kumar S, Singh JP, Li K (2008) The PARSEC benchmark suite: characterization and architectural implications. In: International Conference on Parallel Architectures and Compilation Techniques (PACT), pp 72–81
18. Yadav S, Laxmi V, Gaur MS, Kapoor HK (2019) Late breaking results: improving static power efficiency via placement of network demultiplexer over control plane of router in multi-NoCs. In: 2019 56th ACM/IEEE Design Automation Conference (DAC), pp 1–2
19. Yadav S, Laxmi V, Gaur MS (2020) Multiple-NoC exploration and customization for energy efficient traffic distribution. In: 2020 IFIP/IEEE 28th International Conference on Very Large Scale Integration (VLSI-SOC), pp 200–201
20. Abts D, Jerger NDE, Kim J, Gibson D, Lipasti MH (2009) Achieving predictable performance through better memory controller placement in many-core CMPs. In: Proceedings of the 36th annual international symposium on computer architecture (ISCA). ACM, pp 451–461
21. Zhao H, Zhang F, Chen L, Lu M (2021) A method of fast evaluation of an MC placement for network-on-chip. J Circuits Syst Comput 30(7):2150115
22. Hung W, Addo-Quaye C, Theocharides T, Xie Y, Vijaykrishnan N, Irwin MJ (2004) Thermal-aware IP virtualization and placement for networks-on-chip architecture. In: Proceedings of the International Conference on Computer Design. IEEE, pp 430–437
23. Hu J, Marculescu R (2003) Energy-aware mapping for tile-based NoC architecture under performance constraints. In: Proceedings of the Asia and South Pacific Design Automation Conference (ASP-DAC). IEEE, pp 233–239
24. Srinivasan K, Chatha K (2005) A technique for low energy mapping and routing in network-on-chip architectures. In: Proceedings of the international symposium on low power electronics and design. IEEE, pp 387–392
25. Robaei M, Zhao H (2019) Broadcast-based hybrid wired-wireless NoC for efficient data transfer in GPU of CE systems. IEEE Consum Electron Mag 8(6):62–67
26. Kullu P, Ar Y, Tosun S, Ozdemir S (2020) Mapping application-specific topology to mesh topology with reconfigurable switches. IET Comput Digital Tech 14(1):9–16
27. Bayar S, Yurdakul A (2016) An efficient mapping algorithm on 2-D mesh network-on-chip with reconfigurable switches. In: 2016 International conference on design and technology of integrated systems in nanoscale Era (DTIS), Istanbul, Turkey, pp 1–4
28. Baharloo M, Aligholipour R, Abdollahi M, Khonsari A (2020) ChangeSUB: a power efficient multiple network-on-chip architecture. Comput Electr Eng 83(2020):106578
29. Shahidinejad A, Fathi S (2018) Wireless-assisted multiple network on chip using microring resonators. Microprocess Microsyst 63(2018):190–198
30. Ejaz A, Papaefstathiou V, Sourdis I (2018) DDRNoC: dual data-rate network-on-chip. ACM Trans Archit Code Optim 15(2):1–24
31. Rad F, Reshadi M, Khademzadeh A (2021) A novel arbitration mechanism for crossbar switch in wireless network-on-chip. Cluster Comput 24(2021):1185–1198

32. Ma W, Gao X, Gao Y, Yu N (2021) A latency-optimized network-on-chip with rapid bypass channels. Micromachines 12(6):621
33. Neelkamal, Yadav S, Kapoor HK (2019) Lightweight message encoding of power-gating controller for on-time wakeup of gated router in network-on-chip. In: 2019 9th International symposium on embedded computing and system design (ISED), pp 1–6
34. Balfour J, Dally WJ (2006) Design tradeoffs for tiled CMP on-chip networks. In: Proceedings of the 20th Annual International Conference on Supercomputing. ACM, pp 187–198
35. Carara E, Calazans N, Moraes F (2007) Router architecture for high-performance NoCs. In: Proceedings of the 20th Annual Conference on Integrated Circuits and Systems Design (SBCCI). ACM, pp 111–116
36. Yoon YJ, Concer N, Petracca M, Carloni LP (2013) Virtual channels and multiple physical networks: two alternatives to improve NOC performance. IEEE Trans Comput Aided Des Integr Circuits Syst 32(12):1906–1919
37. Grot B, Hestness J, Keckler SW, Mutlu O (2009) Express cube topologies for on-chip interconnects. In: International symposium on high-performance computer architecture (HPCA). IEEE, pp 163–174
38. Kumar P, Pan Y, Kim J, Memik G, Choudhary A (2009) Exploring concentration and channel slicing in on-chip network router. In: Proceedings of 3rd ACM/IEEE international symposium on networks-on-chip. pp 276–285
39. Gómez C, Gómez ME, López P, Duato J (2008) Exploiting wiring resources on interconnection network: Increasing path diversity. Working on Parallel, Distributed, and Network-Based Processing. pp 20–29
40. Teimouri N, Modarressi M, Tavakkol A, Sarbazi-azad H (2011) Energy-optimized on-chip networks using reconfigurable shortcut paths. In: Conference on Architecture of Computing Systems. pp 231–242
41. Noh S, Ngo V-D, Jao H, Choi H-W (2006) Multiplane virtual channel router for network-on-chip design. In: First International Conference on Communications and Electronics, pp 348–351
42. Gilabert F, Gómez ME, Medardoni S, Bertozzi D (2010) Improved Utilization of NoC channel bandwidth by switch replication for cost-effective multi-processor systems-on-chip. In: Proceedings of fourth ACM/IEEE international symposium on networks-on-chip (NOCS), pp 165–172
43. Xiang X, Sigdel P, Tzeng N (2020) Bufferless network-on-chips with bridged multiple subnetworks for deflection reduction and energy savings. IEEE Trans Comput 69(4):577–590
44. Morgan AA, Hassan AS, El-Kharashi MW, Tawfik A (2020) NoC^2: an efficient interfacing approach for heavily-communicating NoC-based systems. IEEE Access 8:185992–186011
45. Volos S, Seiculescu C, Grot B, Pour NK, Falsafi B, Micheli GD (2012) CC-NoC: specializing On-chip interconnects for energy efficiency in cache-coherent servers. In: Proceedings of Sixth IEEE/ACM international symposium on networks on chip (NoCS), pp 67–74
46. Rabaey JM (1996) Digital integrated circuits: a design perspective. Prentice-Hall Inc., New York
47. Kang KW, Samsung Electronics Co Ltd (2005) Layout structures of data input/output pads and peripheral circuits of integrated circuit memory devices. U.S. Patent 6, 847, 576
48. Huang T-W, Lin C-X, Wong MDF (2021) OpenTimer v2: a parallel incremental timing analysis engine. IEEE Des Test 38(2):62–68
49. Das R, Narayanasamy S, Satpathy SK, Dreslinski RG (2013) Catnap: energy proportional multiple network-on-chip. In: Proceedings of the 40th annual international symposium on computer architecture (ISCA). ACM, pp 320–331
50. Bokhari H, Javaid H, Shafique M, Henkel J, Parameswaran S (2015) SuperNet: multimode interconnect architecture for manycore chips. In: Proceedings of the 52nd Annual Design Automation Conference (DAC), vol 85. ACM, pp 1–6
51. Bai X, Visweswariah C, Strenski PN, Hathaway DJ (2002) Uncertainty-aware circuit optimization. In: Proceedings 2002 Design Automation Conference. (IEEE Cat.No.02CH37324), pp 58–63
52. Fung R, Betz V, Chow W (2008) Slack Allocation and Routing to Improve FPGA Timing While Repairing Short-Path Violations. IEEE Trans Comput Aided Des Integr Circuits Syst 27(4):686–697
53. Dally W, Towles B (2003) Principles and practices of interconnection networks. Morgan Kaufmann Publisher, London
54. Binkert N, Beckmann B, Black G, Reinhardt SK, Saidi A, Basu A, Hestness J, Hower DR, Krishna T, Sardashti S, Sen R, Sewell K, Shoaib M, Vaish N, Hill MD, Wood DA (2011) The gem5 simulator. SIGARCH Comput Archit News 39(2):1–7

55. Agarwal N, Krishna T, Peh L-S, Jha NK (2009) Garnet: a detailed on-chip network model inside a full-system simulator. In: IEEE international symposium on performance analysis of systems and software (ISPASS). IEEE, pp 33–42
56. Kahng AB, Li B, Peh L-S, Samadi K (2012) ORION 2.0: a power-area simulator for interconnection networks. IEEE Trans Very Large Scale Integr (TVLSI) Syst 20(1):191–196
57. Miguel JS, Jerger NE (2015) Data criticality in network on chip design. In: Proceedings of the ninth IEEE/ACM international symposium on network on chip (NOCS), pp 28–30
58. Baharloo M, Khonsari A (2018) A low-power wireless-assisted multiple network-on-chip. Microprocess Microsyst 63(2018):104–115
59. Dasari UK, Temam O, Narayanaswami R, Woo D H, Google LLC (2021) Apparatus and mechanism for processing neural network tasks using a single chip package with multiple identical dies. U.S. Patent 10, 936, 942