

Received August 12, 2021, accepted September 1, 2021, date of publication September 9, 2021, date of current version September 20, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3111294

SB-Router: A Swapped Buffer Activated Low Latency Network-on-Chip Router

MONIKA KATTA¹, T. K. RAMESH¹, (Member, IEEE), AND JUHA PLOSILA², (Member, IEEE)

¹Department of Electronics and Communication Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Bengaluru Campus, Bengaluru, Karnataka 560035, India

²Autonomous Systems Laboratory, Department of Future Technologies, Faculty of Science and Engineering, University of Turku, 20014 Turku, Finland

Corresponding author: Monika Katta (er.monikakatta@gmail.com)

ABSTRACT Switch Allocation (SA) holds a critical stage in Network-on-Chip (NoC) routers, its performance gets affected adversely due to Head-of-Line (HoL) blocking. In traditionally used Input-Queued Routers (IQR), packets are arranged in a particular order in each Virtual Channel (VC). This implementation is vulnerable to HoL blocking, as the switch allocator can allocate only those packets which are available at the head in a VC. In this paper, Swapped Buffer (SB) Router architecture is proposed to schedule packets in input buffers by using SB registers. The VCs are designed as SBs, this allows the packets stored in SB registers along with the head packet of VC to participate in SA. The concept of the SB register minimizes the conflicts in SA and thus reduces HoL blocking, therefore improves the performance of NoC. This paper proposes a priority mechanism to prioritize the non-head packets as compared to head packets in case of conflict between them. Two methods have been proposed in this paper, to enhance the performance of the NoC router. First, a VC allocation technique is proposed to optimize the order of packets in the input buffer. Next, SB-Router is combined with the Fill VC allocation technique to further enhance the performance of NoC routers. The performance of the proposed router is evaluated and the experimental results indicate that our design achieves latency improvement of 68.75% over (Time-Series) TS-Router for uniform traffic at the injection rate of 0.42 flits/cycle for a 64 node mesh network with moderate power consumption and area usage. The performance improvement in packet latency for traces from Princeton Application Repository for Shared-Memory Computers (PARSEC) has also been evaluated. With the achieved reduction in latency, the proposed method has the potential to serve high-speed operations while mapping different applications on multiple core architectures.

INDEX TERMS Network-on-chip, packet scheduling, switch allocation, virtual channel.

I. INTRODUCTION

Network-on-Chip, (NoC) the emerging technology has become popular in interconnecting multiple cores on a chip [1]. Multi-core processors use NoC to interconnect thousands of cores rather than shared buses or point-to-point interconnect wires [2]. Switch Allocation (SA) is used to assign output ports to input ports [3]. SA also makes sure that there is no conflict in the flit transit [4]. In modern NoCs, the most popular router architecture is the Input Queued Router (IQR) [5], [6]. Many works recently have shown that the SA matching efficiency can be improved for the IQRs by using different strategies [7]. These strategies attempt to efficiently improve the matching capacity of SA

by taking the request based on time series [8] or by adding the dedicated circuitry to bypass the packets [9]. However, the queues at the Virtual Channel (VC) still face the problem of Head of-Line (HoL) blocking. HoL blocking impacts the overall performance severely in on-chip networking than in off-chip networking [10]. This is because of the high number of short-length packets in NoCs. The test results for Princeton Application Repository for Shared-Memory Computers (PARSEC) benchmarks, shows that on an average 78.7% of the packets are single flit packets [11]. This, in turn, means that the packet at the head can block more packets behind it.

End congestion due to HoL blocking is one of the main reasons for performance degradation in IQR [12]. The traditional IQRs have input queues organized as VCs [13]. Each VC can be used as a buffer to keep incoming packets. Switch allocator

The associate editor coordinating the review of this manuscript and approving it for publication was Mario Donato Marino .

can allocate only those packets which are at the head of a VC. Therefore, when the packet at the head of a VC cannot be allocated quite a few times, the subsequent packets in the same VC lack the chance of participation in SA. If the subsequent packets would have got the chance to participate in SA, they would have succeeded. This leads to HoL blocking. As shown in Fig. 1, the first and the third VC are competing to send the packet to output port 1. In the SA stage, if the switching fabric chooses to transfer the packet from the first VC, the packet from the third VC cannot be transferred to the output port in the same time slot. Here, the third VC is blocking a packet destined to output port 4, that is available for receiving the packet. This is known as HoL blocking. The head packet cannot progress due to congestion and the subsequent packets that are destined to the output ports available to receive the packets, are blocked.

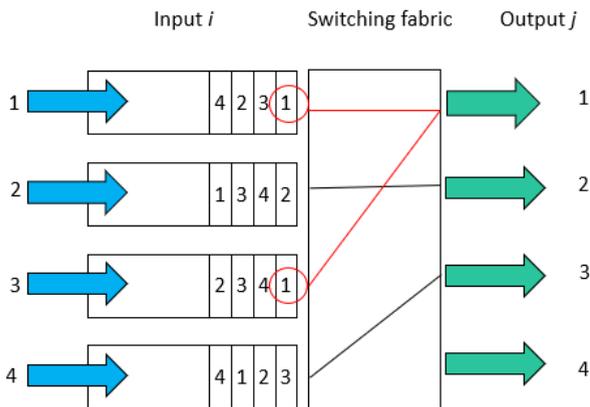


FIGURE 1. An example showcasing head of line blocking in network-on-chip. The third VC is blocking a packet destined to output port 4, that is available for receiving the packet.

NoC real-time applications demand low latency for data transfer. To meet this requirement, a router architecture is required that utilizes buffers to their optimum level and prioritizes the packets which are not the head packets to participate in SA. In this paper, SB-Router architecture is proposed that uses SB registers to hold the status of the non-head packets in VC. The length of SB is decided online depending upon the flits buffered in a VC. This opens more options for the packets to participate in SA. The performance of the proposed router is evaluated and compared with baseline router and other state-of-the-art routers in terms of latency, power and area over both synthetic and application traffic patterns. The contributions of the proposed SB-Router for NoC are as follows:

- SB-Router architecture is proposed to schedule packets in input buffers by using SB registers. The VCs are designed as SBs, this allows the packets stored in SB registers along with the head packet of VC to participate in SA
- A priority mechanism is proposed in this paper to prioritize the non-head packets to get the option of allocation

as compared to head packets, in case of conflict between them. A VC allocation technique is implemented to optimize the order of packets in the input buffer.

- The proposed SB-Router is combined with the Fill VC allocation technique [33] and the implementation results show further improvement in latency.
- The performance bounds of SB-Router are explored by considering SB-Router-1 to SB-Router-4. As compared to the baseline router, SB-Router-1 shows a 68.75% improvement in latency under mesh topology, for uniform traffic at the injection rate of 0.42 flits/cycle.

The paper is organized into six sections. In section II, related work is discussed. Section III describes the baseline router architecture. Section IV demonstrates the detailed design for the proposed SB-Router. In section V, experimental setup and performance evaluation is done. In section VI, the conclusions and future scope for further research are brought out.

II. RELATED WORK

In NoC, packets move from the source node to the destination node traversing many intermediate routers. With the increase in the number of nodes, the number of intermediate routers increases, leading to an increase in transmission latency. Thus, reducing latency has become a major concern for designers. To reduce latency, many strategies have been used in earlier works [14]. The classification of strategies used to reduce latency is illustrated in Fig. 2. Each type of strategy is discussed in this section.

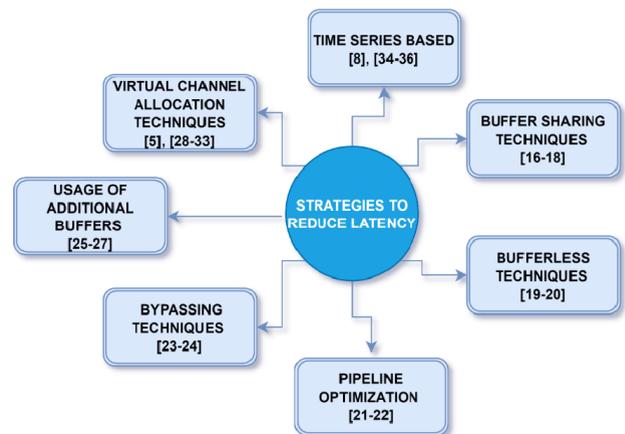


FIGURE 2. Classification of popular strategies used to reduce latency in network-on-chip.

These strategies includes buffer sharing techniques, bufferless techniques, pipeline optimization, bypassing techniques, usage of additional buffers, VC allocation methods and time-series-based methods.

On-chip routers use buffers at input and output ports to store packets temporarily when congestion occurs in the output physical channels. Input Buffered Routers (IBR) store packets in buffers, at the time of congestion [15]. This helps

in saving on to link bandwidth. Authors in [16] have proposed buffer sharing techniques to optimize resource utilization and throughput. However, the design of a complex router with two crossbars and the timestamp-based flow control contributes to high zero load latency. Shared queues, maximizes buffer utilization by allowing the sharing of multiple buffer queues among input ports. However, this contributes to power overhead of 4% and area overhead of 16% as compared to typical VC routers, respectively [17]. Authors in [18] have proposed Elastistore, which uses shared buffers to optimize the available buffer space usage without compromising on performance. However, the allocation logic needs to be simplified to optimize the number of pipeline stages.

Bufferless routing methods discussed in papers [19] and [20], remove buffers from the routers to reduce area usage and power consumption. This leads the network to consume more link bandwidth and increase latency, especially when the packet injection rate is high.

Short-Path [21], a pipelined router architecture achieves high-speed implementations by parallelizing as much as possible. This design is susceptible to use large queues. In application-specific NoC [22], both positions of the routers and the route for each communication trace of the application can be adjusted to suit the requirements. However, it increases the complexity of the system by using a power calculator.

Bypass-based low latency NoC router [23] combines look-ahead routing and parallel switch traversal. It takes only one or two stages to transfer one flit. This design provides good support to a low congestion network. The Look-ahead by-pass router architecture [24], combines many techniques to reduce latency. However, this design relies on the fact that output ports of the router having little contention are often free at low network loads. It also has conditional dependability for pipeline bypassing.

Roundabout router [25] inspired by real-life multi-lane roundabouts, distributes the input ports only to the primary lanes, while the secondary lanes are exploited only whenever congestion occurs. However, it requires an additional buffer. In the Augmented VC buffer design [26], the width of the private VCs is reduced at each port and the trace buffer is used as backup storage. However, it requires additional buffers to store the status of trace buffers for reusing which, in turn, adds to latency. In packet chaining reservation protocol [27], authors maintain more than one packet chaining for each flow to be sent to the network speculatively but small flows are treated preferentially.

To reduce latency, it is required to overcome HoL blocking from the queues at VC. The existing methods include conservative Virtual Channel Allocation (VCA) [28]. In this method, packets are reallocated to a VC only when it is empty. Therefore, at a time only one packet will be allocated to a VC at max, eliminating HoL blocking. However, performance degrades due to low buffer utilization. Other improvements include an aggressive VCA strategy [5], this design reallocates a packet to a VC even when it is not empty.

Thus, efficiently utilizes VC buffers and improves SA. This also showcase the importance of eliminating HoL blocking. An insignificant method would be to allocate more VCs one after another to a router. This will improve HoL blocking but the major disadvantage of this approach would be an increase in latency. The book-keeping will require additional buffers. In the Scheduling Algorithm iSLIP [29] the authors have aimed at optimizing the matches per allocation cycle at the cost of complex design. It requires multiple iterations and therefore overhead of power and area consumption is observed. Virtual input crossbar discussed in paper [30] allows flit transmission from multiple VCs at the input port, in the same time slot. However, the packets in VC of the input buffer are queued. In [31], authors have suggested to pipeline SA stage by dividing them into sub-stages. Another solution is output queued routers, they maintain buffer queues at the output port which are free from HoL blocking but increases write bandwidth when multiple flits are waiting for the same output port. Reorder-Buffer [RoB] router targets low latency by avoiding HoL blocking but the design increases the router's overall critical path delay [32]. In [33] we proposed latency improvement and mitigated HoL blocking by using Fill VC allocation for NoC. Fill VC allocation reduces the number of pipeline stages to reduce latency. The VCA mechanism optimizes the usage of buffer and improves the latency in NoC. This design doesn't use any priority mechanism for arbitration.

The time series-based strategies make use of allocation information. In Pseudo-Circuit [34] and Packet-Chaining [8], the benefits of adding previous allocation information to current allocation have been explored. However, small packets can still increase waiting flits at the head of a VC. Adaptive-effort pipelined switch allocator [35] uses virtual output queued router for allocation at the cost of power and area overhead. TS-Router [36] adds next allocation information to the current allocation but the anticipatory evacuation policy is effective only at low loads when input queue occupancy is low and thus evacuation is feasible.

The size of buffers available at the input port affects the performance of on-chip routers adversely. Several works have shown that a large buffer consumes a significant amount of power and area of a router [37]–[39]. Taking into account the cost of chip area and power [40], NoC routers restrict the count of VCs. Existing techniques fill these buffers in a VC and therefore face HoL blocking. To eliminate HoL blocking, a technique is proposed in this paper to organize packets in each VC. The non-head packets are also given a fair chance to participate in SA along with the head packets. This may increase the matching efficiency in SA. A priority mechanism is proposed for arbitration in NoC to reduce the congestion and network latency in routers. SA plays a key role in dominating the latency of packets in NoC. The matching efficiency needs to be optimized to achieve improvement in latency. The packets should progress from the source node towards their destination node, rather than waiting in

the buffers for their respective turn. The performance of the proposed router is compared with baseline router and other state-of-the-art routers in terms of latency, power and area over both synthetic and application traffic patterns. The details of baseline router architecture is discussed in the next section.

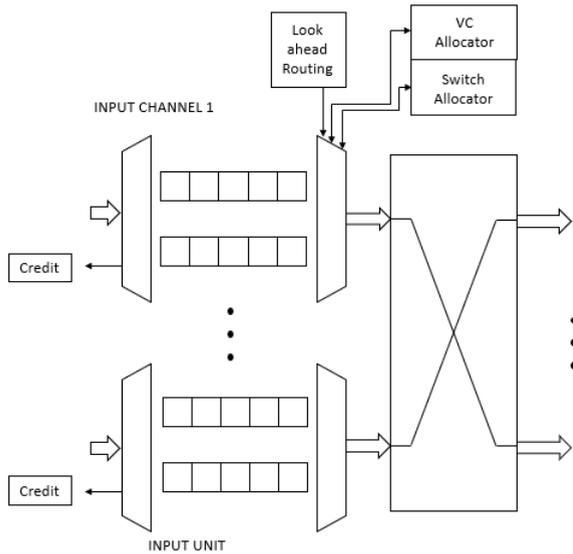


FIGURE 3. Baseline router architecture. The basic router architecture for SB-Router and all the other routers considered for comparison of results.

III. BASELINE ROUTER ARCHITECTURE

The SB register is used to swap the packets in VC at the input port to achieve increased matching efficiency in SA. The proposed design can be applied to other IQRs [5] easily, as the design fits well with the router architecture. The general router architecture popularly used in NoCs is chosen as our baseline router architecture. As shown in Fig. 3, the baseline router selected here is a state-of-the-art router [41] having five pipeline stages. The optimized router adopting look-ahead routing [42] and speculative SA [41] with only 3 pipeline stages is used here. Each pipeline stage is explained here. The first and foremost function is buffering of a flit. Whenever a flit comes through a channel, it occupies a buffer. The second task is Route Computation (RC), the output port is found for the flit stored in the buffer. The process of finding the output port for an incoming packet is called the routing or RC. It is done only for the head flit and the body flits and the tail flit follows the same route has been assigned to the head flit. Look-ahead routing performs RC one hop in advance and hence reduces one pipeline stage. The third task is called VCA. The process of reserving a buffer in the next downstream router is called VCA. To ensure flow control, handshaking signals are used between adjacent routers. A packet that has to go through the north output port after getting an update from the north neighbor reserves a buffer in it. The next step is SA, when multiple flits are competing for the same output port, one of them has to be chosen. It is

an arbitration process and once arbitration is over, the flits travel through the switch. In the switch traversal step, packets travel through the switch at any given clock cycle; at most 5 flits can travel through the switch one going to the east output port; one from the west; one from the north; one from the south and one can go through the processing element. Speculative SA [41] reduces one pipeline stage by performing SA in parallel with VCA. Based upon SA, switch traversal takes place. All these operations happen inside the router. Once switch traversal completes, the switch is connected to the link and it is called the link traversal, after which the router is pipelined. Traditional VC flow control [13] is used here. Multiple VCs per input port account for the whole buffer of an input port. These VCs share the bandwidth of the input port. Each VC acts as a first in first out queue. For all the routers namely, SB-Router, RoB router, Packet-Chaining and TS-Router the baseline architecture is illustrated in Fig. 3.

IV. SB-ROUTER DESIGN

SB-Router uses SB registers to minimize HoL blocking and to maximize the matching efficiency in SA. In this section, first, the description of SB-Router and its scheduling process is given with an example. Next, the concept of online SB is introduced. Then, a priority mechanism to prioritize the packets for SA is discussed. Next, VC buffer scheduling is discussed. Finally, two methods for SB-Router performance enhancement are discussed.

A. SB-ROUTER ARCHITECTURE

The SB-Router is responsible for scheduling packets in the input port. Fig. 4 shows the format of SB registers. Fig. 5 illustrates the steps to add information of a blocked packet to the SB register.



FIGURE 4. Format of SB registers. The four fields depict the information stored in SB registers.

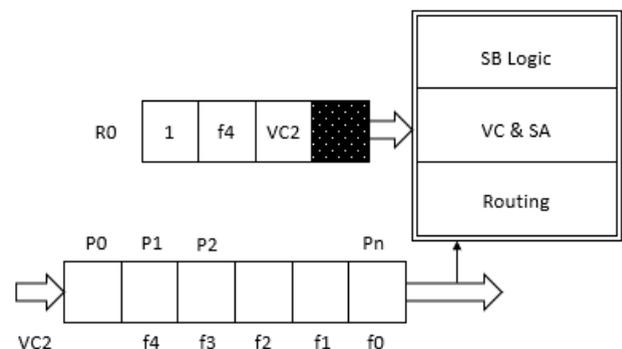


FIGURE 5. Steps to add information of a blocked packet to SB register.

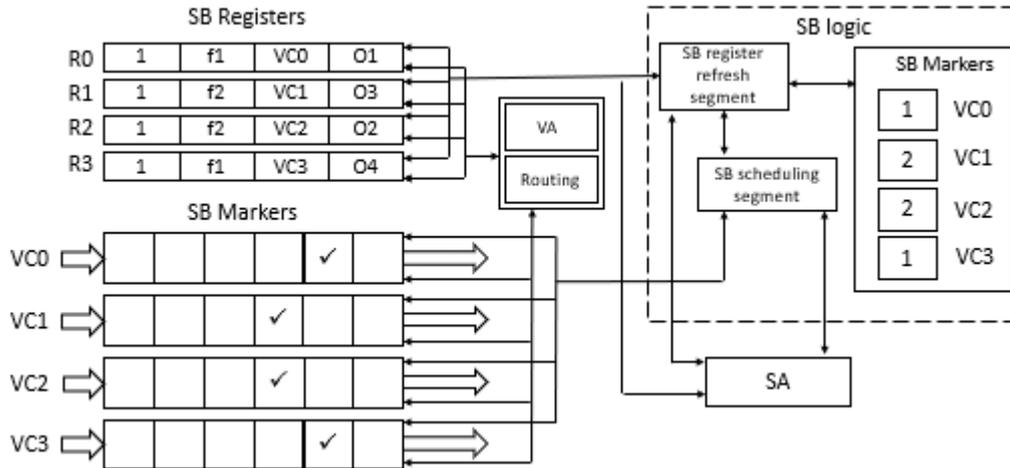


FIGURE 6. Detailed architecture of SB-Router.

The SB registers are available at the input port to hold the status of selected packets. It is assumed that it takes one cycle to add packet information to the SB register. A packet whose information is added to the SB register is regarded to be stored in the SB register. The count of all the flits of packets whose information is added to SB registers for a VC gives the length of SB of that VC. The online SB is used in each VC. The length of SB in each VC is decided from the count of all the flits buffered in that VC. Such a decision is performed only when an SB register is empty. When an SB register is empty, SB circuitry selects a VC having more flit count in its buffer to increase the length of its SB. The VC having more count of buffered flits can use more SB registers. The length of SB in this VC will be more than others. The packets in the SB register can now participate in SA, along with the packets at the head of VC, thereby giving more opportunities to packets to participate in SA.

As shown in Fig. 4, SB registers have four segments which include information about the head flit of a packet, VC number of that packet, packet position in the original VC (packet index) and its validity in terms of its usage (valid). SB registers store the entire information about head flit so that there is no need to access the buffers during the SA stage. It is assumed that the size of the SB register is fixed and is independent of the size of the packet. The size of the SB register is calculated from the sum of all 4 fields in the SB register. For instance, in a system with 64-bit flits having 4 VCs each of 8 flits long, the SB register will be 70 bits long. It is the sum of 64 bits (head flit), 2 bits required to store the VC number of that packet, 3 bits required to store the packet index and 1 bit for valid.

Referring to Fig. 5, suppose that packet P1 in VC2 is decided to be added to SB register R0. Its location information and head flit are stored carefully in the SB register. In R0, the head flit can participate in RC and VCA. Head packet Pn in VC2 and packet P1 after performing RC and VCA,

wait for SA. Here, it becomes really important to decide the priority mechanism among the head packets in VC and the non-head packet stored in SB register R0. The proposed priority mechanism is explained later in this paper. We give priority to SB register R0, so packet P1 is chosen first. In case of failure, the head packets stored in VC are given a chance. Thus, P1 has a higher priority for SA. After the SA stage is completed, each input port is assigned to an output port, even though the selected packet of this input port is a head packet or subsequent packet.

Fig. 6 illustrates the architecture of the input port. SB registers and SB logic circuitry is added to the input port to implement the proposed SB-Router architecture. The three main elements in SB logic are SB register refresh segment, SB scheduling segment and SB markers. SB markers point to the boundary of fixed units and other units available for swapping in a VC. SB register refresh segment adds information of packets to SB register and changes the SB marker accordingly. SB register refresh segment also refreshes the contents of the SB register after the respective packet passes the SA stage. SB scheduling segment performs packet scheduling based on SA. If there are no packets from SB registers, getting passed in SA then the SB scheduling segment passes on the information to the switch. The switch directly allocates head packets of VC through the crossbar. If SB register request is allocated then the SB scheduling segment needs to perform scheduling of packet from the contents of SB registers. SB scheduling segment can get the location information of the packet from the corresponding SB register. Such information contains the VC number of the packet, packet index and valid details of the packet. The packet scheduling process involves updating the corresponding markers of packets in the VC. Once the packet scheduling is completed, the SB scheduling segment drops a notification to transmit the head packets of VCs. If it succeeds in its request, the original VC is updated as allocated and the head packet of the corresponding

VC can be forwarded to the next stage for switch traversal.

Fig. 7 illustrates the flowchart of the steps used for the SA process in SB-Router. Before performing SA, as illustrated in Fig. 7, requests from all the VCs in an input port should be available with the switch allocator. In case of conflicts between various VCs in the same input port, requests buffered in the SB register will be passed onto the switch allocator, following which SA will be performed. For the cases where no conflicts are observed between various VCs in the same input port, SA will be performed instantly. After which, SB logic determines whether the requests in the SB register are successfully allocated. If a request in the SB register is allocated then SB logic starts packet scheduling. It moves the packet in the SB register to the head of VC and updates the SB register with this information. If requests in the SB register are not allocated then SB logic performs switch traversal directly. Packet scheduling is done based on the information stored in SB registers. The information update in SB registers, including releasing the SB register that is not used or adding the packet information to SB registers, should be done only after packet scheduling. Following this, switch traversal takes place and information update in SB registers is activated simultaneously. However, the process of information update in SB registers does not contribute to the critical pipeline stage of the router. Therefore, switch traversal needs to wait only for the completion of packet scheduling. It need not wait for the process of information update in SB registers. The chosen packets are forwarded to the corresponding output port to traverse through the link. To summarize, updating information in SB registers and switch traversal will be done only after packet scheduling. Therefore, updating information in SB registers will not take extra router latency. Instead, it will be done in parallel with the critical pipeline stages of the router.

As shown in Fig. 8, the information of the first packet in VC0 is stored into SB register R0, the second packet in VC1 and VC2 is stored into SB register R1 and R2 respectively and the first packet in VC3 is added to the SB register R3. As the priority is given to SB register, based on the information stored in it, the corresponding packet is chosen for SA. Referring to Fig. 8, the second packet in VC2 is chosen for illustration. This packet is then forwarded for SA and is allowed to transfer to the output port. Fig. 9 illustrates this complete process. The contents of SB register R2 are refreshed.

B. ONLINE SB

Swapping the contents of the entire VC is not an optimized design. The associated circuitry and its complexity increase with the increase in the length of SB. The RC and VCA units require additional support to meet the requests arriving from packets in SBs. In addition to that, as the storage capacity of SB increases, the chances of performance improvement reduce because the number of output ports is fixed. The requests arriving at the switch allocator

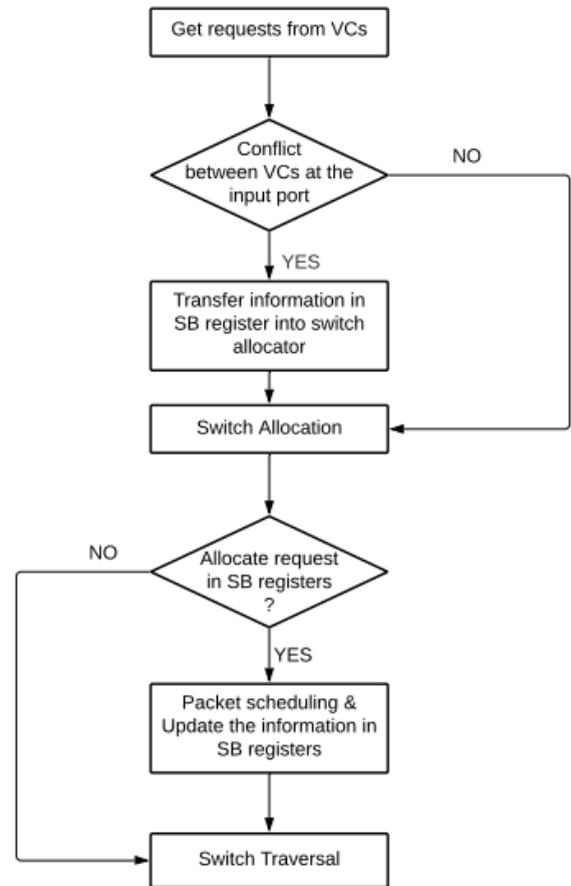


FIGURE 7. Flowchart showing steps for SA in SB-Router.

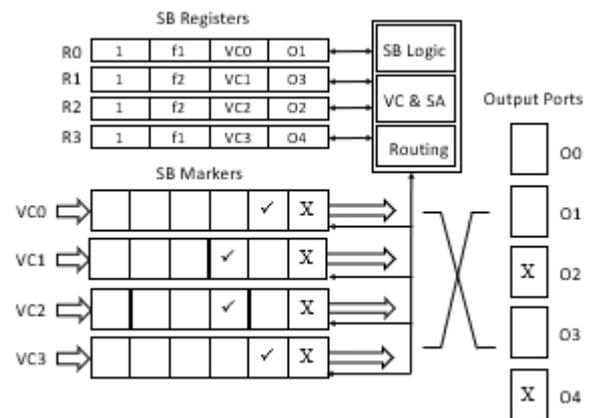


FIGURE 8. Scheduling instance. Second packet in VC2 is chosen to be transmitted to output port.

hardly target different output ports with the increase in the length of SBs.

To reduce the overhead of SBs, a part of VC is organized as an SB. For example, we can organize the first five flits of each VC as an SB. This is an efficient method to reduce the complexity of the circuit. Moreover, it needs to schedule packets in the dedicated units and take care of only these units.

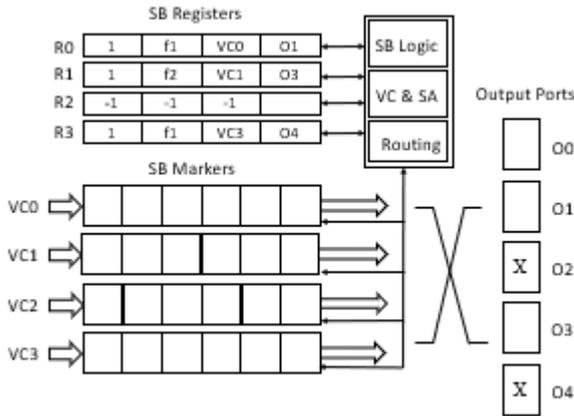


FIGURE 9. Register contents after scheduling the packets and transmitting them to the corresponding output ports.

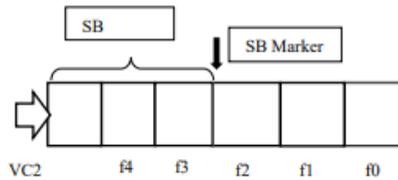


FIGURE 10. Format of online SB. The length of SB in a VC is decided as per the requirement and can be changed online based on the count of flits buffered in the VC.

However, this technique may compromise the performance improvement obtained from SBs because the number of flits in a VC is not known. There may be a few cases where the VC is empty, so SBs are of no use in such cases. On the contrary, in some other cases, VCs may have overflowing packets that cannot be accommodated only in SBs, therefore the length of SBs should be chosen carefully. The balance between performance and overhead can be considered by keeping SBs online. Fig. 10 shows online SBs where the length of SB in a VC is decided as per the requirement and can be changed online. SB marker gives the boundary between fixed VC units and units available for swapping in a VC.

C. PROPOSED PRIORITY

The preference is given to VCs at the input port based on the priorities linked with them. It is assumed that the input port priority is an integer, taken at the time of arrival of flit in the router and the input port with priority p ($p = 0, 1, 2, \dots, 5$) belongs to the p^{th} class. The p^{th} class of input port refers to the assigned priority to a particular input port. The input port with a lower index indicates higher priority. Based on the index value of an input port, priority is given to the lower index one and the output port is allocated to it. Fig. 11 shows the priority mechanism; various queues depict different priority classes. The priority is given to the SB register R0 as compared to the head packet stored in VC. If the packet stored in SB register R0 fails in the SA step, then the head packet in VC will be given a chance based on the priority associated with them.

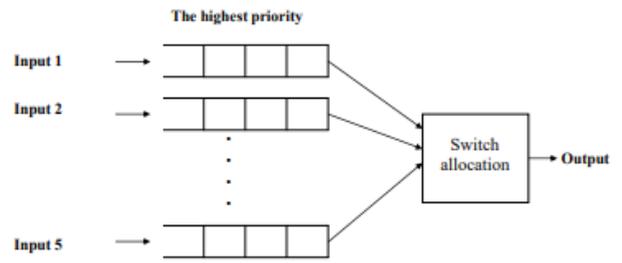


FIGURE 11. Proposed priority mechanism. Priority is given to the input port with a lower index.

D. VC BUFFER ORGANIZATION

Packet scheduling in the SB register is a crucial step in SB-Router architecture. The entire process should get over in one or two cycles. In case, it exceeds the waiting time of the head packet in VC then SB-Router is of no use. The overall circuitry and complex process will give overhead. The efficient scheduling process depends on the organization of VCs. Therefore, VC buffer organization is a critical step. To address this, it is required to maintain a data structure to keep track of packets in memory and free memory locations. Traditional VC buffer organization is done using two data structures namely linked lists and circular buffers. Two registers namely, the first marker and the last marker are used to define the buffer margin lines in the memory. An arriving flit can be included in it by indicating with the tail margin line. A flit can be removed from the buffer when it is read at the head margin line. When a marker points to the end of the buffer, it starts again from the beginning. The circular buffers are always an easy option to maintain but the requirement of a fixed division of buffer space for packet scheduling becomes a little difficult. To schedule packets efficiently, the linked list data structure is used to organize VC buffers. To schedule packets conveniently in VC, only the markers need to be changed.

E. PERFORMANCE-ENHANCING METHODS

In this subsection, the performance-enhancing methods of SB-Router are discussed.

1) VCA METHOD

SB-Router enhances the options for the packets to participate in SA. However, due to the area and power overhead, the number of SB registers is limited and also the length of SB is kept low in every VC. To further enhance the performance, it is required that the output ports of the head packet in VC and the packet buffered in SB should be different. With this method, we can increase the matching efficiency during SA. In our proposed VCA method, a more appropriate packet order in VC is presented for SB. This is achieved by swapping the non-head packets in a VC with those packets having free output ports. For each VC, the output port of the previous allocation is recorded. In the routing stage, while allocating VC for a packet, such VC is chosen whose output port is

different than the last allocated packet. If such VC is not found, a VC having the lowest buffer occupation is chosen and a packet is allocated to it. With this VCA method, it is more feasible that the output port of the non-head packet and the head packet, in the same VC, are different. This opens more options for the SA stage.

2) COMBINING WITH FILL VCA

SB-Router is combined with our previous work Fill VCA [33] to enhance its performance. The design of Fill VCA targets to mitigate HoL blocking and improve the latency of the NoC router. SB-Router opens more options for the SA and combining it with Fill VC utilizes all the requests to minimize the number of conflicts and improve the overall performance.

V. EXPERIMENTAL STUDIES

In this section, the experimental setup is discussed and our proposed design is compared to other routers. The evaluation is done by calculating packet latency under synthetic and application traffic. Area and power consumption are also studied.

A. EVALUATION METHODOLOGY

The tool used is a flit-level, cycle-accurate simulator Booksim [43] and the performance has been evaluated in synthetic traffic and application traffic. Trace-driven simulations [44] are also performed to evaluate the performance under PARSEC benchmarks. 2-dimension mesh, FBFly and torus with 64 nodes are used to measure the effectiveness in different topologies, as these topologies are generally used to test the performance of NoCs. It is assumed that all of the channels are having a delay of a single cycle. Each router is connected to four other routers in Flattened Butterfly (FBFly) & torus and one router in the mesh. In every input port, there are sixteen (16) VCs in torus and FBFly and four (4) VCs in the mesh. Each VC has 4 flits of buffer in the FBFly and torus and 8 flits in the mesh. For the mesh, deterministic (i.e. all events are determined by cause/events external to the will, ultimately) DOR (dimension- order routing) is chosen because it is one of the simple and popular choices. For the FBFly and torus UGAL routing (universal globally adaptive load-balancing) [45] is chosen. The router architecture is described in section IV and is the same for mesh, FBFly and torus. SB-Router is compared with state-of-the-art routers like Packet-Chaining, TS-Router and RoB-Router. One iteration iSLIP is used as the baseline allocator for TS-Router, Packet-Chaining, RoB-Router and SB-Router. Augmenting Path allocator [46] and 2 iteration iSLIP (iSLIP-2) are also compared with our design. Unless specified otherwise, the evaluation is done for packets having single-flit.

The traffic patterns like the uniform, bit reverse, transpose, average and tornado are used for evaluations under different topologies. These traffic patterns are common and cover various types of interconnect patterns in NoC. To optimize the buffer utilization, a VC is allocated till all its buffers

are occupied by new packets. Performance evaluation for each router is done by running simulations of 100,000 cycles with 20,000 warm-up cycles on a 64 node network. An NoC router RTL model and Cadence Encounter RTL compiler with 180 nm standard cell library are used for synthesis scaling with 1.8 V operating voltage. The frequency of operation is 1.0 GHz. The per flit number of bits is 64 same as the link width. The size of the buffer for the router is 320B. The power, area and timings analysis gave us a detailed overview of our design. For dynamic power evaluations, the detailed factors are evaluated from Booksim and are used to feed to Cadence Encounter.

In application evaluations, a trace-driven, cycle-accurate many-core simulator is used with the above-mentioned network details. The traces are collected from full-system simulations of the 64 node network.

B. NETWORK PERFORMANCE

The comparison of SB-Router with other state-of-the-art routers in mesh under different traffic patterns namely, uniform, bitrev and tornado is illustrated in Fig. 12. SB-Router is compared with one of the highest matching allocators augmenting paths allocator. Although iterations take a few cycles to execute but for the evaluations, it is assumed that augmenting path allocator takes a single cycle in SA. State-of-the-art time series-based allocators like Packet-Chaining and TS-Router are also considered for comparison. Fig. 12 (a), (b) and (c) illustrates that under all traffic patterns, SB-Router performs pretty well compared to all the other allocators. For all the traffic patterns considered here, SB-Router always shows improved performance in terms of reduced average packet latency. TS-Router shows improved performance than augmenting paths and Packet-Chaining under uniform traffic. For tornado traffic, the performance of these allocators is almost the same.

iSLIP1 and iSLIP2 doesn't show much improvement in performance as compared to other allocators under tornado and uniform traffics. But under bit rev traffic, these allocators show improved performance than the other three allocators. As compared to the baseline allocator iSLIP1, SB-Router shows much better performance improvement in terms of average packet latency than the other time series-based routers. Under uniform traffic, as shown in Fig. 12 (a), the performance improvement in terms of latency reduction shown by SB-Router is 68.75%, while that for TS router is 12.5% when the injection rate is kept at 0.42 flits/cycle. Performance improvement obtained by SB-Router under bit rev traffic is comparatively less than other traffic patterns as shown in Fig.12 (b). However, SB-Router still achieves lower packet latency at a low injection rate. Under tornado traffic, as shown in Fig. 12 (c), SB-Router shows a 63.82% reduction in the packet latency while TS-router shows about a 6% reduction in the packet latency at the injection rate of 0.27 flits/cycle.

The performance of SB-router is also evaluated in the FBFly topology under tornado and transpose traffic patterns and the latency results are illustrated in Fig. 13 (a) and (b).

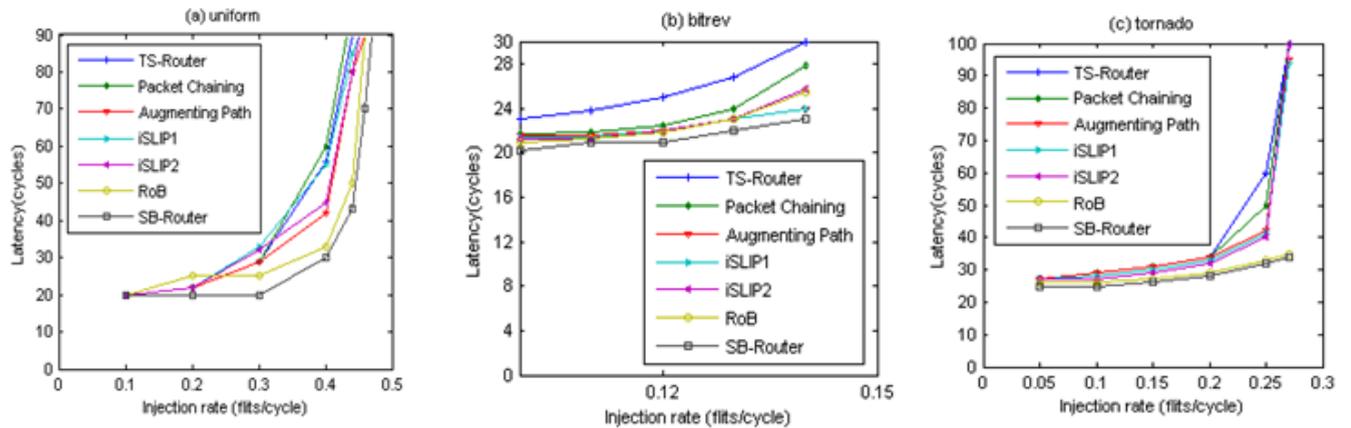


FIGURE 12. The plots show packet latency comparison of proposed router with other state-of-the-art routers for the mesh topology under uniform, bit rev and tornado traffic.

TABLE 1. Maximum injection rate measured in flits/cycle under mesh topology.

Traffic	TS Router	Packet Chaining	RoB Router	SB Router
Uniform	0.44	0.42	0.46	0.47
Bit rev	0.14	0.14	0.14	0.15
Tornado	0.25	0.26	0.28	0.29

TABLE 2. Maximum injection rate measured in flits/cycle under FBFLy topology.

Traffic	TS Router	Packet Chaining	RoB Router	SB Router
Tornado	0.44	0.45	0.46	0.48
Transpose	0.37	0.4	0.45	0.46

In FBFLy, there is more number channels between routers in the same dimension. Hence, the length of waiting queues in the input port is less, due to which lower latency improvement is achieved. However, our proposed router further reduce latency in comparison to other routers as shown in Fig. 13 (a) and (b).

The performance analysis of SB-Router is done in the torus topology under average and transpose traffic patterns and the results are illustrated in Fig. 14 (a) and (b) respectively. The performance of SB-Router strictly depends on the scheduling of packets in VCs. With the increase in the number of packets in a VC, performance improvement may be achieved. Therefore, higher the injection rate, better is the performance of SB-Router.

The improvement in matching number is obtained for SB-Router and TS -Router as compared to iSLIP1 under uniform traffic in a mesh topology. The matching number of SB-Router and TS-Router are almost the same for the injection rate less than 0.45 flits/cycle but improves drastically for the SB-Router when the injection rate is above 0.45 flits/cycle. SB-Router outperforms TS-Router.

When the packets are injected at a rate above 0.45 flits/cycle, the average rise in matching number for SB-Router is 7% higher as compared to that of TS-Router. This is because the choices available with SB-Router increase as it has more options to select packets for SA to maximize the matching efficiency as compared to TS- Router which only relies on the next allocation information.

Table 1 lists the maximum values of injection rate in flits/cycle for various traffic patterns under mesh topology.

TABLE 3. Maximum injection rate measured in flits/cycle under torus topology.

Traffic	TS Router	Packet Chaining	RoB Router	SB Router
Average	0.7	0.71	0.73	0.8
Transpose	0.35	0.36	0.4	0.5

Table 2 lists the injection rate maximum values under FBFLy topology for tornado and transpose traffic patterns when the latency threshold of 1500 cycles is considered. Table 3 lists the injection rate maximum values under torus topology for average and transpose traffic.

To further enhance the matching of SA, a VCA technique is used for SB-Router. As showcased in Fig. 15 (a), the average packet latency of SB-Router further decreases with this VCA strategy for mesh topology under uniform traffic. With the increased options available for SA, there is an improvement in the matching efficiency during SA. The average packet latency improves further for the mesh topology under bit rev and tornado traffic patterns as illustrated in Fig. 15 (b) and (c).

Table 4 presents the performance comparison of our proposed work with other state-of-the-art routers. The performance has been evaluated under three traffics namely uniform, bit rev and tornado under mesh topology. Table 4 indicates that for uniform traffic, the latency improvement obtained from our proposed work is 16.66% at the injection rate of 0.42 flits/cycle. Referring to Table 1, it can be inferred that for measuring the latency improvement over other routers, the injection rate should be at least 0.42 flits/cycle. The improvement in latency for bit rev traffic

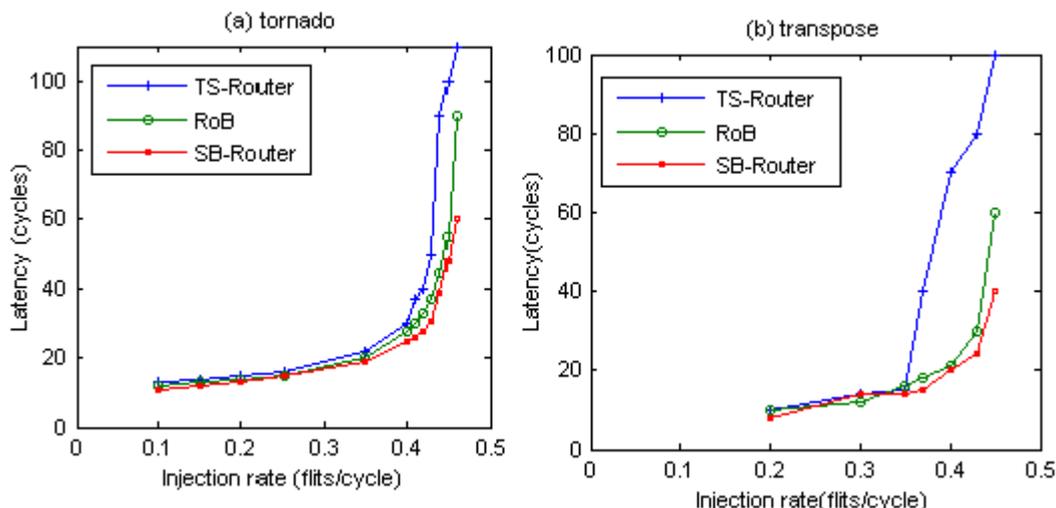


FIGURE 13. The plots show packet latency comparison of proposed router with other state-of-the-art routers for the FBFly topology under tornado and transpose traffic patterns.

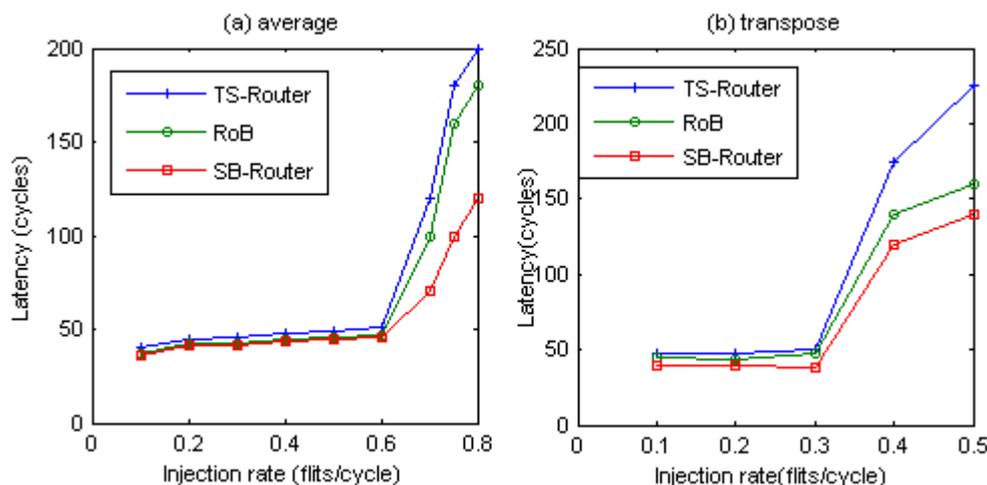


FIGURE 14. The plots show packet latency comparison of proposed router with other state-of-the-art routers for the torus topology under average and transpose traffic patterns.

is 1.57% at the injection rate of 0.14 flits/cycle. The latency improvement achieved under bit rev traffic is relatively lower than the other traffic patterns. However, SB-Router still shows lower packet latency at a low injection rate. For tornado traffic, the improvement in latency achieved by SB-Router is 3.03% at 0.25 flits/cycle.

Table 4 also presents latency improvement under FBFly for two traffic patterns namely, tornado and transpose. The latency improvement achieved for a tornado is 13.33% at the injection rate of 0.44 flits/cycle. Table 2 can be referred to for the selection of injection rate. The latency improvement for transpose traffic is 6.25 % at the injection rate of 0.37 flits/cycle. The latency improvement obtained from our proposed work is pretty good for FBFly topology.

The performance improvement in terms of latency for two traffic patterns namely, average and transpose, under torus topology is also listed in Table 4. The latency improvement for average traffic is 41.66% at 0.7 flits/cycle. For transpose traffic, the latency improvement is 24% at 0.3 flits/cycle. The improvement in latency achieved from our proposed work is relatively higher for torus topology.

The SB-Router is combined with Fill VCA and the combination (SBF) is considered for evaluation. As shown in Fig. 16 (a), (b) and (c) the average packet latency in the mesh under three traffic patterns clearly shows that SBF outperforms SB-Router. SBF can achieve a 17% reduction in average packet latency when the packets are injected at a rate close to saturation.

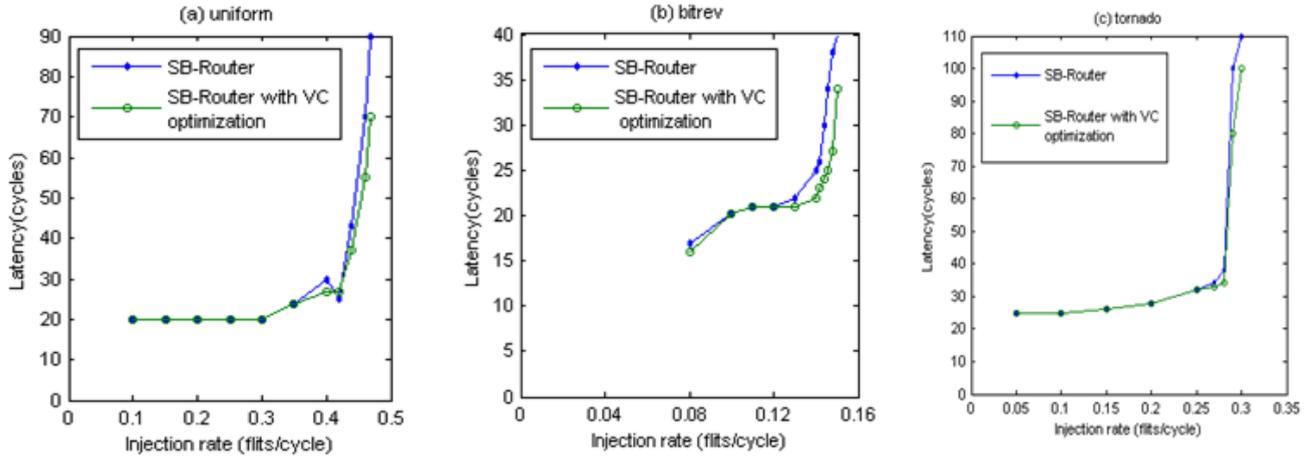


FIGURE 15. The plots show reduction in latency for SB-Router after adopting the VC allocation strategy under uniform, bit rev and tornado traffic for the mesh topology.

TABLE 4. Packet latency comparison (measured in cycles) of our proposed work (SB-Router) with other state-of-the-art routers.

Topology	Traffic	Injection Rate (flits/cycle)	TS-Router [36]	Packet-Chaining [8]	Augmenting Paths [46]	iSLIP1 [29]	iSLIP2 [29]	RoB-Router [32]	Proposed Work	Latency improvement
Mesh	Uniform	0.42	70	90	-	80	65	30	25	16.66%
Mesh	Bit Rev	0.14	30	27.8	24	24	25.7	25.4	25	1.57%
Mesh	Tornado	0.25	60	42	42	41	40	33	32	3.03%
FBFly	Tornado	0.44	90	-	-	-	-	45	39	13.33%
FBFly	Transpose	0.37	40	-	-	-	-	16	15	6.25%
Torus	Average	0.7	120	-	-	-	-	-	70	41.66%
Torus	Transpose	0.3	50	-	-	-	-	-	38	24%

C. SB-ROUTER WITH VARIOUS CONFIGURATIONS

The performance bounds of SB-Router are explored by considering SB-Router-1 to SB-Router-4. SB-Router-1 uses one SB register in each input port. SB-Router-2, SB-Router-3 and SB-Router-4 make use of two, three and four SB registers in each input port respectively. The performance of SB-Router is evaluated by varying the number of SB registers. It is observed that with the increase in the number of SB registers, the area and power increases.

To get the optimized design, we have done a comparative study. As shown in Fig. 17 (a), (b) and (c), the performance of SB-Router-1 is much higher than the baseline router. SB-Router-4 does not show much improvement in performance compared to SB-Router-1. The novelty of our proposed work is that with the help of only a few SB registers, we are able to achieve significant performance improvement in terms of latency, the results have been shown in Table 4 and 5. As compared to baseline router, SB-Router-1 shows 68.75% improvement in latency for uniform traffic under mesh topology, at the injection rate of 0.42 flits/cycle and the same is shown in Fig. 17 (a). The performance

improvement achieved by increasing the number of SB registers is comparatively less as shown in Fig. 17. The baseline router is described in section 3 of this paper. The comparison of various configurations of SB-Router from SB-Router-1 to SB-Router-4 with the baseline router is listed in Table 5.

D. APPLICATION-LEVEL NETWORK PERFORMANCE

The performance of SB-Router is further evaluated for the traces of the realistic traffic of PARSEC benchmarks. The average network latency for each application is evaluated by choosing nine typical PARSEC benchmarks and the average for all applications is also plotted as shown in Fig. 18. The realistic traffic contains a mix of both single-flit packets and multi-flit packets so that the efficiency of SB-Router is evaluated close to performance bounds. For realistic traffic, the injection rate is comparatively low (less than 0.01 flits/cycle on an average) than synthetic traffic, restricting the push in the further improvement in the performance of SB-Router. However, SB-Router still shows improvement in performance in realistic traffic as illustrated in Fig. 18. With the PARSEC benchmark, the average network latency of

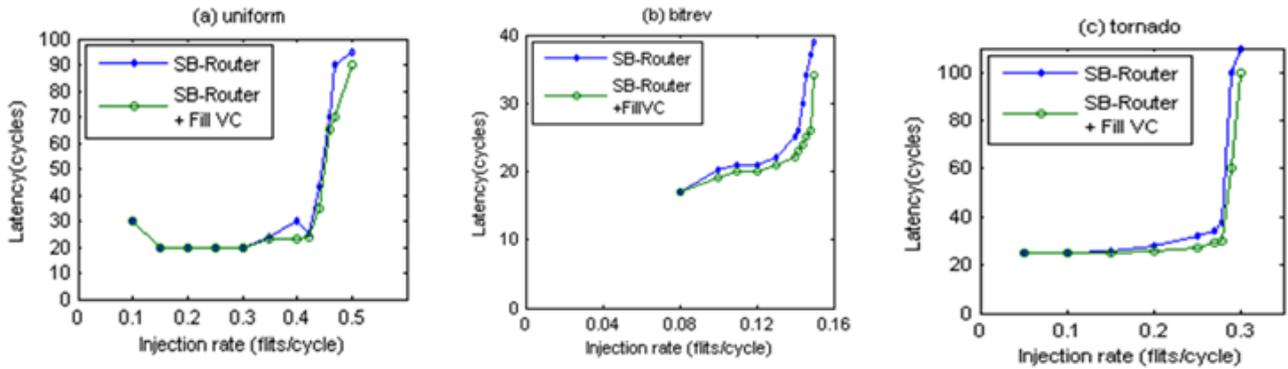


FIGURE 16. The plots show the comparison of packet latency between SB-Router and the combination of SB-Router and Fill VC under uniform, bit-rev and tornado traffic in the mesh.

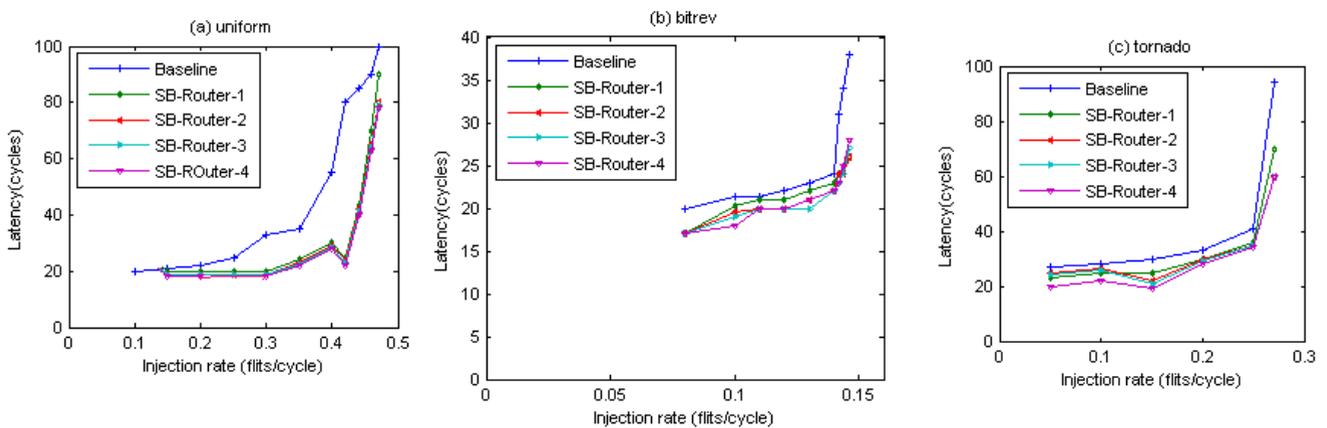


FIGURE 17. The plots show the comparison of various configurations of SB-Routers and the baseline router in terms of latency and injection rate for the mesh topology under uniform, bit rev and tornado traffic.

TABLE 5. Comparison of various configurations of SB-Router with the baseline router in terms of latency (cycles) under mesh topology.

Traffic	Injection rate (flits/cycle)	Baseline Router	SB-Router-1	SB-Router-2	SB-Router-3	SB-Router-4
Uniform	0.42	80	25	24	23	22
Bit rev	0.14	24	22	21	21	21
Tornado	0.25	41	36	35	35	34

the baseline router is compared with SB-Router for the mesh network and the results are shown in Fig. 18.

E. AREA AND POWER

For the evaluations of area usage and power consumption, various components of a router namely switch, buffers, clock and allocators are considered. The power considered here is the summation of dynamic power and static power. For our evaluations, we consider a mesh network of 64 nodes with every single router having 5 output and input ports. Every flit contains 64 bits. Every input port has a count of 4 VCs. Each one of them can store 8 flits. As a result of which, the total required memory in the baseline router is 10,240 bits. Here, the consumption of power and area of buffers include both SB

registers and VC buffers. The area usage of baseline router, SB-Router-1 and SB-Router-4 is illustrated in Fig. 19. For our evaluations, the injection rate is kept the same as that of the average injection rate of PARSEC benchmarks.

To optimize the usage of resources, SB-Router adds only a few registers to the input port and the related logic circuitry so that the additional area is marginal. Since in SB-Router, the changes are being made only in the input port, the area of crossbar and clock remains constant, as compared to the baseline router architecture. Fig. 20 illustrates the power consumption by SB-Router-1, SB-Router-4 and baseline router. It is observed that power consumption by SB-Router-1 is increased only by 8.69% and the area usage shows an increase by 6.66%. The main reason for the marginal increase in

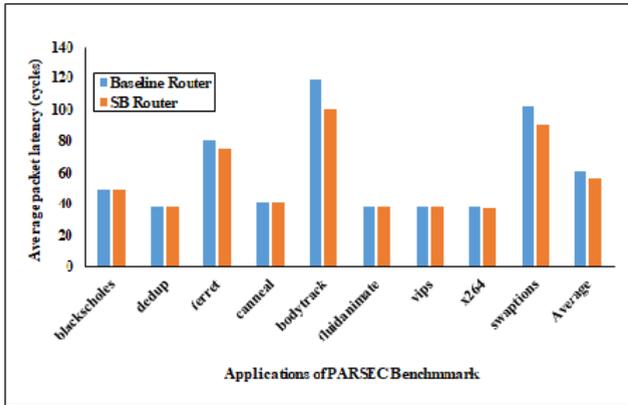


FIGURE 18. Comparison of SB-Router and Baseline router in terms of network latency using PARSEC benchmarks in the mesh topology.

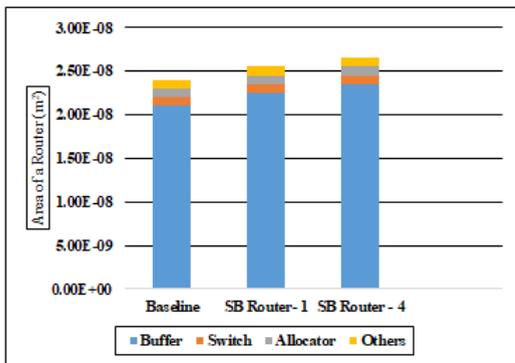


FIGURE 19. Area of SB-Routers and the baseline router.

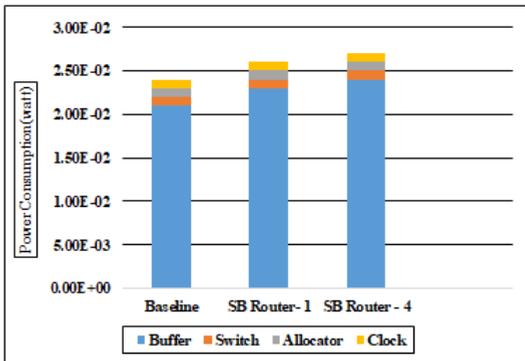


FIGURE 20. Power consumption of SB-Routers and the baseline router.

area and power consumption is the buffer. SB registers are responsible for this marginal increase in area, as the size of the VC remains unchanged. For evaluation, the power consumption and area of SB registers are added into buffers. SB registers consume a significant area to provide room for the head packets. SB register refresh segment and SB scheduling segment are responsible for a small increment in the area but moderate circuit complexity. The power consumption and area of the allocator increases, because the switch allocator and VC allocator deals with SB registers.

VI. CONCLUSION AND FUTURE WORK

SA, the most important stage for NoC routers, adversely affects the communication latency. The low latency SB-Router architecture presented in this paper helps to avoid HoL blocking by implementing efficient SA in NoC. The SB-Router architecture uses SB registers to hold the status of the non-head packets in VC. The length of SB is decided online depending upon the flits buffered in a VC. This opens more options for the packets to participate in SA. The average packet latency under different traffic patterns in mesh, FBFly and torus topology is evaluated using booksim. The proposed design is compared with other state-of-the-art routers namely RoB router, Packet Chaining and TS-Router. Latency improvement is observed in 70% of injection rates used with a moderate rise in power consumption by 8.69% and the area usage by 6.66%. The improvement in latency is effective with an increase in injection rate above 0.45 flits/cycle. With this achieved reduction in latency, several high-speed operations can be served effectively. The performance can be optimized while mapping various applications on multi-core architectures.

The further improvement in latency with a reduction in power consumption and area usage can be focused on in future work. SB-Router is designed for on-chip networks, as most packets in on-chip networks are single-flit. SB-Router is particularly suited for short packets and can be integrated into most IQ routers. In the future, the architecture can be modified to target long packets. In such cases, the mixing of flits at the receiving router can also be addressed.

REFERENCES

- [1] H. Bokhari and S. Parameswaran, "Network-on-chip design," in *Handbook of Hardware/Software Codesign*, S. Ha and J. Teich, Eds. Dordrecht, The Netherlands: Springer, 2017, pp. 461–489.
- [2] B. Aghaei, A. Khademzadeh, M. Reshadi, and K. Badie, "Link testing: A survey of current trends in network on chip," *J. Electron. Test.*, vol. 33, no. 2, pp. 209–225, Apr. 2017, doi: 10.1007/s10836-017-5646-0.
- [3] M. Rashid, N. K. Baloch, M. A. Shafique, F. Hussain, S. Saleem, Y. B. Zikria, and H. Yu, "Fault-tolerant network-on-chip router architecture design for heterogeneous computing systems in the context of Internet of Things," *Sensors*, vol. 20, no. 18, p. 5355, Sep. 2020, doi: 10.3390/s20185355.
- [4] I. Pérez, E. Vallejo, and R. Bevide, "Efficient bypass in mesh and torus NoCs," *J. Syst. Archit.*, vol. 108, Sep. 2020, Art. no. 101832, doi: 10.1016/j.sysarc.2020.101832.
- [5] H. Krutthika and K. Rajashekara, "Network on chip: A survey on router design and algorithms," *Int. J. Recent Technol. Eng.*, vol. 7, no. 6, pp. 1687–1691, Mar. 2019.
- [6] D. R. Melo, C. A. Zeferino, L. Dilillo, and E. A. Bezerra, "Maximizing the inner resilience of a network-on-chip through router controllers design," *Sensors*, vol. 19, no. 24, p. 5416, Dec. 2019, doi: 10.3390/s19245416.
- [7] C. Li, D. Dong, and X. Liao, "Exploiting contention and congestion aware switch allocation in network-on-chips," in *Proc. ACM Turing 50th Celebration Conf. (ACM TUR-C)*, 2017, pp. 1–10, doi: 10.1145/3063955.3063997.
- [8] G. Michelogiannakis, N. Jiang, D. Becker, and W. J. Dally, "Packet chaining: Efficient single-cycle allocation for on-chip networks," in *Proc. 44th Annu. IEEE/ACM Int. Symp. Microarchitecture*, Dec. 2011, pp. 83–94.
- [9] F. Khodaparast, M. Reshadi, and N. Bagherzadeh, "Application partitioning and mapping for bypass channel based NoC," *Comput. Electr. Eng.*, vol. 71, pp. 676–691, Oct. 2018, doi: 10.1016/j.compeleceng.2018.08.016.

- [10] S. M. Mohtavipour, M. Mollajafari, and A. Naseri, "A novel packet exchanging strategy for preventing HoL-blocking in fat-trees," *Cluster Comput.*, vol. 23, no. 2, pp. 461–482, 2020, doi: [10.1007/s10586-019-02940-2](https://doi.org/10.1007/s10586-019-02940-2).
- [11] C. Bienia, S. Kumar, J. P. Singh, and K. Li, "The PARSEC benchmark suite: Characterization and architectural implications," in *Proc. 17th Int. Conf. Parallel Architectures Compilation Techn. (PACT)*, 2008, pp. 72–81.
- [12] K. Jin, C. Li, D. Dong, and B. Fu, "HARE: history-aware adaptive routing algorithm for endpoint congestion in networks-on-chip," *Int. J. Parallel Program.*, vol. 47, no. 3, pp. 433–450, Jun. 2019, doi: [10.1007/s10766-018-0614-6](https://doi.org/10.1007/s10766-018-0614-6).
- [13] F. Rad, M. Reshadi, and A. Khademzadeh, "Flow control and scheduling mechanism to improve network performance in wireless NoC," *IET Commun.*, vol. 14, no. 14, pp. 2231–2239, Aug. 2020, doi: [10.1049/iet-com.2019.1033](https://doi.org/10.1049/iet-com.2019.1033).
- [14] M. Katta and T. K. Ramesh, "Virtual channel and switch traversal in parallel to improve the latency in network on chip," in *Proc. 2nd PhD Colloq. Ethically Driven Innov. Technol. Soc. (PhD EDITS)*, Nov. 2020, pp. 1–2, doi: [10.1109/PHDEEDITS51180.2020.9315315](https://doi.org/10.1109/PHDEEDITS51180.2020.9315315).
- [15] A. Das, A. Kumar, and J. Jose, "Reducing off-chip miss penalty by exploiting underutilised on-chip router buffers," in *Proc. IEEE 38th Int. Conf. Comput. Design (ICCD)*, Oct. 2020, pp. 230–238.
- [16] R. S. Ramanujam, V. Soteriou, B. Lin, and L.-S. Peh, "Extending the effective throughput of nocs with distributed shared-buffer routers," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 30, no. 4, pp. 548–561, Apr. 2011.
- [17] A. T. Tran and B. M. Baas, "Achieving high-performance on-chip networks with shared-buffer routers," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 22, no. 6, pp. 1391–1403, Jun. 2014.
- [18] I. Seitanidis, A. Psarras, K. Chrysanthou, C. Nicopoulos, and G. Dimitrakopoulos, "ElastiStore: Flexible elastic buffering for virtual-channel-based networks on chip," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 12, pp. 3015–3028, Dec. 2015.
- [19] X. Xiang, W. Shi, S. Ghose, L. Peng, O. Mutlu, and N.-F. Tzeng, "Carpool: A bufferless on-chip network supporting adaptive multicast and hotspot alleviation," in *Proc. Int. Conf. Supercomput. (ICS)*, Jun. 2017, pp. 1–11.
- [20] X.-Y. Xiang and N.-F. Tzeng, "Deflection containment for bufferless network-on-chips," in *Proc. IEEE Int. Parallel Distrib. Process. Symp. (IPDPS)*, May 2016, pp. 113–122.
- [21] A. Psarras, I. Seitanidis, C. Nicopoulos, and G. Dimitrakopoulos, "Short-Path: A network-on-chip router with fine-grained pipeline bypassing," *IEEE Trans. Comput.*, vol. 65, no. 10, pp. 3136–3147, Oct. 2016.
- [22] P. Mukherjee and S. Chattopadhyay, "Low power low latency floorplan-aware path synthesis in application-specific network-on-chip design," *Integration*, vol. 58, pp. 167–188, Jun. 2017, doi: [10.1016/j.vlsi.2017.02.010](https://doi.org/10.1016/j.vlsi.2017.02.010).
- [23] P. Guo, Q. Liu, R. Chen, L. Yang, and D. Wang, "A bypass-based low latency network-on-chip router," *IEICE Electron. Exp.*, vol. 16, no. 4, pp. 1–12, 2019.
- [24] K. Parane, P. P. B. M., and B. Talawar, "LBNoC: Design of low-latency router architecture with lookahead bypass for network-on-chip using FPGA," *ACM Trans. Design Autom. Electron. Syst.*, vol. 25, no. 1, pp. 1–26, Jan. 2020.
- [25] C. Effiong, G. Sassatelli, and A. Gamatie, "Distributed and dynamic shared-buffer router for high-performance interconnect," in *Proc. 11th IEEE/ACM Int. Symp. Netw.-on-Chip*, Seoul, South Korea, Oct. 2017, pp. 1–8.
- [26] N. Jindal, S. Gupta, D. P. Ravipati, P. R. Panda, and S. R. Sarangi, "Enhancing network-on-chip performance by reusing trace buffers," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 4, pp. 922–935, Apr. 2020, doi: [10.1109/TCAD.2019.2907909](https://doi.org/10.1109/TCAD.2019.2907909).
- [27] K. Wu, D. Dong, C. Li, S. Huang, and Y. Dai, "Network congestion avoidance through packet-chaining reservation," in *Proc. 48th Int. Conf. Parallel Process.*, Aug. 2019, pp. 1–10, doi: [10.1145/3337821.3337874](https://doi.org/10.1145/3337821.3337874).
- [28] S. Ma, N. E. Jerger, and Z. Wang, "Whole packet forwarding: Efficient design of fully adaptive routing algorithms for networks-on-chip," in *Proc. IEEE Int. Symp. High-Performance Comp Architecture*, Feb. 2012, pp. 1–12.
- [29] N. McKeown, "The iSLIP scheduling algorithm for input-queued switches," *IEEE/ACM Trans. Netw.*, vol. 7, no. 2, pp. 188–201, Apr. 1999.
- [30] S. Rao, S. Jeloka, R. Das, D. Blaauw, R. Dreslinski, and T. Mudge, "VIX: Virtual input crossbar for efficient switch allocation," in *Proc. 51st ACM/EDAC/IEEE Design Automat. Conf. (DAC)*, Jun. 2014, pp. 1–6.
- [31] S. S. Mukherjee, F. Silla, P. Bannon, J. Emer, S. Lang, and D. Webb, "A comparative study of arbitration algorithms for the alpha 21364 pipelined router," in *Proc. 10th Int. Conf. Architectural Support Program. Lang. Operating Syst. (ASPLOS)*, 2002, pp. 223–234.
- [32] C. Li, D. Dong, Z. Lu, and X. Liao, "RoB-router: A reorder buffer enabled low latency network-on-chip router," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 9, pp. 2090–2104, Sep. 2018.
- [33] M. Katta and T. K. Ramesh, "Latency improvement by using fill VC allocation for network on chip," in *Data Engineering and Communication Technology*. Singapore: Springer, May 2021, pp. 561–569.
- [34] M. Ahn and E. J. Kim, "Pseudo-circuit: Accelerating communication for on-chip interconnection networks," in *Proc. 43rd Annu. IEEE/ACM Int. Symp. Microarchitecture*, Dec. 2010, pp. 399–408.
- [35] S. A. R. Jafri, H. B. Sohail, M. Thottethodi, and T. Vijaykumar, "ApSLIP: A high-performance adaptive-effort pipelined switch allocator," School Elect. Eng., Purdue Univ., West Lafayette, IN, USA, Tech. Rep. 451, Oct. 2013.
- [36] Y.-Y. Chang, Y. S.-C. Huang, M. Poremba, V. Narayanan, Y. Xie, and C. King, "TS-router: On maximizing the quality-of-allocation in the on-chip network," in *Proc. IEEE 19th Int. Symp. High Perform. Comput. Archit. (HPCA)*, Feb. 2013, pp. 390–399.
- [37] M. Vinodhini, N. S. Murty, and T. K. Ramesh, "Transient error correction coding scheme for reliable low power data link layer in NoC," *IEEE Access*, vol. 8, pp. 174614–174628, Sep. 2020.
- [38] O. L. M. Srrayvinya, M. Vinodhini, and N. S. Murty, "A unique low power network-on-chip virtual channel router," in *Proc. IEEE Int. Conf. Comput. Intell. Comput. Res. (ICCIC)*, Dec. 2017, pp. 1–5.
- [39] M. Vinodhini and N. S. Murty, "Reliable low power NoC interconnect," *Microprocessors Microsyst.*, vol. 57, pp. 15–22, Mar. 2018.
- [40] I. A. Alimi, R. K. Patel, O. Aboderin, A. M. Abdalla, R. A. Gbadamosi, N. J. Muga, A. N. Pinto, and A. L. Teixeira, "Network-on-chip topologies: Potentials, technical challenges, recent advances and research direction," in *Network-on-Chip*. London, U.K.: InTech, Apr. 2021, doi: [10.5772/intechopen.97262](https://doi.org/10.5772/intechopen.97262).
- [41] L.-S. Peh and W. J. Dally, "A delay model and speculative architecture for pipelined routers," in *Proc. HPCA 7th Int. Symp. High-Performance Comput. Archit.*, 2001, pp. 255–266.
- [42] X. Chen, F. Shi, F. Yin, and X. Wang, "A novel look-ahead routing algorithm based on graph theory for triplet-based network-on-chip router," *J-Stage IEICE Electron. Exp.*, vol. 15, no. 8, p. 15, 2018.
- [43] W. Myung, Z. Qi, and M. Cheng, "Performance analysis of routing algorithms in mesh based network on chip using booksim simulator," in *Proc. IEEE Int. Conf. Intell. Appl. Syst. Eng. (ICIASE)*, Apr. 2019, pp. 297–300, doi: [10.1109/ICIASE45644.2019.9074082](https://doi.org/10.1109/ICIASE45644.2019.9074082).
- [44] G. S. Sangeetha, V. Radhakrishnan, P. Prasad, K. Parane, and B. Talawar, "Trace-driven simulation and design space exploration of network-on-chip topologies on FPGA," in *Proc. 8th Int. Symp. Embedded Comput. Syst. Design (ISED)*, Dec. 2018, pp. 129–134, doi: [10.1109/ISED.2018.8703884](https://doi.org/10.1109/ISED.2018.8703884).
- [45] M. S. Rahman, S. Bhowmik, Y. Ryasnianskiy, X. Yuan, and M. Lang, "Topology-custom UGAL routing on dragonfly," in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal.*, Nov. 2019, pp. 1–15.
- [46] R. Mullins, A. West, and S. Moore, "Low-latency virtual-channel routers for on-chip networks," in *Proc. 31st Annu. Int. Symp. Comput. Archit.*, Jun. 2004, pp. 188–197.



MONIKA KATTA received the B.E. degree (Hons.) in electronics and communication engineering from the University of Rajasthan and the M.Tech. degree (Hons.) in VLSI design from MNIT Jaipur, PGDBM in Finance, Symbiosis International University, Pune. She is currently pursuing the Ph.D. degree with Amrita Vishwa Vidyapeetham, Bengaluru, India. She is currently working as a Senior Assistant Professor with the Department of Electronics and Communication

Engineering, New Horizon College of Engineering, Bengaluru. Her research interest includes on chip networking in multi-core processors.



T. K. RAMESH (Member, IEEE) received the Ph.D. degree in optical networks from Amrita Vishwa Vidyapeetham. He is currently an Associate Professor with the Department of Electronics and Communication Engineering, Amrita School of Engineering, Bengaluru. He has 28 years of teaching and research experience. He has successfully guided four Ph.D. students. He is also guiding eight Ph.D. scholars. He has published more than 80 research publications in peer-reviewed international journals and conferences. His areas of research interests include communication networks and applications, analog and digital devices and circuits, functional safety, artificial intelligence, and network-on-chip. He is a Lifetime Member of Indian Society for Technical Education and a member of The Institution of Electronics and Telecommunication Engineers.



JUHA PLOSILA (Member, IEEE) received the Ph.D. degree in electronics and communication technology from University of Turku (UTU), Finland, in 1999. He is currently a Full Professor in autonomous systems and robotics with the Department of Future Technologies, Faculty of Science and Engineering, UTU. He is also the Head of the EIT Digital Master Program in embedded systems with the EIT Digital Master School, European Institute of Innovation and Technology, and represents UTU in the Node Strategy Committee of the EIT Digital Helsinki/Finland node. His research interests include adaptive multi-processing systems and platforms, and their design, including, specification, development, and verification of self-aware multi-agent monitoring and control architectures for massively parallel systems, machine learning, and evolutionary computing-based approaches, as well as application of heterogeneous energy efficient architectures to new computational challenges in the cyber-physical systems and the Internet-of-Things domains, with a recent focus on fog/edge computing (edge intelligence), and autonomous multi-drone systems.

...