Journal of Electrical and Computer Engineering Innovations
(JECEI)
Journal homepage: http://www.jecei.sru.ac.ir

Research paper

RR-SFVP: A Novel Arbitration Unit Design for NoC Router, Ingeniously Fusing the Round Robin Method with Strong Fairness and Variable Priority

E. Shafigh Fard [1], M. A. Jabraeil Jamali [2,], M. Masdari [1], K. Majidzadeh [1]*

[1]*Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, Iran.*

[2]*Department of Computer Engineering, Shabestar Branch, Islamic Azad University Shabestar, Iran.*

Article Info

Article History:
Received 31 August 2024
Reviewed 12 October 2024
Revised 10 December 2024
Accepted 15 December 2024

Keywords:
Network on Chip
Router
Round robin
Arbiter
virtual channel

*Corresponding Author's Email Address: *m_jamali@itrc.ac.ir*

Abstract

Background and Objectives: A network on Chip (NoC) is a scalable communication framework that supports several cores. In some cases, while designing a customized Network-on-Chip, the communication needs across IP cores are often uneven, resulting in imbalanced loads on the input ports of a router. The arbitration unit plays a crucial role in the design of the NoC micro-router architecture as it substantially influences the performance, chip occupancy, and power consumption of the NoC.

Methods: This article presents a router arbitration architecture that utilizes a mix of variable priority arbitration and round-robin methods. The arbitration process evaluates other channels' requests using the Round Robin index within this architectural framework. A novel approach was suggested to integrate a network router unit onto a single chip, offering several benefits compared to earlier methods. The most significant advantage is its variable priority feature, which allows inputs to be assigned different priority levels regardless of the design circuit. The system is meant to prioritize fairness across all requests by sequentially executing them. The second and primary benefit of the developed circuit is its ability to retain the previously assigned virtual channel ID. This feature preserves the provided virtual channel ID and reduces the time required to verify the requested virtual channels in the subsequent cycle.

Results: The evaluation process occurs after the flit has been requested to quit the virtual channel and the availability of the corresponding virtual channel has been verified. The simulation findings demonstrate that the RR-SFVP arbitration unit's design is 12.1% more compact in space than the standard RR approach, offering a promising solution for space-constrained designs. It exhibits 4.3% lower power consumption, a significant improvement in energy efficiency, and 55.1% reduced critical path time, enhancing the system's overall performance.

Conclusion: The RR-SFVP technique incorporates all favorable elements in the design of the arbitration unit circuit, such as variable priority and equitable arbitration. Its clear benefits make a strong case for its superiority in the field.

Introduction

Network on Chip (NoC) is a suggested architecture designed to address the issues of using a shared bus. This network employs a modular and scalable architecture rather than a conventional bus, mapping IP blocks onto the network as tiles [1]-[4]. Data is

transported across connections using a built-in router in a packed format. Unlike bass-based systems, NoC (Network-on-Chip) is an important innovation that enables increased bandwidth and improved scalability inside each tile [5], [6]. The importance of this technology lies in its capacity to surpass the constraints of shared bus systems. As a result, it has become a crucial focus of study and development in computer engineering. This field deserves our attention and active involvement due to its potential to revolutionize the future of computing. The NoC architecture typically comprises processing cores, routers, and connections. Each router is composed of a switch and many buffers [7]. NoC is a state-of-the-art on-chip interconnect network designed for packet-based communications. Network-on-Chips (NoCs) provide the advantages of reduced packet latency, increased bandwidth, improved throughput, fewer space requirements, enhanced energy efficiency, and increased fault tolerance. Network-on-chip (NoC) systems rely heavily on routers, the primary building pieces [8]. Typically, data in the network system on the NoC chip is transferred in packets, which are then subdivided into flits. The whole packet is stored in the input buffer before being sent to the wormhole switching in the store and forward architecture. The flits are then exchanged between each router. The message comprises an initial head flit, one or more data chunks, and a concluding tail flit. According to Fig. 1, the flits are 16 bits in this article. Network on Chip (NoC) is a suggested architectural solution to address the issues of using a shared bus. This network employs a modular and scalable architecture instead of a conventional bus, where IP blocks are assigned to the network as tiles [1]-[4]. Data is transported across connections using a built-in router in a packed format. Unlike bass-based systems, NoC (Network-on-Chip) is an important innovation that enables increased bandwidth and improved scalability inside each tile [5], [6]. The importance of this technology resides in its capacity to surpass the constraints of shared bus systems. As a result, it has become a crucial area of study and advancement in computer engineering. This topic requires our attention and involvement due to its potential to revolutionize the future of computing. The NoC architecture typically comprises processing cores, routers, and connections. Each router is composed of a switch and many buffers [7]. NoC is a state-of-the-art on-chip interconnect network designed for packet-based communications. Network-on-Chips (NoCs) provide the advantages of reduced packet latency, increased bandwidth, improved throughput, fewer space requirements, enhanced energy efficiency, and increased fault tolerance. Routers serve as the primary components of Network-on-Chips (NoCs) [8]. Typically, in the network system of the NoC chip, data is sent in packets, which are then fragmented into flits. In the store and forward paradigm, the whole packet is stored in the input buffer before being sent to the wormhole switching. The packet is divided into smaller units called flits and passed between each router. The message comprises an initial head flit, one or more data chunks, and a concluding tail flit. According to Fig. 1, the flits have a size of 16 bits in this article.

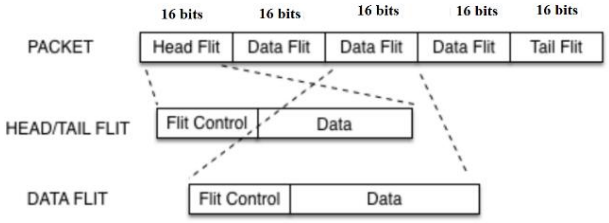
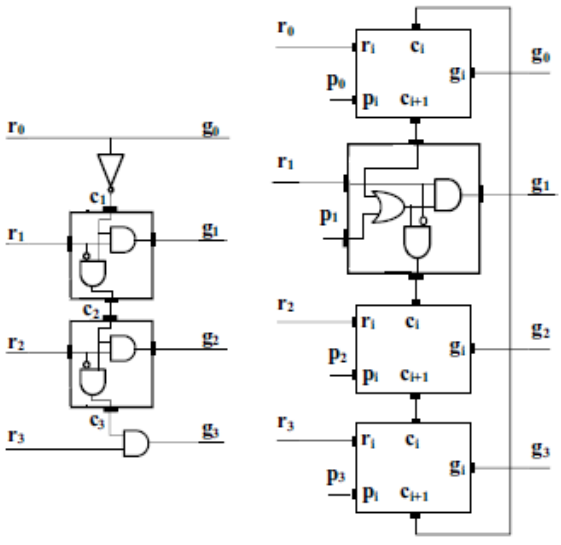


Fig. 1: Packet and flit structure [9].

A router's data pipeline comprises an input port buffer, a crossbar switch structure, and an essential component known as the arbitration unit. The arbitration unit manages traffic by identifying the virtual channel with the greatest priority for sending data under competitive settings [10]-[12]. The configuration of the arbitration unit might be intricate depending on the arbitration priority and the kind of control. The critical path delay in a Network-on-Chip (NoC) router architecture often occurs in the input ports, switches, and arbitrations. This delay is rather considerable compared to other units, primarily because of the intricate construction of the arbitration unit. Therefore, the arbitration unit circuit calculates the highest possible system speed. Hence, the choice of arbitration unit design has a crucial role in determining the performance, characteristics, speed, and power consumption of the NoC system [13]-[16]. Fig. 2 depicts two arbitration units, including four input ports.



(a) Fixed priority (b) Variable priority

Fig. 2: Arbitration architecture with four input ports [3].

These units can resolve conflicts between n requests (r0, r1... rn) using available resources criteria and indications. The winning request on each line is granted (gi). Regarding priority, the arbitration architecture may be categorized into fixed and variable architecture. In the case of an arbiter with fixed priority, the priority of requests is decided linearly. Fig. 2(b) depicts an arbiter with a predetermined priority, where r0 is assigned the most significant priority and r3 is assigned the lowest priority [17], [18]. Variable priority arbitration differs from fixed priority arbitration in that it considers additional indications while determining the allocation of resources and following sequential criteria. Arbitration may be categorized into three classes (weak, firm, or FIFO) [19] based on the fairness requirements and the form of arbitration. In weak, fair arbitration, requests are approved without considering particular criteria and priorities. In a significant and equitable arbitration process, responses to requests are determined according to priorities and criteria contingent upon the unique circumstances. Equitable demands are allocated using the initial-in-first-out (FIFO) method, which prioritizes the earliest arrival of the initial service. Furthermore, a variable priority arbitration unit may ensure equitable arbitration, as seen in Fig. 2(a). When the priority is shifted from one cell to another in a time cycle, this kind of arbitration is called round-robin arbitration, as seen in Fig. 3.

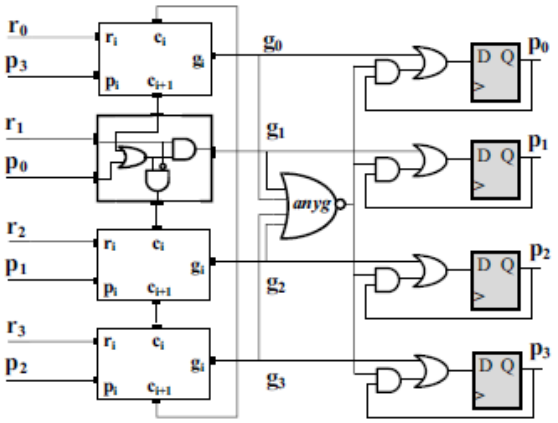


Fig. 3: Round-robin arbiter architecture with variable priority [16].

In this arbitration scenario, when the priority of g1 is high in the present cycle, P_1 is assigned a high priority in the subsequent clock cycle. Consequently, r_2 attains the most significant priority in the following clock cycle, while r_1 is assigned the lowest priority. The round-robin arbitration model [20] is a straightforward and efficient method that does not suffer from starvation. As the number of input requests increases, the round-robin arbitration structure expands, resulting in more space, greater power consumption, and longer critical route delays for big processors. This research has considered additional crucial aspects contributing to a fairer arbitration process based on the RR technique. This approach handles exit request cycles and addresses cycles when no request is generated from any channel (significant arbitration). In addition to this benefit, a circuit with a much reduced critical path latency has been created compared to earlier techniques, such as RR. The notable contributions of this study were as follows: This study effectively decreases the critical route time and power use of Network-on-Chip (NoC) by using variable and equitable priority in Round-Robin (RR) arbitration. The decrease in size allows for improved and environmentally friendly network operations, showcasing the tangible advantages of the suggested RR-SFVP arbitration architecture.

- Utilizing significant determinants to choose a robust arbitration mechanism and providing a grant to the appropriate channel in the presence of competitive situations among many virtual channels for picking an output port. The suggested design provides a cost-effective solution compared to existing arbitration systems due to its reduced hardware demand, alleviating the audience's concerns. The subsequent parts of the paper are structured in the following manner: Section 2 provides a concise overview of the relevant literature. Section 4 presents an elaborate analysis of the simulation of the RR-SFVP arbitration architecture and its corresponding outcomes. This part provides tangible proof of the architecture's efficacy, further confirming its potential. Section 5 ultimately wraps up and suggests avenues for further study, urging the audience to consider investigating the potential of this groundbreaking design.

Related Work

Dally and Towles introduced the Matrix round-robin arbitration unit design. This architecture sequentially evaluates the input requests and is considered one of the arbitration approaches that ensure essential fairness while preserving the prior award. This kind of arbitration is beneficial for a limited number of inputs. Nevertheless, the intricate hardware design and the extensive resource use have compelled researchers to choose a more efficient arbitration unit [21]. The arbitration unit is used in a router with an input port known as a link-list DAMQ (LLD), specifically called an LLD-Matrix router. This approach utilizes lists and table linkages in the input port to facilitate the reading and writing of flits in the buffer. The router equipped with the LLD input port is an option that offers a reasonably affordable hardware cost but exhibits limited performance. This method facilitates concurrent communication, which may be particularly advantageous for NoC routers—a link-list method to update five tables after each write and read operation results in significant delays. Several modifications made to the specified tables often result in substantial time delays. The latency also escalates as the input rate rises [22].

The matrix round-robin arbitration mechanism is used when a router's input port is ViChaR. In contrast to the LINK-LIST DAMQ paradigm, this technique does not use the LINK-LIST. The ViChar controller circuit is expensive in terms of hardware. This technique can accommodate virtual channels with the maximum buffer slot size, necessitating arbitration for allocating slot buffers to virtual channels and switches. Large buffer slots may create a bottleneck on critical pathways, limiting NoC. This approach has many drawbacks, including complexity, restrictions in adjustment, and lengthier pipelines in the entrance and departure of flits [23]. Fu and Ling compared two methods, RoR [19] and Matrix [21], on an FPGA platform [24], focusing on resource consumption, performance, and power consumption. They concluded that Matrix arbitration utilizes a more significant amount of resources. Both methods have equivalent power consumption; however, the Matrix method exhibits superior data processing speed compared to the RoR method. Zheng and Yang introduced a round-robin arbiter technique in which inputs are arbitrated simultaneously. The PRRA algorithm is derived from a straightforward binary search method that utilizes four inputs to enhance latency. A proposed IPRRA design aims to decrease the critical path delay in PRRA architecture [25]. The IPRRA method significantly reduces the time taken for the PRRA critical route. Lee et al. [26] suggested the Round Robin Arbiter (HDRA) method. This method employs individual filter circuits for each input, which utilize indicators within the circuit to determine the order of requests and assign grants to specific inputs. PRRA lacks fairness compared to other methods, including the discussed method. IIR arbitration is an arbitration method that outperforms similar approaches regarding power consumption and delay due to its superior architecture. This architectural design transmits requests r0, r1,… rn to the arbitration unit. This unit evaluates the requests systematically, and if they meet the criteria, a grant will be allocated to the approved request. The IRR_WF method, akin to the IRR method, does not retain the previous grant clock value. This technique is referred to as arbitration with weak fairness. The distinction between IRR and IRR_WF lies in the fact that IRR, in the absence of an exit request, retains the previous request based on the presence of REG, which is deemed equitable arbitration for the subsequent cycle. Nevertheless, in the IRR-WF approach, the absence of a register results in the loss of the previous request if all requests are zero [27]. To ensure that a high-performance network of chip switches is provided [28], an efficient arbiter is needed, especially in terms of fairness. The architecture was proposed based on a tree structure, which divided and distributed the arbitration task to separate nodes, providing high-performance arbitration with excellent scalability. The FSA (Fairness Switch Arbiter) uses a feedback-based parallel priority update mechanism to complete arbitration. The FSA method uses four inputs to achieve a critical path with only an O ($log4N$) delay. This method is similar to IPRRA in structure and performance but fairer than IPRRA.

The Weighted Round Robin Arbiter (WRRA) [29] builds on the Round Robin Arbiter's principles by focusing on fair resource allocation. Each requester receives resources proportional to their assigned weight, distinguishing it from traditional methods. For example, if two requesters have weights of 3 and 7, they would receive 30% and 70% of grants over time, respectively. The arbiter uses a credit-based mechanism to determine allocations, maintaining credit counters for each requester to indicate eligibility. A replenishment process restores credits when no requests are active. Operating on clock signals, it updates counters during grant allocation and replenishment, ensuring effective resource management. Overall, this module enhances fairness and efficiency in resource distribution.

Group decision (GD) [30] method discusses a bus arbiter designed using a group decision algorithm that integrates fixed priority, round-robin, and mixed priority systems to create a new priority sequence. This approach addresses the starvation problem in multi-master systems on a chip (SoC) by swiftly responding to masters requiring bus access, halving their waiting time. Unlike traditional methods that may prioritize specific requests, this algorithm ensures that multiple masters can receive timely responses, albeit with increased area and power requirements. The group decision algorithm offers several advantages over conventional bus arbitration methods, including fairness and improved bus usage while leveraging the benefits of various priority types. Table 1 summarizes the methods employed and their respective benefits and drawbacks, which can be found in this section.

Table 1: Advantages and disadvantages of different types of arbitration units

disadvantages	advantages	how the method works	Method name
Arbitration with fixed priority	-Delay and lower cross-section compared to other methods Simplicity - Less overhead,	All the requests enter the general multiplexer, which, by choosing a request, is registered and leads to a substantial arbitration (no request), and the counter goes to the subsequent request (if there is a request)	IRR [23]

The processing speed is lower than the other methods	The possibility of saving the current priority in the cycle without a request	Works as a variable priority, and the request chain enters the arbitration process along with the i0 and p0 priority and proceeds in order.	ROR [10]
Higher resource consumption	- Faster data processing speed -Matrix fair arbiter	Resets the bits of row i and sets the bits of column i. Matrix	Matrix [7]
Poor adjudication (failure to save current grant) in the absence of request	-faster adjudication request than the IRR -The consumption area is less compared to IRR	As the IRR method minus the presence of SF multiplexer IRR_WF	IRR_WF [23]
Poor adjudication (failure to save the current grant) in the absence of an application	Less delay in critical paths	Requests are of fixed priority type, where each request consists of a flip-flop and multiplexer, where r0 has the highest priority at the beginning, and chainwise if r0 is not requested, the priority reaches other r. HDRA	HDRA [20]
The most extended delay among all arbitration methods Poor arbitration (no current grant record) -High consumption power	Reduction of critical path delay	RR method, round trip method based on binary (parallel) search algorithm PRRA (Zheng & Yang, 2007)	PRRA [18]
High latency The largest cross-sectional area compared to other methods.	Reducing the execution time of the PRRA method	The improved model of the PRRA IPRRA method	IPRRA [18]
Combination loop	Faster than the base rra	Connecting each cell to the next (s) Cell	Timing speculative arbiter [11]
Has a long critical path due to transport propagation through fixed and variable priority cells.	However, it avoids the compound loop.	A fixed-priority cell chain replaces the connection between the last and first cells.	Acyclic arbiter [12]
Big area circuit	Shorter critical path, fairness	Tree structure search algorithm	FSA [28]
MORE POWER CONSUMED	Superior fairness in resource allocation compared to the RR.	Based on RRA with resources proportional to their assigned weight	WRRA [29]
High power consumed More area	Fairness	combines the advantages of fixed priority, round-robin,	GA [30]

The Proposed Method

There are N arbiters in an N × N fabric switch, each responsible for arbitrating requests from all input ports directed to an output port. Due to the uniform construction of an arbiter compared to others, we will investigate just one arbiter.

A. VC Arbitration

VC arbitration is a crucial organizational component of a router that significantly affects the effectiveness of a NoC system. The Arbiter conducts arbitration among the competing VCs over a singular resource, such as an output port.

This dissertation presents innovative methods for dynamic virtual channel flow control techniques and virtual channel arbitration. The first two methods rely on the adaptability of virtual channels at the router input port, enhancing the effectiveness of the network-on-chip system. In both systems, the input port consists of a centralized buffer with slots dynamically assigned to virtual channels based on real-time traffic conditions. The use of several virtual channels with low buffering resources achieves performance enhancement. The VC arbitration method relies on an efficient and rapid arbiter that operates depending on the index of its input ports (or VC requests).

In the preceding part, input-port VC arbitrations were often executed with input-arbiter modules during the switch allocator phase. Nonetheless, the NoC arbiter architecture in this approach employs VC-Selector modules at the input port for two primary reasons: a central buffer retains all VC flits from an input port, and arbitration occurs within a single clock cycle. Upon transmitting a grant signal to an input port, the read pointer is either already aligned with the victorious VC flit, or the winning flit is positioned at the output port of the buffer. The design indicates that the VC-Selector selects a VC for the arbiter while concurrently loading the flit at the buffer output. The VC_req signals in the local link data (LLD), and the request data queue (RDQ) input ports provide the read pointer and facilitate the blocking mechanism, respectively. Although one may contemplate integrating VC-Selector modules inside the switch allocator, this would establish a reliance between the input port and arbiter, sharing segments of their critical routes. To prevent this, the architecture incorporates the

input-port VC arbitration inside the input-port, obviating the need for input-arbiter modules in the switch allocator, as seen in Fig. 4.

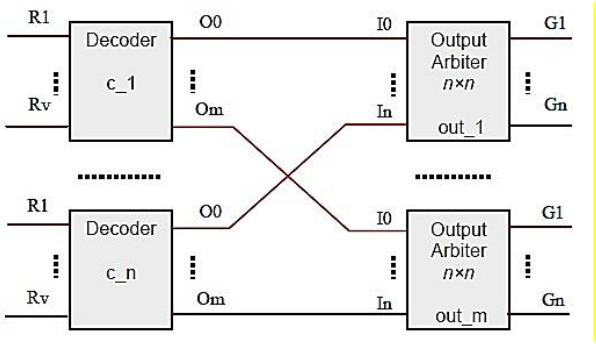


Fig. 4: *n*×*m* S.A. architectures, *n*= # of inputs, *m*= # of outputs, *v*=# of V.C.s per input-port.

The description of SA Micro-Architecture corresponds to the n input ports, as seen in Fig. 5.

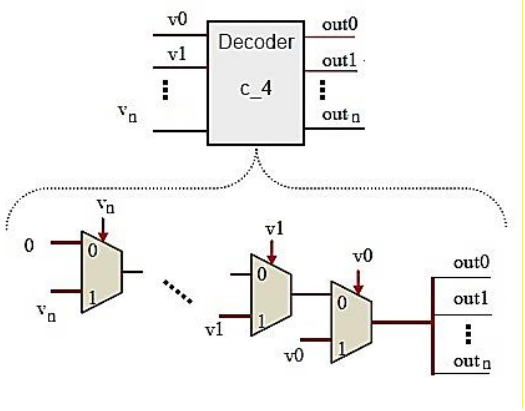


Fig. 5: Separable SA Micro-Architecture.

B. RR-SFVP Arbiter Micro-Architecture

The proposed arbitration architecture combines arbitrations with variable priority based on round robin. In the RR-SFVP method, all positive aspects have been used in the design of the arbitration unit circuit, including variable priority and fair arbitration. A particular virtual channel does not always have the highest priority, which is part of the variable priority characteristics. Referring to the previous methods of the arbitration unit, the requests are dealt with in order of fixed priority. In the RR-SFVP method, a (requested) channel does not always have the highest or lowest priority. To arbitrate more fairly, the accepted (granted) applications of the previous cycle are saved and examined in the next arbitration cycle. In the proposed method, when there is competition between several virtual channels to choose an output port, several important factors are involved in arbitration and awarding a grant to a virtual channel. The request to leave the virtual channel, the value of the previous grant cycle of the virtual channel, and requests from other channels are essential parameters in the arbitration process among virtual channels. The RR-SFVP method employs a variable priority mechanism associated with each channel, which is dynamically adjusted based on the history of requests and the current state of the channels. This design choice leads to a just and equitable system, ensuring fairness in arbitration requests and reassuring users. The RR-SFVP method reserves the details about the requests granted in the previous cycle, ensuring no requestor is indefinitely blocked. This feature, coupled with the system's ability to grant access to the shared resource over successive cycles, instills a strong sense of reliability, making the audience feel secure and confident in its performance. The RR-SFVP method allows the arbitration unit, a component responsible for managing and resolving conflicts over resource access, to simultaneously assess and make decisions for multiple channels. This parallel processing capability leads to a faster selection process, enhancing the system's ability to evaluate and make decisions for various channels simultaneously and making the audience feel the speed of the method. In the following, if r is used in the figures or text of the article, it is the abbreviation of request. Fig. 6 shows the logic circuit of the router arbiter logic block diagram.

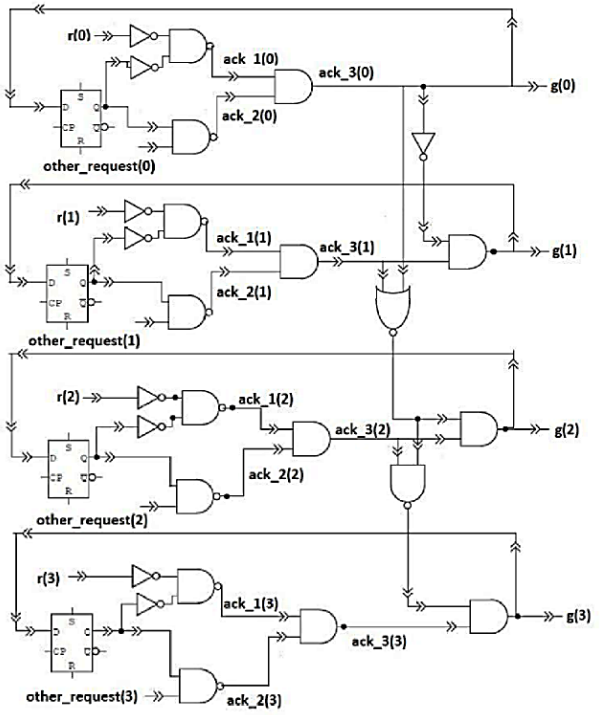


Fig. 6: RR-SFVP method arbiter circuit.

As the circuit in Fig. 6 shows, the parameters ri, Flip-Flop (i), and Other-request (i) affect the result of Gi. The flip-flops related to the circuit are initially set to 0. After each arbitration series, the rst (i) signal of the Flip-Flop (i) is adjusted by the circuit in Fig. 7.

This circuit is designed to reset a channel's flip-flop after it has been given access to avoid the indefinite blockage of other channels.

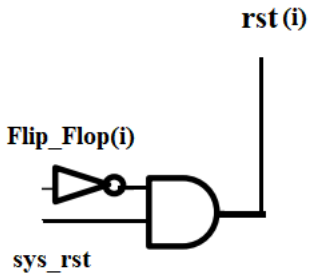


Fig. 7: Restart circuit of flip-flop.

Based on Fig. 7, any request that receives approval will not cause the flip-flop of the corresponding channel to reset during a general restart. Implementing this scenario will elevate the precedence of other virtual channels (which have not been provided). The sys_rst signal is a system-wide reset signal triggered at each arbitration process's conclusion. If the current request, rn, is active (1) and the flip-flop, fn, from the previous cycle, has a value of 0, indicating that the last cycle's gn was zero, the grant will be sent to n. This ensures that rn is not starved, even if another request, request (n), is active (equal to 1). The value of the variable "other_request" is set to 1 when at least one of the other ports of the virtual channel has submitted an exit request. In other words, the result may be expressed as the logical OR operation of all the individual exit requests (r0 OR r1 OR r2 ... OR rn-1).

If the current request, rn, is not active (0) and the flip-flop, fn, has a value of zero, regardless of the value of the parameter, other_request(n), the value of gn will be zero.

If the current request, rn, is active (1) and the flip-flop, fn, from the previous cycle, has a value of 0, indicating that the last cycle's gn was zero, then the grant will be given to n. This ensures that rn is not deprived, even if another _request (n) is active (equal to 1). The value of other_request is set to 1 when any other virtual channel ports have submitted an exit request. In other words, it may be expressed as the logical OR operation of all the exit requests (r0 or r1 or r2 ….. or rn-1).

If the rn request is disabled (0) and the flip-flop fn is set to zero, regardless of the value of the parameter other_request(n), the value of gn will be zero.

The procedure shown in Fig. 8 is used to find the appropriate factors for each channel. Upon receiving the request signal to exit channel i, the arbitration unit examines the contents of Flip-Flop (i) and the requests from other channels (i), respectively.

The suggested technique offers a distinct benefit, as seen in the algorithm depicted in *. Up to the line indicated with *, all the channels are processed simultaneously and without interdependence. However, the value of the grants is still influenced by the values of many other grants, as seen in circuit diagram 4. However, it should be noted that the principal channels do not have a high priority because the two channels have different funding.

```
Step 1: Set all Flip-Flop(N) =0
Step 2: Start
Step 3: Declare variables Flip-Flop[i],
        r[i], Other-request [i]
Step 4: Initialize variables
Step 5: Repeat the steps until i=N
   5.1: If i > N
          i <=0
        Else
read the value of Flip-Flop[i]
read the value of r [i]
read value of Other-request [i]

    ack_1 [i] = not(Flip-Flop[i]) nand
          not(r [i] )
    ack_2 [i] = Flip-Flop[i] nand
         Other-request [i])

 ack_3 [i] = ack_1 [i] and  ack_2[i]  *

        If  ack3[i]=1

          If (ack3[i-1] and ack[i-2],…)=0
          Gn=1

       Else

          Gn=0

  i<= i+1

step 6: Flip-flop(i) = Gi
step 7: Restart Flip-Flip(N)
step 8: Stop
```

Fig. 8: The algorithm of RR-SFVP method.

The conventional round-robin approach is not susceptible to hunger. However, this technique offers a benefit over the round-robin method since it does not need a sequential grant and allows for variable priority. Consequently, this method significantly improves arbitration time and throughput rate by checking all channels in every clock cycle. There is no need to verify their turn.

The suggested technique has a benefit, as seen in the algorithm depicted in Up to the line indicated with *; all the channels run concurrently and autonomously, without dependence on each other. However, following the intended line and circuit diagram 4, the magnitude of the grants is influenced by the magnitudes of the prior awards. However, it should be noted that the principal channels do not have a high priority because the two channels have distinct grants. The conventional round-

robin approach does not suffer from famine. However, this technique offers a benefit over the round-robin method since it does not need a sequential grant and allows for variable priority. Consequently, this method significantly improves arbitration time and throughput rate by checking all channels in every clock cycle. There is no need to verify their turn.

Based on the information provided in Fig. 9, two methods exist to get the virtual channel grant (G_i=1). In the first stage, Flip-Flop (i) has a value of 1, r(i) associated with exiting the virtual channel is likewise 1, and other_request(i) is 0.

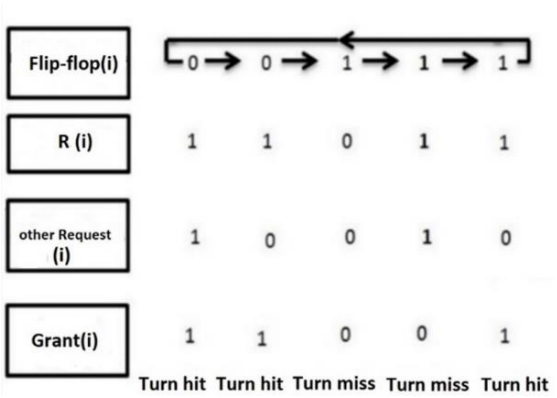


Fig. 9: Scenario of RR-SFVP method.

The second criterion for establishing the virtual channel grant is when Flip-flop(i) is equal to zero, r(i) associated with leaving the virtual channel is equal to 1, and the value of the other request is irrelevant. Furthermore, based on the shown situation in Fig. 9, two situations exist in which no grant is allocated for virtual channel I (Gi=0). The initial state occurs when all three values of signals Flip-flop (i), r (i), and Other-request (i) are simultaneously set to one. Alternatively, when Flip-flop (i) has a value of 1, the signal r(i) corresponds to the act of exiting the virtual channel, while the other request (i) is set to 0.

C. Functional and Fairness Analysis RR-SFVP

The circuit design is fair from two points of view. We consider scenarios based on the circuit in Fig.8 to prove this issue. First, I think a scenario where we want to prove that if a request is repeated repeatedly, the circuit acts fairly and repeated grants to It does not make requests and does not starve other requests; for example, in the first clock, channel number 1 only made a request and other channels did not make a request, according to the designed circuit and the logical analysis of the parameters of the zeroth virtual channel, g_0 is equal to zero, but in the circuit corresponding to the first virtual channel Considering that the previous value of the flip-flop is zero and r_1=1, the value of the virtual channel grant becomes one. In the next cycle, we assume that the first and third virtual channels reissue a request simultaneously. In this case, according to the circuit of the virtual channel, I am zero. According to logical analysis and r_0=0, the amount of grant g_0 is equal to zero, but in the first virtual channel, even though the request is issued But according to the logic analysis (other requests = 1 and the flip-flop value of virtual channel 1 = 1), therefore the grant is not assigned to the first virtual channel and the second virtual channel is the owner of the grant. This performance shows that the circuit is pretty fair and performs logical analysis in parallel, but it also pays attention to the requests of other channels.

According to the second point of view, which is a seal of approval on the fairness of the designed circuit, if no request is issued from any virtual channel with four inputs (0000), according to the circuit and digital analysis, the last flip-flop of the virtual channel, which is one, is used as ID grant. It is shown whether the virtual channel request is 1 or 0. This type of circuit operation reduces the circuit search delay and specifies the next clock from which the virtual channel ID of the arbitration operation should start.

One key factor contributing to the fairness of this arbitration unit is the reserve from the previous award cycle. When the time reaches zero, all requests are stored to enhance the evaluation efficiency and provide a more equitable selection process for the final award cycle. To better comprehend this subject matter, Fig. 10 and its corresponding timing diagram visually show the suggested approach's performance and behavior.

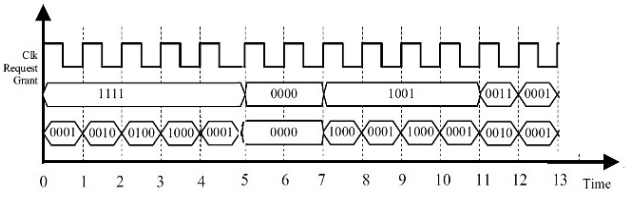


Fig. 10: Timing diagram for input request scenarios of strong fairness proposed method [26].

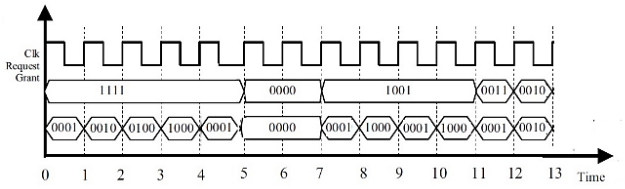


Fig. 11: Timing diagram for some input request scenarios of weak fairness arbiters [26].

Fig. 10 shows that from times 1 to 5, a constant input request of "1111" is applied, and each bit is given in every clock cycle. However, at time 5, the request will be modified to "0000", indicating that no request will be submitted. In the absence of a request, the priority of the previous request is logged and stored until a new request is initiated. For instance, at time 5, the priority of the

second bit of the save request is implemented, and at time 7, it is used for verification. Therefore, the request assumes grant ownership at time 7 in the fourth iteration. This implies that, as with other unjust systems, there is no need to initiate arbitration from the first request.

Fig. 11 indicates that approaches with inadequate fairness do not account for a scenario without requests. If, at time five, the system does not have any pending requests since the previous grant was not saved, and a request is submitted at time seven, the arbitration procedure restarts from the beginning of the relevant circuit. This condition presents a deficit in fairness. Non-compliance with the final priority, when not explicitly asked for, significantly affects the overall fairness of this suggested arbitration unit, guaranteeing a dependable and just procedure.

Fig. 10 illustrates the timing diagram for our round-robin arbiter to demonstrate its functionality and performance. A constant input request, "1111," is implemented throughout time intervals 1-5 and is partly satisfied in each clock cycle. The request is altered to "0000" at time 5, indicating no request is sent. In the absence of a request, the most recently authorized request's priority is acknowledged and included in any subsequent request. The second-bit priority of the request is recorded at time 6 and implemented at time 7. Consequently, at time 7, the fourth request is executed. We assessed our arbiter alongside many others (RoR, Matrix, PRRA, IPRRA, and HDRA) under identical testbench and request conditions. The time findings are shown in Figs. 10 and 11. The RoR, Matrix, and our IRR arbiters document the current priority shown in Fig. 8 when no request is submitted. However, the PRRA, IPRRA, and HDRA arbiters could neither exhibit the diverse waveforms seen in Fig. 11 nor document the priority. In the absence of a request, the least significant request bit for PRRA, IPRRA, and HDRA waveforms is assigned the greatest priority. The lack of a circuit to address the no-request scenario is the reason for the arbitration conducted by PRRA, IPRRA, and HDRA. The impartiality of an adjudicator is directly influenced by upholding the lowest priority under the no-request condition. The key advantage of our RR-SFVP arbitrator is its capacity to provide a more rigorous fairness arbitration.

The RR-SFVP technique utilizes parallel processing of all channels using the suggested algorithm. It also considers other requests and examines the request history of one channel. This approach ensures a fairer and more robust arbitration procedure. This form of arbitration has used the input port of the mechanism described in [34]. Upon entry, each flit is assigned to a specific virtual channel after verifying the availability of buffer slots and the emptiness of each section using the write pointer and header flit. The identifier associated with the header flit is allocated to the matching flit identifier.

To accept a new flit, it is necessary to identify the vacant slots in the shared buffer during the second step after selecting one of the virtual channels.

To assess the arbitration scheme discussed, the input port in the router must be used. As described in [31], this input port is responsible for reading and writing a portion of the input port to and from the buffer. One of the benefits of this approach is its utilization of a table and two straightforward read-and-write circuits. The input port of this approach has a hardware design that facilitates parallel processing. Simultaneously, the first vacant position in the buffer is located for writing, and straightforward and parallel hardware, as described in [28], is used to read from the address specified in the buffer.

D. Throughput Analysis RR-SFVP

Based on the circuit designed in Fig. 6, we check the throughput rate from 3 points. The first one is that since the throughput rate has an inverse relationship with the delay, the critical path delay of the circuit is less compared to other circuits, according to Table 3, which increases the throughput rate. The second case refers to the hardware structure that does not exist in the block operating circuit. As said in the fairness assessment section, every virtual channel with the highest priority in the current cycle has the lowest priority in the next cycle. This issue causes the operation Arbitration to be done faster, increasing the throughput rate. The third case goes back to the issue of the fairness of the circuit, that the ID of the last granted virtual channel is taken, considering that the corresponding flip-flop is also 1. To facilitate finding the following grant, it will be easier to find the following grant, and in a way, the time to see it will be faster, with less delay as a result. The permeability increases.

Evaluation of the Proposed Method

This part involves simulating the RR-SFVP router and comparing it with comparable scenarios based on its structure and architecture.

The primary performance indicators for the designed circuit are its speed, power consumption, and area. These metrics often quantify an arbitration circuit's speed, latency, or maximum frequency (F_{max}). The frequency of the arbitration circuit is determined by the most extended delay (critical route) between two registers at any given moment. The RR-SFVP approach is quite significant in this particular circumstance.

A. RR-SFVP Hardware Requirements

We have assessed the NoCs above based on primary hardware attributes, including power consumption, chip size, and speed, as determined by Verilog implementation using Synopsys Design Compiler. Evaluation results are obtained using ASIC technology libraries, such as 90 nm NanGate [32]. The configuration for the input-port ASIC's

power and area use the CMOS technology specifications from the Synopsys Generic 90nm Library, with a global operating voltage of 1.2V and a period of 400MHz.

Table 2 shows that the electrical characteristics of the logic gates are derived from the standard Synopsys 90 nm Digital library.

Table 2: Characteristics of gates

Gate name St.	Propagation Delay (ps)	Power Dy. (nW)	*Power (nW/MHz)*	Area (µm2)
INVX1	38	88	*12*	6.5
AND2X1	85	298	*19*	7.4
AND3X1	119	297	*34*	8.3
NAND2X1	51	336	*15*	5.5
OR2X1	85	226	*23*	7.4
OR3X1	114	250	*39*	9.2
OR4X1	137	261	*56*	10.1
NOR2X1	64	170	*15*	6.5
MUX21X1	107	815	*43*	11.1
MUX41X1	168	827	*58*	23.0
DEC24X1	119	1238	*66*	29.5
XOR2X1	133	454	*26*	13.8
DFFARX1	217	620	*100*	32.2

We conduct a hardware overhead study to evaluate the anticipated speed and hardware overhead of the previously described round-robin arbiters compared to our suggested arbiter. We do not use any algorithms to optimize the circuits like Electronic Design Automation software does.

The primary performance metrics of an arbiter circuit are speed, area, and power consumption. The standard metric for the speed of an arbiter circuit is the delay time or the maximum clock frequency (fmax). The clock frequency of an arbiter is determined by the maximum delay (critical path) between two concurrently timed registers. The circuits of 4-input arbiters are analyzed at the gate level. The electrical characteristics of the logic gates are obtained from the Synopsys 90nm Digital Standard Cell Library, as shown in Table 2.

We computed the aggregate of the areas and powers of all the cells for each arbiter to assess their power and area, as shown in Table 3.

The power encompasses both static and dynamic components. The critical route delay between two registers in each circuit is computed for speed estimate. The critical route for each circuit is shown by the numbers in parenthesis in the last column of Table 3. There is a consistent correlation between power consumption, critical route, and consumption area. These three variables are crucial in determining the most effective design of the arbitration unit. The typical metric to quantify an arbiter circuit's speed is the clock frequency's time or value (Fmax).

Table 3: Characteristics of 4-Input Arbiters based on Table 2

Type of 4-input arbiters	Area (µm2)	Power (µW)	Critical Path Delay (PS)
IRR	294	296 (282d)	625 (217+133+168+107)
RoR	328	298 (289d)	1242 (217+5*(85+85) +137+38)
Matrix	556	479 (465d)	747 (217 +2*38+3*85+114+85)
IRR_WF	280	274 (262 d)	518 (217+133+168)
HDRA	431	360 (348d)	609 (217 +64+51+85+85+107)
PRRA	510	493 (479d)	861(217+2*38+3*85+85+2*114)
IPRRA	528	488 (473d)	747 (217+2*38 +3*85+85+114)
RR-SFVP	288	285 (271d)	557 (217+38+3*51+85+64)
Arbiters	Saving	Saving	Faster
RR-SFVP / IRR	2% (better)	3.7% (better)	10.88%(better)
RR-SFVP / RoR	12.1% (better)	4.3% (better)	55.15%(better)
RR-SFVP / Matrix	48% (better)	40% (better)	25.5%(better)
RR-SFVP / IRR_WF	2.8% (worse)	4% (worse)	7.5%(worse)
RR-SFVP / HDRA	33% (better)	20% (better)	8.2%(better)
RR-SFVP / PRRA	44% (better)	42% (better)	35%(better)
RR-SFVP / IPRRA	45% (better)	41% (better)	25%(better)

B. Hardware Parameter Analysis

We evaluate the parameters of the designed circuit from two dimensions: 1- hardware and 2- network on chip. From the hardware point of view, the parameters of the circuit area, power consumption, and critical path are comparable. The type and size of the desired gate affect the area and power consumption of the circuit.

The critical path length should be shorter to reduce the delay and increase the throughput of the circuit. According to Fig. 6, gates have optimized the critical path delay. Table 3 pertains to circuit design. The first column displays the space taken up by the arbiter unit, depending on the gate used. The second column indicates the power spent, while the final column represents the delay on the critical route.

Table 3 demonstrates that the RR-SFVP technique has lower power consumption, area, and critical path than other ways, except the IRR_WF method, based on the hardware design. This technique, which incorporates weak fairness, exhibits a reduced footprint, leading to decreased power consumption and a shorter critical path than the RR-SFVP Three primary elements. This rigorous examination method guarantees the comprehensiveness of our study.

C. Performance Evaluation of RR-SFVP NoC

Latency and throughput are the primary performance metrics for assessing RR-SFVP NoC. The NoCs are built in System Verilog, and we use the ISE 14.4 simulation environment to get these performance metrics. A 8*8 mesh topology with five input/output and wormhole switching is considered. There are four VCs in every input port. According to (1), throughput is measured by the rate of receiving packets to the maximum number of packets injected at a given time. Time, which is 20ns in this evaluation as (1). The packet communication utilizes wormhole switching, with the channel width corresponding to the flit size of 16 bits. A packet has 16 flits; each input port contains a central 8-slot buffer. Each input port has four virtual channels, except for ViChaR, which includes four virtual channels corresponding to the number of buffer slots in the input port. Throughput and delay are assessed based on flit injection rates per time unit. Fig. 12 shows the Simulation waveform of the proposed arbiter architecture in the Xilinx ISE 14.4 simulator for the delayed flit departure from a router.

The suggested technique is assessed using synthetic benchmarks and actual application traffic, showcasing its potential advantages and performance enhancements.

Table 4 displays the comprehensive attributes of the redesigned NoC architecture.

Table 4: Structure of simulation parameters

Network Size	**8 x 8**
Packet/Flit/Data	16-bits
VC and I/O ports	4 VCs for each of the five ports
Switching mode	Wormhole
Topology	Mesh
Routing Algorithm	XY routing
Traffic Patterns	Tornado, Complement, Random MPEG, AV.

The different metrics, such as latency and throughput, have been measured and thoroughly evaluated using microarchitecture and Verilog simulation. We conducted measurements of both throughput and delay. Throughput is determined by calculating the rate at which packets are received compared to the maximum number of packets injected at a certain period. This may be represented as follows (1):

$$\text{Throughput}=\frac{\text{number of received packets} \times \text{ size of one packet}}{\text{number of nodes} \times \text{number of cycles}} \quad (1)$$

Equation (2) measures the average delay resulting from the average latency associated with the entrance and departure of a certain number of packets in a Network-on-Chip (NoC) during each clock cycle.

$$\text{latency}= |departure\ (time) - arrival(time)| \quad (2)$$

Fig. 12: Simulation waveform of router architecture.

The results of Fig. 12, which is done in the simulation environment of the ise 14.4 software, show the transfers of flits between the ports of different routers; for example, three flits have simultaneously requested in output virtual channel 3, which, according to the explanations about the priorities of the designed circuit, all the data enters the desired output channel with the least delay and without data loss. This circuit behavior plays a significant role in increasing the circuit's throughput.

D. RR-SFVP NoC Analysis Parameters

In this part, we will examine the circuit designed for its application in the network on the chip. The two parameters of delay and throughput are among the parameters that affect the performance of a good arbiter unit. Suppose we analyze the delay in the proposed method based on the designed circuit and the existing scenario. In that case, we will see that the delay between two flip-flops is optimal compared to the previous methods, even though the flip-flop in the circuit has a relatively high delay compared to other gates. But, considering that it increases the fairness of the arbitration unit and preserves the previous grant cycle, it is one of the advantages of this circuit.

On the other hand, considering that the ID of the last cycle grant is preserved, it saves time when searching for the next cycle. It reduces the delay and somehow increases the throughput of the arbitration unit. On the other hand, as mentioned in the hardware review section, the lower the hardware overhead, the lower the critical path delay, and as a result, according to (1), increases the throughput.

E. Evaluation of the Performance of the New Arbitration Unit in Synthetic Benchmarks

Packet communication relies on the use of wormhole switching. The channel width is equivalent to the flit size, which is 16 bits. A packet consists of 16 flits; each input port has a central buffer that can hold up to 8 slots. Each input port has four virtual channels, except for ViChaR, which has eight. The number of virtual channels in ViChaR equals the number of buffer slots in the input port. The flit injection rates per time unit are used to quantify the throughput and delay. For instance, a flit injection rate of 8 indicates that each node, namely the source core, injects eight flits every unit of time. The maximum injection rate is determined by the capacity of the NoC routers to transmit the flits. As previously stated, the arrival and departure of flits using this approach with input port [28] routers takes one cycle, but it takes two cycles for LLD and ViChaR-based routers.

Thus, considering a time unit equivalent to 16 clock cycles, the LLD and ViChaR-based sources may inject a maximum of eight flits. It is not feasible to inject more than eight flits. Nevertheless, the method-based source cores can only inject a maximum of 16 flits every time unit, making it impossible to inject more than 16 flits. Our simulation considers a maximum of 8 flit injection rates for RR-SFVP to provide a fair comparison.

The performance metrics of each Network-on-Chip (NoC) are influenced by the functional behavior of the data flow mechanism and the temporal characteristics of the router. When analyzing the performance of a Network-on-Chip (NoC), it is essential to consider the delays associated with the router on the critical path. Consequently, we evaluate these Network-on-Chips (NoCs) at various clock frequencies based on the essential path delays linked to their routers. The experiment assesses the performance parameters of the NoCs mentioned above, and the findings are shown in Figures 13, 14, and 15. There is a direct correlation between the clock rate and the performance measures. Let's consider that n packets transit through the NoC system during t, with a clock rate of f. During a period of t, the NoC system will transmit p×n packets at a clock rate of p×f. Figs. 13, 14 and 15 demonstrate the superior performance of our strategy compared to others. When the injection rates are increased, more flits are injected into the NoCs, resulting in a higher population level and more disagreement.

Regarding functionality, the LLD-RoR, LLD-Matrix, and LLD-HDRA operate similarly at the same frequency because the RoR arbiter is extremely similar to the HDRA arbiter covered in the function and fairness section. Thus, in terms of performance, the LLD-Matrix with a faster clock rate outperforms the LLD-RoR and LLD-HDRA NoCs. The ViChaR-Matrix NoC leads to the same conclusion. Consequently, four fast NoCs are chosen for assessment and comparison: LLD-Matrix, ViChaR-Matrix, EDVC-IRR, and router with input port [31] and RR-SFVP arbitration. With a 4-VC setup, the LLD-Matrix, ViChaR-Matrix, EDVC-IRR, and RR-SFVP operate at 514, 451, 820, and 1000 MHz clock, respectively. The frequencies above are derived from the critical path of the delays listed in Table 3.

The performance of a Network-on-Chip (NoC) is determined by the behavior of its data flow mechanism and its timing. Attributes of a router the evaluation of NoC performance must consider the critical path delays associated with the router. Consequently, we evaluate these Network-on-Chips (NoCs) at several clock frequencies based on the critical route.

The test evaluates the delays of their routers and the performance characteristics of the NoCs. The findings are shown in Fig. 13, Fig. 14 and Fig. 15. The performance measurements are directly proportional to the clock rate and NoC frequency.

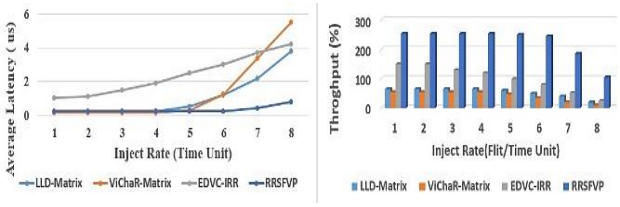


Fig. 13: Latency and average throughput for random traffic.

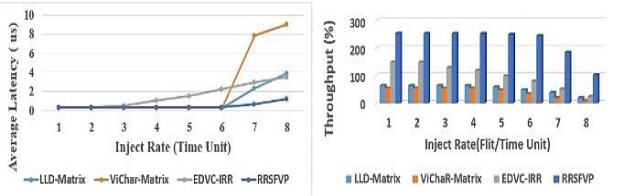


Fig. 14: Latency and average throughput for tornado traffic.

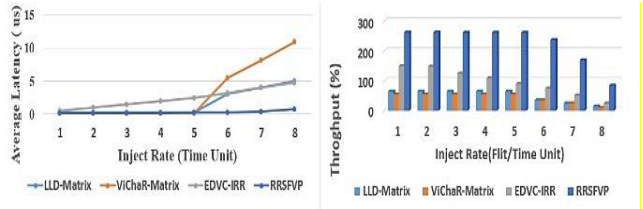


Fig. 15: Latency and average throughput for complement traffic.

Figs. 13, 14 and 15 show the mean delay and throughput requirements for the mesh topology (8×8) about the Complement and Random Tornado traffic patterns [33]-[35], respectively. Equations (3), (4) and (5) calculate the source address (Sx, Sy) and destination address (Dx, Dy) for Tornado, Complement, and random traffic patterns in a mesh topography of size m × m, where $0 < x, y \leq m-1$.

For Tornado:

$$Dx = Sx + (m/2)\text{-}1, Dy = Sy + (m/2)\text{-}1 \quad (3)$$

For Complement:

$$Dx = m\text{-}Sx\text{-}1, Dy = m\text{-}Sy\text{-}1 \quad (4)$$

For Random:

$$Dx = 1/m, Dy = 1/m \quad (5)$$

In XY routing and tornado traffic, all routers experience uniform congestion. Conversely, in complement traffic, the congestion is higher in side routers compared to middle routers. In the case of a random type, the packet is equally likely to be sent to other nodes. The experiment is conducted on an 8×8 network, where each packet comprises 16 flits. Additionally, each input port has a central buffer containing eight slots.

Arbitration systems that include diverse inputs exhibit superior efficiency compared to alternative arbiters. Typically, the chip area is smaller, the power consumption is decreased, and the critical path value is reduced. Due to fewer gates, this technique often has the lowest power consumption compared to other arbiters. Using fewer gates further streamlines the chip's architecture and arrangement.

Based on the data shown in Figs. 13, 14 and 15, the RR-SFVP's test results surpass those of other techniques, particularly at high rates. The number of flits rises with greater doses, intensifying rivalry among them.

There is a consistent correlation between the level of delay and the level of throughput. However, it should be noted that the suggested technique of fake traffic mentions at the beginning of the assessment section that the flits acceptance capacity at the entrance port of the relevant arbitration unit is twice as much as other approaches, namely two cycles. Consequently, the latency may be increased due to this factor, but this leads to a high throughput rate, as shown by expression 2.

Implementing ViChaR [23] for 4VC and 8-slots demonstrates significantly improved output NoC performance and reduced average latency compared to both Link-List and ViChaR NoC for various traffic scenarios. The acceleration is attributed to the input port's performance, directly reducing the number of executable cycles. Furthermore, the channels are simultaneously and concurrently verified by the arbitration unit in a specific section of the circuit, resulting in a better processing speed owing to the circuit's architecture. In contrast to other methods, the RR-SFVP selects the desired port in a parallel and simultaneous manner across multiple channels. This allows for efficient processing of requests and enables the system to make the best choice based on the criteria outlined in the RR-SFVP section.

F. Evaluation of the Performance of the Arbitration Unit in Real Benchmarks (Applications):

To further examine the effectiveness of the suggested technique, we assessed the proposed method's impact on the performance of two Network-on-Chip (NoC) applications, namely MPEG-4 and AV [33].

We conducted measurements of throughput and latency. Throughput was determined by calculating the rate at which packets were received and the maximum number of packets injected during a specific time frame. The average latency is determined by calculating the average time delays per clock cycle when a certain amount of packets are sent and received in the NoC. We vary the packet injection rates to assess the performance of the application-specific traffic. The rate of packet injection is modified per unit of time. The maximum bandwidth of the source cores defines the time unit. For example, the MPEG4 decoder's Source Core#8 has a maximum bandwidth of 1580 flits. Hence, the time unit will be 1580 clock cycles, assuming that each source core injects one flit each clock cycle. The AV (Audio-Video) application requires precise measurement. Specifically, Core#14 has a maximum bandwidth of 192078 flits per 192078 clock cycles.

MPEG-4 AV programs consisting of audio and video are mapped onto 2D chips with a mesh topology. The programs are mapped to dimensions of 3x4 and 4x4, respectively. Fig. 16 and Fig. 17 show diagrams of MPEG-4 and AV applications.

Packet communication utilizes wormhole routing and adheres to a specific XY routing algorithm. The arrow lines indicate the packet's route from the source cores to the destinations.

The assessment criteria in this experiment are identical to those in the previous one, with a total of 4 virtual channels and a 16-bit per flit.

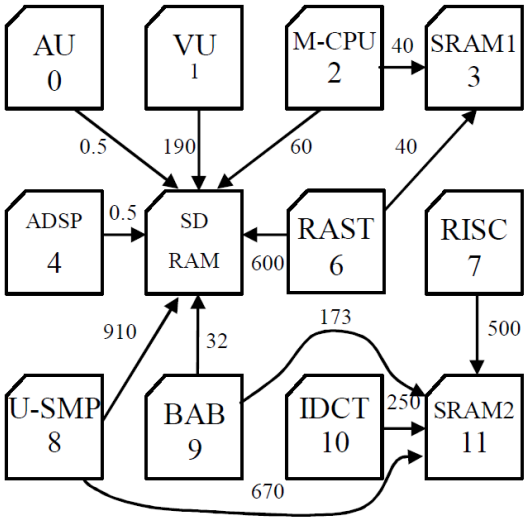


Fig. 16: Mapping of MPEG-4 core graphs to a 3×4 Mesh Topology NOC [36].

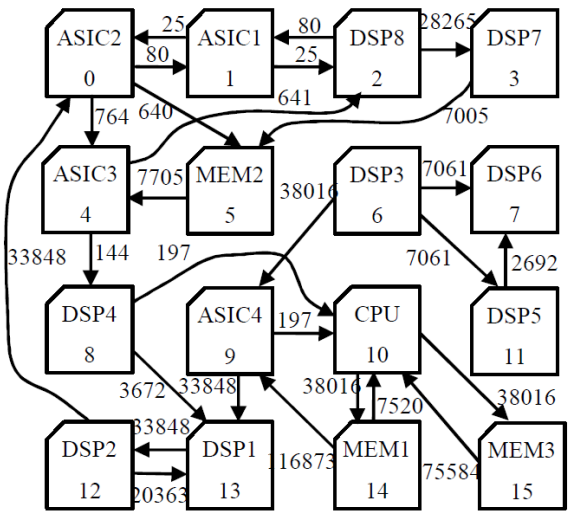


Fig. 17: Mapping of AV core graphs to 4×4 Mesh Topology NOC [36].

The delay occurs throughout the transmission of packets to all destinations, such as MPEG-4 and AV targets, with corresponding packet sizes of 55,472 and 380,128. The disparities among these three Network-on-Chip (NoC) applications are insignificant, primarily due to two specific circumstances. Initially, we configured VC-4 for every input port, surpassing the maximum number of required VCs in these applications. Fig. 16 and Fig. 17 show the most requested VCs in MPEG-4, AV, and 3 and 2. Furthermore, the packet traverses many ways.

They are not experiencing congestion. For example, the input channel on the western side of MPEG-4 router #11 has the most significant packet flow.

According to Fig. 18 and Fig. 19, the findings indicate that the delay caused by RR-SFVP is 72%, 92%, and 78% less than the average delay caused by LLD-Matrix for MPEG-4, AV, and applications.

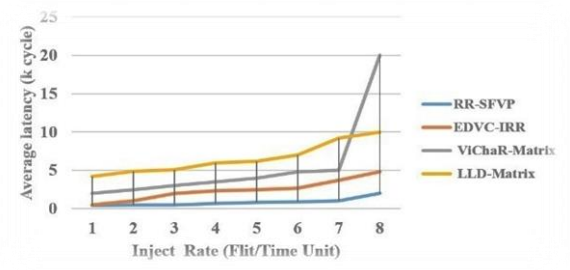


Fig. 18: Latency for MPEG-4, for 3×4 mesh NoCs.

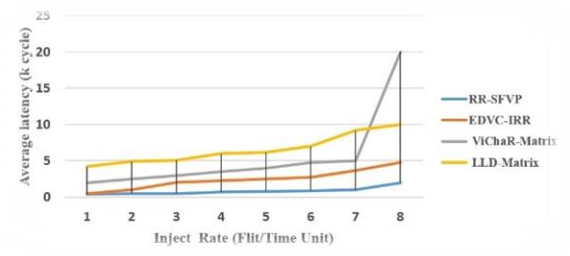


Fig. 19: Latency for AV, for 4×4 mesh NoCs.

Results and Discussion

In this article, a method for having a network router unit on a chip was proposed, which had several advantages over the previous techniques, the most important of which was its variable priority, which means that input does not always have the highest priority, even though the design circuit It is done sequentially, that is, the system is designed in such a way that fairness is considered among all requests. The second and most important advantage of the designed circuit is to save the last granted virtual channel ID, which, in addition to keeping the granted virtual channel ID, saves time in the next cycle to check the requested virtual channels. The third advantage of the designed circuit is shortening the critical path, saving area, and power consumption.

Conclusion

This study introduces the RR-SFVP microarchitecture, a modified version of the RR technique. The buffer employs an arbitration unit to pick a port from several ports based on essential priorities among various virtual channels (VCs).

This ensures that no port is deprived of resources and that no port is given greater priority than others. Assessments indicate that the RR-SFVP approach exhibits lower area and power usage than other methods. Compared to the RR arbitration unit, one of the conventional techniques, it has achieved a 55.1% reduction in critical route latency and a 12.1% drop in power consumption, improving space efficiency by 4.3%. However, the simulation findings demonstrate that the RR-SFVP approach, compared to the IRR method, a relatively recent arbitration method, exhibits a 2% reduction in area, a 3.7% decrease in power consumption, and a 10.88% decrease in critical route latency. Future work might include suggesting changes to the arbitration unit design, such as using shared comparison gates to decrease vital path delays.

Author Contributions

Elnaz Shafigh Fard conceived and designed the analysis and contributed data or analysis tools. Mohammad Ali Jabraeil Jamali performed the analysis. Mohammad Masdari collected the data. Kambiz Majidzadeh contributed to the interpretation of the results. All authors reviewed the final manuscript.

Acknowledgment

The article's authors have not received any financial support from any organization concerning this research.

Conflict of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Abbreviations

NOC	*Network On Chip*
VC	*Virtual Channel*
DAMQ	*Dynamically Allocated Multi-Queue*
ViChaR	*Virtual Channel Regulator*
EDVC	*Efficient Dynamic Virtual Channel*
FIFO	*First-in First-out*
FPGA	*Field programmable gate array*
HoL	*Head Of Line blocking problem*
HDRA	*High-speed and Decentralized Round robin*
IPRRA	*Improved parallel round-robin arbiter*
IRR_WF	*Index round-robin weak fairness*
HDRA	*High-speed and decentralized round-robin arbiter*
LLD	*Linked-List based DAMQ*
MPEG	*Moving Picture Experts Group*
AV	*Audio-Video Benchmark*

References

[1] L. Benini, G. D. Micheli, T. Tao Ye, Networks on chips, Vol. 1. Burlington: Morgan Kaufmann, 2006.

[2] R. Akbar, F. Safaei, “A novel heterogeneous congestion criterion for mesh-based networks-on-chip,” Microprocess. Microsyst., 84: 104056, 2021.

[3] A. T S, B. M, U. Sankar S M, R. Thiagarajan, A. Daltan G, P. Raja Rajeshwari, A. Sai Kumar et al., “Evaluation of low power consumption network on chip routing architecture,” Microprocess. Microsyst., 82: 103809, 2021.

[4] F.Rad, M. Reshadi, A. Khademzadeh, “A novel arbitration mechanism for crossbar switch in wireless network-on-chip,” Cluster Comput., 24: 1185-1198, 2021.

[5] S. M. Mamaghani, M. A. Jabraeil Jamali, “A load-balanced congestion-aware routing algorithm based on time interval in wireless network-on-chip,” J. Ambient Intell. Hum. Comput., 10: 2869-2882, 2019.

[6] M. Nirmala, V. Shylaja, “FPGA implementation of priority arbiter-based router design for NOC systems,” Int. J. Adv. Res. Eng. Technol. (IJARET), 10(2): 509-516, 2019.

[7] M. Rezaei-Zare, M. Fathy, A. Rezaei-Z, “Design of a high-performance router with distributed shared-buffer for load balancing for on-chip networks,” Microelectronics Journal, 132(c): 105647, 2023.

[8] B. Bahrami, M. A. Jabraeil Jamali, S. Saeidi, “ A novel hierarchical architecture for wireless network-on-chip,” J. Parallel Distrib. Comput., 120: 307-321, 2018.

[9] Y. Long Lan, V. Muthukumar, “Efficient virtual channel allocator for noc router microarchitecture,” in Proc. 30th IEEE International System-on-Chip Conference (SOCC), 2017.

[10] M. T. Balakrishnan, T. G. Venkatesh, A. Vijaya Bhaskar, “Design and implementation of congestion aware router for network-on-chip,” Integration, 88: 43-57, 2023.

[11] W. Zhou, Y. Ouyang, Y. Lu b, H. Liang, “A router architecture with dual input and output channels for Networks-on-Chip,” Microprocess. Microsyst., 90: 104464, 2022.

[12] L. Liu, Z. Zhu, D. Zhou, Y. Yang, “A fair arbitration for Network-on-Chip routing with odd-even turn model,” Microelectron. J., 64: 1-8, 2017.

[13] J. Rostami Monfared, A. Mousavi, “Design and simulation of nano-arbiters using quantum-dot cellular automata,” Microprocess. Microsyst., 72: 102926, 2020.

[14] R. Uma, H. Sarojadevi, V. Sanju, "Network-On-Chip (NoC) - routing techniques: A Study and analysis," in Proc. 2019 Global Conference for Advancement in Technology (GCAT), 2019.

[15] S. Sreekumar, “Literature survey on Network on Chips,” J. Emerging Technol. Innovative Res., 8(3): 1836-1840, 2021.

[16] W. J. Dally, B. Towles, Principles and Practices of Interconnection Networks, Morgan Kaufmann Publishers, Elsevier, 2004.

[17] J. K. Murthy, Nishanth. B. N, “Design and analysis of arbiters for the NoC's routers,” Int. J. Adv. Res. Electr., Electron. Instrum. Eng., 6(9): 6785-6792, 2017.

[18] A. Osovsky, D. Starov, O. Stukach, N. Maltseva, D. Surkov, “Crossbar switch arbitration with traffic control for NoC,” in Proc. International Siberian Conference on Control and Communications (SIBCON): 1-5, 2022.

[19] A. Toe, S. A. Dianat, Design and Verification of a Round-Robin rbiter, Theses 8-2018.

[20] T. Nadu, J. Arjana, “Low latency Noc with dynamic priority based matrix arbiter,” Indian J. Sci. Technol., 9(29), 2016.

[21] M. Evripidou, C. Nicopoulos, V. Soteriou, J. Kim, "Virtualizing virtual channels for increased network-onchip robustness and upgradeability," in Proc. IEEE Computer Society Annual Symposium on VLSI, 2012.

[22] C. A. Nicopoulos, P. Dongkook, K. Jongman, N. Vijaykrishnan, M. S. Yousif, C. R. Das, "ViChaR: A dynamic virtual channel regulator for network-on-chip routers," in Proc. 39th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO'06), 2006.

[23] Z. Fu, X. Ling, “The design and implementation of arbiters for Network-on-chips,” in Proc. 2nd Int. Conf. Industrial and Information Systems: 292-295, 2010.

[24] S. Q. Zheng, M. Yang, “Algorithm-Hardware codesign of fast parallelround-robin arbiters," IEEE Trans. Parallel Distrib. Syst., 18(1): 84-95, 2006.

[25] Y. L. Lee, J. M. Jou, Y. Y. Chen, "A high-speed and decentralized arbiter design for NoC," in Proc. IEEE/ACS Int. Conf. on Computer Systems and Applications: 350-353, 2009.

[26] M. Oveis-Gharan, G. N. Khan, "Index-based round-robin arbiter for NoC routers," in Proc. IEEE Computer Society Annual Symposium on VLSI: 62-67, 2015.

[27] O. Gharan, G. N. Khan, "Efficient dynamic virtual channel organization and architecture for NoC systems," IEEE Trans. Very Large Scale Integr. VLSI Syst., 24(2): 465-478, 2016.

[28] J. Luo, W. Wu, Q. Xing, M. Xue, F. Yu, Z. Ma, “A low-latency fair-arbiter architecture for network-on-chip switches,” Appl. Sci., 12(23): 12458, 2022.

[29] A. Mohanty, G. Chiranjeevi, “Comparative analysis of round robin arbiter and weighted round robin arbiter,” in Proc. 2024 Asia Pacific Conference on Innovation in Technology (APCIT), 2024.

[30] X. Cheng, Y. Wang, H. Jin, P. Li, “A novel bus arbiter based on group decision algorithm,” in Proc. IEEE 4th International Conference on Circuits and Systems, 2022.

[31] E. Shafigh Fard, M. A. Jabraeil Jamali, M. Masdari, K. Majidzadeh, “An efficient NoC router by optimal management of buffer read and write mechanism,” Microprocess. Microsyst., 89: 104440, 2022.

[32] S. Das, J. R. Doppa, P. P. Pande et al., “Energy-efficient and reliable 3D network-on-chip (NoC): Architectures and optimization algorithms,” in Proc. IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2016.

[33] N. Alfaraj, J. Zhang, Y. Xu, H. J. Chao, "HOPE: Hotspot congestion control for clos network on chip," in Proc. IEEE/ACM International Symposium on Networks on Chip: 17-24, 2011.

[34] Y. Xu, B. Zhao, Y. Zhang, J. Yang, "Simple virtual channel allocation for high throughput and high-frequency on-chip routers," in Proc. HPCA-16 2010 The Sixteenth International Symposium on High-Performance Computer Architecture: 1-11, 2010.

[35] V. Dumitriu, G. N. Khan, "Throughput-oriented NoC topology generation and analysis for performance SoCs," IEEE Trans. VLSI Syst., 17(10): 1433-1446, 2009.

[36] M. Oveis-Gharan, G. N. Khan, "Statically adaptive multi fifo buffer architecture for network on chip," Microprocess. Microsyst., 39(1): 11–26, 2015.

Biographies

Elnaz Shafigh Fard received a B.Sc. degree in Software Engineering from the Azad University of Shabestar, Iran, in 2001 and an M.S. degree in Computer Systems Architecture from the Azad University of Najaf Abad branch, Isfahan, Iran, in 2014. She is currently pursuing a Ph.D. degree at the Azad University of Urmia branch, Urmia, Iran. Her research interests include embedded system design, intelligent system design and modeling, system-on-chip design, and performance and latency optimization in network-on-chip architectures.

- Email: elnazsha9427@gmail.com
- ORCID: 0009-0004-2749-4044
- Web of Science Researcher ID: NA
- Scopus Author ID: NA
- Homepage: NA

Mohammad Ali Jabraeil Jamali received his B.Sc. in Electrical Engineering from Urmia University, Urmia, Iran, his M.Sc. from Tabriz University, Tabriz, Iran, the M.Sc. in Computer Engineering from Islamic Azad University, Science and Research Branch, Tehran, Iran and the Ph.D. in Computer Engineering from Islamic Azad University, Science and Research Branch, Tehran, Iran, in 1994, 1997, 2003 and 2009, respectively. He is an Assistant Professor of Computer Engineering at Islamic Azad University, Shabestar branch. He is the author/co-author of over 50 technical journals and conference publications. His current research interests are processor and computer architectures, chip multiprocessors, multiprocessor systems-on-chip, networks-on-chip, ad hoc and sensor networks, security, and the Internet of Things.

- Email: m_jamali@itrc.ac.ir
- ORCID: 0000-0001-7687-5469
- Web of Science Researcher ID: NA
- Scopus Author ID: 57193977114
- Homepage: https://scimet.iau.ir/MohammadAli_JabraeilJamali?%20highlyCited

Mohammad Masdari received his B.Tech. He earned a degree in Computer Software Engineering from Islamic Azad University, Qazvin Branch, Iran, in 2001 and an M.Tech degree in Computer Software Engineering from Islamic Azad University, South Tehran Branch, Tehran, Iran, in 2003. He received his Ph.D. in Computer Software Engineering from Islamic Azad University, Science and Research Branch, Tehran, Iran, in 2014. Since 2003, he worked as a faculty member of Islamic Azad University, Urmia branch, Iran. He is an Assistant Professor in the Department of Computer Engineering of Islamic Azad University, Urmia branch, Iran. His research interests include Distributed Systems and Network Security.

- Email: mo.masdari@iau.ac.ir
- ORCID: 0000-0002-7093-2204
- Web of Science Researcher ID: NA
- Scopus Author ID: NA
- Homepage: https://urmia.iau.ir/file/download/faculty/64f44124367a5-masdari.pdf

Kambiz Majidzadeh was born in Urmia, Iran, in 1980. He received a B.Sc. degree in Software Engineering from the Islamic Azad University of Khoy, Khoy, Iran, in 2002, M.Sc. degree in Computer Networking, and Ph.D. degree in Information Technology from Baku State University (BSU), Baku, Azerbaijan, in 2005 and 2009, respectively. His research interests are very large-scale.

- Email: k.majidzadeh@iaurmia.ac.ir
- ORCID: 0000-0001-5118-8910
- Web of Science Researcher ID: NA
- Scopus Author ID: 54880938600
- Homepage: https://urmia.iau.ir/file/download/faculty/67063cb4bdbd9-majidzadeh.pdf

How to cite this paper:
E. Shafigh Fard, M. A. Jabraeil Jamali, M. Masdari, K. Majidzadeh, "RR-SFVP: A novel arbitration unit design for NoC router, ingeniously fusing the round robin method with strong fairness and variable priority," J. Electr. Comput. Eng. Innovations, 13(2): 283-298, 2025.

DOI: 10.22061/jecei.2024.11230.779

URL: https://jecei.sru.ac.ir/article_2238.html