IJICTR
International Journal of Information & Communication Technology Research

Volume 11- Number 3 – Summer 2019 (31 -41)

DTIS: Destination Traffic Based Input Selection Strategy Based on Traffic Pattern for Network-on-Chip Systems

Amin Mehranzadeh
Department of Computer Engineering
Science and Research Branch,
Islamic Azad University
Tehran, Iran
a.mehranzadeh@srbiau.ac.ir

Ahmad Khademzadeh*
Iran Telecommunication
Research Center
Tehran, Iran
zadeh@itrc.ac.ir

Midia Reshadi
Department of Computer Engineering
Science and Research Branch,
Islamic Azad University
Tehran, Iran
reshadi@srbiau.ac.ir

Received: 3 March 2019 - Accepted: 12 June 2019

Abstract—**An input selection strategy is an important part of a router that is done by an arbitration process. When an output channel is requested by two or more input channels simultaneously, the best input channel will be selected by the input selection strategy. This research presents a new input selection strategy called DTIS (Destination Traffic based Input Selection). The DTIS uses local and non-local congestion information on the path to distribute traffic more evenly over the network. Also, a global congestion aware method called DCA is used to give priority to an input channel according to the destination. The simulation results prove that DTIS improves the average latency and throughput for various synthetic and real traffic patterns with acceptable overhead in terms of area consumption. The simulation results show the average delay improvements of DTIS to the CAIS and Round Robin strategies are 26% and 77%, respectively.**

Keywords-component; Network on Chip; Arbitration; Input selection strategy; Destination traffic; Performance evaluation;

I. INTRODUCTION

The network on chip (NoC) represents a flexible and scalable solution for designing the parallel and chip multiprocessor systems. Energy consumption, latency and throughput are the limiting factors that influence performance and efficiency of the NoC systems design [1-2]. The input selection strategies are one of the critical design issues that influence the packet latency and throughput of the network. When two or more input channels request an output channel simultaneously, an arbitration process is used for resolving conflicts between them. The arbiter gives priority to an input channel to get access to the output channel by using an input selection strategy [3]. This paper presents a new simple and efficient input selection strategy based on network traffic pattern for NoC systems. Also, in the proposed input selection strategy a global congestion aware method called DCA is used to give priority to an input channel according to the destination nodes. The key contributions of this paper are as follows:

1) Introducing a new complex arbiter scheme based on the network traffic. The proposed input selection strategy uses a hybrid arbitration scheme. When the traffic on the network is low the round-robin (RR) algorithm is used as a simple arbiter and when the

* Corresponding Author

traffic is high a new complex arbiter is used to manage the traffic distribution.

2) Introducing a new priority scheme based on the traffic pattern and also to provide quality of service requirements.

3) Distributing the packets along the network by using a global congestion aware method called DCA based on destination node. In fact, the proposed input selection strategy can improve the network performance by routing the packets through the non-congested paths.

The rest of this paper is organized as follows: Section II presents a review of related works and various input selection strategies. In sections III, the proposed input selection strategy is introduced. Section IV presents the proposed complex arbiter. Section V presents proposed input selection pseudo code. Section VI presents the simulation environment, traffic scenarios, evaluation metrics, experimental results, and performance evaluation of the proposed input selection strategy. Section VII presents the area overhead. Finally, Section 8 concludes this paper.

II. RELATED WORKS

The input selection is an important part of the router architecture that influences the distribution of traffic over the network. Choosing an input selection strategy can affect the average packets delay and performance of the network. The input selection is done by an arbitration process according to a fixed-priority or a variable priority strategy [4-5]. In a fixed-priority strategy, when there are multiple input port requests for the same output port, the arbiter uses a fixed-priority policy to grant access to one input port. A fixed-priority strategy is not fair to all input channels and it has a low performance under high load rate condition of the network. In variable priority strategies, the arbiter would grant access to the input port request which has the highest priority level [5]. The variable priority input selection strategies also can be classified into oblivious and congestion-aware [6]. The oblivious input selection strategies do not consider the network status. A congestion-aware input selection strategy selects an input channel based on the network congestion information. The following subsections present different types of variable priority input selection strategies.

A. Oblivious input selection strategies

The oblivious input selection strategies do not consider the network status. For example, in FCFS (First Come First Served), an input channel which requested earliest has a higher priority and can access to the output channel [5]. The random input selection function randomly chooses an input channel from candidate channels [7]. A round robin arbiter by providing an equal chance to access an output port provides a high degree of fairness among the input ports in a cyclic order [8][9]. The implementation of oblivious input selection strategies is simple, but for time-variant and non-uniform traffic pattern they cannot balance the traffic load over the network and degrades the overall network performance [10].

B. Congestion-aware input selection strategies

A congestion-aware input selection strategy selects an input channel based on the network congestion information. For example, the contention-aware input selection (CAIS) is a variable priority congestion-aware strategy that uses the CL (Contention Level) parameter to give priority to an input channel. CL parameter is the number of requests for output port from input ports of the current router that will be sent to a downstream router. The CAIS requires to compute and send CL by extra wires from the output channel to the input port of a downstream router. Actually, CAIS grants busier input channel higher priority to access the output channel [11]. The CARS input selection strategy uses a priority arbiter which depends on the congestion status of the upstream routers. In CARS, the access is given to the input channel that shows the most congested status (Cs) [12]. GLB (Global Load Balancing) uses the global congestion information as a metric in arbitration in order to reduce the network congestion [13]. The ISF is an input selection strategy for virtual-channel based NoCs. The ISF selects one input channel that has the largest free buffers in the downstream virtual channel [14]. In PBWR [15], a position-based weighted round-robin arbitration strategy is presented for providing equality of service (EoS) [16]. EoS is a subset of quality of service (QoS) to provide equal service to every flow in the network. Also, AWRR is an adaptive version of PBWR that is presented in [17]. In PDBA, a probabilistic distance-based arbitration [18] is proposed as an approximation of the age-based arbitration [19].

This paper focuses on the congestion-aware input selection strategies and it presents a new simple and efficient input selection strategy. The proposed input selection strategy uses a hybrid arbitration scheme and it uses local and non-local information based on traffic pattern to give priority to an input channel.

III. PROPOSED INPUT SELECTION STRATEGY

When network congestion is high, if input channels requests are prioritized based on network congestion conditions, this can lead to a more uniform distribution of traffic over the network. By doing this, an arbiter becomes more complex and it will increase the hardware overhead and power consumption. On the other hand, an input selection strategy should be simple and fast and not delay the arbitration. A complex arbiter, is a slow arbiter, but it is aware of the congestion, and a simple arbiter is faster. Therefore, there is a trade-off between selecting a complex arbiter and a simple arbiter. The proposed input selection strategy uses a hybrid arbitration scheme that it is a combination of a simple arbiter and a complex arbiter. The structure of the proposed input selection strategy is illustrated in Fig. 1.

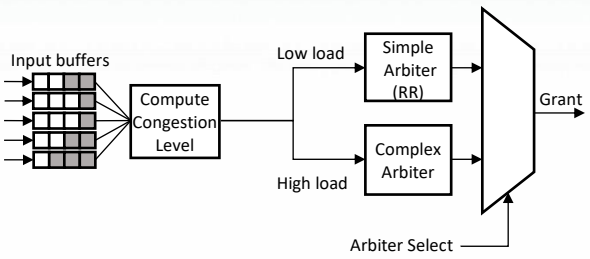


Figure 1. Proposed input selection strategy structure.

As shown in Fig. 1, the simple arbiter is used when the network load is low and when the network load is high, a complex arbiter is used to distribute the traffic more evenly. To determine the amount of network load, the number of occupied input buffer slots is sequentially checked, and if more than half of the buffer slots are empty, the network is in the low load state and otherwise in the high load state. By doing this, the arbitration will switch between a simple and a complex arbitration based on network load. In the proposed input selection strategy, a round-robin arbiter will be used in the low load and a complex proposed arbiter in the high load. In this study, the name of the proposed input selection strategy is named DTIS (Destination Traffic based Input Selection).

IV. PROPOSED COMPLEX ARBITER

In the proposed input selection strategy, the complex arbiter gives a score to each input channel. An input channel that has a higher score wins the competition. This score helps to find a path that has the lower congestion between the current and destination nodes. Each input channel score can be computed as:

$$Score = Occupy\_Slots\_Score + DCA\_Score + Priority\_Score + AGE\_Score \quad (1)$$

The complex arbiter in the proposed input selection strategy (DTIS) selects an input channel with the higher channel congestions level (*Occupy_Slots_Score*). DTIS uses a new global congestion aware scheme based on destination node called DCA method (*DCA_Score*) [20]. Also, it presents a new priority scheme based on the traffic pattern to provide quality of service requirements (*Priority_Score*). To reduce the starvation, DTIS uses an *AGE* parameter and each input channel can win the competition, according to this parameter (*AGE_Score*).

A. Occupy slots score parameter

According to the analysis which is presented in [21], the channel congestion level of an input buffer can affect the amount of delay that a packet experiences in the network. When an input buffer cannot hold the newly arrived packets because of exceeding the input buffer space, then the channel congestion will occur. The channel congestion includes two main delays: The buffer shift time (T_{Shift}) and the buffer transfer time ($T_{Transfer}$). TTransfer is a constant delay that happens when a flit transfers through an upstream router to an input buffer of a downstream router. T_{Shift} is a time duration that an incoming flit experiences during its shifts from the current position to the front position in an input buffer [22]. The congestion level of a channel ($CL_{Channel}$) can be written as follows:

$$CL_{Channel} = T_{Transfer} + T_{Shift} \quad (2)$$

If all routers of the network have a same architecture, then they have the same buffer architecture. Therefore, $T_{Transfer}$ remains unchanged for all routers and the congestion level of a channel can be written as

$$CL_{Channel} = T_{Shift} \quad (3)$$

T_{Shift} depends on the number of occupied input buffer slots (n_{Occupy}) and the router service time ($T_{RouterService}$). Therefore, the congestion level of a channel can be written as:

$$CL_{Channel} = T_{Shift} = n_{Occupy} \times T_{RouterService} \quad (4)$$

$T_{RouterService}$ is a duration time that a router performs the routing and arbitration processes. If all network's routers have a same architecture, then $T_{RouterService}$ will be a constant time when there is no contention between the input ports of a router. Therefore, the $CL_{Channel}$ can be written as:

$$CL_{Channel} = n_{Occupy} \quad (5)$$

In other words, the number of occupied input buffer slots can show the congestion level of a channel. The complex arbiter in the proposed input selection strategy uses the *Occupy_Slots_Score* parameter to give higher priority to an input channel that has more channel congestion. The *Occupy_Slots_Score* value of an input channel is the number of occupied slots in an input buffer of a router.

B. DCA score parameter

The DCA (Destination Congestion Awareness) is a method to distribute traffic more equally over the network based on the packet destination address [20]. The DCA by using only local information, without using any additional wires, tries to send flits to the destination nodes and helps to distribute traffic more evenly over the network. As Figure 2 shows, in DCA the network is divided into four regions (*East-North*, *West-North*, *West-South*, *and East-South*). Also, each region is divided to three parts (*bottom*, *middle*, and top*)*. The DCA method counts the sent flits from the current node to a destination node which is located at one of the related regions and parts. For example, if the routing algorithm selects the *East* channel as output for the current header flits and its destination is located at the *East-North* region, then the one of the *East-North.bottom* or *East-North.middle*, or *East-North.top* values for the current node will be updated according to the destination related part. Contrary, if the routing algorithm selects the *North* Channel as output for the current header flits and its destination is located at the *North-East* region, then the one of the *North-East.bottom* or *North-East.middle*, or *North-East.top* values for the current node will be updated according to the destination related part. Further details of the DCA method are presented in our previous research [20].

Fig. 3 shows the DCA method motivation which is used in our proposed input selection strategy. Suppose

that the current node is located at (3, 3) and it wants to grant access to one of the two requests from west and south input ports. West input request from node (2, 3) wants to used east output channel of the current node and goes to the destination node that is located at (7, 5). Also, south input request from node (3, 4) wants to use east output channel of the current node and goes to the destination node that is located at (6, 2).

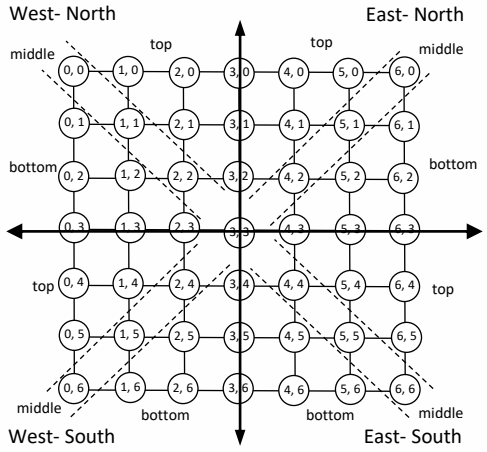


Figure 2. Network division of the DCA method.

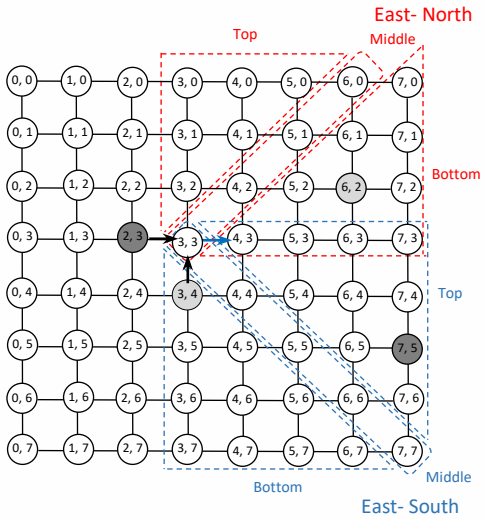


Figure 3. The motivation of the DCA method.

The motivation of the DCA method is to choose an input channel with the lower previously sent flits through the current router and from the corresponding output channel to the destination node. Hence, according to the Fig. 3, the DCA method compares the value of the *East-North.bottom* (for node (6, 2)) with the value of the *East-South.top* (for node (7, 5)). For example, if the value of the *East-South.top* is lower than the value of *East-North.bottom*, then the DCA method gives more priority to the west input channel (node (2, 3)) with the *DCA_Score*. Actually, *DCA_Score* shows the amount of previously sent flits from the current node to a destination node. In DTIS input selection strategy a request from an input channel with the lower previously sent flits to a destination has a bigger score. Further detail of the score computation of DCA method is presented in our previous research [20].

C. *Priority score parameter*

A traffic pattern can affect the network performance and it determines how data is sent and received over the network. In fact, a traffic pattern affects how the router's input buffers are occupied.

TABLE I. THE POSSIBLE VALUES OF OLD AND NEW PARAMETERS.

OLD	NEW	Occupied slots changes of an input buffer	
		Previous cycle	*Current cycle*
0	0	unchanged	Unchanged
0	1	unchanged	Increased
0	-1	unchanged	Decreased
1	0	increased	Unchanged
1	1	increased	Increased
1	-1	increased	decreased
-1	0	decreased	unchanged
-1	1	decreased	increased
-1	-1	decreased	decreased

TABLE II. PRIORITY VALUES ACCORDING TO THE OLD AND NEW PARAMETERS FOR UNIFORM AND TRANSPORT TRAFFIC PATTERNS.

OLD	*NEW*	*Priority values of input buffers*	
		Uniform traffic	*Transport traffic*
0	0	0	0
0	1	2	2
0	-1	0	1
1	0	0	0
1	1	0	0
1	-1	1	1
-1	0	0	0
-1	1	4	3
-1	-1	3	4

In the proposed input selection strategy the performance of different traffic patterns was evaluated. The results of multiple simulations for different traffic patterns showed, by giving priority to the changes in the number of occupied slots of an input buffer in the current and previous cycles and selecting an input channel according to these priorities, the average latency of the packets was improved. For this purpose, in the proposed input selection strategy, the numbers of occupied slots of an input buffer in the current and previous cycles are considered as two parameters called *NEW* and *OLD*, respectively. In the proposed strategy, we used values of 1, -1, and 0 for NEW and *OLD* parameters to indicate states of increasing, decreasing, and unchanging of an input buffer, respectively.

Table 1 shows the possible values of *OLD* and *NEW* parameters for the current and previous cycles. Also, for different traffic patterns, multiple simulations were performed and priorities were determined based on the network improvement results according to the

values of the *NEW* and *OLD* parameters for each input buffer.

Table 2 shows the priority values for uniform and transport traffic patterns according to the *OLD* and *NEW* parameters. Note that the values in Table 2 may be different for various traffic patterns. The proposed input selection strategy uses these priorities as *Priority_Score*.

For example (last row of the Table 2), giving priority to an input buffer that the number of occupied slots in the current and previous cycles was decreased (*OLD*= -1, *NEW*= -1), will reduce the average packet latency in the uniform traffic pattern. The priority values will be obtained by an offline process (multiple simulations) and they will be used at the runtime. Hence, the proposed input selection strategy is a fast input selection strategy and has an acceptable implementation overhead. The priority values of input buffers in Table 2 show the impact of *OLD* and *NEW* parameters in the packet average latency reduction (zero values mean no effect, higher values mean higher impact, and lower values mean lower impact). Also, in proposed input selection strategy, the priority values can be used to provide quality of service requirements. For example, in some applications, it is useful to divide network traffic into a number of classes and different classes of packets may have different levels of importance. By giving more priority to these classes of packets, they will take more priority over the other packets. These quality of service priorities can be added to the priority values in Table 2.

D. AGE score parameter

In input selection strategies starvation is generally a result of unfair arbitration. Actually, the starvation is a situation that an input channel cannot access to an output channel because other input channels have higher priority. To avoid starvation, the complex arbiter in the proposed input selection strategy uses a parameter called *AGE*. In other words, each input channel has an *AGE* parameter, and when it wins competition with other input channels, the value of the *AGE* will be set to 0, otherwise the value of the *AGE* parameter will be increased one unit. Actually, proposed input selection strategy with the *AGE_Score* parameter increases the score of an input channel that could not access to an output channel and give it a chance to win the competition at the next cycles.

E. Proposed complex arbiter structure

Fig. 4 shows the proposed complex arbiter structure. As shown in Fig. 4, the "*Priority Score Compute Unit*" calculates the *Priority_Scores* based on the *OLD* and *NEW* values at each instant, based on the type of the traffic and previously calculated priority table. In the "*DCA Score Compute Unit*", front header flits from each input buffer are received as inputs, and *DCA_Scores* according to the destination address information of these flits and the history data stored in the router registers will be computed for each input channel. The "*AGE Compute Unit*" updates the ages based on the failed requests and the received service request and calculates the *AGE_Scores*. Also, the number of occupied input buffer slots is used as the *Ocuppy_Slots_Score* for each input channel. Finally, using the equation (1), the sum of all scores associated with each input channel is calculated. At last, an input channel with the higher score is sent to the output as the winner of the competition (*Selected_channel*).

V. PROPOSED INPUT SELECTION PSEUDO CODE

Fig. 5 shows the pseudo code of the proposed input selection strategy. The input parameters are the set of available input channels (*Available_input_channels*), the previously computed priority of input channels based on the type of traffic pattern (*Priority_Table*), the current router address, and the last selected input channel (*last_selected_channel*). In pseudo-code, the input buffers information such as *AGE*, *OLD*, *NEW* and the free slots are internal properties and they are in the "*Buffers*" variable of each router. The output parameter is the best channel that is chosen by the proposed input selection strategy (*Selected_channel*). At first, the amount of network load is computed from the number of occupied slots of input buffers (line 2). If more than half of the buffer slots are empty (Threshold), then the network is in the low load state and a round-robin arbiter will be used (lines 4 and 5). Otherwise, the complex proposed arbiter will be used in the high load (line 6). In this case, at first, the scores array is set to zero and for each available input channel *Ocuppy_slot_score*, *Priority_score*, and *DCA_score* will be computed (lines 7 to 13). Finally, the score of each input available input channel is computed according to the sum of the *Ocuppy_slot_score*, *Priority_score*, *AGE_score*, and *DCA_score* (line 14). The best input channel, is selected by the maximum value of the scores array values and it will be sent to the output by the Selected_channel (line 16). At final, age values of all input channels will be updated (lines 18 to 22).

VI. EXPERIMENTS AND SIMULATION ENVIRONMENT

For evaluating the DTIS input selection strategy, we modified an open source SystemC based simulator named Noxim [23]. The environment of the simulation uses a 10×10 mesh network topology. The wormhole switching [24] and the odd-even algorithm [25] are used as switching and routing strategies, respectively. In our simulation each input channel has a four flits length FIFO buffer and each packet consists of eight flits. We run the simulation for 200,000 cycles and the first 20,000 cycles is considered as warm up duration. For increasing the accuracy of the evaluations, we repeated all simulations ten times and then used the averaged results.

A. Traffic scenarios

We used both synthetic and real traffic pattern scenarios to evaluate the DTIS input selection strategy. Under random traffic scenario, the pattern of the traffic is uniform and a node sends data to the other nodes randomly with the same probability. In the transpose traffic pattern, if i and j are the number of the column and row of a node that is located at (j, j), this node only sends its flits to a node that is located at (n-1-j, m-1-i) on the network, where n and m are the number of columns and rows in a mesh network, respectively.

Under the hotspot traffic pattern, we send 10% more traffic than regular random pattern to a node that is located at (5, 5). For evaluating the real traffic scenario, we used the MPEG4 communication system traffic pattern [24].

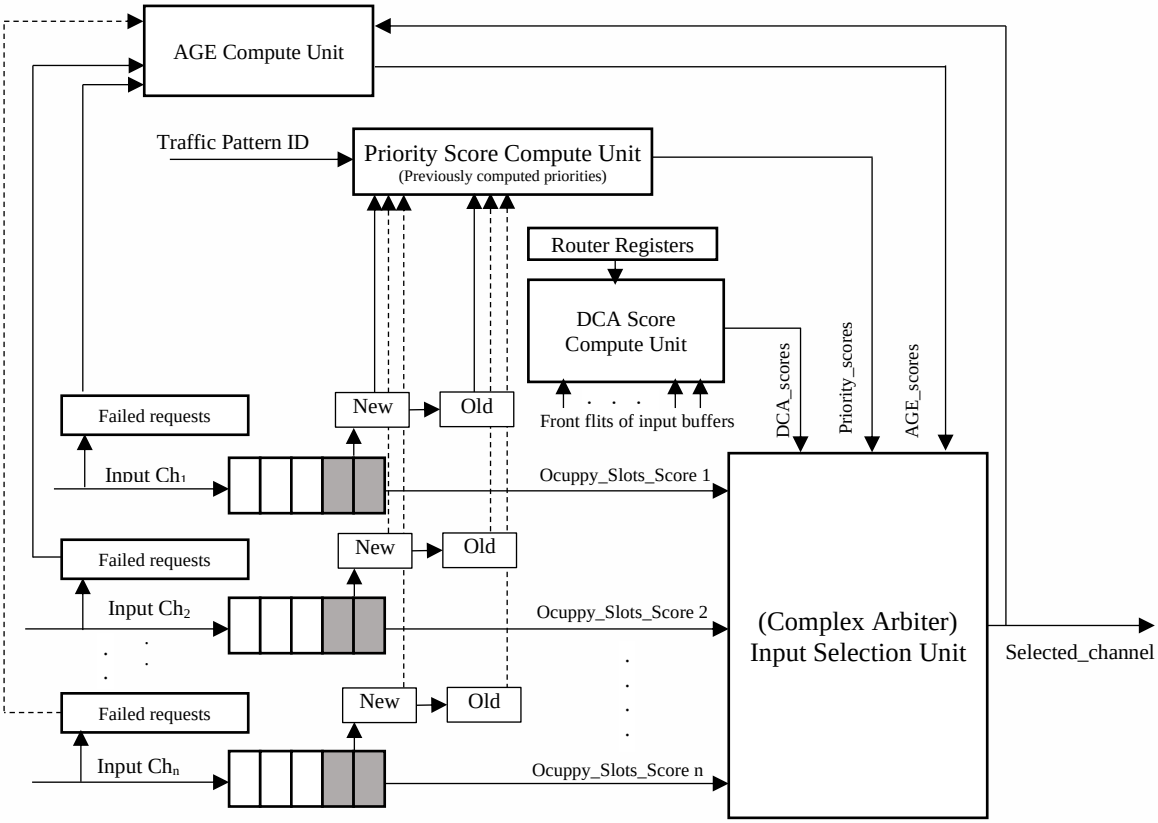


Figure 4. Proposed complex arbiter structure.

```
01: DTIS_input_selection (in  :  Available_input_channels, Priority_Table, current,
                                 last_selected_channel;   Out:  Selected_channel) {
02: Load_Level = get_buffer_ocuppy_level (Buffers);
03: Threshold = (buffer_depth * (input_channel_numbers))/2;      // 50 Percent of the buffer slots are occupied.
04: if (Load_Level <= Threshold) {   // use round robin as simple arbiter
05:    Selected_channel = Round_Robin (Available_input_channels, last_selected_channel);
06 :} else {   // use proposed algorithm as complex arbiter
07:   Scores [Available_input_channels] =0;
08:   for each Ci ϵ Available_input_channels {
09:       Ocuppy_slot_score= Max_buffer_length – Buffers [Ci].Free_slots;
10:       Priority_score = Priority_Table [Buffers [Ci].old] [Buffers [Ci].new];
11:       AGE_score = Buffers [Ci].AGE;
12:       destination = getDestination (Buffers [Ci].getFrontFlit ());
13:       DCA_score = DCA_get_score (Ci, current, destination);
14:       Scores [Ci] = Ocuppy_slot_score + Priority_score + AGE_score + DCA_score;
15:    }
16:   Selected_channel = Max (Scores [Available_input_channels]);
17 :} // updates AGES
18:   for each Ci ϵ Available_input_channels {
19:      if (Ci== Selected_channel) {
20:         Buffers [Ci]. AGE = 0;
21:       } else Buffers [Ci]. AGE ++;
22:    }
23 :}
```

Figure 5. Proposed input selection pseudo code.

B. Evaluation metrics

For evaluating the performance of the proposed input selection strategy, we used the network throughput and average packet latency metrics [26]. The network throughput is defined as the maximum number of the delivered packets within a specific period over the network, and it can be defined as follow:

$$Throughput = \frac{Total\ received\ flits}{Number\ of\ network\ nodes \times Total\ cycles} \quad (6)$$

Where *Total received flits* shows the total numbers of delivered flits to the destination node and *Total cycles* refers to the number of the passed clock cycles between the first injected message and the last delivered message. The average packet latency can be defined as the average of delivered packets latency and the latency of a packet can be defined as the duration time between a header flit injection time of a packet into the network and the time that a tail flit is delivered at the destination. The average packet latency can be defined as follows:

$$L = \frac{1}{K}\sum_{i=0}^{k} L_i \quad (7)$$

Where L_i is the latency of the message i and K is the total number of delivered messages at the destination.

C. Experiment results

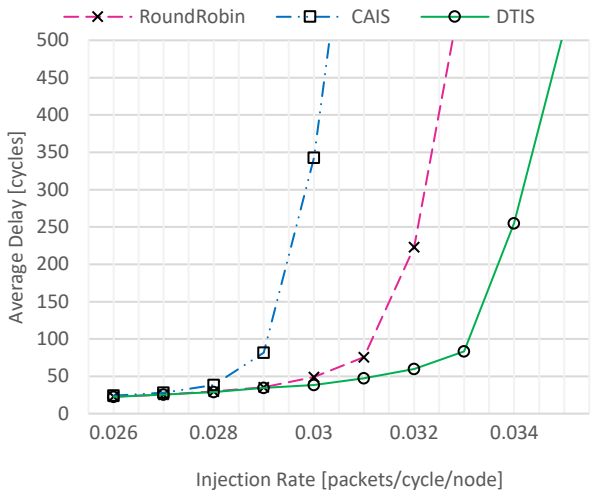
This section of the paper shows the improvement of the average packet latency, throughput, and energy consumption of the proposed input selection strategy for each traffic pattern with different packet injection rate.

Fig. 6a shows results of the average delay for transpose traffic pattern. As shown in Fig. 6a, the average delay improvements of DTIS to the RoundRobin and CAIS are 94% and 37%, respectively.

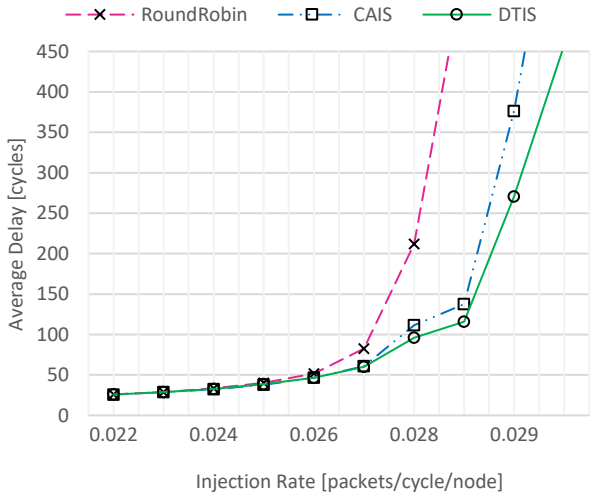
Fig. 6b shows results of the average delay for random traffic pattern. As shown in Fig. 6b, the average delay improvements of DTIS to the RoundRobin and CAIS are 54% and 13%, respectively. Fig. 6c shows results of the average delay under the hotspot traffic pattern.

As shown in Fig. 6c, the average delay improvements of DTIS to the RoundRobin and CAIS are 84% and 27%, respectively. DTIS performs better than RoundRobin and CAIS for transport, random, and hotspot traffic scenarios. This is because DTIS uses the new congestion aware scheme called DCA and can distribute traffic more evenly over the network.

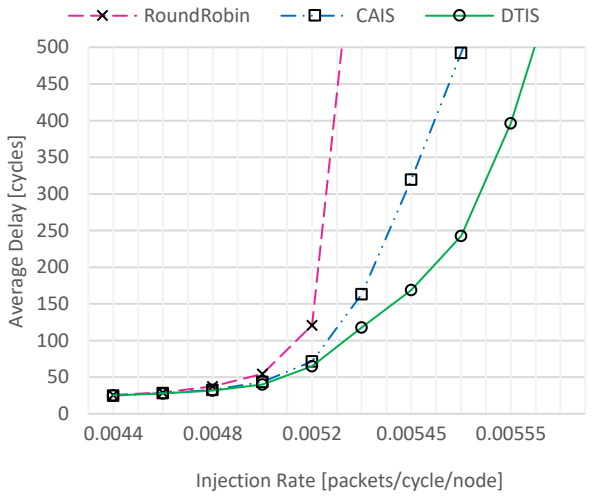
Table 3 shows the latency improvements results of DTIS for random, transpose, and hotspot traffic scenarios at a packet injection rate of non-saturated traffic. As can be seen, DTIS input selection strategy has an improvement, on the average, ranging from 26.24% to 77.83% compared to the other input selection strategies. Table 4 shows the saturation throughput improvements of DTIS in detail. Simulation results show that DTIS has a higher saturation throughput than other strategies, with an improvement of 12.18%–14.92%.



(a)



(b)



(c)

Figure 6. Average packet latency results for different traffic patterns: (a)Transpose (b) Random (c) Hotspot.

Fig. 7 shows results of the throughput for all traffic scenarios. The simulation results show that improvement on the average delay can improve the network throughput. As observed from the results, the DTIS strategy leads to the lowest average delay for all traffic scenarios, because the proposed input selection strategy has a more accurate knowledge about the network congestion status by using the local and non-local traffic information. So, it can distribute traffic more efficiently than other strategies.

The improvement in the network throughput can improve the total energy consumption. We used Noxim [23] simulator to evaluate the energy consumption of the DTIS input selection strategy. Noxim is an architecture-level simulation tool based on SystemC, a system description language based on C++. Noxim can evaluate the energy consumption of the major operations of a router including input selection, routing, and forwarding the flits. In Noxim, the energy of a given element "e" at each cycle "c" is defined as:

$$E(e, c) = \alpha(e, c) \times P_{avg}(e) \times T_{CK} \quad (8)$$

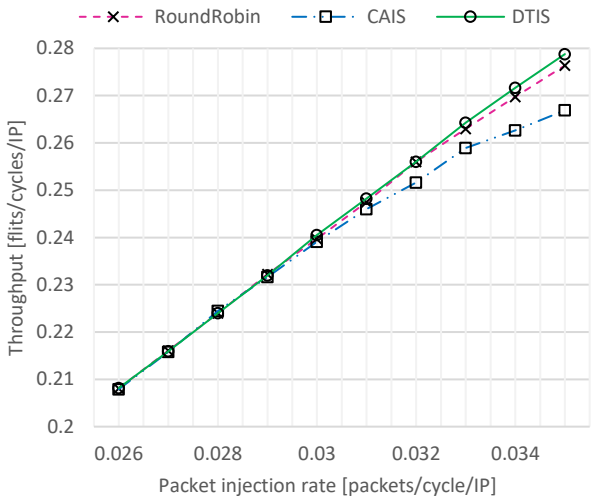
Where α (e, c) is the activity function and if e not active in cycle c is 0, otherwise is 1. P_{avg} (e) is the average dynamic power and in Noxim it has been estimated for each component.

We implemented DTIS input selection strategy in VHDL and synthesized using the Synopsys Design Compiler and added the average power of DTIS to the Noxim. In evaluating the energy consumption, the overhead of DTIS logic and wiring is taken into account because Noxim is a signal level simulator and in it each wire is defined as a signal.

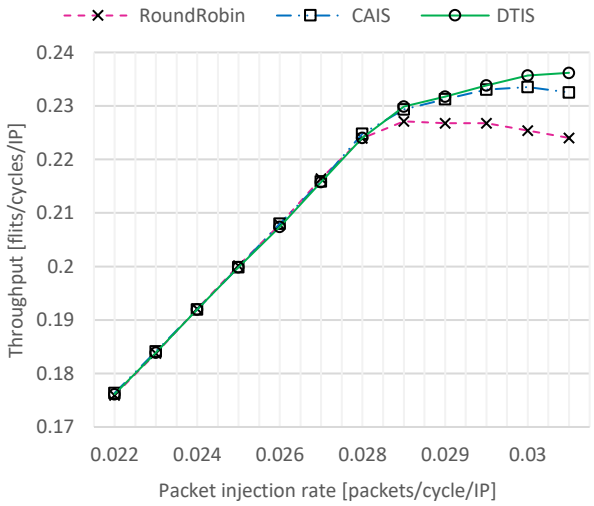
According to Fig. 8, if DTIS input selection strategy is used, the total energy consumption of the network will decrease for all traffic patterns. For example, DTIS has a lower energy consumption with an average improvement of 6% for transpose traffic pattern than other strategies before saturation point. This improvement in energy consumption is the results of avoiding from congested routes and accurate traffic information that is provided by DCA.

For evaluating the DTIS input selection strategy under a real traffic scenario, we used an MPEG4 communication system traffic pattern [24].

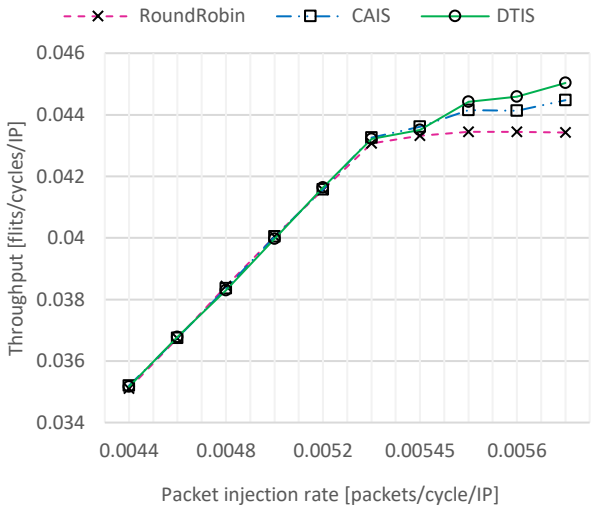
Fig. 9a shows the MPEG4 block diagram with communication bandwidth in MBps that is mapped onto 3×4 mesh topology (Fig. 9b). As it can be seen in Fig. 9c, the DTIS input selection strategy again outperforms the other strategies in a real traffic scenario and the average delay improvements of DTIS to the RoundRobin and CAIS are 42% and 12%, respectively.



(a)

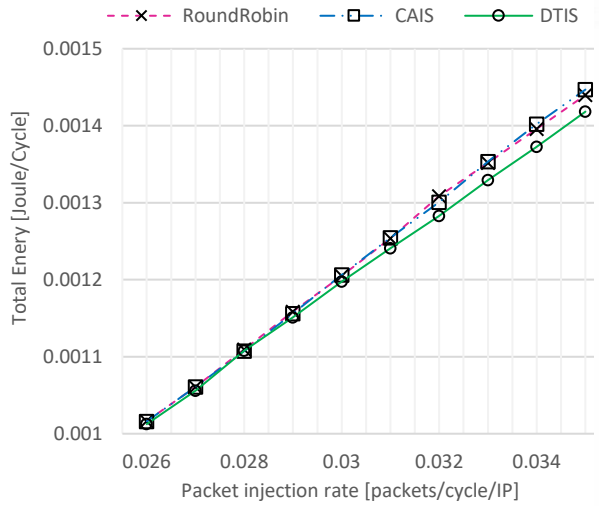


(b)

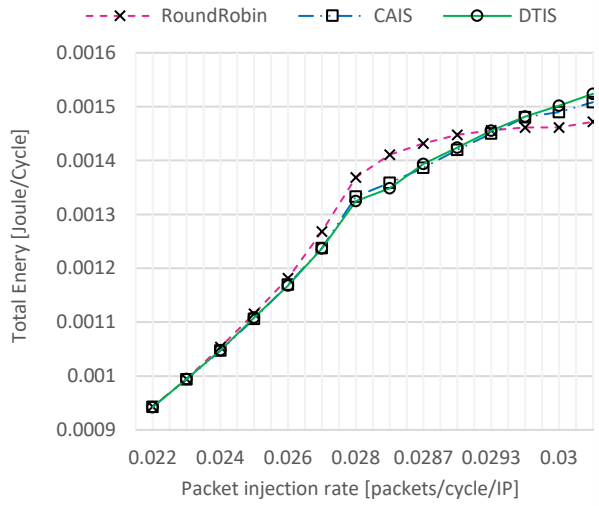


(c)

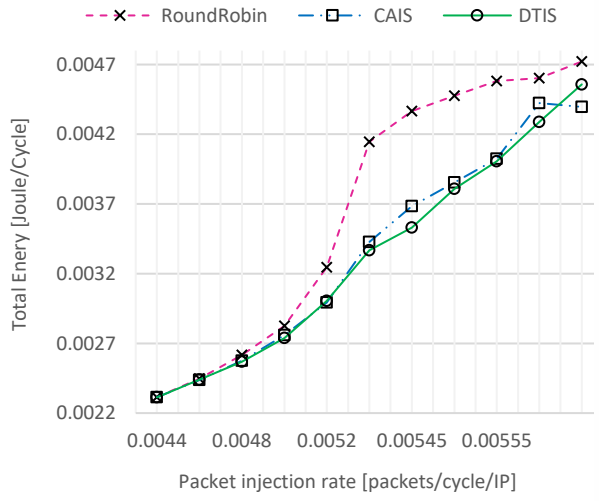
Figure 7. Throughput results for different traffic patterns: (a) Transpose (b) Random (c) Hotspot.



(a)

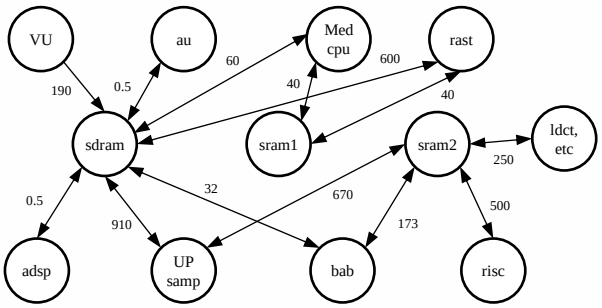


(b)

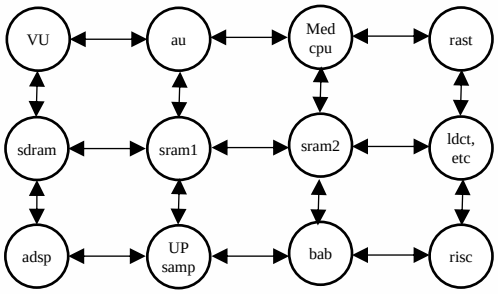


(c)

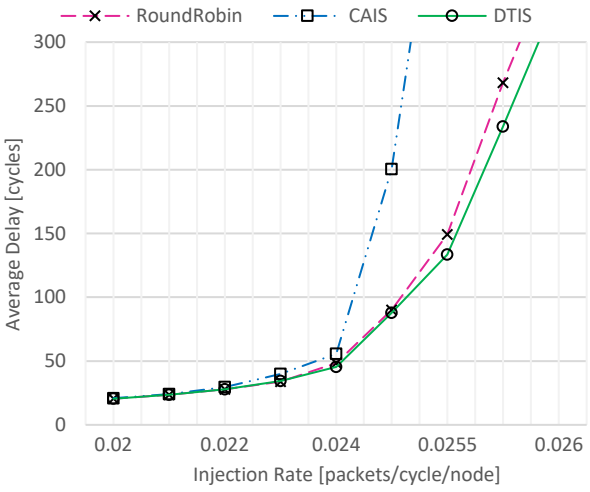
Figure 8. Total network energy consumption results for different traffic patterns: (a)Transpose (b) Random (c) Hotspot.



(a)



(b)



(c)

Figure 9. MPEG4 decoder block diagram, with communication BW annotated (in MB/s) and it's mapping onto mesh topology [24], and experiment results:

(a) MPEG4 decoder block diagram,
(b) Mapping onto mesh topology, and
(c) Average packet latency for MPEG4 traffic pattern

TABLE III. LATENCY AND DTIS IMPROVEMENT.

Traffic patterns	Packet injection rate (packet/cycle/node)	Average latency (cycles)			Latency reduction by DTIS	
		RoundRobin	*CAIS*	*DTIS*	*vs. RoundRobin*	*vs. CAIS*
Random	0.028	212.196	111.442	96.037	54.74%	13.82%
Transpose	0.031	850.072	75.450	47.385	94.43%	37.20%
hotspot	0.0054	752.560	163.185	117.966	84.32%	27.71%
				Average latency reduction	77.83%	26.24%

TABLE IV. SATURATION THROUGHPUT AND DTIS IMPROVEMENT.

Traffic patterns	Saturation throughput (packet/ns/node)			DTIS improvement	
	RoundRobin	*CAIS*	*DTIS*	*vs. RoundRobin*	*vs. CAIS*
Random	0.2041	0.2201	0.2462	20.63%	11.86%
Transpose	0.2563	0.2489	0.2913	13.66%	17.03%
hotspot	0.0420	0.0431	0.0464	10.48%	7.66%
			Average improvement	14.92%	12.18%

VII. AREA OVERHEAD

The implementation of a router in the NoC systems as compared to the IP cores, because of resource constraints, should not consume a large area. In this paper, to evaluate the area overhead of the DTIS input selection strategy, we designed three routers based on the DTIS, CAIS, and RoundRobin input selection strategies in VHDL. Then we used the Synopsys Design Compiler and a SAED 90 nm EDK technology to synthesize them to provide the area breakdown of the different elements of the router overhead (μm^2). We used 64-bit flits and 4-entries FIFO buffers in all implementations. As can be observed from the Fig. 10, the major contribution of silicon area is due to the FIFO buffers. As it can be seen in Fig. 10, the RoundRobin router needs the lowest area as compared to the CAIS and DTIS input selection strategies. Also, the DTIS has a little more area overhead because of needing additional logic circuits. The area overhead of DTIS is a linear function of port numbers and due to its good performance is negligible.

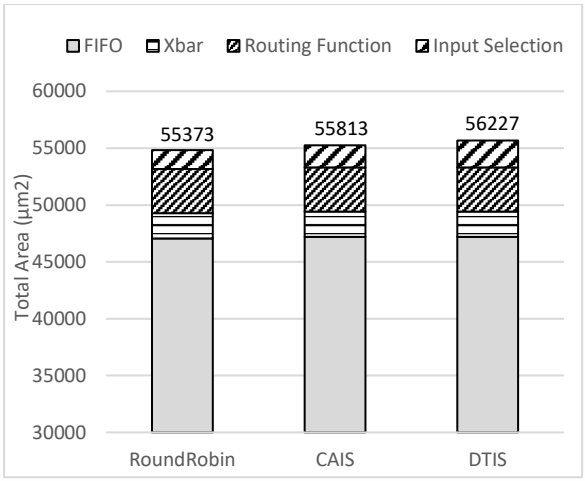


Figure 10. Total area of routers for different input selection strategies (µm2).

VIII. CONCLUSIONS

In this research, a new input selection strategy called DTIS was proposed that it uses different parameters as input selection metrics to improve the performance of the NoC. The proposed input selection strategy can detect the network congestion more accurately because it uses both offline and online congestion information based on the traffic pattern, and the congestion on the path to the destination. The simulation results depicted an improvement in the average delay, network throughput, and energy consumption for both real and synthetic traffic scenarios. Also, the proposed input selection strategy can be applied to each network size because it is general in nature.

REFERENCES

[1] L. Benini, and G. De Micheli, "Powering networks on chips: energy-efficient and reliable interconnect design for socs", Proceedings of the 14th International Symposium on Systems Synthesis, ACM, 2001, pp.33–38.

[2] W.J. Dally, and B.P. Towles, "Principles and Practices of Interconnection Networks", Elsevier Press, USA, 2004.

[3] A. Jantsch, H. Tenhunen, "Networks on Chip", Springer Press, Boston, USA, 2003, Vol. 396.

[4] J. Duato, S. Yalamanchili, and L.M. Ni, "Interconnection Networks: An Engineering Approach", Morgan Kaufmann Press, USA, 2003.

[5] R. Marculescu, U.Y. Ogras, L. Peh, N.E. Jerger, and Y. Hoskote, "Outstanding Research Problems in NoC Design: System, Microarchitecture, and Circuit Perspectives", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Jan. 2009, Vol. 28, No. 1, pp.3-21.

[6] A. Mehranzadeh, A. Khademzadeh, and A. Mehran, "FADyAD- Fault and congestion aware routing algorithm based on DyAD algorithm", 5th International Symposium on Telecommunications, Tehran, 2010, pp.274-279.

[7] U.Y. Ogras, P. Bogdan, and R. Marculescu, "An Analytical Approach for Network-on-Chip Performance Analysis", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Dec. 2010, Vol. 29, No. 12, pp.2001-2013.

[8] L. Benini, and G. De Micheli, "Networks on chip: a new paradigm for systems on chip design", Proceedings 2002 Design, Automation and Test in Europe Conference and Exhibition, Paris, France, 2002, pp.418-419.

[9] E. Chang, H. Hsin, S. Lin, and A. Wu, "Path-Congestion-Aware Adaptive Routing With a Contention Prediction Scheme for Network-on-Chip Systems", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Jan. 2014, Vol. 33, No. 1, pp.113-126.

[10] J.C. Martinez, F. Silla, P. Lopez, and J. Duato, "On the influence of the selection function on the performance of networks of workstations", Proceedings of the International

Symposium on High-Performance Computing, Springer Berlin Heidelberg, 2000, pp.292–299.

[11] D. Wu, B.M. Al-Hashimi, and M.T. Schmitz, "Improving routing efficiency for network-on-chip through contention-aware input selection", Asia and South Pacific Conference on Design Automation, Japan, Yokohama, 2006, pp.6.

[12] M. Daneshtalab, A. Pedram, M.H. Neishaburi, M. Riazati, A. Afzali-Kusha, and S. Mohammadi, "Distributing Congestions in NoCs through a Dynamic Routing Algorithm based on Input and Output Selections", 20th International Conference on VLSI Design held jointly with 6th International Conference on Embedded Systems (VLSID'07), Bangalore, 2007, pp.546-550.

[13] M. Daneshtalab, M. Ebrahimi, P. Liljeberg, J. Plosila, H. Tenhunen, "A systematic reordering mechanism for on-chip networks using efficient congestion-aware method", Journal of Systems Architecture, 2013, Vol. 59, Issues 4–5, pp.213-222.

[14] W. Xinyu, Y. Zhigang, and X. Huazhen, "Improving Routing Efficiency for Networks-on-Chip through an Efficient Input Selection Strategy", Future Control and Automation book, Springer Berlin Heidelberg, 2012, pp.445-452.

[15] H. Park, and K. Choi, "Position-based weighted round-robin arbitration for equality of service in many-core network-on-chips", Proceedings of the Fifth International Workshop on Network on Chip Architectures (NoCArc '12). ACM, New York, NY, USA, 2012, pp.51-56.

[16] M.M. Lee, J. Kim, D. Abts, M. Marty, and J.W. Lee, "Probabilistic distance-based arbitration: providing equality of service for many-core CMPs", 43rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO-43), Georgia, 2010.

[17] H. Park, and K. Choi, "Adaptively weighted round-robin arbitration for equality of service in a many-core network-on-chip", IET Computers & Digital Techniques, 2016, Vol. 10, Issue 1 , pp.37-44.

[18] J.W. Lee, M.C. Ng, and K. Asanovic, "Globally-Synchronized Frames for Guaranteed Quality-of-Service in On-Chip Networks", International Symposium on Computer Architecture, Beijing, 2008, pp.89-100.

[19] G. Kim, M.M. Lee, J. Kim, J.W. Lee, D. Abts, and M. Marty, "Low-Overhead Network-on-Chip Support for Location-Oblivious Task Placement", IEEE Transactions on Computers, June 2014, Vol. 63, No. 6, pp.1487-1500.

[20] A. Mehranzadeh, A. Khademzadeh, N. Bagherzadeh, and M. Reshadi, "DICA: destination intensity and congestion-aware output selection strategy for network-on-chip systems", IET Computers & Digital Techniques, 2019, Vol. 13, No. 4, p.335-347.

[21] S. Foroutan, Y. Thonnart, and F. Petrot, "An iterative computational technique for performance evaluation of networks-on-chip", IEEE Transactions on Computers, 2013, Vol. 62, No. 8, pp.1641–1655.

[22] W.J. Dally, and C.L. Seitz, "The torus routing chip", Journal of Distributed Computing, 1986, Vol. 1, No. 4, pp.187–196.

[23] V. Catania, A. Mineo, S. Monteleone, M. Palesi, and D. Patti, "Cycle-Accurate Network on Chip Simulation with Noxim", ACM Transactions on Modeling and Computer Simulation, 2016, Vol. 27, Issue 1, Article 4, 25 pages.

[24] E.B.Van der Tol, and E.G.T. Jaspers, "Mapping of MPEG-4 decoding on a flexible architecture platform", Proc. SPIE, San Jose, CA, USA, January 2002, pp.1–13.

[25] G.M. Chiu, "The odd-even turn model for adaptive routing", IEEE Transactions on Parallel Distributed System, 2000, Vol. 11, No. 7, pp.729–738.

[26] M. Dehyadegari, M. Daneshtalab, and M. Ebrahimi, "An adaptive fuzzy logic-based routing algorithm for networks-on-chip", NASA/ESA Conference on Adaptive Hardware and Systems (AHS), San Diego, CA, 2011, pp.208–214.

Amin Mehranzadeh received the B.Sc. and M.Sc. degrees in Computer Engineering. He is currently pursuing the Ph.D. degree at Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran. His current research interests include Embedded Systems and VLSI Hardware Design, and Power and Performance Optimization in Network-on-Chip Architectures.

Ahmad Khademzadeh was born in Mashhad, Iran, in 1943. He received the B.Sc. degree in Applied Physics from Ferdowsi University, Mashhad, Iran, in 1969 and the M.Sc. and Ph.D. degrees respectively in Digital Communication and Information Theory and Error Control Coding from the University of Kent, Canterbury, UK. He is currently a Full professor in ICT Research Institute (ITRC). He is a member of the Iranian Electrical Engineering Conference Permanent Committee. Dr. Khademzadeh has received four distinguished national and international awards including Kharazmi International Award, and has been selected as the national outstanding researcher of the Iran Ministry of Information and Communication Technology. His research interests include VLSI Design, Interconnection Network, Fault Tolerant and Computer Architectures.

Midia Reshadi is currently an Assistant Professor in Computer Engineering Department at Science and Research Branch of Azad University since 2010. His research interest is Network-on-Chip including Performance and Cost Improvement in Topology, Routing and Application-Mapping Design levels of various types of NoCs such as 3D, Photonic and Wireless. Recently, he has started carrying out research in NoC Based Deep Neural Network Accelerators and Silicon Interposer Based NoC with his team which is consist of M.Sc. and Ph.D. students.