Scientific Research and Essays Vol. 6(15), pp. 3252-3260, 11 August, 2011
Available online at http://www.academicjournals.org/SRE
DOI: 10.5897/SRE11.482
ISSN 1992-2248 © 2011 Academic Journals

Full Length Research Paper

A fuzzy-based performance-enhancing input selection technique for network-on-chip

Ebrahim Behrouzian Nejad[1]*, Ahmad Khademzadeh[2], Kambiz Badie[2] and Amir Masoud Rahmani[1]

[1]Computer Engineering Department, Islamic Azad University, Science and Research Branch, Tehran, Iran.
[2]Iran Telecommunication Research Center, Tehran, Iran.

Accepted 3 June, 2011

The performance of network-on-chip (NOC) largely depends on the underlying routing techniques. A routing technique has two constituencies: output selection and input selection. This paper focuses on the improvement of input selection part. Two traditional input selections have been used in NOC, first-come-first-served (FCFS) input selection and Round-Robin input selection. Also, recently a contention-aware input selection (CAIS) technique has been presented for NOC, but there are some problems and defections in this technique. In this paper we eliminate the problems and defections of CAIS technique to develop a simple yet effective input selection technique named Fuzzy-based contention-aware input selection (FCAIS). When there are contentions of multiple input channels competing for the same output channel, FCAIS decides which input channel obtains the access depending on the two parameters: contention level of the upstream switches and AGE of the all input channels. In this scheme each switch selects one of the input channels with highest priority, which is calculated by a fuzzy controller. The simulation results with different traffic patterns show that FCAIS can achieves better performance than the FCFS and CAIS input selections, when combined with either deterministic or adaptive output selection.

Key words: Network-on-chip, routing algorithm, input selection, fuzzy control system.

INTRODUCTION

In the past decades, system-on-board (SOB) has been the dominant methodology for designing complex digital systems. As the complexity of applications and their required algorithms have grown so rapidly, SOB has been replaced by system-on-chip (SOC) methodology (Mehran et al., 2007). SOC consist of a number of pre-designed intellectual property (IP) assembled together using electrical bus to form large chips with very complex functionality. But future generations of SOC will consist of hundreds of pre-designed IPs assembled together to form large chips with very complex functionality. As technology scales and chip integrity grows, on-chip communication is playing an increasingly dominant role in SOC design. To deal with the increasingly difficult problem of on-chip communication, it has been recently proposed to connect the IPs using a Network-On-Chip architecture.

NOC has been proposed as a solution to provide better modularity, scalability, reliability and higher bandwidth compared to bus-based communication infrastructures (Dally and Towles, 2001; Benini and De-Micheli, 2002; Ivanov and De-Micheli, 2005). In NOC each core is connected to a switch by a network interface. Cores communicate with each other by sending packets via a path consisting of a series of switches and inter-switch links. Figure 1 shows an abstract view of a NOC. As shown in Figure 1, a typical NOC consists of four major components: cores (C), network interface (NI) units, switches (S) and physical links. Each core can be a

*Corresponding author. E-mail: behrouzian.e@srbiau.ac.ir or behrouzian.e@gmail.com.

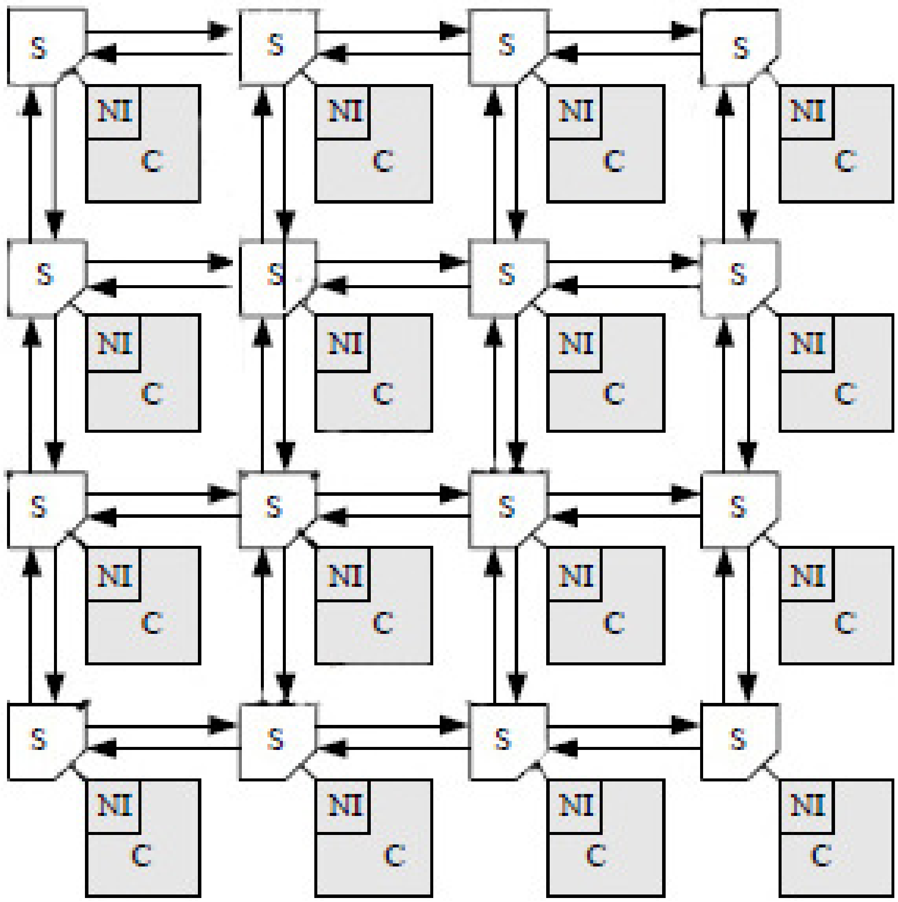

Figure 1. Abstract view of a 4*4 2D mesh-based NOC.

processing element (PE), embedded memory, DSP or etc. Other components constitute the communication fabric.

In a NOC, switches are responsible for routing the packets between nodes. Each switch has a set of bidirectional ports through which it is connected to neighboring switches or PEs. It also contains a routing logic to define a path between input and output ports, buffers to store intermediate data and an arbiter to grant access to a given port when multiple input requests arrive in parallel.

The problem of defining communication protocols for these NOCs is not an easy matter since the resources used in traditional networks are not available on-chip. The performance of NOC largely depends on the underlying routing technique, which chooses a path for a packet and decides the routing behavior of the switches. Routing algorithms can be generally classified into two types: deterministic and adaptive. In deterministic routing, the path is completely determined by the source and the destination address. On the other hand, a routing technique is called adaptive if, given a source and a destination address; the path taken by a particular packet depends on dynamic network conditions (example, congested links due to traffic variability). A routing technique has two constituencies: output selection and input selection. A packet coming from an input channel may have a choice of multiple output channels. The output selection chooses one of the multiple output channels to deliver the packet. Similarly, multiple input channels may request simultaneously the access of the same output channel; the input selection chooses one of the multiple input channels to get the access (Nejad and Khademzadeh, 2009; Wu et al., 2006). Most researches on routing techniques for NOC have focused on the improvement of output selection. Two input selection methods have been used in NOC, FCFS input selection and Round-Robin input selection. In Wu et al. (2006), this paper investigates the impact of input selection, and presents a contention-aware input selection (CAIS) technique for NOC that improves the routing efficiency, but there are some problems and defections in this technique. In this paper we eliminate the problems and defections of CAIS technique to develop a simple yet effective input selection technique named FCAIS. When there are contentions of multiple input channels competing for the same output channel, FCAIS decides which input channel obtains the access depending on the two parameters: contention level of the upstream switches and AGE of the all input channels. In this scheme each switch selects one of the input channels with highest priority, which is calculated by a fuzzy controller.

Related works

The performance of NOC is highly dependent on throughput and latency properties of the on-chip routers. Routing strategies have a key role on communication and performance in on-chip interconnection networks and several efforts have been done attempting to improve the performance of them in on-chip interconnection networks. In Glass and Ni (1992), a partially adaptive routing algorithm, called turn model which is based on prohibiting certain turns during routing packets to prevent deadlock is presented. In Chiu (2000) a routing algorithm called odd-even (OE) was proposed based on turn model. It restricts some locations where turn can be taken so that deadlock can be avoided. In comparison with previous methods, the OE is more adaptive than other routing algorithms which are based on turn model. A routing scheme called DyAD was proposed in Hu and Marculescu (2004). This algorithm is the combination of a deterministic routing algorithm and an adaptive routing algorithm. The router can switch between these two routing modes based on the network's congestion. Another adaptive routing named DyXY along with an analytical model based on queuing theory for a 2D mesh has been proposed in Li et al. (2006). The authors claim that DyXY ensured deadlock-free and livelock-free routing and it can achieve better performance compared with static XY routing and OE routing. In Pirretti et al. (2004) and Dumitras et al. (2003) some fully fault tolerant routing algorithms are explained, one of them is named directed flooding algorithm. In this algorithm a message is sent to each outgoing link with probability p which is not fixed but varies based on the destination of the packet. In Asad et al. (2009) a source routing algorithm called Predominant Routing was proposed which exploits the advantages of both deterministic and adaptive routing algorithms. Also, in Salehi and Dana (2010) a routing algorithm for avoiding congested areas using a fuzzy-based routing decision is proposed.

All of these routing techniques focused on the output selection. Two input selections have been used in NOC, FCFS input selection and Round-Robin input selection. In FCFS, the priority of accessing the output channel is granted to the input channel which requested the earliest. Round-robin assigns priority to each input channel in equal portions on a rotating basis. FCFS and Round-robin are fair to all channels, but do not consider the actual traffic condition. In Wu et al. (2006), this paper investigates the impact of input selection and presents a new contention-aware input selection (CAIS) technique for NOC that improves the routing efficiency. When there are contentions of multiple input channels competing for the same output channel, CAIS decides which input channel obtains the access only depending on the contention level of the upstream switches, which in turn removes possible network congestion. But there is a starvation possibility in this technique. The motivation of this paper is to eliminate the problems and defections of CAIS technique to develop a simple yet effective input selection technique named FCAIS. In this scheme, FCAIS, each switch selects one of the input channels with highest priority, which is calculated by a fuzzy controller.

Fuzzy logic control system

The fuzzy logic was introduced by Lotfi Zadeh as a generalization of the Boolean logic (Zadeh, 1965). The difference between these logics is that fuzzy set theory provides a form to represent uncertainties; that is, it accepts conditions partially true or partially false. Fuzzy logic is a good logic to treat random uncertainty, that is, when the prediction of a sequence of events is not possible. A fuzzy control system (Driankov et al., 1993) is a rule based system which a set of so-called fuzzy rules represents a control decision mechanism to adjust the effects of certain causes coming from the system. The aim of the fuzzy control system is normally to substitute for or replace a skilled human operator with a fuzzy rule based system. Specifically, based on the current state of a network an inference engine equipped with a fuzzy rule base determines an online decision to adjust the system behavior in order to guarantee that it is optimal in some certain senses.

The fuzzy logic controller which is used in this paper, is shown in Figure 2. The output of the fuzzy logic controller in Figure 2 is used to tune the controlled system's parameters based on the state of the system.

The design process of a fuzzy control system consists of a series steps. The first step in fuzzy control is to define the input variables and the control variables. Each variable must be quantified. Then quantifications of the variables are assigned a membership function. Then a fuzzy rule base must be designed. This rule base determines what control action take place under what input conditions. The rules are written in an if-then format. An implication formula is used to evaluate the individual if-then rules in the rule base (Salehi and Dana, 2010; Driankov et al., 1993). A composition rule is used to aggregate the rule results to yield a fuzzy output set. In the proposed fuzzy system, Mamdani minimum inference method (Ying, 2000) was used as the fuzzy inference method. A defuzzification method is then applied to the fuzzy control action to produce a crisp control action. We use center of gravity as defuzzification method.

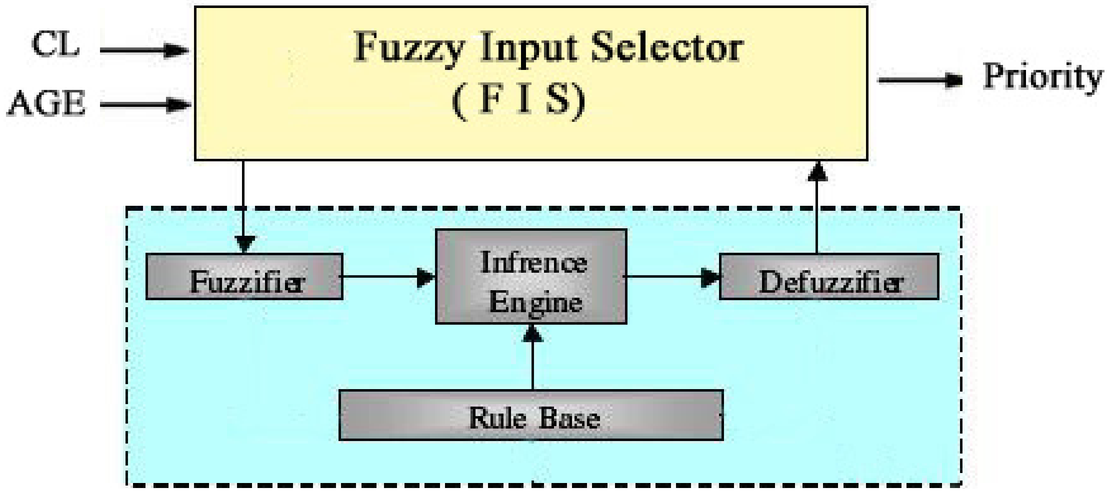

Figure 2. Fuzzy control system for input selection.

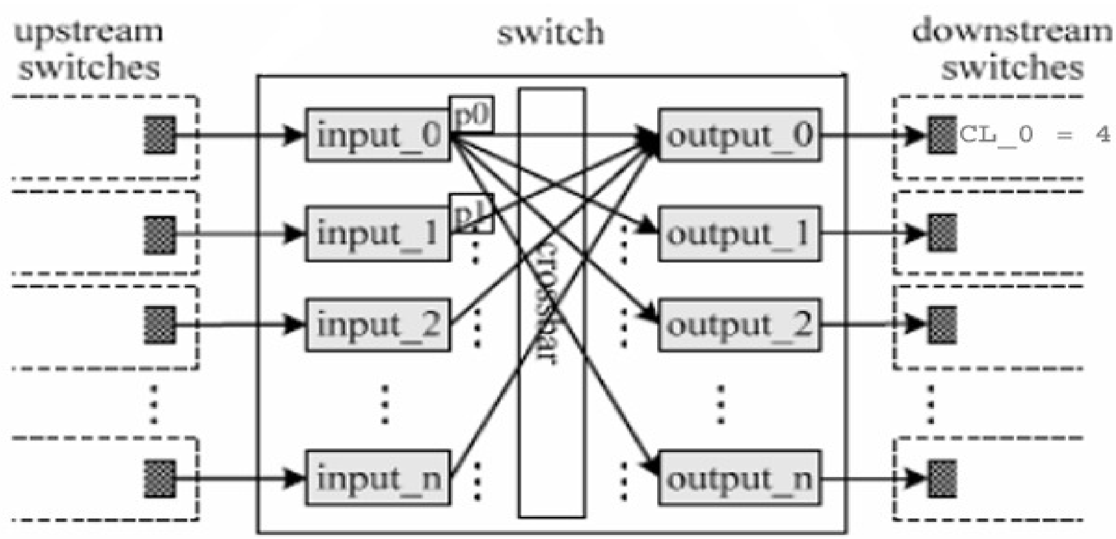

Figure 3. Concept of contention level (CL) in a switch.

MATERIALS AND METHODS

Proposed input selection technique

Each routing algorithm has two constituencies: output selection and input Selection. In this study, we will present a novel input selection technique for NOC. The proposed input selection technique can be combined with an output selection, either deterministic or adaptive, to complete the routing function. In this paper, the XY routing (Glass and Ni, 1992) is used as a representative of deterministic output selection for its simplicity and popularity in NOC. To avoid deadlock, the minimal odd-even (OE) routing (Chiu, 2002) is used as a representative of adaptive output selection.

Multiple input channels may request simultaneously the access of the same output channel, example, packets p0 of input_0 and p1 of input_1 can request output_0 at the same time. The input selection chooses one of the multiple input channels to get the access. Two input selections have been used in NOC, FCFS input selection and Round-Robin input selection. In FCFS, the priority of accessing the output channel is granted to the input channel which requested the earliest. Round-Robin assigns priority to each input channel in equal portions on a rotating basis. FCFS and Round-Robin are fair to all channels but do not consider the actual traffic condition. This section presents an input selection that performs more intelligent, by considering the actual traffic condition, leading to higher routing efficiency. In this paper we consider NOCs with 2D mesh topology. Wormhole switching is employed because of its low latency and low buffer requirement. Similar to Wu et al. (2006), the basic idea is to give the input channels different priorities of accessing the output channels. The priorities are decided dynamically at run-time, based on the actual traffic conditions of the upstream switches. More precisely, each output channel within a switch observes the contention level (CL) (the number of requests from the input channels) and sends this contention level to the input channel of the downstream switch, where the contention level is then used in the input selection (Figure 3). When multiple input

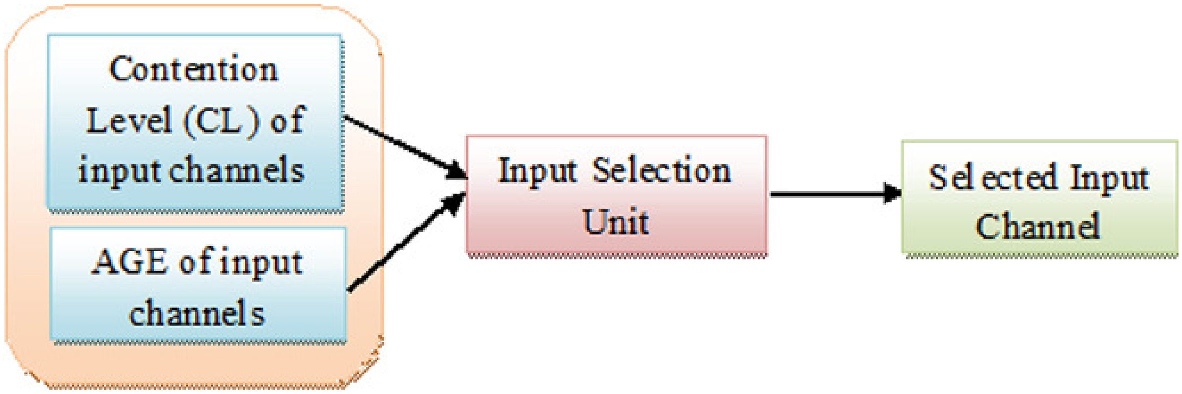

Figure 4. Input selection parameters.

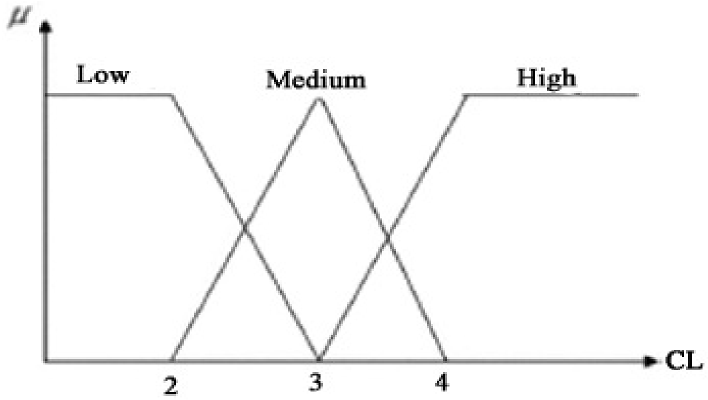

Figure 5. Membership function for CL.

channels request the same output channel, the access is granted to the input channel which has the highest contention level acquired from the upstream switch. This input selection removes possible network congestion by keeping the traffic flowing even in the paths with heavy traffic load, which in turn improves routing performance. Such an input selection helps reduce the number of waiting packets in congested areas. This removes possible network congestions and leads to better NOC performance.

Based on this observation CAIS input selection was developed. For the input channels connected to the cores, there are no upstream switches transmitting CL to them. The CL value is set to 0 for these input channels. Therefore, the packets already in the network have higher priority than the packets waiting to be injected into the network.

With a little attention to above input selection technique (CAIS), we notice an important problem. If an input channel which has lower CL continuously competing with channels which have higher CL, obviously will be defeated any time. The packets in this channel will not be able to get their required output channel and face with starvation and this will cause the problem of decreasing network efficiency. Thus, there is a starvation possibility in this input selection technique, because it performs input selection only based on the highest contention level (CL) and the channels with low CL have a little chance for winning. So, now we try to consider priority parameter in a way that input channels with low CL, have the opportunity to win. Therefore, in addition to CL, another parameter with the name of AGE for every input channel is taken into consideration and measure of priority will be a compound of CL and AGE (Figure 4). The initial value of AGE for every channel is zero. The AGE shows the number of unsuccessful attempt of an input channel to get access its desired output channel.

We have assumed that NOC's router architecture consisting of only input buffers. Priorities of input channels are calculated by CL and AGE with a fuzzy controller as follows:

(a) In each switch, CL and AGE parameters of all input channels are computed.
(b) Then an input selection function is used to select one input channel. The input channel with highest priority is selected to get access to desired output channel. Input selection function is a fuzzy controller. The priority of each input channel is calculated by CL and AGE parameters of input channels with this fuzzy controller. Design of a fuzzy control system for input selection consists of 5 steps:

Step 1: Define the input and output variables, respectively CL and AGE are considered as inputs and input channel priority as the output.
Step 2: Each variable quantified, for instance the CL is quantified as low, medium and high. Also AGE is quantified as small, medium and large. Quantifications of the variables are assigned a membership function as shown in Figures 5 and 6. Membership function for priority is chosen to be singletons.
Step 3: A fuzzy rule base designed. These rule bases determine

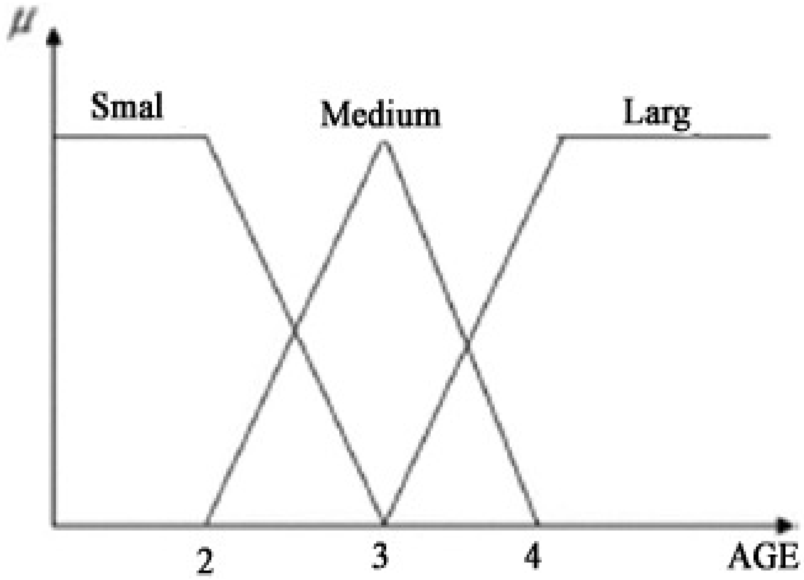

Figure 6. Membership function for AGE.

Table 1. Rule base for FCAIS.

AGE \ CL	Low	Medium	High
Small	Very Low	Low	Medium
Medium	Low	Medium	High
Large	Medium	High	Very High

what action take place under what input conditions. The rules are written in if-then formats as shown in Table 1.

Step 4: A defuzzification method is applied to the fuzzy control action to produce a crisp priority. We use one of the famous defuzzification method "center of gravity" to produce a crisp output.

Step 5: The input channel with highest priority is selected to get access to the output port.

(c) After selection of appropriate input channel, the access to output port is granted to it.

When some input channels compete each other to achieve a specific output channel, finally only one channel will succeed. After this, the AGE of the winner channel will reset to zero and AGE of the other channels entering in the competition will increase for one unit. In this way, each time that an input channel compete with other input channels to achieve specific output channel, in case of failure, its AGE increase one unit and this increase its priority for the following competitions and this increase the opportunity of success. In this fashion finally this input channel will be able to gain its desired output channel. If multiple output channels have the equal priority, the output channel will be granted to that input channel which has higher AGE. And finally in case that all AGE be equal, FCAIS selects an input channel randomly. In this fashion we can eliminate any starvation probability.

Figure 7 illustrates the detailed architecture of a switch with new input selection technique (FCAIS). As can be seen, the structure of this switch is similar to CAIS switch, with slight difference. Here, a very small unit to compute the AGE parameter is added to each port. Also a fuzzy input selector (FIS) is added to each port of the switch.

RESULTS AND DISCUSSION

To evaluate the performance gains that can be achieved with our new input selection technique (FCAIS), we developed a C++ based simulator. Experiments are conducted to evaluate the performance of the FCAIS input selection technique and give a comparison between FCAIS, CAIS and traditional input selections. Due to the advancement of FCFS over Round-Robin, FCFS is selected to compare with CAIS and FCAIS. All input selection techniques are combined with a deterministic output selection (XY routing) and an adaptive output selection (OE routing). The network size during simulation is fixed to be 6×6 tiles. It is assumed that the packets have a fixed length of 5 flits and the buffer size of input channels is 5 flits. The efficiency of each type of routing is evaluated through latency-throughput curves. Similar to other works in the literature, we assume that the packet latency spans the instant when the first flit of

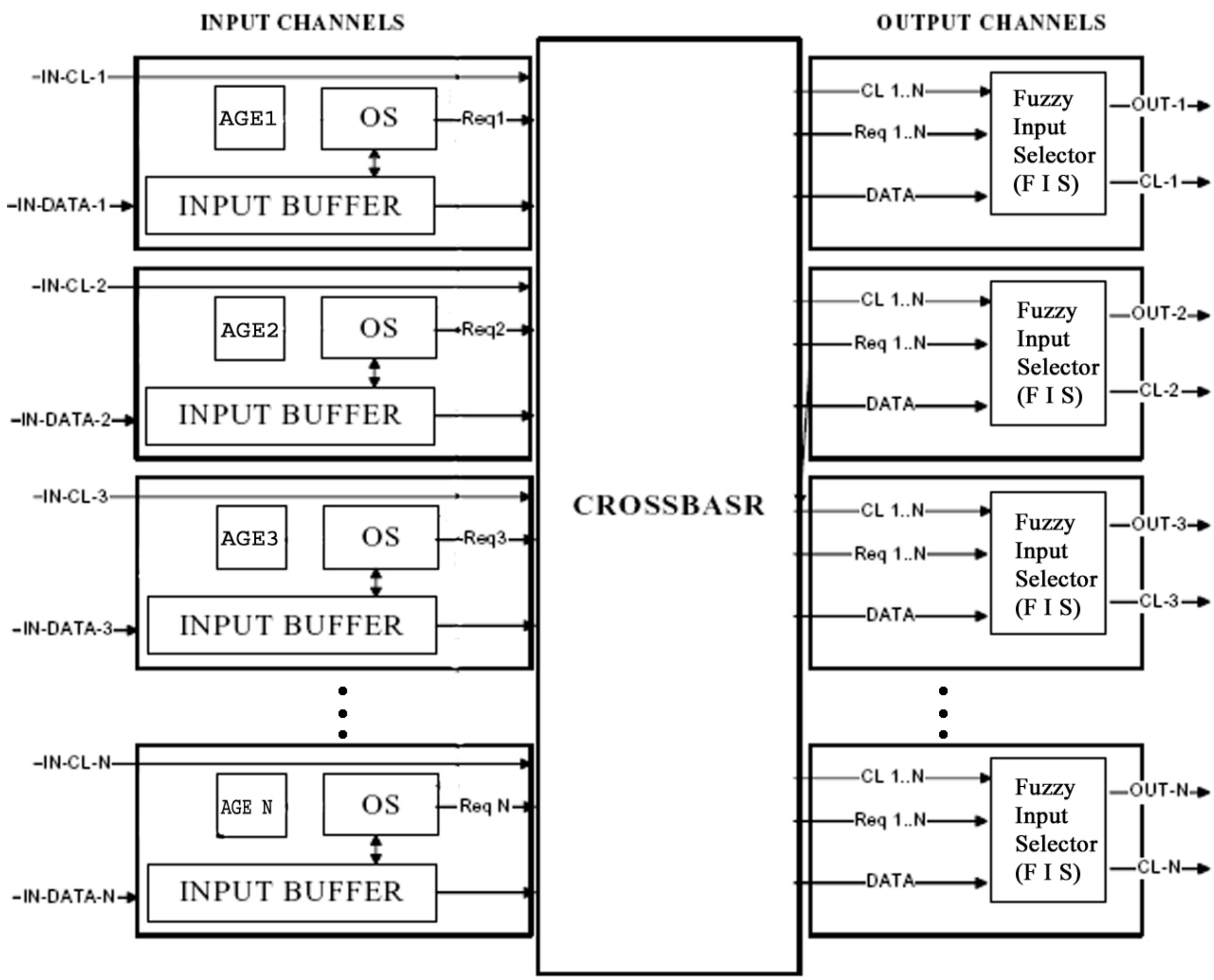

Figure 7. Switch architecture with FCAIS input selection technique.

the packet is created, to the time when last it is ejected to the destination node, including the queuing time at the source.

For each simulation, the packet latencies are averaged over 50,000 packets. Latencies are not collected for the first 5,000 cycles to allow the network to stabilize. Since the network performance is greatly influenced by the traffic pattern, in this set of experiments we consider three synthetic traffic patterns: uniform, transpose and hot spot. In the uniform traffic pattern, a core sends a packet to any other cores with equal probability. In the transpose traffic pattern, a core at (i, j) only send packets to the core at (5-j, 5-i). In the hot spot traffic pattern, the core at (3, 3) is designated as the hot spot, which receives 10% more traffic in addition to the regular uniform traffic.

Figure 8 shows the performance of the six routing schemes under uniform traffic. As can be seen from the figure, the four schemes have almost the same performance at low traffic load (<0.040 packets/cycle). As the traffic load increases, the packet latency rises dramatically due to the network congestion. Comparing the curves of OE+FCFS, OE+CAIS and OE+FCAIS it can be seen that, using the OE output selection, FCAIS performs better than CAIS and FCFS. Similarly, the curves of XY+FCFS, XY+CAIS and XY+FCAIS show that FCAIS also outperforms FCFS and CAIS when using XY output selection. Figure 9 shows the performance of the six routing schemes under transpose traffic. It can be seen that FCFS and CAIS have the same performance when using the XY output selection; FCFS works slightly better than CAIS and FCAIS when using the OE output selection. This is because with transpose traffic, it is rarely the case that more than one input channels compete for the same output channel. Therefore, the input selection policy has little impact on the routing performance. Figure 10 shows the routing performance under hot-spot traffic. Once again, it can be seen that

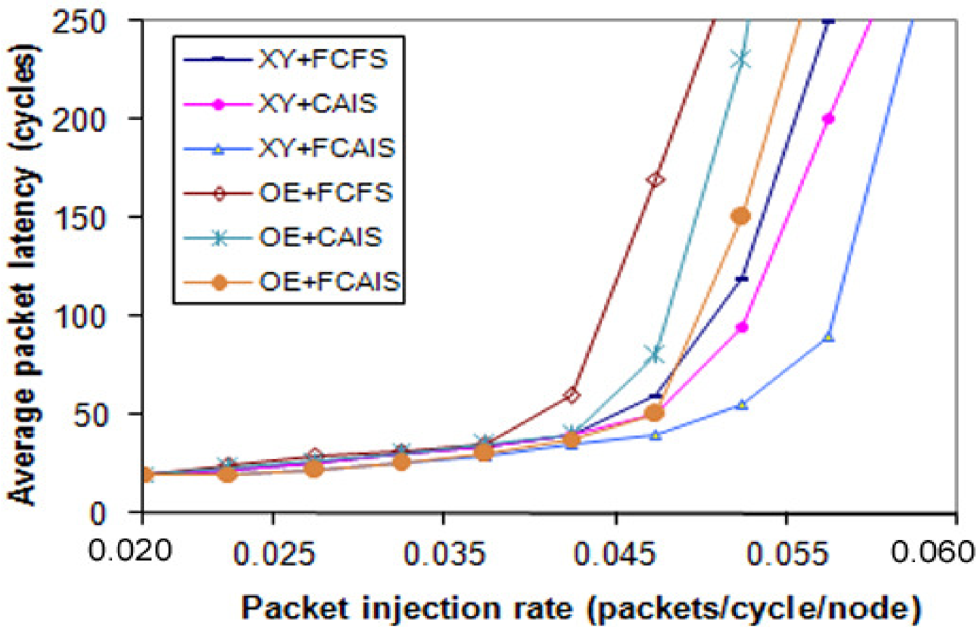

Figure 8. Performance of the routing schemes under uniform traffic.

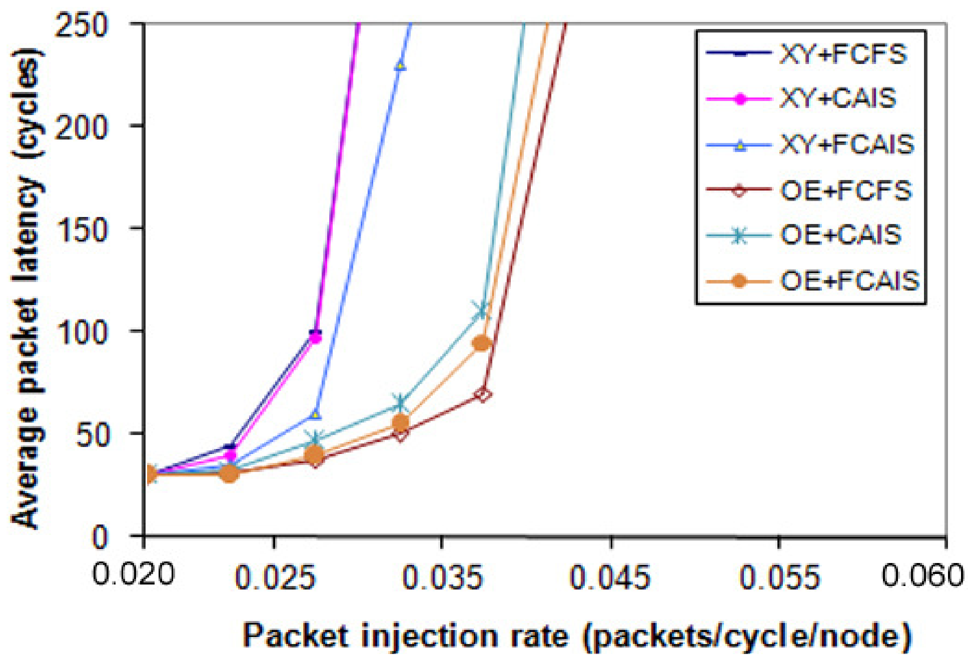

Figure 9. Performance of the routing schemes under transpose traffic.

FCAIS significantly outperforms FCFS and CAIS, either using XY or OE output selection.

Conclusion

The performance of Network-on-Chip largely depends on the underlying routing techniques. A routing technique has two constituencies: output selection and input selection. This paper has shown the importance of input selection in routing efficiency. In this paper a new efficient Fuzzy-based Input Selection technique, FCAIS, is presented which performs more intelligent, by considering the actual traffic condition of the network,

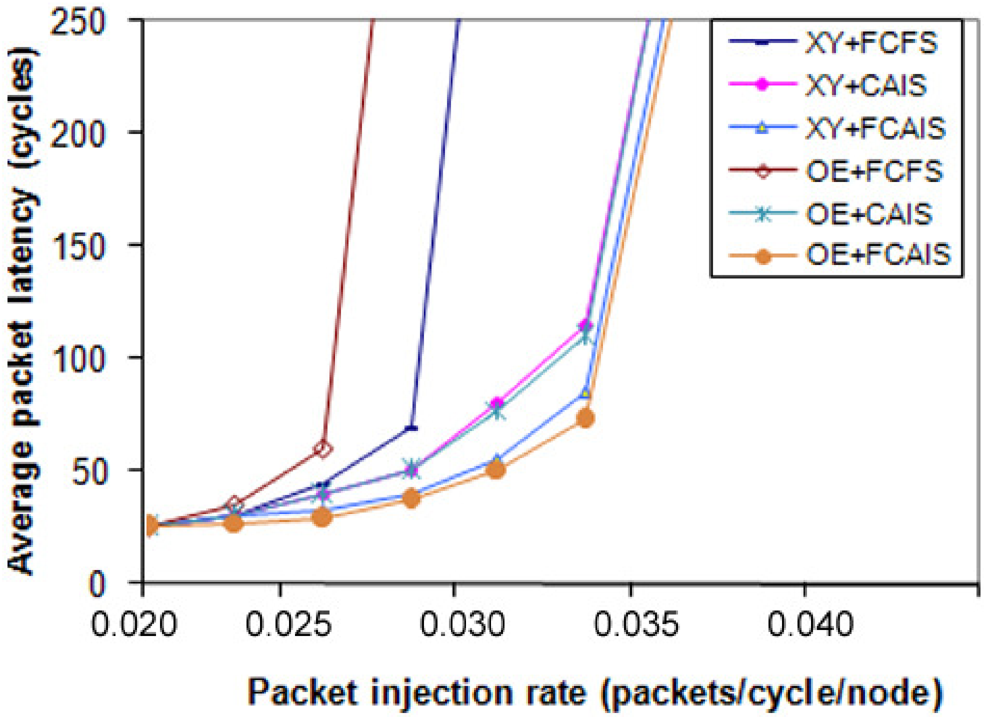

Figure 10. Performance of the routing schemes under hot spot traffic.

leading to higher routing efficiency. In this paper we improve the problems and defections of CAIS technique to develop a simple yet effective input selection technique, FCAIS. The simulation results show the effectiveness of FCAIS by comparing it with CAIS and FCFS traditional input selection techniques.

REFERENCES

Asad A, Seyrafi M, Zonouz EA, Seyrafi M, Soryani M, Fathy M (2009). A Predominant Routing for On-Chip Networks. Paper presented at the proceeding of 4th Internationa workshop on Design and Test (IDT), Riyadh, pp. 1-6.

Nejad BE, Khademzadeh A (2009). BIOS: A New Efficient Routing Algorithm for Network on Chip. J. Contemp. Eng. Sci., 2(1): 37-46.

Benini L, De-Micheli G (2002). Networks on-chip: a new soc paradigm. IEEE J. Comput., 35: 70-78.

Chiu GM (2002). The Odd-Even Turn Model for Adaptive Routing. IEEE Trans. Parall. And Dist. Sys., 11(7): 729-738.

Dally WJ, Towles B (2001). Route Packets, Not Wires: On-Chip Interconnection Networks. Paper presented at the proceeding of the Design Automation Conference (DAC), Stanford Univ., CA, USA, pp. 684-689.

Driankov D, Hellendoom H, Reinfrank M (1993). An introduction to fuzzy control. (2nd ed): Springer-Verlag press.

Dumitras T, Kerner S, Marculescu R (2003). Towards on-chip fault tolerant communication. Paper presented at the proceeding of the of Asia and South Pacific Design Automation Conference, pp. 225-230.

Glass CJ, Ni LM (1992). The Turn Model for Adaptive Routing. Paper presented at the proceeding of the 19th annual international symposium on Computer architecture, Gold Coast, Australia, pp. 278-287.

Hu J, Marculescu R (2004). DyAD-Smart Routing for Networks-on-Chip. Paper presented at the proceeding of the proceeding of Design Automation Conference (DAC), San Diego, California, USA, pp. 260-263.

Ivanov A, De-Micheli G (2005). The Network -on-Chip Paradigm in Practice and Research. IEEE J. Des. Test of Comp., 22: 399-403.

Li M, Zeng QA, Jone WB (2006). DyXY- A Proximity Congestion-Aware Deadlock-Free Dynamic Routing Method for Networks-on-Chip. Paper presented at the proceeding of the ACM/IEEE Design Automation Conference, pp. 849-852.

Mehran A, Saeidi S, khademzadeh A (2007). Spiral: A heuristic mapping algorithm for network on chip. IEICE J. Elec. Exp., 4(15): 478-484.

Pirretti M, Link GM, Brooks RR, Vijaykrishnan N, Kandemir NM, Irwin MJ (2004). Fault tolerant algorithms for network-on-chip interconnect. Paper presented at the proceeding of the IEEE Computer society Annual, pp. 46-51.

Salehi N, Dana A (2010). A fuzzy-based power-aware routing algorithm for network on-chip. Paper presented at the proceeding of the ICACT 2010, Phoenix Park, pp. 1159-1163.

Wu D, Al-Hashimi BM, Schmitz MT (2006). Improving Routing Efficiency for Network-on-Chip through Contention-Aware Input Selection. Paper presented at the proceeding of the 11th Asia and South Pacific Design Automation Conference, Japan.

Ying H (2000). Fuzzy Control and modelling- analytical foundation and applications. (First ed.): IEEE press.

Zadeh LA (1965). Fuzzy sets. J. Inf. Cont., 8: 338-353.