IJBPAS, July, 2015, 4(7): 4311-4324

ISSN: 2277–4998

International Journal of Biology, Pharmacy
and Allied Sciences (IJBPAS)
'A Bridge Between Laboratory and Reader'

www.ijbpas.com

FPASR: A NEW EFFICIENT FUZZY POWER-AWARE SOURCE ROUTING ALGORITHM FOR NETWORK-ON-CHIP

EBRAHIM BEHROUZIAN NEJAD[1*], HAMID MEHDI[2], AMIN EINIPOUR[3]

1: Department of Computer, Andimeshk Branch, Islamic Azad University, Andimeshk, Iran

2: Department of Computer, Andimeshk Branch, Islamic Azad University, Andimeshk, Iran

3: Department of Computer, Andimeshk Branch, Islamic Azad University, Andimeshk, Iran

***Corresponding Author: E Mail: Behrouzian.e@gmail.com**

ABSTRACT

Network-on-Chip (NoC) has been proposed as a solution to provide better modularity, scalability, reliability and higher bandwidth compared to bus-based communication infrastructures. The performance of Network-on-Chip largely depends on the underlying routing techniques. In this paper a new on-chip fuzzy-based source routing algorithm (FPASR) has been presented which try to combine the advantages of both deterministic and adaptive routing schemes. In this scheme each router selects one of the shortest paths with minimum cost which is calculated by a fuzzy controller and considering the free slots input buffer and power consumption of the neighboring switches. Experimental results show that proposed routing algorithm has lower average packet latency and power dissipation than the others routing methods, Also distributes power consumption better than other traditional routing algorithms.

Keywords: Network-on-Chip, Routing algorithm, Source routing, Power consumption, Fuzzy control

INTRODUCTION

Next generations of systems-on-chip (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 meet the performance and design productivity requirements, it has been recently proposed to connect the IPs using a Network-on-Chip (NoC)architecture. NoC [Dally and Towles 2001] [Benini and De Micheli 2002] [Kolson et al. 2002] [Ivanov and Micheli 2005] has been proposed as a solution to provide better modularity, scalability, reliability and higher bandwidth compared to bus-based communication infrastructures. In NOC each core is connected to a switch by a network interface. Cores communicate with each other by sending packets. 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 processing Element (PE), embedded memory, DSP or etc. Other components constitute the communication fabric.

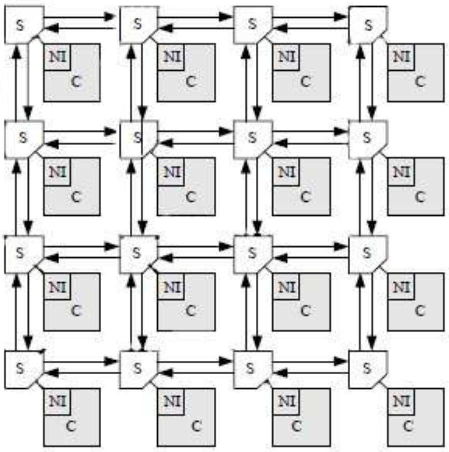


Figure 1: The typical structure of a 4×4-2-D mesh NoC

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. Routing is the process of transmitting data from one node to another in a given network. 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. In a deterministic routing algorithm, messages with the same source and destination addresses always take the same network path. One main advantage of using deterministic routing is its simplicity in term of switch design. Because of the simple logic, the deterministic routing provides low routing latency when the network is not congested. However, as the packet injection rate increases, the deterministic routing algorithms are likely to suffer from throughput degradation as they cannot dynamically respond to network congestion. In contrast, adaptive routing algorithms increase the chances of packets to avoid congested links by using alternative routing paths; this leads to higher throughput. However, because of the extra logic needed in order to decide on a good routing path, adaptive routing has a higher routing latency compared to the deterministic routing, at low levels of network congestion.

In adaptive routing the path between the source and the destination is determined node by node depending on the

Network status.In these types of algorithms, information about the state of the network are used to make routing decisions. As shown in Fig.2 first a routing function computed the set of admissible output channels. Then a selection function selects one output channel from the set of admissible output channels depending on dynamic network conditions [Murali et al. 2006].

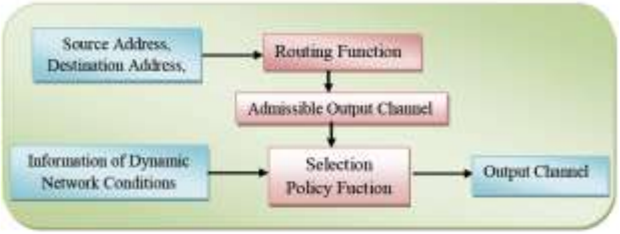


Figure 2: Adaptive Routing Algorithm mechanism

Note that adaptive routing algorithms do not make any guarantees on the arrival order of the packets to the destination. Therefore, when packets order is important the arriving packets must be reordered using some kind of reorder buffer. Also since adaptive routing methods select the routing paths on the basis of short-term local information, it is possible that selected path between each pair of source and destination nodes is not the best available path from a global point of view. On the other hand, power is a critical issue in interconnection network driven by power-related design constraints such as power distribution optimization and thermal Protection design. The focus of this paper is to propose a routing algorithm to solve the above mentioned problems in conventional deterministic and adaptive routing algorithms. We try to combine the advantages of both deterministic and adaptive routing schemes and omit their disadvantages in this proposed routing method.

In this work we present an on-chip fuzzy source routing algorithm that the route is determined by minimum cost which is calculated by fuzzy controller and considering the free slots input buffer and power consumption of the neighboring switches as shown in figure3. A distributed run-time mechanism estimates the power dissipation of the router and shares this information among adjacent routers. Using FPASR algorithm the router selects one of the shortest paths by balancing the power consumption across all its neighbors and avoiding routing packets to congested node.

FPASR routing scheme combines a power model and odd-even routing [Chiu 2000], each router routes the packet toward the destination with considering the cost value which is calculated by fuzzy controller.therefore the router selects one of the shortest paths with lowest power consumption and non-congested node.

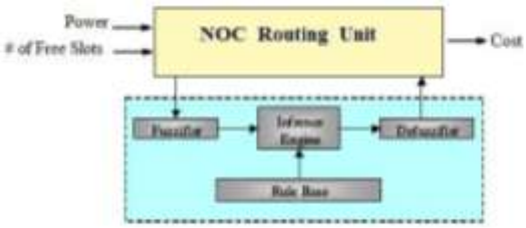


Figure 3: Structure of fuzzy Controller

The rest of the paper is organized as follows. In Section 2, we review related work. In Section 3 the used power model is described. Section 4 describes fuzzy logic control system concepts. In section 5 proposed routing methods (FPASR) is described in detail. Experimental results in Section 6 validate the performance improvements of the proposed routing method. Finally, Section 7 concludes this work.

Related works

The performance of Networks-on-Chip (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 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 degree of routing adaptiveness provided by the model is more even for different source destination pairs. 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 [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 odd-even 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.

3. Power Model

In an interconnection network power is dissipated when flits traverse routers and links. As shown in figure4 (a), flits Flow through an interconnection networks with stream of operations in a typical router without Virtual-channel. (1)

Buffer write, (2) routing, (3) switch arbitration,(4)buffer read, (5) switch traversal and (6) link traversal(figure 4.(a)). But compared to other operations, the power consumption of arbitration and routing logic is negligible [Shang et al. 2003]. thus a simplified router operation for power estimation is derived.(l) buffer write, (2) buffer read, (3) switch traversal and (4) link traversal as shown in **Figure 4.(b).**

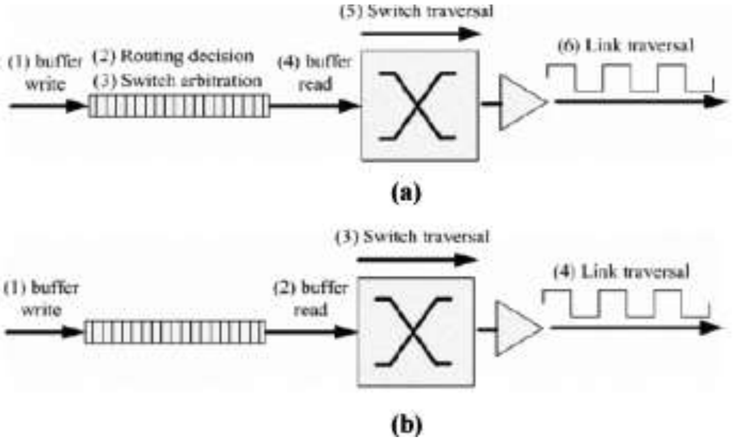


Figure 4: Router operations for on-chip power estimation

In routers, power consumption is highly correlated with the switching activity of the arriving flits at the boundaries of each functional block, as demonstrated in [Wang et al. 2002]. Hence capturing aggregated switching activities at the edge of each functional block is sufficient for on-line power estimation. We thus propose a switching activity based approach for modeling the average power consumption of each operation in figure 4(b). The power model has been used as described in [salehi and Dana 2010] [Yang et al. 2007] which are explained as follows: Input buffers: in a router, buffers read and write are the two operations that trigger dynamic power dissipation in input buffers. The average power consumption for read *(Pbuff_rd)* and *(Pbuff_wr)* operations over time T can be approximately estimated based on four variables: number of read operations *Nbuff_rd* , number of write operations *Nbuff_ wr* , and aggregated switching activities of each read (write) operation, *S(i)buff_rd* , *S(i)buff _ wr* As follows:

$$P_{buff\_rd}=\left(C_{buff\_rd}\times\sum_{i=1}^{N_{Buff\_rd}}S(i)_{buff\_rd}\right)/\mathrm{T} \tag{1}$$

$$P_{buff\_wr}=\left(C_{buff\_wr}\times\sum_{i=1}^{N_{Buff\_wr}}S(i)_{buff\_wr}\right)/\mathrm{T} \tag{2}$$

Cbuff_Wr ,Cbuff_rd is power coefficient of buffer write (read) operation.

Crossbar switch: the crossbar switch relays flits from input buffers to output ports. The boundary switching activities of the crossbar switch *S(i)crossbar* and the number of switch traversal operations *Ncrossbar* are used to model power consumption in a crossbar switch as follows:

$$P_{crossbard}=\left(C_{crossbar}\times\sum_{i=1}^{N_{crossbar}}S(i)_{crossbar}\right)/\mathrm{T} \tag{3}$$

C*crossbar* is power coefficient of switch traversal.

Links: The power consumption of link circuitry is similarly modeled based on input switching activity, *S (i) link* and the number of link traversal operations, *N link:*

$$P_{link}=\left(C_{link}\times\sum_{i=1}^{N_{link}}S(i)_{link}\right)/\mathrm{T} \tag{4}$$

C*link* is power coefficient of link traversal. The total router power consumption in figure.4(b) is thus estimated as:

$$Ptotal = Pbuff\_rd+ Pbuff\_wr+ Pcrossbar+ Plink \tag{5}$$

The aggregated switching activity parameters in the different functions are not independent. Switching activity is the hamming distance between adjacent flits, which changes when either flit data change or when the flit sequence shuffled. In the operation sequence shown in figure 4.*(b),Sbuff _wr ≡ Scrossbar ≡*

$$P_{total}=C_1\times\sum_{i=1}^{N_{flit\_wr}}S(i)_{buff\_wr}+C_2\times\sum_{i=1}^{N_{flit\_rd}}S(i)_{buff\_rd} \tag{6}$$

Slink

The router power model can now be simplified as follows:

Where *N flit_wr (N flit _ rd)* is the number of flits entering (leaving) this router during timeT. Only two aggregate switching activity parameters *Sbuff_wr* and *Sbuff_rd* need to be captured on line. The router power model is simplified as follows:

And

(8)

(7)

$$P_{total}=\left(C_1\times\overline{S_{Buff\_wr}}\times N_{flit\_wr}+C_2\times\overline{S_{buff\_rd}}\times N_{flit\_rd}\right)/T$$

$$P_{total}=\left(W_{wr}\times N_{flit\_wr}+W_{rd}\times N_{flit\_rd}\right)/T$$

Where *Wwr,Wrd* is the power weight of write (read) operation, which mainly depends on the technology adopted in the design, the design itself, and it is a little influenced by applications[salehi and Dana 2010] [Yang et al. 2007].

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, i.e., 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.

There are generally two kinds of fuzzy logic controllers. One is the feedback controller, which is not suitable for the high performance communication networks. Another one, which is used in this paper, is shown in Figure 3. The output of the fuzzy logic controller in Figure 3 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 each quantification of the variable is assigned a membership function. Then a fuzzy rule base must be design, 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.

FPASRRouting Algorithm

The proposed routing algorithm (FPASR) is a kind of source routing algorithms which is a simple and efficient routing protocol designed specifically for use in Network-on-chip. Also, FPASR exploits the advantages of both deterministic and adaptive routing, i.e. in this FPASR method packets can arrive in order as same as the order of their transmission by the source (like deterministic routing). Similar to[Asad et al. 2009]The route between each pair of source and destination nodes is not fixed and when a new flow is detected and needs a route to the destination, the best path is selected based on the current traffic characteristics of the on-chip network (like adaptive routing). The selected paths are the best globally in spite of adaptive routing which selects the routing path on the basis of short term local information. It is possible that this selected path by adaptive routing is not the best route at all globally.The FPASR routing method is composed of two phases:

1. First phase: Path Discovery

2. Second phase: Data Transmission

In the following we describe the two phases in detail.

Path Discovery

When a source node (S) originates a new flow destined to another node D (destination node), it places in the header of the packets a *source route* giving the sequence of the hops that the packets should follow on their way to D. this new created flow needs a route destined to the destination. At this time FPASRrouting algorithm arrives into path discovery phase. When the path discovery phase is initiated, the source node generates two copies of a PATH DISCOVERY (PD) flit and sends them to each minimal outgoing link destined to the destination. Each PD flit is composed of source id, destination id, a record for listing the direction of each router output port on the selected path (see figure 7). Listing the direction of each router output port on the selected path instead of writing the whole address of each router in PD flits is suitable to reduce the overhead of used source routing compared with the traditional source routings. As you can see in Table 1, for encoding each output port direction East, West, North and South two bits are considered. FPASR finds paths along of the shortest paths with minimum cost between the source and destination nodes.

Table 1: Output port encoding

Output port direction	Code
East	00
West	01
North	10
South	11

Cost of a path is calculated by free slot input buffer and power of neighbors with a fuzzy controller. We have assumed that NoC's router architecture consisting of only input buffers. When an intermediate node other than destination receives a PD flit at first time, it does the following steps:

a) At first a routing function computed the set of admissible output channels toward which the packet can be forwarded to reach the destination. We use minimal adaptive odd-even routing algorithm as routing function. The deadlock free and livelock free features of this algorithm are incorporated by limiting a packet to traverse the network only by one of the shortest paths.

b) Then an output selection function is used to select one output channel from the set of admissible output channels. The port with lowest cost is selected to route PD flit. Output selection function is a fuzzy controller. The cost of each port is calculated by free slot input buffer and power of neighbors with this fuzzy controller. Design of a fuzzy control routing system consists of 5 steps:

Step 1: Define the input and output variables, respectively power of neighbors and free slot input buffers as inputs

And cost as the output.

Step 2: Each variable quantified, for instance the power quantified as low,medium and high.Each quantifications of

the variable is assigned a membership function as shown in figure 5 and 6. Membership function for cost is chosen to be singletons.

Step 3: A fuzzy rule base designed. These rule bases determine what action take place under what input conditions. The rules are written in if-then formats as shown in table 2.

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

Step5: The output port with lowest cost is selected to route PD flit.

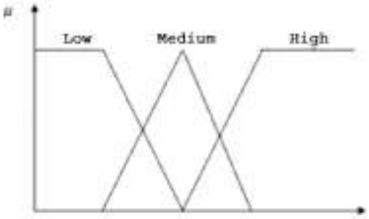


Figure 5: Membership Function for free slots input buffer

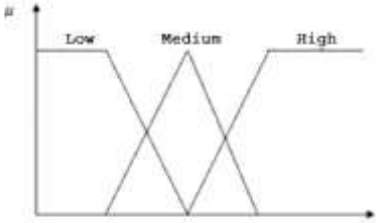


Figure 6: Membership Function for power

Table 2: Rule Base for FPASR routing algorithm

power Free slot	Low	Medium	High

Buffer			
Low	Medium	High	Very High
Medium	Low	Medium	High
High	Very Low	Low	Medium

c) After selection of appropriate output port, direction of selected output port of the current router appended to each PD flits. Then PD flit passed toward the destination through selected output port.

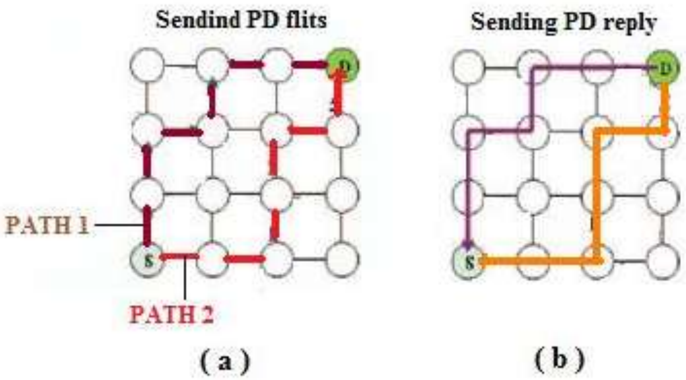
If an intermediate node receives the PD flit which has recently seen another PD flits from the same source node to the same destination; it discards the PD duplicate with higher cost. Ignoring of duplicate PD flits from the same source node to the same destination and discarding them in the intermediate nodes is preferred as it avoids the overhead of PD flits traffic.

Finally two/multiple copies of the PD flit that traversed through different low-cost routes reach to the destination.

5.2.Data Transmission Phase

After sending PD flits by the source node for exploring the best routes to the destination, it receives the PD flits reply. Then it derives the routing information of the PD flits which are representative of a low-cost pathsfor sending subsequent data packets to the destination. In returning the PD flits reply to the source node, the destination node simply reverses the sequence of output ports direction of hops saved in the route record and then uses this as the source route on the PD flit reply back to the source node. When the reply reaches to the source node via the chosen route, the source node receives it, exploits the routing information of it, reverses the sequence of output ports direction of hops saved in the route record again and sends subsequent packets to the destination via the chosen route. Each intermediate node is responsible for sending the received traffic along the source route in Data Transmission phase. The use of source routing allows packet routing to be trivially loop-free and avoids the need for up-to-date routing information in the intermediate nodes.

Fig. 7 shows an example, in which node S is attempting to discover a best route to node D. As you can see in Fig. 7(a), node S generates two copies of a PD flit which include source id S and destination id D and passes them to the minimal output ports destined to the destination. Finally the low cost paths are found by the destination.Fig.7(b) illustrates that two PD reply is transmitted to the source node by the destination for sending subsequent packets via the chosen routes.



	Source ID	Dest. ID	Routers' output port direction					
PATH 1 :	S	D	N	N	E	N	E	E
PATH 2 :	S	D	E	E	N	N	E	N

(c)

Figure7: (a)process of sending PD flits from source(S) node toward Destination(D) node.(b) sending PD reply from Destination(D) node toward Source(S) node . (c). structure of PD flit

Because node S sends two/multiple copies of the PD flits, multiple routes will be discovered .As shown in Fig. 7(a), two/multiple copies of the PD flits which they are representative of two low-latency and low-power paths between node S and node D are received by the destination node. In this step node D does not discard the higher cost PD flit and sends its reply together lower cost PD reply to the source node. Node S sets up the route whichever comes first. If the first route is broken, node S chooses the second one. The node S can have multiple routes for alternative routes.As a result it can achieve the fault tolerant. Also source node can split traffic and send a different portion of it into the two found best routes to minimize the maximum traffic on each link of the NoC topology.

Experimental Results

To evaluate the performance that can be achieved with proposed method, we have implemented a NoC simulation platform developed in SystemC coupled with Orion simulator [Wang et al. 2002] as a separate plug-in power analysis tool, to our network simulator. This simulator can calculate the average delay and the total energy for the packet transmission. An 8×8 2D-mesh is considered for simulation. Each packet is composed of 8 flits and FIFO buffers have a capacity of four flits. Flits size is assumed to be 48 bits. To compare performance of the FPASR method we implemented three other routing algorithms include DyAD, XY, and Odd-Even. Similar to other work in the literature, we assume that the packet latency spans the instant when the first flit of the packet is created, to the time when last it is ejected to the destination node, including the queuing time at the source. Each simulation is run for a warm-up period of 1000 cycles. Thereafter, performance data are collected after 20000 packets are sent. Since the network performance is greatly influenced by the traffic pattern, in this set of experiments we consider three 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 (N-1-j, N-1-i). In the hot spot traffic pattern, the cores at (3, 3),(3,4),(4,3) and (4,4) are designated as the hot spot, which receives 20% more traffic in addition to the regular uniform traffic.

As shown in Fig.8, XY routing performs better than odd-even, DyAD and FPASR routing algorithms under uniform traffic load. This result is consistent with other results reported in the other literature. The reason why XY performs the best under uniform traffic is that it embodies global, long-term information about this traffic pattern. From a global, long-term point of view, the uniform traffic pattern starts with message traffic spread evenly across the mesh; later on the XY routing strategy maintains that evenness. On the other hand, the adaptive algorithms select the routing paths based on local, short-term information. The decision benefits only the packets in the immediate future, which tend to interfere with other packets. Thus, the evenness of uniform traffic is not necessarily maintained in the long run.

However, for most of the applications in real world, each node will communicate with some nodes much more compared to others. XY routing has serious deficiency in dealing with such non-uniform traffic patterns because of its determinism. As shown in figure.9 and figure.10, taking the results using transpose and hot-spot traffics, the network using XY saturates at very low injection. But odd-even, DyAD and presented routing algorithm(FPASR) are able to achieve a higher throughput than XY.

As shown in **Figure 9 And Figure 10**, for the same traffic pattern and the injection rate, FPASR routing algorithm achieves shorter average packet latency and higher throughput in these experiments compared to XY, odd-even and DyAD. The effectiveness of proposed routing algorithm(FPASR) is confirmed by the fact that it continuously outperforms odd-even and DyAD in terms of sustainable throughput in all these experiments.

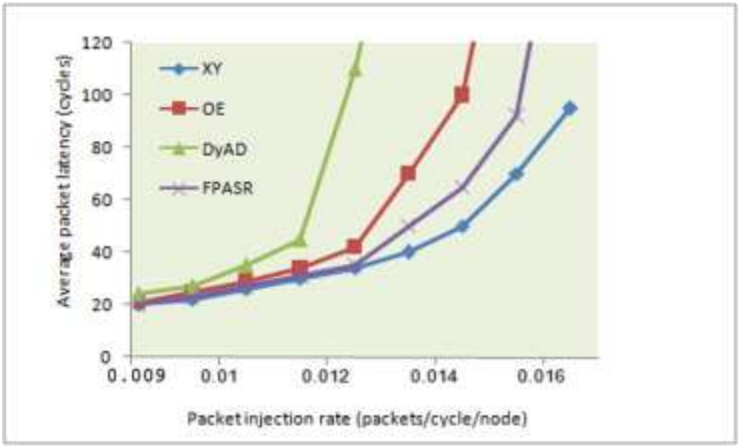


Figure 8: Performance of different routing algorithms under uniform traffic

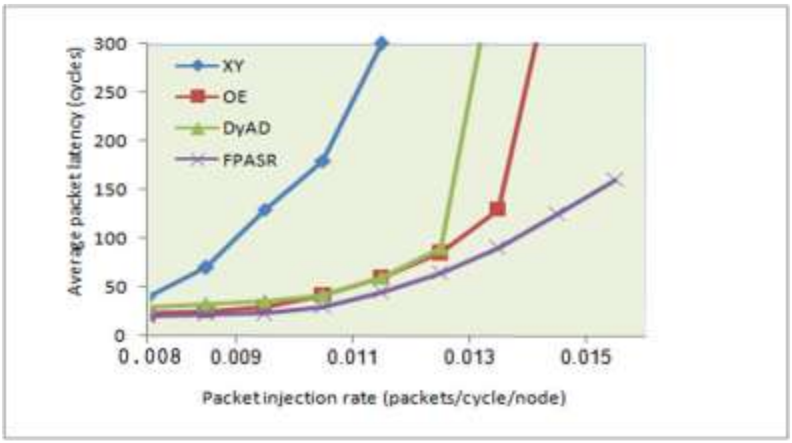


Figure 9: Performance of different routing algorithms under transpose traffic

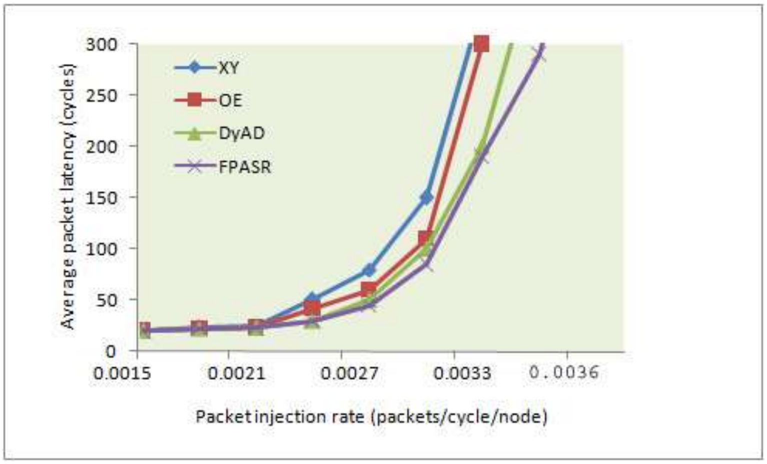


Figure 10: Performance of different routing algorithms under hot-spot traffic

CONCLUSION

In this paper a new efficient power-aware source routing algorithm (FPASR) for Network-on-Chip is presented. FPASR routing algorithm is a kind of source routing algorithm which is based on a fuzzy controller to select the best output port in each step. Also, proposed algorithm exploits the advantages of both deterministic and adaptive routing algorithms. FPASR routing algorithm is composed of two phases: Path Discovery and Data Transmission. It is able to find low-latency and low-power paths between each pair of source and destination nodes. The simulation results show that proposed routing algorithm achieves lower average packet latency and higher throughput in compared to other routing algorithms, under different traffic patterns. Also using fuzzy-based output selection, it is able to distribute power consumption better than traditional routing algorithms.

REFERENCES

[1] A. Asad, M. Seyrafi,A.Ehsani Zonouz, M. Seyrafi, M. Soryani, M. Fathy,"A Predominant Routing for On-Chip Networks", IDT 2009, pp. 1 – 6, Riyadh 15-17 Nov. 2009.

[2] L. Benini and G. De Micheli. Networks on-chip: a new soc paradigm. *IEEE Computer,* January, 2002.

[3] G. M. Chiu, "The Odd-Even Turn Model for Adaptive Routing," IEEE Transactions on Parallel and Distributed Systems, vol.1, no.7, 2000.

[4] W. J. Dally and B. Towles, "Route Packets, Not Wires: On-Chip Interconnection Networks,"DAC, June 2001, pp. 684-689.

[5] D. Driankov, H. Hellendoo m and M. Reinfrank, An introduction *to fuzzy control, Springer- Verlag,* Berlin, New York, 1993.

[6] T. Dumitras, S. Kerner, and R. Marculescu, "Towards on-chip faulttolerant communication", in Proceedings Asia and South Pacific Design Automation Conference, 2003.

[7] C. J. Glass, L. M. Ni, "The Turn Model for Adaptive Routing," in ISCA '92: Proceeding of the 19th annual international symposium on Computer architecture, 1992.

[8] J. Hu and, R. Marculescu, "DyAD-Smart Routing for Networks-on-Chip," DAC 2004, pp. 260 – 263, San Diego, California, USA, 2004.

[9] A. Ivanov and G.D. Micheli, "The Network -on-Chip Paradigm in Practice and Research," IEEE Design and Test of Computers, vol. 22, no. 5, pp. 399-403, Sept./Oct. 2005.

[10] S. Kolson, A. Jantsch, et al.A NoC architecture and design methodology. In *Proc. Annual Symp.VLSI,*2002.

[11] M. Li, Q.-A.Zeng, and W.-B. Jone, "DyXY- A Proximity Congestion-Aware Deadlock-Free Dynamic Routing Method for Networks on-Chip," Proc. ACMlIEEE Design Automation Conf., pp. 849-852, July 2006

[12] S. Murali, D. Atienza, L. Benini, and G. De Micheli, “A multipath routing strategy with guaranteed in-order packet delivery and faulttolerance for networks on-chip,” in Proceedings of the 43rd ACM/IEEE Design Automation Conference (DAC ’06), pp.845–848, San Francisco, Calif, USA, July 2006.

[13] M. Pirretti, G. M. Link, R. R Brooks, N. Vijaykrishnan, N, M. Kandemir, M. J. Irwin, “Fault tolerant algorithms for network-on-chip interconnect”, in Proceedings IEEE Computer society Annual Symposium on 19-20, pp. 46 – 51, Feb. 2004.

[14] N. salehi, A. Dana, “A fuzzy-based power-aware routing algorithm for network on-chip”,ICACT 2010, pp 1159-1163, Phoenix Park 7-10 Feb. 2010.

[15] L. Shang, L. S. Peh and N. K. Jha. PowerHerd: Dynamic Satisfaction of Peak Power Constraints in Interconnection Networks.ICS'03, 2003.

[16] H. Wang, X. Zhu, L.-S. Peh and S. Malik, “Orion: A Power- Performance Simulator for Interconnection Networks”, Proc. MICRO, pp. 294-395, 2002.

[17] Sh.Yang,L. Li Y. Xu Y.Zhang B. Zhang. A power-aware adaptive routing scheme for network on a chip .ASIC, 2007.ASICON'07.pp.13011304, 22-25 Oct. 2007

[18] HaoYing. Fuzzy Control and modelling-analytical foundation and applications,IEEE press, 2000

[19] L. A. Zadeh, Fuzzy sets.Information and Control, Vol. 8, No. 3, pp. 338-353, 1965.