

BIOS: A New Efficient Routing Algorithm for Network on Chip

Ebrahim Behrouzian-Nezhad¹ and Ahmad Khademzadeh²

¹ Computer Engineering Department, Islamic Azad University
Science & Research Branch, Tehran, Iran
Behrouzian.e@srbiau.ac.ir

² Iran Telecommunication Research Center, Tehran, Iran
zadeh@itrc.ac.ir

Abstract

The performance of Network-on-Chip (NoC) largely depends on the underlying routing techniques, which have two constituencies: output selection and input selection. Previous research on routing techniques for NoC has focused on the improvement of output selection. In this paper a novel routing algorithm for network on chip, which called BIOS, is presented. BIOS improves both input and output selection. In fact, BIOS is based on the best input and output selection. Simulation results with different traffic patterns show that our new routing algorithm(BIOS) achieves significant better performance than the other deterministic and adaptive routing algorithms.

Keywords: system-on-chip, network-on-chip, routing algorithm, input selection, output selection.

1. 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 As the complexity of applications and their required algorithms have grown so rapidly, SOB has been replaced by System on Chip (SOC) methodology[1]. 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 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 System-on-Chip (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 (NoC) architecture. 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.

Fig. 1 shows an abstract view of a NOC in this architecture. As shown, each tile is composed of a resource(R) and a switch or router(S). The router is connected to the four neighboring tiles and its local resource via channels. Each channel consists of two directional point-to-point links between two routers or a router and a local resource.[5,7,9]

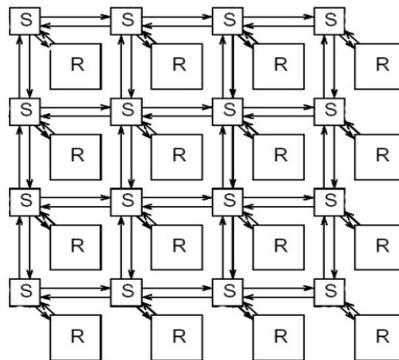


Fig.1 . The typical structure of a 4*4 NOC

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 behaviour of the switches[11]. 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 (e.g. congested links due to traffic variability). One main advantage of using deterministic routing is its simplicity in terms of routers design. Because of the simplified 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 can not 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 [6,12]. In this paper, we will present a novel routing algorithm called BIOS which combines the advantages of both deterministic and adaptive routing algorithms.

2. Related Work

The idea of NoC is derived from large-scale computer networks and distributed computing. However, the routing techniques for NoC have some unique design considerations besides low latency and high throughput. Due to tight constraints on memory and computing resources, the routing techniques for NoC should be reasonably simple. Several switch architectures have been developed for NoC [2,3,8], employing XY output selection and wormhole routing. The routing technique proposed in [3] acquire information from the neighboring switches to avoid network congestion and uses the buffer levels of the downstream switches to perform the output selection. A routing scheme which is based on Odd-Even routing algorithm[4] and combines deterministic and adaptive routing is proposed in [6], where the switch works in deterministic mode when the network is not congested, and switches to adaptive mode when the network becomes congested. All of these routing techniques focused on the output selection. Two input selections have been used in NoC, First-Come-First-Served (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 [12] an input selection technique is presented, but there is a starvation possibility in this technique. The motivation of this paper is to improve both input selection and output selection to develop a simple yet effective routing algorithm.

3. Proposed Routing Algorithm

Each routing algorithm has two constituencies: output selection and input selection. For output selection part, similar to [6], We will present an approach to combining the advantages of both deterministic and adaptive routing algorithms. For input selection part, we will present a novel technique which is based on the actual traffic conditions of the upstream switches and performs more intelligent than FCFS and Round Robin input selection. The platform under consideration is composed of a $n \times n$ array of tiles which are interconnected by a 2D mesh network. In our experiments, the XY routing scheme is picked up as a representative deterministic routing scheme because of its simplicity and wide popularity. In short, for 2D mesh networks, the XY routing first routes packets along the X-axis. Once the packets reach the column wherein lies the destination tile, they are then routed along the Y -axis. Obviously, XY routing is a minimal path routing

algorithm and is free of deadlock and livelock. Also, odd-even and DyAD routing algorithms[4,6] are picked up as adaptive routing schemes. Compared to other adaptive routing algorithms without virtual channel support, odd-even and DyAD provide more even adaptiveness. Thus, in this paper, minimal version of these routing algorithms are used. The use of minimal routing helps not only in reducing the energy consumption of communication, but also to keeping the network free from the livelock.

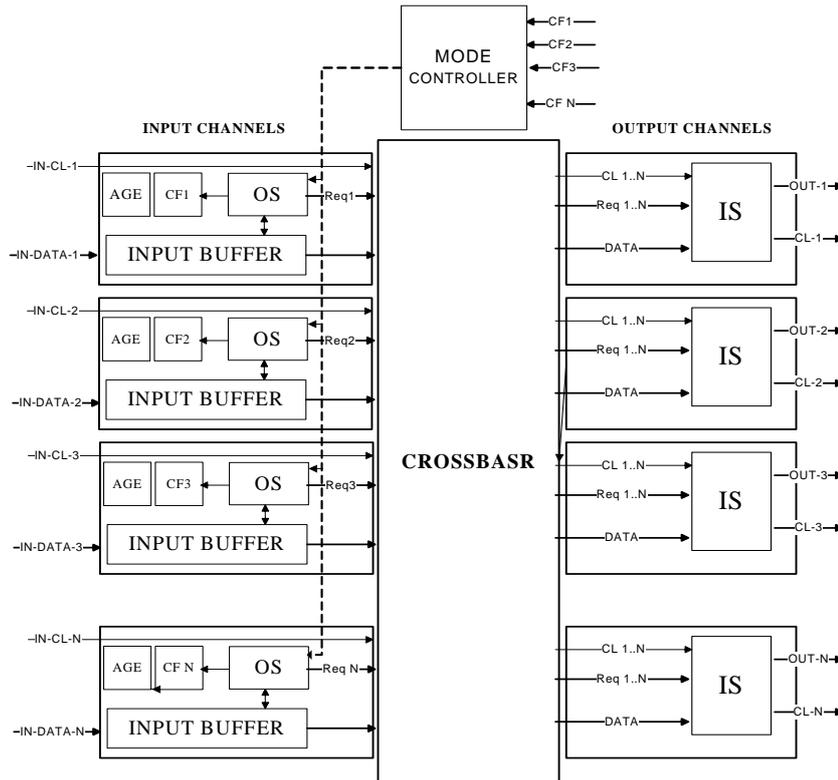


Fig.2. BIOS router architecture

In this subsection, we present the actual router design, which implements the concept of BIOS routing algorithm. Combining odd-even and XY to form a router may lead to deadlock problem. Thus, we develop a new routing scheme, called DOE, as the deterministic routing mode in new algorithm. DOE is indeed a deterministic version of odd-even based on removing the odd-even's adaptiveness. Fig.2 illustrates the detailed architecture of a switch for BIOS. Each input port in Fig. 2 has a separate FIFO (typically the size of several flits) which buffers the input packets before delivering them to the output ports.

3.1. Output Selection

In this section we envision a new routing technique which judiciously switches between deterministic and adaptive routing based on the network's congestion conditions. Each router in the network continuously monitors its local network load and makes decisions based on this information. When the network is not congested, router works in a deterministic mode and thus enjoys the low routing latency enabled by deterministic routing. On the contrary, when the network becomes congested, the router switches back to the adaptive routing mode and thus avoids the congested links by exploiting other

routing paths; this leads to higher network throughput.

When a new header flit is received, the output selector(OS) processes that flit and determines which output port the packet should be delivered to. When the router works in the odd-even mode(adaptive mode), there can be more than one output direction to route packets. In this case, the output selector(OS) will choose the direction in which the corresponding downstream router has more empty slots in its input FIFO. For instance, let us suppose the router located in the position of (1; 1) as in Fig. 3 receives a packet from router (1; 0) with the destination address (3; 3). Under odd-even mode, the packet can be routed either to north or to east in this router. In this case, the output selector compares the occupancy of the south input FIFO at router (2; 1) with that of the west input FIFO at router (1; 2). If the former has more flits in the FIFO, then the packet will be routed to the east to router (1; 2); otherwise, it will be routed to the north to router (2; 1). Once the router has made its decision on which direction to route, the output selector sends the connection request to the input selector(IS) in order to set up a path to the corresponding output port.

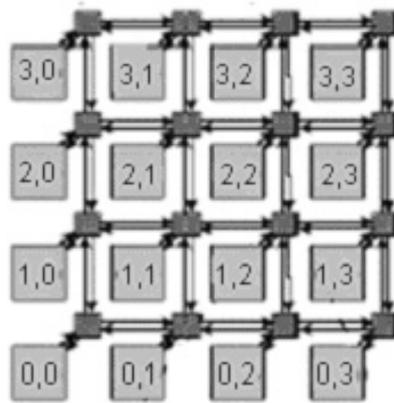


Fig.3. Output selection with BIOS in a typical NOC

Except for the local output selector, each output selector also monitors its FIFO occupation ratio. If the ratio reaches the specified congestion threshold, a value 1 will be asserted on the corresponding congestion flag. Otherwise, a value of 0 will be asserted. If the FIFO is full, a value 2 will be asserted on the corresponding congestion flag. Intuitively, a value of 1 in the congestion flag indicates to the upstream router that the downstream router is congested, and it is better to use adaptive routing in this case in order to avoid possible congested links. On the other hand, a value of 0 tells the upstream router that congestion is not an issue and it should dump the packets out as fast as possible for minimum latency. a value of 2 tells the upstream router that congestion in downstream router is very high and this router is not able to route packets at this time. The mode controller continuously monitors its neighboring congestion to determine whether the deterministic or the adaptive routing mode should be used. If any congestion flag from its neighboring routers are asserted 1 or 2, then the mode controller commands all the output selectors to work at the adaptive (odd-even) mode; otherwise, it switches the output selectors to deterministic (DOE) mode. If all congestion flag from its neighboring routers are asserted 2, the packet should

sent back to previous upstream switch that come from(backtracking).

3.2. Input Selection

In this section we will present a novel input selection technique for NOC. Multiple input channels may request simultaneously the access of the same output channel, e.g., packets p_0 of input_0 and p_1 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, First-Come-First-Served (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 [12], 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. When multiple input 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 a new input selection is 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, 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, won't 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 new 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+AGE. The initial value of AGE for every channel is zero. 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. With this new criteria(CL+AGE) each time that an input channel compete with other input channels to achieve specific output channel, in case of failure, it's AGE increase one unit and this increase its priority for the following competitions and this itself increase the opportunity of success and finally this channel be able to gain its desired output channel.

In competition the following conditions may occurs :

- a) if the priority(CL+AGE) of an input channel be higher than of other input channels, then the desired output channel will be granted to it and then its AGE will reset to zero and the AGE of all other input channel will increase for one unit.
- b) If multiple output channels have the equal priority, the output channel will be granted to that input channel which has higher AGE, then its AGE will be reset to zero and the AGE of other input channels will increase for one unit.

```

Output selection :
OS function : for ( ; ; ){
Check local input buffer;
Occ = # of packets into local buffer;
If occ <= T then assert 0 into local CF
Else if occ > T && buffer is not full then
    Assert 1 into local CF
    Else assert 2 into local CF; }
For each arrival packet do : {
Check destination addr of packet;
Check congestion of downstream switchs in all allowed
output directions :
If all CF are 2 then backtrack packet
Else if all CF are 0 then use DOE & deterministic mode
Else use adpaptive mode to select best output channel;}
    
```

```

Input selection :
IS function : for ( ; ; ){
CL = # of access request to j th output channel;
send CL to downstream switch;
for each input channel i do{ AGE i = # of unsuccessful try ;
M i = observed (CLi) + AGE i;}
In each competition do {
if M i > M j,Mk,... then { output channel granted to channel i;
AGE i = 0; increament AGE of M j,Mk,...}
if M i = M j then
if AGE i > AGE j then { grant the access to channel i;
AGE i = 0; increament AGE of M j,Mk,...; }
else if AGE i < AGE j then{
get the access to channel j; AGE j=0;
increament AGE of M i,Mk,...; }
else { get acces to a chanel randomly;
winner AGE = 0;
increament AGE of other channel.} }}
    
```

Fig. 4. Pseudo code of input selection and Output selection parts of BIOS algorithm

In this way, we have been able to remove the problem of starvation. Therefore, BIOS routing algorithm is deadlock and starvation free and has a high degree of adaptiveness and is able to make decision according to real traffic conditions of the network. Fig. 4 shows the pseudo code of BIOS algorithm, which consists of two parts, Input selection and Output selection.

4. Experimental Results

To evaluate the performance gains that can be achieved with BIOS, we developed a C++

based simulator on standard template libraries(STL). Simulations are carried out on a 6×6 mesh NoC. we simulate several square mesh networks with XY, odd-even, DyAD and BIOS routing algorithms. The efficiency of each type of routing is evaluated through latency-throughput curves. 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 2000 cycles. Thereafter, performance data are collected after 20,000 packets are sent. 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. The network size during simulation is fixed to be 6×6 tiles. All of the input ports have a FIFO size of 5 flits, with the congestion threshold set at 60% of the total FIFO capacity.

As shown in Fig.5(a), XY routing performs better than odd-even, DyAD and BIOS routing algorithms under uniform traffic load. This result is consistent with other results reported in the other literature. The reason why XY performs 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. Taking the results using transpose traffic, the network using XY saturates at very low injection. But odd-even, DyAD and BIOS routing algorithm are able to achieve a higher throughput than XY. As shown in Fig.5, for the same traffic pattern and the injection rate, BIOS routing algorithm achieves shorter average packet latency and higher throughput in all these experiments compared to odd-even and DyAD. The effectiveness of BIOS routing algorithm is confirmed by the fact that it continuously outperforms odd-even and DyAD in terms of sustainable throughput in all these experiments. In fact, for the same traffic pattern and the injection rate, BIOS routing algorithm achieves shorter average packet latency compared to odd-even and DyAD throughout the experiments. Another interesting fact to observe is that BIOS routing algorithm does keep the advantage of deterministic routing when network is not congested. As shown in Fig. 5(a) and Fig 5(b) and Fig 5(c), BIOS routing algorithm has the same average packet latency when network is not

congested. On the other hand, compared to BIOS routing algorithm, the average latency a packet experiences in odd-even and DyAD, at least is 18% higher compared to that in BIOS, when the network is lightly loaded. It appears that implementing a BIOS routing algorithm router requires an almost negligible additional cost compared to an odd-even router. To justify our claim, we actually implemented all four versions of routers (XY, odd-even, DyAD and BIOS) using a 0.16 μ m technology, with a clock rate of 333MHz. Their respective area (in gates) is shown in Fig.5(d) for odd column routers with FIFO size of 8 flits. In all the designs, each input port has a fixed link width of 32 bits. The flit size is set to be 32 bits as well. As shown in Fig.5(d), the overhead of implementing the extra logic for BIOS routing algorithm is indeed negligible compared with odd-even and DyAD implementation and is below 5%.

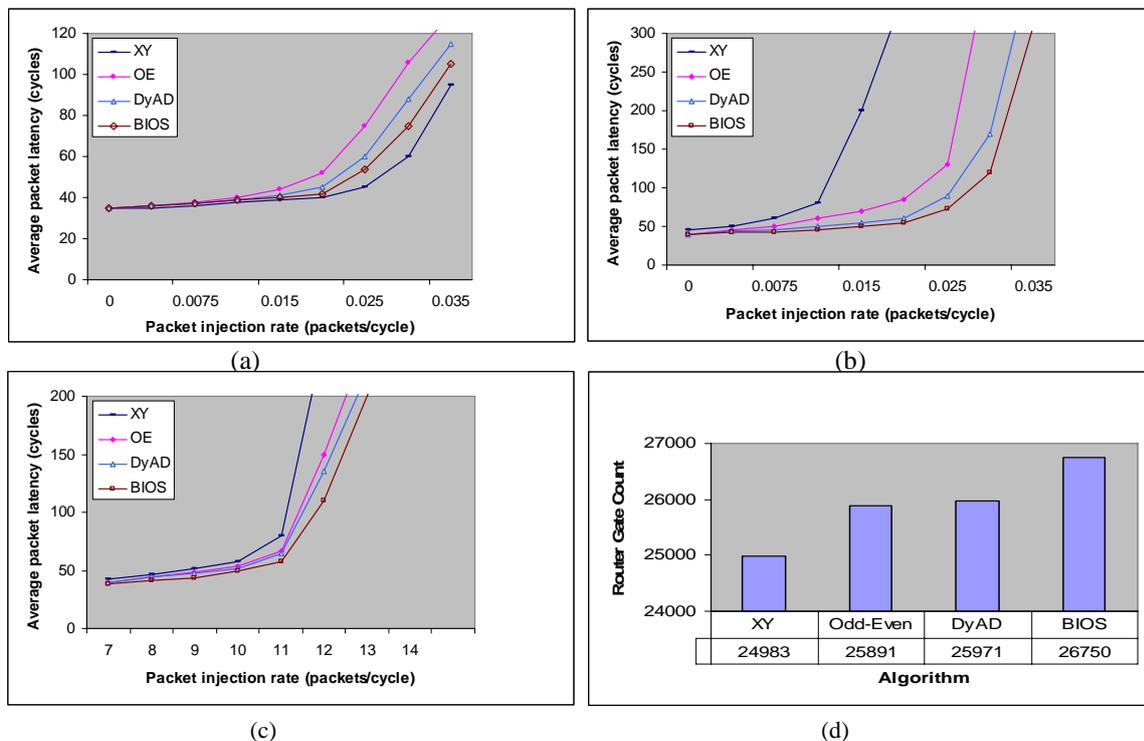


Fig.5. (a) Routing performance under uniform traffic, (b) Routing performance under transpose traffic, (c) Routing performance under hot spot traffic, (d) complexity of routers (gate count)

advantages of both deterministic and adaptive routing algorithms. we improved both output selection and input selection. This paper has shown the importance of both input selection and output selection in routing efficiency. The simulation results show the effectiveness of BIOS by comparing it with purely deterministic and adaptive routing algorithms under different traffic patterns. Moreover, a prototype router based on the BIOS idea has been designed and evaluated. Compared to purely adaptive routers, the overhead of implementing BIOS is negligible (less than 5%), while the performance is consistently better.

References

- [1] A.Mehran, S.Saeidi, A.khademzadeh, "Spiral: A heuristic mapping algorithm for network on chip", IEICE Electronic Express, Vol.4, No.15, 478, Japan, Aug 2007
- [2] C. A. Zeferino, M. E. Kreutz, and A. A. Susin, "RASoC: A router soft-core for Networks-on-Chip," Designers Forum - DATE, pp. 198-203, France, 2004.
- [3] E. Rijpkema, K. Goossens, A. Radulescu, J. Dielissen, J. Van Meerbergen, P. Wielage, and E. Waterlander, "Trade-offs in the design of a router with both guaranteed and best-effort services for networks on chip," IEE Proceedings: Computers and Digital Techniques, vol. 150, pp. 294-302, 2003.
- [4] G.-M. Chiu, "The odd-even turn model for adaptive routing," IEEE Transactions on Parallel and Distributed Systems, vol. 11, pp. 729-38, 2000.
- [5] J. Henkel, W. Wolf, and S. Chakradhar, "On-chip networks: A scalable, communication-centric embedded system design paradigm," VLSI Design, pp. 845-851, India, 2004.
- [6] J. Hu and R. Marculescu, "DyAD - Smart routing for networks-on-chip," DAC, pp. 260-263, USA, 2004
- [7] L. Benini and G. De Micheli, "Networks on chips: A new SoC paradigm," Computer, vol. 35, pp. 70-78, 2002.
- [8] N. Kavaldjiev, G. J. M. Smit, and P. G. Jansen, "A virtual channel router for on-chip networks," IEEE International SOC Conference, pp. 289-93, USA, 2004.
- [9] S. Kumar, A. Jantsch, J.-P. Soininen, M. Forsell, M. Millberg, J. Oberg, et al, "A network on chip architecture and design methodology," ISVLSI, pp. 117-24, USA, 2002.
- [10] T. T. Ye, L. Benini, and G. De Micheli, "Packetization and routing analysis of on-chip multiprocessor networks," Journal of Systems Architecture, vol. 50, pp. 81-104, 2004.
- [11] W. J. Dally and B. Towles, "Route packets, not wires: On-chip interconnection networks," DAC, pp. 684-689, USA, 2001.
- [12] Wu, D., Al-Hashimi, B. M. and Schmitz, M. T. (2006) Improving Routing Efficiency for Network-on-Chip through Contention-Aware Input Selection. In: 11th Conference (ASP-DAC 2006), 2006, Japan.

Received: June 11, 2008