



# Adaptive Input-Output Selection Based On-Chip Router Architecture

M. Daneshtalab<sup>1,2,\*</sup>, M. Ebrahimi<sup>1</sup>, S. Mohammadi<sup>2</sup>, and A. Afzali-Kusha<sup>2</sup>

<sup>1</sup>Department of Information Technology, University of Turku, Finland

<sup>2</sup>School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran

(Received: 24 June 2011; Accepted: 10 October 2011)

In this paper, we propose a novel on-chip router architecture, named Adaptive Input-Output Selection (AIOS), for networks-on-chip. The architecture employs efficient input and output selection methods in order to reduce the maximum power consumption and latency of the network. The output selection of AIOS utilizes an adaptive minimal and non-minimal routing algorithm which relies on the congestion condition of neighboring routers to circumvent the congested areas in the network. Moreover, the presented routing scheme is capable of supporting both unicast and multicast communication. When multiple input ports competing for the same output port, the input selection of AIOS serves each input port according to its congestion level to diminish possible network congestion. The simulation results show that in synthetic and realistic traffic profiles the presented router architecture reduces both average latency and maximum power consumption compared to baseline architectures.

**Keywords:** Networks-on-Chip, Routing Algorithm, Arbitration Mechanism, Power Consumption.

## 1. INTRODUCTION

In future MultiProcessor System-on-Chip (MPSoC) designs, the interconnection imposes some complexities ranging from scalability and energy, to device reliability.<sup>1</sup> A possible approach for coping with this problem is to use an on-chip interconnection network instead of an ad-hoc global wiring.<sup>2</sup> The tile-based Network-on-Chip (NoC) architecture is known as a suitable solution for overcoming communication problems in future VLSI circuits.<sup>2-4</sup> Such chips are composed of many tiles regularly positioned in a grid where each tile can be a memory, CPU core, DSP core, video stream processor and high-bandwidth I/O unit. These components are connected to their adjacent tiles through on-chip routers (Fig. 1).<sup>2</sup> In this communication platform, data is transferred between tiles through packets without requiring dedicated wirings. That is, NoCs not only offer a scalable performance needed by systems which grow with each new generation but also allow to mitigate the energy consumption by avoiding the use of long global wires.<sup>1</sup> Furthermore, NoCs are reusable platforms and aid to reduce the design productivity gap. Finally, none of the current on-chip interconnect

approaches (buses and dedicated point-to-point channels) meet the requirements of future SoCs, as NoCs promise to do.

The performance and efficiency of NoCs largely depend on the output-selection and input-selection methods exploited by on-chip routers. The output-selection method, using a routing algorithm, determines which output channel should be chosen for a packet arrived from an input channel. The input-selection method chooses one of input channels to get access to the output channel, which is performed by an arbitration process.

In this paper, we present a novel router architecture, named Adaptive Input-Output Selection (AIOS) to improve the network performance and reduce the maximum power consumption of network. AIOS employs efficient input-selection and output-selection methods in the routing process where the key ideas are twofold.

The first idea is to circumvent possible congested areas in the network. Therefore, we introduce an adaptive output-selection method using both minimal and non-minimal paths based on the congestion condition of neighboring routers. Since multicast communication is used in many parallel applications (e.g., cache coherency, clock synchronization, replication, etc.).<sup>5,6</sup> we enhance a Hamiltonian path routing scheme not only to support both unicast and multicast traffic but also to increase the

\* Author to whom correspondence should be addressed.  
Email: masdan@utu.fi

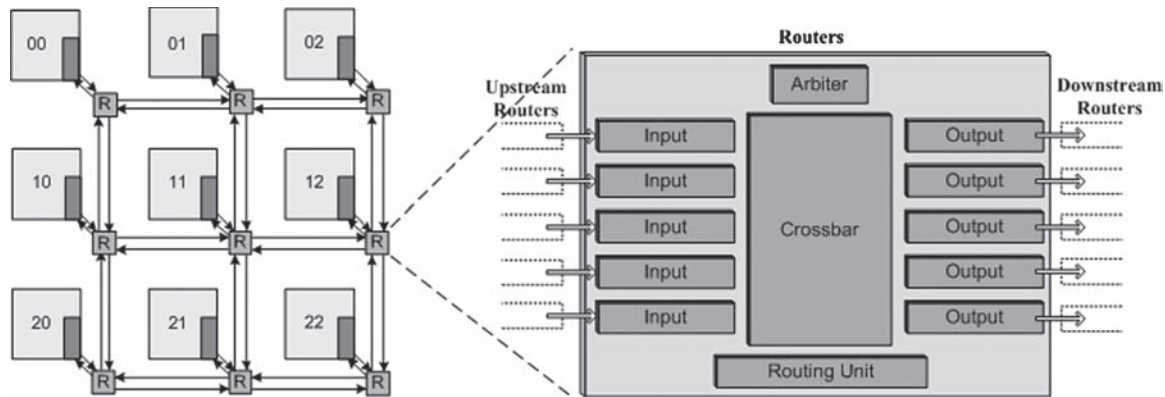


Fig. 1. Mesh topology based on tile.

alternative paths by introducing non-minimal paths which can mitigate congestion links.

When multiple input ports competing for the same output port, the input selection might consider the congestion level of the upstream routers. Giving busier input port higher priority to access the output port to keep the traffic in busy paths flowing, increases the possibility of the starvation. Thus, for the input-selection method we adopt the Weighted Round Robin (WRR) arbitration mechanism which serves each input port according to its congestion level. Exploiting this technique helps to avoid possible network congestion.

To evaluate AIOS, we compare it with the other router schemes under several synthetic traffic profiles along with Video Object Plane Decoder (VOPD), i.e., an example of a real traffic profile.

The paper is organized as follows. In Section 2, a brief review of previous works and background is presented while the implementation of minimal and non-minimal routing model for unicast and multicast traffic is described in Section 3. The proposed router architecture is presented in Section 4 and the experimental results are discussed in Section 5. Finally the summary and conclusion are given in the last section.

## 2. PREVIOUS WORK AND BACKGROUND

In this section, we review the previous works on adaptive input-selection and output-selection methods.

### 2.1. Input-Selection Methods

Choosing one of input channels to get access to the output channel is performed by an input-selection method using an arbitration mechanism. The arbiter could follow either non-priority or priority scheme.<sup>7-9</sup> In the non-priority scheme when there are multiple input port requests for the same available output port, the arbiter does not consider the traffic condition of the input channels to grant access to one input port. First-Come-First-Served (FCFS)<sup>7,10</sup> and Round-Robin (RR)<sup>7,8</sup> are two approaches

using non-priority arbitration. Therefore, these methods can avoid starvation on different ports. Despite the non-priority scheme, in the priority scheme when there are multiple input port requests for the same available output port, the arbiter would grant access to the input port request which has the highest priority level, e.g., Contention-Aware Input Selection (CAIS).<sup>9</sup> In CAIS, the busiest input channel obtains the highest priority to access the output channel. The input channel is given priority proportional to the number of requests arrived from the upstream routers. Thus, the traffic can be kept flowing in busy channels to avoid the network congestion. This scheme increases the possibility of the starvation so that in this paper, we have presented an efficient arbitration mechanism using WRR.<sup>11</sup> The presented input-selection method is able to avoid starvation while serving each input channel according to its traffic condition.

### 2.2. Output-Selection Methods

The routing algorithms, employed by the output-selection method, are classified as deterministic or adaptive.<sup>12,13</sup> In deterministic routing models, the path between a source and a destination is determined by the source and destination nodes positions without considering the traffic status of the network. *XY* is a deterministic wormhole routing algorithm that is deadlock and livelock free.<sup>12-15</sup> In this routing algorithm, each packet first travels along the *X* and then the *Y* direction to reach the destination. In adaptive algorithms, however, the path between a source and a destination is determined node by node depending on the network status as packets move toward the destination. That is, the output-selection method utilizes adaptive routing algorithms based on the congestion condition of the neighboring routers.<sup>7,16</sup> This causes packets to be forwarded to routers with lower traffic load. The adaptive nature of this type of routing algorithms makes them very attractive.<sup>13</sup> The Odd-Even turn model is an adaptive routing algorithm based on the turn model.<sup>14,15</sup> To avoid deadlock, Odd-Even restricts the position where turns are allowed in the mesh topology. Another adaptive routing

algorithm, named DyAD is introduced in Ref. [7]. This algorithm is a combination of a static routing algorithm called oe-fix, and the adaptive Odd-Even turn model. Depending on the congestion condition of neighboring routers, one of the routing algorithms is chosen.

Hot-potato (or deflection routing)<sup>10,17</sup> is based on the idea of delivering a packet to an output channel at each cycle. If all the minimal path channels are occupied, then the packet is misrouted. In the hot-potato routing, if the number of input channels is equal to the number of output channels at every router, packets can always find an exit channel. In fact, this method is deadlock free in virtual cut through switching networks and cannot be used in wormhole switching. However, livelock is a potential problem in this routing which increases message latency in high traffic loads considerably. Accordingly, performance of hot-potato is not as efficient as other adaptive routing algorithms.<sup>12</sup> All the aforementioned adaptive routing algorithms are utilized without using virtual channels. However, virtual channels can be employed to gain performance.

Communication in network-based MPSoCs can be either unicast (one-to-one) or multicast (one-to-many).<sup>18,19</sup> In unicast communication, a message is sent from a source node to a single destination node, while in multicast communication a message is sent from a source node to an arbitrary set of destinations. Multicast communication is employed in many MPSoC applications, e.g., replication, barrier synchronization, cache coherency in distributed shared-memory architectures, and clock synchronization.<sup>19,20</sup> Although multicast communication can be implemented by multiple unicast messages, this alternative method produces too much unnecessary traffic and probably increases the latency and congestion in the network.<sup>5,18</sup> Multicast routing algorithms can be classified as unicast-based, tree-based, and path-based.<sup>5</sup> However, despite the path-based method may force packets to follow longer paths under low to medium traffic load, it is more efficient than the others under high traffic load.<sup>6</sup> If turn model algorithms are adopted to route multicast packets, some forbidden turns might occur. More precisely, after reaching an intermediate destination the message header is routed toward the next destination. As a result, the packet must be forwarded toward its next destination regarding the turn model rules. However, there might be a dependency between the input ports (in which the packet is arrived from the previous destination) to output channel (in which the packet is forced to take to the next destination). In other words, packets obey the turn model rules when travelling between two consequent destinations, while occasionally they have to break the rules when reached a destination and forced to select certain output channels for the next destination. To cope with the forbidden turns the absorb-and-retransmission mechanism can be used.<sup>5,21</sup> In this method, the packet is absorbed by the

router when arriving to a destination and then retransmitted to the next destination. However this technique degrades the performance significantly. In Ref. [20] authors utilized the Odd-Even routing algorithm to route multicast packets. The more frequently forbidden turns occur, the more performance is degraded. Accordingly, the Hamiltonian Adaptive Multicast and Unicast Model (HAMUM), explained in the following subsection, has been recently proposed to support both unicast and multicast traffic adaptively.<sup>18</sup> The adaptivity of HAMUM is identical to the adaptivity of Odd-Even for the unicast traffic while for the multicast traffic the adaptivity of HAMUM is higher than conventional multicast routing algorithms.<sup>18</sup> Hence, HAMUM is exploited as the output-selection method for AIOS because it provides adaptivity for both unicast and multicast traffic efficiently.

### 2.3. The HAMUM Routing Model

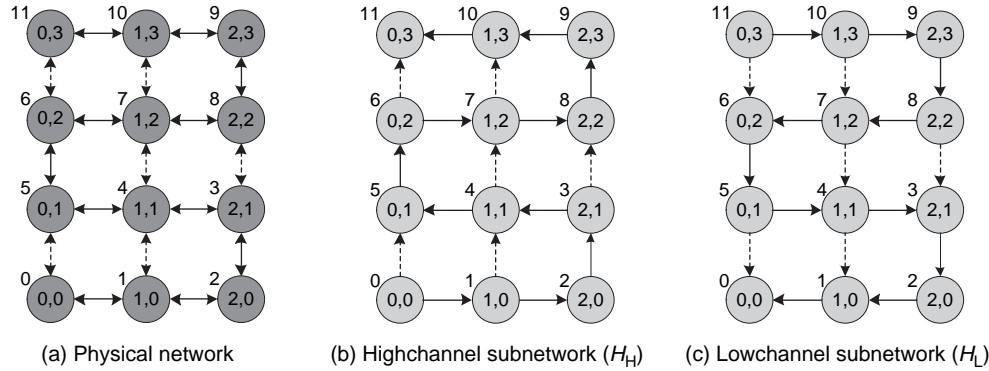
The path-based routing algorithm is established as the Hamiltonian path strategy. In the Hamiltonian path-based approach, every node in a graph is visited exactly once.<sup>5</sup> In this strategy, each node is assigned a label as follows (Fig. 2(a)):

$$L(x, y) = \begin{cases} y \times n + x & \text{if } y \text{ is even} \\ y \times n + n - x - 1 & \text{if } y \text{ is odd} \end{cases}$$

where  $x$  and  $y$  are node's coordinates and  $n$  is the number of nodes in the network.

As exhibited in Figures 2(b) and (c), two directed Hamiltonian paths are constructed by labeling. The high channel subnetwork ( $H_H$ ) starts at node 0, and the low channel subnetwork ( $H_L$ ) ends at node 0. In case the label of the destination node is greater than the label of the source node, the routing always takes place in the  $H_H$  subnetwork; otherwise it takes place in the  $H_L$  subnetwork. The destinations are placed into two groups. One group contains all the destinations that could be reached using the  $H_H$  subnetwork, and the other contains the remaining destinations that could be reached using the  $H_L$  subnetwork. To reduce the path length, the vertical channels that are not part of the Hamiltonian path (the dashed lines in Fig. 2) could be used in appropriate directions.

HAMUM is based on the Hamiltonian path in 2D mesh networks-on-chip with wormhole switching technique.<sup>18</sup> In the conventional path-based routing models, such as Multi-Path (MP) and Column-Path (CP) algorithms,<sup>5</sup> the unicast and multicast messages are routed using deterministic routing algorithms, degrading the network performance. These path-based routing algorithms can be replaced by HAMUM, a minimal adaptive scheme to route both unicast and multicast messages through the network. To break all cycles in HAMUM, similar to the Odd-Even model, the locations at where certain turns can be taken are restricted, so that deadlock is avoided.



**Fig. 2.** (a) 3 × 4 mesh physical network with the label assignment and the corresponding (b) high channel and (c) low channel subnetworks. The solid lines indicate the Hamiltonian path, and dashed lines indicate the links that could be used to reduce path length.

The rules regulating HAMUM are categorized in the high channel subnetwork and low channel subnetwork as follows:

*For the high channel subnetwork:*

*North-East bound packets:* North direction is allowed to be taken in all intermediate nodes. East direction can be used by the nodes in even rows.

*North-West bound packets:* North direction is allowed to be taken in all intermediate nodes. West direction can be used by the nodes in odd rows.

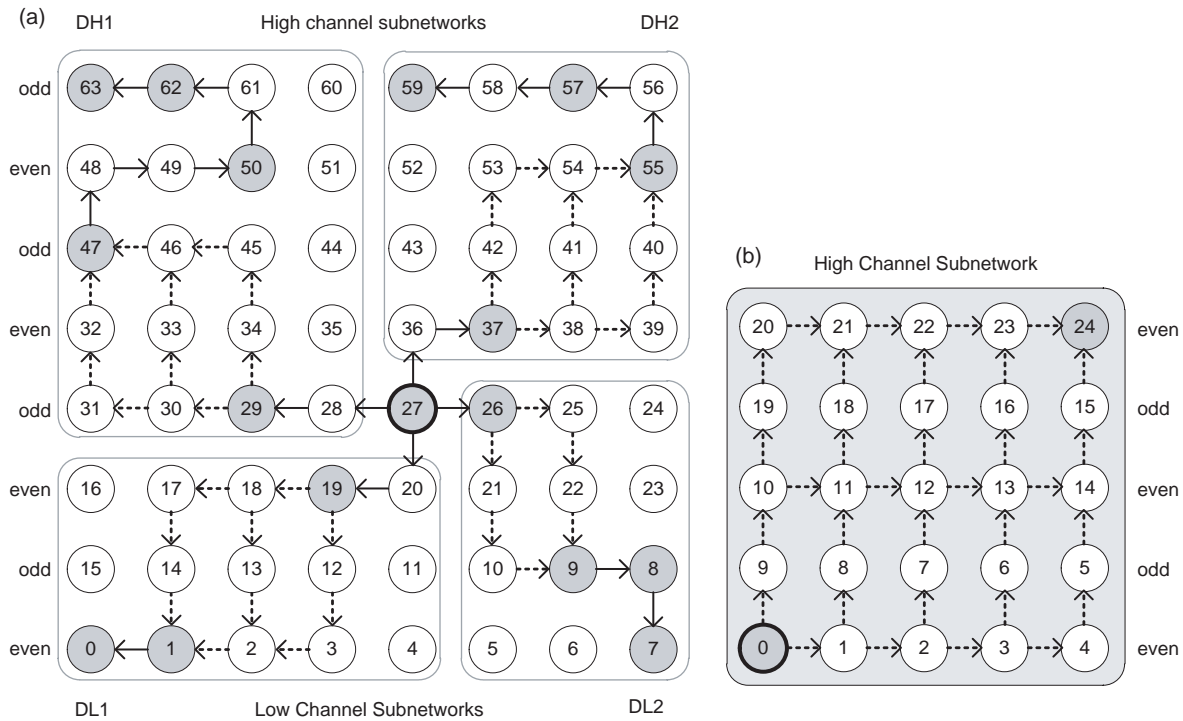
*For the low channel subnetwork:*

*South-East bound packets:* South direction is allowed to be taken in all intermediate nodes. East direction can only be used by the nodes in odd rows.

*South-West bound packets:* South direction is allowed to be taken in all intermediate nodes. West direction can only be used by the nodes in even rows.

Notice that a message will be forwarded to the destination as in the deterministic Hamiltonian strategy, when the current node is located one row to in the south (north) of the destination row in the high channel subnetwork (low channel subnetwork). Inasmuch as the rules keep the messages traveling through the Hamiltonian paths, it prevents the occurrence of deadlock.

Figure 3(a) shows how HAMUM brings adaptivity to the MP method. In the MP routing algorithm, the destination set is partitioned into two subsets,  $D_H$  and  $D_L$ , where every node in  $D_H$  has a higher label than that of the source node and every node in  $D_L$  has a lower label

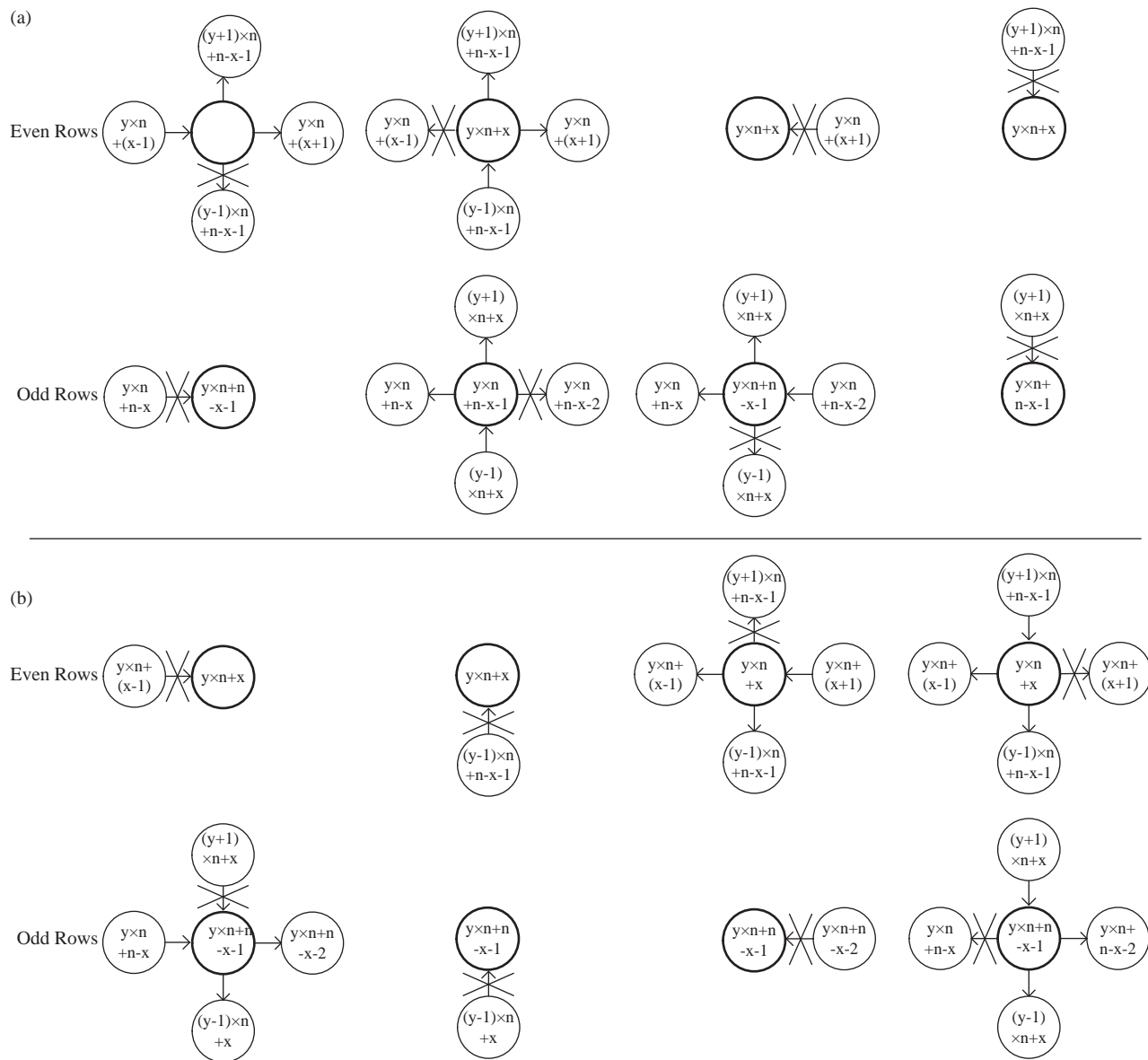


**Fig. 3.** Examples of (a) multicast aspect and (b) unicast aspect of HAMUM.

than the source node. Thus, multicast messages from the source node are sent to the destination nodes in  $D_H$  using the high channel subnetwork and to the destination nodes in  $D_L$  using the low channel subnetwork.<sup>5, 18</sup> To reduce the path lengths,  $D_H$  and  $D_L$  are also partitioned. The set  $D_H$  is divided into two subsets. When the source node is located in the even row (in the odd row), one subset consists of the nodes whose  $X$  coordinates are greater than (greater than or equal to) that of the source node and the other subset contains the remaining nodes in  $D_H$ . The set  $D_L$  is partitioned in a similar way. Hence, all destinations of a multicast message are grouped into four disjoint subsets. Consider the example in figure where node 27 sends its multicast message to destinations  $D = \{0, 1, 7, 8, 9, 19,$

$26, 29, 37, 47, 50, 55, 57, 59, 62,$  and  $63\}$ . As exhibited in Figure 3(a),  $D_H$  is divided into two subsets, which are  $D_{H1} = \{29, 47, 50, 62, 63\}$  and  $D_{H2} = \{37, 55, 57, 59\}$ . In the same way  $D_L$  is divided into two subsets, with  $D_{L1} = \{19, 1, 0\}$  and  $D_{L2} = \{26, 9, 8, 7\}$ . Finally, one packet per subset should be created and sent from the source node to the corresponding subnetwork. All packets must follow the Hamiltonian path and reach to destinations in the order they are arranged.

Using HAMUM multiple paths can be taken by packets in the network. For example, the packet delivered to subnetwork  $D_{H2}$  can be forwarded in three different ways from the node 37 to the node 55 (similar cases between nodes 29 to 47, 19 to 1, and 26 to 9). Figure 3(b) exhibits all



**Fig. 4.** All possible turns in the Enhanced HAMUM. (a) high channel subnetwork (b) low channel subnetwork. Note that the numbers indicated the labels of the previous, current and next routers.



possible minimal routing paths of a unicast packet where the packet is sent from the node 0 to the node 24 in the high channel subnetwork.

As mentioned earlier, the output-selection method of AIOS uses either the minimal or non-minimal path

depending on the congestion condition of the neighboring routers. Inasmuch as HAMUM is a minimal routing algorithm, in this paper, we extend this routing model in order to support the non-minimal paths as well as the minimal paths.

```

Algorithm Enhanced_HAMUM is
--(Cx,Cy): Current node, (Dx,Dy): Destination ode
MinPath1 = NULL; MinPath2 = NULL; NonminPath = NULL;
Begin
If (Dy = Cy) then                                --Current & Dest. are in the same row
If (Dx = Cx) then                                --Current & Dest. are in the same column
    MinPath1 <= Local;                            --Packet sends to the Local direction
Elsif (Dx > Cx) then
    MinPath1 <= East;                             --Dest. is to the East of the Current node
Else
    MinPath1 <= West;                             --Dest. is to the West of the Current node
End if;
Elsif (Dy > Cy) then                               --high channel Subnetwork
    If (Cy mod 2 = 0) then                         --rule1 in the even rows
        If (Dx > Cx) and (Dy - Cy = 1) then      --Dest. is in the East & 1 row to the Current node
            MinPath1 <= East;
Elsif (Dx > Cx) and (Dy - Cy > 1) then            --Dest. is in the East of the Current node
            MinPath1 <= East;
            MinPath2 <= North;
Else
            MinPath1 <= North;                   --Dest. is in the West of the Current node
NonminPath <= East;
End if;
Elsif (Cy mod 2 /= 0) then                         --rule2 in odd rows
        If (Dx < Cx) and (Dy - Cy = 1) then      --Dest. is in the West & 1 row to the Current node
            MinPath1 <= West;
Elsif (Dx < Cx) and (Dy - Cy > 1) then            --Dest. is in the West of the Current node
            MinPath1 <= West;
            MinPath2 <= North;
Else --Dest. is in the East of the Current node
            MinPath1 <= North;
NonminPath <= West;
End if;
End if;
Elsif (Dy < Cy) then                               --low channel Subnetwork
    If (Cy mod 2 = 0) then                         --rule1 in even rows
        If (Dx < Cx) and (Cy - Dy = 1) then      --Dest. is in the West & 1 row to the Current node
            MinPath1 <= West;
Elsif (Dx < Cx) and (Cy - Dy > 1) then            --Dest. is in the West of the Current node
            MinPath1 <= West;
            MinPath2 <= South;
Else
            MinPath1 <= South;                   --Dest. is in the East of the Current node
NonminPath <= West;
End If;
Elsif (Cy mod 2 /= 0) then                         --rule2 in odd rows
        If (Dx > Cx) and (Cy - Dy = 1) then      --Dest. is in the East & 1 row to the Current node
            MinPath1 <= East;
Elsif (Dx > Cx) and (Cy - Dy > 1) then            --Dest. is in the East of the Current node
            MinPath1 <= East;
            MinPath2 <= South;
            --Two minimal paths are suggested
Else
            MinPath1 <= South;                   --Dest. is in the West of the Current node
NonminPath <= East;
End if;
End if;
End Enhanced_HAMUM;

```

Fig. 5. The pseudo VHDL code of the Enhanced HAMUM.

### 3. ENHANCED HAMUM

HAMUM can be extended to support the non-minimal path routing in the network, we call this method Enhanced HAMUM. All allowable turns that can be taken by a packet in Enhanced HAMUM are illustrated in Figure 4. According to the labeling mechanism described in Section 3.3, in  $n \times m$  mesh network, the label of a node in an even or odd row is given by  $(y \times n) + (x)$  and  $(y \times n) + (n - x - 1)$ , respectively. Therefore, if a node is located in an even row, the labels of its neighboring nodes in east, north, west, and south directions are  $(y \times n) + (x + 1)$ ,  $((y + 1) \times n) + (n - x - 1)$ ,  $(y \times n) + (x - 1)$ , and  $((y + 1) \times n) + (n - x - 1)$ , respectively. Similar approach is used for the nodes in odd rows. In the Hamiltonian path, packets should be routed entirely in ascending (in the high channel subnetwork) or descending order (in the low channel subnetwork), so that in the high channel subnetwork and in the even row (Fig. 4(a)), the arrived packet from the west direction can be routed to the nodes with the larger labeling numbers (i.e., neighboring nodes in the north and east directions). In sum, in the high channel subnetwork, the arrived packets from south or west (south or east) direction can be transmitted minimally or non-minimally to the north or east (north or west) direction in even (odd) rows. In contrast, in the low channel subnetwork, the packets can only be received from north or east (north or west) direction in even (odd) rows; and they can be delivered to south or west (south or east) direction in even (odd) rows.

Figure 5 depicts the implementation of the Enhanced HAMUM. Once the presented algorithm is performed, three output variables, MinPath1, MinPath2, and NonMinPath, are evaluated. The variables of MinPath1 and MinPath2 are the minimal directions that can be chosen by a packet while the NonMinPath indicates the allowable non-minimal direction. For example if the source node is located in the even row and the destination node is in the northeast position of the source node, two minimal directions (i.e., east and north) are suggested by the algorithm; while in the similar case, if the source node is located in the odd row, one minimal direction (i.e., north) and one non-minimal direction (i.e., west) are supplied by the algorithm.

Two examples of the Enhanced HAMUM, using minimal and non-minimal directions, are shown in Figure 6. In the first example, the source node 1 sends a message to destination 23 while the nodes 11 and 18 are faulty or congested. The Enhanced HAMUM allows the packet to route around the congested areas by selecting the non-minimal path at node 8. As another example in Figure 6, a packet can turn around the congested region when traveling from the source node 2 to the destination 16.

#### 3.1. Communication Channel Deadlock Avoidance

Deadlock is a situation where network resources continuously wait for each other to be released. To show that

the proposed algorithm are deadlock free, it is required to prove that there is no cyclic dependency between channels.<sup>16,22</sup>

Now we prove that Enhanced HAMUM is deadlock free. At the source node, the network is divided into two disjoint subnetworks, High channel ( $H_H$ ) and Low channel ( $H_L$ ). Since each of the high channel and low channel subnetworks uses separate sets of channels, no cyclic dependency will be created among channels. If we could prove that the message routing algorithm in the high channel subnetwork is deadlock free, that would be sufficient to establish that the low channel subnetwork is also deadlock free, and since  $H_H \cap H_L = \Phi$ , the whole network will be free of deadlocks. So, we take the high channel subnetwork into consideration.

The multicast message can be represented by Multicast =  $(u, D)$ , where  $u \in V$  is the source node,  $D = \{d1, d2, \dots, dx\}$  is the set of ordered destination nodes, and  $x$  is the number of destination nodes. Each node in the graph has a label (L) determined by the Hamiltonian path labeling mechanism. Since a unicast message is the special case of a multicast message, we prove that the algorithm is deadlock free for the multicast messages, and then it is obvious for the unicast messages.

To show that the Enhanced HAMUM routes every packet in strictly increasing order in the high channel subnetwork, it is sufficient to show that for an arbitrary router, the label of the upstream router is smaller than the current router, and the label of the current router has lower value than the downstream router. All possible turns between input and output channels in Enhanced HAMUM along with the node labels are illustrated in Figure 4. Examination shows that all allowable turns take place in ascending order. On top of that, in the high channel subnetwork, the destination nodes are ordered in ascending order in the header of the packet,

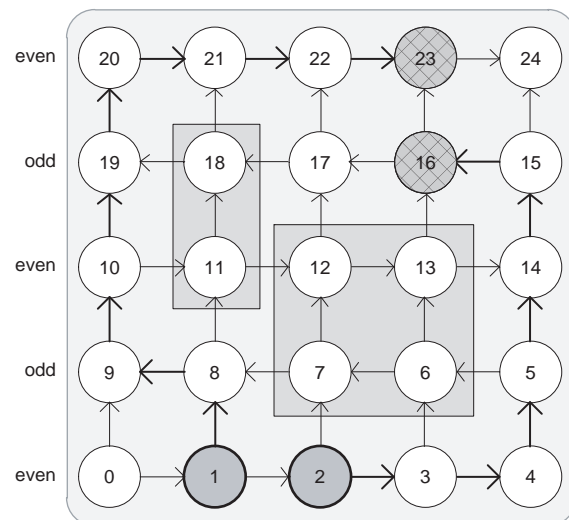


Fig. 6. An example of the enhanced HAMUM.

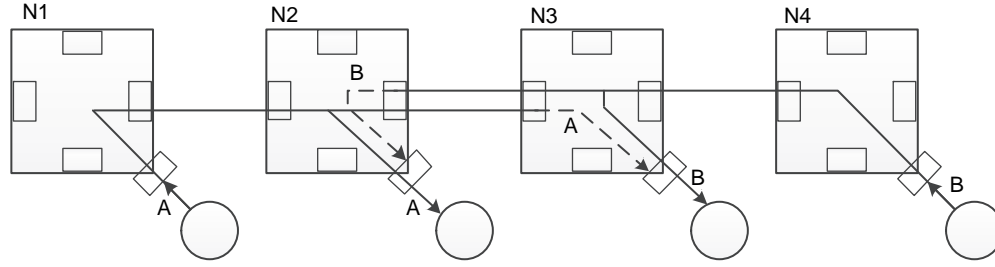


Fig. 7. Deadlock formed by using a single delivery channel.

$L(u) < L(d1) < L(d2) < \dots < L(dx)$ . If we suppose that a minimal or non-minimal multicast message path is Path  $(u, D) = (u, a_1, a_2, \dots, a_x, d_1, a_{x+1}, a_{x+2}, \dots, a_y, d_2, a_{y+1}, a_{y+2}, \dots, a_z, d_x)$ , then intermediate nodes (either in the minimal or non-minimal paths) are selected in a way that the packet follows the path only in ascending order, so:

$$\begin{aligned} L(v_0) \leq L(u) \leq L(a_1) \leq L(a_2) \leq \dots \leq L(a_x) < L(d1) \\ \leq L(a_{x+1}) \leq L(a_{x+2}) \leq \dots \leq L(a_y) < L(d2) \leq L(a_{y+1}) \\ \leq L(a_{y+2}) \leq \dots \leq L(a_z) < L(dx) \leq L(v_{n-1}) \end{aligned}$$

Note that the Hamiltonian path guarantees the existence of at least one possible path between each pair of nodes. According to the above facts, there cannot exist any link like  $(a_i, a_{i+1})$ , where  $L(a_i) > L(a_{i+1})$ , so no cyclic dependency can occur between channels. Moreover, all unicast and multicast packets in the high channel subnetwork are routed in entirely ascending order, thus the traveled paths of all packets cannot create any dependency cycles. The similar proof can be applied to the low channel subnetwork.

### 3.2. Consumption Channel Deadlock Avoidance

In the path-based multicast routing deadlock can be avoided by transmitting messages through intermediate nodes in such a way that packets follow a path either in ascending or descending order. However, deadlock is still possible because messages transmitted in the high channel and low channel subnetworks use the same delivery channels at every destination node.<sup>12</sup>

Figure 7 shows a deadlock configuration in which two multicast messages A and B are destined for nodes  $N_2$  and  $N_3$ . Packets A and B are traveling in the high channel and low channel subnetworks, respectively. Assuming that  $L(N_1) < L(N_4)$ , packet A first reached node  $N_2$ , then node  $N_3$ . Also, packet B first reached node  $N_3$ , then node  $N_2$ . As there is a single delivery channel at each node, each packet has reserved one consumption channel and is waiting for the other consumption channel to become free. In this example, packet A acquires consumption channel  $N_2$  and communication channel  $(N_2, N_3)$  while waiting for consumption channel  $N_3$ . Packet B acquires consumption channel  $N_3$  and communication channel  $(N_3, N_2)$

while waiting for consumption channel  $N_2$ . This cyclic wait creates a deadlock. In general, deadlocks may arise because messages traveling in the high channel and low channel subnetworks share the same delivery channels, thus producing cyclic dependencies between them. This cyclic dependency can be easily broken by using different delivery channels for each subnetwork. In this case, two delivery channels at each node are enough to avoid deadlocks.<sup>5,12</sup>

## 4. THE AIOS ROUTER ARCHITECTURE

The idea of the proposed router architecture is based on spreading the traffic to prevent congestion (remove hot-spots). Using adaptive input-selection and output-selection methods can improve the network performance significantly and reduce the maximum power consumption of the network. The output-selection method utilizes both the minimal and non-minimal schemes of HAMUM. When congestion (hotspot) is formed close to a router, a non-minimal direction may be selected to deliver a packet while a minimal direction is taken when there is no congestion. In AIOS, the input-selection exploits the Weighted

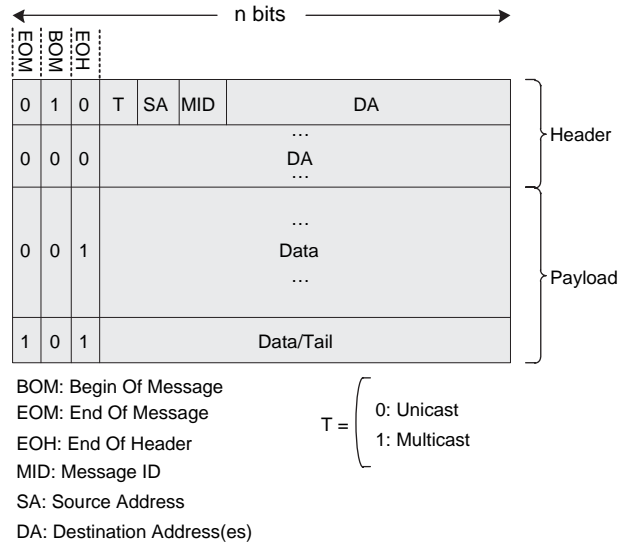


Fig. 8. Message format.



Round Robin (WRR) policy which makes the routing algorithm non vulnerable to starvation. Also, WRR increases the performance of the network by monitoring the traffic condition.

### 4.1. Switching Method

In this work, we consider a  $n \times n$  network of interconnected tiles with a mesh topology.<sup>13, 22, 23</sup> Each tile is composed of a PE (Processing Element) and a router in which the router is connected to its four adjacent routers in addition to the PE of the tile through channels. Each channel consists of two unidirectional point-to-point links. To have pipelined flow of messages and small buffers, we use wormhole scheme for the switching.<sup>13</sup> In this method, a packet is divided into smaller segments called FLITs (FLow control digIT) which are routed successively until they reach their destination.

### 4.2. Message Format

The header of multicast messages must carry the addresses of the destination nodes. Two famous encoding schemes are all-destination encoding and bit string encoding.<sup>24</sup> The all-destination encoding is a simple scheme in which all destination addresses are carried by the header. This scheme is efficient for a small number of addresses because the header length is proportional to the number of addresses. However, it produces an overhead when the number of destinations is large. One way to limit the size of the header is to encode destination addresses as a bit string, where each bit corresponds to a destination. This bit string encoding scheme is efficient when the average number of destinations is large. However, it is inefficient when the system is large and the number of destinations is small.<sup>12</sup>

In this paper, the packet format is based on the all-destination encoding method. The message format is

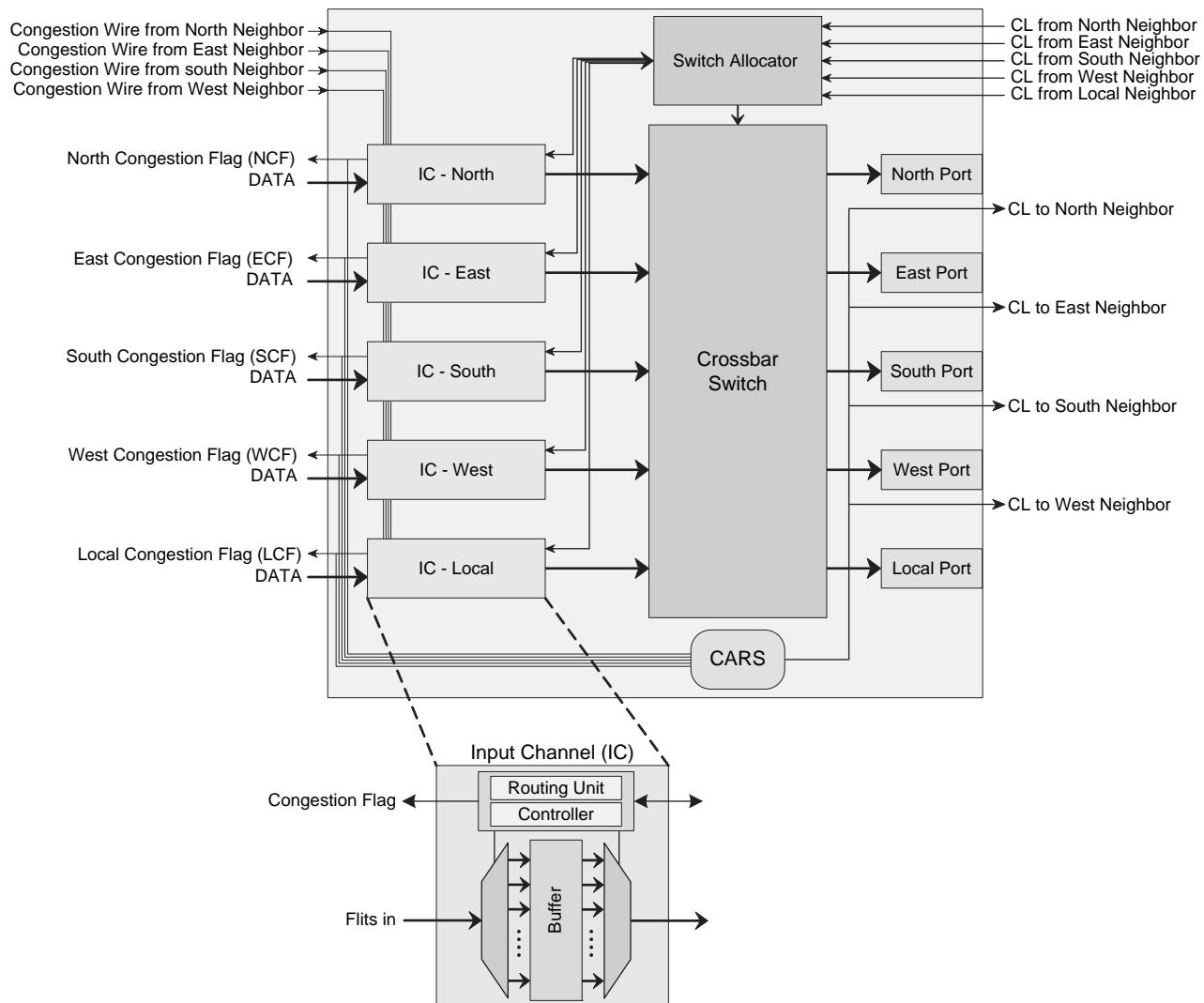


Fig. 9. The proposed routing structure.

shown in Figure 8; it includes a header flit(s) and a parametric number of payload flits. Each flit is  $n$ -bit wide and the  $n$ th bit is the EoM (End of Message) sign, the  $(n - 1)$ th bit is the BoM (Begin of Message) sign, and  $(n - 2)$ th bit is the EoH (End of Header) sign. In the header flit(s), the fourth field,  $T$ , is used to describe the type of the message. There are two types of message: unicast ( $T = 0$ ) and multicast ( $T = 1$ ). The specific addresses of the source node and the destination node(s) are placed in the last field of the header(s) in a row and the content of the message is located in the rest of flits (Payload). Moreover, the message identifier (MID) is used for the message ordering.

**4.3. Router Structure**

As shown in Figure 9, each input channel has a routing unit, a controller for handshaking and an input buffer. The flits of the packets are stored in the input buffer. The routing unit determines the output channel to route packets. The controller controls the buffer status including empty and full states as well as detects the sign of the rate at which the buffer is becoming occupied. A positive rate indicates that the buffer is filling up while a negative rate reveals that the buffer is draining. Each input channel has a Congestion Flag (CF) signal (i.e., ECF, WCF, NCF, SCF and LCF corresponding to East, West, North, South and Local input channel, respectively) to inform its adjacent routers about its congestion condition so that the congested input channel should not be selected by the upstream router until the congestion condition is ceased.

The router has a crossbar to establish a connection path from an input port to an output port. For each output port the router uses an arbiter for selecting among simultaneous input requests to access the same output port. In order to detect whether the buffer status is critical or not, the arrival and departure rates of the buffer should be measured. For this purpose, the circuit shown in Figure 10 is used.  $N_{new}$  is the number of occupied slots of the input buffer in the current cycle of the router clock and  $N_{old}$  is the same number but in the previous cycle of the router clock. To determine the rate at which the buffer becomes full, the number of filled buffer cells at each rising edge of the router internal clock ( $N_{new}$ ) is compared to that of the

previous rising edge ( $N_{old}$ ). If  $N_{new} > N_{old}$  ( $N_{old} > N_{new}$ ), it shows that the buffer is filling up (draining). The sign is compared to the buffer status to activate the CF.

The status signal of the buffer becomes full when the number of occupied cells of the buffer is more than a threshold value. In this case, for warning the full status, the signal  $W\_Full$  is activated indicating that most buffer cells are full. This suggests that the congestion condition is traced using the signal  $W\_Full$  to indicate the filling of the buffer. As shown in Figure 10, CF is asserted when both the  $W\_Full$  signal and the positive rate for occupying the input buffer slots ( $N_{new} > N_{old}$ ) are detected. The Congestion Level (CL) of each router is computed by a module called Contention Aware Routing Selection (CARS). The CL is a 3-bit binary number as a result of summing up four CF values from four input ports (see Figs. 9 and 11). The CL for each router indicates its load level. For example, if the north and east input buffers of the router are congested ( $NCF = 1$  and  $ECF = 1$ ), then the CL value of the router will be equal to “010”. As illustrated in Figure 11, the output of the CARS module is sent to the corresponding input channels of its adjacent routers (downstream routers).

**4.3.1. Output-Selection**

In the output-selection method, each input channel has a routing unit decoding the header flit of packets coming from an input port. The modified HAMUM, based on the minimal and non-minimal paths, is used to determine the output port to deliver packets. If the route(s) determined from the minimal path routing is(are) congested, the routing unit uses instead the non-minimal path.

First, based on the modified HAMUM in Figure 5, the output port(s) specified by the minimal path (MinPath1 and MinPath2) are examined and if the congestion flag of the neighboring routers of the selected output ports is active, the congestion condition of the non-minimal path is checked. If the non-minimal direction is not congested, the packet is sent to the output port determined by the non-minimal path (NonMinPath). If the neighboring routers are not congested, the packet will be sent through the first minimal path output port (MinPath1). Figure 12 shows the address decoder circuit.

The procedure of selecting the suitable output port among all output ports that have been specified by the

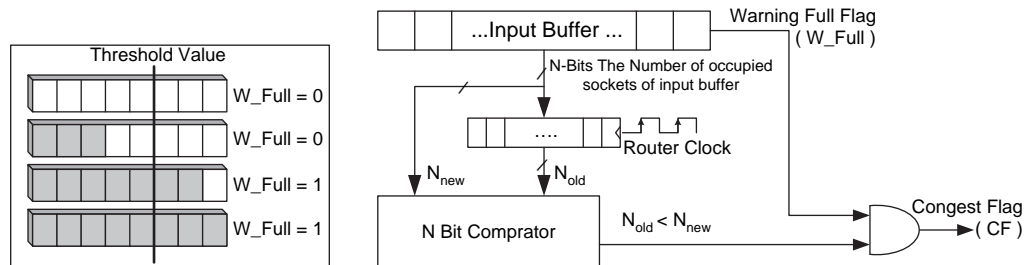


Fig. 10. Congestion detection circuit for the input buffer.

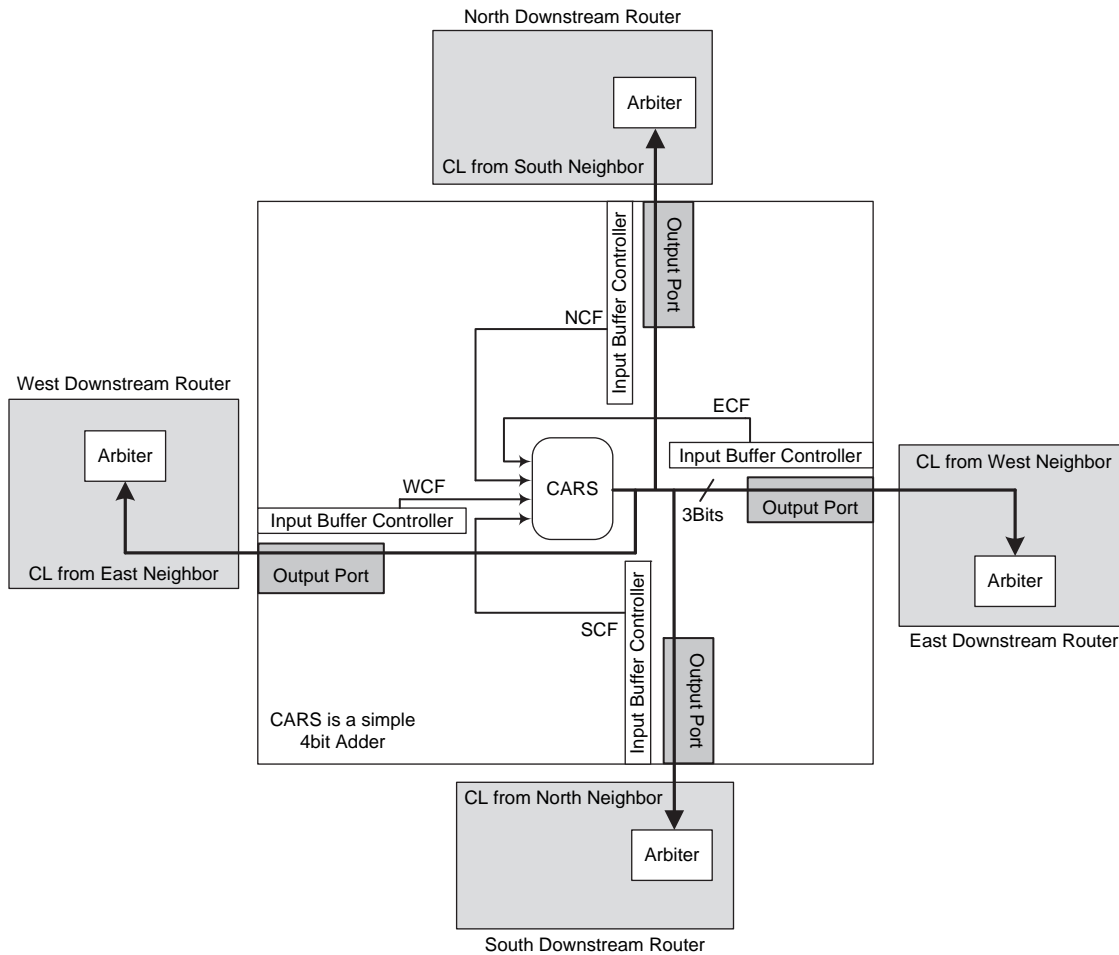


Fig. 11. Congestion level computation and transmission scheme.

routing unit (Fig. 5) is exhibited in Figure 13. In fact, the routing unit chooses the direction in which the corresponding downstream router has not raised its congestion flag. For instance, if a packet with a given source and destination could be routed to both output ports p1 (CF = 1) and p2 (CF = 0), then it will be routed to p2. If p1 and p2 happen to have both their congestion flag raised, if

the routing unit has specified a non-minimal path, p3, the packet will be routed to p3, otherwise it will be routed to p1. On the other hand, if both p1 and p2 are minimal output directions and the congestion flags of their corresponding downstream router have not risen, the routing unit will route the packet to p1 direction. If the header type is a multicast message, the routing unit fetches the

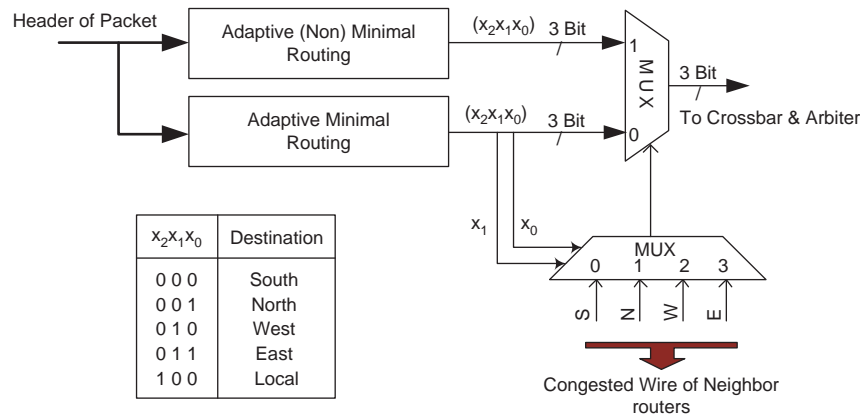


Fig. 12. Routing unit circuit.

```

If (MinPath1 /= NULL and CF(MinPath1) = '0') then
  Select <= MinPath1;
Elsif (MinPath2 /= NULL and CF(MinPath2) = '0') then
  Select <= MinPath2;
Elsif (NonminPath /= NULL and CF(NonminPath) = '0') then
  Select <= NonminPath;
Else
  Select <= MinPath1;
End if;

```

Fig. 13. The procedure of selecting the suitable output port.

destination address from the header. After fetching the destination address from the header, if the destination address is the current node, the routing unit will request the local output port. Meanwhile, the routing unit fetches the next destination address from the header and runs the adaptive routing procedure to determine the output port(s) corresponding to the next destination address.

#### 4.3.2. Input-Selection

The arbiter provides services for each input channel in turn in the Round Robin (RR) order. If the input channel buffer is empty, it will be skipped without being serviced. Figure 14 shows a block diagram of a round robin arbiter.<sup>25-27</sup> The arbiter uses a Programmable Priority Encoder (PPE) unit to choose one highest priority request from  $n$  incoming requests ( $Req$  bus). In every arbitration cycle, PPE, which takes  $n$  1-bit-wide requests and the  $\log n$ -bit-wide pointer ( $P\_enc$ ) pointing to the current highest-priority request as its inputs, chooses the first nonzero request value beyond (and including)  $Req[P\_enc]$ . The output of the PPE is an  $n$ -bit-wide  $Gnt$  (grant) which has at most one nonzero bit and a 1-bit wide  $anyGnt$  signal which indicates if there has been at least one request. For updating the pointer,  $Gnt$  is loaded and rotated right one bit in  $rr1$  unit (rotate right 1-bit register) whose output is encoded using the  $Enc$  unit and then latched for storing the next  $P\_enc$ .

The proposed arbiter uses the Weighted Round Robin (WRR) scheme derived from the round robin policy for the input-selection method. The presented scheme allows a weight to be assigned to each input port. The weight

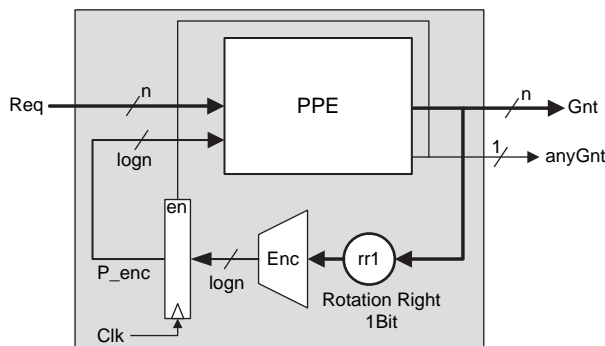


Fig. 14. Block diagram of a round-robin arbiter.

of each input port, which specifies the number of packets to be transmitted, is proportional to the CL of the upstream router. This will assign different weights to the input channels of the routers for accessing the output channels through the arbitration process. Figure 15 shows a block diagram of the Weighted Round Robin arbiter derived from the Round Robin scheme. The main difference between the two schemes is that WRR serves to the input port based on its CL. There are five registers four of which contain the CL of their upstream routers and one register is for the local router. The registers have three inputs and one output. If the register enable ( $En$ ) is set, then the new CL value, which shows the CL of the upstream router, will be loaded in the register. After loading, the register operates as the down-counter for the service provided for this input port. The register value is decremented in each packet transmission cycle until the register value reaches zero. Whenever the register value reaches zero or the register enable ( $En$ ) is reset, the signal  $Zero$  will be asserted and subsequently the signal  $Enable$  of the  $rr1$  unit is activated. Finally, similar to the RR scheme,  $P\_enc$  is updated. In the situations where there are multiple input requests to the same output channel, each output channel arbiter will service the incoming requests according to their CL (weight). This mechanism resolves any possible starvation that might occur in arbiters based on priority scheme such as in CAIS.<sup>9</sup> In the worst case, four input ports can request for an output port simultaneously and the granted input port can transmit at most four packets. Thus, an input port will receive a grant after 16 packet transmissions. However, since the congestion values of upstream routers are dynamically changed, the assigned weight to each input port is updated.

It is worth noting that as a result of using the adaptive routing algorithm, the messages of the same data may traverse different paths reaching at the destination out-of-order. Hence, a technique is required to reorder the packets at the destination. In the proposed technique in this work, the packets that reach the destination node have the information about the packet source node (SA) and the packet order (MID). Using the SA and MID, the destination core may store each packet in its proper location in the core memory such that the original source order can be achieved with negligible overhead. Note that the data in the memory might not be processed by the core unless all parts of the data are received. This is also true for deterministic multicast routing algorithms. Also, the use of the source address enables the destination to concurrently handle different data coming from different sources.

## 5. RESULTS AND DISCUSSION

To assess the efficiency of AIOS, four other routers, defined in Table I, are also implemented. We have developed a flit level event driven wormhole NoC simulator

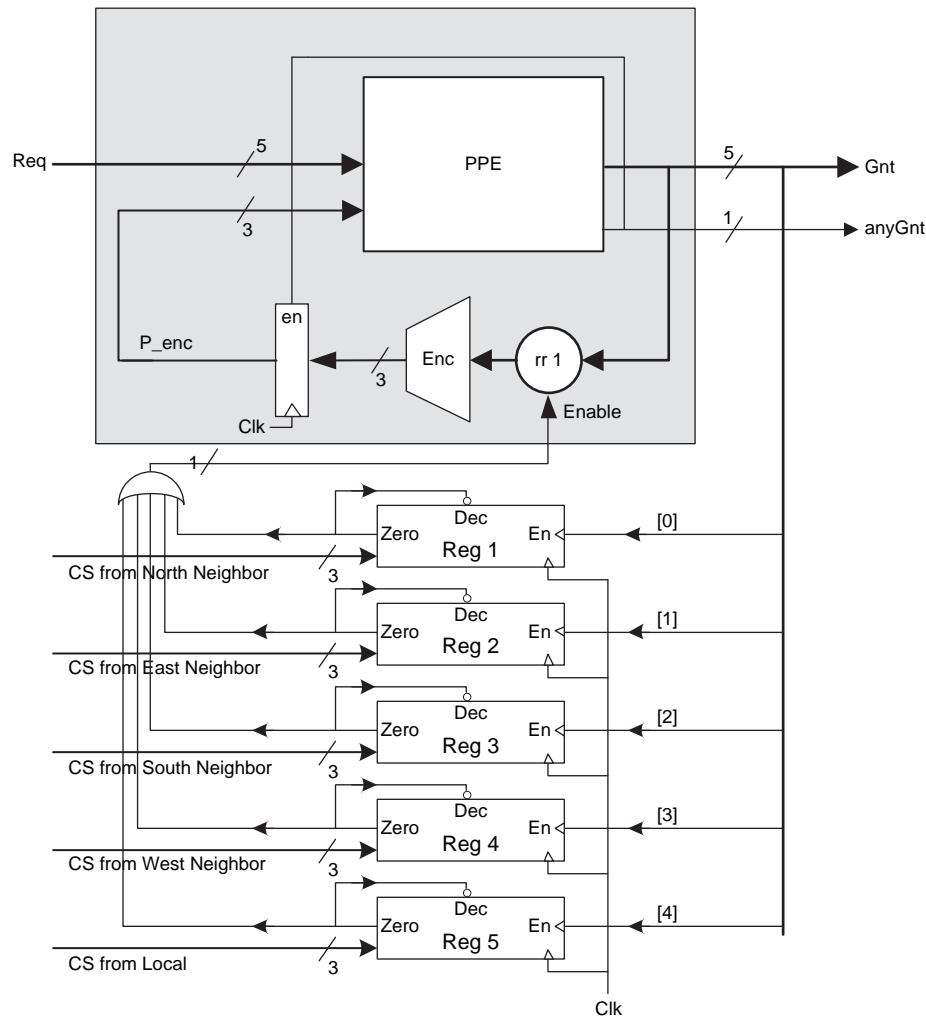


Fig. 15. Block diagram of a weighted round robin arbiter.

implemented with VHDL. The simulator computes the average latency and the power consumption for the packet transmission. As a performance metric, we use latency defined as the number of cycles between the initiation of a message operation issued by a PE and the time when the message is completely delivered to the destination PE. The request rate is defined as the ratio of the successful message injections into the network interface over the total number of injection attempts.

A two dimensional mesh configuration is used for the NoC. Each router consists of 8 unidirectional channels (four incoming and four outgoing channels). The simulator inputs include the array size, the router operation frequency, the input and output selection methods, the link width, and the traffic type.

Table I. Structure of other four routers.

Router	P-OE	P-MP	RR-OE	RR-MP
Input-Selection	Priority (CAIS)	Priority (CAIS)	Round Robin	Round Robin
Output-Selection	Odd-Even	Multi-Path	Odd-Even	Multi-Path

To estimate the power consumption, we use values reported by Orion library functions.<sup>28</sup> Since some components such as the routing unit and WRR circuits have not been modeled in Orion, we have modified the Orion library for computing their power consumptions. The data width and the frequency are set to 32 bits and 1 GHz, respectively, which leads to a bandwidth of 32 Gb/s. Each input channel has a buffer (FIFO) size of 8 flits with the congestion threshold set at 75% of the total buffer capacity. During the simulation, the packet length is variable and selected randomly from 5 to 25. The time needed to generate multicast messages is not considered, because we assumed the multicast messages are generated in the processing elements (PE). The array size of  $8 \times 8$  has been considered.

### 5.1. Performance Evaluation

#### 5.1.1. Multicast Traffic Profile

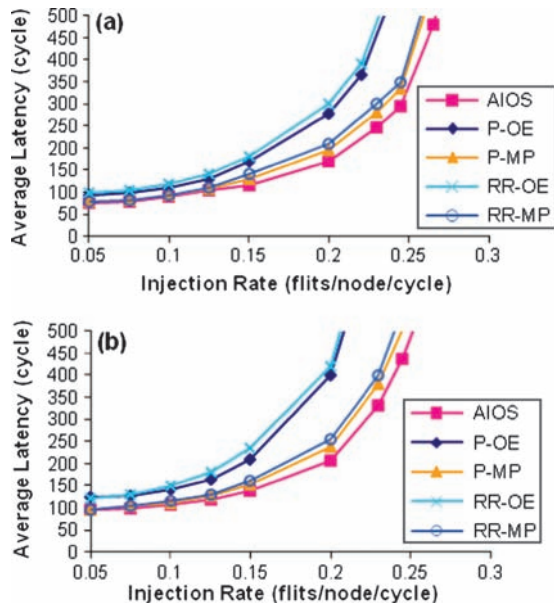
This simulation is performed using a uniform-based multicast traffic profile pattern. Each Processing Element (PE)



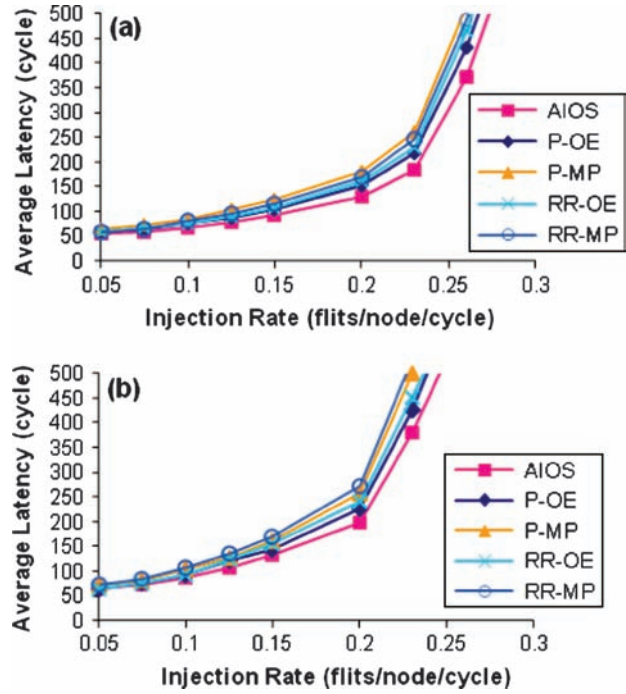
generates messages and injects them into the network using the time intervals which are obtained based on the exponential distribution. In the multicast traffic profile, each PE sends a message to a set of destinations. A uniform distribution is used to construct the destination set of each multicast message.<sup>5</sup> The number of destinations is set to 10 and 20. The average latency as a function of the average flit injection rate is shown in Figure 16(a) and (b). As shown in the results, AIOS leads to lower delay particularly not only in high traffic loads but also with more multicast destinations. As described before and can be seen from Figures 16(a) and (b), unicast-based routing algorithms, e.g., Odd-Even, are not efficient for multicast traffic.<sup>5,21</sup> The average latency of each network has been computed near saturation point (0.15). As a result, compared with the P-OE, P-MP, RR-OE, and RR-MP, the average latency of AIOS is reduced by 34%, 9%, 41%, and 15%, respectively.

**5.1.2. Unicast and Multicast (Mixed) Traffic Profile**

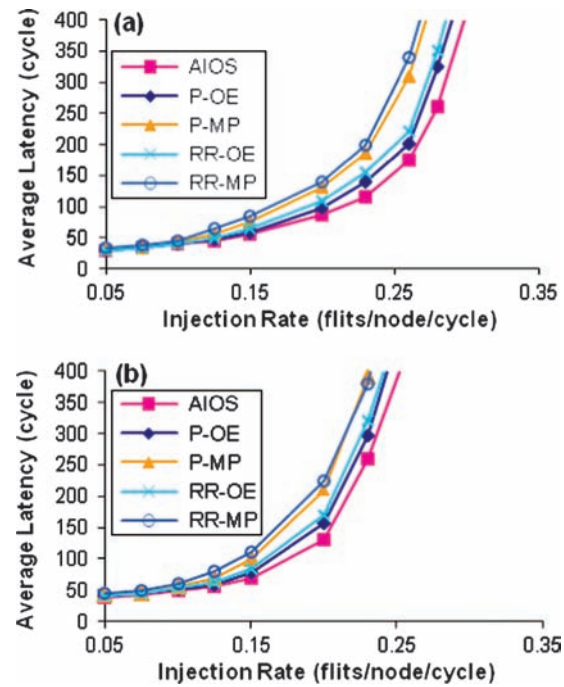
In this experiment, we employ a mixture of unicast and multicast traffic, where 80% of injected messages are unicast messages and the remaining 20% are multicast messages. This pattern may be representative of the traffic in a distributed shared-memory multiprocessor where updates and invalidation produce multicast messages and cache misses are served by unicast messages.<sup>5</sup> Both unicast and multicast messages are routed using HAMUM. The number of destination for multicast messages is set to 10 and the array size of the network is equal to the previous traffic profile. Uniform and Hotspot synthetic traffic patterns<sup>14,28,29</sup> are used to generate the unicast traffic in



**Fig. 16.** Performance results in  $8 \times 8$  2D-mesh under multicast traffic profile with (a) 10 destinations, (b) 20 destinations.



**Fig. 17.** Performance with different loads in  $8 \times 8$  2D-mesh under mixed traffic (20% multicast and 80% unicast). Unicast traffic in (a) is based on the uniform pattern and in (b) is based on the Hotspot pattern with  $h = 10\%$ .



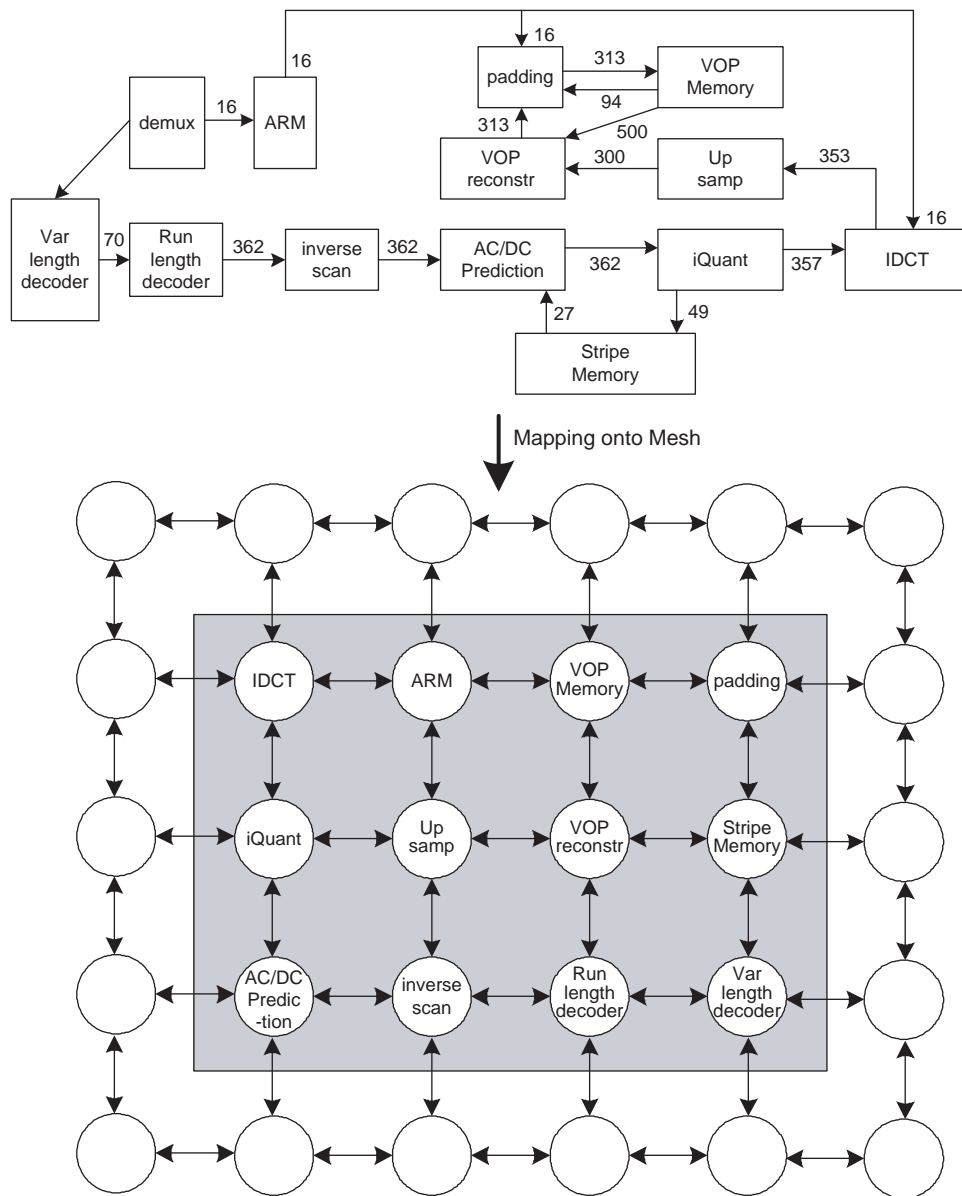
**Fig. 18.** Performance with different loads in  $8 \times 8$  2D-mesh under unicast traffic: (a) the uniform pattern and (b) the Hotspot pattern with  $h = 104\%$ .

the network. In the uniform traffic profile, each PE sends a message to any other PE with equal probability. This is determined randomly using a uniform distribution. Under the hotspot traffic pattern, one or more nodes are chosen as hotspots receiving an extra portion of the traffic in addition to the regular uniform traffic. In Figure 17(a) the average communication latency of different routers under the uniform traffic model for unicast traffic are shown. In this traffic, AIOS performs better than all of the other three algorithms. The average latency of each network has been computed near saturation point (0.2) where the unicast traffic is uniform. As a result, compared with the P-OE, P-MP, RR-OE, and RR-MP, the average latency of AIOS is reduced by 15%, 27%, 19%, and 24%, respec-

tively. Under the hotspot traffic model, given a hotspot percentage of  $h$ , a newly generated message is directed to each hotspot node with an additional  $h$  percent probability. We simulate hotspot traffic with a single hotspot node. The hotspot node is chosen to be node (4, 4) in the  $8 \times 8$  2D-Mesh with  $h = 10\%$ . As observed from Figure 17(b), AIOS shows considerably smaller delays compared to the other router models.

### 5.1.3. Unicast Traffic Profile

For appraising the unicast efficiency of AIOS, The uniform and hotspot traffic profiles, where 100% of injected messages are unicast messages have been considered.



**Fig. 19.** The VOPD block diagram, with communication BW annotated (in MB/s) and its mapping onto mesh topology (ARM and demux are assumed to be integrated in one core).

Figures 18(a) and (b) show the simulation results for the uniform and hotspot traffic profiles. As depicted, when the injection rate is increased, AIOS is superior to all of the other schemes. In brief, as the injection rate increases, the proposed algorithm leads to smaller average delays. This is due to the fact that the input-selection method uses WRR scheme which allows packet flows coming from congested paths to be serviced more often according to their congestion level. In contrast, in the RR scheme no matter how congested a path is, all packet flows are serviced equally. In the mechanism based on CAIS (priority), congested input channels which have higher numbers of request are serviced more often while the input channels with lower traffic may not be serviced leading to the starvation problem.

#### 5.1.4. Video Object Plane Decoder (VOPD) Traffic Profile

To evaluate the performance of the proposed algorithm under more realistic traffic loads, we use Video Object Plane Decoder (VOPD) Traffic Profile.<sup>30</sup> Although our algorithm is not proposed for real-time applications, we only use VOPD traffic profile as an example of real traffic profile without considering its real-time constraints. In this experiment, PE's generate packets and inject them into the network in the intervals determined by an exponential distribution. The mesh array size is assumed to be  $6 \times 5$ . In Figure 19, we show the core graph and its mapping onto the mesh for the VOPD. The other cores around the grey box generate uniform traffic, with the same average flit injection rate as the VOPD cores. As the results shown in Figure 20 reveal, for this traffic model, in the central areas of the chip, some congestion may occur. Therefore, since the presented routers are based on the adaptive routing algorithms, they do not send packets in the central areas when these areas are congested and thus distribute the traffic over the rest of the chip area. This strategy reduces the average delay of the packet transportation.

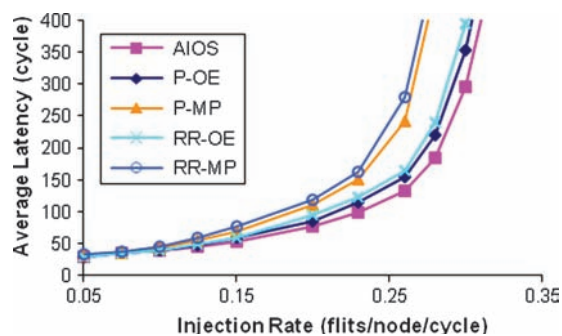


Fig. 20. The performance of different algorithms under VOPD traffic model.

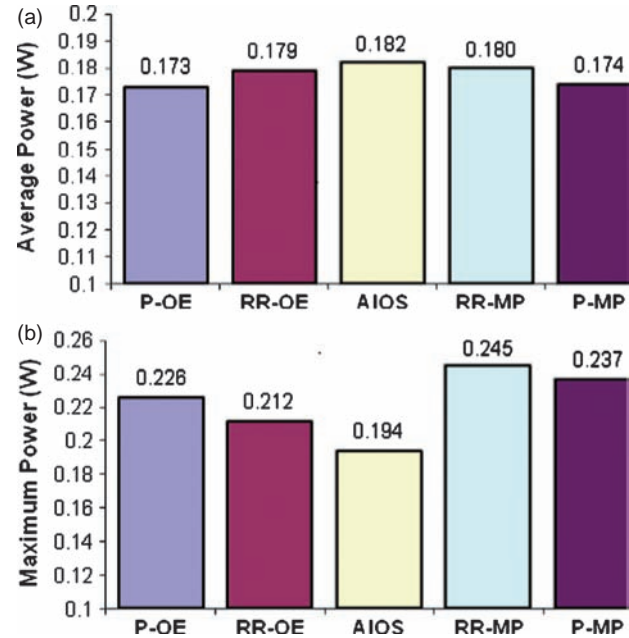


Fig. 21. (a) Average and (b) Maximum power dissipation results in  $8 \times 8$  2D-mesh under mixed traffic profile.

## 5.2. Power Dissipation

Using the simulator, the power dissipation of each model is calculated and compared under the mixed traffic profile. The results for the average and the maximum power under mixed traffic are shown in Figures 21(a) and (b), respectively. Both average and maximum power values are computed near the saturation point, 0.23 (flits/cycle), under mixed traffic. As the results presented in the figures, the average power dissipation of the network with AIOS is 5%, 4%, 1.5%, and 1% larger than those of P-OE, P-MP, RR-OE, and RR-MP, respectively. Also, the results of the maximum power of AIOS is 16%, 22%, 10%, and 26% less than those of the P-OE, P-MP, RR-OE, and RR-MP, respectively. We can notice that the maximum power, compared to other routers, is considerably lowered in our proposed router. This is achieved by smoothly distributing the power consumption over the network using the output

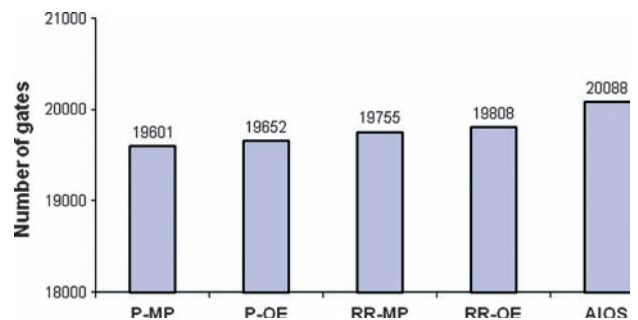


Fig. 22. Area cost of routers for implementing different input-output selections.

selection scheme which reduces the number of the hotspots and, hence, lowering the maximum power.

### 5.3. Hardware Overhead

To evaluate the area overhead of the presented model and demonstrate the performance/area trade-off, RTL models of aforementioned routers are implemented with four different input-output selection schemes using VHDL. The routers are described in VHDL and synthesized with Synopsys D.C. using the CMOS STMicroelectronic 65 nm technology. For all routers, the data width is set to 32 bits, and each input channel has a buffer size of 10 flits. The FIFOs are implemented in our design using registers in order to achieve better performance/power efficiency. We perform place-and-route via Cadence SoC-Encounter for more accurate area estimation. 21 shows the area cost of the switches. Comparing the area cost of AIOS with P-MP, P-OE, RR-MP and RR-OE introduces 2.4%, 2.1%, 1.6% and 1.3% additional overhead respectively.

## 6. SUMMARY AND CONCLUSION

In this paper, a router architecture based on the adaptive input and output selection is proposed. The output selection of the presented router utilizes an adaptive routing algorithm supporting both unicast and multicast traffic while the input selection part of the router uses the weighted round robin arbitration. Also, the adaptive output selection algorithm supporting both minimal and non-minimal paths uses congestion flags to route packets through less congested paths and consequently helps balance the traffic. The WRR input selection also assists in relieving nodes where congestion is formed. A simulator was used to evaluate the efficiency of the proposed router. Under the multicast, unicast, mixed, and VOPD traffic models and in high flit injection rates, the proposed architecture has the lowest average communication delay in comparison with the other router models. It also reduces the maximum power dissipation of the network compared to other models under mixed traffic model.

## References

1. L. Benini and G. De Micheli, Networks on chips: A new SoC paradigm. *IEEE Computer* 35, 70 (2002).
2. W. J. Dally and B. Towles, Route packets, not wires: On-chip interconnection networks. *DAC* 684 (2001).
3. A. Hemani, A. Jantsch, S. Kumar, A. Postula, J. Oberg, M. Millberg, and D. Lindqvist. Network on chip: An architecture for billion transistor era, *Proc. of the IEEE NorChip Conf.* November (2000), pp. 120–124.
4. S. Kumar, A. Jantsch, M. Millberg, J. Oberg, J. Soininen, M. Forsell, K. Tiensyrj, and A. Hemani. A network on chip architecture and design methodology, *Proc. Symposium on VLSI* April (2002), pp. 117–124.
5. R. V. Boppana, S. C., and C. S. R., Resource deadlock and performance of wormhole multicast routing algorithms. *IEEE Transactions on Parallel and Distributed Systems* 535 (1998).
6. P. Abad, V. Puente, and J. Á. Gregorio, MRR: Enabling fully adaptive multicast routing for CMP interconnection networks. *High Performance Computer Architecture (HPCA)* (2009).
7. J. Hu and R. Marculescu, DyAD-Smart Routing for Networks-on-Chip, *DAC* 2004, San Diego, California, USA (2004), pp. 260–263.
8. C. A. Zeferino, M. E. Kreutz, and A. A. Susin, RASoC: A router soft-core for networks-on-chip. *Designers Forum—DATE*, France (2004), pp. 198–203.
9. D. Wu, B. M. Al-Hashimi, and M. T. Schmitz, Improving routing efficiency for network-on-chip through contention-aware input selection. *Proc. of 11th ASP-DAC* (2006), pp. 36–41.
10. E. Nilsson, M. Millberg, J. Oberg, and A. Jantsch, Load Distribution with the Proximity Congestion Awareness in a Network on Chip, *DATE*, Germany (2003), pp. 1126–7.
11. A. Demers, S. Keshav, and S. Shenkar, Analysis and simulation of a fair queuing algorithms, *Proceedings of SIGCOMM '89*, August (1989), pp. 3–12.
12. J. Duato, C. Yalamanchili, and L. Ni, Interconnection Networks: An Engineering Approach, Morgan Kaufmann Publishers (2003).
13. L. M. Ni and P. K. McKinley, A survey of wormhole routing techniques in direct networks. *IEEE Tran. on Computers* 26, 62 (1993).
14. G. Chiu, The odd-even turn model for adaptive routing. *IEEE Tran. on Parallel and Distributed System* 729 (2000).
15. C. J. Glass and L. M. Ni, The Turn Model for Adaptive Routing, *Proc, Symp, Computer Architecture* May (1992), pp. 278–287.
16. T. T. Ye, L. Benini, and G. De Micheli, Packetization and routing analysis of on-chip multiprocessor networks. *Journal of Systems Architecture* 50, 81 (2004).
17. U. Feige and P. Raghavan, Exact analysis of hot-potato routing, *Proceedings of the 33rd Annual Symposium on Foundations of Computer Science* (1992), pp. 553–562.
18. M. Ebrahimi, M. Daneshtalab, P. Liljeberg, and H. Tenhunen, HAMUM—A novel routing protocol for unicast and multicast traffic in MPSoCs, *Proceedings of 18th IEEE Euromicro Conference on Parallel, Distributed and Network-Based Computing (PDP)*, Italy, February (2010), pp. 525–532.
19. E. A. Carara and F. G. Moraes, Deadlock-free multicast routing algorithm for wormhole-switched mesh networks-on-chip. *Prof. of ISVLSI* (2008), pp. 341–346.
20. M. Daneshtalab, M. Ebrahimi, S. Mohammadi, and A. Afzali-Kusha, Low distance path-based multicast algorithm in NOCs. *IET Computers and Digital Techniques, Special issue on NoC* 3, 430 (2009).
21. P. Mohapatra and V. Varavithya, A hardware multicast routing algorithm for two-dimensional meshes, *proc. Int. Conf. SPDP, New Orleans* (1996), pp. 198–205.
22. X. Li, P. K. Mckinley, and L. M. Ni, Deadlock-free multicast wormhole routing in 2-D mesh multicomputers, *IEEE transactions on Parallel and Distributed Systems* v.5, i.8 793 (1994).
23. J. Liang, S. Swaminathan, and R. Tessier. aSOC: A scalable, single-chip communication architectures, *IEEE Int. Conf. on Parallel Architectures and Compilation Techniques* October (2000), pp. 37–46.
24. C.-M. Chiang and L. M. Ni, Multi-address encoding for multicast, *Proceedings of the Workshop on Parallel Computer Routing and Communication*, May (1994), pp. 146–160.
25. P. Gupta and N. McKeown. Designing and implementing a fast crossbar scheduler. *IEEE Computer Society Press* 20, January (1999).
26. K. Lee, S. Lee, and H. Yoo, A distributed crossbar switch scheduler for on-chip networks, *IEEE Int. Conf. on CICC*, September (2003), pp. 671–674.

27. D. C. Stephens, J. C. R. Bennet, and H. Zhang, Implementing scheduling algorithms in high-speed networks. *IEEE Journal on Selected Areas in Communications* 17, 1145 (1999).
28. A. Kahng, B. Li, L. Peh, and K. Samadi, ORION 2.0: A fast and accurate NoC power and area model for early-stage design space exploration, *Proc. of DATE*, France, April (2009).
29. R. V. Boppana and S. Chalasani, A comparison of adaptive worm-hole routing algorithms, *Proc. Int'l. Symp. Computer Architecture*, May (1993), pp. 351–360.
30. E. B. Van der Tol and E. G. T. Jaspers, Mapping of MPEG-4 decoding on a flexible architecture platform, *Proc. SPIE 2002*, January (2002), pp. 1–13.

### **Masoud Daneshtalab**

Masoud Daneshtalab received his bachelor degree in computer engineering from Shahid-Bahonar University of Kerman in 2002, and Master degree in computer architecture from School of Electrical and Computer Engineering, University of Tehran in 2006. Since autumn 2008 he has been working in the Embedded Computer Systems laboratory, University of Turku and from May 2009 he is a doctoral candidate of Graduate School in Electronics, Telecommunications and Automation (GETA). He is expected to get his Ph.D. degree in 2011. His current research interests include on/off-chip interconnection networks for multiprocessor architectures, networks-on-chips (NoC), dynamic task allocation, 3D systems, and low-power digital design. His Ph.D. thesis is focused on adaptive implementation of on-chip networks. Masoud is a member of IEEE and has published more than 50 refereed international journals and conference papers. He has served as a program committee member in different conferences, including PDP, ICCESS, and DATICS.

### **Masoumeh Ebrahimi**

Masoumeh Ebrahimi received her bachelor degree in computer engineering from School of Electrical and Computer Engineering, University of Tehran in 2005, and the master degree in computer architecture from Azad University, Science and Research branch, in 2009. Since spring 2009 she has been working in the Embedded Computer Systems laboratory, University of Turku. She has expertise in interconnection networks, networks-on-chip, 3D integrated systems, and systems-on-chip. Her Ph.D. thesis is focused on routing protocols in 2-D and 3-D NoCs. Masoumeh is a member of IEEE and has published more than 20 international refereed journals and conference papers.

### **Ali Afzali-Kusha**

Ali Afzali-Kusha received his B.Sc., M.Sc., and Ph.D. degrees in Electrical Engineering from Sharif University of Technology, the University of Pittsburgh, and the University of Michigan in 1988, 1991, and 1994, respectively. From 1994 to 1995, he was a Post-Doctoral Fellow at the University of Michigan. Since 1995, he has been with the University of Tehran, where he is currently a Professor in the School of Electrical and Computer Engineering and the Director of the Low-Power High-Performance Nanosystems Laboratory. Also, while on a research leave from the University of Tehran, he was a Research Fellow at the University of Toronto and the University of Waterloo in 1998 and 1999, respectively. His current research interests include low-power high-performance design methodologies from the physical design level to the system level for the nanoelectronics era. Dr. Afzali-Kusha is a senior member of IEEE.