FACTA UNIVERSITATIS
Series: Electronics and Energetics Vol. 29, No 2, June 2016, pp. 309 - 323
DOI: 10.2298/FUEE1602309S

IN-CHANNEL MISROUTING SUPPRESSION TECHNIQUE FOR DEFLECTION-ROUTED NETWORKS ON CHIP

Igor Z. Stojanovic, Goran Lj. Djordjevic

Faculty of Electronic Engineering, University of Niš, Serbia

Abstract. *Deflection routing, where port-contentions in routers are resolved by intentionally misrouting some of packets along unwanted directions instead of storing them, has been proposed as a promising approach for improving power and area efficiency of large-scale networks on chip (NoCs). However, at high network load, when packets are misrouted more frequently, the cost and energy benefits of this simple routing scheme are offset by the performance degradation. To address this problem, we propose a technique that uses small in-channel buffers to capture some of deflected packets before they take a misrouting hop. The captured packets are then looped-back to the routers where they suffered deflection and routed again. To improve the efficiency of this in-channel misrouting suppression scheme we also slightly modify the routing function of the deflection router by restricting the choice of productive directions for misrouted packets. Evaluations on synthetic traffic patterns show that the proposed misrouting suppression mechanism yields an improvement of 36.2% in network saturation throughput when implemented into the conventional deflection-routed network.*

Key words: *Network-on-chip, multi-core, deflection routing, misrouting suppression*.

1. INTRODUCTION

Network-on-chip (NoC) has been proposed as an efficient and scalable solution to the challenging on-chip interconnection problems in modern many-core systems on chip (SoCs). To accommodate the communication needs of tens or even hundreds of processing elements (PEs) integrated on a single chip, this architecture employs dedicated routers interconnected by some form of network topology. NoCs typically use wormhole routing with virtual channel (wormhole/VC) flow control to route data packets from the source to the destination PE. This flow control scheme enables deadlock avoidance, optimize channel utilization, improve performance and provide quality of service [1, 2]. Although wormhole/VC routing needs considerably less amount of buffer storage then other traditional flow control schemes (e.g. virtual cut-through and store-and-forward), the in-router buffers are still a significant source of area and energy overhead. For a static random access memory

Received June 3, 2015; received in revised form August 3, 2015
Corresponding author: igor.stojanovic@elfak.ni.ac.rs
Faculty of Electronic Engineering, University of Niš, A. Medvedeva 14, 18000 Niš, Serbia
(e-mail: mita@iritel.com)

(SRAM) buffer implementation, the input buffers can consume 46% of the total on-chip network power while occupying 17% of the total area [3]. To address the issue, several *bufferless* NoC architectures have recently been proposed. In these architectures, in-router buffers are removed and contentions among packets are handled by employing the deflection routing [4-13].

With deflection routing, data packets are divided into *flits* (flow control units) which are then routed independently through the network and reassembled at their destination. Flits arrive synchronously on the router's input ports, and each flit is routed via the output port that offers the shortest path to its destination. When two incoming flits require the same output port, the router deflects one of the flits to an alternative output port (this is always possible as long as the router has as many outgoing as incoming ports). In this way, port contentions cause flits to be misrouted temporarily, in contrast with the wormhole/VC scheme where such flits must be buffered.

Deflection routing has several advantages over wormhole/VC scheme. First, since the number of incoming ports is equal to the number of outgoing ports, and flits move between routers synchronously, deadlock cannot occur. The adaptive nature of deflection routing also enables hot spots avoidance and provides fault-tolerance in the network [4]. This approach also eliminates the need for backward status links to implement flow control, and thus the design of the router is greatly simplified. Finally, the deflection routing permits the use of as few as one flit-wide register per inter-router link, thereby realizing significant savings in hardware cost and power consumption over wormhole/VC NoCs, which must provide ample buffers in each router. Recent studies have shown that in the deflection-routed NoCs, the power consumption is reduced by 20-40%, and the router area on die is reduced by 40-75% [6].

Deflection routers target mainly low-latency operation at low network load [5]. Under such load conditions, deflections are rare so that flits rapidly advance toward their destinations over shortest paths. On the other hand, under high load, frequent deflections might cause flits to deviate significantly from their shortest paths, leading to early saturation and poor energy efficiency. The issue of limited maximum throughput of deflection-routed networks has been addressed by several prior works. One line of research is aimed at improving the design of router's port allocator and switching (PAS) stage. Within this stage, input flits are first permuted and then passed to the router's output ports so that as many flits as possible are directed toward their desired directions. BLESS router uses the PAS stage composed of a 4×4 crossbar switch controlled by an allocator unit that arbitrates the flits to output ports based on oldest-first arbitration policy [6]. The full priority ordering of flits results in fewer deflections, but it incurs a long critical path delay, thus limiting router operation to low clock frequencies. CHIPPER router speeds up the critical path of the router by replacing the crossbar with a two-stage permutation network composed of four independently controlled 2×2 switch modules [7]. However, the simplicity of this design results in an increased deflection rate, and consequently lowers the maximum network throughput.

Another line of research deals with techniques for reducing the overhead of flit deflection. Such *misrouting suppression* mechanisms try to prevent deflected flit to take a misrouting hop by temporary holding the flit at its current route position. The minimally buffered deflection router (MinBD) achieves the misrouting suppression by a small side-buffer attached between the output and the input of the router's PAS stage [8]. At each clock cycle, the side-buffer can accept up to one of deflected flits from PAS output, and

resubmit that flit to the PAS input at some later cycle. By preventing a fraction of deflected flits to leave the router, this technique significantly improves the maximum network throughput. However, it also introduces the contention between the buffered flits and the new flits waiting for injection, which can cause the injection unfairness among routers in a highly loaded network. In our previous work, we proposed an in-channel misrouting suppression technique, referred to as the dual-mode channel, which uses a lightweight link-control mechanism to force deflected flits, when possible, to loop-back to their current routers instead of being misrouted [9]. This simple and effective method improves performances without compromising the injection fairness, but the obtained maximum network throughput is lower than that obtained with the side-buffering technique.

In this paper, we further improve the misrouting suppression efficiency of the dual-mode channel by adding small buffers at both ends of the channel. These buffers temporary store deflected flits that cannot be looped-back during the same clock cycle when they are entering the channel. Also, we slightly modify the routing function of the baseline deflection router to remove the tendency of misrouted flits to take immediate reverse hops. This modification is motivated by our observation that such hops have an adverse effect on how often the channel is able to loop-back the deflected flits. When combined, the proposed mechanisms suppress more than 50% of misrouting hops, raising the maximum throughput by 36.2% with respect to the baseline deflection-routed network. The throughput improvement is 8.7% higher than with the side-buffering technique, and is achieved without compromising the injection fairness in the network.

The remainder of the paper is organized as follows. Section 2 provides a background on deflection routing including the overview of two representative misrouting suppression techniques: the side-buffering and the dual-mode channel. Section 3 presents the novel misrouting suppression scheme for deflection-routed NoCs. In Section 4, evaluation and results are presented. Section 5 concludes this paper.

2. DEFLECTION-ROUTED NOC ARCHITECTURE OVERVIEW

In this section, we first provide a generic model of deflection-routed NoC architecture, which includes only the essential features reported in several previous proposals [5-13]. In particular, we consider a network of 2D mesh topology composed of non-pipelined (i.e. combinational) deflection routers connected by synchronous bidirectional communication channels. Then we also discuss two existing techniques to improve the performance of the baseline deflection-routed network via misrouting suppression.

2.1. Baseline 2D mesh deflection network

Figure 1 illustrates the fundamental elements of a generic 2D mesh deflection-routed NoC. The NoC is constructed as a grid of routers where each router is connected by bidirectional communication channels only to its neighbors. Each router is also connected to a local PE, which serves as a source and sink for data packets. Before being injected to the router, packets are split into smaller flow control units, so called flits, and each flit is routed independently through the network. In the most basic form, the deflection router is a pure combinational logic module, which directs the incoming flits from the input ports to the proper output ports. The inter-router communication channel includes a pair of

oppositely oriented flit-wide edge-triggered registers. Since there are no in-router buffers, these so-called flit-registers are the only memory elements for storing flits in transit. Therefore, during traveling towards their destinations, flits are always on the move, by hopping between the flit-registers and propagating through the routers.

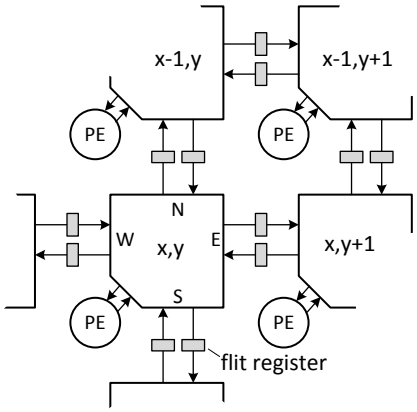


Fig. 1 2D mesh deflection-routed NoC architecture

Routers attempt to route each flit along a shortest path to its destination. A router forwards a flit through a productive output port in a productive direction if the distance between the current flit position and its destination decreases. In 2D mesh network, when a flit reaches a router, there are at most two productive directions (i.e. output ports) to its destination. If the router is not able to grant the productive output port, the flit is deflected to any free but non-productive output port. Deflection occurs within the internal router structure when multiple incoming flits contend for the same output port. On the other hand, the term misrouting refers to an external manifestation of the flit deflection. It corresponds to a transfer of a deflected flit over the inter-router channel one hop further in a non-productive direction. The cost of misrouting is two clock cycles since each non-productive hop must be compensated by one productive hop in the opposite direction. Let note that in the baseline deflection-routed network, every flit deflection leads to a flit misrouting.

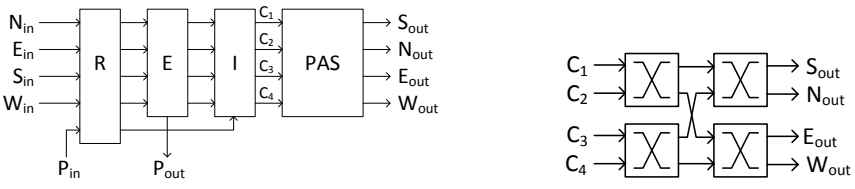
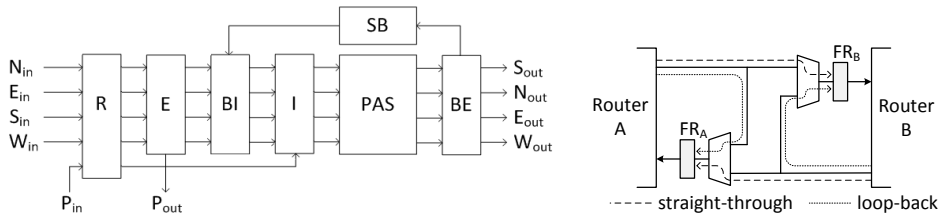


Fig. 2 Architecture of baseline deflection router:
a) internal structure, and b) PAS based on permutation network

Figure 2a shows the architecture of the deflection router with four pairs of input and output network ports (denoted as *N - North*, *S - South*, *W - West* and *E - East*) and a pair of inject and eject ports (denoted as P_{in} and P_{out}) which are connected to the local PE. The router is composed of four consecutive stages: the routing stage (R), the eject stage (E), the inject stage (I), and the port allocation and switching stage (PAS). Through these stages, four internal flit-channels, C_1, ..., C_4, are established to guide flits from the set of input to the set of output ports. The routing stage associates a set of productive ports to each incoming flit. The routing function is based on offsets in *X* and *Y* dimensions between the current router and the flit's destination router. The number of productive ports assigned to a flit can be: 0 (flit is addressed to the local PE, i.e. both *X*- and *Y*-offset are zero), 1 (flit is already at one of the axes of its final destination, i.e. either *X*- or *Y*-offset is zero) or 2 (both *X*- and *Y*-offset are different than zero). The eject stage picks randomly one of locally-addressed flits (if any), and directs that flit to the local PE. The inject stage detects the presence of a free flit-channel and directs the new flit (generated by the local PE) to that channel. If the new flit is not injected into the network because all flit-channels are occupied, then that flit remains in the PE's transmission queue and is resubmitted in the next clock cycle. The PAS stage permutes and passes the flits from flit-channels to output network ports. Here, we adopt a PAS stage introduced in CHIPPER router [7], which consists of four two-input switch modules arranged into two stages (Fig. 2b). Each switch module is controlled by an arbitration logic which first, decides the winner between two flits, and then, sends the winning flit toward its productive output port. The losing flit is directed to the other output of the module. The winner between two input flits is determined according to the silver-flit arbitration policy [8]. In this arbitration scheme, a single randomly selected flit is designated as a silver flit, i.e. it is prioritized above the others. The silver flit always wins in arbitration. The winner between any two non-silver flits is decided randomly.

2.2. Misrouting suppression techniques

The term misrouting suppression refers to any technique for reducing delay overhead incurred by flit deflection in deflection-routed networks [9]. These mechanisms cannot cancel flit deflection, which occurs within the PAS stage of the router, but they can recognize a deflected flit and force it to temporary stay at its current route position instead of making a non-productive hop. The misrouting suppression can be implemented either within the deflection router or within the inter-router communication channel.



a) b)

Fig. 3 Misrouting suppression techniques: a) in-router misrouting suppression with side-buffer, and b) in-channel misrouting suppression with dual-mode channel

Side-buffering. The side-buffering [8] is an in-router misrouting suppression technique which uses a small buffer memory (so-called side buffer) attached to each router to buffer some deflected flits that otherwise would be misrouted. The side buffer can be implemented either as a single flit-register or as a small-size FIFO (composed of several flit-registers). As shown in Fig. 3a, the side buffer (SB) is attached to the deflection router via two additional stages: the buffer-eject stage (BE) and the buffer-inject stage (BI). The BE stage recognizes deflected flits at the output of the PAS stage, and puts one of them into the side buffer if the side buffer is not full. This flit is picked randomly among the deflected flits. The buffered flit will be re-ejected through the BI stage in some later clock cycle, when there is a free flit-channel after flit ejection.

Previous studies have shown that even adding the smallest side-buffer (1-flit in size) can reduce the misrouting rate by 50%, and can improve the maximum network throughput by 26% [8]. However, the studies have also shown that the performance improvement of this technique does not scale with the increasing side-buffer size because increasing the buffer size over 2 flits leads to only marginal performance gain. More importantly, as pointed out in [9], the presence of side buffers can cause an imbalance between the injection and ejection bandwidth available to PEs in the areas of the network congested with in-transit traffic. This occurs because of the arrangement of stages within the side-buffered deflection router, which gives injection precedence to the flit residing in the side-buffer over the new flit waiting at the PE inject port. When the router is overloaded with in-transit flits, a free flit-channel appears rarely and is occupied by buffered flit in most cases, leaving the new flit to wait for another chance.

Dual-mode channel. The dual-mode channel is an in-channel misrouting suppression technique which prevents some non-productive network hops by forcing deflected flits, when possible, to loop-back to their current routers instead of being misrouted [9]. The datapath for this design is shown in Fig. 3b. The approach is based on enhancing the inter-router communication channel with the capability to dynamically (i.e. on a cycle-by-cycle basis) switches between two modes of operation. If deflected flits are present on both ends of the channel, or one flit is deflected and the other one is absent, then the channel activates the loop-back mode (indicated by dotted lines in Fig. 3b). In this mode, the flits are returned back to the corresponding input ports of their current routers. Otherwise, the channel is configured in the straight-through mode (indicated by dashed lines) allowing both flits to make one network hop. With this scheme, a deflected flit will be misrouted only if there is a productively-routed flit on the opposite side of the channel. In all other cases, the deflected flit will stay at its current route position. It is important to note that the loop-back mechanism is transparent for productively-routed flits, which flow as in a network with the conventional inter-route channels.

Our previous simulation results show that this simple in-channel misrouting suppression mechanism offers 14.3% performance improvement in terms of maximum network throughput when implemented in the baseline deflection-routed NoC [9]. The improvement is smaller when comparing with the side-buffering technique, but is accomplished with lower implementation cost (i.e. there is no need for additional buffer memory) and without any modification to the underlying router microarchitecture. An important advantage of the dual-mode channel approach over the side-buffering is that it preserves the injection fairness in the network.

3. MISROUTING SUPPRESSION WITH IN-CHANNEL BUFFERING

The limited misrouting suppression efficiency of the dual-mode channel is a consequence of the fact that the channel cannot save a deflected flit from misrouting if a productively-routed flit is present on the opposite end of the channel. Under high traffic, when the inter-router channels are almost fully utilized, the loop-back mode can only be activated when both ends of the channel are occupied by deflected flits, which occur rarely. In this section we propose two techniques to mitigate the performance limitation of the dual-mode channel. The first one relates to modifying the routing function of the baseline deflection router with goal to increase the frequency of simultaneous appearance of deflected flits at both sides of the channel. The second technique deals with adding a small in-channel buffer memory for temporary storing deflected flits that cannot be looped-back immediately.

3.1. Optimized routing function

According to the results of our simulation experimentation with 2D mesh deflection networks under saturated load with uniform random traffic pattern, a deflected flit appears at a router's output port with the probability of $\delta = 0.3$. Assuming that flit deflections occur in neighboring routers independent, the probability that both sides of an inter-router channel are fed with deflected flits should be $\delta^2 = 0.09$. However, the simulation results show that this probability is actually 0.05. That is, the loop-backs in dual-mode channels occur less frequently than would be expected.

A closer examination of the patterns of inter-router communication reviles that the discrepancy between expected and measured loop-back probability is caused by the tendency of the misrouted flits to return back to the routers wherein they have suffered deflection during the previous clock cycle. Suppose that a flit *f* is deflected in a router *A* and then misrouted to a router *B* over channel C_{AB}. Upon arriving at router *B*, the flit *f* is assigned with at most two productive ports. Because flit *f* is misrouted, one of its productive ports must be the port through which it just has entered the router *B*. Therefore, during the next clock cycle, there is a high chance that flit *f* will be returned back to the router *A* over the channel C_{AB}, but now as a productively-routed flit, thus forcing the straight-through configuration of the dual-mode channel. If happens that router *A* sends deflected flit to channel C_{AB} during the next clock cycle, that flit will be misrouted, too. Thus, the net effect of such behavior is that the likelihood of flit misrouting depends on whether a flit sent by the same router over the same channel during the previous clock cycle was misrouted or not.

In order to resolve this performance issue, we slightly modify the routing function of the baseline deflection router by restricting the choice of productive ports for misrouted flits. In particular, we extend the routing function of the deflection router with the following rule:

Rule 1: Let flit *f* has entered a router *A* through the input port $T \in \{N,S,E,W\}$, and let $P \subset \{N,S,E,W\}$be the set of productive output ports for flit *f* in router *A*. If the size of *P* is two, then remove *T* from *P*.

Rule 1 only impacts the implementation of the routing stage of a deflection router (Fig. 3a). It is applied after the incoming flits are assigned with productive ports. If flit *f* has reached router *A* by a productive hop, then Rule 1 has no effect on the routing decision regarding *f* because *T* cannot be in *P*. Otherwise, if flit *f* has arrived at router *A*

by a misrouting hop, then port *T* must be in *P*. In this case, port *T* will be preserved in *P* if *T* is the only productive option for *f*. Otherwise, *T* is removed form *P*. Without *T* in the set of its productive ports, flit *f* will not be intentionally returned back to the previous router, unless it is deflected within the PAS stage of router *A*. It should be noted that Rule 1 does not preclude the possibility that a misrouted flit will be returned back to the previous router; it only decreases the likelihood of such event to occur.

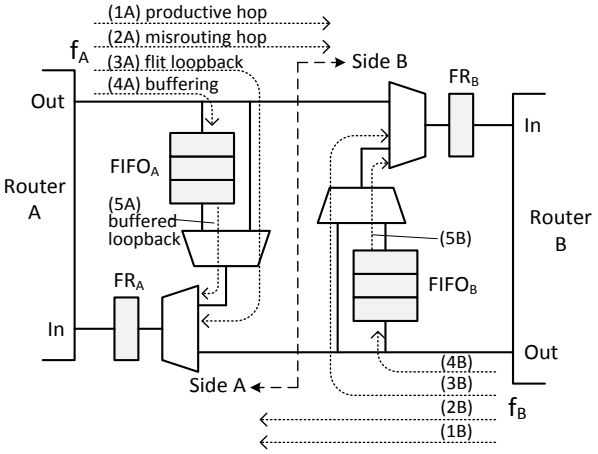
3.2. In-channel buffering

The main motivation for using the in-channel buffers is to decouple the operations of the two sides of the dual-mode channel by enabling each side to buffer incoming deflected flits which cannot be looped-back immediately. Thus, instead of being misrouted to a neighboring router, the buffered deflected flit will be kept at its current route position until the condition for looping-back is met. When eventually looped-back to the router that has caused its deflection, the flit will get a new chance to continue traveling along a productive direction toward its destination.

The datapath of the proposed inter-router channel with in-built flit-buffers is shown in Fig. 4a. In comparison to the dual-mode channel (Fig. 3(b)), the buffered channel contains two additional small-sized FIFO sections which parallel direct loop-back paths. With FIFOs included, the dual-mode channel is enhanced with several new options on how to handle the incoming and buffered flits. As indicated by dotted lines in Fig. 4a, the buffered channel can carry out one or more of ten different flit-transfer actions during each clock cycle. The choice of the actions depends on the routing statuses of the incoming flits as well as the statuses of the two FIFOs. The first set of options is for transferring of an incoming flit straight-through to the flit-register on the opposite side of the channel. If the incoming flit is productively-routed, this action leads to a productive hop (actions labelled as 1A/1B); otherwise, if the flit is deflected, the straight-through transfer causes a misrouting hop (2A/2B). The second set of options is those that keep an incoming flit on the same side of the channel. The flit loop-back action (3A/3B) allows an incoming flit to bypass the FIFO and immediately reach the flit-register (FR_A/FR_B) on the same side of the channel. The incoming flit can also be buffered (4A/4B), and a buffered flit can be looped-back (5A/5B).

A C-like pseudo code describing the operation of the dual-mode channel with in-built buffers is shown in Fig. 4b. Consider the operation of the A-side part of the channel in more details. The B-side part operates analogously. The A-side part of the channel can be configured in either the straight-through or the loop-back mode. The straight-through mode moves the opposite-side flit, f_B, into the A-side flit-register, FR_A. In the loop-back mode, either the A-side incoming flit, f_A, or the flit taken form $FIFO_A$ is written into the FR_A. The straight-through mode is prioritized over the loop-back mode, and occurs in two distinct situations: when the flit f_B is productively-routed (1B), and when the flit f_B is deflected and must be misrouted (2B). The deflected flit f_B is misrouted if there are no other options for handling that flit, i.e. the loop-back path of B-side is blocked by a productively-routed flit f_A and the $FIFO_B$ is full. Even if the A-side part of the channel is configured in the straight-through mode, a deflected flit f_A can still be saved from misrouting by storing into $FIFO_A$ if $FIFO_A$ is not full (4A). If the A-side loop-back path is enabled, the flit-register FR_A receives either a flit from $FIFO_A$ (5A) if $FIFO_A$ is not empty or an incoming flit f_A, if that flit is deflected and $FIFO_A$ is empty. In the case of buffered loop-back action (5A),

the incoming flit f_A, if deflected, is written into $FIFO_A$ (4A). It should be noted that a situation where both incoming flits are misrouted is not possible with this scheme. The critical case is one where both FIFOs are full, and both incoming flits, f_A and f_B, are deflected. According to the algorithm, in this case, both sides of the channel activate the buffered loop-back operation (5A/5B), which enables buffering of both flits (4A/4B) regardless of the current FIFOs statuses.



a)

```
        Side A:                                  Side B:
        if(fB.P ||                               if(fA.P ||
           fB.N && fA.P && FIFOB.Full){             fA.N && fB.P && FIFOA.Full){
1B/2B      FRA ← fB;                        1A/2A     FRB ← fA;
           if(fA.N && !FIFOA.Full){                   if(fB.N && !FIFOB.Full){
 4A           FIFOA ← fA;                     4B          FIFOB ← fB;
           }                                          }
        } else if(!FIFOA.Empty){                 } else if(!FIFOB.Empty){
 5A        FRA ← FIFOA;                      5B        FRB ← FIFOB;
           if(fA.N){                                  if(fB.N){
 4A           FIFOA ← fA;                     4B          FIFOB ← fB;
           }                                          }
        } else if(fA.N){                         } else if(fB.N){
 3A        FRA ← fA;                         3B        FRB ← fB;
        }                                        }
```

b)

Fig. 4 Misrouting suppression with in-channel buffering: (a) datapath; (b) pseudo code.
Notice: *f.P* is true if flit *f* is productively-routed; *f.N* is true if flit *f* is deflected;
sign "←" denotes a register transfer operation.

In-channel buffering vs. side-buffering. The rationale of using in-channel buffering is similar to that of using side-buffering – to buffer some deflected flits that otherwise would be misrouted. In difference to the side-buffering, which picks and buffers deflected flits before they leave the router, the in-channel buffers store deflected flits that have entered

the inter-router channel but cannot be looped-back immediately. By placing buffers within the channels instead of within the routers brings the following advantages. As opposite to the side-buffering that can accept up to one deflected flit per router per clock cycle, the buffered dual-mode channel can loop-back/store up to two deflected flits at each clock cycle. In a 2D mesh network with dimension of $N \times N$, the number of routers is N^2 and the number of inter-router channels is $2N^2$ - $2N$. Because the number of inter-router channels is almost two times greater than the number of routers, the opportunities to capture deflected flits are more frequent with the in-channel buffering than with the side-buffering. Moreover, being stored outside the routers, the flits buffered into the in-channel FIFOs re-enter the routers via network ports, and consequently they do not block the new flits generated by PE to enter the router. In this way, the problem of injection unfairness is avoided. The minimum delay overhead of a deflected flit which is buffered into an in-channel FIFO is two clock cycles: the first cycle is used for buffering, and the second one for looping-back the buffered flit. Although the delay overhead is the same as in the case of misrouting, the in-channel buffering is still beneficial since the buffered flit does not occupy the resources of the neighboring router.

4. Performance Evaluations

In order to evaluate the performance impact of the proposed misrouting suppression technique, we have developed a discrete-event, cycle-accurate simulator for modeling deflection-routed NoC using SystemC [14]. It provides support to experiment with deflection NoC with various options available, such as network topology and size, router/ channel architecture, buffer parameters, and traffic modelling. The simulator provides output performance metrics, such as latency, throughput, transport delay, and deflection rate for a given set of choices. The main building blocks of the simulator are: 1) processing element, 2) deflection router, and 3) inter-router channel (IRC). The processing element block generates and injects flits into the network according to the user-specified configuration, including the traffic pattern and injection rate. It is also responsible for ejecting flits from the destination endpoints and collecting appropriate statistics. The router block mimics the behavior of the generic non-pipelined deflection router described in Section 2. It can be configured in the bufferless mode (i.e. without side-buffer) or the buffered mode (with side-buffer of configurable size). The configuration options for the IRC block are the following: conventional channel (a pair of oppositely oriented flit-registers), dual-mode channel (see Fig 3*b*), and buffered channel (see Fig 4). The simulation results presented in this section are obtained for 2D mesh network with size of 8×8 nodes. The default buffer size in buffered architectures was set to 1 flit. Each simulation run was started with a warm-up period of 1,000 cycles followed by a measurement period of 20,000 cycles.

4.1. Performance under saturation load

The first set of evaluations was carried out in a saturation mode under uniform random traffic pattern. In this mode, the transmission queue of each PE is assumed to be always nonempty. Under such overloaded conditions, each PE injects a new flit into the network in every clock cycle when a free flit-channel is available in the router inject stage. The

injected flits are destined randomly to other PEs with an equal probability. A summary of the results is given Table 1.

Table 1 Comparison of saturation performance of baseline deflection-routed NoC architecture and architectures with misrouting suppression support

	th	*td*	*h*	δ_r	μ_r	μ_e
Baseline	0.265	13.216	13.216	0.298	0.298	0
Side-buffering	0.332	11.016	8.696	0.295	0.143	51.5%
Dual-mode channel	0.303	11.555	10.889	0.298	0.240	19.36%
In-channel buffering	0.361	14.541	8.144	0.305	0.145	52.3%

The details of the performance measures reported in Table 1 are as follow. The saturation throughput (*th*) is defined as the average number of flits received per PE per clock cycle. It is the single most important network-level performance indicator, which being measured under saturation load provides an absolute limit reached by the throughput of a deflection-routed network. The transport delay (*td*) is the time, measured in clock cycles, elapsed from the instant when the source PE injects a flit to the network to the instant when the destination PE receives it. Both the saturation throughput and the transport delay are correlated with the average hop count (*h*), which is defined as the average number of hops (i.e. channels traverses) a flit takes from source to destination. The average hop count accounts for both productive and non-productive (i.e. misrouting) inter-router hops. In networks where deflected flits are misrouted more often, the average hop count is larger, and consequently the transport delay is longer and throughput is lower. Deflections occur within the routers due to inability of PAS stage to grant productive ports to all incoming flits. The tendency of the PAS stage to produce deflections is measured with the deflection rate (δ_r), which is defined as $\delta_r = n_d / n_r$, where n_d is the total number of deflected flits, and n_r is the total number of flits that are processed by PAS stages of all routers during the simulation. Similarly, the misrouting rate (μ_r) is defined as $\mu_r = n_m / n_r$, where n_m is the total number of flits that are misrouted after deflection. The baseline deflection-routed network misroutes every deflected flit, thereby $\mu_r = \delta_r$. With a misrouting suppression mechanism implemented, not all deflected flits are misrouted. The misrouting suppression efficiency is defined as $\mu_e = ((\delta_r - \mu_r) / \delta_r) \times 100\%$.

The results in Table 1 show that the implementation of misrouting suppression techniques brings a significant improvement in saturation throughput over the baseline architecture. The dual-mode channel, as the simplest misrouting suppression technique, raises the throughput by 14.3% over the baseline, while the improvement reaches 25.3% for the side-buffering technique. The highest throughput of 0.361 *flits/cycle* is achieved with the in-channel buffering, which represents an improvement of 36.2% over the baseline.

In the baseline architecture, a flit takes 13.2 inter-router hops on average to reach its destinations. Misrouting suppression techniques decrease the average hop count (and thus increase the throughput) by temporary holding some of deflected flits at their current route positions. This way, in the network with dual-mode channels, the average hop count is reduced for 2.33 hops with respect to baseline, while the reduction for 4.52 hops has achieved with the side-buffering. As expected, the lowest average hop count of 8.14 hops

is achieved with the in-channel buffering, which represents a decrease of 5.07 inter-router hops per flit (or, 38.4%) with respect to the baseline.

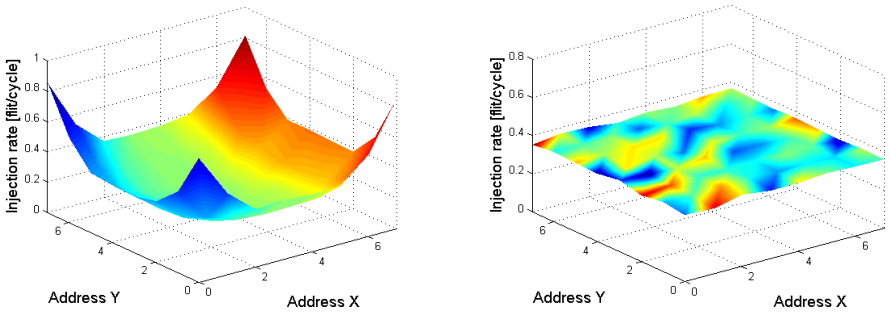
In the baseline architecture, the transport delay equals the average hop count because each hop (either productive or misrouting) takes one clock cycle. In the networks with a misrouting suppression support, the transport delay incurred by a flit is the sum of two components: the hop count and the time the flit spends blocked by a misrouting suppression mechanism. For example, each time the dual-mode channel activates the loop-back mode, it adds one clock cycle to the transport delay of the looped-back flits. However, since the loop-back saves two hops, the total transport delay is lower than in the baseline NoC. In difference to the dual-mode channel, deflected flits captured by the side-buffering or in-channel buffering mechanism may stay buffered at their current route positions for several clock cycles before they get a chance to make the next inter-router hop. A closer examination of the simulation statistics revealed that flits, while traveling toward their destinations, spend 2.32 clock cycles in the side-buffers on average, which is low enough to provide a 16.6% lower total transport delay than in the baseline network. On the other hand, with the in-channel buffering the average buffer delay is 4.85 clock cycles. The high buffer delay is the reason why the transport delay with the in-channel buffering is larger than in the baseline architecture, despite a significant reduction in hop count. Note that the in-channel buffering achieves a high saturation throughput even with a high transport delay. This is because buffered flits waiting to be looped-back do not block other flits that could otherwise make forward progress. Let note that the transport delay can be reduced by limiting the time (i.e. the number of clock cycles) that flits are allowed to spent in in-channel buffers – when the time limit is reached, the buffered flit is forced to loop-back, regardless of the routing status of the flit arriving from the opposite side of the channel. However, an inevitable consequence of such buffering policy will be reduction of the network throughput due to lower utilization of in-channel buffers. For this reason, we have excluded this design option from further consideration.

The results in Table 1 do not show significant difference in deflection rates between the baseline and NoCs architectures with the misrouting suppression support. This is because the same PAS stage (i.e. permutation network with silver flit arbitration policy) is used in all investigated NoC configurations. On the other hand, the misrouting rate depends not only on how often flits deflect, but it also depends on how efficiently the misrouting suppression mechanism prevents the deflected flits to make misrouting hops. The side-buffering technique reduces the misrouting rate by preventing some of deflected flits to leave the router. In this way, 51.5% of misrouting hops are prevented. The dual-mode channel uses the loop-back mode to return some of deflected flits back to their current routers. With this strategy, the dual-mode channel succeeds to prohibit about 19.36% of all deflected flits to make misrouting hops without adding extra buffers. By adding buffers into the dual-mode channels and optimizing the routing function of the deflection router, the proposed in-channel buffering technique reaches the misrouting suppression efficiency which is slightly higher than that of the side-buffering technique.

4.2. Injection fairness

As emphasized out in Section 3, the arrangement of stages within the side-buffered deflection router may create injection unfairness in the network, in sense that some PEs get to transmit more flits than others. This phenomenon can be best observed in Fig. 5a, which shows distribution of the injection rate (i.e. the number of flits injected by each PE per clock

cycle) over all PEs in the side-buffered deflection NoC under saturated load with uniform traffic pattern. As can be seen, the injection rate differences between PEs are significant: while corner PEs can inject their flits at almost every cycle, the PEs in the middle of the mesh get a chance to inject a flit on every tenth cycle. As shown in Fig. 5b, the in-channel buffering provides almost uniform injection rate distribution under the same load conditions. This advantage occurs because the in-channel buffering is transparent for the deflection router, which treats each incoming flit equally, regardless of whether the flit is looped-back by the in-channel misrouting suppression logic or it comes from a neighboring router.



a) b)

Fig. 5 Injection rate distribution under saturation load in deflection-routed 2D mesh NoCs with misrouting suppression support: a) side-buffering; b) in-channel buffering

4.3. Sensitivity to buffer size

The second set of simulations deal with the impact of buffer size on the effectiveness of the side-buffering and in-channel buffering techniques. Observed form Table 2, although increasing the buffer size improves the throughput and misrouting suppression efficiency under saturated traffic load, this improvement is relatively small and rapidly saturates. Doubling the buffer size from 1-flit to 2-flits increases the saturation throughput by only 2.71% for side-buffering, and 4.15% for in-channel buffering technique. In addition to high hardware cost, the price paid for this throughput improvement is 10% longer transport delay for side-buffering, and even 28% longer for in-channel buffering. Further increase of buffer size increases the saturation throughput only marginally, while the transport delay continues to steadily increase. These results suggest that buffers with size larger than 1 flit increases hardware complexity and wastes power without significant performance benefit.

Table 2 Comparison of saturation performance of baseline deflection-routed NoC architecture and architectures with misrouting suppression support

Buffer size	Side-buffering			In-channel buffering		
	th	*td*	μ_e	*th*	*td*	μ_e
1 flit	0.332	11.016	51.5%	0.361	14.541	52.3%
2 flits	0.341	12.126	57.2%	0.376	18.613	58.6%
3 flits	0.344	13.476	59.2%	0.382	22.899	61.2%
4 flits	0.346	14.915	60.0%	0.386	27.201	62.4%

4.3. Latency analysis

Finally, we evaluate the impact of different misrouting suppression schemes on the latency performance of deflection-routed network. Latency is defined as the time (in clock cycles) since the flit is generated at the source PE until it arrives at the destination PE, including the time the flit spends in the source PE's transmission queue. In these simulations, each PE generates flits following Poisson distribution with mean rate λ (λ is also called the average flit injection rate for the NoC). Generated flits remain in its queue until they are successfully injected to the network. For each network configuration, the flit injection rate is varied from zero to the point when the first transmission-queue in the network becomes saturated.

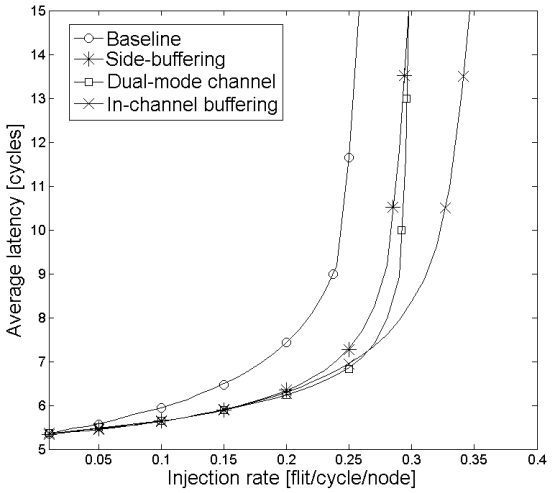


Fig. 6 Latency comparison of baseline deflection NoC architecture and architectures with misrouting suppression support under uniform traffic pattern

Figure 6 contains load-latency graph under uniform traffic pattern. As observed, at low injection rates, deflection-routed networks with the misrouting suppression support experience almost the same average flit latency as the baseline deflection network. This is because of the fact that the network is free from congestion. However, as load in the network increases, the effect of misrouting suppression technique adopted becomes more visible. The graph in Fig. 6 shows that the proposed in-channel buffering technique significantly improves the routing performance by providing low-latency communication at higher injection rates.

As can be observed in Fig. 6, for every deflection scheme, except for the side-buffering technique, the maximum injection rate achieved closely matches the saturation throughput reported in Table 1. This is because these schemes provide injection fairness so that all transmission queues in the network become saturated at approximately the same injection rate. On the other hand, in the side-buffered deflection network, the transmission queues of PEs in the middle area of the network become saturated at much lower injection rate than those of boundary PEs, leading to early saturation.

5. CONCLUSIONS

In this paper, a misrouting suppression technique for deflection-routed networks on chip was presented. The presented technique avoids misrouting hops by looping-back or capturing deflected flits into small in-channel buffers, immediately after they have appeared at router's output ports. The efficiency of the technique is further improved by modifying the routing function of deflection router in a way to prevent misrouted flits to take immediate reverse hops. The simulation results show that the proposed schemes improves performance of the baseline deflection-routed NoC by 36.2% in terms of saturation throughput. Results also show that the misrouting suppression with the in-channel buffering offers higher saturation throughput than with the in-router buffering (i.e. side-buffering) although with a penalty in terms of hardware cost. Moreover, the performance improvement is achieved without incurring injection unfairness among network nodes, which characterizes the side-buffering approach.

Acknowledgement: *This work was partially supported by the Serbian Ministry of Science and Technological Development Project No. TR-33035.*

REFERENCES

[1] W. J. Dally "Virtual-channel flow control", *IEEE Trans. Parallel Distributed Syst.*, 1992, vol. 3, no. 2, pp. 194-205.
[2] T. Bjerregaard, S. Mahadevan, "A survey of research and practices of network-on-chip", *ACM Comput. Surv*., vol 38, no. 1, 2006.
[3] A. Kumar, P. Kundu, A. Singh, L. S. Peh and N. Jha, "A 4.6 Tbits/s 3.6 GHz single-cycle NOC router with a novel switch allocator in 65 nm CMOS", In Proc. of 25th International Conference on Computer Design, ICCD, 2007, pp. 63-70.
[4] A. Kohler and M. Radetzki, "Fault-tolerant architecture and deflection routing for degradable NoC switches", In Proc. of the 3rd IEEE International Symposium on Networks-on-Chip, 2009, pp. 22–31.
[5] G. Michelogiannakis, D. Sanchez, W.J. Dally, C. Kozyrakis, "Evaluating bufferless flow control for on-chip networks", In Proc. of the 4th ACM/IEEE Int. Symposium on Networks-on-Chip, 2010, pp. 9-16.
[6] T. Moscibroda and O. Mutlu, "A Case for Bufferless Routing in On-Chip Networks", In Proc. of the 36th annual international symposium on Computer architecture, ACM, New York, 2009, pp. 196-207.
[7] C. Fallin, C. Craik and O. Mutlu, "CHIPPER: A low-complexity bufferless deflection router", In Proc. of the 17th International Symposium on High Performance Computer Architecture (HPCA), 2011, pp. 144–155.
[8] C. Fallin, G. Nazario, X. Yu, K. Chang, R. Ausavarungnirun and O. Mutlu, "MinBD: Minimally-Buffered Deflection Routing for Energy-Efficient Interconnect", In Proc. of the 6th IEEE/ACM International Symposium on Networks on Chip, 2012, pp. 1-10.
[9] I. Z. Stojanovic, M. D. Jovanovic and G. Lj. Djordjevic, "Dual-mode inter-router communication channel for deflection-routed networks-on-chip", *The Journal of Supercomputing*, Springer US, Published online: March 2015.
[10] Y. Li, K. Mei, Y. Liu, N. Zheng, Yi Xu, "LDBR: Low-deflection bufferless router for cost-sensitive network-on-chip design", *Microprocessors and Microsystems*, 2014, vol. 38, no. 7, pp. 669-680.
[11] M. Hayenga, "SCARAB: A single cycle adaptive routing and bufferless network", In Proc. of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO-42), 2009, pp. 244-254.
[12] J. Jose, B. Nayak, K. Kumar and Muyam M, "DeBAR: Deflection based adaptive router with minimal buffering", In Proc. of the Design, Automation & Test in Europe Conference & Exhibition (DATE), 2013, pp. 1583–1588.
[13] C. Feng, J. Li, Z. Lu, A. Jantsch, M. Zhang, "Evaluation of Deflection Routing on Various NoC Topologies", In Proc. of IEEE 9th International Conference on ASIC (ASICON 2011), pp. 163-166.
[14] Open SystemC Initiative. SystemC v2.1 Language Reference Manual, 2005. http://www.systemc.org/