Improved Deflection Routing Method for Bufferless Networks-on-Chip

Igor Stojanovic[1], Milica Jovanovic[2], Sandra Djosic[3] and Goran Djordjevic[4]

Abstract – **In conventional deflection routers, port-contention is resolved by misrouting packets. It leads to a significant performance loss at higher loads. To address this problem, we propose a simple link-control that allows a deflected packet to return back to its current router instead of being misrouted. Evaluations show that the proposed mechanism yields an improvement of 7 – 12% in network saturation throughput when coupled with recent bufferless deflection-based routers.**

Keywords – **Network-On-Chip (NoC), Deflection routing, multi-core, loopback.**

I. INTRODUCTION

With the constant growing complexity of modern System on Chip (SoC) architectures, the issue of interconnection becomes more important for overall system performances and efficiency. Incorporating multiple processors along with a wide range of IP (Intellectual Property) cores in a single SoC makes traditional bus-based interconnect architectures inefficient and difficult to use [1]. A *network on-chip* (NoC), based on traditional network routing, has been suggested as a scalable alternative [2]. With its scalability and regularity, NoC easy handles growing complexity, thus reducing both the design effort and time to market. It also offers significantly higher aggregated bandwidth and lower energy consumption compared to conventional bus-based systems, which makes NoC a competitive solution for using in low power SoC designs.

A NoC consists of multiple routers interconnected with each other using point-to-point physical channels (links) to form a suitable network topology [3]. Each router is also connected to an IP core which serves as a source and sink of data. Most current NoC designs employ wormhole packet switching in combination with either deterministic or minimal adaptive routing policies. To gain higher throughput and avoid potential deadlock situations, wormhole routers relay on virtual channel flow control mechanism such that the input buffer is organized as several independent flit buffers allocated to different packets. Although in-router buffering improves the bandwidth efficiency, virtual channel buffers draw a significant fraction of NoC power and area, and can increase router latency.

The need for power- and area-efficient on-chip interconnect infrastructure in modern many-core systems has recently initiated a new line of research in the field of NoC, which advocates the use of routers with minimum amount of internal storage [4-6]. In such *bufferless* NoCs, each flit of a packet is routed independently of every other flit, and deflection routing is used to avoid the need for in-router buffers. Since there is no buffers to store flits in transit, fits must pass through the router without waiting or stalling, and any port-contention between multiple arriving flits results in one flit being routed through the desired (i.e. productive) output port and others being deflected to another (i.e. non-productive) output ports. The high cost of buffers makes deflection routing attractive, especially for low-to-medium network loads. However, at high network load, the energy benefits of this bufferless scheme are offset by performance degradation [7]. This is because port-contentions occur more frequently, and each deflection sends a flit further from its destination causing unproductive network hops.

In this work, we present a simple extension to the basic deflection routing scheme, which allows some deflected flits to be returned back to the current router, instead of being transferred to the next. By preventing, when possible, unnecessary link traversals, the proposed mechanism is able to improve the performance of deflection routing under higher network loads. It is orthogonal to other techniques for optimizing bufferless router design, such as insertion of a small central buffer [8] and improving port-allocation logic [5], and it can be applied to any bufferless deflection NoC provided that neighboring routers are connected by full-duplex links (i.e., two unidirectional links).

II. BUFFERLESS DEFLECTION ROUTING

The basic deflection routing scheme mandates that all flits that arrive to a router at the current routing cycle must be switched to output ports and sent out immediately in order to make room for the new set of incoming flits. Consider two neighbouring routers shown in Fig 1. If router *A* decides not to send flit f_{AB} to router *B*, it could happen that in the next cycle it has to send out more flits than it has output ports, causing some of them to be dropped. Therefore, in a conventional bufferless deflection-based router design, neighbouring routers must exchange flits in every routing cycle, regardless of whether the flits make productive hops or not.

[1]Igor Stojanovic is PhD student at Faculty of Electronic Engineering, University of Nis, A. Medvedeva 14, 18000 Nis, Serbia, E-mail: igor.stojanovic@elfak.ni.ac.rs

[2] Milica Jovanovic is teaching assistant at Faculty of Electronic Engineering, University of Nis, A. Medvedeva 14, 18000 Nis, Serbia, E-mail: milica.jovanovic@elfak.ni.ac.rs

[3] Sandra Djosic is teaching assistant at Faculty of Electronic Engineering, University of Nis, A. Medvedeva 14, 18000 Nis, Serbia, E-mail: sandra.djosic@elfak.ni.ac.rs

[4] Goran Djordjevic is professor at Faculty of Electronic Engineering, University of Nis, A. Medvedeva 14, 18000 Nis, Serbia, E-mail: goran.lj.djordjevic@elfak.ni.ac.rs.

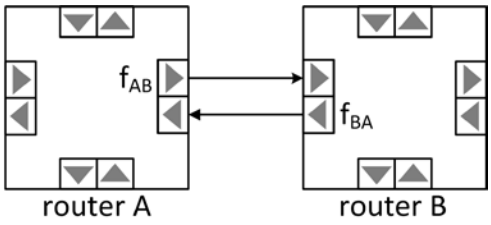


Fig. 1. Router-to-router connection with fixed full-duplex link.

The deflection routing scheme can be implemented on any network topology with the same number of input and output ports per router. Although there are a number of network topologies satisfying this constraint, we consider 2D mesh topology, only. The reason is that the 2D mesh is the most commonly used topology in NoC design due to its simplicity, regularity, and scalability. 2D mesh topology is composed of routers with five bidirectional ports. A router uses four ports to connect with its neighboring nodes (two per dimension, one in each direction). The fifth port is used by the local IP core to inject/eject flits (the minimal routable units of packets) to/from the network, respectively.

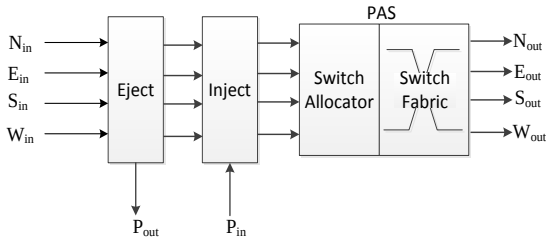


Fig. 2. Bufferless router organization.

Fig. 2 shows the basic bufferless deflection router organization. Bufferless deflection routing has three stages: Eject, Inject, and Port Allocation and Switching (PAS) stage. The eject stage removes one of locally-addressed flit (if any) from the router and forwards it to the local IP core. The inject stage injects a new flit form the local IP core if one of four inputs does not have a flit present in a given cycle. The Port Allocation and Switching (PAS) stage first maps the set of input flits to the set of output ports (task of Switch Allocator). The flits mapping tries to assign productive port to each flit in order to get flit routed in productive direction (the shortest path to the destination). Note that the number of productive ports assigned to a flit in the 2D mesh NoC can be: 0 (flit is addressed to the local router), 1 (flit is already at one of the axes of its final destination) or 2 (otherwise). After output ports are allocated, the input flits are actually moved to output ports through the Switch Fabrics. Note that the PAS stage actually makes the differences between the various bufferless NoC designs.

There are two representative bufferless router designs: BLESS [4] and CHIPPER [6]. BLESS router uses 4x4 crossbar switch controlled by an allocator unit. Output port allocation is performed sequentially, using oldest-first priority scheme. The scheme achieves relatively low deflection rate, but with high penalty in terms of hardware cost and latency. On the other side, CHIPPER greatly simplifies router architecture by replacing global allocator and crossbar with a partial two-stage permutation network that is composed of four arbiter block. Each arbiter block comprises 2x2 crossbar switch and allocator unit. Compared to BLESS, CHIPPER has much simpler hardware but also worse network performances.

There are several related works that try to improve deflection router design by offering a specific tradeoff between router complexity and performance. One prior work, AFC (Adaptive Flow Control) [9] proposes hybrid design that incorporates in-router flit buffers (as in buffered routers), but can switch it off under low network load and work as bufferless router. However it seems that under high network load, frequent buffer switching degrades network performance because buffers require noticeable time to switch on, which is followed by additional energy consumption.

Another prior design, minimally-buffered deflection (MinBD) router uses deflection routing but incorporates small flit in-router buffer called side buffer. When a flit is deflected to an unproductive output port, it can be temporary stored in the side buffer avoiding leaving the router. This preserves unnecessarily flit misrouting. In the next cycle, flit stored in the side buffer enters the network, if there is a free input slot, and participates the new contention.

III. DEFLECTION ROUTING WITH LOOP-BACK

The proposed solution is based on the following observation: If two adjacent routers agree not to exchange their flits, the number of flits contained in either of the two routers in the next cycle will not exceed the number of their output ports. Moreover, the exchange of flits can be temporarily prohibited over any subset of full-duplex links in the network without exceeding the allowable number of flits in any router. To exploit this property, we replace the fixed full-duplex links with links that support two different operating modes: the exchange mode, and the loop-back mode (Fig. 3).

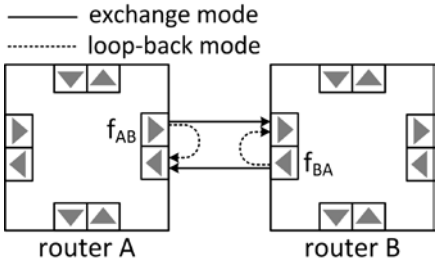


Fig. 3. Router-to-router connection with two mode full-duplex link.

The exchange mode allows the flow of flits as in the conventional deflection routing. In the loop-back mode, flits form output ports on both sides of the link are returned back to the corresponding input ports of their current routers.

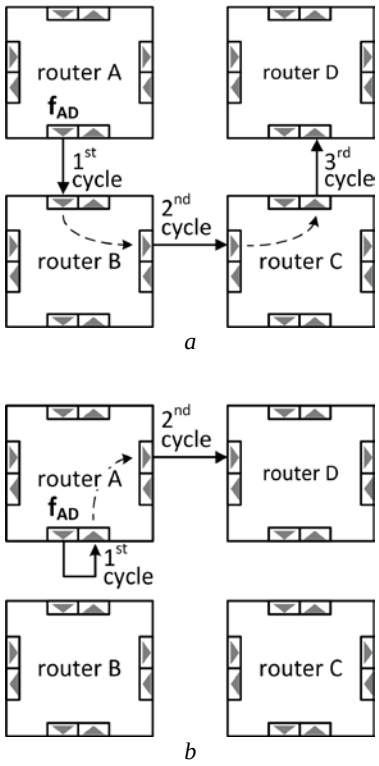


Fig. 4. Deflection overheads: a) Overhead due to flit misrouting; b) overhead due to flit loop-back.

The loop-back feature allows a deflected flit to avoid misrouting (i.e. being moved one hop further form its destination), thus conserving cycles needed to reach its destination. In fact, the loop-back operation reduces the cost of deflection for one routing cycle, as exemplified in Figs. 4*a* and *b*. Both figures show the path of flit f_{AD} from router *A* to its destination router *D* after it suffers deflection in router *A* toward router *B*. The path in Fig. 4*a* corresponds to the conventional deflection routing: once misrouted to router *B*, flit f_{AD} needs two additional productive cycles to reach its destination – router *D*. On the other hand, if flit f_{AD} is deflected-back to router *A*, it will be delivered to router *D* with one cycle of delay, as shown in Fig 4*b*.

Although the loop-back operation can be beneficial for deflected flits, it may disturb the transfer of productive flits. To prevent this, we formulate the following *flit-deflection rule*: if at least one output port on either side of the link holds a productive flit, the link is configured in exchange mode; otherwise, the link is configured in loop-back mode. According to this rule, a deflected flit will be misrouted only if there is a productive flit on the opposite side of the link.

IV. Implementation

Figure 5 shows the hardware implementation of the two-mode full-duplex link. At each side of the link, a simple link controller is appended, which regulates the transfer of flits between two routers. A flit is injected into the router via a two-input multiplexer, which passes either the flit coming from the output port of the same router (i.e. loop-back mode), or the flit sent by the opposite router (i.e. exchange mode). To implement the flit-deflection rule, each router's output port need to be extended with a flag (denoted as *p* in Fig. 5) indicating the routing status of the flit that is present on that port. The flag *p* is set to '1' if the corresponding output port is occupied by a flit and it is the productive port for the flit. Otherwise, if the output port does not hold a flit, or the flit is routed in a non-productive direction through that port, the value of *p* is '0'. The loop-back mode is selected only if both routers indicate non-productive routing status, i.e. the productivity flags of both routers are set to '0'. Note that in Fig. 3 the link controllers are shown as blocks separated from routers. In fact, they can be placed either outside or inside of the router, thus minimally impacting the regular router microarchitecture.

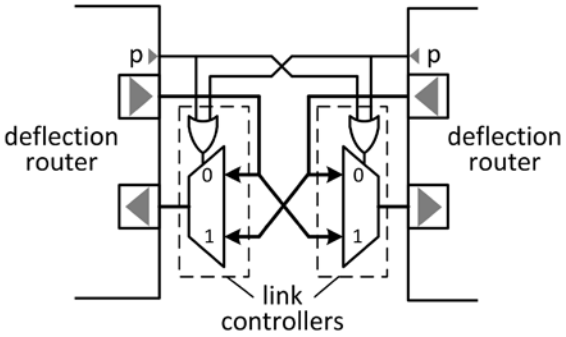
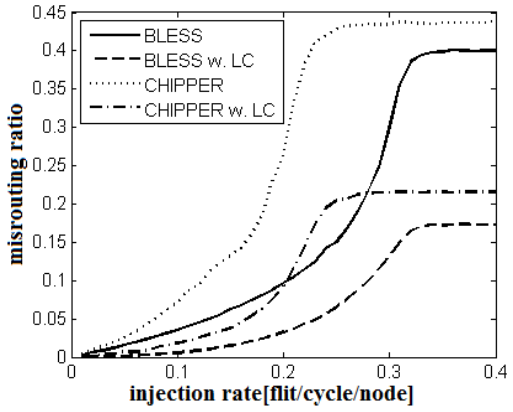


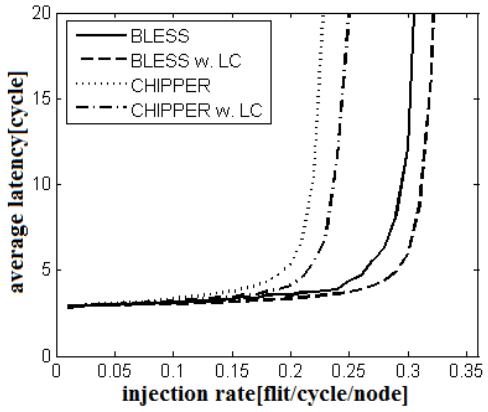
Fig. 5. Implementation of two-mode full-duplex link.

V. Evaluation

We have evaluated our proposed link control mechanism using an in-house cycle-accurate NoC simulator developed in SystemC. We have simulated two 2-D mesh NoCs with size of 8x8 nodes, which are built using a single-cycle flit-level models of two conventional bufferless deflection router architectures: BLESS [4], and CHIPPER [6]. These routers differ mainly in port allocation policy. BLESS router forwards incoming flits to output ports through a full 4x4 crossbar switch by giving higher priority to older flits. On the other hand, CHIPPER router employs a partial permutation network of four 2x2 crossbars and random flit priorities to accomplish the same task. The only design parameter varied between simulations is the flit-deflection rule, i.e. the link configuration with or without inserted link controllers. We have used synthetic traffic with flits addressed in uniformly random manner. Flits are generated and injected into the network following a Poisson distribution. In all our simulations, the flit injection rate (i.e. the average inter-arrival time of flits at injection port of each router) is swept from zero to network saturation.



a



b

Fig. 6. Performance of deflection NoCs with fixed and two-mode full-duplex router-to-router links: a) Misrouting ration vs. injection rate; b) Average latency vs. injection rate

To quantify how efficiently the link controllers suppress flit misrouting we introduce a metric, named *misrouting ratio*, which is defined as the total number of non-productive hops divided by the total number of hops taken by all the flits injected into the network. From Fig. 6*a*, we can observe that the proposed mechanism significantly reduces the misrouting ratio in both networks. The relative improvement is greater at low loads, since almost all flit deflections can be handled with the loop-back operation. As injection rate increases, it becomes more common that a link controller must exchange a deflected flit with the productive flit on the opposite side of the link, which increases the misrouting ratio. However, even at saturation load, link controllers reduce the misrouting ratio for 57% in BLESS-NoC, and for 51% in CHIPPER-NoC network. By avoiding some non-productive hops, the proposed link control mechanism is able to recover some performance loss due to deflections. From the Fig 6*b*, we can see that the proposed mechanism reduces the average flit latency, and improves the network throughput at high injection rates. As shown in Table 1, the proposed mechanism provides a 7% higher saturation throughput for BLESS-NoC, and 12% higher for CHIPPER-NoC network.

TABLE I
NOC THROUGHPUT COMPARISON

BLESS	BLESS with link controllers	CHIPPER	CHIPPER with link controllers
0.327	0.351	0.242	0.271

VI. CONCLUSION

In this work we have introduced a link-control strategy for reducing overhead of flit deflection in bufferless networks-on-chip. The results presented show that the router-to-router connection with two-mode full-duplex links improves network throughput with an area overhead of a two-input flit-wide multiplexer per router's input port.

ACKNOWLEDGEMENT

This work was partially supported by the Serbian Ministry of Science and Technological Development Project No. TR-32009, TR-33035 and Project Grant III44004.

REFERENCES

[1] S. Borkar: "Future of interconnect fabric: a contrarian view", Proc. 12th ACM/IEEE international workshop on System level interconnect prediction, 2010, pp. 1-2

[2] W. Dally, W. James, and B. Towles: "Principles and practices of interconnection networks", Elsevier, 2004

[3] W. Dally, W. James, and B. Towles: "Route packets, not wires: On-chip interconnection networks.", Proc. Design Automation Conference, IEEE, 2001, pp. 684-689

[4] T. Moscibroda, and O. Mutlu: "A Case for Bufferless Routing in On-Chip Networks", Proc. 36th International Symposium on Computer Architecture, 2009, pp. 196–207

[5] M. Hayenga: "SCARAB: A single cycle adaptive routing and bufferless network". Proc. 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO-42), 2009, pp. 244-254

[6] C. Fallin, C. Craik, O. Mutlu: "CHIPPER: A low-complexity bufferless deflection router". Proc. 17th International Symposium on High Performance Computer Architecture (HPCA), 2011, pp. 144–155

[7] G. Michelogiannakis, D. Sanchez, W. J. Dally, and C. Kozyrakis: "Evaluating bufferless flow control for on-chip networks". Proc. 4th ACM/IEEE International Symposium on Networks-On-Chip, 2010, pp. 9–16.

[8] C. Fallin, G. Nazario, X. Yu, K. Chang, R. Ausavarungnirun, and O. Mutlu: "MinBD: Minimally-Buffered Deflection Routing for Energy-Efficient Interconnect". Proc. 6th IEEE/ACM International Symposium on Networks on Chip, 2012, pp. 1-10

[9] S. Jafri, Y. Hong, M. Thottethodi, and T. Vijaykumar: "Adaptive flow control for robust performance and energy", Proc. 43rd Annual IEEE/ACM International Symposium on Microarchitecture , 2010, pp. 433-444