Switch Folding: Network-on-Chip Routers with Time-Multiplexed Output Ports

G. Dimitrakopoulos
Electrical and Computer Engineering
Democritus University of Thrace
Xanthi, GR67100, Greece

N. Georgiadis and C. Nicopoulos
Electrical and Computer Engineering
University of Cyprus
1678 Nicosia, Cyprus

E. Kalligeros
Information and Comm. Systems Eng.
University of the Aegean
Samos, GR83200, Greece

***Abstract*—On-chip interconnection networks simplify the increasingly challenging process of integrating multiple functional modules in modern Systems-on-Chip (SoCs). The routers are the heart and backbone of such networks, and their implementation cost (area/power) determines the cost of the whole network. In this paper, we explore the time-multiplexing of a router's output ports via a folded datapath and control, where only a portion of the router's arbiters and crossbar multiplexers are implemented, as a means to reduce the cost of the router without sacrificing performance. In parallel, we propose the incorporation of the switch-folded routers into a new form of heterogeneous network topologies, comprising both folded (time-multiplexed) and unfolded (conventional) routers, which leads to effectively the same network performance, but at lower area/energy, as compared to topologies composed entirely of full-fledged wormhole or virtual-channel-based router designs.**

I. Introduction

Modern computing devices, ranging from mobile phones and tablets to powerful servers, rely on complex silicon chips that integrate a huge number of computational, storage, and communication elements. The design of such systems is not an easy task. Efficient design methodologies are needed that organize the designer's work and reduce the risk of an inefficient system [1].

One of the main challenges that the designer faces is how to connect the components inside the silicon chip, both physically and logically, without compromising performance. The Network-on-Chip (NoC) paradigm tries to answer this question by applying at the silicon-chip-level various well established networking principles, after suitably adapting them to the silicon chip characteristics and application demands [2].

The routers are the heart and the backbone of the NoC. Their main function is to route data from network sources to network destinations. They achieve this goal by providing arbitrary connectivity between their input and output ports, thus allowing for the implementation of arbitrary network topologies. The nodes of the system are attached to the NoC via a network interface (NI) that handles any protocol bridging and data (de)packetization.

NoC routers follow roughly the architectures depicted in Fig. 1 [3]. Fig. 1(a) shows a typical Wormhole (WH) router. The routing computation logic un-wraps the headers of incoming packets and determines their output destination. Such decoding and routing computation can be prepared in the previous switch and used in the current one. This optimization is called look-ahead routing (LRC) and allows the routing computation to be performed in parallel with the rest of the tasks [4]. At the same time, a packet's header competes for the selected output port, since the other input queues may have a request for the same output port. If the packet header wins this stage, called switch allocation (SA), it will traverse the crossbar (ST – switch traversal), and, one cycle later, it will travel on the output link (LT – link traversal) towards the adjacent switch. SA and ST can be performed concurrently when employing merged arbiter multiplexer structures [5].

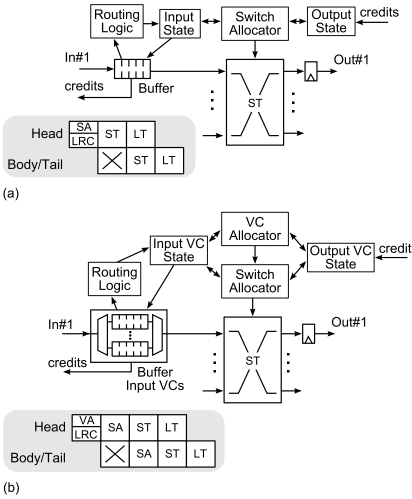


Fig. 1. Two typical NoC router architectures and their pipelines: (a) 3-stage wormhole (WH), and (b) 4-stage switch based on virtual channels (VC).

Both LRC and SA are performed only for the head flit of each packet. The remaining body and tail flits will follow the same route that has already been reserved by the head flit. Therefore, in a WH router, if a packet at the head of a queue is blocked, either because it loses access to an output port, or because the downstream buffer is full, all packets behind it also stall. This head-of-line (HoL) blocking problem can be solved by separating each input buffer into multiple parallel queues. Each queue is called a virtual channel (VC) and allows packets from different queues to bypass each other and advance to the crossbar, instead of being blocked by a packet at the head of the queue [6]. Due to the presence of multiple VC queues in each input, each packet has to choose a VC at the input of the adjacent router (known as an "output VC" from the perspective of the current router) before the SA stage. Matching input VCs to output VCs is performed by the VC allocator (VA).

The work of the Univ. of Cyprus was supported by the Cyprus Research Promotion Foundation's Grant ΤΠΕ/ΠΛΗΡΟ/0609(ΒΙΕ)/09 (co-funded by the Republic of Cyprus and the European Regional Development Fund).

978-3-9815370-0-0/DATE13/©2013 EDAA

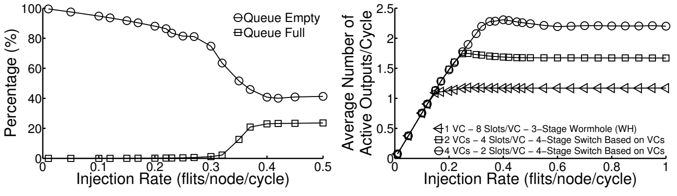
Typically, the VCs of each input share a common input port of the crossbar via a local multiplexer (see Fig. 1(b)). Although this is enough for most cases, it significantly complicates the SA stage relative to a WH router. In VC routers, SA is organized in two phases, since both per-input (local) and per-output (global) arbitrations are needed. Even though the per-input and per-output arbiters operate independently, their eventual outcomes in SA are very much dependent, with each one affecting the aggregate matching quality of the router [7].

When VCs are used for performance, they offer a clear throughput benefit over WH routers. The downside, however, is the increased complexity and latency exhibited by such VC-based routers, as compared to WH routers, assuming the same clock frequency. The lower latency of WH routers may prove beneficial to latency-critical applications. When the system requires a mechanism for traffic isolation, or deadlock avoidance, then either VC-based routers should be used – following the architecture shown in Fig. 1(b) – or multiple physical networks of WH switches can be employed.

A. Motivation

Inside each router, the main energy/area contributors are the buffers and the crossbar. At the network level, the links also constitute a significant part of overall power consumption. Buffers can either be register-based, or built inside SRAM blocks that improve area and density, but incur latency overhead [8]. Low-cost networks are built with small register-based shallow buffers, sized just to cover the round-trip credit delay of the link [9]. Although buffers are expensive in terms of both power and area, they are not utilized well. Low buffer utilization is demonstrated in the diagrams of Fig. 2(a), for a VC-based router with 4VCs/input and 4 buffer slots per VC in an 8×8 2D mesh network employing XY dimension-ordered routing. At low loads, most of the buffers are empty and a significant amount of them remain empty even at higher loads.

The first reaction of the designers to low buffer utilization is to employ some form of buffer sharing that would allow the always-empty queues to share their empty space with the full buffers, thus increasing NoC efficiency. This approach can start from sharing the VC buffer space among all VCs of a single input port [10], or it can move to complete sharing of the buffering among different input ports, transforming the input-buffered switch shown in Fig. 1 to a shared-memory one [11].



(a) (b)

Fig. 2. (a) Full/empty queue percentages, and (b) average number of active output links per cycle for different number of VCs in an 8×8 2D mesh.

Low buffer utilization also translates to low output utilization. When the network operates at low loads, only a few inputs have valid flits, and, consequently, only a few outputs are utilized. This is the main argument for proposing bufferless networks [12] and other low-latency switch designs that involve speculative switch allocation [13]. Similarly, at high loads, the utilization of the output ports is also small. This happens not because there are no available flits at the inputs, but because the majority of the output ports are blocked due to back-pressure flow control. In other words, a full input buffer blocks the upstream output port, which inevitably remains idle waiting for a credit to return.

Therefore, only a small fraction of the output ports are actually utilized in each clock cycle (both at low and high loads), as shown in Fig. 2(b). For the first time – to the best of our knowledge – we would like to explore this characteristic of low output utilization and design low-cost routers that involve fewer arbiters and crossbar output ports (multiplexers) than the actual output ports of the baseline routers. Switching different flits to the various outputs of the router is time-shared, by re-utilizing the available hardware (datapath) resources. We call this approach *switch folding*. Note that switch folding and buffer sharing are orthogonal techniques that can be used together for further reduction of the cost per switch. The main contributions of this paper can be summarized as follows:

- The introduction of switch-folded routers, which yield significant area/energy benefits over unfolded (conventional) routers, and the exploration of their design space.
- The evaluation of heterogeneous topologies that employ folded switches in parts of the network, thereby offering the same performance in terms of latency and throughput, while reducing the overall energy/area cost.

Previous work on reducing the area and power of the crossbar relied on *static* techniques, such as decomposition and segmentation [14], [15]. Decomposition restricts connectivity between certain input-output ports, which can also lead to network routing restrictions. Segmentation utilizes only parts of the crossbar under certain traffic conditions, without reducing the crossbar's hardware requirements and area. Switch folding reduces the hardware requirements and can *dynamically* serve all output ports without any connectivity restrictions.

The rest of the paper is organized as follows: Section 2 describes the design details of switch folding. Section 3 introduces the heterogeneous topologies employed in this work, while the experimental results are shown in Section 4 with analysis and comparison against baseline routers. Conclusions are drawn in Section 5.

II. Switch folding

The operation of switch folding is better understood via two simple examples that also highlight the cycle-by-cycle differences between the proposed and a conventional router. The conventional WH router involves a full crossbar and switch allocator that can support any input-output connection in each cycle, provided that only one input is connected to each output port. The baseline datapath for a WH router with switch folding involves only one arbiter and multiplexer module, which will be re-used in a time-shared fashion to serve all output ports of the router.

An example of the operation of both routers is shown in Fig. 3(a). In the conventional case (top), the first flits of packets A and B, which are buffered at inputs 0 and 1, respectively, can reach their corresponding outputs 0 and 2 in the first cycle, since they do not experience any contention. In the second cycle, the second flit of A will also move to output

2. Thus, the internal datapath is utilized by 2/3 in the first cycle and by 1/3 in the second one. This translates to almost 50% of average internal datapath utilization. Of course, higher datapath utilizations can occur when all input buffers have flits to send, output ports are free to accept new flits, and the flits themselves do not contend for the same resources.

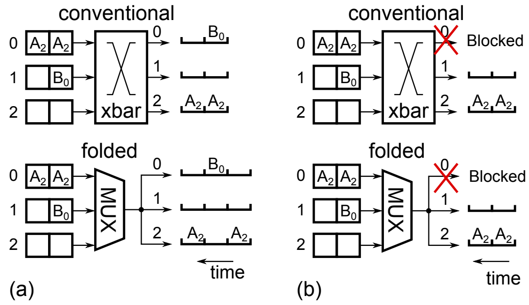


Fig. 3. Two conceptual examples of the operation of a conventional router (top), as compared to a switch-folded router (bottom).

On the contrary, a switch-folded router (bottom of Fig. 3(a)) would need 3 cycles to switch the same number of flits. In the first cycle, where the switch serves – for instance – output 2, the flit from input A will leave the router. Arbitration and multiplexing will be performed only for output 2, transferring to the output the first flit of packet A. This flit will be broadcast with a direct wire connection (no extra logic) to all outputs, but, it will be registered only at output 2. This is done via the control logic of the switch-folded router that keeps track of which output is served in each cycle and asserts the corresponding flip-flop-enable signal. In the next cycle, the folded switch may continue to serve output 2, since it has more flits to send, or it can change output and serve output 0, following a cyclic serve order. The latter choice is the one made in the example of Fig. 3(a). In the second cycle, the flit from input 1 will move to output port 0, following the same procedure. If all outputs are served cyclically, the next cycle should be devoted to output 1. However, since no input requests output 1, it can be skipped so as to serve output 2 again, thus delivering the second flit of A in the 3rd clock cycle. In all three clock cycles, the internal datapath is fully (100%) utilized, since it delivers a new flit every cycle.

An output can be skipped when there are no requests for this output, as shown in the example of Fig. 3(a), or when it is blocked by link-level flow control, as shown in the next example (Fig. 3(b)). In this case, we assume that output 0 is blocked due to back-pressure, both for the conventional and the proposed time-shared router. In this case, which appears very often at high loads, both routers behave exactly the same in terms of cycles needed to serve the incoming flits. Therefore, in such cases, the folded switch offers the same transfer rate of flits to the outputs, achieving again 100% internal datapath utilization. The same behavior appears even at low loads in the case of heavily directed traffic (hot-spot), where many inputs want to access the same output port and the arbiter is obliged to serialize their accesses in time.

Switch folding does not change the per-packet operation of the switch. For example, in the case of wormhole switching, the arbitration decisions that are taken once per packet are kept fixed until all the flits of the packets pass to the corresponding output. This also happens in routers with switch folding, where the arbitration decisions do not change every time the router serves a new output. The folded datapath remembers which decision was taken in the past for each output. If no other output is utilized, the flits of the same packet leave the switch un-interrupted in consecutive cycles. If other outputs have valid traffic, then the flits of the same packet appear on the output link with some idle cycles between them. However, when an output is blocked due to back-pressure, another ready output is selected.

The area/power benefits of the folded switch are evident, since the majority of the crossbar's logic and wiring are missing. Additionally, the folded switch can increase the router's clock frequency. This is attributed to the lower fanout seen by the output of the input buffers and the increased layout density that shortens wire connections. With the folded switch, the output of each input buffer drives only one arbiter/multiplexer pair, which is N times less than an N-output router.

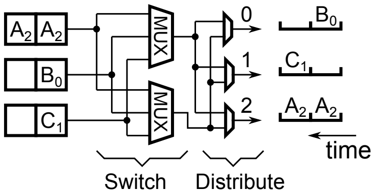


Fig. 4. An example of a switch-folded router employing 2 switch elements.

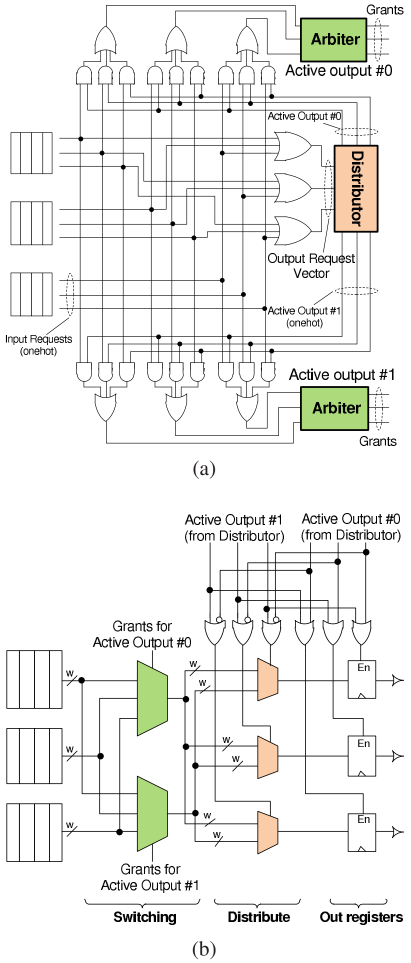
We can tradeoff some of the speed benefit offered by the reduced fanout and add a second switch element per router, as shown in Fig. 4. In this case, the router is able to deliver two independent flits to any two available outputs. The internal datapath of the router now consists of two arbiters and multiplexers that prepare two output results. These two output results are actually distributed to all outputs. In this case, since two results arrive per output, one 2-to-1 multiplexer is needed per output to distinguish between the two. The selection lines of these multiplexers are pre-driven by the selection logic of the folded switch, and are prepared beforehand, together with the output enable signals that inform each output of the arrival of a new flit. From our experiments, this configuration of 2 switching elements (arbiter and mux) per router with an additional multiplexer per output has the same delay as the conventional router that employs a full crossbar. However, the folded router achieves this with a lot less area.

The routers with switch folding in Figs. 3 and 4 do not impose any limitation on the routing algorithm, or the topology of the network, and can dynamically switch at the output a certain number of flits (1 or 2), by utilizing a lower-cost folded datapath.

A. The folded-switch datapath and control

The basic operation of a switch allocator for both WH and VC-based routers that employ switch folding with 2 switch elements should be slightly altered in order to identify efficiently which 2 outputs will be served in each cycle. The same procedure can be generalized for more outputs.

The modified switch allocator of a switch-folded router is shown in Fig. 5(a). In the switch allocation stage, we first form the Output Request Vector. This vector contains a 1 at a certain bit position, if the corresponding output is requested



(a)

(b)

Fig. 5. The organization of (a) the folded switch allocator, and (b) the folded datapath for a folded router with 2 switch elements.

by at least one flit stored at the head-of-line position of the input buffers/VCs. This output request vector is passed to a 2-output distributor, built from simple cells, which decides which 2 outputs will receive new flits in the next cycle. For the distributor, we employ the circuit proposed in [16], after appropriately adding to it some extra priority state, in order to keep a round-robin order in the outputs' activation. [1]

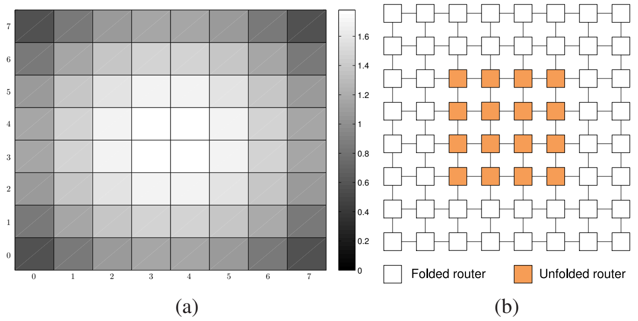
Each Active Output bit vector – derived from the distributor circuit – declares which output will be active in the next cycle, and it is used to (a) mask the input requests that refer to different outputs from those already selected, and (b) to configure the distribution of the results of the folded switch to the outputs of the router during the ST stage (see Fig. 5(b)). The pair of masked requests are passed to the only 2 available arbiters of the folded switch, which choose the winning inputs that will send their flits to the 2 active output ports. The decisions of the 2 arbiters, along with the one-hot bit vectors that declare which outputs will be active in the next cycle (the Active Output vectors), are passed to the next pipeline stage (ST), which handles the folded switch traversal.

As shown from the examples of Figs. 3 and 4, the folded switch traversal is organized in two steps. In the first step, only 1 or 2 switch elements (wide multiplexers) are used, and, in the next step, the one or two results are distributed to the appropriate outputs. The datapath of a folded switch with 2 switch elements is shown in detail in Fig. 5(b). The multiplexers of the switch elements are configured by the arbiter's decision produced in the previous stage. On the contrary, the output multiplexers are controlled by the Active Output bit vectors, which are also used to generate the enable signals for each output register.

III. Heterogeneous topologies

Switch folding is a time-multiplexed variant of the crossbar switching logic, which enables full routers to be implemented at a lower cost. However, depending on the symmetry of the topology and the characteristics of the router algorithm, some parts of the network are more stressed than other parts with lower utilization. For example, it is well known that the 2D mesh center needs to handle more traffic demand than its periphery. This conclusion is quantified by the heat map shown in Fig. 6. As can be seen, for the simulated conventional (i.e., comprising only unfolded routers) 8×8 mesh, the four central nodes have the greatest number of active links/cycle (i.e., maximum utilization), whereas the routers' output link utilization is gradually reduced as we move away from the mesh's center. Similar figures can also be found in [17].

As a consequence of the above experiment and discussion, we argue that switch folding with 1 or 2 outputs served per cycle can be applied to the majority of the network, while other heavily utilized parts can still employ routers with a full crossbar. This approach leads to heterogeneous topologies. We have to note here that switch folding offers a new form of heterogeneous network topologies, in which the low-cost routers do not have fewer VCs, shorter buffers, and/or smaller widths, but, instead, they are time-multiplexed.



(a)

(b)

Fig. 6. (a) The number of active output links per cycle in an 8×8 2D mesh under uniform random traffic, for a VC-based router with 4 VCs/input port and 4 flit slots/VC at an input load of 0.2 flits/node/cycle; (b) The square heterogeneous topology.

Motivated by the heat map of Fig. 6(a), we make use of the square heterogeneous 2D mesh topology shown in [17] (see Fig. 6(b) for convenience), but in its folded/unfolded variant, as a means to tradeoff area/energy efficiency offered by switch folding and possible throughput losses by time sharing of the outputs. Conventional routers are used at the center of the mesh, whereas the remaining routers use switch folding. Other configurations are possible, but our experiments have shown that they do not offer any significant benefit.

As a final remark, note that the use of folded routers and the employment of heterogeneous topologies based on them

[1] We use only 2 columns of the (load) distributor of [16], which are always enabled, and, in place of the deflection requests, we add the output request vector.

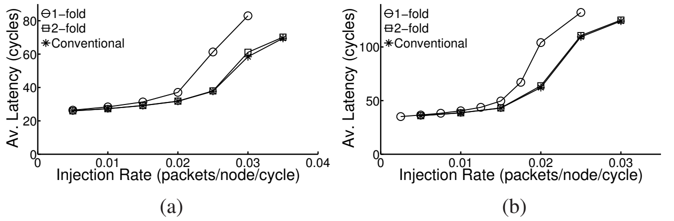
are orthogonal approaches. As it will be demonstrated in the Evaluation Section, homogeneous topologies consisting of only folded routers are sufficient for NoCs with up to a number of VCs per router input port, whereas for more VCs, heterogeneous topologies should be utilized.

IV. Evaluation

In this section, we evaluate the folded-switch micro-architecture against conventional input-queued routers, using an 8×8 2D mesh network with XY dimension-ordered routing and the GARNET cycle-accurate NoC simulator [18] that models all micro-architectural components of a NoC router. The proposed folded-switch micro-architecture was also cycle-accurately implemented within GARNET. Synthetic traffic pattern results from *uniform random* and *bit-complement* traffic are presented. Other permutation traffic patterns, such as *tornado* and *transpose*, follow trends very similar to bit-complement traffic, with respect to the relative performance of conventional routers and routers with switch folding. Hence, these results are omitted for brevity.

For all router configurations under comparison – both WH and VC-based routers – we use the pipeline organization shown in Fig. 2, assuming 8 buffer slots per input in all configurations: (a) 8 slots per input for a WH router; for VC-based routers: (b) 2 VCs per input with 4 slots/VC, and (c) 4 VCs per input with 2 slots/VC. The injected traffic consists of two types of packets to mimic realistic system scenarios [9], [19]: 1-flit short packets (just like request packets in a chip multi-processor), and longer 5-flit packets (just like response packets carrying a cache line). For the latency-throughput analysis, we assume a bimodal distribution of packets with 50% of the packets being short, 1-flit packets, and the rest being long, 5-flit packets. This percentage is in accordance with recent studies of cache traffic in chip multi-processors running real application workloads [9], [19].

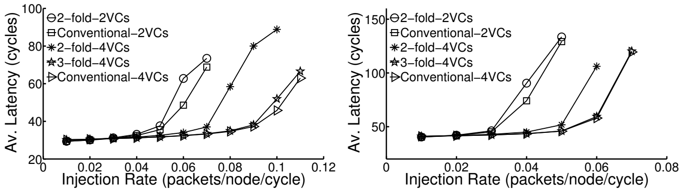
In all experiments, except for the last one, we assume that the routers with switch folding utilize the square heterogeneous topology (see Section 3), where 25% of the routers in the center of the mesh are conventional routers, while the rest are folded routers employing the significantly lower-cost folded datapath. This option (using the heterogeneous square topology with 1/4 conventional/folded router ratio) stems from our intention to have a good balance between area/energy savings and network performance. Results for homogeneous folded networks will be provided in our last set of experiments.



(a) (b)

Fig. 7. A comparison of average network latency (in cycles) vs. injection rate, between WH folded and unfolded routers, under (a) uniform-random and (b) bit-complement traffic.

Fig. 7 compares the folded WH routers inside the heterogeneous square topology with the conventional WH routers participating in a homogeneous 2D mesh. For the folded switches, we assume k switch elements per router (k-fold). Clearly, the 2-fold router behaves almost identically to the conventional router and achieves the same latency, both at low and high loads. This conclusion holds for both uniform-random traffic and for bit-complement traffic. The 1-fold case is sufficient at low loads, but it saturates earlier than the conventional router. From additional experiments, we verified that this trend of the 2-fold WH router (when applied to the heterogeneous square topology) is followed even with smaller, or larger, buffers in each input port.



(a) (b)

Fig. 8. A comparison of average network latency (in cycles) vs. injection rate, between VC-based folded and unfolded routers, under (a) uniform-random and (b) bit-complement traffic.

Next, we proceed with an assessment of the behavior of VC-based routers, which follow the switch-folding paradigm relative to conventional VC-based routers. The results are depicted in Fig. 8. At low loads, all alternatives with equal buffers per input port behave exactly the same. The routers with 2 VCs/input saturate earlier than the rest and their performance is almost indistinguishable. The folded router with 2 VCs exhibits slightly higher delay close to saturation, as compared to the conventional routers with 2 VCs. Both the folded router and the conventional one with 4 VCs/input saturate at higher loads, with the conventional case offering more throughput, which is almost the same as the 3-fold router.

In the last set of experiments, we aim to quantify the tradeoff arising from the chosen topology and the number of VCs when considering only routers with folded switches. The results gathered for 2-fold routers are shown in Fig. 9. We measure the latency of two network topologies: the square topology, whereby conventional routers are placed at the center of the network, and a homogeneous 2D mesh, where each router is 2-folded. We vary the input load and the number of VCs per router. The results reveal that networks with high number of VCs are more sensitive to the chosen topology, necessitating the adoption of the heterogeneous square topology for high throughput. In the case of WH routers, or routers with 2 VCs, a homogeneous 2D mesh behaves almost as well as the square topology, thus allowing for more cost reductions. Moreover, the results of Fig. 9 demonstrate that it is advantageous to use folded routers with 4 VCs in a homogeneous fully-folded topology, as opposed to a network with 2 VCs per input port using the square topology. Our experiments show that this trend is followed even for a larger number of VCs, i.e., the homogeneous fully-folded topology with 8 VCs per input port is better than the square topology with 4 VCs.

A. Hardware analysis

The area/energy benefits of the folded routers arise predominantly from the reduction of the complexity of the crossbar, and, to a lower degree, from the removal of per-output

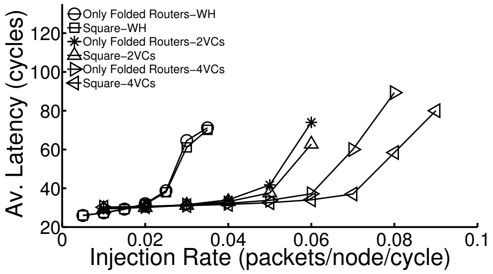


Fig. 9. A comparison of average network latency (in cycles) vs. injection rate, between homogeneous and heterogeneous topologies using 2-folded routers, under uniform-random traffic.

arbitration logic. The 1-fold router utilizes only 1 of the 5 ports of the crossbar – in the case of a 2D mesh network – while the 2-fold router re-uses 2 output ports of the crossbar on a cycle-by-cycle basis. Of course, in the case of the 2-fold router, some additional hardware is also spent in the distribution network, which consists of one 2-input multiplexer per output port. Based on our synthesis and layout results (using a 65 nm standard-cell library), 1-fold routers reduce the crossbar's area by more than 80%, while the 2-fold routers offer a reduction of 45%. In both cases, besides the actual reduction of the hardware dedicated to the crossbar (i.e., the number of multiplexers), additional area benefits are gained due to the improvement in the layout density of the circuit. The layout density is improved by the reduced wiring connections between the inputs and the crossbar's multiplexers, as a result of switch folding. Such gains are more pronounced in the case of wide datapaths of 128 bits, or more. The energy gains follow a similar trend in terms of dynamic power, but are significantly higher in terms of leakage (static) power, due to the reduced crossbar hardware per router.

The actual area/energy benefits of the folded routers depend on how much the crossbar contributes to the area/energy of the whole router. On low-cost routers with low buffer-count per input port, the crossbar plays an important role and it consumes more than 35% of the router's area [20]. In high-end routers with increased buffer count, the area dedicated to the crossbar is less than 20% [21]. Therefore, depending on the implementation, switch folding may save between 16% and 28% of the area budget in the 1-fold case, and between 9% and 15% in the 2-fold case. Heterogenous topologies (that make partial use of unfolded routers) enjoy less area savings, but without sacrificing performance.

The switch-folded routers follow exactly the same pipeline organization as the conventional routers. In order to quantify the delay overhead imposed by the distributor module to the folded switch allocator (see Fig. 5(a)), we synthesized the proposed circuit using a 65 nm standard-cell library. When the synthesizer is constrained for high-clock frequencies, the delay of the folded switch allocator is roughly 11% longer than the delay of a conventional switch allocator. Under less strict delay constraints, this delay overhead is hidden and traded off for some extra area (larger gate sizes) in the switch-folded implementation, which is perfectly acceptable due to the already reduced area of the folded design. The switch traversal stage does not increase the delay; the delay saved by reducing the fanout of the input buffer stage is given back to the distribution multiplexers at the output ports of the router.

V. Conclusions

The observed underutilization of the output ports of NoC routers – both at low and high traffic loads – has motivated us to explore the potential of routers with time-multiplexed output ports, i.e., switch-folded routers. Switch folding is explored for the first time, and it opens up a new direction for cost reduction. Our results demonstrate that switch folding can achieve similar, or nearly identical, network performance with conventional (unfolded) router designs, albeit at a lower area/energy cost. The magnitude of the reaped benefits hinges on the performance-cost tradeoffs arising from key design decisions, such as flow control policies (buffered/buffer-less) and the network topology.

References

[1] J. Handy, "NoC interconnect improves SoC economics," *Objective analysis - Semiconductor market research*, 2011.

[2] W. J. Dally and B. Towles, "Route Packets, Not Wires: On-Chip Interconnection Networks," in *Proc. of the 38th Design Automation Conference (DAC)*, Jun. 2001.

[3] M. Azimi, D. Dai, A. Mejia, D. Park, R. Saharoy, and A. S. Vaidya, "Flexible and adaptive on-chip interconnect for tera-scale architectures," *Intel Technology Journal*, vol. 13, no. 4, pp. 62–77, 2009.

[4] M. Galles, "Spider: A high-speed network interconnect," *IEEE Micro*, vol. 17, no. 1, 1997.

[5] G. Dimitrakopoulos et al., "Merged Switch Allocation and Traversal in Network-On-Chip Switches,", to appear in *IEEE Trans. on Computers*.

[6] W. J. Dally, "Virtual-Channel Flow Control," in *Proc. of the Intern. Symp. on Computer Architecture*, May 1990, pp. 60–68.

[7] D. U. Becker and W. J. Dally, "Allocator implementations for network-on-chip routers," in *Proc. of the Intern. Supercomputing Conf.*, 2009.

[8] P. Salihundam et al., "A 2Tb/s 6x4 Mesh Network with DVFS and 2.3Tb/s/W router in 45nm CMOS," in *Symp. VLSI Circuits*, 2010.

[9] J. Kim, "Low-cost router microarchitecture for on-chip networks," in *Intern. Symp. on Microarchitecture*, Dec. 2009.

[10] C. Nicopoulos et al., "Vichar: A dynamic virtual channel regulator for network-on-chip routers," in *IEEE/ACM Intern. Symp. on Microarchitecture*, 2006, pp. 333–346.

[11] A. T. Tran and B. M. Baas, "RoShaQ: High-performance on-chip router with shared queues," in *IEEE Intern. Conf. on Computer Design*, Oct. 2011, pp. 232–238.

[12] T. Moscibroda and O. Mutlu, "A case for bufferless routing in on-chip networks," in *Proc. of ISCA*, 2009, pp. 196–207.

[13] R. D. Mullins, A. F. West, and S. W. Moore, "Low-latency virtual-channel routers for on-chip networks," in *Proc. of the Intl. Symp. on Computer Architecture*, 2004, pp. 188–197.

[14] J. Kim et al., "A gracefully degrading and energy-efficient modular router architecture for on-chip networks," in *33rd Intern. Symp. Computer Architecture*, 2006, pp. 4 –15.

[15] H. Wang, L.-S. Peh, and M. S., "Power-driven design of router microarchitectures in on-chip networks," in *Proc. of the Intern. Symp. on Microarchitecture*, 2003, pp. 105 – 116.

[16] G. Dimitrakopoulos and K. Galanopoulos, "Switch allocator for bufferless network-on-chip routers," in *Proc. of the Intern. Work. on Intercon. Network Architecture: On-Chip, Multi-Chip*, 2011, pp. 19–22.

[17] A. K. Mishra, N. Vijaykrishnan, and C. R. Das, "A case for heterogeneous on-chip interconnects for cmps," in *Proc. of the intern. symp. on Computer architecture*, 2011, pp. 389–400.

[18] N. Agarwal, T. Krishna, L. Peh, and N. Jha, "GARNET: A detailed on-chip network model inside a full-system simulator," in *Proc. of the IEEE Intern. Symp. on Performance Analysis of Systems and Software*, April 2009, pp. 33–42.

[19] S. Ma, N. E. Jerger, and Z. Wang, "Whole packet forwarding: Efficient design of fully adaptive routing algorithms for networks-on-chip," in *Proc. of HPCA*, Feb. 2012, pp. 467–478.

[20] D. Wentzlaff et al., "On-Chip Interconnection Architecture of the Tile Processor," *IEEE Micro*, pp. 15–31, Sep./Oct. 2007.

[21] J. Balfour and W. J. Dally, "Design tradeoffs for tiled CMP on-chip networks," in *Proc, of the 20th ACM Intern. Conf. on Supercomputing (ICS)*, Jun. 2006.