

# BIXBAR: A Low Cost Solution to Support Dynamic Link Reconfiguration in Networks on Chip

Pablo Abad, Pablo Prieto, Valentin Puente, Jose-Angel Gregorio

University of Cantabria

Santander, Spain

{abadp, prietop, vpuente, monaster}@unican.es

**Abstract**— Improving link utilization is a key aspect in interconnection network design. Reconfigurable-direction inter-router links optimize network resource utilization, which substantially increases the maximum achievable throughput. In the case of On-chip Networks, the short distance between adjacent routers makes feasible fast link arbitration, which makes dynamic link reconfiguration an attractive solution. In this paper we propose a low-cost router micro-architecture that is able to deal with reconfigurable links with a marginal cost over a conventional router. The key element of the proposal is a bi-directional crossbar, which enables reconfiguration of links, without significantly increasing router area and energy. The results obtained indicate that with this proposal, system performance could be improved, for some selected workloads, by up to 25% while energy-performance tradeoff is reduced by 20%, avoiding the additional costs entailed in other state-of-the-art routers capable of performing dynamic link reconfiguration.

**Keywords**- *On-chip memory hierarchy; on-chip networks; bidirectional link; router microarchitecture.*

## I. INTRODUCTION

On-chip network infrastructures devoted to general purpose CMPs have a set of requirements which could be considered to be specific to the communication substrate of memory hierarchy. Sharing hardware resources with the rest of system components makes energy and area efficiency a first-order design issue, while coherence protocols determine the special characteristics of network messages [10][26][21]. Obtaining an optimal tradeoff among performance, area and energy is a difficult task, limiting the applicability of some performance-oriented design decisions due to their associated overhead. This fact has provided some homogeneity to the baseline router structure in many on-chip network proposals [3][6][9][18][23]. Among the different deterministic routing policies, Dimension-Order Routing [7] is the most common one. It enables fast operation and minimizes the buffering overhead (in the shape of additional Virtual channels [8]). However, this baseline router structure can lead to poor resource utilization, wasting more than half of the raw network bandwidth.

An efficient solution able to improve link utilization is the use of reconfigurable network channels [19]. This kind of link has demonstrated its performance advantages when compared to conventional fixed-link configurations [5][19], dynamically adapting router input and output bandwidth to the traffic requirements. This leads to much better resource utilization and therefore important performance benefits. However, an important limitation of bi-directional networks is the necessity

to duplicate crossbar ports in order to accommodate reconfigurable links [19].

In contrast to off-chip networks, in on-chip networks link bandwidth is profuse. In such latency-sensitive scenarios, link availability is usually employed to build wide channels [13], reducing the serialization penalty of communications. In order to evaluate the implications of this fact, we performed a simple experiment measuring how the behavior of a network with reconfigurable links similar to BINOC [19] evolves as inter-router links become wider. Through a cycle-accurate simulator, connected to the Orion tool [15], we measured the evolution of performance, area-delay efficiency and energy-delay efficiency [3] for an  $8 \times 8$  mesh network under different synthetic traffic patterns [7]. Results, shown in Figure 1., have been normalized to those obtained by a network with conventional links (TNOC) and a similar configuration. As can be seen, performance benefits remain nearly constant for any link width. However, in terms of area and energy efficiency, it can be seen that the results for reconfigurable networks quickly degrade, providing poorer performance even for widths less than those used in the on-chip environment. The source of this poor cost evolution is the crossbar. In a matrix-like crossbar, both power and area exhibit quadratic growth as link width is increased, while performance remains constant. The penalty of a crossbar with twice the number of ports becomes dominant, making reconfigurable link direction an unattractive approach.

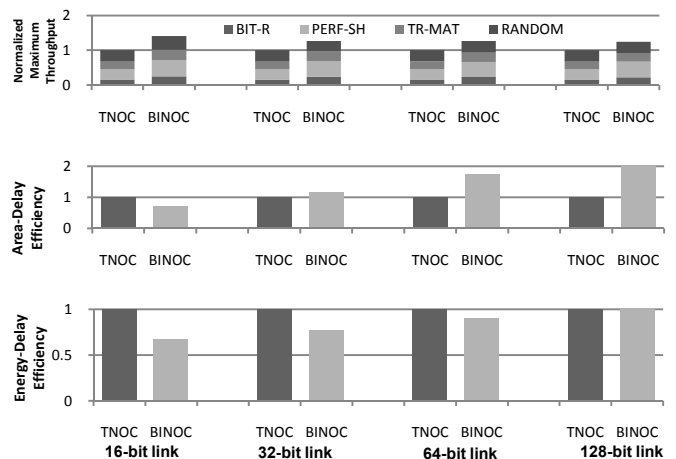


Figure 1. Performance evolution. (a) Throughput, (b) Area-delay efficiency (lower is better) and (c) Energy-delay efficiency (lower is better).

Nevertheless, a known effect of FIFO input-buffered routers is that head-of-line blocking limits the utilization of all

crossbar output ports [16]. If we combine this observation with the area and energy overhead numbers provided in Figure 1, we reach the conclusion that it may be interesting to employ the unused output ports to accommodate those links which modify their direction, instead of increasing crossbar input/output ports. In this way, we will maintain all the advantages derived from better link utilization, while minimizing the area and energy impact on router micro-architecture.

In this work, we propose a new crossbar organization in which the internal wiring will also re-configure the propagation direction. Thus, every crossbar port will be dynamically configurable as an input or output port, requiring the same size as a conventional crossbar to efficiently work with reconfigurable links. In order to quantify the efficiency of our proposal, we compare our router micro-architecture with two different counterparts: a typical CMP network router with conventional fixed links and the BINOC router proposed in [19], which makes use of reconfigurable links and a crossbar configuration with twice the number of ports. The rest of the paper is organized as follows: In Section II we introduce the new crossbar configuration proposed, and the resulting router architecture. In Section III the evaluation methodology is described and the proposal is evaluated. Finally, Section IV states the main conclusions of the paper.

## II. BI-DIRECTIONAL ROUTER

### A. Bi-directional Crossbar

Different organizations are possible for crossbar construction. An  $n \times n$  crossbar can be built through  $n(n-1)$  multiplexers. The same functionality can be provided through a regular matrix of wires. Multiplexer-based crossbars are the default designs generated from RTL synthesis [24]. However, the majority of place-and-route tools can lead to non-optimal designs [17]. When custom layouts are employed to design the router crossbar, matrix-like configuration seems to provide many advantages over multiplexers, better exploitation of crossbar regularity and energy reduction [17]. In fact, real designs such as Intel's 80-core NoC [11] make use of matrix crossbars implemented through custom layouts. Making use of matrix-like crossbars, router data-path duplication imposes a significant area and energy overhead. However, the potential performance benefits of reconfigurable links still make it worthwhile to search for solutions able to efficiently accommodate bidirectional wires. In this section we propose an alternative crossbar implementation that can operate with reconfigurable links while maintaining the wiring required by the crossbar used in a network with fixed links.

Figure 2. (a) shows a simple schematic of a  $2 \times 2$  matrix crossbar where an input driver (inverter chain) drives each horizontal wire. Additionally, each vertical wire employs crosspoint logic at both transmission switch and tri-state driver [17]. With this configuration, each crossbar line is driven independently. The proposed crossbar represented in Figure 2. (b), shares a similar matrix-like configuration. However, two major differences can be observed. First, through the inclusion of a hardware logic similar to the one used at cross-points, each switch port can act both as an input or output port, resulting in a combined input/output structure at each crossbar edge.

Second, each cross-point requires the inclusion of an additional tri-state buffer. Tri-state logic is not bi-directional, making it necessary to add this second connection to allow packets to travel in opposite directions through the same wires. In order to eliminate the overhead introduced by additional tri-state drivers, the wire grid could be interconnected through pass-gate cross-points [17].

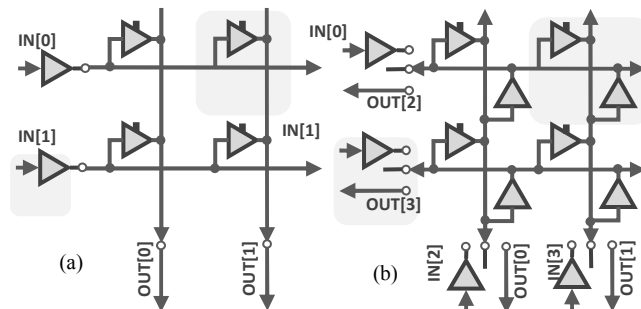


Figure 2. Simplified structure of a  $2 \times 2$  matrix crossbar with tri-state buffers as cross-points. (a) Conventional model and (b) bi-directional structure.

### B. Switch Arbitration

Control logic must be adapted to implement the functionality of the bi-directional switch structure. The arbitration process of our bi-directional crossbar requires a two-phase process. First, crossbar wires must be assigned according to the requests received at the arbitration unit. Second, crossbar ports must be configured to act as input or output ports, according to the decisions taken in the arbitration unit. Additionally, the arbitration unit must be aware of some input-output connections that might require the simultaneous utilization of three different data wires. For example, in Figure 2. (b), a message traversing the crossbar from  $in(0)$  to  $out(3)$  must drive two horizontal lines and a vertical one.

In a unidirectional crossbar, switch arbitration is performed on a per-output-port basis (for the sake of simplicity we will consider that a Virtual Channel Allocator filters multiple requests from a single input port). A granted request means that an input port obtains the exclusive utilization of one horizontal and one vertical wire. In the case of the bi-directional crossbar, requests ask for a variable number of wires, it being necessary to change the way arbitration is performed. In this case, we need to arbitrate each wire independently, doubling the number of arbiters needed. Now, a port requesting a switch traversal must send one request to each wire required, and only when all the wires required are granted will the port be granted. Additionally, as each router port can act either as an input or an output, each arbiter needs to double its number of inputs, to accommodate those cases where a message is present at every router port ( $in$  or  $out$ ).

A preliminary estimation of the overhead induced by this arbitration unit in terms of area, energy and delay can be made using Orion [15] power and area models and Peh and Dally's delay model [25]. For 45nm technology, Orion estimates a  $562 \mu\text{m}^2$  area for the control logic of a conventional crossbar. In the case of our proposal, this value grows up to  $2531 \mu\text{m}^2$ . A  $5 \times$  increase in area could be considered significant, but it must be considered in the context of total router area. Orion also

provides total router area, which in the case of the conventional crossbar returns a value of  $9.7 \cdot 10^5 \mu\text{m}^2$ . For this router value,  $2 \cdot 10^3 \mu\text{m}^2$  of additional area represents an overhead of less than 1%. Power requirements provided by Orion lead to a similar result. The power consumed by bi-directional crossbar arbitration more than doubles the values of a conventional crossbar, but in both cases this power is various orders of magnitude below the power consumed by a buffer or crossbar traversal.

A more critical aspect of crossbar arbitration could be the effect of arbiter delay. This could force the router to operate at lower frequencies or oblige the redesign of the pipeline in order to accommodate additional stages for arbitration. To obtain a first approximation of the delay overhead introduced by our new arbitration unit, we will make use of the formulation in [25]. For a typical clock cycle of  $20\tau_4$ , a switch allocator of a 5-port conventional router has a delay of  $11.8\tau_4$ . Employing the same formula for our arbiter, we obtain a new delay of  $14.6\tau_4$ . As can be seen, arbitration fulfils the cycle delay without problems, and in a typical design no additional stages will be required for bi-directional crossbar arbitration.

### C. Datapath Area and Energy Overhead

The overhead introduced by additional tri-state buffers at each cross-point derives from the area overhead associated with tri-state logic. However, no additional control signals are required, because the same control line can drive both tri-states at each cross-point through an inverter generating opposite values for each tri-state. Additionally, with deep-submicron technology, crossbar area is usually wire-dominated, as the area devoted to logic associated with each cross-point is much smaller than wire pitch [3][18]. Consequently, the area and energy overhead of extra tri-states is negligible.

Energy dissipation of a crossbar wire is mainly determined by line capacity  $C_{wire}$ . Considering a switching activity  $A$  and a link width  $W$ , crossbar traversal energy for a conventional crossbar can be formulated as

$$E_{xb-uni} = (C_{wire-in} + C_{wire-out}) \cdot A \cdot W + E_{tri-state}$$

In the case of our bi-directional crossbar, we maintain the same number of input and output ports, so the same wire length (and therefore capacity) can be assumed for our implementation. On the other hand, some communications need to drive three wires, which imposes some energy overhead. If we pessimistically assume a uniform distribution in input-output connections, half of data traversals will require three wires. Therefore, considering tri-state energy negligible and horizontal and vertical wires to be the same length:

$$E_{xb-bi} = 0.5(2C_{wire} \cdot A \cdot W) + 0.5(3C_{wire} \cdot A \cdot W)$$

$$E_{xb-bi} = C_{wire} \cdot A \cdot W + 1.5C_{wire} \cdot A \cdot W = 1.25 E_{xb-uni}$$

Assuming the use of low-power techniques such as segmented crossbar [27], we will be able to reduce this preliminary energy overhead. As can be seen in Figure 3, each

wire can be divided into two segments by tri-state buffers. Thus, only part of the wire must be driven in those cases where the cross-point is before the segment limit. In general, for an  $n \times n$  crossbar and a two-segment horizontal wire, we can compute the energy required to traverse it. Assuming a uniform utilization of crossbar wires,  $1/n$  of crossbar traversals only need to drive the first segment of horizontal wires. Now, uni-directional crossbar energy formulation can be re-written as

$$E_{xb-uni}^{Segm} = \frac{2n^2 - n + 1}{n^2} C_{wire} \cdot A \cdot W$$

In the case of the bi-directional crossbar, control logic will prioritize the utilization of the vertical wire closest to the input drivers. In this way, only in those cases when two or more three-wire transactions are performed simultaneously will it be necessary to drive the whole horizontal wire. Again, we will assume that three-wire traversals represent half of crossbar traversals. Moreover, we will assume that only 50% of 3-wire traversals can benefit from crossbar segmentation (a pessimistic fraction as we will see in the evaluation section). We can consider that this value is an energy upper bound because the fraction of cycles where two or more three-wire traversals coincide is very low. Anyway, under these assumptions, energy can be calculated as

$$E_{xb-bi}^{Segm} \leq [0.5(E_{xb-uni}) + 0.25(3C_{wire}) + 0.25(1.5C_{wire})] \cdot A \cdot W$$

Therefore, for a bi-dimensional network ( $5 \times 5$  crossbar),

$$E_{xb-bi}^{Segm} \leq (0.5 \cdot 1.84 + 0.25 \cdot 3 + 0.25 \cdot 1.5) C_{wire} \cdot A \cdot W = 1.11 E_{xb-uni}^{Segm}$$

With this worst-case analysis, crossbar segmentation allows us to reduce maximum energy overhead from 25% to 11% without any significant performance or area overhead [27].

The formulae from this section have been extracted assuming a fixed fraction for each type of crossbar traversal. However, through simulation we have determined that three-wire traversals represent only 20-30% of total traversals for the most common set of synthetic traffic patterns. Additionally, the three-wire traversals which have to drive three whole crossbar wires are a minimal fraction, below 5%, in most cases it being possible to take advantage of the segmented crossbar feature.

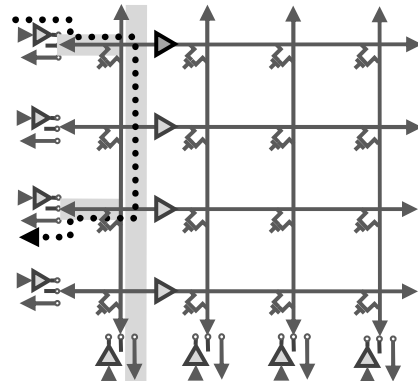


Figure 3. 4x4 bi-directional crossbar with two segments per wire. Detail of the wire-segments driven for a 3-wire packet traversal.

<sup>1</sup>  $\tau_4$  represents the delay of an inverter driving four other inverters **Error!**  
Reference source not found.

Finally, it should be noted that bi-directional crossbar cost parameters have only been compared to those obtained by a unidirectional crossbar with the same number of ports. Conventional crossbar solutions for reconfigurable links require twice the number of ports, consuming four times more area and 50% more energy according to Orion estimation (using a 128-bit link).

#### D. Router Architecture (BiXbar)

The router structure employed as the starting point is similar to other architectures proposed for on-chip environments [3][6][23][27]. Input buffers are employed at each port to hold incoming flits. These input buffers are dynamically distributed into multiple virtual channels through a buffering structure similar to [18]. Routing and flow-control policies are selected with the aim of minimizing cost and delay. Wormhole flow control enables the minimization of buffering requirements [7], while DOR routing is characterized by low complexity in the routing and arbitration processes. Latency optimizations such as those present in [18][23] can be considered to be unrelated to the utilization of reconfigurable links. We will employ a conventional 5-stage pipeline organization [25] for our router structures: 1) Route Computation (RC) computes message paths. 2) Virtual Channel Allocation (VA) grants a single virtual channel from each input port. 3) Switch Allocation (SA) is the last arbitration stage. 4) Switch Traversal (ST) advances flits from the input buffers to their granted output ports. Finally, 5) Link Traversal (LT) corresponds to the cycle required to traverse the wire.

Router ports have been modified to build dynamically reconfigurable channels. Link direction is governed through a channel-control module (Link Arbiter), which arbitrates the link. A detailed description of this module's behavior, as well as the logic required to build the FSMs controlling it, has been borrowed from [19]. In order to fit the logic associated with channel arbitration into a pipeline with the same number of stages as a conventional router, the required information (output port) to perform Link arbitration (LA) is generated after the RC stage. The logic associated with link direction can be executed in parallel with VA, and the results of both VA and LA forwarded to the SA stage simultaneously. Assuming 1-cycle delay links, this approach enables channel reconfiguration to be performed without incurring in any idle cycle penalty in the link.

### III. PERFORMANCE EVALUATION

#### A. Counterpart Routers

Two router configurations have been used in this section for comparison with our BIXBAR proposal. All the configuration parameters beyond the scope of this work have been equalized for the three configurations proposed, in an attempt to perform a fair comparison. Thus, topology, storage capacity, operation frequency, link width, etc. are fixed parameters for all the router structures. The first structure, denoted as traditional NoC (TNOC), represents the baseline configuration for our experiments. Two fixed links with opposite directions connect the router to each of its neighbors, employing a conventional 5x5 crossbar to implement this connectivity. As well as BIXBAR, this router also implements input-port buffering,

wormhole flow control and DOR routing policies, as well as operating under a fixed five-stage pipeline.

The second router configuration we employ for comparison, denoted BINOC, is a bi-directional structure mimicking the one presented in [19]. Routing, flow control and buffering management strategies are equivalent to those routers mentioned above. Its link arbitration process is the same as in our proposal, facilitating a fair comparison between the two routers. Each port in router BINOC requires two switch ports in order to accommodate dual behavior of network links. This keeps the same router radix but requires a crossbar with twice the number of ports.

#### B. Evaluation Framework

The evaluation process was performed through a framework composed of three different simulation tools. The full system simulator is based on SIMICS [20] augmented with GEMS [22]. The original network simulator of GEMS was replaced with TOPAZ [1], which accurately models the network architecture. The network simulator was coupled with the power modeling tool ORION [15]. The simulated system is a 16-processor CMP with static shared S-NUCA L2 [12]. The main parameters of the simulated system are shown in TABLE I. The system layout employs a 4x4 mesh to connect each processor and its private L1 with the 16 L2 banks. The link width selected for system configuration was 128 bits, which is a common value in this kind of environment [18].

TABLE I. MAIN SYSTEM CONFIGURATION PARAMETERS

<b>cores / issue width</b>	16@2GHz / 4	<b>Network Topology</b>	4x4 Mesh
<b>W.size/out. req.</b>	64/16	<b>Network link/delay</b>	128 bits / 1 cycle
<b>L1 I/D cache</b>	32KB, 2-way	<b>Message Size (flits)</b>	2 (command), 5 (data)
<b>L2 cache</b>	16MB, 16 banks	<b>Router Storage</b>	120 flits
<b>Main Memory</b>	4GB, 250 cycles	<b>Router delay</b>	4 cycles

Energy parameters for each router configuration were extracted through the power and area simulation tool Orion [15]. Results were obtained for a 2GHz operating frequency, 1V operating voltage and 45nm technology. Storage for every router was modeled as SRAM blocks. The switch model selected was a Matrix-Crossbar with Tri-state Gates as crossbar connectors. Finally, the length selected for inter-router links was 3mm, which seems to be reasonable for the size of system employed. TABLE II. summarizes the energy consumption of the main router events for the three counterparts.

TABLE II. ENERGY REQUIREMENTS

TNOC		BINOC		BIXBAR	
	E (pJ/flit)		E (pJ/flit)		E (pJ/flit)
<b>Buffer Write</b>	1.566	<b>Buffer Write</b>	1.026	<b>Buffer Write</b>	1.026
<b>Buffer Read</b>	7.727	<b>Buffer Read</b>	6.367	<b>Buffer Read</b>	6.367
<b>SW Traversal</b>	14.39	<b>SW Traversal</b>	24	<b>SW Traversal</b>	15.83
<b>Link</b>	50.9	<b>Link</b>	50.9	<b>Link</b>	50.9

#### C. Synthetic Traffic Patterns

Before full system simulation, we will provide performance results for communication patterns representative of a broad amount of applications (synthetic traffic patterns). In an attempt to mimic the characteristics of a coherence protocol, the packet size selected for these experiments has a bimodal distribution, 1 flit for 75% of the messages and 5 flits for the remaining 25%. Additionally, the reactive nature of coherence protocols is also emulated. Several synthetic patterns have been

considered. Figure 4. shows the evolution of total latency in steady state for three of the patterns (Random, Bit Reversal and Perfect Shuffle [7]). These results correspond to an 8×8 mesh topology. As can be seen, the advantages of bi-directional channel utilization are clear for non-uniform traffic distributions. While the performance advantage of BINOC is apparent for every traffic pattern analyzed, the advantages of BIXBAR clearly depend on the form of the traffic analyzed. In the case of non uniform traffic patterns, BIXBAR is able to obtain results closer to those obtained by the BINOC router but employing a quarter the crossbar size.

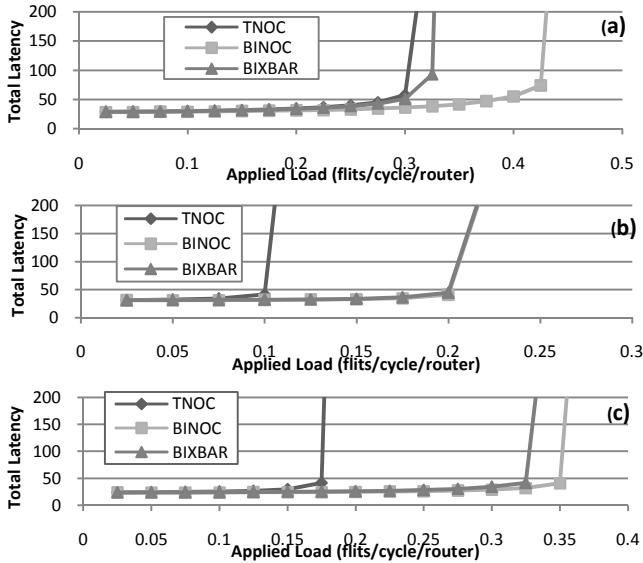


Figure 4. Latency results for the three routers evaluated under synthetic traffic patterns. (a) Random, (b) Bit reversal, (c) Perfect Shuffle.

In the second part of the synthetic traffic analysis, we evaluated each router configuration based on its area and energy efficiency [3]. To do so, we incorporated the area and energy estimations provided by ORION [15] into our network simulator. For this experiment, a burst of messages is injected at maximum rate, and the time required to consume all of them is determined. Area efficiency was then calculated as the product of the burst completion time and the router area. Similarly, energy efficiency was computed as the product of burst time and the energy consumed by the network. Figure 5. (a) shows the time required to consume the burst for each router configuration for 4×4 and 8×8 meshes. As can be seen, time results match the performance curves provided in Figure 4. , the fastest configuration being the one with the best latency curve. However, when taking into account area and energy, Figure 5. (b) shows that the performance advantage of BINOC is not able to compensate for the area and energy overhead introduced by a crossbar with twice the number of ports.

Additionally, as wire length increases, wire capacity is also increased, which has a negative impact on the energy consumed by a message in traversing crossbar wires. These metrics reflect the poor area and energy efficiency of the BINOC structure. On the other hand, the BIXBAR configuration uses a crossbar configuration with the same size as TNO, while maintaining a great part of the performance advantages provided by reconfigurable links. Thus, BIXBAR

proves to be the most efficient configuration in terms of both area-delay and energy-delay tradeoffs.

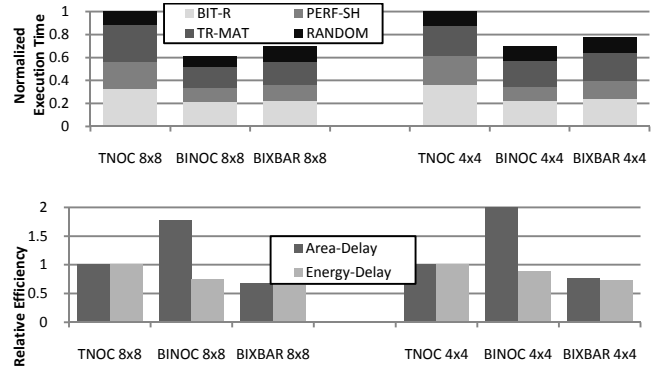


Figure 5. (above) Time required to consume a 100,000 message burst (TNO normalized). (below) Energy-delay and Area-delay product for each router evaluated (lower is better).

#### D. Full System Evaluation

The simulation infrastructure employed in this work allows us to measure the effects of our proposals under much more realistic conditions than synthetic traffic patterns. The multithreaded numerical applications are part of the NAS Parallel Benchmarks (OpenMP implementation version 3.2 [14]) and the PARSEC software [4]. Transactional applications correspond to the Wisconsin Commercial Workload suite [1]. Finally, desktop workloads belong to the SPEC CPU2006 suite [28] running in rate mode (multi-programmed). All the results provided have a 95% confidence interval.

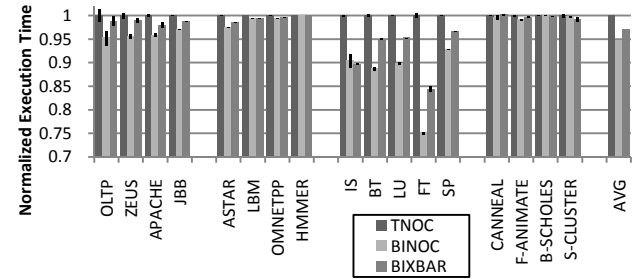


Figure 6. Full-system evaluation results. TNO-normalized execution time.

The results provided in Figure 6. represent execution time values normalized to those obtained by the conventional TNO. As can be seen, the performance differences of the three structures are attenuated when compared to synthetic traffic results. Applications have different levels of communication requirements, and even those workloads with intensive utilization of network resources also have phases with light load over the network. According to performance results, applications can be categorized in three different groups. First, some applications use network resources lightly throughout the execution (*omnetpp*, *hmm*, *PARSEC*). In this case, no performance benefits can be extracted from reconfigurable links. A second group corresponds to those applications with intensive communications that use the network resources uniformly (*oltp*, *zeus*, *jbb*). As seen in the synthetic traffic pattern analysis, BIXBAR is not able to achieve large performance benefits when traffic is uniformly distributed. In

this case, only BINOC is able to provide performance benefits. Finally, we also find applications with considerable bandwidth utilization and mixed phases with uniform and non-uniform communications (*astar*, *apache*, *NAS*). In these cases, the advantages of both bi-directional routers are clearly observed.

Finally, the evaluation of energy efficiency in Figure 7. puts the tradeoff between network cost and system performance into perspective. Although it is the configuration with the best performance, the hardware overhead provokes a strong penalty in BINOC when including energy results. Only with the FT and LU applications, where performance differences are large, is there a better energy-delay value in the case of BINOC. Our proposal provides a lower performance improvement, but this is obtained with a much smaller energy overhead, as this is the structure with the most efficient design. As can be seen, BIXBAR improves Energy-Delay Product (EDP) by 5% on average when compared to a conventional router implementation with similar characteristics, and by 15% when compared to the BINOC structure.

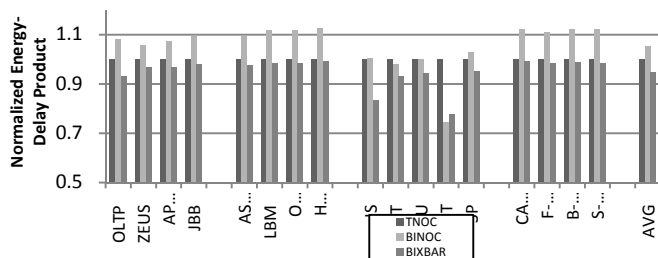


Figure 7. TNOC-normalized Energy-delay efficiency (EDP).

#### IV. CONCLUSIONS

We present a router with a bi-directional crossbar structure whose crossbar wires are dynamically re-configured. Through this organization, every crossbar port can behave as either an input or an output port. In this way, we are able to build a router architecture which does not need to duplicate crossbar ports in order to accommodate reconfigurable channels. Results show that our proposal can obtain similar performance results to BINOC (a true duplicate crossbar) under non-uniform traffic patterns, and the best energy-performance tradeoff of the three counterparts compared.

#### ACKNOWLEDGEMENTS

The authors would like to thank Jose-Angel Herrero for his valuable assistance with computing environment, and the anonymous reviewers for many useful suggestions. This work has been supported by the Spanish Ministry of Science and Innovation, under contract TIN2010-18159, and by the HiPEAC European Network of Excellence.

#### REFERENCES

- [1] P. Abad, et. al. "TOPAZ: An Open-Source Interconnection Network Simulator for Chip Multiprocessors and Supercomputers", NOCS 2012
- [2] A.R. Alameldeen, et. al., "Simulating a \$2M Commercial Server on a \$2K PC", IEEE Computer, February 2003.
- [3] J. Balfour, W. J. Dally, "Design Tradeoffs for Tiled CMP On-chip Networks", International Conference on Supercomputing (ICS), 2006.

- [4] C. Bienia, "Benchmarking Modern Multiprocessors", Ph.D. Thesis. Princeton University, January 2011.
- [5] M.H. Cho, et. al., "Oblivious Routing in On-Chip Bandwidth-Adaptive Networks", International Conference on Parallel Architectures and Compilation Techniques (PACT), September 2009.
- [6] W.J. Dally, B. Towles, "Route Packets, Not Wires: On-Chip Interconnection Networks", Design Automation Conference, June 2001.
- [7] W.J. Dally, B. Towles, "Principles and Practices of Interconnection Networks", Morgan Kaufmann Publishers, Inc., 2004.
- [8] W.J. Dally, "Virtual-channel Flow Control", IEEE Transactions on Parallel and Distributed Systems, vol. 3, no. 2, March 1992.
- [9] P. Gratz, C. Kim, R. McDonald, S.W. Keckler, D. Burger, "Implementation and Evaluation of On-Chip Network Architectures", International Conference on Computer Design (ICCD), October 2006.
- [10] A. Hansson, K. Goossens, A. Radulescu, "Avoiding Message-Dependent Deadlock in Network-Based Systems on Chip", VLSI design, 2007.
- [11] Y. Hoskote, et. al., "A 5-GHz Mesh Interconnect for a Teraflops Processor", IEEE Micro, vol. 27, issue 5, November 2007.
- [12] J. Huh, C. Kim, H. Shafi, L. Zhang, D. Burger, S.W. Keckler, "A NUCA Substrate for Flexible CMP Cache Sharing", International Conference on Supercomputing (ICS), June 2005.
- [13] N.E. Jerger, L.S. Peh, "On-Chip Networks, Synthesis Lectures on Computer Architecture", Morgan & Claypool Publishers, 2009.
- [14] H. Jin, M. Frumkin, J. Yan, "The OpenMP Implementation of NAS Parallel Benchmarks and its Performance", NAS Tech. Report, 1999.
- [15] A. Kahng, B. Li, L.S. Peh, K. Samadi, "ORION 2.0: A Fast and Accurate NoC Power and Area Model for Early-Stage Design Space Exploration", Design Automation and Test in Europe (DATE), 2009.
- [16] M. Karol, M. Hluchyj, S. Mogan, "Input versus output queuing on a space-division packet switch," IEEE Trans. Commun., Dec. 1987.
- [17] T. Krishna, L.S. Peh, B.M. Beckmann, S.K. Reinhardt, "Towards the Ideal On-Chip Fabric for 1-to-Many and Many-to-1 Communications" International Symposium on Microarchitecture (MICRO), 2011.
- [18] A. Kumar, et. al., "A 4.6 Tbits/s 3.6GHz Single-cycle NoC Router with a Novel Switch Allocator in 65nm CMOS" International Conference on Computer Design (ICCD), October 2007.
- [19] Y. Lan, S. Lo, Y. Lin, Y. Hu, S. Chen, "BINOC: A Bi-directional NoC Architecture with Dynamic Self-Reconfigurable Channel", International Symposium on Networks-on-Chip (NOCS), May 2009.
- [20] P.S. Magnusson, M. Christensson, J. Eskilson, D. Forsgen, G. Hallberg, J. Hogberg, F. Larsson, A. Moestedt, B. Werner, "Simics: A Full System Simulation Platform", IEEE Computer, Vol. 35, no. 2, February 2002.
- [21] M.M.K. Martin, M.D. Hill, D.A. Wood, "Token Coherence: Decoupling Performance and Correctness", International Symposium on Computer Architecture (ISCA), June 2003.
- [22] M.M.K. Martin, et. al., "Multifacet's General Execution-driven Multiprocessor Simulator (GEMS) Toolset", Computer Architecture News (CAN), September 2005.
- [23] R. Mullins, A. West, S. Moore, "Low-Latency Virtual-Channel Routers for On-Chip Networks", International Symposium on Computer Architecture, June 2004.
- [24] C.H. Owen Chen, S. Park, T. Krishna, L.S. Peh, "A Low-Swing Crossbar and Link Generator for Low-Power Networks-on-Chip", International Conference on Computer-Aided Design, November 2011.
- [25] L.S. Peh, W.J. Dally, "A Delay Model and Speculative Architecture for Pipelined Routers", International Symposium on High Performance Computer Architecture (HPCA), January 2001.
- [26] Y.H. Song, T.M. Pinkston, "A Progressive Approach to Handling Message-Dependent Deadlock in Parallel Computer Systems", IEEE Transactions on Parallel and Distributed Systems, Vol. 14, no. 3, 2003.
- [27] H. Wang, L.S. Peh, S. Malik, "Power-driven Design of Router Microarchitectures in On-chip Networks", International Symposium on Microarchitecture (MICRO), December 2003.
- [28] The Standard Performance Evaluation Corporation. SpecCPU2006, <http://www.spec.org/cpu2006>