Balancing Performance and Cost in CMP Interconnection Networks

Pablo Abad, Valentin Puente, Jose-Angel Gregorio

Abstract—This paper presents an innovative router design, called Rotary Router, which successfully addresses CMP cost/performance constraints. The router structure is based on two independent rings, which force packets to circulate either clockwise or counterclockwise, traveling through every port of the router. These two rings constitute a completely decentralized arbitration scheme that enables a simple but efficient way to connect every input port to every output port. The proposed router is able to avoid network deadlock, live-lock and starvation without requiring data-path modifications. The organization of the router permits the inclusion of throughput enhancement techniques without significantly penalizing the implementation cost. In particular, the router performs adaptive routing, eliminates HOL-blocking and carries out implicit congestion control using simple arbitration and buffering strategies. Additionally, the proposal is capable of avoiding end-to-end deadlock at coherence protocol level with no physical or virtual resource replication while guaranteeing in-order packet delivery. This facilitates router management and improves storage utilization. Using a comprehensive evaluation framework that includes full-system simulation and hardware description, the proposal is compared with two representative router counterparts. The results obtained demonstrate the Rotary Router's substantial performance and efficiency advantages.

Index Terms— Rotary Router, router architecture, interconnection networks, Chip Multi-processors, coherence protocol, routing deadlock, coherence protocol deadlock.

1. Introduction

CMPs are a leading alternative for dealing with increasing design complexity in current and future microarchitectures. In such systems, memory hierarchy plays an important role in achieving the expected performance. A critical component of the on-chip memory hierarchy is the interconnection network. Like the bonding substrate of other components in the chip, the interconnection network must scale with future transistor budget increments. Nowadays, centralized structures are common in commercial CMPs [19][35], but an increasingly large number of functional blocks in the system will eventually require decentralized structures. Packet-switched point-to-point networks have been postulated as the best candidate to accomplish this challenge [9]. Some academic proposals [7] or commercial solutions [5] are already using these networks.

Although point-to-point networks are a well-known topic in multiprocessor systems, it is mandatory to bear in mind the on-chip technological constraints before applying the knowledge to CMP systems. Correctness guarantee mechanisms, such as detection or avoidance of anomalies at network level, are similar in off-chip and on-chip networks. However, in the CMP domain the network operates in a more imbricate way with the rest of the system than in the off-chip domain. Due to the implementation nature a significant reduction in latency and an important increment in raw bandwidth are available. Nevertheless, the implementation cost and power consumption limits are bounded by other major components on the chip [4]. The growing number of processor cores and restrictions in heat dissipation devices limit the feasibility of using complex interconnection networks [33]. An excessively complex implementation of the network will decrease the total budget of transistors devoted to other crucial components of the system, interfering with its overall performance. On the contrary, overly simple network approaches could hinder system performance. For example, it makes no sense to reduce the buffering capacity on the routers to a point where the network contention is severely penalized.

The CMP prevalence in commodity systems makes it necessary to parallelize general purpose applications, which is challenging. A shared-memory model seems to be the most suitable for parallel programming because it is easier to reason about a global address space than about a partitioned one. In order to support the shared-memory with large on-chip multi-level cache hierarchies CMP systems usually require the presence of hardware-based cache coherency. The coherence protocol, applied by the coherency controllers, is responsible for guaranteeing the coherence invariants in each cache block. This makes the cache hierarchy tangle transparent to the software.

To ensure coherence invariants, each coherency controller could react to the incoming memory or coherency transaction, sending new messages to other controllers in the system. In this way, correlation among different classes of protocol messages could coexist in the network, with dependence among them. The reactive nature of these different types of messages and the limited capacity of consumption queues can generate end-to-end deadlock [41]. When a particular class of messages over-

- *F.A. Authors are with the University of Cantabria, Santander, Spain. E-mail: {pablo.abad, vpuente, joseangel.gregorio}@unican.es.*

Manuscript received –xxxx-2010.

flows a consumption queue, the progression of subsequent actions at the coherency controller cannot be assured. Under these circumstances, if the network cannot guarantee the delivery of some messages because overflowed traffic is blocking the responses, pending actions at the coherency controller cannot advance. When this cyclic dependency between messages in the network and actions in the coherency controller appears the system will stall. This problem is known as end-to-end deadlock or message-dependent deadlock. The most common solution to deal with the problem is to devote a partition of network resources to each class of messages through the utilization of separate networks, both virtually [30] or physically [23].

On the other hand, some protocols require that consecutive messages between two given controllers arrive at their destination in the same order as they entered into the network. Therefore, the network must be capable of guaranteeing the in-order delivery of the traffic that requires it. The simplest solution is to use deterministic routing with this kind of traffic.

The Rotary Router can cope with the aforementioned performance, correctness and cost requirements. The router structure, as can be seen in Fig. 1, is based on two counter-rotating rings, which force packets to travel sequentially through the ring until finding a profitable output port available. This crossbar-less scheme does not require the utilization of virtual channels to guarantee correctness or improve performance. The proposal can: (1) reduce Head-of-Line blocking, (2) use adaptive routing, (3) be topologically agnostic, (4) scale with network degree, and (5) have reasonable power consumption and implementation cost in the CMP context.

Routing and flow control algorithms work synergically to avoid the appearance of potential anomalies in the interconnection network. Based on occupation control, three different flow control mechanisms ensure progression of all traffic under any circumstances. The Rotary Router guarantees that the resulting interconnection network is not only routing deadlock free, but it can also circumvent end-to-end deadlock without splitting, physically or virtually, the resources used by different classes of messages in the dependence chain of a memory transactions. The idea is based on restricting the amount of resources per router available for each kind of traffic. In contrast to conventional solutions, our idea maximizes buffer utilization, not requiring allocation of exclusive storage resources for each traffic class. Additionally, the management of each class favors most priority traffic in a natural way. At first glance, the distributed nature of the router seems to preclude the in-order delivery capability. However, we have found that for a small proportion of messages it is possible to achieve this capability with little effort and with low performance impact, simply by restricting network resource availability combined with a port-to-port book-keeping mechanism.

To demonstrate the claims and quantify the performance enhancement of the Rotary Router over conventional routers a comprehensive performance analysis has been carried out which ranges from place-and-route implementation cost estimation to full-system simulation. The results show that the advantage of our proposal is up to 50 % in terms of raw performance and nearly 60 % in terms of energy-delay product. Finally, an estimation of the feasibility of the implementation is provided.

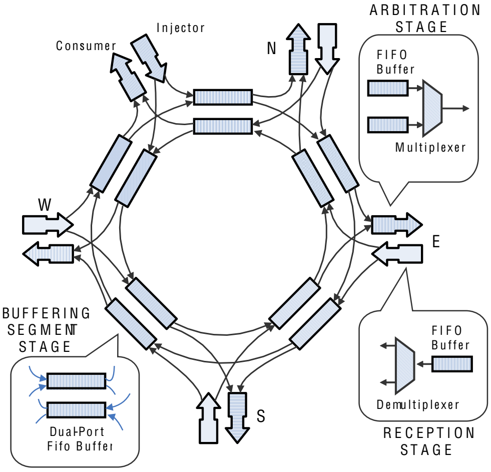


Fig. 1. Rotary Router sketch.

The seminal work describing the basis of this proposal was introduced in [1][2].

The rest of the paper is organized as follows: Section 2 introduces the Rotary Router architecture and operation. Section 3 describes how network anomalies are avoided. Section 4 introduces the adjustments required to guarantee coherence protocol correctness. Section 5 presents the evaluation framework. Section 6 shows comparative performance and results and Section 7 states the main conclusions of the paper. Finally and for space reasons Appendices explain in a deeper way important details for a complete understanding of the proposal.

2. Router Architecture & Operation

Next we will describe the router architecture and its operation. On the one hand, in order to minimize contention effects on performance, the Rotary Router should not make use of centralized arbitration mechanisms. For this reason, arbitration will be performed at each router output port and with fixed cost independently of the number of ports. On the other hand, non-FIFO buffers involve a high cost, so it is necessary to deal with the head-of-line (HOL) blocking problem while maintaining buffer FIFO policy. Another alternative is a mechanism to enable the packets at the head of the FIFO queue to be overtaken when profitable output port access has been denied, making it possible to advance the packets waiting behind. Finally, it would be desirable that neither the number of router ports nor the routing algorithm increases switch fabric complexity, thus achieving the best possible utilization of the expensive buffering resources. In order to address all the aforementioned requirements, the way of connecting the components inside the router has to be redefined, while some common elements present in conventional architectures should disappear.

Fig. 1 shows a diagram of the router for a bi-dimensional topology. The structure of the Rotary Router is based on two independent rings, which force packets to circulate either clockwise or counterclockwise, traveling from port to port of the router. Each ring is built with a group of Dual-port FIFO Buffers (DFB). The operation of the Rotary Router is simple, when a packet arrives at a router input port it is sent to one of the rings which forms the router. The packet starts moving towards its output port using the DFBs of the ring. Once the packet reaches a profitable output port, there are two possibilities. If the output port is available, the packet will advance to the next router. Otherwise, the packet will keep on circulating in the ring until reaching another profitable output port[1] in the current router. When necessary, the packet will be forced to complete a full circuit of the ring and start a second one. The packet must perform as many full circuits as needed before leaving the router (later, we will see that this happens very rarely in practice).

The circulation of packets inside the router has numerous advantages. Each packet is able to go through any output port avoiding the usage of centralized arbitration, and consequently reducing contention. As no centralized crossbar is required, the router data-path implementation could be simplified. Allowing packets to leave the router by any profitable output port constitutes an implicit application of adaptive routing. The reduction of the HOL blocking is inherent; when a packet at the head of a DFB is not able to obtain access by an output port, it moves toward the next DFB in the ring, allowing the advance of the rest of the packets behind it. Most of the buffering capacity of the network is available to any packet, regardless of its class or destination.

Structurally, the Rotary Router is made up of three independent classes of components or stages, denoted in Fig. 1 by Reception, Arbitration and Buffering Segment. The first and second constitute the input and output ports of the router and the third forms the rings. The structure and complexity of each block are both independent of the node degree, making the router easily scalable. Next, the basic function of each block is described.

Reception Stage

This block is constituted by a de-multiplexer and a FIFO buffer capable of storing, at least, the largest possible packet in the network. For each packet, depending on network topology, current node and destination node, this stage is responsible for determining the profitable output ports. This routing information is appended to the packet header. The reception stage also selects the ring direction, trying to reduce the packet intra-router delay. The decision depends mainly on two factors, the number of DFBs traversed and the occupation level of the two accessible DFBs. At low load, ring selection minimizes the number of DFBs needed to reach the nearest profitable output. At medium-high load, the time spent traversing a DFB becomes the dominating factor, the selection is dominated by the occupation level of each ring. In order to select the ring direction, the reception stage only checks the occupation level of the DFBs connected to it, because the occupation of all the buffers in the same ring is similar.

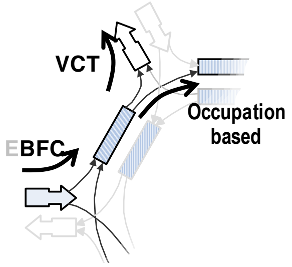


Fig. 2. Flow control mechanisms in the Rotary Router.

Arbitration Stage

This stage must manage the ejection of packets from the rings. It is the only point where two packets could collide at the access to a common resource (the output port). Two buffers and one multiplexer are provided in order to share the single physical channel. The multiplexer employs a random policy to guarantee fair usage of both rings. Besides detaching the arbitration process from the inner-ring packet movement, the output buffers are essential to maximize network performance. These output buffers store packets capable of using the link as soon as it becomes available, improving the link's utilization. Combined with the reception module, this stage is responsible for applying flow control mechanisms between contiguous routers.

Buffering Segment Stage

This module is made up of two DFBs connecting every two router ports. Each DFB has two pairs of Read/ Write (R/ W) ports. One pair is used to build a ring (connecting with the previous and next DFBs) in which packets circulate, while the other one connects the buffer to the Reception and Arbitration Stages. Two independent rings are found on each router, each made up of a number of DFBs equal to the number of input ports. The two rings have opposite directions in order to minimize the number of DFB traversals. This stage must decode the routing information included in every packet header and generated by the reception stage. If a profitable output port of the router is available, the Buffering Segment Stage must use the read port connected to the corresponding Arbitration Stage. Otherwise, the read port connected to the write port of the next DFB should be used. This module can be considered as the essential component of the router, making all the intended goals achievable.

3. NETWORK CORRECTNESS

Next, we will describe how the routing and flow control algorithms work synergistically to avoid the appearance of potential anomalies in the interconnection networks. We will demonstrate that the network is free of deadlock, livelock or starvation. The theoretical foundations are supported by S. Konstantinidou and L. Snyder's work [20].

[1] Which brings the packet toward its destination

3.1. Flow Control and Routing Algorithm

In the Rotary Router, flow control and routing mechanisms are strongly bounded to the deadlock avoidance method. As shown in Fig. 2, three flow control mechanisms coexist in the network; one of them controls the advance between routers, another one manages the inner-rings' packet movement and the last one controls the access to the rings. These three mechanisms are responsible for ensuring progression of all traffic under any circumstances.

To control the packet movement between DFBs, occupation-based flow control is used. Each DFB makes its own occupation level available to the previous DFB in the ring. In an on-chip environment this is not a problem, due to the router blocks' proximity [10]. In this way, we can allow a packet to advance to the next DFB only if the occupation of the destination buffer is less or equal to the occupation of the current one. This flow control helps to balance the occupation of all the DFBs in the ring and to equalize the ring injection probability at each input port. In order to control the advance of packets among routers Virtual Cut Through (VCT) [15] is employed. To apply it, the reception stage has capacity for at least one of the biggest packets of the system, typically a cache block plus some control information. Finally, the packet injection to the rings is basically regulated by Bubble Flow Control (BFC) [37]. In order to allow the access of a packet to any ring, the buffer requested must have room for at least two packets. If the input port is connected to a processing element, the number of required "holes" increases to three. This flow control guarantees the packet movement inside any ring.

However, in some situations the in-transit traffic generated by the system may need to fine tune admission to the ring. Under realistic working conditions, the traffic pattern is composed of time fluctuating flows between the same source-destination pair. Intense fixed packet flows can cause unfair resource utilization by high activity input ports, it being more difficult for low activity input ports to access the rings. Due to this, unbalanced link utilization may arise. Fortunately, it is possible to avoid this situation by balancing buffer occupancy among every router input port. The least used router input ports will be given ring injection preference over the most used. This mechanism works as follows; when the number of packets in a router from the same transit input port surpasses a certain limit, the flow control applied on the port is restricted, increasing the number of packet holes required to inject a packet into the ring. This gives the rest of the input ports more chance to inject. Once the number of packets from the restricted port falls below the limit, the restricted flow control returns to its original value. With this modification in the flow control, it is possible to improve the mixture of different traffic flows in the router, achieving a better output link utilization even under adverse traffic configurations. The resulting flow control is denominated Equalized Bubble Flow Control or *EBFC* (see Appendix 1).

With respect to the routing algorithm, the packet always tries to leave each router through, according to routing information, a minimal path to the destination. When a profitable port is not available at the exit of the current DFB, it must advance to the next DBF. If a packet makes a predetermined (and large enough) number of unsuccessful complete circuits of the rings, it will be tagged for misrouting. From that moment, this packet must use the first available output transit port. Once the packet leaves the router, the tag is cleared and the packet will advance following minimal path again.

Under these working conditions, as shown in Appendix 1, intra and inter-router packet movement is ensured, guaranteeing the absence of deadlock, livelock and starvation anomalies.

4. Coherence Protocol Requirements

In addition to adequately covering the requirements of any network on chip, the network that is part of a CMP system must also fulfill the coherence protocol requirements, normally present in any CMP. Two of the most important of these requirements are the avoidance of end-to-end deadlock and the in-order-delivery of certain types of protocol messages.

4.1. End-to-end Deadlock Avoidance for Rotary Router

In a CMP cache coherence protocol, most messages involved in a memory transaction depend on each other, since the injection of some message types is a consequence of the delivery of other types. For example, when a coherency controller sends a message requesting a cache line, this will generate a message from another coherency controller providing that block. This relationship, known as the message dependency chain [41], could potentially cause deadlock at the controller level. This new kind of deadlock, known as message-dependent deadlock or end-to-end deadlock, appears at the endpoints of the interconnection network because of the limited capacity of the consumption queues. In contrast to network-dependant deadlock, message-dependant deadlock has no relation with routing and or topology.

As a simple example, consider an interconnection network with a request-reply protocol. If at some point network resources are congested by request messages due to consumption queue overflow, reply messages might not be able to make progress. If replies cannot finalize, the processing of pending requests is stalled due to the exhaustion of resources allocated in upper levels of the system to pending replies. The cyclic dependency between reply and request could stall the whole system. Realistic coherence protocols have much longer dependency chains between messages[2], because depending on the state in which a cache line is in, different operations will be performed in order to allow the requesting processor to access the data. For example, the communica-

[2] *As we are assuming a cache coherent system, each message is composed of just one packet (a command or a command plus a cache block) and consequently we will indistinctly employ the terms "message" or "packet"*

tion protocol in the Alpha 21364 [30] has a dependency chain length of seven, which means that some operations need seven messages to complete. To deal with this situation, the utilization of traffic in different virtual networks is commonly used. For such long dependency chains, the hardware overhead derived from virtual-network utilization becomes significant, encouraging the search of more efficient mechanisms. Software solutions based on avoiding consumption queue overflow by dumping pending messages into host main memory are not suitable for CMP systems.

The solution for the Rotary Router is based on its facility to avoid HOL blocking. A blocked message cannot indefinitely delay the access of other messages to an available output port and this capability, adequately exploited, will allow us to avoid message-dependent deadlock with almost zero cost. This method is deeply explained in Appendix 2.

4.2. In-Order Delivery

Another typical requirement for some coherence protocols [27] (or specific maintenance tasks [30]) is in-order delivery support. In-order delivery is the property by which any set of packets with the same pair source-destination should arrive at the destination node in the same order as the injection sequence [10]. Fulfilling this requirement is straightforward for an input buffered router. A deterministic routing algorithm can be used in ordered messages, forcing them to follow a fixed path to the destination. The proportion of traffic that requires this is small enough to use this approach with nearly no impact on performance. Usually the data-path prevents packet reordering inside this kind of routers and so in-order delivery is guaranteed. Paradoxically, the majority of the Rotary Router's advantages derive from its on-router reordering. To guarantee in-order packet delivery in the Rotary Router, we will use the fact that it is possible to guarantee deadlock freedom for in-order traffic if a sub-routing function with this property is applied. For example, if we are employing a suitable topology for CMP, such as a bi-dimensional mesh, it is enough to apply a dimension order routing (DOR) to this traffic to ensure deadlock freedom. In general, for any *k*-ary *n*-cube network, the combination of not allowing the use of wraparound links and DOR will be enough to guarantee deadlock freedom. Although sub-optimal, it is also generally possible to maintain the in-order delivery implementation topology agnostic, employing some topology agnostic deadlock-free routing, such as up/down, for in-order traffic.

In the Rotary Router packets traveling in opposite directions of the same dimension can share the same buffering space. This could lead to deadlock when in-order traffic is present. For example, if we reach a configuration where two neighboring routers devote all the buffering capacity to in-order traffic, none of the traffic involved could progress. To circumvent this situation, different buffer rings will be used for in-order traffic flowing in opposite directions, as can be seen in a 2-ary 3-cube in Fig. 3. Packets traveling in opposite directions will make use of different rings. Note that the previously depicted intra-router and inter-router flow controls remain unaltered. Under the above conditions, it can be ensured that packets with the same source-destination pair will advance along the same path. However, we still need to ensure that packets do not change their order while moving inside the router rings. A mechanism based on lookup tables is developed for this purpose. The operation of this mechanism is explained in Appendix 3.

5. Evaluation Methodology

5.1. Counterpart Routers

In order to understand better the benefits of the Rotary router, two counterpart routers with different cost ranges have been selected (see Appendix 4 for details). Both routers use worm-hole flow control [10], implementing deterministic routing. The first router, based on the model in [34] and denoted WH-BASE, represents a minimal cost implementation. It will include minimal buffering capacity statically distributed among each virtual channel (2flits/VC) and a classical pipeline of 5 stages. This router represents the minimal cost design point and will provide a reference to determine the cost-performance ratio of the Rotary Router.

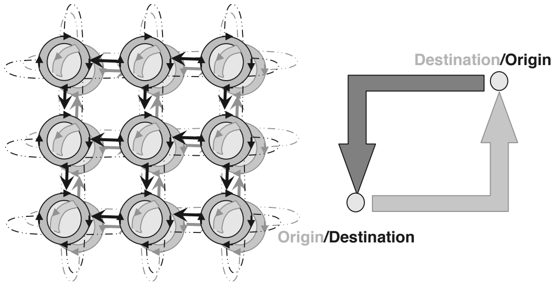


Fig. 3. Ring buffer and link utilization for in-order traffic for a *2-ary 3-cube*.

The second counterpart denoted WH-ADV, will use the same flow control and routing mechanisms as WH-BASE, but with more generous buffering and hardware and pipeline optimizations. WH-ADV implements speculative switch arbitration, reducing router pipeline to only three stages. It also employs a shared buffer per port similar to [21], dynamically adapting VC capacity. Although the design is quite similar to [21], our implementation does not have low-load pass-through technique to keep the comparison fair. This class of techniques is orthogonal to the router comparison and similar solutions can be also used with the Rotary Router. WH-ADV represents a different design point with a similar implementation cost to the Rotary Router and it will allow us to compare raw performance face-to-face.

Like in [30], to support different numbers of message classes without end-to-end deadlock we use separate virtual networks. We will assume a coherence protocol with six classes of messages, being six virtual channels

TABLE 1 MAIN SIMULATION PARAMETERS

Number of Cores	16@2GHz
Window Size / outstanding req. per CPU	64 / 16
Issue Width	4
L1 I/D cache	Private, 32KB, 4-way, 64-Byte block, 1-cycle
Direct Branch Predictor	4KB YAGS
Indirect Branch Pred.	256 entries (cascaded)
L2 cache	16MB SNUCA, token coherence 6 message length chain protocol, 16 banks, 1 banks/router,
L2 cache bank	1MB, 16-way, 5-cycle, pseudo LRU, 64-Byte block
Main Memory	4GB, 260 cycles, 320 GB/s
Command size	16 bytes
Network Topology	4-ary 2-cube (16 banks + 16 cores)
Network Link (width/latency)	128 bits / 1 cycle
Packet Length	5 flits or 2 flits

the minimal number required. However, in the case of a torus network the number of virtual channels must grow to twelve, because the routers use Dally's deadlock avoidance mechanism which requires two virtual channels to avoid network-dependent deadlock.

Nevertheless, in a mesh network where only six virtual channels are needed to avoid end-to-end deadlock, the number of virtual channels could be doubled trying to improve performance. Our performance evaluation will also explore this possibility.

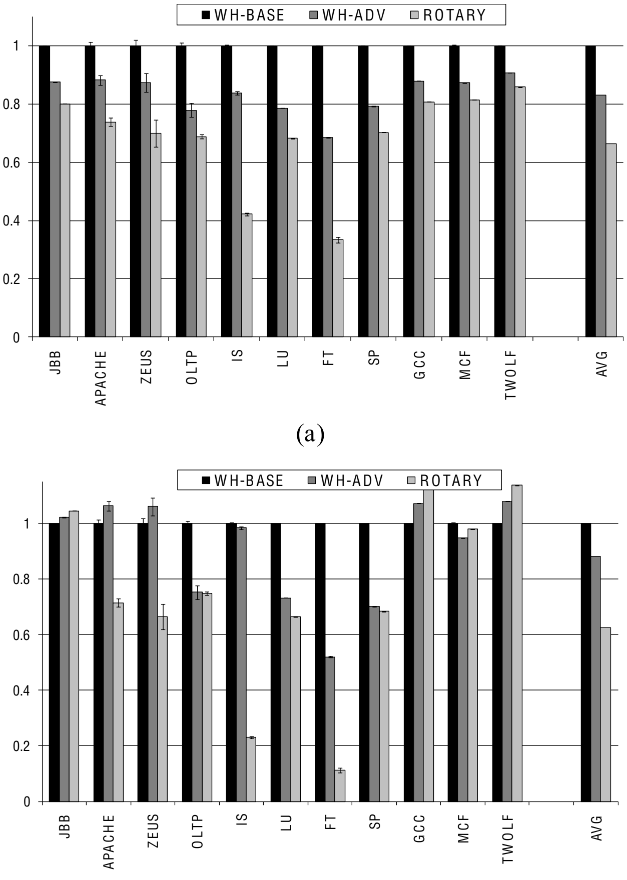
5.2. Simulation Framework

We are employing a working framework composed of three simulators. The interconnection network simulator SICOSYS [38] is used to enable precise modeling of the network behavior. This simulator will be embedded in a full system simulator, in order to demonstrate the advantage of the proposal under realistic working conditions. The full system simulator is based on SIMICS [25] augmented with GEMS [26]. GEMS provides detailed models of both the memory system and a state-of-the-art processor. SICOSYS has been integrated into the GEMS simulator, replacing its original network simulator.

The complete framework will allow us to perform exhaustive full-system simulation with complex workloads with detailed modeling of the most relevant system modules at architectural level.

The simulated system is a 16-processor CMP with shared S-NUCA L2 based on [16]. Token Coherence [27] is used, requiring a hierarchy of six classes of messages to be implemented. In this protocol, persistent request activation and deactivation must be point-to-point ordered. With this configuration, we will demonstrate the advantages of our proposal in terms of performance and correctness, with a large number of message classes. The main parameters of the simulated system are shown in Table 1.

The workloads considered in this study are three multi-programmed and eight multithreaded workloads running on top of the Solaris 9 OS. The numerical applications are part of NAS Parallel Benchmarks (OpenMP implementation version 3.2.1 [11]). The transactional benchmarks correspond to the Wisconsin Commercial Workload suite [3], released by the authors of GEMS in the 2.1 version. The other classes are multi-programmed workloads using part of the SPEC CPU2000 [39] applications. The benchmarks are evaluated in rate mode (one instance of the program per available processor) and with reference inputs. For each simulation point a variable number of runs is performed with pseudo-random perturbation in order to estimate workload variability [3]. All the results provided have a 95% confidence interval.



(a)

(b)

Fig. 4. WH-BASE Normalized (a) Execution Time, (b) Network ED2P.

6. PERFORMANCE EVALUATION

As well as this full-system evaluation, we previously performed an exhaustive analysis using synthetic workloads and different network configurations in order to obtain results under specific conditions (see Appendix 5). Although the results provided in Appendix 5 clearly show the advantage of our proposal, it is essential to also provide full system results running real workloads to present conclusive results.

Fig. 4(a) presents the WH-BASE normalized execution time for the workloads considered. In general, the simplicity of the WH-BASE router has significant impact in performance with more than 10% increment in execution time for any application versus other routers. Comparing its behavior with WH-ADV, in some situations such as the FT case, the time for executing the application increases by up to 40%. For this particular system, even though it is not especially aggressive in terms of the number of cores or their configuration, it seems that an excessive emphasis in saving network resources implies performance degradation. The range of loss depends upon the application nature. Applications that exhibit a higher network pressure undergo more performance loss.

For this realistic configuration, our proposal achieves the best performance, the benefit being substantial in some network-demanding applications such as FT. Under realistic working conditions, the Rotary Router is not just better because of its better high-contention behavior, as was proved in the previous section, but also because of the lower latency of the protocol messages with highest priority. Due to the flow control of in-transit messages in the Rotary Router, messages with higher priority in the message dependence chain advance faster, because they can occupy a bigger portion of buffering resources.

Finally Fig. 4(b) shows the interconnection network Energy Delay^2 Product. On average, the ED2P reduction achieved by the Rotary Routers is clear. The Rotary Router requires continuous packet movement between DFB. This fact increases the dynamic energy consumption of the network. Nevertheless, on average the number of ring turns per router ranges from 0.4 at low load to 0.75 at high load. The significant performance benefit can compensate that fact, being able to reduce the ED2P almost 40% on average.

In some applications, especially those with low network utilization, the Rotary Router has a poor ED2P. Nevertheless, note that the rest of the system (processors, cache, memory, etc…) is identical for the three routers considered. It seems to be accurate to assume that the average power of the rest of the system when executing the application will be very similar. Consequently, the execution time reduction will have a larger impact on full system ED2P.

In summary, these results clearly indicate that saving network power by reducing its performance has undesirable effects. When the performance-energy tradeoff is considered a high performance router is desired.

7. Conclusions

A new interconnection network approach has been presented, based on novel router architecture, which is especially suitable for CMP systems. The router utilizes a decentralized and scalable structure based on two rings which force packets to circulate either clockwise or counterclockwise, traveling from port to port of the router. Packet circulation has important benefits such as HOL blocking avoidance, which provides the appropriate substrate to implement a deadlock avoidance mechanism that is topologically independent and does not require virtual channels. The elimination of the necessity for virtual-channels almost completely eliminates the hardware overhead required to implement performance-oriented features such as adaptive routing or congestion control policies.

Additionally, A VC-less structure such as the one presented in this work not only reduces the arbitration complexity and optimizes the usage of the scarce buffering resources available, but it also allows the implementation of a low-cost solution for the end-to-end deadlock problem induced by traffic dependencies inherent to any coherence protocol. Without hardware replication, router complexity is maintained constant independently of the number of message classes.

Finally, a representative group of synthetic traffic patterns and a wide range of realistic applications demonstrate the significant advantages of the Rotary Router when compared to other performance-optimized or cost-optimized router architectures.

8. References

[1] P. Abad, V. Puente, J.A. Gregorio, P. Prieto, "Rotary router: an efficient architecture for CMP interconnection networks", Int. Symposium on Computer Architecture (ISCA), 2007.

[2] P. Abad, V. Puente, J.A. Gregorio, P. Prieto, "Reducing the Interconnection Network Cost of Chip Multiprocessors, (NOCS), 2008.

[3] Alameldeen, A.R., Mauer, C.J., Xu, M., Harper, P.J., Martin, M.M.K., Sorin, D.J., Hill, M.D., and Wood, D.A. "Evaluating non-deterministic multi-threaded commercial workloads". Proceedings of the Fifth Workshop on Computer Architecture Evaluation Using Commercial Workloads (2002), 30–38

[4] J. Balfour, W. Dally, "Design Tradeoffs for Tiled CMP On-Chip Networks", International Conference on Supercomputing (ICS) 2006.

[5] S.Bell, et al., "TILE64TM Processor: A 64-Core SoC with Mesh Interconnectc ", ISSCC 2008.

[6] D.M. Brooks, et al, "Power-Aware Microarchitecture: Design and Modeling Challenges for Next-Generation Microprocessors", IEEE Micro. Volume 20, Issue 6, November 2000.

[7] D. Burger, S. Keckler, K. McKinley, M. Dahlin, L. John, C. Lin, C. Moore, J. Burrill, R. McDonald, W. Yoder "Scaling to the end of Silicon with EDGE Architectures", IEEE Computer. vol. 37, no 7, pp.44-55, July 2004.

[8] L. Cheng, N. M., K. Ramani, R. Balasubramonian, J.B. Carter, "Interconnect-Aware Coherence Protocols for Chip Multiprocessors". International Symposium on Computer Architecture (ISCA) 2006.

[9] W. Dally, B. Towles, "Route Packets, Not Wires: On-Chip Interconnection Networks", Design Automation Conference (DAC) 2001.

[10] W. Dally, B. Towles, "Principles and Practices of Interconnection Networks". Morgan Kaufimann, 2004.

[11] Jin, H., Frumkin, M., and Yan, J., "The OpenMP Implementation of NAS Parallel Benchmarks and its Performance", NASA Ames Research Center, editor, Technical Report NAS-99-01 (1999)

[12] R. Gonzalez, M. Horowitz, "Energy Dissipation In General Purpose Microprocessors", IEEE Journal of Solid-State Circuits, Vol. 31, No. 9, pp. 1277-1284, September 1996.

[13] M. Hayenga, N.E. Jerger, M. Lipasti, "SCARAB: A Single Cycle Adaptive Routing and Bufferless Network", International Symposium on Microarchitecture, December 2009.

[14] .Intel, An introduction to Quick Path Interconnect http:/ / www.intel.com/ technology/ quickpath/ introduction.pdf

[15] P. Kermani, L. Kleinrock, "Virtual Cut-Through: A New Computer Communication Switching Technique". Computer Networks, Vol. 3, pp. 267-286, September 1979.

[16] C. Kim, D. Burger, S. W. Keckler. "An Adaptive, Non-Uniform Cache Structure for Wire-Dominated On-Chip Caches", International Confe-

rence on Architectural Support for Programming Languages and Operating Systems (ASPLOS), 2002.

[17] J. Kim, “Low-Cost Router Microarchitecture for On-Chip Networks”, International Symposium on Microarchitecture, December 2009.

[18] J. Kim, C. Nicopoulos, D. Park, V. Narayanan, M. S. Yousif, C. R. Das, “A Gracefully Degrading and Energy-Efficient Modular Router Architecture for On-Chip Networks”, Int. Symposium on Computer Architecture (ISCA), 2006.

[19] P. Kongetira, K. Aingaran, K. Olukotun, “Niagara: A 32-way Multithreaded SPARC Processor”, IEEE Micro. Vol. 25, No. 2, pp. 21-29, March/ April 2005.

[20] S. Konstantinidou, L. Snyder, “The Chaos Router”, IEEE Trans. Computers, Vol. 43, No. 12, pp. 1386-1397, December 1994.

[21] A. Kumar, P. Kundu, A. P. Singh, L-S. Peh, N.K. Jha, “A 4.6Tbits/ s 3.6GHz single-cycle NoC router with a novel switch allocator in 65nm CMOS“.International Conference on Computer Design, October 2007.

[22] S. E. Lee, N. Bagherzadeh, “Increasing the Throughput of an Adaptive Router in Network-on-Chip (NoC)” CODES+ISSS’06, 2006.

[23] D. Lenoski, J. Laudon, K. Gharachorloo, W. Weber, A. Gupta, J. Hennessy, M. Horowitz, M. Lam, “The Stanford Dash Multiprocessor”, IEEE Computer Vol. 25, No. 3, pp. 63-79, March 1992.

[24] J. Laudon, D. Lenosky, “The SGI Origin: A ccNUMA Highly Scalable Server”, Int. Symposium on Computer Architecture (ISCA), 1997.

[25] P. S. Magnusson, M. Christensson, J. Eskilson, D. Forsgren, F. Larsson, A. Moestedt, B. Werner, “Simics: A Full System Simulation Platform”. Computer, Vol. 35, No.2, pp. 50-58, February 2002.

[26] M. Martin, D. Sorin, B. Beckmann, M. Marty, M. Xu, A. Alameldeen, K. Moore, M. Hill, D. Wood, “Multifacet’s General Execution-driven Multiprocessor Simulator (GEMS) Toolset”, SIGARCH Comput. Archit. News, Vol.33, No.4, pp.92–99, November 2005.

[27] M. Martin, M. Hill, and D. Wood, “Token Coherence: Decoupling Performance and Correctness”, Int. Symposium on Computer Architecture (ISCA), June 2003.

[28] T. Moscibroda, O. Mutlu, “A case for bufferless routing in on-chip networks”, Int. Symposium on Computer Architecture (ISCA), 2009.

[29] G. Michelogiannakis, J. Balfour, and W. J. Dally, “Elastic-buffer flow control for on-chip networks”, Int. Symposium on High-Performance Computer Architecture (HPCA) 2009.

[30] S. Mukherjee, P. Bannon, S. Lang, A. Spink, D. Webb, “The Alpha 21364 Network Architecture”, IEEE Micro, vol. 22, no. 1, pp 26-35, Jan-Feb 2002.

[31] R. Mullins, A. West, S. Moore “Low-Latency Virtual-Channel Routers for On-Chip Networks”, Int. Symposium on Computer Architecture (ISCA), 2004.

[32] C. Nicopoulos, D.Park, J.Kim, N.Vijaykrishnan, M.S. Yousif, C.R. Das, “ViChaR: A Dynamic Virtual Channel Regulator for Network-on-Chip Routers”, MICRO 2006.

[33] K. Olukotun, L. Hammond, “The future of Microprocessors” ACM Queue, Vol. 3, No. 7, September 2005.

[34] L. Peh, W. Dally, “A Delay Model and Speculative Architecture for Pipelined Routers”, Int. Symposium on High-Performance Computer Architecture (HPCA) 2001.

[35] H. Hofstee, "Power Efficient Processor Architecture and The Cell Processor", Int. Symposium on High-Performance Computer Architecture (HPCA), 2005.

[36] P.Pande, C.Grecu, M. Jones, A.Ivanov, R.A. Saleh, “Performance Evaluation and Design Trade-Offs for Network-on-Chip Interconnect Architectures”, IEEE Trans. Computers, Vol. 54, No. 8, pp.1025-1040, February 2005.

[37] V. Puente, C. Izu, R. Beivide, J.A. Gregorio, F. Vallejo, J.M. Prellezo, “The Adaptive Bubble Router”, Journal of Parallel and Distributed Computing, Vol. 61, No. 9, September 2001.

[38] V. Puente, J.A. Gregorio, R. Beivide, “SICOSYS: An Integrated Framework for studying Interconnection Network in Multiprocessor Systems”, Euromicro Workshop on Parallel and Dist. Processing, 2002.

[39] SPEC2000, http:/ / www.spec.org/ cpu2000/

[40] S. Scott and G. Thorson, “The Cray T3E Network: Adaptive Routing in a High Performance 3D Torus”, Hot Interconnects IV, August 1996.

[41] Y.H. Song, T.M. Pinkston, “A Progressive Approach to Handling Message-Dependent Deadlock in Parallel Computer Systems”, IEEE Trans. on Parallel and Distributed Systems, Vol. 14, No. 3, pp 259-275, March 2003.

Pablo Abad was born in Reinosa, Cantabria. He received his BS, MS and PhD degree from the University of Cantabria, Spain, in 2003 and 2010 respectively. He currently works as assistant professor of Computer Architecture. His research interests are focused on on-chip interconnection network design, as well as their interaction with the rest of system components as part of CMP memory hierarchy.

Valentin Puente was born in Vendejo, Cantabria. He received the BS, MS and PhD degree from University of Cantabria, Spain, in 1995 and 2000 respectively. He is currently an Associate Professor of Computer Architecture at the same University. His research interests include interconnection networks, multithreaded architectures, and performance evaluation. He is a member of the IEEE Computer society.

José Angel Gregorio was born in Bareyo, Cantabria (Spain). He received his BS, MS and PhD in Physics (Electronics) from the University of Cantabria, in 1978 and 1983, respectively. He is currently a professor of computer architecture in the Department of Electronics and Computers in the same University. His research interests include parallel and distributed computers, interconnection networks, and performance evaluation of computers and communication systems. He is a member of the IEEE Computer society.