2008 14th IEEE International Conference on Parallel and Distributed Systems

Quarc: a Novel Network-on-Chip Architecture

M. Moadeli[1], W. Vanderbauwhede[1], A. Shahrabi[2]
1: Department of Computing Science
University of Glasgow, Glasgow, UK
Email: {mahmoudm, wim}@dcs.gla.ac.uk
2 : School of Engineering and Computing
Glasgow Caledonian University, Glasgow, UK
Email: a.shahrabi@gcal.ac.uk

Abstract

This paper introduces the Quarc NoC, a novel NoC architecture inspired by the Spidergon NoC [16]. The Quarc scheme significantly outperforms the Spidergon NoC through balancing the traffic which is the result of the modifications applied to the topology and the routing elements. The proposed architecture is highly efficient in performing collective communication operations including broadcast and multicast. We present the topology, routing discipline and switch architecture for the Quarc NoC and demonstrate the performance with the results obtained from discrete-event simulations.

1 Introduction

The Network-on-Chip (NoC) concept is an emerging communication-centric architecture for future complex System-on-chip (SoC) design providing scalable, energy efficient and reliable communication. In a NoC-based system, different components such as computation elements, memories and specialized IP blocks exchange data using a network as a communication infrastructure.

Designing a flexible on-chip communication network for a NoC platform, which can provide the desired bandwidth and at the same time be reused across many applications, is a challenging task which requires trading-off between a number of cross-cutting concerns such as performance, cost and size. In addition to the technology in which the hardware is implemented, the topology, switching method, routing algorithm and the traffic pattern are some other key factors which have direct impact on the performance of a NoC platform.

To meet these challenges, research carried out in the field has proposed the idea of using a packet switched communication network for on-chip communication. A packet switched NoC consists of an interconnection of many routers that connect IPs together to form a given topology in order to enable a large number of units (cores) to communicate with each other. The underlying topology of this architecture is the key element of on-chip network, since it provides a low latency communication mechanism and, when compared to traditional bus-based approaches, resolves physical limitations due to wire latency providing higher bandwidth and parallelism.

Deterministic routing and wormhole switching are regarded as the dominant routing and switching mechanism in the NoC domain [22]. Those options mainly originate from the resource constraints at intermediate routers [8, 22].

Most recent proposed NoC architectures have been founded on top of ring, fat-tree or 2D mesh topologies as they have an area efficient layout on a two dimensional surface which is most suitable for NoC design. Nostrum [18], Æthereal [9], and Xpipes [17] are some examples of architectures used for on-chip networks. The Spidergon NoC [16] is also one of the ring-based architectures proposed recently.

By adopting wormhole switching, deterministic routing and homogeneous, low-degree routers; the Spidergon scheme aimed to address the demand for a fixed and optimized network on-chip architecture to realize cost effective MPSoC development. However, the edge-asymmetric property of the Spidergon causes the number of messages that cross each physical link varies severely, resulting in an unbalanced traffic on network channels and, thus, leading to poor performance of the whole network. This situation is even exacerbated when the network is under bursty traffic as a result of some operations such as broadcast.

In this paper we propose the Quarc (Quad-arc) scheme; a novel NoC architecture. The novelty of the Quarc NoC lies both in the topology it adopts and its router architecture. While preserving all features of the Spidergon, the

1521-9097/08 $25.00 © 2008 IEEE
DOI 10.1109/ICPADS.2008.53

Quarc scheme introduces an extra physical link to the cross link of the Spidergon to separate right-cross-quarter from left-cross-quarter to balance the traffic. It also employs an all-port router architecture to reduce the message blocking latency during collective communication operations. The Quarc NoC's features result in a NoC that is highly efficient in exchanging all types of traffic. In particular as paper shows the Quarc NoC is highly efficient for performing collective communication operations.

The rest of the paper is organized as follows. Section 2 introduces the Quarc NoC. It then investigates the architecture of the switches. Routing discipline, including unicast and broadcast, is also presented in this section. Section 3 studies the performance of the Quarc scheme compared to the Spidergon NoC. Finally, we make concluding remarks in Section 4.

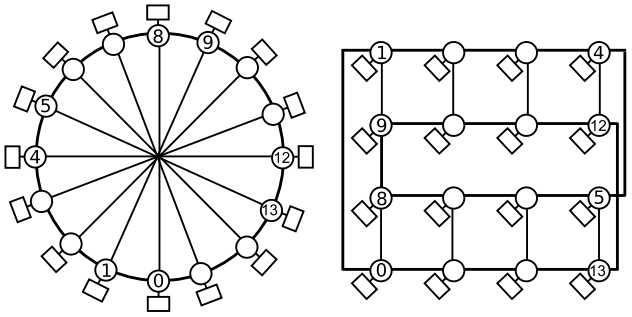

Figure 1. The Spidergon topology and the on chip layout.

2 Quarc: A NoC Architecture

The topology of an on-chip network specifies the structure in which routers connect the IPs together. Fat tree, mesh, torus and variations of rings are among the topologies introduced or adopted for the NoC domain.

Typically, a particular topology is selected in order to trade-off between a number of cross-cutting measures such as performance and cost. A number of important characteristics that affect the decision on adopting a particular topology are network diameter, the highest degree of nodes in the network, regularity, scalability and synthesis cost for an architecture.

The topology of the Quarc NoC is quite similar to that of the Spidergon NoC. Therefore, the next section presents a brief description of the Spidergon NoC, followed by introduction of the Quarc NoC.

2.1 The Spidergon NoC

The Spidergon NoC [16] is a network architecture which has been recently proposed by STMicroelectronics [20]. The objective of the Spidergon topology has been to address the demand for a fixed and optimized topology to realize low cost multi-processor SoC implementation. In the Spidergon topology an even number of nodes are connected by unidirectional links to the neighboring nodes in clockwise and counter-clockwise directions plus a cross connection for each pair of nodes. Each physical link is shared by two virtual channels in order to avoid deadlock. Fig. 1 depicts a Spidergon topology of size 16 and its layout on a chip.

The key characteristics of the this topology include good network diameter, low node degree, homogeneous building blocks (the same router to compose the entire network), vertex symmetry and simple routing scheme. Moreover, the Spidergon scheme employs packet-based wormhole routing which can provide low message latency at a low cost. Furthermore, the actual layout on-chip requires only a single crossing of metal layers.

In the Spidergon NoC, two links connecting a node to surrounding neighboring nodes carry messages destined for half of nodes in the network, while the node is connected to the rest of the network via the cross link. Therefore, the cross link can become a bottleneck. Also, since the router at each node of the Spidergon NoC is a typical one-port router, the messages may block on occupied injection channel, even when their required network channels are free. Moreover, performing broadcast communication in a Spidergon NoC of size N using the most efficient routing algorithm requires traversing $N - 1$ hops.

2.2 The Quarc

We propose the Quarc (Quad-arc) NoC, which improves on the Spidergon by making following changes: *(i)* adding an extra physical link to the cross link to separate right-cross-quarter from left-cross-quarter, *(ii)* enhancing the one-port router architecture to an all-port router architecture and *(iii)* enabling the routers to absorb-and-forward flits simultaneously. The Quarc preserves all features of the Spidergon including the wormhole switching and deterministic shortest path routing algorithm, as well as the efficient on-chip layout.

The resulting topology for an 8-node NoC is represented in Fig. 2.

Unlike the Spidergon NoC, in the Quarc architecture a messages will be blocked only when its requested network resources are occupied. This feature significantly enhances the performance of the network by reducing the waiting time at source node. Moreover, adding another physical

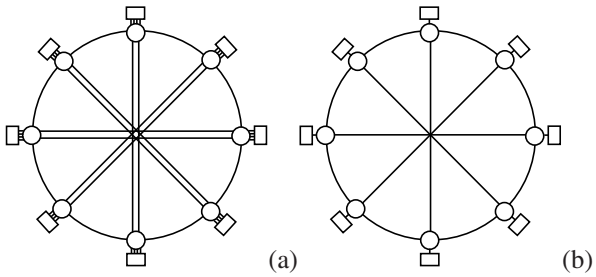

Figure 2. Quarc topology (a) vs Spidergon (b)

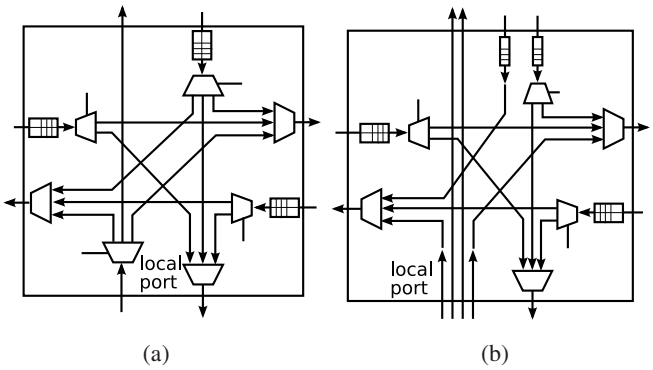

Figure 3. Minimal switch architectures for Spidergon (a) and Quarc (b) with deterministic routing

link to the cross network links improves access to the cross-network nodes. And last but not the least, the effect of the modification manifests itself most clearly when performing broadcast or multicast communication operations. In the Spidergon NoC, deadlock-free broadcast can only be achieved by consecutive unicast transmissions. The NoC switches must contain the logic to create the required packets on receipt of a broadcast-by-unicast packet. In contrast, the broadcast operation in the Quarc architecture is a true broadcast, leading to much simpler logic in the switch fabric; furthermore, the latency for broadcast traffic is dramatically reduced.

The next section demonstrates that, surprisingly, the modifications proposed to the Spidergon topology and switch architecture to obtain the Quarc do not adversely affect area consumption of the resulting NoC compared to the original Spidergon. On the contrary, we demonstrate that the proposed modifications lead to both smaller switches and simpler routing logic.

2.3 Switch architecture

In this section we compare the switch architectures of the Quarc and Spidergon NoCs. Fig. 3 shows simplified diagrams for a Spidergon 4×4 switch with 1 local channel and 3 network channels (Fig 3(a)) and the Quarc architecture (Fig 3(b)). Both diagrams show minimal architectures for use with deterministic routing, i.e. the hardware is tailored to the paths allowed by the routing discipline.

The main differences are the number of local ingress ports (4 for Quarc) and the doubling of the cross-network link. Further differences are not obvious from the figure: the Quarc switch performs a true broadcast, so the ingress multiplexers have a state that clones the flit; the decision logic is very simple (see 2.5). The Spidergon switch can only broadcast by unicast, and therefore needs a more complex logic to decide if a switch needs to clone a broadcast packet; furthermore, the ingress packet is not simply cloned but the header flit needs to be rewritten.

2.4 Cost Analysis of the Quarc Architecture

In this section, we argue that the Quarc switch is both smaller and less complex than the Spidergon switch, and that this saving more than outweighs the overheads caused by the additional ports, both in terms of the complexity of the processing element and the area consumed by the additional links.

Virtual Channels in the NoC are implemented as a small bus in parallel with the data; the bus controls a demux at the receiving node which directs flits into the buffer for each Virtual Channels. A separate bus signals flit buffer occupancy to the transmitting node. Note that the Quarc only requires Virtual Channels for the non-cross links, as the cross links are not used for forwarding.

We assume that every node serves a processing element (PE), typically a microprocessor with local memory. The difference in resource utilisation at the PE between the Quarc and the Spidergon is very small: in both cases the packets are stored in RAM, the addresses of the packets are queued. For Quarc, the PE will queue the addresses in 4 queues, effectively making the routing decision by doing so. For the Spidergon, the PE will put the addresses in a single queue. As the variance on the occupation of the individual queues (σ for Quarc), is twice as large as the variance on the occupation of the combined queue ($\sigma/\sqrt{4}$ for Spidergon), the queues will need to be twice as deep. This is a small memory overhead as the address size is a fraction of the packet size. Note that the actual packet memory requirements are identical for Quarc and Spidergon.

The key difference between the Quarc and the Spidergon switches is the local ingress port. In terms of the complexity and size of the buffers, multiplexers and demultiplexers

required by the other ports, there is not difference. The 4 ports of the Quarc translate to 4 addresses on the processor bus, instead of a single address for the Spidergon. The two non-cross ports require a flit buffer, but only a single buffer (i.e. no separate buffers per virtual channel) as there is only one destination. By comparison, the Spidergon's local port demultiplexer requires a flit buffer per virtual channel. Consequently, the hardware requirements for the Quarc switch are lower than for the Spidergon switch. Futhermore, the Spidergon switch needs to calculate the output port based on the flit header (see 2.5); the Quarc only needs to compare the destination address with the switch address to decide if the packet needs to be delivered locally or forwarded. Thus the routing infrastructure in the Quarc switch is almost non-existent, which again reduces the complexity and area of the switch.

For the TSMC 90 nm CMOS process, the metal pitch is $240\,nm$ [13]. Consequently, a bidirectional 64-bit NoC channel will be about $32\,\mu m$ wide. Let's assume we would like to use our NoC to create a 16-core Cell processor [7] SoC. The Cell's area in 90 nm is $221\,mm^2$ [14]. This includes the IO ring, so the actual core size would be slightly smaller, $200\,mm^2$ assuming an IO ring of $400\,\mu m$. The length of the NoC channel would then be $14\,mm$, so the area would be $14.32/1000\,mm^2$. The ratio of the total channel area to the total area (core+channels) would be $(24 \times 14 \times 32/1000)/(200 \times 16 + 24 \times 14 \times 32/1000)$, in other words the NoC channels consume less than 0.35% of the total area. Increasing the number of channels from 24 to 32 (a 16-node Spidergon or mesh has 24 links, a 16-node Quarc has 32) would take this figure up to 0.45%. Consequently the area cost of the additional link is very small. One could argue that the additional links will consume additional power, but that would only be the case if the links are clocked when idle.

2.5 Routing algorithm

2.5.1 Unicast routing

Spidergon On the Spidergon, deterministic routing is quite simple: for any packet arriving from the cross-network link and not destined for the local port or arriving from the local port, the router calculates the quadrant of the destination relative to its own address.

Calculating the quadrant (q) is simple. We first give the algorithm and then an implementation at bit level suitable for hardware.

- Let N be the number of nodes, N_s the absolute source node address, N_d the absolute destination node address.
- Renormalise the destination address (N_r):

$$\begin{aligned} N_d > N_s &\Rightarrow N_r = N_d - N_s \\ N_d < N_s &\Rightarrow N_r = N_d - N_s + N \end{aligned}$$

- Determine the quadrant q:

$$N_r \leq \frac{N}{4} \Rightarrow q = 0$$

$$\frac{N}{4} < N_r \leq \frac{N}{2} \Rightarrow q = 1$$

$$\frac{N}{2} < N_r \leq 3\frac{N}{4} \Rightarrow q = 2$$

$$N_r > 3\frac{N}{4} \Rightarrow q = 3$$

The following possible approach uses 2's complement arithmetic with bit shifts to illustrate the small amount of hardware required to perform the operations.

Let $n = N - 1$, $b = log_2(N)$

Then, assuming d to be a word of b bits wide, c a single bit and q two bits:

$$\begin{aligned} N_r &= ((N_d + (\sim N_s + 1))\&n) \\ c &= N_r\hat{}(n \gg 2)?(\sim 1 + 1) : 0 \\ q &= ((N_r + c)\&(3 \ll (b-2))) \gg (b-2) \end{aligned}$$

Packets arriving on the cross-network link are sent either left ($q = 3$) or right ($q = 0$); packets arriving on the local port are sent left ($q = 3$), right ($q = 0$) or up ($q = 1$ and $q = 2$).

For packets received from the left or right nodes, the packet may be sent to the PE of the local node or it may be further transmitted along the rim.

Quarc For the Quarc, the surprising observation is that there is no routing required by the switch: packets are either destined for the local port or forwarded to a single possible destination. Consequently, the proposed NoC switch requires no routing logic. The route is completely determined by the port in which the packet is injected by the source. Of course, the NoC interface (transceiver) of the source processing element (PE) must make this decision and therefore calculate the quadrant as outlined above. However, in general the PE transceiver must already be NoC-aware as it needs to create the header flit and therefore look up the address of the destination PE. Calculating the quadrant is a very small additional action.

2.5.2 Broadcast operation

Collective communications operations have been traditionally adopted to simplify the programming of applications for parallel computers, facilitate the implementation of efficient communication schemes on various machines, and promote the potability of applications across different architectures [11]. These communication operations are particularly useful in applications which often require global data movement and global control in order to exchange data and synchronize the execution among nodes. The most widely used collective communication operations are *broadcast*, *multicast*, *scatter*, *gather* and *barrier synchronization*.

The support for collective communication may be implemented in *software* or/and *hardware*. The software-based approaches [10] rely on unicast-based message passing mechanisms to provide collective communication. They mostly aim to reduce the height of multicast tree and minimize the contention among multiple unicast messages.

Software-based approaches typically have limitations in delivering the required performance. Implementing the required functionality partially or fully in hardware has proved to improve the performance of collective operations. Depending on required performance, the hardware support for collective communication may be achieved by customizing the switching [4], routing, number of ports [6] or even allocating a dedicated network for collective communication operations.

Hardware-based multicast schemes can be broadly classified into *path-based* and *tree-based.* In a path-based approach, the primary problem for multicasting is finding the shortest path that covers all node in the network [11]. After path selection, the intermediate destinations perform absorb-and-forward operations along the path. Hamilton path-based algorithm [5] and the Base Routing Conformed Path (BRCP) approach [1] are examples of path-based algorithms utilizing absorb-and-forward property at hardware layer.

In the tree-based scheme, the multicast problem is finding a Steiner tree with a minimal total length to cover all network nodes [2]. The tree operation introduces additional network resource dependencies which could lead to deadlock which is difficult to avoid if global information is not available. Hence, in wormhole-routed direct networks, the tree based multicast is usually undesirable, unless the messages are very short.

Broadcast and multicast traffic in Networks on Chip is an important research field that has not received much attention. A multicasting scheme for a circuit-switched network on chip proposed in [12]. Since the scheme relies on the global network state using global traffic information it is not easily scalable. Multicast operation is provided by Æthereal NoC [15]. However, Æthereal relies on a logical notion of global synchronicity which is not trivial to implement as the system scales. In [3] a multicast scheme in wormhole-switched NoCs is proposed. By this scheme, a multicast procedure consists of establishment, communication and release phase. A multicast group can request to reserve virtual channels during establishment and has priority on arbitration of link bandwidth.

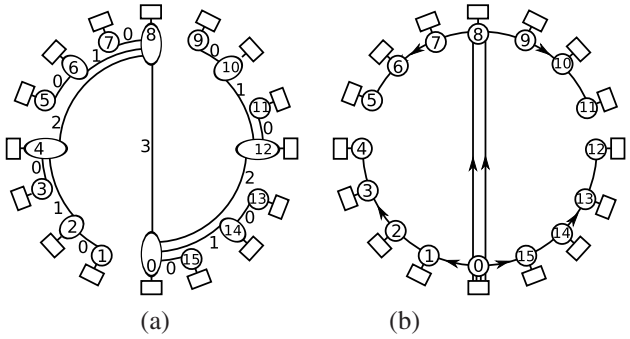

Figure 4. Broadcast in Spidergon (a) and Quarc (b) NoCs

Spidergon Broadcast in the Spidergon most efficiently may be handled by unicast with a "unicast tree" algorithm depicted in Fig. 4(a). The initiating node 0 sends a packet to node $N/2$; nodes 0 and $N/2$ send a packet to $N/4$ and $N/2 + N/4$; all 4 nodes send a packet to nodes $N/8$, $N/4+N/8$, $N/2+N/8$, $N/2+N/4+N/8$ and so on. Because this is a multi-stage process ($log_2 N$ stages) the broadcast packet needs a decrementing count field to identify the stage of the broadcast process. When a NoC switch receives a broadcast packet, it must take following decisions:

1. Is the current node a destination node or a forwarding node? The rule for this decision is: if the distance between the source address and the node address is smaller than the value of the count field, the packet must be forwarded (on the rim). Otherwise, the packet is received by the local node. So the actions to perform are:

   - Renormalise the address $N_d \rightarrow N_r$ (see above)
   - Compare N_r against the value of the count field

   If the packet is received, proceed to the next step.

2. Is further broadcast required? The rule for this decision is: if the count field is 0, no further broadcast is required.

3. If further broadcast is required, how many packets need to be sent? The number of packets to be sent is

given by the count field of the ingress packet. Essentially, the switch decrements the count field and forwards the packet along the rim. This means that the switch must buffer the packet for the duration of the broadcast and decrement the count field in the buffered packet before each transmission, until the count is 0.

The problem with this scheme (and in general with broadcast-by-unicast) is that the switch requires buffer space for every broadcast packet. In a large network with a number of concurrent broadcasts, the buffer requirements will significantly increase the area of the switch.

Quarc Broadcast in the Quarc is much more elegant and efficient: The Quarc NoC adopts a BRCP (Base Routing Conformed Path) [1] approach to perform multicast/broadcast communications. BRCP is a type of path-based routing in which the collective communication operations follow the same route as unicasts do. Since the base routing algorithm in the Quarc NoC is deadlock-free, adopting BRCP technique ensures that the broadcast operation, regardless of the number of concurrent broadcast operations, is also deadlock-free.

To perform a broadcast communication the transceiver of the initiating node has to broadcast packet on each port of the all-port router. The transceiver tags the header flit of each of four packets destined to serve each branch as broadcast to distinguish it from other types of traffic. The transceiver also sets the destination address of each packet as the address of the last node that the flits stream may traverse according to the base routing. The receiving nodes simply check if the destination address at the header flit matches its local address. If so, the packet is received by the local node. Otherwise, if the header flit of the packet is tagged as broadcast, the flits of the packet at the same time are received by the local node and forwarded along the rim. This is simply achieved by setting a flag on the ingress multiplexer which causes it to clone the flits.

The broadcast in a Quarc NoC of size 16 is depicted in Fig. 4(b). Assuming that Node 0 initiates a broadcast, it tags the header flits of each stream as broadcast and sets the destination address of packets as 4, 5, 11 and 12 which are the address of the last node visited on left, cross-left, cross-right and right rims respectively. The intermediate nodes receive and forward the broadcast flit streams, while the destination node absorbs the stream.

3 Performance analysis

This sections describes in details the simulator developed to evaluate the performance of the system followed by presenting the average latencies in the Quarc and the Spidergon in a variety of working configurations.

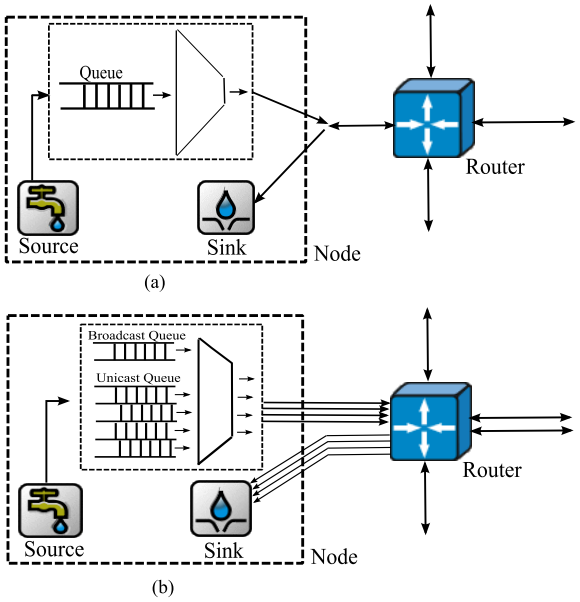

Figure 5. The schematic of a sample simulation node in the Spidergon (a) and Quarc (b) NoCs.

3.1 NoC Simulator

To evaluate the performance of the Quarc NoC architecture we have developed a discrete event simulator operating at flit level using OMNET++ [21]. The simulator has been verified extensively against analytical models for the Spidergon and mesh topologies employing wormhole routing [19]. The schematic of the components of each node in the Quarc and the Spidergon NoCs are shown in Fig. 5. The source produces the messages according to a Poisson distribution. The passive queue has queues to store the messages and sends the messages based on their creation time. The passive queue is connected to the router through four injection channels in the Quarc NoC and via one injection channel in the Spidergon NoC. The router is connected to three neighboring routers, a sink and a passive queue. It receives the flits of the messages and sends them to the appropriate routers or its corresponding sink. The sink absorbs the messages destined for it from the router.

It is worth mentioning that in the Quarc NoC a broadcast message starts its transmission only when all injection channels are free. Therefore, if one or more channels are occupied by broadcast or unicast messages the possible broadcast transmission are not performed.

The simulator operates on the following assumptions. A network cycle is defined as the time required that a flit traverse between two adjacent router or between a router and a sink or passive queue. The time consumed in the routers

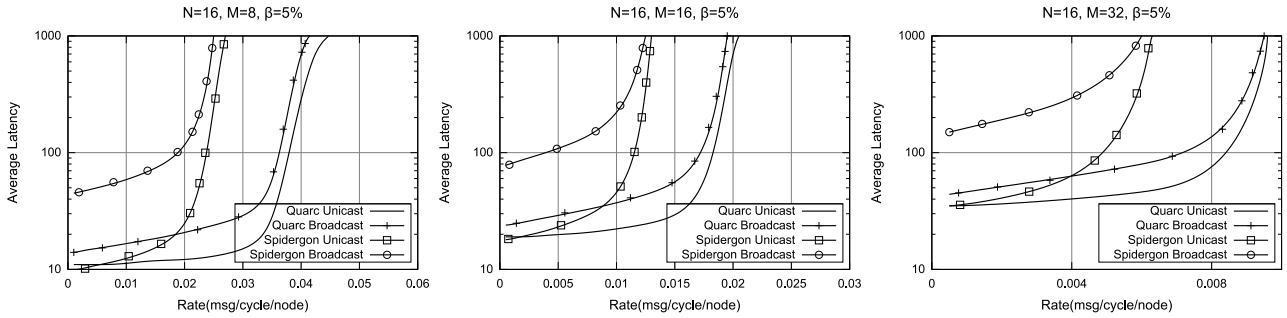

Figure 6. Comparison of Quarc and Spidergon for M=8,16,32

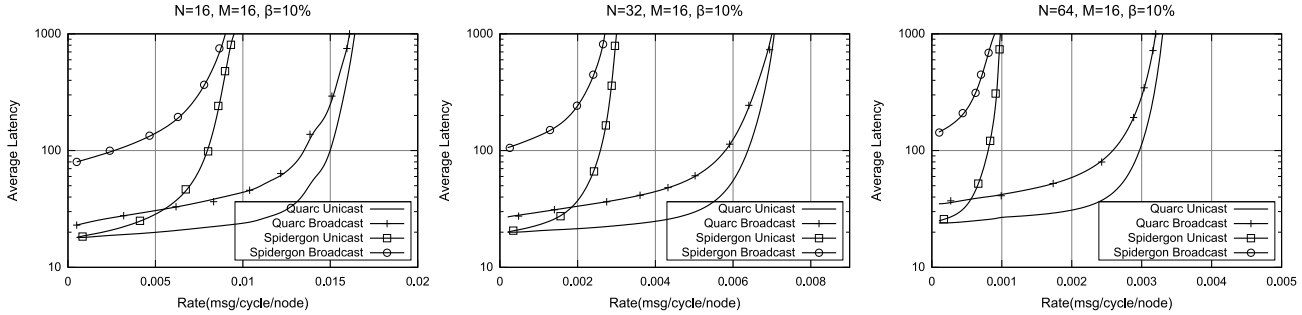

Figure 7. Comparison of Quarc and Spidergon for N=16,32,64

is ignored in simulation. Also all messages are assumed to be of equal size.

Destinations of unicast messages at each node are selected randomly. The latency of a unicast message is regarded as the time from generation of unicast message at the source node until the time when the last flit of the message is absorbed by the sink at destination. Broadcast message latency is the time from generation of the broadcast message at the source node until the time when the last flit of the message is absorbed by the sink at the last receiver of the broadcast message.

3.2 Analysis of the simulation results

The performance of the Quarc architecture has been evaluated against the Spidergon for numerous configurations by changing the network size, message length and the rate of broadcast traffic. In graphs, N, M and β represent the number of nodes, message length and rate of broadcast traffic respectively. The horizontal axis in the figures shows the message rate per node while the vertical axis describes the latency.

Fig. 6 shows the average latency experienced by unicast and broadcast traffic in the Quarc and Spidergon NoCs in configurations where network size $N = 16$ and broadcast rate, $\beta = 5\%$ are fixed while the message length can be 8, 16 and 32. Fig. 7 compares the simulation results against the analysis for the networks ranging from 16 to 64 nodes with a fixed message length of 16 and 10% broadcast traffic.

As can be seen from the figures the Quarc NoC outperforms the Spidergon over the complete range of N, M and β . The most striking performance difference is clearly observed for broadcast traffic, with almost an order of magnitude improvement on the latency. However, the unicast latency is overall at least a factor of 2 lower. Also, the graphs clearly show that the Quarc NoC is capable of sustaining a much higher load before it saturates. This in turn indicates that the throughput of the Quarc NoC is significantly higher than the Spidergon NoC.

The graphs in Fig. 8 compare the average latency in the Quarc and Spidergon NoC for the configuration where the network size ($N = 64$) and message length ($M = 16$) are fixed while the broadcast rate, B, is varying between 0 to 10%. The graphs reveal the Quarc NoC is highly capable of sustaining the broadcast traffic. As can be seen the injection of the broadcast traffic into the Spidergon NoC severely reduces the sustainable load in the network. In the Quarc NoC the adverse impact of the broadcast traffic on the sustainable load and on the performance of the unicast is hardly appreciable.

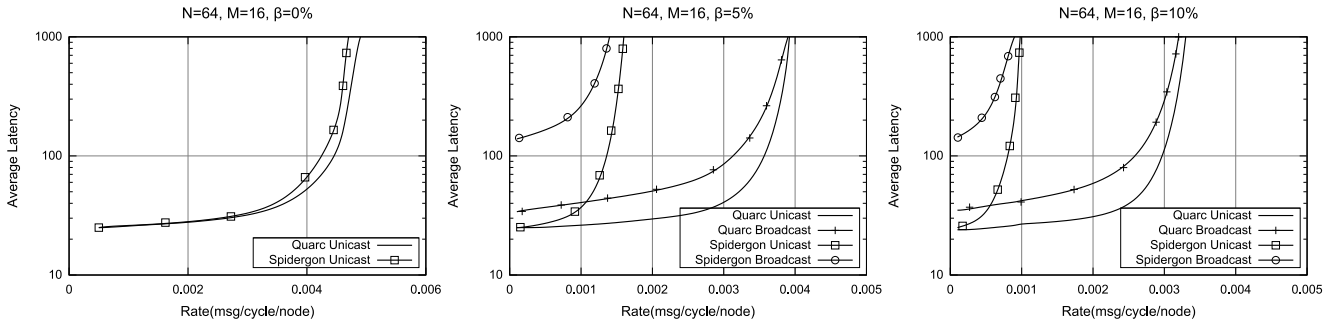

Figure 8. Comparison of Quarc and Spidergon for β= 0%, 5%, 10%

4 Conclusion

In this paper we have proposed a novel Network-on-Chip architecture, the Quarc NoC, which is inspired by the Spidergon but introduces a number of modifications which significantly enhance the performance of the unicast and collective communication. The Quarc addresses a key issue with the Spidergon architecture: unbalanced traffic due to its edge-asymmetric property and consequently to poor performance under bursty traffic, such as broadcast. The major achievement of the Quarc NoC and thus the main contribution of this paper is that the Quarc topology balances the traffic. The performance of the modified topology has been evaluated using extensive simulation experiments. The Quarc outperforms the Spidergon both in terms of latency and throughput over the complete range of number of nodes, message length and broadcast rate.

Our next objective is to compare the performance of the Quarc against other widely used NoC architectures such as mesh and torus.

References

[1] D.K. Panda et al. Multidestination Message Passing in Wormhole k-ary n-cube Networks with Base Routing Conformed Paths. *IEEE Transactions on Parallel and Distributed Systems*, 1995.

[2] Ju-Young Park. Construction of Optimal Multicast Trees Based on the Parameterized Communication Model. *Int'l Conf. on Parallel Processing*, 1996.

[3] Lu Zhonghai , Yin Bei , and A. Jantsch. Connection-oriented multicasting in wormhole-switched networks on chip. *IEEE Computer Society Annual Symposium on Emerging VLSI Technologies and Architectures*, 2006.

[4] William J. Dally and Charles L. Seitz . The torus routing chip. *Journal of Distributed Computing*, 1986.

[5] X. Lin , A.-H. Esfahanian , and A Burago. Adaptive Wormhole Routing in Hypercube Multicomputers. *Journal of Parallel and Distributed Computing*, pages 274–277, 1998.

[6] D. F. Robinson et al. Efficient multicast in all-port wormhole-routed hypercubes. *Journal of Parallel and Distributed Computing*, 1995.

[7] D. Pham et al. Overview of the architecture, circuit design, and physical implementation of a first-generation cell processor. *Solid-State Circuits, IEEE Journal of*, 2006.

[8] E. Bolotin, et. al. QoS architecture and design process for Networks-on- Chip. *Journal of Systems Arch*, 2004.

[9] E. Rijpkema, K. Goossens, and P. Wielage. Router Architecture for Networks on Silicon. *Progress , 2nd Workshop On Embedded Systems*, 2001.

[10] Hong Xu et al. Optimal software multicast in wormhole-routed multistage networks. *IEEE Transactions on Parallel and Distributed Systems*, 1997.

[11] J. Duato et al. *Interconnection networks: An Engineering Approach.* Morgan Kaufmann, 2003.

[12] J. Liu, L.-R. Zheng, and H. Tenhunen . Interconnect intellectual property for network-on-chip. *Journal of System Architectures*, 2003.

[13] D. James. 2004 - the year of 90-nm: a review of 90 nm devices. *Advanced Semiconductor Manufacturing Conf. and Workshop, 2005 IEEE/SEMI*, pages 72–76, April 2005.

[14] H. Jeschke. Chip size estimation for SOC design space exploration. *Journal of Systems Architecture*, 2007.

[15] K. Goossens, J. Dielissen, and A. Radulescu. Aethereal network on chip: concepts, architectures, and implementations. *IEEE, Design and Test of Computers*, pages 414–421, 2005.

[16] M. Coppola, R. Locatelli, G. Maruccia, L. Pieralisi, and A. Scandurra. Spidergon: a novel on-chip communication network. *Int'l Symposium on System-on-Chip*, 2004.

[17] M. Dall'Osso et al. xpipes: a Latency Insensitive Parameterized Network on-Chip Architecture for Multi-Processor SoCs. *Int'l Conf. on Computer Design*, 2003.

[18] M. Millberg, E. Nilsson, R. Thid, S. Kumar, and A. Jantsch. The nostrum backbone-a communication protocol stack for Networks on Chip. *Int'l Conf. on VLSI Design*, 2004.

[19] M. Moadeli et al. Communication Modeling of the Spidergon NoC with Virtual Channels. In *ICPP*, 2007.

[20] STMicroelectronics. www.st.com.

[21] A. Varga. Omnet++. *IEEE Network Interactive, in the column Software Tools for Networking*, 2002.

[22] W. J. Dally and B. Towles. Route packets, not wires: On-chip interconnection networks. *Design Automation Conf. (DAC)*, pages 683–689, 2001.