2009 International Conference on Advanced Information Networking and Applications

Quarc: a High-Efficiency Network on-Chip Architecture

Mahmoud Moadeli[1], Partha Maji[2], Wim Vanderbauwhede[1]
1: Department of Computing Science
University of Glasgow
Glasgow, UK
Email: {mahmoudm, wim}@dcs.gla.ac.uk
2 : Institute for System Level Integration
Livingston, UK
Email: partha.maji@sli-institute.ac.uk

Abstract

Th novel Quarc NoC architecture, inspired by the Spidergon scheme [5] is introduced as a NoC architecture that is highly efficient in performing collective communication operations including broadcast and multicast. The efficiency of the Quarc architecture is achieved through balancing the traffic which is the result of the modifications applied to the topology and the routing elements of the Spidergon NoC. This paper provides an ASIC implementation of both architectures using UMC's $0.13\mu m$ *CMOS technology and demonstrates an analysis and comparison of the cost and performance between the Quarc and the Spidergon NoCs.*

1 Introduction

The Network-on-Chip (NoC) is proposed as a scalable, structured, packet-switched, energy efficient and reliable communication medium to address the increasing communication demands of the future complex System-on-chip (SoC) . 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 is a formidable 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 through exploiting more parallelism.

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 [12], Æthereal [6], and Xpipes [11] are some examples of architectures used for on-chip networks. The Spidergon [5] and the Quarc [14] NoCs are two 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.

This paper presents an introduction to the Quarc and the Spidergon schemes along with their ASIC implementations using UMC's $0.13\mu m$ CMOS technology. Moreover, the paper compares two architectures from cost and performance points of view,

The rest of the paper is organized as follows. Section 2

1550-445X/09 $25.00 © 2009 IEEE
DOI 10.1109/AINA.2009.64

IEEE computer society

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 presents a comparison between the Quarc and the Spidergon schemes in terms of performance and cost. Finally, we make concluding remarks in Section 4.

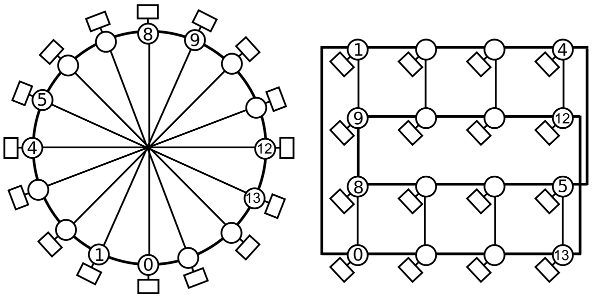


Figure 1. The Spidergon topology and the on chip layout.

2 Quarc: A NoC Architecture

The topology of an on-chip network specifies the structure in which routers connect the IPs together. Typically, a particular topology is chosen as a result of trading-off between performance and cost. Fat tree, mesh, torus and variations of rings are among the topologies introduced or adopted for the NoC domain.

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.

Due to similarity of the Spidergon and the Quarc NoCs, the next section presents a brief description of the Spidergon NoC, followed by an introduction to the Quarc NoC.

2.1 The Spidergon NoC

The Spidergon NoC [5] has been recently proposed by STMicroelectronics [15] 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.

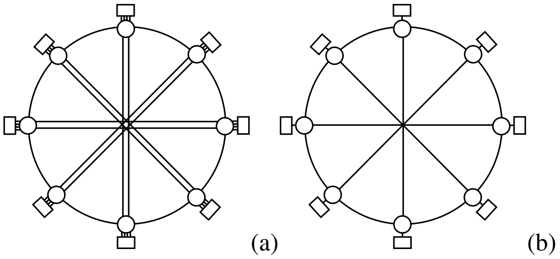


Figure 2. Quarc topology (a) vs Spidergon (b)

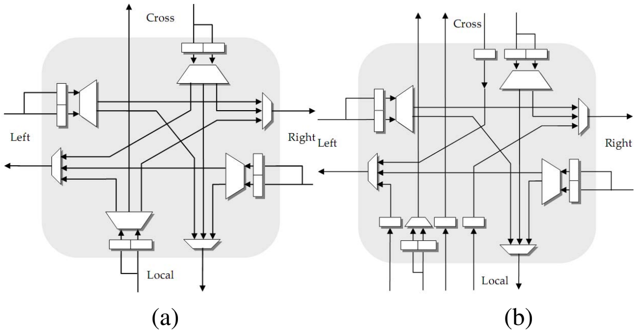
In the Spidergon NoC, two links connecting a node to surrounding neighboring nodes carry messages destined for half of nodes in the network, while the node is connected to the rest of the network via the cross link. Therefore, the cross link can become a bottleneck. Also, since the router at each node 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 Architecture

The Quarc NoC [14] shares significant similarities with Spidergon NoC. The Quarc preserves all features of the Spidergon architecture and improves on the Spidergon scheme 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.

Implementing the Quarc as an all-port router, significantly enhances the performance of the network by reducing the waiting time at source node. Moreover, adding another physical 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



(a) (b)

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

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 analysis in Section 3 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 present the switch architectures of the Quarc and the 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 2.4). 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.

A top level block diagram of the Quarc switch is shown in Fig.4. The Quarc switch architecture consists of three fundamental modules, namely, *Input Port Controller* (IPC), *Switch*, and *Output Port Controller* (OPC). While IPC contains input buffer to store the flits, OPC does not contain any output buffer. This significantly reduces overall area of the Quarc switch. Any flit enters to the Quarc switch pass through four stages, namely, input buffering, routing, switching, and virtual channel allocation. The different modules responsible for controlling each of these stages are shown in Fig.4. The routing logic inside the Quarc switch is very minimal as a flit can either be destined for local node or needs to be forwarded on the same direction on the rim. Hence, the area occupied by the crossbar is very small due to its simplicity.

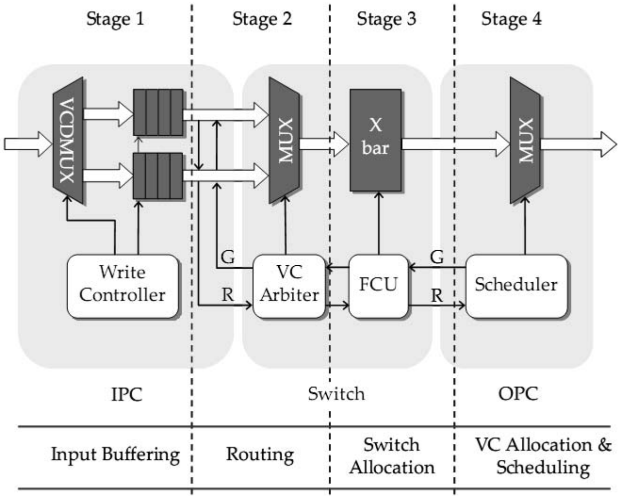


Figure 4. Functional block diagram of the Quarc Switch

2.4 Routing algorithm

2.4.1 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):

$$N_d > N_s \Rightarrow N_r = N_d - N_s$$
$$N_d < N_s \Rightarrow N_r = N_d - N_s + N$$

- Determine the quadrant q:

$$N_r \leq \frac{N}{4} \Rightarrow q = 0$$
$$\frac{N}{4} < N_r \leq \frac{N}{2} \Rightarrow q = 1$$
$$\frac{N}{2} < N_r \leq 3\frac{N}{4} \Rightarrow q = 2$$
$$N_r > 3\frac{N}{4} \Rightarrow q = 3$$

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.

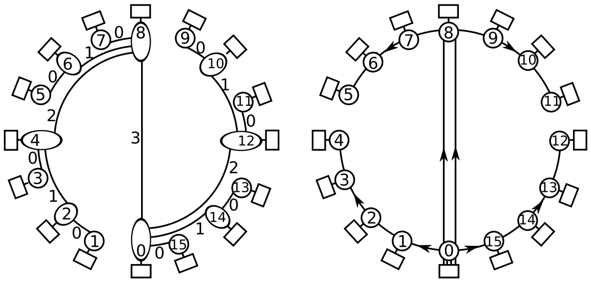
Quarc For the Quarc, the surprising observation is that there is no routing required by the switch: packets are either destined for the local port or forwarded to a single possible destination. Consequently, the proposed NoC switch requires no routing logic. The route is completely determined by the 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.

2.4.2 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 [8].

The support for collective communication may be implemented in *software* or/and *hardware*. The software-based approaches [7] 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. 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 [8]. After path selection, the intermediate destinations perform absorb-and-forward operations along the path. *Hamilton* path-based algorithm [4] and the *Base Routing Conformed Path* (BRCP) approach [1] are examples of path-based algorithms utilizing absorb-and-forward property at hardware layer.



(a) (b)

Figure 5. Broadcast in Spidergon (a) and Quarc (b) NoCs

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 [9]. 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 [10]. 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.

Broadcast is regarded as the most fundamental collective communication operation. Therefore, in the rest of this section the paper demonstrates how the Spidergon and the Quarc NoCs perform a broadcast communication.

Spidergon Broadcast in the Spidergon most efficiently may be handled by unicast with a “unicast tree” algorithm depicted in Fig. 5(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.

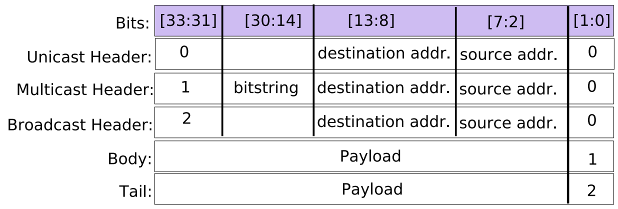
To perform a broadcast communication the transceiver of the initiating node has to broadcast packet on each port of the all-port router. The transceiver tags the header flit of each of four packets destined to serve each branch as broadcast to distinguish it from other types of traffic. The transceiver 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. 5(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.

2.5 Packet Format in the Quarc NoC

The Quarc scheme is a packet switched network employing wormhole switching. In wormhole switching a packet is divided into elementary units called flits, each composed of a few bytes for transmission and flow control. The header flit governs the route and the remaining data flits follow it in a pipelined fashion. If the header flit blocks, the remaining flits are blocked in situ.

Since the Quarc scheme adopts a simple deterministic routing, the packet format for unicast and collective communication is quite simple. For a Quarc NoC employing flit size of 34 bits various flit types composing a packet are depicted in Fig.6. Bits $[1:0]$ denote the flit types namely: *header*, *body* and *tail*. And the last 3 bits of header flits represent traffic types which are shown for *unicast*, *multicast* and *broadcast*. Each packet must have the header and tail flits.



Bits:	[33:31]	[30:14]	[13:8]	[7:2]	[1:0]
Unicast Header:	0		destination addr.	source addr.	0
Multicast Header:	1	bitstring	destination addr.	source addr.	0
Broadcast Header:	2		destination addr.	source addr.	0
Body:	Payload				1
Tail:	Payload				2

Figure 6. Flit type formats in the Quarc NoC

In broadcast/multicast operations, the last node to be visited must be specified as destination address in the header flit. For broadcast all nodes in the path from source to destination are the receiver nodes. While, in case of multicast the target addresses are specified in the *bitstring* field. Each bit in the *bitstring* represents a node which its hop-distance from the source node corresponds to position of the bit in the *bitstring*. Status of each bit indicates whether the visited node is a target of the multicast or not. Please note that due to the scalability issues of the Quarc NoC, it is assumed that the network size may be up to 64 nodes. However, larger networks may employ flits of larger size or to use multi flit headers for specifying multi-addresses for multicast operations.

3 Cost and Performance Analysis

Employing a particular NoC architecture typically involves trading-off between a number of cross-cutting measures such as performance and cost. This section presents a comparison of the cost and performance between the Quarc and the Spidergon NoCs.

3.1 Cost Analysis

In this section, we argue that the Quarc switch is smaller in size and at the same time is less complex than the Spidergon switch and this saving in area outweighs the overheads incurred by additional ports and the area for additional links.

SWITCH MODULES	AREA(mm^2)
Input Buffers	0.0397
Write Controller & VCDMUX	0.0021
Crossbar & MUX	0.0026
VC Arbiter	0.0026
Flow Control Unit	0.0018
OPC (Scheduler & MUX)	0.0143
Total Area of a 32-Bits Quarc Switch	**0.0631**

Table 1. Module-wise cost analysis of a 32-bits Quarc switch

We assume that every node of NoC hosts a processing element (PE), typically a microprocessor with local memory. The difference in resource utilization at the PE between the Quarc and the Spidergon NoCs is very small. In both cases the packets are stored in RAM and the address of the packets are queued. For the Quarc NoC, the PE queues the addresses in four separate queues, effectively making the routing decision by doing so. For the Spidergon NoC, 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 has to be twice as deep. This is, of course, a small memory overhead as the address size is a fraction of the packet size. Also, note that the actual packet memory requirements are identical for both the Quarc and the Spidergon NoCs.

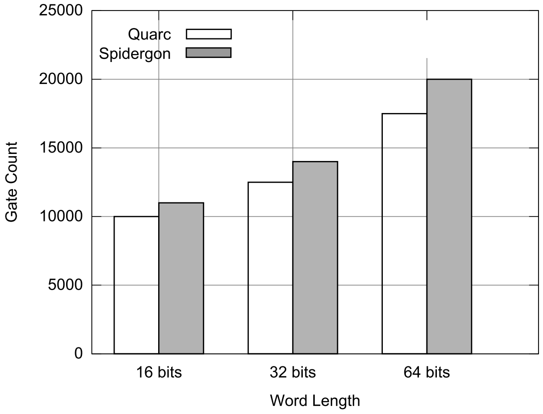


Figure 10. Cost comparison between Quarc and Spidergon switches

In terms of costs, the key differences between the Quarc and the Spidergon switches are the local ingress ports, crossbar, and routing logic. The four ingress ports of the Quarc translate to four addresses on the processor bus, instead of a single address for the Spidergon NoC. Each of the three extra local ports in the Quarc switch requires one flit buffer, but only a single buffer (i.e. no separate buffers per virtual channel) as there is only one destination. The other cross port demultiplexer of the Quarc NoC requires a flit buffer per virtual channel which is the same as the Spidergon NoC's local port. This extra area overhead of input buffers in the Quarc switch is compensated by the area of crossbar and routing logic required for the Spidergon NoC.

The Spidergon switch needs to calculate the output port based on the flit header. In comparison, the Quarc only needs to compare the destination address with the switch address to decide if the packet needs to be delivered locally or to be forwarded. Thus the routing infrastructure in the Quarc switch is almost non-existent, which reduces the complexity and area of the switch. Furthermore, in the Spidergon switch, the local and the cross ports require crossbar of dimension 2×3 which occupies a large area. But in comparison, the Quarc switch does not require any crossbar at all at the local port. The cross port of the Quarc NoC requires a crossbar of dimension of 2×2 which is similar to other input ports. This saving of area due to crossbar and routing logic in the Quarc switch considerably outweighs the overhead due to the additional local ports.

To present a comparison between the two architectures, we have implemented 16, 32, and 64-bits versions of both the Quarc and the Spidergon switches using UMC's $0.13\mu m$

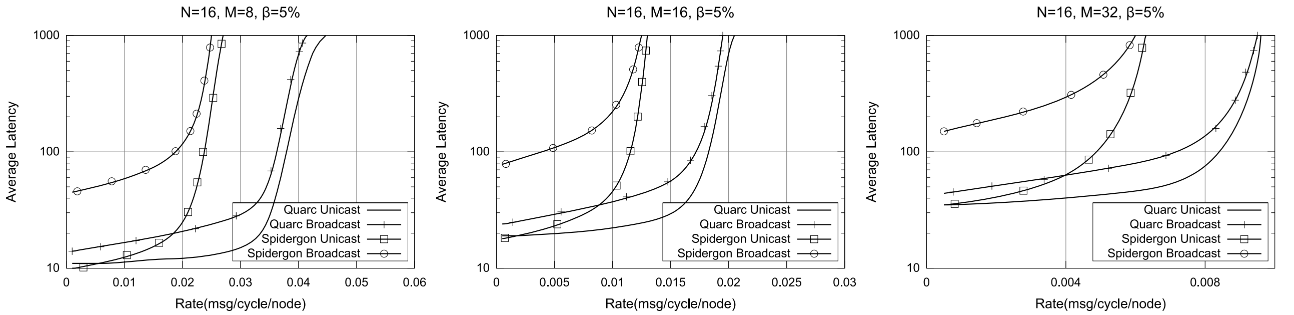


Figure 7. Comparison of Quarc and Spidergon for M=8,16,32

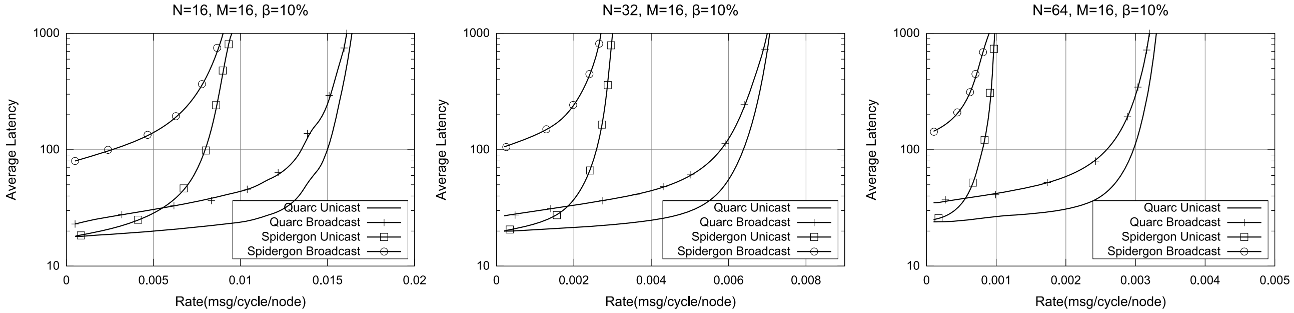


Figure 8. Comparison of Quarc and Spidergon for N=16,32,64

CMOS technology library. In order to make assembling and upgrading of the switch simple, the switch architecture is designed in a modular fashion as shown in Fig.4.

For 32-bits version of a Quarc switch the pre-layout area is $0.063mm^2$, whereas similar version of the Spidergon switch occupies $0.071\ mm^2$. A more detailed module-wise area occupancy for a Quarc switch of 32-bits version is shown in the Table1. Note that the amount of area occupied by the crossbar and FCU are very minimal. This result supports the argument that the Quarc NoC does not have complex crossbar or routing logic, which saves the area of the switch. The gate density of UMC's $0.13\mu m$ CMOS process is up to $200K$ gates/mm^2. From the obtained synthesis results using this technology library, we have calculated the gate count for various configurations of the Quarc and the Spidergon switches. A comparison of the cost analysis in terms of gate count for various versions between the two switches is shown in the Fig.10.

3.2 Performance Analysis

To evaluate the performance of the Quarc NoC architecture we have developed a discrete event simulator operating at flit level using OMNET++ [16]. The simulator has been verified extensively against analytical models for the Spidergon and mesh topologies employing wormhole routing [13].

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. 7 shows the average latency experienced by unicast and broadcast traffic in the Quarc and the 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. 8 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.

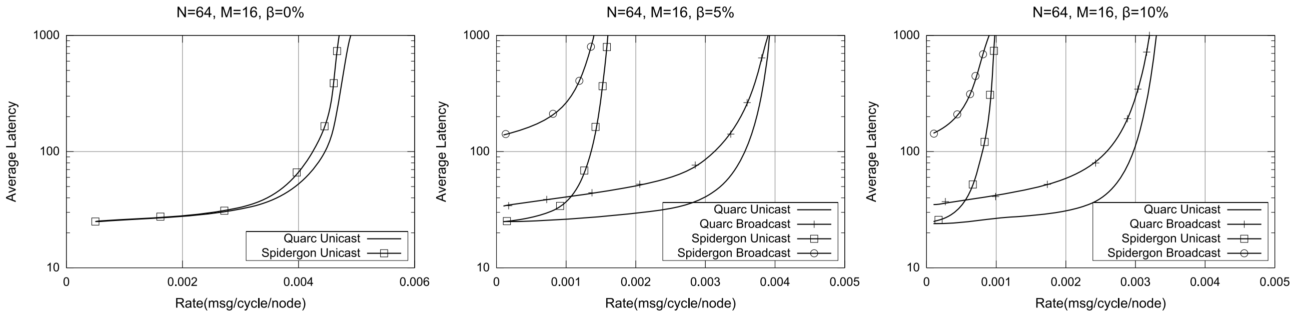


Figure 9. Comparison of Quarc and Spidergon for β= 0%, 5%, 10%

The graphs in Fig. 9 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, β, 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.

4 Conclusion

In this paper we have presented an ASIC implementation of the Quarc NoC. 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 performance of the Quarc NoC has been evaluated using extensive simulation experiments. The Quarc outperforms the Spidergon over the complete range of number of nodes, message length and broadcast rate. Equally important, our cost analysis based on ASIC implementation of the two architectures showed that, surprisingly, the additional performance gain obtained at no extra cost compared to the Spidergon NoC.

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

[5] M. Coppola, M. D. Grammatikakis, R. Locatelli, G. Maruccia, and L. Pieralisi. *Design of Cost-Efficient Interconnect Processing Units: Spidergon STNoC*. CRC Press, Inc., Boca Raton, FL, USA, 2008.

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

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

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

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

[10] 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.

[11] 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.

[12] 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.

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

[14] M. Moadeli, W. Vanderbauwhede, and A. Shahrabi. Quarc: A Novel Network 0n-Chip Architecture. *Parallel and Distributed Systems, International Conference on*, 2008.

[15] STMicroelectronics. www.st.com.

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