Characterization of Costs and Performance of Communication Mechanisms for a Network-on-Chip

Rubens Vicente de Liz Bomer, Cesar Albenes Zeferino
University of Vale do Itajaí - UNIVALI
Embedded and distributed systems laboratory-LSED
Itajaí, Brazil
{rubensliz, zeferino}@univali.br

***Abstract*— A Network-on-Chip (NoC) applies a set of mechanisms to enable communication among the cores that attached to it in a NoC-based system. These communication mechanisms include routing, flow-control, buffering, arbitration and switching. For each mechanism, the design space of NoC offers alternative techniques with impact on the costs and performance of the NoC. This paper presents a study that aimed at identifying the costs and performance of different techniques for the communication mechanisms applied in a Network-on-Chip. A parameterizable and synthesizable model of a NoC was used to obtain the silicon costs in a 90 nm ASIC technology, while a simulation model was applied to obtain the performance metrics. Results identify the less expensive techniques and compare the performance metrics of alternative techniques for some communication mechanisms.**

Keywords—Network-on-Chip; Logical Synthesis; Performance.

I. INTRODUCTION

The problem of interconnecting components in a System-on-Chip (SoC) has received an increasing attention from research and development departments of universities and industries. This is due to the fact that future SoCs with tens to hundreds of cores will require communication architectures that offer scalable bandwidth (which increases with the size of the system) and, also, are easily reused in order to minimize the time necessary for designing new systems. The traditional bus-based architectures satisfy the reusability requirement, but do not offer scalable performance. So, new solutions have been discussed in the literature.

An example of the NoC is SoCIN (System-on-Chip Interconnection Network) [1], which was developed by researchers of the University of Vale do Itajaí and Federal University of Rio Grande do Sul. This network uses low-cost solutions to provide scalable and reusable architecture for SoCs, providing basic services of the three lower layers of the OSI model (Open Systems Interconnection): Physical, Data Link and Network layers.

SoCIN has already been synthesized to FPGA (Field Programmable Gate Array), but this technology is not suitable for a cost analysis due to the extra area used to enable its configurability. Furthermore, the only performance metric that can be obtained with the FPGA synthesis tools is the maximum operating frequency of the target circuit.

In order to overcome these limitation, in this work, a set of experiments were carried out in order to obtain the costs of the communication mechanisms used in SoCIN in an ASIC (Application Specific Integrated Circuit) technology, which does not have the extra cost of FPGAs. Furthermore, in order to improve the performance analysis, this work applied a cycle-accurate SystemC RTL simulator [2] of SoCIN (which is named BrownPepper) to assess the effect of each alternative mechanism in the performance of communication.

Our goal was to characterize the cost and performance of communication mechanisms used SoCIN, identify the mechanisms with smaller costs and better performance, and determine the configuration with best relationship between cost and performance.

The remain of this paper is organized as follows. In Section II, the communication mechanisms used in SoCIN are described. Section III presents the methodology used in this work to obtain the results, which are presented and discussed in Section IV (cost) and Section V (performance). Section VI presents the final remarks.

II. COMMUNICATION MECHANISMS

SoCIN is a NoC that uses a parameterizable router (named ParIS – Parameterizable Interconnect Switch) in order to allow the use of different techniques or implementations for each one of the following communication mechanisms: routing, arbitration, flow control and buffering.

The routing techniques supported are: XY (deterministic), in which the packet must first move on the X-axis and then on the Y-axis, and WF (partially adaptive), in which the packet must first move in the X-direction (West-First) if its destination is at West. If the destination is at East, it can be forwarded in the X-direction as in the Y direction, adaptively.

Three alternatives are available for the arbitration: Static, Random and Round-Robin. The Static arbiter ensures a constant priority level for a particular request. The Random arbiter uses a ring counter and gives the higher priority to a different request every clock cycle. Finally, the Round-Robin arbiter changes the priority criteria at each arbitration and ensures a fair distribution of the scheduled resource [3].

Flow control can be of two types: Handshake or Credit-based. In the first one, the sender sends a validation signal and

waits for an acknowledgment from the receiver indicating that it is able to receive the data. In the Credit-based technique, the sender keeps a counter of credits representing the free space of the receiver buffer. For each data it sends, the counter is decremented until it reaches zero, when the sender is no more allowed to send data. For each data consumed by the receiver, is sent a credit to the sender, which increments the counter.

Buffering is based on FIFO buffers and two implementations are available: Shift and Ring (or Circular). In the first approach, a data is always written in first FIFO position and the data previously written into the buffer are shifted right. In other words, the write pointer is fixed and the read pointer varies. In the second approach, both pointers are variable and there is no data shifting inside the buffer.

III. METHODOLOGY

In order to obtain the silicon costs, Synopsys Design Compiler tool and SAED 90nm technology were used. All the synthesis were done using a 100 MHz clock with typical operation conditions. The maximum operating frequency was calculated on the basis of the slack in the critical path. Power results were automatically obtained by the synthesis tool (none switching activity file was used).

Additionally, we used the BrownPepper simulator [3] in order to compare the performance of the different communication mechanisms. Experiments were based on a series of simulations varying the operating frequency from 1 to 3000 MHz at intervals of 50 MHz using a traffic pattern based on the on described in [4] (more details in the Section V).

IV. SILICON COSTS

In order to compare the costs of different alternatives for the communication mechanisms, firstly, a reference architecture was defined (Table I) and its costs (area and power) were determined (Table II).

TABLE I. REFERENCE ARCHITECTURE

Parameter	Value
Channel Width	32 bits
Input FIFO depth	4 Flits
Output FIFO depth	0 Flits
FIFO Type	Ring
Flow control	Handshake
Routing	XY
Arbitration	Round Robin

TABLE II. SILICON COSTS OF THE REFERENCE ARCHITECTURE

Metric	Value
Total Area (μm2)	48598,96
Total Power (μW)	895,25

Then, one parameter (arbitration, routing, flow control, and FIFO implementation) was changed at a time in order to compare its costs with the one of the reference architecture and thus determine the alternative that occupies a smaller silicon area and dissipates less power.

Table III presents the results obtained with the reference architecture using Ring and Shift implementations for the FIFO buffers. Due to the use of two variable pointers, the Ring implementation occupies a larger area. On the other hand, the Shift implementation dissipates more power because of the internal shifting at each writing operation. The maximum operating frequency was the same for both implementations.

TABLE III. BUFFERING: RING X SHIFT

Metric	RING	SHIFT
Total Área (μm2)	48598,96	47905,37
Total Power (μW)	895,25	1140,60

Table IV compares the costs of using the two flow control techniques. Due to use of a counter, the Credit-based approach is the most expensive, occupying a larger silicon area and dissipating more power.

TABLE IV. FLOW CONTROL: HANDSHAKE X CREDIT-BASED

Metric	Handshake	Credit -based
Total Área (μm2)	48598,96	49061,28
Total Power (μW)	895,25	1169,50

Table V compares the two routing algorithms and shows that the XY routing occupies a smaller area than the WF approach, because it uses a simpler circuit. However it dissipates a bit more power.

TABLE V. ROUTING: XY X WF

Metric	XY	West First
Total Área (μm2)	48598,96	48848,16
Total Power (μW)	895,25	885,36

Table VI compares the three alternatives for the router arbiter. The Round-Robin occupies the larger area because it uses a bit more complex circuit to update its priority criteria. The Random arbiter dissipates more power because it updates its priority criteria at each clock cycle, while the Round-Robin does it only at each arbitration. The Static arbiter dissipates less power because it does not update its priority criteria.

TABLE VI. ARBITRATION: ROUND-ROBIN X STATIC X RANDOM

Metric	Round Robin	Static	Random
Total Area (μm2)	48598,96	46715,41	48439,17
Total Power (μW)	895,25	886,97	923,03

Based on the performed experiments, the router architecture that occupies less area is shown in Table VII. As it is shown in Table VIII, this configuration is 5,3% smaller than the reference architecture, but dissipates 26,4% more power.

TABLE VII. ARCHITECTURE WITH THE SMALLEST AREA

Parameter	Value
Channel Width	32 bits
Input FIFO Depth	4 Flits
Output FIFO Depth	0 Flits
FIFO Type	Shift
Flow Control	Handshake
Routing	XY
Arbitration	Static

TABLE VIII. COSTS OF THE ARCHITECTURE WITH THE SMALLEST AREA

Metric	Value
Total Area (μm2)	46004,58
Total Power (μW)	1132,40

On the other hand, the configuration that dissipates less power is shown in Table IX. It dissipates 2% less power than the reference architecture and, at the same time, is 3,4% smaller, as one can see in Table X.

TABLE IX. Architecture dissipating less power

Parameter	Value
Channel Width	32 bits
Input FIFO depth	4 Flits
Output FIFO depth	0 Flits
FIFO Type	Ring
Flow Control	Handshake
Routing	WF
Arbitration	Static

TABLE X. COSTS OF THE ARCHITECTURE DISSIPATING LESS POWER

Metric	Value
Total Area (μm2)	46964,61
Total Power (μW)	876,76

V. PERFORMANCE EVALUATION

BrownPepper [3] simulator was used to obtain performance metrics of a 4x4 2D mesh with 32-bit wide data channels.

BrownPepper is a tool that simulates a SoCIN network and allows its evaluation under different configurations and traffic patterns. The NoC is described in cycle-accurate SystemC at the register-transfer level (RTL). A traffic generator and a traffic meter, also modeled in SystemC, are used to inject and collect packets, respectively. In BrownPepper, packets must have at least 4 flits of payload, which are used to carry the information necessary for performance evaluation.

The traffic model used in the experiments is based on the one applied in [4] and is described in Table XI. Four traffic classes are defined with different temporal distributions for traffic injection. Spatial distribution was based on a non-uniform approach in which the probability of sending packets to a given destination reduces with the distance to the source node. This approach is more realistic, because, in general, designers try to place closer the nodes that communicate more frequently.

TABLE XI. TRAFFIC MODEL

Traffic class	Payload length (flits)	Interval between packets	Required bandwidth	Deadline	Packets per flow
Signaling	4	400 ns	320 Mbps	40 ns	125
Real-Time	20	2000 ns	320 Mbps	125 μs	25
RD/WR	4	50 ns	2,56 Gbps	300 ns	990
Block-Transfer	1000	12500 ns	2,56 Gbps	50 μs	4

Fig. 1 and Fig. 2 compare the average latency and the accepted traffic of the routing algorithms. One can observe that the XY routing accepts more traffic than the WF because it reaches the saturation point later. It also presents a better jitter at lower frequencies than the routing WF (Fig. 3).

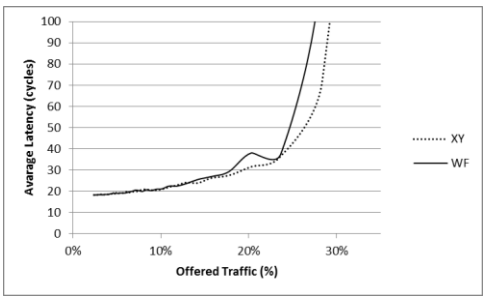

Figure 1 Flow control – Average Latency x Offered Traffic

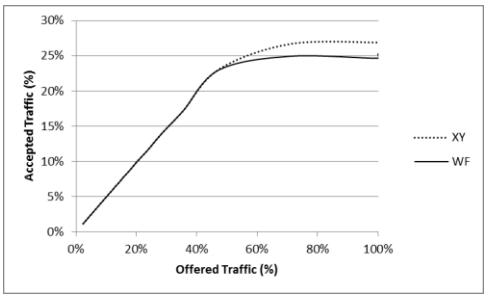

Figure 2 Fow Control – Accepted Traffic x Offered Traffic

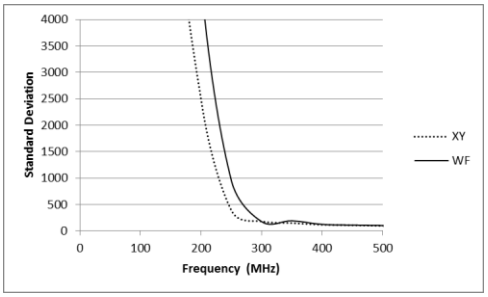

Figure 3 Flow Control – Standard Deviation x Frequency

Fig. 4 and Fig. 5 compare the latency and the accepted traffic of the three arbiters. The Static arbiter has a higher average latency and accepts less traffic. There was no great difference between the arbiters with respect to jitter as is visible by the curve shown in Fig. 6.

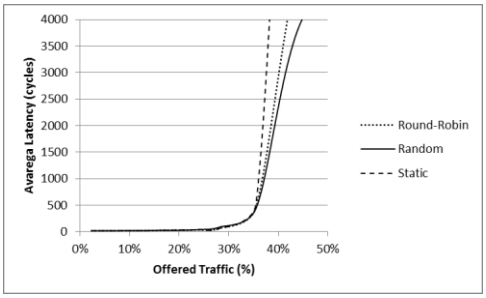

Figure 4 Arbiter – Avarage Latency x Offered Traffic

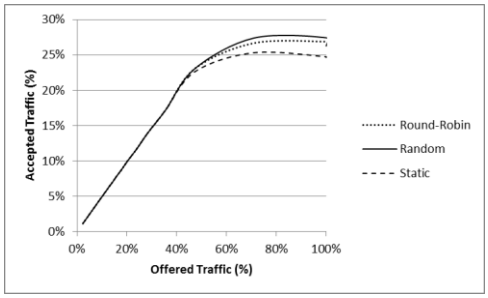

Figure 5 Arbiter – Accepted Traffic x Offered Traffic

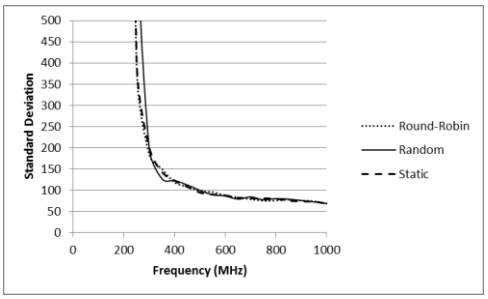

Figure 6 Arbiter – Standard Deviation x Frequency

VI. CONCLUSION

Based on experiments conducted in this work it was possible to identify the alternatives of communication mechanisms with the best cost and performance metrics.

Logical synthesis in ASIC technology made possible to identify that the Shift FIFO type and XY routing occupy less silicon area, while the FIFO Ring and WF routing dissipate less power. Handshake flow control and the Static arbiter had the lowest cost on both metrics.

Regarding performance, XY routing presented better results than WF in all metrics. For the arbiters, the Static presented the worst results in latency and throughput, and the Random presented the better results. There was no great difference among them in relation to the jitter metric.

As future work we intend to use the switching activity files evaluate the power dissipation with more accuracy, as well as implementing new alternative communication mechanisms in SystemC and VHDL.

ACKNOWLEDGMENT

This research was funded by UNIVALI (University of Vale do Itajaí) and INCT NAMITEC (National Institute for Science and Technology on Micro and Nanoelectronics Systems).

REFERENCES

[1] A. A. Susin; C. A. Zeferino; F. G. M. E. Santo. ParIS: A Parameterizable Interconnect Switch for Networks-on-Chip. In: sbcci symposium on Integrated Circuits and System Design, 17., 2004, Pernambuco. Annals… New York: ACM, 2004. p. 204-209.

[2] C. A. Zeferino; J. V. Bruch; M. R. Pizzoni. BrownPepper: a SystemC-based Simulator for Performance Evaluation of Networks-on-Chip. In: 17th ifip/ieee int. Conference on Very Large Scale Integration – vlsi-soc, 2009. Proceedings… Florianópolis: [S.L.], p. 1 – 4.

[3] A. A. Susin; C. A. Zeferino; F. G. M. E. Santo. Modelos Parametrizáveis de Árbitros Centralizados para a Síntese de Redes-em-Chip. Hífen (Uruguaiana), Uruguaiana, p. 91-96, 2003. (*in portuguese*)

[4] A. Kolodny; I. Cidon; R. Ginosar; E. Bolotin. QnoC: QoS architecture and design process for network on chip. In: Journal of Systems Architecture, 50, 2004, p. 105-128.