A CORE FOR NETWORK-ON-CHIP LATENCY-BASED PERFORMANCE ANALYSIS

Miklécio Bezerra da Costa, Ivan Saraiva Silva

Departamento de Informática e Matemática Aplicada
Universidade Federal do Rio Grande do Norte, Natal, Brasil
miklecio@lasic.ufrn.br, ivan@dimap.ufrn.br

ABSTRACT

This paper proposes a traffic generator core for performance analysis of a NoC, based on the latency observed by copies of this core coupled on the network's routers. The core was implemented in VHDL, integrated with the NoC SoCIN and with some Altera's components, like Avalon bus, Nios II/e processor and JTAG UART I/O component, synthesized and prototyped on an Altera's FPGA chip. An application implemented in the language C configured the cores in order to obey the traffic pattern Complement. This application collected the sums of latencies measured by the cores on the FPGA and returned results of average latencies from the communication between SoCIN's nodes. The results showed that this core's project is on the right way for a consistent and complete NoC performance analysis.

1. INTRODUCTION

The increase of transistors' integration on a chip capability allowed the development of MP-SoCs (Multi-Processor System-on-Chip). MP-SoCs have been used to solve the problem of the growing complexity on applications, because the traditional mono-processor systems present a theoretical physical limit in their processing capability [1].

Based on that, the concept of NoC (Network-on-Chip) came up, a mechanism of modules' interconnection on MP-SoC. NoC is considered more adequate to the purposes of a MP-SoC when compared to mechanisms like dedicated point-to-point hires, shared bus or bus hierarchy, because of characteristics like high scalability and reusability [2].

Many NoC projects have been developed, trying to attend this growing demand. These projects differ from each other because of the large number of possible NoC implementations. NoCs may present different topologies and routing, arbitration, switching, memorization and flow control algorithms.

Some performance measures are used to evaluate the quality of these NoC projects in order to decide which one attends the requisites of a MP-SoC. The most important measures to evaluate an interconnection network, like a NoC, are latency and throughput [3].

This paper proposes a core for latency-based performance analysis of NoC. The proposal consists in traffic generator cores that measure the latency proportioned by this traffic on the NoC and send the results to an Altera's system formed by a Nios II processor [4] and a memory, all interconnected by an Avalon bus [5]. These traffic generator cores can be coupled on the routers of a NoC in order to evaluate it.

This paper is divided in the following sections: section 2 presents a general view about NoC; section 3 presents the concept of NoC performance analysis and the measures involved in this process; section 4 describes the project of the core for NoC latency-based performance analysis, proposed in this paper; section 5 presents the results obtained by the use of this core on NoC SoCIN; and finally, section 6 presents the conclusion.

2. NETWORK-ON-CHIP (NOC)

A Network-on-Chip is formed by a set of routers and point-to-point links that interconnect the cores of an integrated system [6]. It's a concept inherited from the traditional computer interconnection networks. The main characteristics of a NoC are also inherited: topology, routing, arbitration, switching, memorization and flow control. The next paragraphs show a brief definition of these characteristics.

The topology of a NoC indicates how the routers are spatially interconnected. It may be classified in direct or indirect topology [7]. In the direct one, each router is connected on a core, for example: 2-D grid, 3-D cube and hypercube. And in the indirect one, some routers just connect other routers, without coupled cores, for example: crossbar and multistage network [8].

The pathway followed by the packet depends on the routing strategy. The main routing algorithms are the deterministic, which always follows the same pathway for a pair of routers, like XY, and the adaptive one, which may change the pathway according to the network situation, like West-First and North-Last [9].

The arbitration mechanism controls the access to output ports in the router. The arbitration may be centralized, with a central arbiter deciding the disputes, or distributed, with arbiters on each output port of the router.

The switching mechanism decides how the packets are transmitted on the link. The main switching mechanisms are store-and-forward, virtual-cut-through and wormhole.

The memorization strategy is important to store data

when a dispute for access occurs. It may be used on the input or on the output port of the router, implemented by structures like FIFO (First In First Out), SAFC (Statically Allocated Fully Connected), SAMQ (Statically Allocated Multi-Queue) or DAMQ (Dinamically Allocated Multi-Queue).

The flow control is a protocol of synchronism between the transmission and the reception of data [3]. The main types of flow control are credit-based, on/off and ack/nack [10].

3. NOC PERFORMANCE ANALYSIS

The performance analysis of an interconnection network, like a NoC, is based on two main measures: the latency and the throughput.

The latency is the interval of time between the start of the message transmission and the reception of it on the target node [11]. However this is a vague concept [3]. It may be the time between the injection of the packet's header into the router and the reception of its last information or it may count the time just on the routers, not on the nodes. The latency is measured in cycles or another time unit.

The throughput is the quantity of information transmitted on the network per time unit [10]. It may be measured in number of bits transmitted on each pair source-target or in percentage of this link's utilization. In this last case, a communication link works on its maximum capability if it is used 100% of the time [3].

In the recent years, some works have been done to evaluate NoCs performance [12, 13]. They combine bit level accuracy and flexibility, implementing on FPGA (Field Programmable Gate Arrays) many kinds of NoCs, traffics and performance measures.

4. CORE FOR NOC LATENCY-BASED PERFORMANCE ANALYSIS

The proposal of this paper consists in presenting traffic generator cores coupled on the local interface of each router of the NoC and an Altera's system connected on one of these cores. Each core generates packets that are received by another core. So the latency of these packets transference on the NoC is calculated and sent to the Altera's system.

One of these cores also communicates with an Avalon bus, receiving configurations and sending results. This core implements a wrapper mechanism between the protocols of Avalon bus and the NoC.

A Nios II/e processor and a 16-bit memory with 512 KB of storage capability compose the Altera's system interconnected by the Avalon bus. That allows to create customized applications in order to analyze the NoC performance. Figure 1 shows the components of an example of the core utilization.

The core is formed by 2 main modules: the sender and the receiver. Both implement handshake flow control algorithm. There is also a local timer, which counts the time in cycles and is synchronized with the other cores' timers. Figure 2 shows the sub-components of the core and the one with wrapper mechanism.

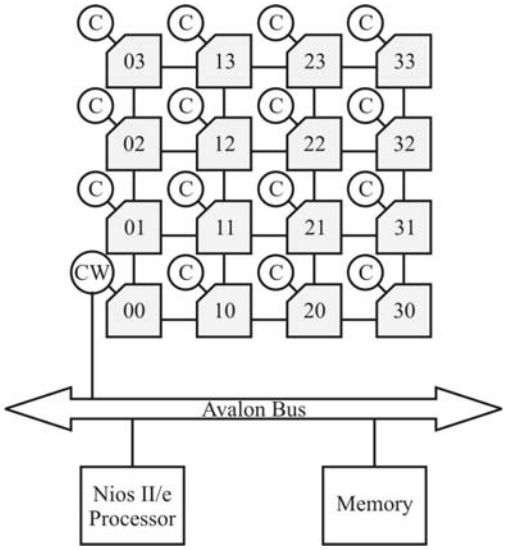

Figure 1 – The components of the core utilization for latency-based performance analysis of a NoC 4x4. **C** represents the traffic generator core and **CW** represents the core that implements the wrapper mechanism.

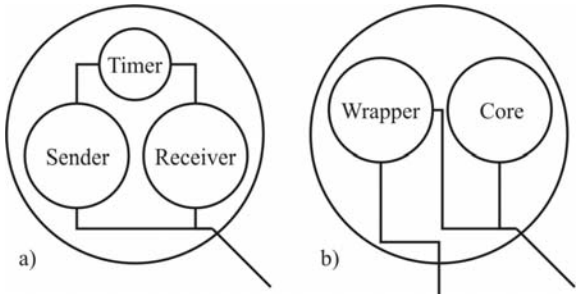

Figure 2 – **a)** The sub-components of the traffic generator core; **b)** The core with a wrapper module between the NoC and the Avalon bus.

First, the application on the Nios II processor sends the configurations to the cores through the Avalon bus. Each core receives 2 configuration packets: one to the receiver and one to the sender. The receiver's configuration packet has the number of packets that the receiver shall collect to generate each result packet. And the sender's one has the target receiver address, the size of the data packets (number of data words) that the sender shall generate and the time delay in cycles between two packets.

Next, the sender starts to inject data packets into the NoC destined to determined receiver. These data packets of determined size contain the sending time, registered by the local timer. Then the receiver collects these packets and calculates their latencies also using the timer's value. After collecting determined number of packets, the receiver sends a result packet to the core with wrapper. This result packet contains the sum of latencies accumulated and the number of packets received. These 4 types of packet – receiver configuration, sender configuration, data and result – used by the cores are formatted according to parameters inherited from the

NoC's packet format, like the widths of words, control bits and addresses (an example is presented in section 5).

The application on the Nios II processor receives interruptions in order to read the result packets through the Avalon bus again. Depending on the application, the analysis may be the average latency between two nodes of the NoC. So the application may divide the sum of latencies by the number of packets.

The memory is used by the application and it maps the components of the Altera's system, which supports a set of I/O components. The Nios II processor is the only one that implements an Avalon master interface. The wrapper module and the memory implement an Avalon slave interface each one.

5. RESULTS

The core was applied on the NoC SoCIN, a very well recognized and documented interconnection network that implements the router ParIS [14], to obtain a valid evaluation of NoC performance. The NoC SoCIN has 2-D grid (or mesh) topology and wormhole packet-based switching. Other aspects are parameterizable. In this case, a NoC of dimensions 4x4 was configured to use deterministic (XY) routing, handshake flow control, dynamic (round roubin) arbitration and FIFO memorization on the input ports.

Following the SoCIN packet format, the 4 types of packet used during this performance analysis are listed in figure 3.

These packets are formed by 32-bit words. SoCIN reserves the msb (most significant bit) of each word to indicate the end of the packet, the second msb to indicate the start of it (the header) and the 4 lsbs (least significant bits) of the header to address the target node on the NoC, which could be a receiver address (RA), a sender address (SA) or the core with wrapper (address "0000").

Every header of these packets has two bits to identify the type of the packet. The receiver configuration packet has a 30-bit field with the number of packets (NoP) received per result. The sender configuration packet has a 3-bit field with the identification of the traffic pattern (TP) implemented in the core's hardware. No traffic pattern has been implemented in the core yet. Therefore this packet uses a 4-bit field with the target receiver address (RA) to indicate manually which traffic shall be generated. There are also the 30-bit fields SoP (size of packet) and Delay.

The data packet also indicates the type of traffic generated, using the same fields TP and SA of the sender configuration packet. The quantity of data words (which are not header or end of packet) is defined by the field SoP and the first one of these words is always the sending time (ST), which means the cycle that the data packet begins to be injected into the NoC. The result packet also indicates the type of traffic generated, using the same fields TP, RA and SA of the data packet. It has two 30-bit fields in order to return the sum of latencies (SoL) and the number of packets (NoP).

The implementations in VHDL of the core and of the NoC SoCIN were integrated and synthesized in the Altera Quartus II IDE. The complete system was prototyped on an Altera DE2 FPGA, the device EP2C35F672C6 of the family Cyclone II, using the Altera SOPC Builder IDE. The maximum operating frequency of the system was 96.26 MHz. Table 1 lists the silicon results obtained from the prototyping.

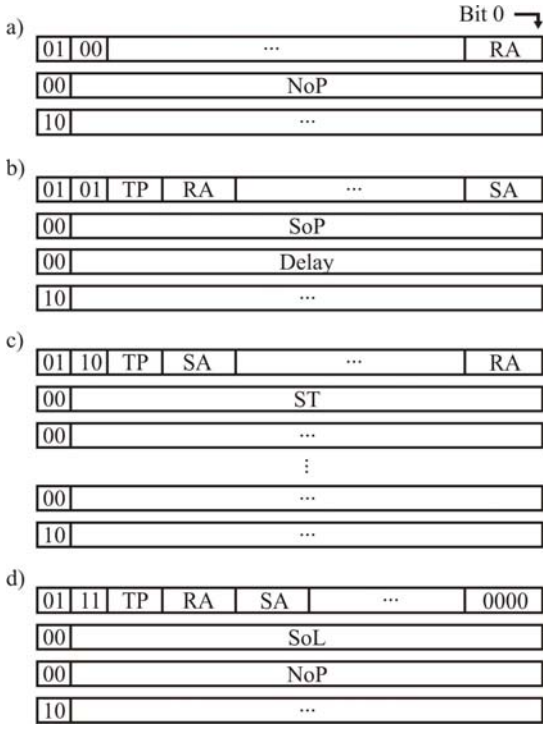

Figure 3 – The 4 types of packet used by the core on the NoC SoCIN: **a)** The receiver configuration packet; **b)** The sender configuration packet; **c)** The data packet; **d)** And the result packet.

Component	Logic Cells
Core	718
Core with wrapper	1193
Router with 3 ports	577
Router with 4 ports	900
Router with 5 ports	1150
Nios II/e processor	964
Whole system	**25072**

Table 1 – Silicon costs for each component of the prototyped system.

An application in language C was implemented to analyze the latencies of the traffic pattern Complement. The application configures the senders to generate packets to their respective complements. The Altera Nios II IDE compiled and injected the application into the system through the JTAG UART I/O component. This component was also used to obtain the average latencies calculated by the application.

The average latencies, calculated by the application on the Nios II, for the sums of latencies sent by each core, using the traffic pattern Complement and applied to the NoC SoCIN, are represented in the graphics of figures 4 and 5.

These graphics show results of packets with respectively 8 and 1024 words, keeping 8 cycles of delay between each data packet sent and a number of 1000 packets accumulated per each result. The graphics contain values of the 50th result obtained from each core, to guarantee that all senders are injecting data packets.

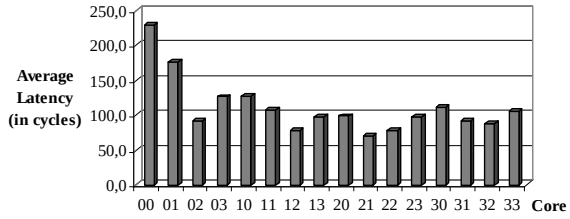

Figure 4 – The latency-based performance analysis of the NoC SoCIN using the traffic pattern Complement. The cores were configured to generate packets of size 8.

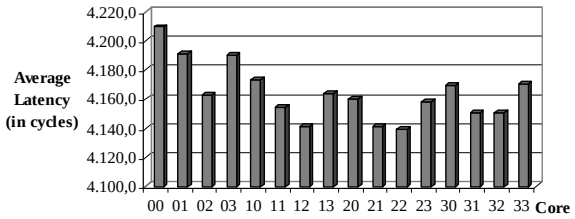

Figure 5 – The same analysis of figure 4, but, in this case, the cores were configured to generate packets of size 1024.

In both cases, the results show that cores near the center of the NoC, like 12, 21 and 22, had lower latencies, explained by the proximity between the sender and the receiver in the pattern Complement for these nodes [11]. But the latencies measured on cores near the one with wrapper, like on 01 and obviously on 00, were overvalued, because they also count the latencies generated by the result packets sent to the Avalon bus through the core 00.

6. CONCLUSION

In this paper it was proposed a core for performance analysis of a NoC, based on the latency observed by copies of this core coupled on the network's routers. The core was implemented in VHDL, integrated with other components, synthesized and prototyped on an Altera's FPGA chip.

The core can be applied on any NoC that implements the handshake flow control, changing just a few parameters of the core's packets formats, for instance: the widths of the words and addresses. In the context of this work, it was analyzed the NoC SoCIN, a very well recognized and documented interconnection network. It was also implemented an application that configured the cores, based on the traffic pattern Complement, and collected its results.

The results showed that this core's project is on the right way for a consistent and complete NoC performance analysis. Except for the interference of the result packets on the latency measured on some cores, they followed the tendency of the pattern used.

They also appointed the next steps of this project. The future works are the separation between the moments of measuring latency and sending result, the insertion of the measure throughput into the project, making it more complete, and the implementations in the core's hardware of traffic patterns and other flow control algorithms, allowing a better analysis of a bigger number of NoCs.

7. REFERENCES

[1] Loghi, M. et al. "Analyzing On-Chip Communication in a MPSoC Environment". In: Design, Automation and Test in Europe Conference and Exhibition, 2004. Proceedings… [S.l.:s.n.], 2004.

[2] Benini, L.; Micheli, G. D. "Networks on Chips: A New SoC Paradigm". IEEE Computer Society Press: 2002; Vol. 35, pp 70-78.

[3] Duato, J.; Yalamanchili, S.; Ni, L. "Interconnection Networks". Elsevier Science, 2002, 600 p.

[4] Altera Corporation, "Nios II Processor Reference Handbook", http://www.altera.com, 2008.

[5] Altera Corporation, "Avalon Interface Specifications", http://www.altera.com, 2008.

[6] Zeferino, C. A. "Redes-em-Chip: arquiteturas e modelos para avaliação de área e desempenho". Dissertação de Mestrado, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2003.

[7] Carara, E. A. "Uma Exploração Arquitetural de Redes Intra-chip com Topologia Malha e Modo de Chaveamento Wormhole". 65f. Trabalho de Conclusão II, Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, 2004.

[8] Rego, R. S. L. S. "Projeto e Implementação de uma Plataforma MP-SoC usando SystemC". Dissertação de Mestrado em Sistemas e Computação, Universidade Federal do Rio Grande do Norte, Natal, 2006.

[9] Araújo, S. R. F. "Estudo da Viabilidade do Desenvolvimento de Sistemas Integrados Baseados em Redes em Chip sem Processadores: Sistema IPNoSys". Dissertação de Mestrado em Sistemas e Computação, Universidade Federal do Rio Grande do Norte, Natal, 2008.

[10] Dally, W.; Towles, B. "Principles and Practices of Interconnection Networks". Morgan Kaufmann Publishers Inc.: 2003.

[11] Mello, A. V. "Canais Virtuais em Redes Intra-Chip: um Estudo de Caso". 44f. Trabalho Individual I, Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, 2005.

[12] Wolkotte, P. T.; Hölzenspies, K. F.; Smit, J. M. "Fast, accurate and detailed NoC simulations". In: International Symposium on Networks-on-Chip (NoCS), 2007.

[13] Genko, N.; Atienza, D.; De Micheli, G.; Mendias, J. M.; Hermida, R.; Catthoor, F. "A Complete Network-On-Chip Emulation Framework". Design, Automation and Test in Europe (DATE), 2005.

[14] Zeferino, C. A.; Santo, F. G. M. E.; Susin, A. A. "ParIS: A Parameterizable Interconnect Switch for Networks-on-Chip". In: 17th Symposium on Integrated Circuits and Systems Design (SBCCI), 2004, Porto de Galinhas. Proceedings. New York: ACM Press, 2004. p. 204-209.