A PREDICTIVE NETWORK-ON-CHIP WITH STAR-MESH TOPOLOGY

Omitted due to the blind review process

Department
University
Address
email@university.country

Omitted due to the blind review process

Department
University
Address
email@university.country

ABSTRACT

The concept of Network-on-Chip (NoC) always was considered an adequate solution to the challenges presented by Systems on Chip (SoC). These systems are continually growing in terms of quantity of interconnected cores. Different NoC topologies have been proposed so far, searching for solutions in terms of performance or reduction in silicon consumption. Following this approach, a NoC with star topology that may be organized as a mesh network, connecting several routers and allowing building large SoCs, is presented in this paper. Its main features, though, is that it was designed aiming to allow the previous knowledge of maxim latency for any traffic and may be used on regular or non-regular mesh networks. This NoC was implemented on FPGA and compared with other similar topologies in terms of performance and silicon cost, and proved to be a good alternative for interconnections on SoC that demanding for real time aspects.

1. INTRODUCTION

The uses of Network-on-Chip emerged as a solution capable to attend the needs of SoC in terms of scalability, re-usability and to minimize electric problems. These electric problems may occur in the connections between cores inside the integrated circuit and are due to the growing in the communication speed in these devices [1].

The first's proposals of NoC were applied to regular mesh 2-D topology, as Ætherial [2] and QNoC [4]. On that NoCs, each router is connected to a process element (PE) or core. This approach implies on three potential problems: *i*) the growth of communication latency between two cores that are located on the opposite boarders of the network, when the network receive more devices and grows its size; *ii*) the latency of a flow may be strongly influenced by other flows that are concurring by the same communication channel, making impossible the knowledge of latency limits on the network; and iii) the growth of silicon cost, due to the growth of routers that are incorporated to network to build a SoC with more cores.

The network QNoC presented a solution to minimize the impact in growth of silicon consumption. Its approach was allocate the cores on the network to minimize the spatial traffic density, allowing that routers and communication channels that were not used to be discarded, on design time.

Another network that presented solution to minimize the silicon consumption was the network SoCIN [5]. Unlike the QNoC network designed for non-regular mesh 2-D topology, the network SoCIN was designed for the regular mesh 2-D topology and shown as its differential that it may be scaled to attend the requirements of the target application, using the technique of routing and arbitration distributed.

The initiatives of QNoC and SoCIN networks attacked the silicon cost problem, though they remain with the potential problem of significant big latency. The star topology, proposed by networks like Mesh-Star [6] and Star-Wheels [7], emerge as an efficient alternative to reduce the latency on big SoCs. Moreover, [8] showed that topologies based on star provide a reduction on silicon consumption if compared to other topologies.

The networks with star topology that were presented so far, do not offer treatment to a potential problem: how to determine the latency limits in a network without previous knowledge of the flows that may be flowing through the network. Is in this context that is proposed in this paper a novel approach of network with star topology that offer predictability of the latency limits for any flow on it, offering a communication structure with worst case execution time (WCET) to SoCs that demanding for this kind of real-time aspect in the communication among its cores.

The network, here called Real Time Star Network-on-Chip (RTSNoC), implements a scheduling algorithm like round-robin and may be used in regular or non-regular mesh, minimizing the impacts of silicon consumption. The Sections that follows describes the work that was done with this network, showing related works, describing the functional structure of the network proposed and evaluating the obtained results, in terms of silicon consumption and the latency to transmit packages through the network.

2. RELATED WORKS

The Network-on-Chip that uses the star topology allows up to six or eight connections of elements in the communication channels of its routers, and may be cores or communications channels from other routers.

The Star-Mesh NoC uses the regular mesh 2-D with eight communications channels. On that network, four communication channels are used to connect cores, while the others four are used to communicate with other routers in the network. The packages from Star-Mesh are composed by word with eighteen bits, being that the two most significant bits indicates if the FLIT (Flow control unit) is a package header, a package tail or a regular package. The routing adopted is XY and the scheduling is based on the CDMA technique.

The Star-Wheels NoC presents a heterogeneous topology that combines the star topology with a wheel topology. On that network, the cores are connected in a router with two bi-directional channels, called Subswitch, forming a network with wheel topology. This Subswitches also are connected to a specific router, called Superswitch, that is used when is necessary the communication between two cores that are connected in different Subswitches. The interconnection of these Superswitches is done by a special router called Rootswitch. The packages of that network are composed by word with fifteen bits, being that the three most significant bits are used as header to differentiate the six modes of control allowed in that network.

The Multi-Cluster Network-on-Chip, proposed by [9], is based on the architecture of router with six bi-directional communication channels. It was designed to work with multiples processors, connected in its routers, forming clusters. The clusters may be configured to communicate using four types of standards: ring, star, pipeline and tree. The package of that network are composed by words with thirty two bits, being that in the package header are reserved three bits to determine the address of a core that generates a package inside the cluster and seven bits to identify the cluster where this package have been generated. Similarly, are reserved more three and seven bits to represent the core and the cluster that will receive the package, respectively.

The RTSNoC network proposed in this paper also uses a router with eight bi-directional communication channels. The number of bits that compound a package word is parametrizable and may be defined in design time. In the Section that follows, the RTSNoC will be showed, detailing the main features and describing the differential over the other networks so far published.

3. THE RTSNOC ROUTER

The Network-on-Chip RTSNoC was developed to create an intercommunication structure with knowledge of latency limits, being appropriate to interconnect the real-time cores that need to know the WCET of the communication environment. The main element that compounds this network is its router.

The router of RTSNoC network has eight interconnections points, as showed in the Figure 1. The eight corners receives the names of compass cardinal points: NN to the north corner, NE to the northeast corner, EE to the east corner, SE to the southeast corner, SS to the south corner, SW to the southwest corner, WW to the west corner and NW to the northwest corner. On each corner of the RTSNoC router have a communication link that are the communication channels of the network.

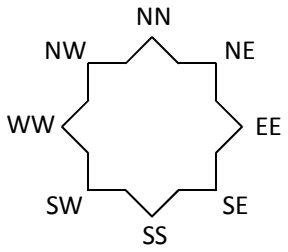

Fig. 1. Diagram of the RTSNoC router. Each corner is a point of connection to the network**.**

The communication links of RTSNoC are implemented by two opposite unidirectional channels, as shown the Figure 2. The size of each channel may be configured by the designer according the application needs. The signal buses called DIN and DOUT refers to the input data and output data buses, respectively. The signal RD and WR are strobes used to read or write a data in the router, respectively. Finally, the signals WAIT and ND are used to the flow control in the network. The WAIT signal indicates to the core that it has to wait before write a new data in the router input. The signal ND indicates to the destination channel that there is a new package available to be routed.

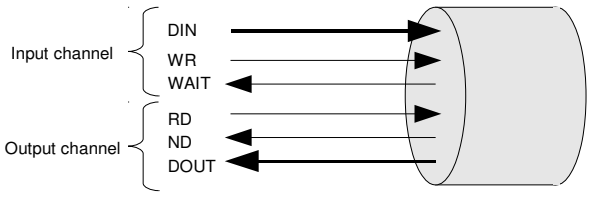

Fig. 2. Diagram of RTSNoC communication link.

Using just one router and eight cores is possible to build a sub-network RTNoC. The interconnection of more routers, beyond the cores, allows establishing a regular or

irregular mesh 2-D, as showed in Figure 3. Networks like SoCIN do not allow implementing irregular mesh networks. In that case, routers without cores must be inserted in the network to keep it regular.

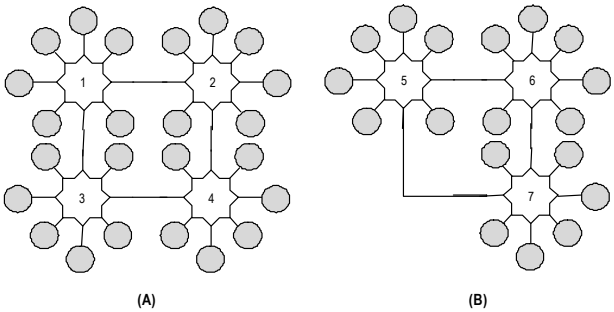

Fig. 3. Examples of RTSNoC: (A) - a regular mesh with 4 routers (1 to 4) and 24 cores interconnected; (B) - a non-regular mesh RTSNoC with 4 routers (5 to 7) and 18 cores interconnected.

The concept of sub-networks inside a network is important in the RTSNoC architecture, because the structure of the packages that flow through the network were built based on this concept.

3.1. Packet format on RTSNoC

The router of RTSNoC network is parametrizable on design time. The adjustable parameters are the data size field and the Cartesian coordinates of the routers involved in some communication. The Figure 4 shows this packet format.

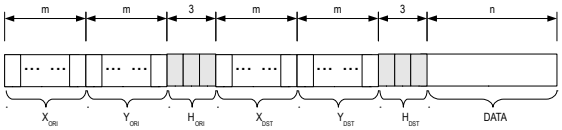

Fig. 4. Format package of RTSNoC. Two parameters may be change before the synthesis.

As the RTSNoC network works with regular or non-regular mesh 2-D, two fields called X_{ORI} and Y_{ORI}, are used to indicate the coordinates X and Y of the router where the package was generated, i.e., the origin address of the packet. The same happens to destination router, that have the fields called X_{DST} and Y_{DST}, to indicate the coordinates that defines the destination of the package. The field H_{ORI} refers to port address of router where the package has been sent. Similarly, H_{DST} is the information of destination core.

The field DATA showed in Figure 4 is the pay load that has to be sent to the cores in the network. The limitation of one PHIT (Physical unit) per package is contained in the network level of NoC. Packages with non-limited number of data may be built logically among the cores, transparently to the NoC.

In the RTSNoC the packages have fixed one FLIT size and each FLIT has one PHIT. This approach was chosen to minimize the silicon cost at the router, and the responsibility to implement buffers belongs to the cores, as the case of processors that have DMA mechanisms with embedded memories. Another motivation to this chosen was the fact of using this approach the maximum latency is minimized, thereby ensuring a higher throughput for all flows that are competing for the same communication channel.

Consider the scenario of a regular mesh network that works with best-effort service, shown in Figure 5. In this case, a core, called A, wishes to send a packet to another core, named C. At the same time, a core called B also initiates the transmission of a packet destined to the core C. As the number of hops between cores A and C is lower than in the case of the path between the cores B and C, the core A will catch the channel and may transmit its packet to C. Considering the NoCs that works with wormhole switching mechanism, the flow of B must wait until the flow A has finished and the core A release the channel allocated for transmission. In this case, the core B has no way to know how long will be the transmission.

Consider another scenario, using the same configuration as Figure 5 but now with routers that works with time-division multiplexing technique and have mechanisms to guarantee service, as Æthereal. In this case, the predictability is guaranteed due to allocation of time slots to priority flows. This architecture is free of contention and has lower variation on the jitter that the RTSNoC. Though, according [2], the performance of connections with guaranteed services imply in reserves of resources as buffers, for instance. To achieve 100% of guarantees, the reserve of resources has to be done for the worst case use, which imply on some wasting of bandwidth not used and bigger silicon consumption.

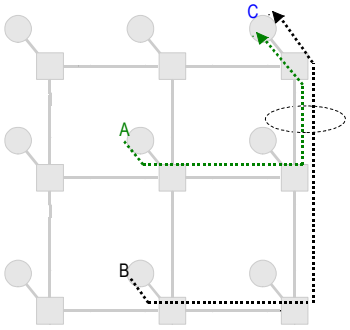

Fig. 5. A example of two cores (A and B) that are requesting the same channel and intending to send packages to the same destination (C) on a NoC with regular mesh topology.

As could be observed in this Subsection, the RTSNoC can ensure the maximum and minimum limits of latency for any flow through the network, due to the approach of consider all packages with a single PHIT. Another important feature that could be observed is the perspective to have a lower silicon cost, proportionate by absence of memories blocks. The Subsection that follows describe in details the internal structure of the RTSNoC router that was implemented to achieve the objectives purposed in this work.

3.2. Internal structure of router

The proposed router has eight channel bi-directional, which may be connected to cores or to channels from other routers. The Figure 6 shows the internal structure of the RTSNoC router.

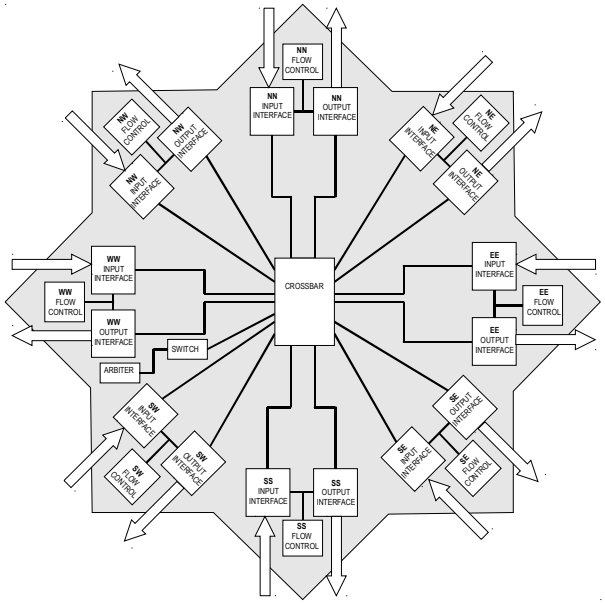

Fig. 6. A block diagram showing the internal structure of RTSNoC router.

The routing algorithm adopted is XY and for this reason the channels North, South, East and West are priorized to connections to channels from other routers, when is necessary, to form a mesh 2-D network.

Each input channel showed in Figure 6 is composed by an input interface, an output interface and by a flow controller. The input interface has a register that may store a PHIT. When a core desires to send packages through the network, it has to write the package in this register. Then, the flow controller will identify in the package header which one is the destination of the message and also will check if the destination is able to receive a new package. Meanwhile, the flow controller informs to the arbitration block that there is a new routing request.

The arbitration block implements a round-robin algorithm. After the system reset, all channels receives a priority level, different from each other. A core will has its routing request accepted just if it has priority greater than the other cores that are sending requests to the arbiter at the same time, or else, if there is no other request pending in the arbiter than your request. Once the request is accepted, the channel that ordered the dispatch will have lower priority in the next arbitration and just may send another package in sequence if there is no other channel requesting a routing to the same destination.

Once have been defined which channel receive the priority in routing, the arbitration block sends a command to the switch block, informing which routing has to be done at that time. This algorithm is used in NoCs with regular mesh topology and in its operation the flow has to be routed first in the X axis and then perform the routing in the Y axis. This approach is used to avoid the problem of deadlock caused by the allocation of resources made by two or more flows.

The packages of RTSNoC have only a PHIT and there are no needed to allocate channels over the network to send packets, turning impossible the occurrence of deadlock problem, which may occur with other NoCs that works with wormhole switching type.

Due to XY algorithm be used in routing in regular mesh networks 2-D, it is necessary to allocate specific channels of the routers so that we can ensure the operation of this type of algorithm in an irregular mesh. In the example shown in Figure 3B, the West channel of the router numbered seven is connected to the channel south of the router numbered five. If this connection had not been performed, a packet generated by any core allocated in the router numbered seven could not reach any cores connected to the router numbered five.

Once the arbitration has been performed, the arbiter block sends a command to the switch block. The switch block is responsible for controlling the crossbar that connects any input channel to any output channel, performing the package switching process.

The RTSNoC router needs two clock cycles to perform a package routing. One clock cycle is wasted with the arbitration process and other clock cycle is wasted with the routing process. The section that follows will present the methods used to implement the RTSNoC router and its results are compared with other NoCs in literature.

4. IMPLEMENTATION AND RESULTS

The implementation and validation of RTSNoC network was done using the synthesis tool for FPGAs, from the manufacturer Xilinx®, called ISE® version 12.1. Test benches files were written and applied simulation experiments to validate the models, which were synthesized targeting a FPGA from Xilinx® Virtex II. This family of

FPGAs was chosen because some NoCs from others authors were published with synthesis results for these components. Therefore, in this case is possible accomplish some comparison with others synthesis results from that NoCs and the results from RTSNoC.

The Sub-section that follows shows the results from the synthesis tool concerning to silicon consumption when compared to declared by the others networks.

4.1. Silicon consumption

As reference, was synthesized just one router inside the FPGA. The ISE® tool does not support to Virtex II family, though, it allows to the synthesis of Virtex 4 family. The devices from the family Virtex 4 have an internal structure very close to the devices from Virtex II family. Therefore, was chosen the device XC4VFX100-12FF1152 from Virtex 4 family, thus ensuring a close comparison to the silicon consumption with the NoCs synthesized on the Virtex II by others authors.

Three networks were used as reference for silicon consumption. The networks were chosen because they published implementation details on FPGAs from Xilinx®, also used in this paper.

The first network was the Hermes network. It was proposed by [10] as an infrastructure as capable to generate Network-on-Chip with regular 2-D topology. In that NoC, the designer defines the desired network from configurable parameters, such as flow control mechanism, the size of network and communication channel bandwidth. On the topology of Hermes network, each router has a different number of communication channels, depending on their position with regard to the limits of the network. At least one network router may have five channels. For example, in one Hermes network, 3x3 size, the central network router has five channels. Each edge router, located at the ends of the diagonals of the network, have only three communication channels. The rest of the routers have four communication channels each. The network was implemented in XC2V100 device, Virtex II family, manufactured by Xilinx ®, and all communication channels have been implemented with 8bits.

The second network was the SoCIN. According [5], that NoC can be scaled to meet the requirements of cost and performance of the target application. Originally it was designed as best-effort (BE) network for the treatment of data flows. The SoCIN network uses the ParIS router (Parametrizable Interconnect Switch). That router has up to five bidirectional communication ports. Similar to the Hermes network, the network SoCIN allows the user to set some configuration parameters of the network as a mechanism for flow control, network size and channel width. The Paris router was implemented on the device XC4VLX25 Virtex-4 family 12FF676, manufactured by Xilinx ®, and was configured to work with communication channels of 8 bits.

The third network was the Star-wheel NoC. It presents a heterogeneous topology that combines the star topology with a wheel topology. In this network, there are three different types of routers: Subswitch, Superswitch and Rootswitch. Due to their operational characteristics, each router has a different consumption of silicon. Packages for this network are composed of 8-bit words and its synthesis was done at the XC4VF100 Xilinx®.

The three networks are compared sizes in communication channels with 8 bits size. In addition, Star-Wheel has three different routers that are used according to the need for network design. Table 1 shows the consumption of silicon for routers and reported in the references are compared with versions of the router RTSNoC communication channels with 8 bits. Furthermore, Table 1 shows the consumption of silicon for the three possible versions of the Star-Wheel routers.

Table 1. Consumption of silicon resources per router.

Router	Silicon resources					
	SLICEs		LUTs		FFs	
	Other	RTSNoC	Other	RTSNoC	Other	RTSNoC
Hermes	*	807	631	429	200	1131
SoCIN	201		386		186	
Subswitch	588		*		*	
Superswitch	2318		*		*	
Rootswitch	1689		*		*	

* Not declared by the authors.

The values in Table 1 refer to the silicon consumption of a router for each NoC. However, in a SoC, is expected that the number of cores should be large and requires a larger number of routers per chip.

Consider the example presented by [11] in the paper "Performance Evaluation of a NoC-Based Design for MC-CDMA Telecommunications using NS-2. It is a 4G modem with high performance used in mobile telecommunications systems and is based on the technique of Multi-Carrier Code Division Multiple Access (MC-CDMA). The Figure 7, extracted from that article, presents this application scenario.

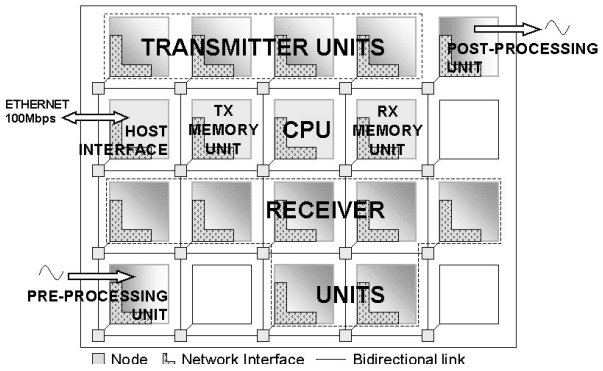

Fig. 7. The MC-CDMA implemented on a NoC with regular mesh 2-D topology. Font: [11].

In that paper, the system was implemented in a NoC and was divided into seventeen different cores. If those cores were connected to Hermes or to SoCIN networks, each one would take twenty routers, because these two networks working with regular mesh. Since the Star-Wheel Rootswitch need a three and seventeen Superswitch Subswitch. On the other hand, the RTSNoC requires only three routers to implement that Modem 4G. Table 2 shows the values of silicon resources estimated for these scenarios.

Table 2. Consumption of silicon for SoC based on the implementation of a MODEM 4G using different NoCs.

NoC	Silicon resources		
	SLICEs	LUTs	FFs
Hermes	*	12620**	4000**
SoCIN	4020	7720	3720
Star-Wheel	18639	*	*
RTSNoC	2421	1287	3393

* Not declared.

** Declared consumption only for communication channels with 8 bit size.

The Figure 8 illustrates graphically the data presented in Table 2. The column of SLICES (blue) was not mentioned by the authors of Hermes network and for this reason it was not shown in the chart. The same applies to the information of LUTs (red) and FFs (green) of Star-Wheels.

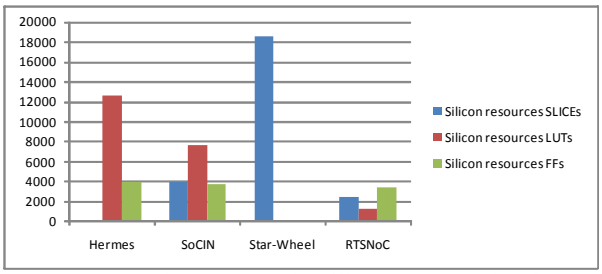

Fig. 8. A graphic based on the silicon consumption from Table 2.

For a real SoC, the RTSNoC shows up as a lower cost alternative to silicon consumption, when compared with other systems presented. The next subsection will present the results of the validation process of RTSNoC and their performance will be evaluated taking into account the criterion of latency.

4.2. Functional evaluation and latency analisys

To confirm the correctness of the RTSNoC network router, was used the simulation tool of Xilinx®, called ISE®. Different patterns of packet traffic were generated in order to confirm the proper operation of the circuit.

Figure 9 shows a diagram of waveforms of one simple simulation showing that the operation of the priority mechanism on arbitration was guaranteed. In that Figure, four flows compete for the use of communication channel SE. After the initial reset of the system, the order of priority to access the same channel is (from the highest to lowest priority): NN, NE, SE, SE, SS, SW, NW and WW. In the proposed simulation, cores located in the channels NN, NE, ES and SS initiated the requests to send packets at the same time. The packets sent by each communication channel are (in hexadecimal values):

- `08811300AAh` (channel NN);
- `08891300CCh` (channel NE);
- `08911300EEh` (channel EE); and
- `08A11300FFh` (channel SS).

In this case, given the priority criterion, the core located at the NN channel sent its package to the SE channel, as shown in Figure 9(1). Following the order of priority, other packets were sent from NE, EE, and ending with the dispatch of the SS package, as shown in Figure 9, (2) up to (4). The order of packet delivery is confirmed in Figure 9(5).

Other scenarios were tested using different configurations of traffic and different network topologies using more than one router. In all cases the maximum latency to the network was respected, independently of the traffic or topology used in the test.

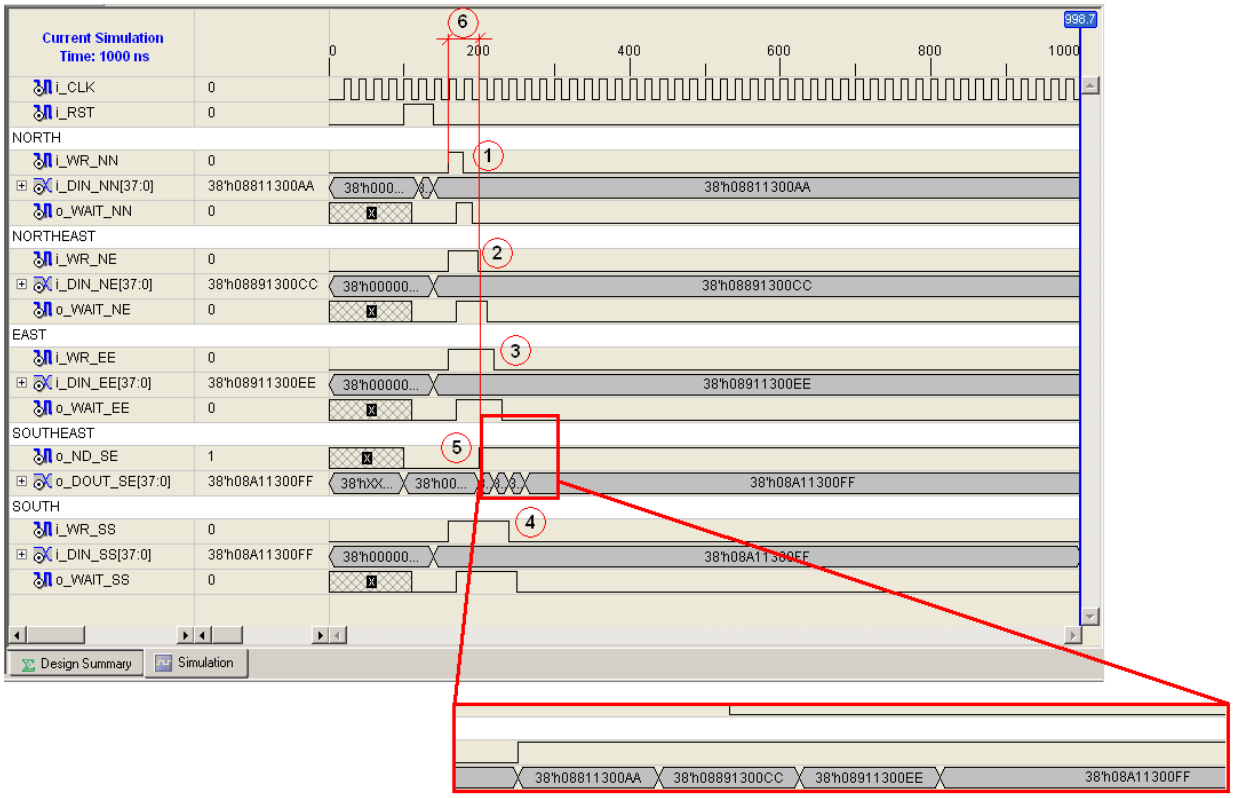

Fig. 9. A waveform showing the competition for a channel.

The latency provided by the router RTSNoC may be seen in Figure 9(6). On that Figure, the delay of the packet sent by the NN channel is two clock cycles. A cycle is due to the arbitration process and the second cycle is due to the routing process. Then, the router of RTSNoC network offers a latency of two clock cycles to routing a packet. The packages of channels NE, EE and SS that are competing for channel SE receive one more cycle for each channel with higher priority than yours. For example, the package sent by SS has one clock cycle for arbitration, one more cycle of delay due to the NN package, one more cycle delay due to the NE packet, one more cycle due to the EE package and another one due to the sending of its own package, totaling five cycles of delay.

Therefore, the limits of network latency that the RTSNoC may submit to a particular flow can be calculated considering that the minimum latency is twice the number of routers between the source of the package and its final destination, as shows (1).

$$L_{MIN} = N_{ROU} * 2, \qquad (1)$$

where L_{MIN} is the minimum latency and N_{ROU} is the number of routers on the path between the source of packet and the destination of packet. The maximum latency is determined adding a unit to the largest number of cores that may compete by some destination channel in a given instant, for each router on the path between the source of packet and the destination of packet. The expression that determines the maximum latency is given by (2):

$$L_{MAX} = \sum (N_{REQ} + 1), \qquad (2)$$

where L_{MAX} is the maximum latency and N_{REQ} is the largest number of channels (or cores) that may request the same communication channel as the destination of your messages at the same time, for each router in the path between the packet source and final destination.

For the scenario of 4G modem, shown in Figure 7, the RTSNoC network gives as minimum latency two clock cycles (two cores located in the same router) and maximum latency as twenty clock cycles (L_{MAX} = [6 + 1] + [6 + 1] + [5 + 1]). The other networks cannot offer latency predictability unless the cores establish a ceiling for the size of their packages. In the case of RTSNoC, the cores may have

packages of all sizes because these packets are processed in the upper layers of the network, and the NoC treats each PHIT as an independent package.

5. CONCLUSION AND OUTLOOK

This paper presented an original model of Network-on-Chip based on star topology and associated regular 2-D mesh. The router proposed in this network allows it to be connected up to eight cores or connected to other routers to form a regular or non-regular mesh network. The approach in limiting the network packet size allows predictability important for systems that work with real-time. Moreover, the structure proved to be a low cost alternative to silicon for the interconnection of complex SoCs.

6. REFERENCES

[1] Benini, L.; De Micheli, G.; , "Networks on chips: a new SoC paradigm," Computer , vol.35, no.1, pp.70-78, Jan 2002.

[2] Goossens, K.; Dielissen, J.; Radulescu, A.; , "AEthereal network on chip: concepts, architectures, and implementations," Design & Test of Computers, IEEE , vol.22, no.5, pp. 414- 421, Sept.-Oct. 2005

[3] Bobda, C.; Ahmadinia, A.; Majer, M.; Teich, J.; Fekete, S.; van der Veen, J.; , "DyNoC: A dynamic infrastructure for communication in dynamically reconfugurable devices," Field Programmable Logic and Applications, 2005. International Conference on , vol., no., pp. 153- 158, 24-26 Aug. 2005.

[4] Bolotin, E. et al. QNoC: QoS architecture and design process for network on chip. Journal of Systems Architecture, v. 49, 2003.

[5] Zeferino, C.A.; Susin, A.A.; , "SoCIN: a parametric and scalable network-on-chip," Integrated Circuits and Systems Design, 2003. SBCCI 2003. Proceedings. 16th Symposium on , vol., no., pp. 169- 174, 8-11 Sept. 2003.

[6] Woojoon Lee; Sobelman, G.E.; , "Mesh-star Hybrid NoC architecture with CDMA switch," Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on , vol., no., pp.1349-1352, 24-27 May 2009.

[7] Gohringer, D.; Bin Liu; Hubner, M.; Becker, J.; , "Star-Wheels Network-on-Chip featuring a self-adaptive mixed topology and a synergy of a circuit - and a packet-switching communication protocol," Field Programmable Logic and Applications, 2009. FPL 2009. International Conference on , vol., no., pp.320-325, Aug. 31 2009-Sept. 2 2009.

[8] Song, Z.; Ma, G.; Song, D.; , "Hierarchical Star: An Optimal NoC Topology for High-Performance SoC Design," Computer and Computational Sciences, 2008. IMSCCS '08. International Multisymposiums on , vol., no., pp.158-163, 18-20 Oct. 2008.

[9] Freitas, H.C.; Navaux, P.O.A.; Santos, T.G.S.; , "NOC architecture design for multi-cluster chips," Field Programmable Logic and Applications, 2008. FPL 2008. International Conference on , vol., no., pp.53-58, 8-10 Sept. 2008.

[10] Moraes, F.; Calazans, N.; Mello, A.; Möller, L.; Ost, L. "Hermes: an Infrastructure for Low Area Overhead Packet-switching Networks on Chip". Integration the VLSI Journal, 38(1), 2004, pp. 69-93.

[11] Lemaire, R.; Clermidy, F.; Durand, Y.; Lattard, D.; Jerraya, A.A.; , "Performance evaluation of a NoC-based design for MC-CDMA telecommunications using NS-2," Rapid System Prototyping, 2005. (RSP 2005). The 16th IEEE International Workshop on , vol., no., pp. 24- 30, 8-10 June 2005.