International Journal of Soft Computing 2 (2): 249-256, 2007

A New Generic GALS Router with Multiple QoS for NoC

Abdelkrim Zitouni, Mounir Zid, Mejdi Kerkeni, Sami Badrouchi and Rached Tourki
Laboratoire EμE, Faculté des Sciences de Monastir, Rue de l'Environnement,
5019 Monastir, Tunisie

Abstract: In the System on Chip (SoC) design, the synthesis of communication architecture constitutes the bottleneck which can affect the performances of the system. The Quality of Service Network on Chip (QNoC) is the most perferment solution that provides low latency transfers and power efficient System on Chip (SoC) interconnect. This study presents new asynchronous and generic router (GeRouter) for use in the new NoC generation architectures (Spider, Polygon, Octagon, etc). This router integrates a generic number of interconnected input and output ports, a sophisticated speed independent dynamic arbiter; a CRC based checking scheme, an Aloha retransmission protocol and two different routing techniques (Wormhole and Virtual Cut Through). This makes it possible to improve the Quality of Service (QoS) required by the NoC that integrate it. The performance study show that the proposed router enables higher data rate and low latency transfers.

Key words: GALS SoC, NoC, generic router, dynamic arbiter

INTRODUCTION

Networks on Chip (NoC) have received considerable attention recently as a solution to the interconnection problem in highly-complex chips. The major reason is that NoC help resolve the electrical problems in new deep-submicron technologies, as they structure and manage global wires.

Now a days, the majority of the NoC use the Mesh 2D topology as communication architecture (Liang *et al*., 2000, 2004; Goossens *et al*., 2005; Kumar *et al*., 2002; Millberg *et al*., 2004; Wiklund and Liu, 2003; Uriel and Prabhakar, 1992), to name but few. Due to their fixed architectures and data structures the majority of these topologies have a limited band-width.

In order to allow an efficient and extendible band-width, other topologies have been proposed by other NoC designers. The networks, such RSPIN (Karim *et al*., 2002) or OCTAGON (Alho and Nurmi, 2003) use a model of distributed communication and try to resolve the problems of latency, flexibility and reactivity which are more and more required by the multimedia applications of the future generation such as MPEG-4, software radio and mobile telephony, etc. The networks Proteo (Bainbridge and Furber, 2002) and Chain (Bainbridge and Furber, 1998) are two networks which support the Globally Asynchronous Locally Synchronous (GALS) formalisms and thus allow the power minimization. The network suggested by Dally, (1992), Dally and Towles (2001) is based on a Tore 2D topology and uses the virtual channels technique to improve the necessary quality of services. The networks STNoC (Benini andMicheli, 2002) and GeNoC (Schmaltz and Borrione, 2005), respectively developed by the STMicroelectronics Company and the TIMA Laboratory make the possibility of meeting the increasing requirements of the designs of current and future SoC. These NoC are based on flexible and evolutionary packets, designed according to a layers based methodology. The notable advantage of STNoC compared to GeNoC is that it integrates adapters of interface making it possible to convert any protocol IP, (OCP, 2003) or STBus out of communication packets.

In order to develop networks whose dimensions of architectures are adaptable, by fixing the number and the sizes of the communication links according to the traffic of the application, we propose in this study a new generic router architecture called GeRouter. This router can be interconnected to a series of NoC architectures going from the tree structure to the simple ring. It integrates a sophisticated dynamic arbiter, two different routing techniques (Wormhole and Virtual Cut Through) and the Aloha retransmission technique and allows the error cheeking according to the CRC technique. The sizes and the depths of the FIFO contained in this router, the number of input/output ports, the number and the time of retransmission and the maximum numbers of the requests sent to the arbiter are generics. All these characteristics make the proposed router flexible and extensible according to the applicative aspect and thus improve the quality of service required by the application to be mapped on it.

Corresponding Author: Abdelkrim Zitouni, Laboratoire EμE, Faculté des Sciences de Monastir, Rue de l'Environnement, 5019 Monastir, Tunisie

Generic router: We have adopted a GALS router with N generic input/output ports, having each a bi-directional exchange bus suitable for the new NoC architectures generation. All inter-module communications are carried out in messages witch are divided in packets. These packets are partitioned into small flits, which are sent through the NoC using the Wormhole or the VCT (Virtual Cut Through) routing depending on the message length. This router implements many communication protocols such as flow control, Aloha retransmission, dynamic arbitration, CRC checking, etc.

Packet fields: To support varying communication requirements, three types of flits are considered in the proposed router: a header flit, body flit and a tail flit, indicating End-of-Packet (EOP). Each packet header (Fig. 1) carries information such as the nature of the flit Nat (control or data), related to data communication requirements.

The header contains also the quality identifier (QoS-id) of the service to be assured by the router, the address of the target router, the address of the source router, the priority of the packet, the number of the flits forming it and the CRC code. The priority field is composed of two bits. The "11" code is associated to the signalling packets such as urgent messages, short packets, interrupt and control signals that require low transport latency and represents the highest priority packets. The 10 code is associated to the real-time application packets. The 01 code is associated to the RD/RW packets such as short memory and register access. The 00 code is associated to the block transfer packets such as long messages and blocks of data that represents the lowest priority packets.

A data flit is composed of 4 fields Fig. 2. The data field contains the data to be transmitted. The Nbre field indicates the flit ordering number. The CRC field indicates the CRC checking code.

The EOP flit is a particular data flit which carries the last data of the packet. The flits format presented in Fig. 2 are dimensioned in the case of 32 bits but in general it is generic.

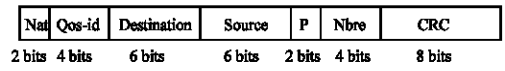

Fig. 1: The packet header fields

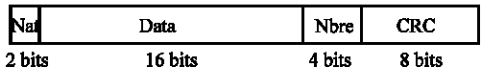

Fig. 2: Data flit fields

ROUTER PROTOCOLS AND MODULES

In the proposed router each node of the network is identified by a single number which is used as source or destination addresses in the packet heading according to whether the router is transmitter or receiver. This architecture is constituted by 4 modules (Port Management Unit (PMU), dynamic arbiter, routing table and the switch) that communicate with each other by using an asynchronous 4-phases protocol.

Port Management Unit (PMU): The port management unit (PMU) implements the routing mechanisms and the services which the router must offer to the communication. It integrates a routing function that calculates according to the information transported by the packet header, the address of the output port to which the flit will be transmitted. It also manages the data traffics through the port to which it is associated and performs the communication with the PMU units of the neighbour routers.

The PMU units of each router operate concurrently so it minimizes the routing latency of the NoC by allowing the flits to arrive simultaneously by any input port. As illustrated in Fig. 3 in the case of a router with 4 input/output ports each PMU unit is composed by four modules: a flow control unit, a clock generator, a memory unit (FIFO) and a routing and services unit.

The clock generator is intended to generate a stoppable clock signal to rhythm the PMU modules. When a given data transaction is initiated, the clock

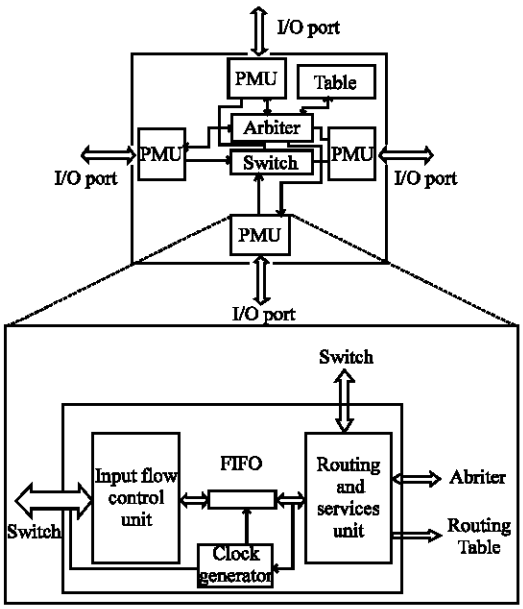

Fig. 3: The router and the PMU internal architectures

signal is activated. This signal is stopped only when all the flits contained in the FIFO are evacuated towards the routing and services unit.

The flow control unit, implements the flow control mechanism that ensure that the received flits are well ordered and they are not duplicated. Such a mechanism allows to a router receiving a flit to indicate to its neighbour router, sender of this flit, if it has or not a sufficient memory to store this flit. If the receiver has a sufficient memory it stores the transmitted flit and sends an acknowledge signal to the sender. The detection of this signal by the sender allows it to liberate the buffer or to reload it to send a new flit if it has others about it. If a given deterioration occurs on the level of a flit or that the input FIFO of the receiver are full, this flit is rejected by the router and no acknowledge signal is sent to the sender.

The flits integrity checking mechanism is based on flits counting and the CRC calculation. The flit counting mechanism uses the Nbre field of the header of the packet to be transmitted and allow the identification of EOP. The CRC checking uses an 8 degree polynomial code that is added as a suffix in each flit. The receiving router that has recomputed the CRC compares it with the suffix of the received flit. If the value of the recomputed CRC code is different from the transmitted one, the receiving router wait until this flit is transmitted again based on the Aloha retransmission technique ore reject it when the number of possible transmission is reached.

The flit counting mechanism, the Aloha retransmission technique, the dynamic arbitration and the CRC cheeking are actually the unique QoS services that are implemented in the proposed router. Other QoS can be added and they are coded in the QoS-id field of the header. For example the CRC code is 0001, etc.

The role of the generic FIFO module is to store the transmitted packets. The Wormhole switching technique is used if the size of the packet is higher than the depth of the FIFO and the VCT technique is used if the FIFO can store a whole packet. All the flits coming from the flow control unit are pipelined through the FIFO to the routing and service unit when they are transmitted. The pipeline nature of the Wormhole and the VCT techniques makes the latency quasi-insensitive to the length of the path separating the source and the destination.

After receiving a given flit and if this flit is a header, the routing and service unit takes the necessary information to determine the direction towards the data will be sanded. Then it consults the arbitration unit which manages the access to the selected port. If the routing and service unit receives an acknowledge signal from the arbiter module it sends the header in the corresponding direction and all the flits will be sanded in the same direction. The requester that is granted by the arbiter switches the heading towards the given output port. This port remains reserved until all packet flits are carried out. If during the routing step, this output port is blocked, the router stores the received flits on the corresponding input port in order to send them later when the output port becomes free. Each requester not having access to the output port and whose request was rejected by the arbiter increments its internal counter and wait for a Td time. Once this time is over, the requester starts again the same request with the arbiter. When the counter reaches a maximum value Nmax, the routing and service unit determines other output port and a new request cycle will begin. The Td time and the Nmax value are generics.

The aim of the Aloha retransmission mechanism is to resolve the problem of loss or the reception delaying of the acknowledgment signals transmitted by the router. This problem is fatal for the NoC since the transmitter can stop the transmitting, for lack of emission credit. The conflict which results from it can retro propagated until blocking the NoC completely. The advantage of the Aloha retransmission protocol lies in its simplicity. The flow chart of this protocol is given by Fig. 4 where Nmax represents the maximum number of retransmission and time-out indicates the maximum latency of the acknowledgement signals. The finite state machine of the routing unit is presented in Fig. 5.

Routing table and switch: After the arbitration phase, the arbiter sends a request to a table which checks the state

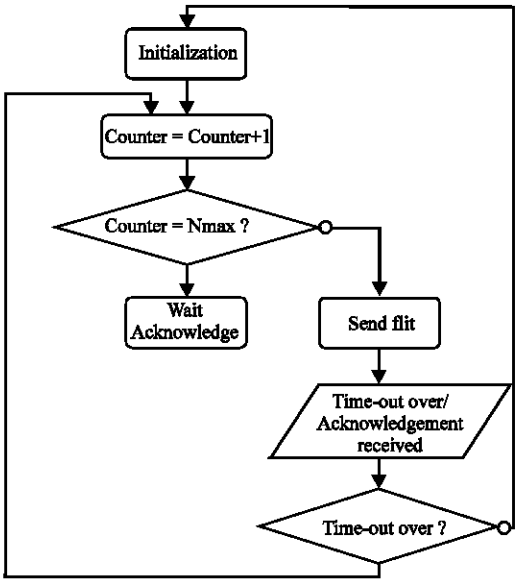

Fig. 4: Flow chart of the retransmission Aloha protocol

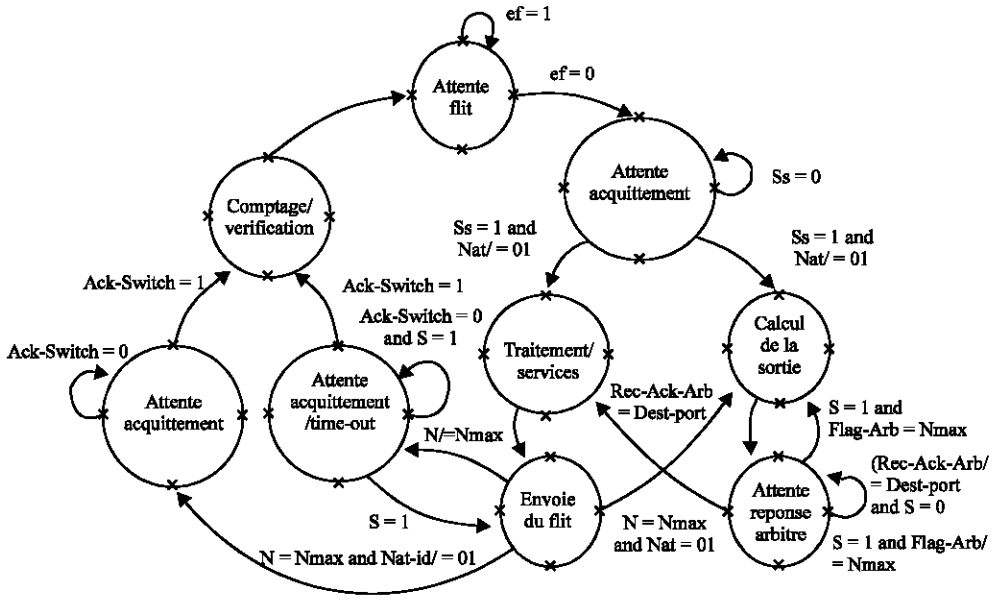

Fig. 5: Finite state machine of the routing unit

of the concerned output ports and acknowledges the arbiter if one of these ports is free. The state of each output port (free or occupied) is memorized in a register.

The switch allows commutating the flits coming from the PMU units towards the wearing of selected destinations. Each PMU communicates with the switch thanks to two signals UX and ADRX. UX is the port with 32 bits which conveys the flits coming from the PMU unit X and ADRX is the address coded on 3 bits chosen by the PMU unit. Address ADRX is communicated by the PMU unit X with the switch to choose the address of the output port through which the flit must be transferred. This address also makes it possible to the switch to acknowledge the external module (IP, etc.,) towards which the flit is intended.

Generic aspect of the router: The proposed router is a flexible, easily extensible and offers a variety of services to the communication owning to the fact that it is completely generic. It makes the possibility of choosing and modifying parameters such as the width of interconnection, the depth of the FIFO, the sizes of the fields of the flits, the maximum number of retransmission of a flit and a request of the arbiter and the propagation times of the signals which prove very important for the correct operation of the total system. Moreover, it is generic in terms of supported number of IP. The development of this router is based on a library of generic models of VHDL blocks. The files of this library contain protocol (number of retransmissions, allowed requests, Time out and size of each field forming the various types of flits) and physic (width and depth of the FIFO, number of input/output of the routers, etc.,) parameters. These files also contain all the functions used by the VHDL blocks like the path calculation function, the CRC checking function, etc.

Dynamic routing arbiter: In order to resolve access conflict to the output ports, we have designed a dynamic arbiter that allows the resolution of access conflict problems of each output port starting from the priority information of each incoming packet.

Each dynamic arbiter is constituted by a speed independent round-robin arbiter and a comparator module. Thus, a router with N input/output ports integrates N dynamic arbiters with N round-robin arbitration modules and N comparators as presented by Fig. 6.a in the case of a 2D mesh NoC arbiter that contains 5 input/output ports (Local, East, West, North and South). Each dynamic arbiter serves the 4 demands that are addressed to every port. The requests effectively allocated by the different comparators are treated by the corresponding arbiter modules.

Each dynamic arbiter is constituted by m comparators and m round robin arbitration modules interconnected by m C-elements (m represents the number of requesters) and an OR gate as shown by Fig. 6.b in the study of an arbiter with three requesters. After receiving the request signals from the requesters, the priority comparator stores the priority values from the priority busses of the active

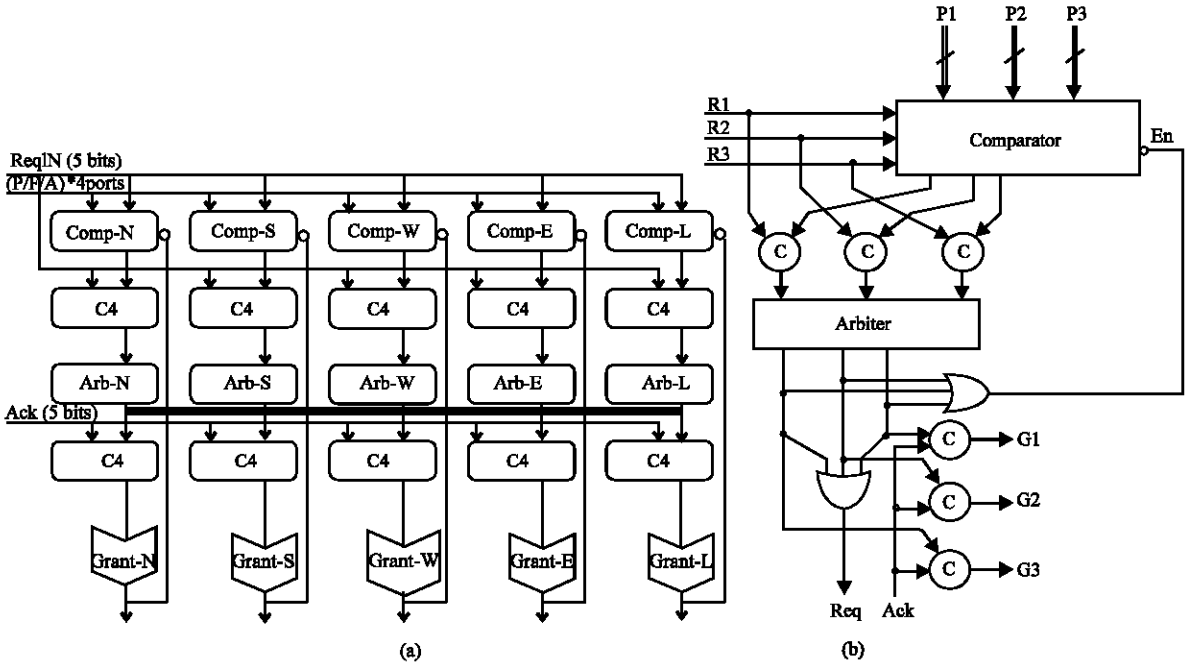

Fig. 6: Architecture view of the dynamic arbiter

requests and compares them. After the comparison step, the comparator sends out a set of internal signals that correspond to the requesters that have the highest priorities.

After being combined with three C-elements, the output signals are transmitted to the arbiter module. Thus, only the requesters that have requested the access and that have been selected by the comparator module will be treated by the arbiter. If we have only one requester that has been selected by the comparator it will be granted the access automatically by the arbiter and the priority list is shifted in a circular way. When we have two or three requesters that have the same priorities orders, then the requester that has the last highest priority order will gained the access and the priority list is shifted in a circular way. We notice that the outputs of the C-elements that enter to the arbiter module will be activated only if the requesters have requested the access and they are selected by the comparator. If a requester has gained the access the comparator module will deactivate the generated internal signals. This scheme is implemented by an enable (En) signal that is generated by an OR gate with three inputs (G1, G2, G3). The outputs of the C-elements that enter to the arbiter still high until the active requests become low. Also the comparator will be activated again only if the enable signal becomes active (the actual requester has released the bus).

Asynchronous arbiter synthesis: The specification model of a round robin arbiter is constituted by n (n is the number of requesters) communicating STG where each STG corresponds to a static priority graph. The activation and the inactivation of each STG are done by n internal signals as detailed below in the case of a round robin arbiter with two requesters.

In the study of a round-robin arbiter with two requesters we have two possible priority lists (two communicating STG). The STG(12) corresponds to the case where the requester R1 has the highest priority and the STG (21) corresponds to the case where the requester R2 has the highest priority. Starting from the communicating STG (Fig. 7) we insert two internal signals S12 and S21. The STG(12) is activated only if the internal signal S12 is activated (S12+). When G1 is activated and STG(12) is active, it activates the S21 signal and deactivates the S12 one. If G2 is activated and STG(21) is active, it activates the S12 signal and deactivates the S21 one. From this graph, the C-element production rules of all signals (G1, G2, S12, S21) are generated:

$$\begin{cases} G1+ = S12.R1.\overline{G2} + S21.R1.\overline{G2}.\overline{R2} \\ G1- = \overline{R1} \end{cases}$$

$$\begin{cases} G2+ = S21.R2.\overline{G1} + S12.R2.\overline{G1}.\overline{R1} \\ G2- = \overline{R2} \end{cases}$$

$$\begin{cases} S12+ = G2 + Reset \\ S21- = G1 \end{cases}$$

$$\begin{cases} S21+ = G1 \\ S21- = G2 \end{cases}$$

We notice that the requester R1 gains the access (G1+) only if this requester has requested the access (R1 high), the requester R2 is not on the bus (G2 low) and S12 is active (S12+, STG(12) is active =>R1 has the highest priority) or if the requester R1 requests the bus access (R1 high) and the requester R2 is not on the bus (G2 low) and does not request the bus access (S21+, STG(21) is active =>R1 has the lowest priority). The arbiter releases the bus from R1 (G1-) only if this requester has released the access (R1 low).

The C-element based implementation of this arbiter is presented in Fig. 8. The S12 and the S21 activation signals are generated by an RS flip-flop. For the S12 generation

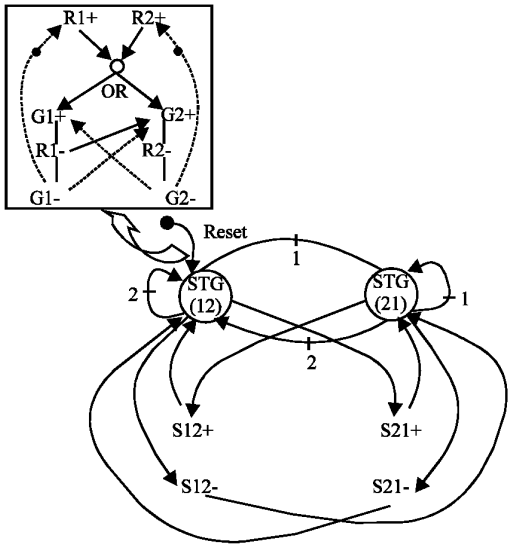

Fig. 7: Specification of a round-robin arbiter with two requesters

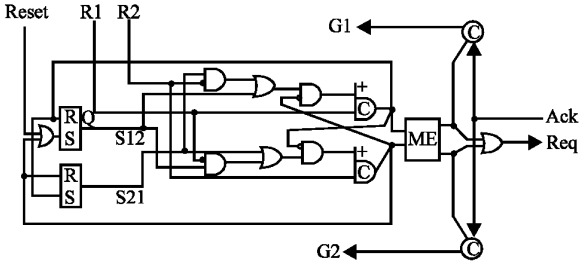

Fig. 8: C-element based implementation of a round robin-arbiter with two requesters

flip-flop, (G2+Reset) represents the Set entry and G1 represents the Reset entry:

$$S12 = \overline{G1}[(G2 + Reset) + S12]$$

In fact, at the beginning of the functionality we have assumed that the requester R1 has the highest priority and then the STG(12) is activated by the Reset signal. Starting from the production rules, the next state functions of the outputs G1 and G2 are generated:

$$G1 = R1.\overline{G2}(S12 + S21.\overline{R2}) + R1.G1$$

$$G2 = R2.\overline{G1}(S21 + S12.\overline{R1}) + R2.G2$$

The proposed method can be easily extended into an asynchronous round-robin arbiter with n requesters.

RESULTS AND DISCUSSION

The proposed GeRouter has been described at RTL level by using the VHDL language. We have checked their functionalities correctness by applying a set of test vectors for different parameters (sizes of the FIFO, time of retransmission, etc.,) for different NoC architectures.

Simulation of a 2D mesh NoC arbiter: Figure 9 presents the simulation results of the dynamic arbiter in the case of a 2D mesh NoC. It is then immediate to see, according to the demands, the behaviour of the arbiter. For example, the demands being to 01111 means that all ports are claimants except the North port. If we examine the asked ports, we notice that the port asks for the West and the Local ports. The West port asks for the South port. The last port asks for the East port. We notice also that ReqE, ReqS and ReqW are activated, since they are asked for. The demands pass then by the arbiters to adjust possible simultaneous demands (case of the demands Local and West toward the South port), it is sufficient to wait for an acknowledge signal to allow the access to every port.

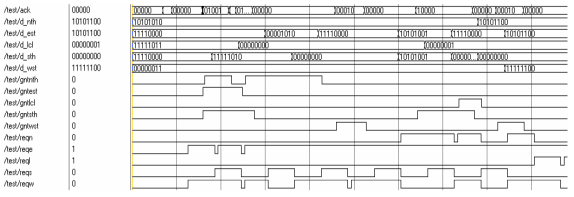

Fig. 9: Example of simulation results of the dynamic arbiter in a the case of a 2D mesh NoC

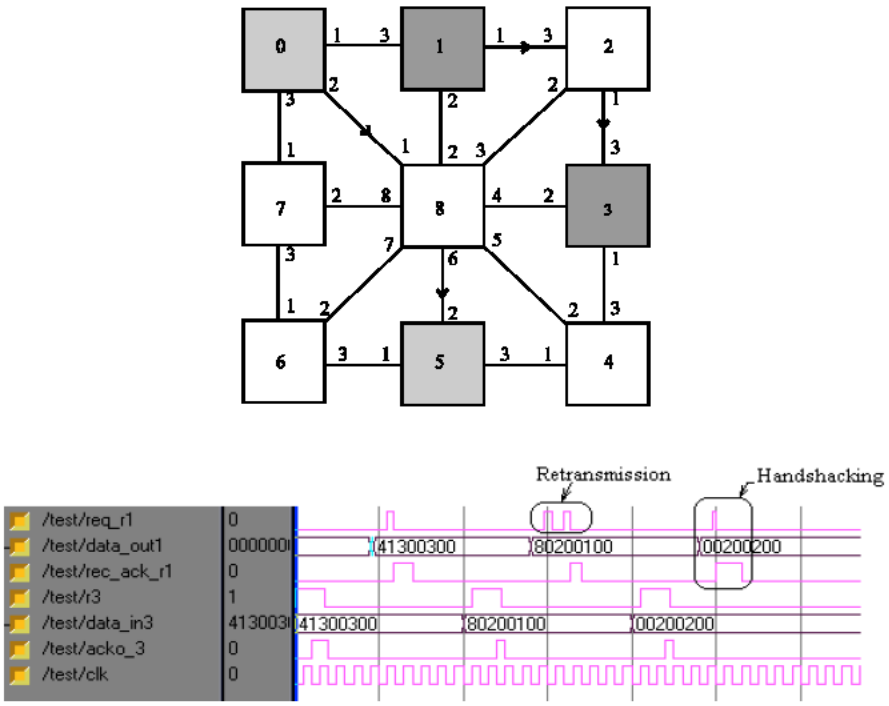

Fig. 10: (b) Example of simulation results of the router in a polygon NoC example (a)

Simulation of a polygon NoC router: Figure 10a illustrates the routing of a packet of three flit by the proposed router in the study of polygon NoC architecture of Fig. 10b with the following parameters:

- The number of input/output ports is fixed at 4.
- The maximum number of retransmissions of the flits and of the request signals is fixed at 5.
- The size of the data path (size of the flit) is of 32 bits.
- The depth of the FIFO is 4 flits.
- The fields of the flit are fixed at: WNat =2, WQoS = 4, Wdes = 6, WSou = 6, WP = 2, WNbre = 4 and WCRC = 8.

To 50 MHz this router offers a flit rate of 145 Mo/S. the latency of routing obtained is about 110ns (from 5 to 6 clock cycles). The latency of data flit routing is of 85ns (4 clock cycles).

Physical design of a 2D mesh NoC router: In the case of a 2D mesh, we have designed a synchronous and an asynchronous router with the same functionality by using a CMOS 0.35μm technology. The asynchronous router operates with 80 Mbytes/s as a data rate. A (5x5) 2D mesh NoC based on the asynchronous router occupies a 4881 mm by 4881 mm silicon area. Our design show that the asynchronous router enables a notable grater data rate than their synchronous counterparts (66 Mflits/s) but only a small difference less than the synchronous NoC in silicon area (4200 mm by 4200 mm). This is evident since asynchronous NoC naturally benefit from asynchronous interconnects.

CONCLUSION

This study presents a flexible and easily extensible asynchronous NoC router that offer a variety of SoC communication services owing to the fact that it is completely generic. This router is based on a dynamic routing function and allows to modify and to choose starting from a parameterized library different protocol and physic parameters. The protocol parameters consists on the Aloha retransmission time-out, the requests and their retransmission numbers, the size of each field forming the various types of flits, the routing technique (Wormhole or VCT), etc. The physic parameters consist on the width and the depth of the FIFO, the number of the input/output ports, etc.

Compared with other existing router, the value added by the proposed router, resides in its capacity to handle a suitable cost/performance compromise in the field of NoC. This thanks to its wide generic character.

In order to minimize the communication latency in the proposed architecture we are under integrating virtual channels. Moreover, in order to allow the use of the GeRouter at various levels of abstraction we are under

modelling it in SystemC language at TLM (Transaction Level Modelling) that is suggested by OCP-IP (Open Protocol-International Core Partnership).

REFERENCES

Alho, M. and J. Nurmi, 2003. Implementation of Router IP for Proteo Network-On-Chip, IEEE Int. Workshop on design and diagnostics of electronics circuits and systems DDECS'03, Poznan, Polandl, pp: 4-16.

Bainbridge, W.J. and S.B. Fuber, 2002. Chain: A delay insensitive chip area interconnect, IEEE Micro, 22: 16-23.

Bainbridge, W.J. and S.B. Furber, 1998. Asynchronous Macrocell Interconnect Using Marble, Intternational Symposium Advanced Research in Asynchronous Circuits and Sys., IEEE Press, pp: 122-139.

Benini, L., G. De Micheli, 2002. Networks on chips: A new SoC paradigm, IEEE Computer, 1: 70-80.

Dally, W.J., 1992. Virtual-Channel Flow Control, IEEE Trans. Parallel and Distributed Sys., 3: 194-205.

Dally, W.J. and B. Towles, 2001. Route Packets, Not Wires: On-chip Interconnection Networks, Design Automation Conference, Las Vegas, USA, pp: 684-689.

Goossens, K., J. Dielissen and A. Radulescu, 2005. Æthereal network on chip: Concepts, Architectures and Implementations, IEEE Design and Test of Computer.

Karim, F., A. Nguyen and S. Dey, 2002. An interconnect architecture for networking systems on chips, IEEE Micro, pp: 36-45.

Kumar, S., A. Jantsch, J.P. Soininen, M. Forsell, M. Millberg , J. Oberg, K. Tiensyrja and A. Hemani, 2002. A Network on Chip Architecture and Design Methodology, IEEE Computer Society Annual Symposium on VLSI. IEEE Computer Soc. Press, pp: 105-112.

Liang, J., A. laffely, S. Srinivasan and R. Tessier, 2004. An architecture and compiler for scalable on-chip communication, IEEE Transactions on Very Large Scale Integration Systems.

Liang, J., S. Swawaminathan and R. Tessier, 2000. aSOC: A Scalable, Single-Chip Communications Architecture, in the IEEE International Conference on Parallel Architectures and Compilation Techniques, pp: 524-529.

Millberg, M., E. Nilsson, R. Thid, S. Kumar and A. Jantsch, 2004. The Nostrum Backbone-a Communication Protocol Stack for Networks on Chip, VLSI Design Conference, Janvier.

OCP, 2003.International partnership. Open Core Protocol Specification. 2.0 Release Candidate.

Schmaltz, J. and D. Borrione, 2005. A generic network on chip model, Research reports, TIMA Laboratory, Grenoble, France.

Wiklund, D. and D. Liu, 2003. SoCBUS: Switched Network on Chip for Hard Real Time Embedded Systems, Proc. Int'l Parallel and Distributed Processing Symp., pp: 78-85.

Uriel, F. and R. Prabhakar, 1992. Exact Analysis of Hot-Potato Routing, The 33rd Annual Symposium on Foundations of Computer Science, IEEE Computer Soc. Press, pp: 553-562.