# 6X6 Crossbar Architecture of SOC based on NOC by Interleaving Technique

*K. Varun Kumar*

*M. Tech scholar at SITS Khammam affiliated to JNTUH*

**Abstract- Systems on a single chip(SoC) employed communication systems are an important contribution to their overall performance. Bus based paradigm employed previously has many problems. Scalability, Power performance still a problem . To overcome above problems, switch-based paradigm has been introduced. In addition, its changeable communication-pattern property depending on various applications at run-time is advantageous. Thus, the possibility of multiple applications running on single platform can be achieved. Nevertheless, many-to-one (gathering) functionality based on the switch-based paradigm has disadvantage at output delay. In this paper, interleaving mechanism is introduced to solve this problem. Moreover, 6x6 interleaving switch-based crossbar architecture along the proposed mechanism is implemented by using XILINX ISE tool.**

**Keywords: System On Chip(SOC), Network On Chip(NOC), Field Programmable Gate Array(FPGA), Finite State Machine (FSM).**

## I. INTRODUCTION

Buses in SOC are an increasingly inefficient way to communicate, since only one source can drive the bus at a time, thus limiting bandwidth. NoC: Network On Chip's are growing popular due to their advantages: higher bandwidth, and lesser power dissipation via small segments of wire. SoCs of larger types are very important which made many designers to adapt NoC approach. The best connectivity and throughput are the challenges with the simplest and cheapest architecture of methodology. This is well illustrated in [4], where researchers propose a two-level FIFO approach in order to simplify the design of the arbitration algorithm and improve the bandwidth. The idea of designing the Reconfigurable Crossbar Switch for NoCs to gain a high data throughput and to be capable of adapting topologies on demand was presented in [5]. Their evaluation results showed that output latency, resource usage, and power consumption were better than a traditional crossbar switch. Nevertheless, they did not focus the problem while operating many-to-one and one-to-many data communication. Our proposed crossbar architecture is designed with two contributions: First, our crossbar has to be lightweight, requiring few FPGA resources, and thus suitable for both small and large resources with high bandwidth efficiency on FPGA based systems. Second, the proposed crossbar architecture has to solve the output delay (output latency) problem while multicasting on all source situations. Moreover, in order to obtain the work efficiency, 6x6 interleaving switch-based crossbar is realized by Verilog, and verified on the Xilinx ISE tool. In respect of the verification, the test environment has introduced in this paper. The rest of this paper is organized as follows: section. Section 2 presents switch based crossbar structure. In section 3, we implement the primitive component and 6x6 interleaving switch-based crossbar, and compare the proposed switch-based crossbar with general propose switch-based system and bus-based system. Test environment and experimental result are presented in section 3. Finally, conclusion is summarized in section 4.

## II. SWITCH-BASED CROSSBAR STRUCTURE

one-to-one (unicast), one-to-many (multicast), many-to-one(gathering) and many-to-many communication problems are overcome by interleaving technique to switch-based crossbar architecture on FPGA, and compare resource usage as well as estimated frequency with [1,2,3].

The major components that make up the switch-based crossbar consist of Input Port module, Switch module and Output Port module. Depending on the kind of channels and switches, there are two alternatives for designing switch-based crossbar: unidirectional and bidirectional [6]. In this paper, unidirectional switch-based crossbar is selected and simplified with 6x6 switch-based crossbar structure shown in Fig. 1 because of resource constraint.

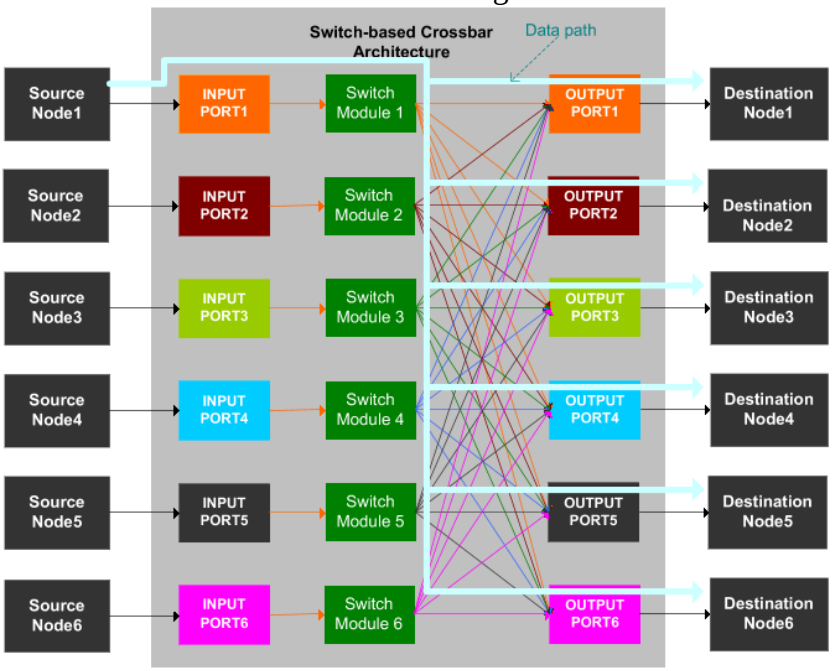



Fig:1 6x6 switch-based crossbar structure

Obviously, transmitting data from a Source Node to a Destination Node require crossing the link between the Source Node and the Input Port module, and the link between the Output Port module and the Destination Node where the Switch module in data path will dynamically establish the link for the Output Port module according to switching protocol. According to the 6x6 switch-based crossbar architecture, its switching protocol shows in Fig. 2. Because of resource constraint on the target FPGA [2], data width, the number of word register, the destination port registers are 16,10 and 6 bits respectively. Moreover, synchronous protocol is applied to synchronize all modules in the switch-based crossbar-Clock signal, Ready signal and Acknowledge signal.

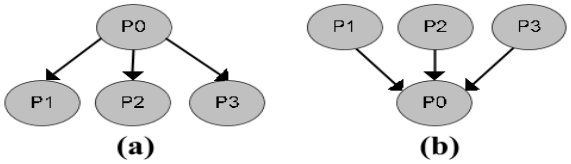



Fig 2: (a) One to Many (b) Many to One

Fig 3 shows the 10-bit Last Significant Bit (LSB) of switching protocol, which defines the number of packets required transferring from a Source Node to a Destination Node, where the maximum is 1,024 packets per time, and the rest define the Destination Node. For

example, when the Source Node1 wants to transfer 100 packets to the Destination Node number 3 and 4, the 16-bit switching protocol has to be 0011_0000_0110_0100 (0x3064H).

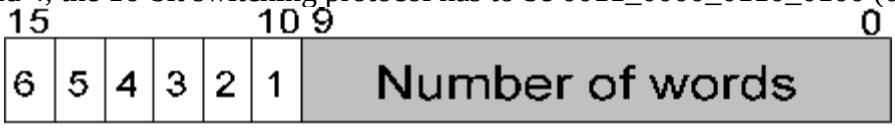



Fig 3: 16-bit switching protocol

**A. Input Port module**

In this section, the architecture and behaviour of the Input port module are detailed. As shown in Fig. 4(a), there are three components in this module comprised of a 16-bit tri-state buffer, a 1-bit tri-state buffer and a finite state machine (FSM) component where the N.of word register and the destination port register are resided. Its behavior shows in Fig. 4(b) based on switching conceptual [1]. At the beginning, the state is in an idle state, and then the header ready signal enables HIGH to inform a Source Node that ready to read the switching protocol. As soon as the 16-bit switching protocol is asserted at the Data in signal and the Data in valid signal, the state goes to the next check state. In this state, the 16-bit switching protocol is separated and written on the N.of word register and the destination port register resided in FSM module, where 10-bit low and 6-bit high are
written on the N.of word register and the destination port register; meanwhile, the register port signal is read to check, whether or not the required destination ports are free. If it is free, the state will go to the req state.

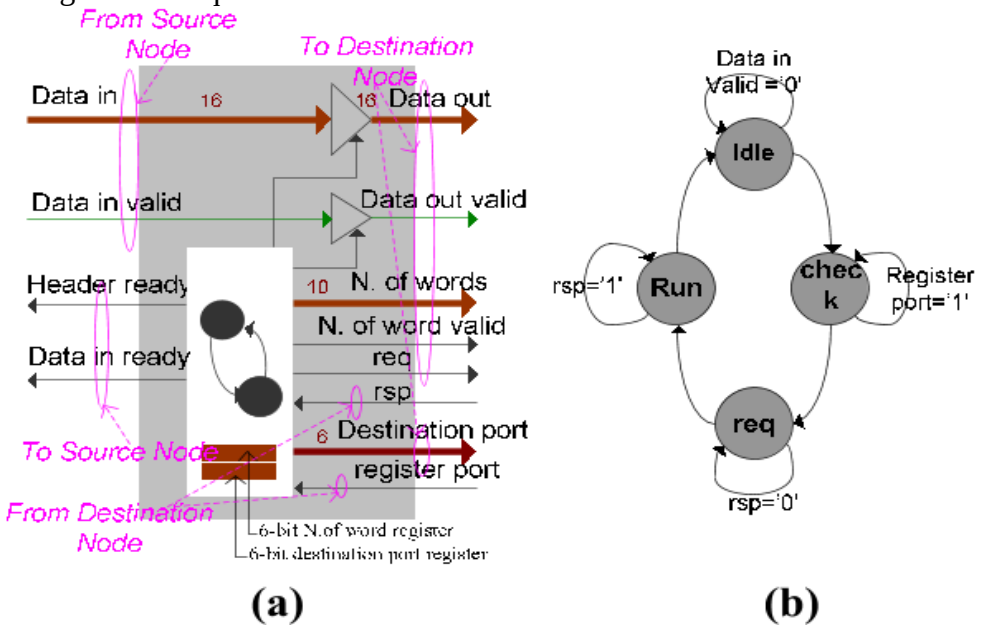



Fig. 4. a) Input Port module architecture b) State-machine

In the req state, the N.of words register, the destination port register and the req signal are read and presented on the N.of words signal and the N.of word valid signal. When the rsp signal enables HIGH, the channel for transferring data is guaranteed, and the state goes to the Run state. In the Run state, the data in ready signal enables HIGH. The 16-bit data in signal and the data in valid signal are sequentially asserted into the channel. When the lasts data is completely forwarded, the rsp signal from the Output port module will enable LOW, and the state will start to the next cycle.

## B. Output Port module

In this section, the Output Port module architecture and the flow diagram are explained in detail. As shown in Fig. 5(a), there are three components of this module, comprised of a 16-bit six-port multiplexer component which used to multiplex the Data in signals coming from the Switch Module, the Round-Robin component assigned to guarantee the fairness of all input requirements and the FSM component where the 10-bit counter register and the 6-bit Port Status register used to count down a number of through packets and to identify that can occupy the path of this Output port module are resided. Since interleaving mechanism bases on Time-Division-Multiplexer (TDM)[7], the
16-bit six-port multiplexer component can multiplex all Data in signals depended on Time Slot. For instance, Fig. 6(a) and 6(b) show the timing diagram, and the 6-bit Port Status register where all Source node and the Source Node number 1,2,5 and 6 require transferring data to one Destination Node, the 6-bit Port Status register in the Destination Node will be 111111 and 110011.
Fig. 5(b) explains its behaviour. At first, the state is in an Idle state, and the mode signals are read where there are two kind of possible modes, normal mode and interleaving mode. In the normal mode, the Round Robin component will be enabled but will be disabled in the interleaving mode. Whenever a Source Node requires transferring its data to a Destination Node, the req signal of the Input Port module connected with the Source Node will enable HIGH. Then, at the Output-Port module connected with the Destination Node, each bit of the 6-bit grant register related with each req signal of each Input Port module will set "1". The N.of words signal and the N.of words
valid signal will be read and stored into the temporal register. Then, the state goes to the Write state. In the Write state, each bit of the grant register is read to check the Source Node's requirement and to identify the location of the next state. In order to simplify, supposing the grant register contains 101100;

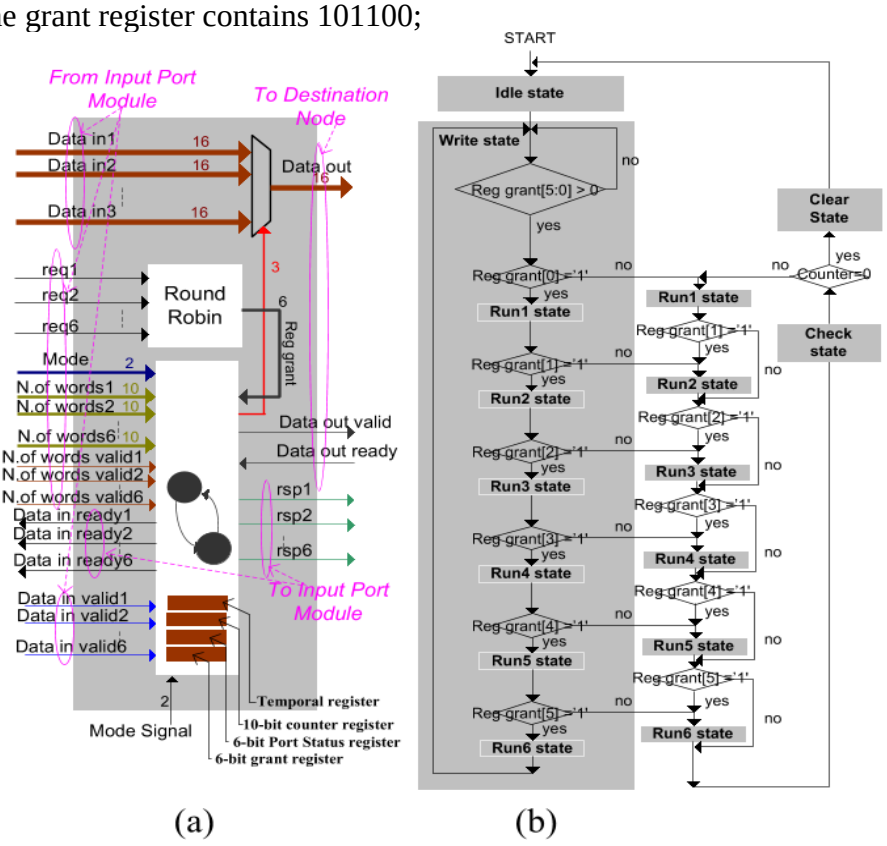
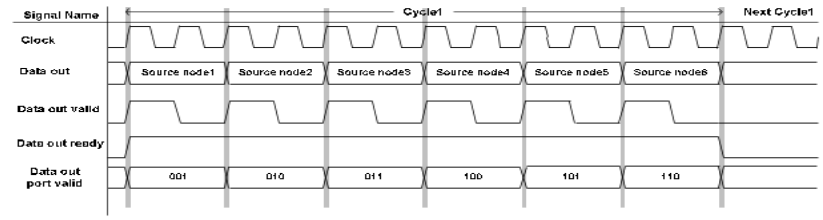
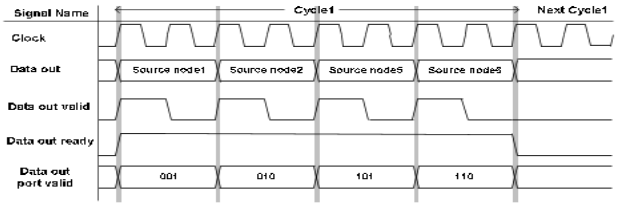



Fig. 5. a) Output Port module Architecture b) Flow control diagram

It implies that the input port modules number 3,4 and 6 will forward their data to the output port module. Therefore, the state will start at the Run3 state, and the Data in ready signals number 3,4 and 6 will be enabled HIGH. Until the Run6 state is executed, the state goes to the Check state. The counter register counting the forwarded packets is checked, whether or not the forwarded packets has been completely sent. Whenever, they have been sent properly this counter register will be "0000000000" and the Clear state will be done. At the Clear state, the rep signals at 3,4 and 6 are enabled LOW and the grant register will be "000000", as well as the state will begin the next cycle.




(a) *6-bit Port Status register* equals to 111111




(b) *6-bit Port Status register* equals to 110011

Fig. 6. Timing diagram a) 6-port to one-port b) 4-port to one-port

## C. Switch Module

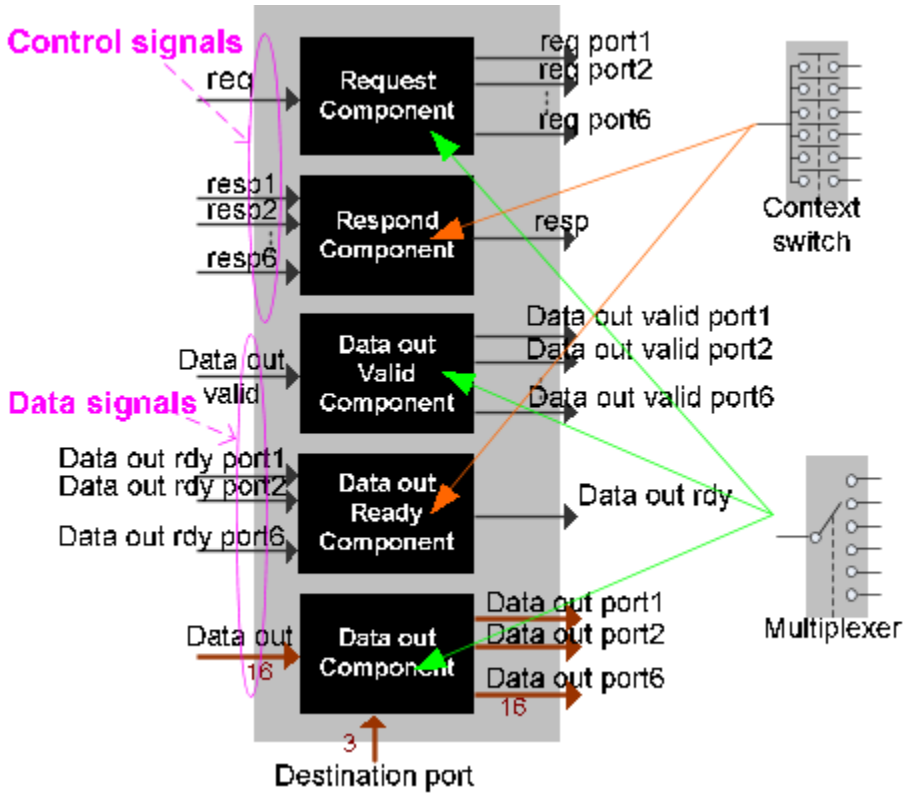



**Fig. 7. Switch module architecture**

Switch module crosses the link between the Input Port module and the Output Port module. Underneath the Switch module, there are five components connecting with two sets of signal. First, the control signal consists of the req signals and the resp signals. Second, the data signal composes of the data out valid signal, the data out rdy port signals and the data out signal. Two types of switch component are context switch and multiplexer. All components of each set can be mapped with two types of switch component as shown in Fig. 7, where they are controlled by the 3-bit Destination port signals of Input Port module.

## III. TEST ENVIRONMENT AND EXPERIMENTAL RESULT

The 6x6 interleaving switch-based crossbar designed and implemented in section 4 is evaluated on test environment as shown in Fig. 8. The Source Nodes and the Destination Nodes connecting with our crossbar can reconfigurable where several IP cores can take place. Before testing, 16-bit counter modules are placed on all Source Nodes, and the available FPGA I/O pins are mapped to all Destination Nodes. The environment is set as following: 1) the system operates at 100 MHz, 2) the interleaving mode is set to default, 3) all 16-bit counters are ready to generate the switching protocol and the 256 packets, 4) TLA5204 logic connects to this environment to measure and capture the 16-bit data out signal and data out valid signal at the Destination node.

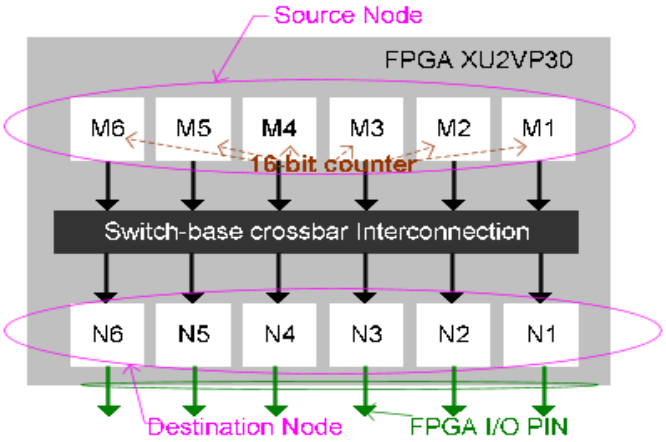



Fig. 8. Test environment on Xilinx FPGA

Since the bandwidth and time period are considered, when more than one Source Nodes require transferring a large number of packets to the same Destination Node simultaneously, the number of the Source Node is increased one at a time in test cases. Fig. 9 show the captured data while one Source Node and six Source Nodes are transferring the 256 packets to the same Destination Nodes with interleaving functionality on our crossbar.

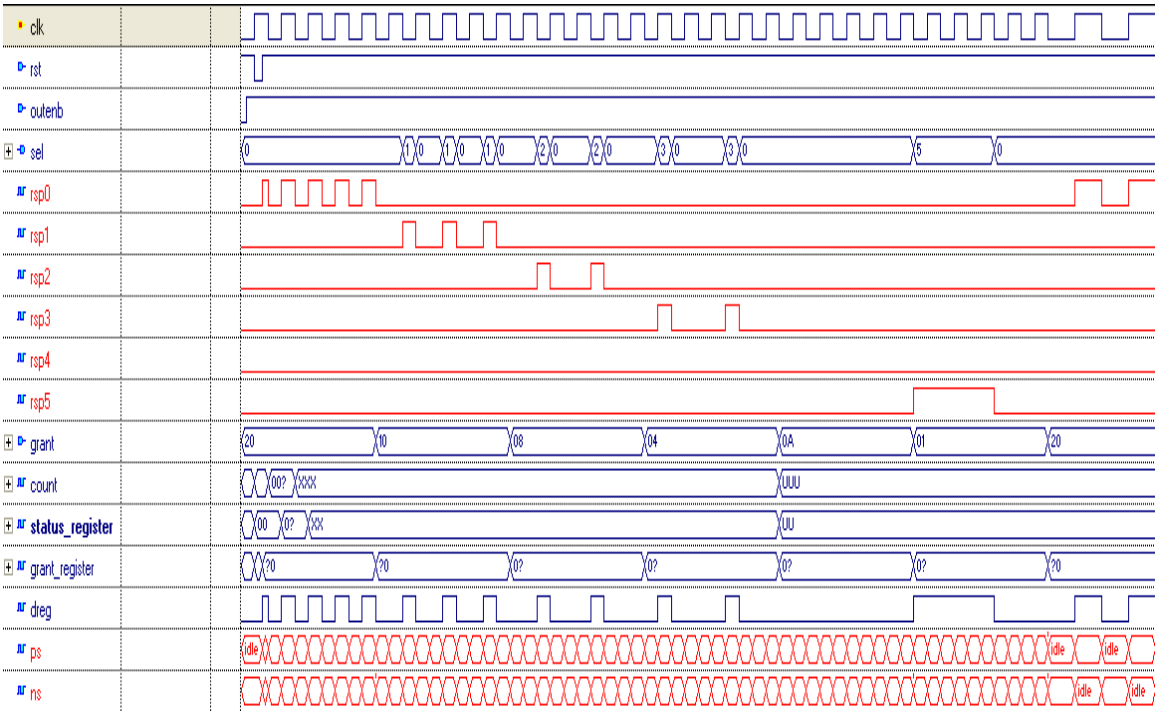



Fig 9: Output Waveform

Band Width is given by the formula

$$BW = Nof\ Packet * bit * Nof\ Inputdata\ /TimePeriod$$

where BW is bandwidth, Nof Packet is the number of packets, bit is data width, N.of Input data is the number of input data, and Time period is the time period of data out signal measured at the Destination Node.

## IV Conclusion

This paper proposes and implements the interleaving mechanism to overcome the output delay (latency) while operating many-to-one and one-to-many data communication. Moreover, resource usage and bandwidth are acceptable when it is compared with the general propose switch-based system and bus-based system.

## V REFERENCES

[1] D. Bafumba-Lokilo, Z. Savaria, J. David , "Generic Crossbar Network on Chip for FPGA MPSoCs", Circuits and Systems and TAISA Conference, 2008, pp 269-272.
[2] Xilinx , "On-chip Peripheral Bus V2.0 with OPB arbiter", "Processor Local Bus(PLB)v4.6(v1.03a)", http://www.xilinx.com.
[3] H. C. d. Freitas, P. Navaux, "On the Design of Reconfigurable Crossbar Switch for Adaptable On-Chip Topologies in Programmable NoC Routers", ACM Special Interest Group on Design Automation, 2009, pp.129-132.
[4] H.Po-Tsang, H. Wei , "2-Level FIFO Architecture Design for Switch Fabrics in Network-on-Chip", Circuits and Systems, ISCAS 2006, Proceedings 2006, IEEE International Symposium on, pp.4863-4866.

[5] M. Hübner, L. Braum,D. Göhringer, J. Becker, "Run-Time Reconfigurable Adaptive Multilayer Network-On-Chip for FPGA-Based systems", IEEE International Symposium on In Parallel and Distributed Processing, 2008, pp. 1-6.
[6] C. Hilton, B. Nelson, "PNoC: a flexible circuit-switched NoC for FPGA based systems", IEE Proceeding Computing Vol. 153, 2006, pp. 181-188.
[7] C. Qiao, R.Melhem, "Reconfiguration with Time Division Multiplexed MIN's for Multiprocessor Communication", IEEE Transactions Parallel and Distributed System, Vol. 5, 1994, pp. 337-352.
[8] . P. Surapong, S. Singhaniyom, and M. Glesner. "An Interleaving Switch-based Crossbar Architecture for MP SoC on FPGA" . In International Conference on ECTICON, pages 188–192, 2010.