I J C T A, 9(17) 2016, pp. 8687-8692

A Comprehensive Digital Cross Connect (DCS) for Buffered and Unbuffered Switching Applications

Gunjan Thakur[*], Mrinal Sarvagya[**] and Preeta Sharan[***]

ABSTRACT

A Digital Cross Connect (DCS) switch is highly popular for on-chip communication as well as telecommunication network due to its simple architecture and non-blocking behavior. The switch architecture not only impacts the performance and efficiency of the overall network but also has an influence on device application. A buffered crossbar switch (shared memory) provides synchronization and retransmission of data at cost of complex architecture. An unbuffered crossbar switch reduces transmission latency and gives simple architecture but does not compensate for loss data. This means that a network configuration must satisfy the requirements for the type of traffic. The existing DCS IP Cores demands lot of customization in order to make it suitable for any application. In this work we have proposed a comprehensive DCS which supports different modes such as buffered (shared memory), non-buffered, shared bus, synchronous and asynchronous on a single platform. The proposed modes of DCS has been simulated using Verilog Hardware Description Language (HDL) in Xilinx ISE9.1i version software. Further, it can be implemented on FPGA which provides a hardware based approach and requires minimum customization.

Keywords: Buffering, customization, non-blocking, IP cores

I. INTRODUCTION

With the advancement in communication network and increase in mobile traffic the demand for high performance switches has immensely raised up. Cross Connect Switches are highly popular for on-chip communication as well as telecommunication network due to its simple architecture and non-blocking behaviour. The basic function of a Digital Cross Connect (DCS) system is to provide a path for the transfer of data or packets from source to destination. The Crossbar schedulers generate control signals to configure the switch matrix and ensures high throughput as well as fairness among various devices sharing a common resource. Actually, the switching architecture not only impacts the performance and efficiency of the entire network, but also has an influence on the device application.

Buffered switch architecture (shared memory) provides synchronization in case of different data rates at the sending and receiving end and allows for retransmission of data at the time of packet loss. At the same time buffering results in reduction of overall bandwidth and increases the system complexity and cost. An unbuffered switch architecture provides higher operating speed by reducing the transmission latency and has a simple design to implement. In this paper we have proposed a comprehensive DCS which supports various modes of operation such as asynchronous, synchronous, buffered, non-buffered and shared bus on a single platform. The proposed modes of the comprehensive DCS has been simulated using Xilinx ISE9.1i version and further can be implemented on FPGA in order to provide a hardware based approach.

[*] Vemana Institute of Technology, Bangalore, Karnataka, India, *E-mail: gunjanmoni@gmail.com*
[**] Reva Institute of Technology & Management, Bangalore, Karnataka, India, *E-mail: mrinalsarvagya@gmail.com*
[***] Oxford College of Engineering, Bangalore, Karnataka, India, *E-mail: sharanpreeta@gmail.com*

II. SURVEY

Previous work shows that a buffered crossbar switch can support integration of unicast and multicast traffic and perform efficiently [1]. In [2-4] authors have proved that buffering at cross points provides desired QoS and 100 % throughput without speed up .Buffered switch results in elimination of input and output contention and gives high scalability [5]. In [6] authors have proved that unbuffered switch reduces transmission latency and provides higher operating speed for NoC. Few researchers has proposed and implemented crossbar switches on FPGA platform [7][8][9] which is considered suitable for SoC network. Therefore, buffering and unbuffering of data is application specific and must be performed as per the traffic requirement. Existing DCS IP cores demands lot of customization in order to fit into any particular application and thus are complex. This shows a requirement to develop a switch architecture which can support both the modes (buffered and unbuffered) of operation. The proposed comprehensive DCS can be switched to different modes depending on the application and requires minimum customization.

III. PROPOSED WORK

The internet traffic can be divided into two major categories: real time traffic (live telecast of video) and non-real time. Real time traffic are highly susceptible to delay but can accept a small amount of data loss. At the same time, non-real-time traffic can allow delay but are quite sensitive to data loss. This means that it is important to know the characteristics of the traffic and selection of switch architecture must fulfill the requirements.

3.1. Buffered Switching Mode (Shared Memory)

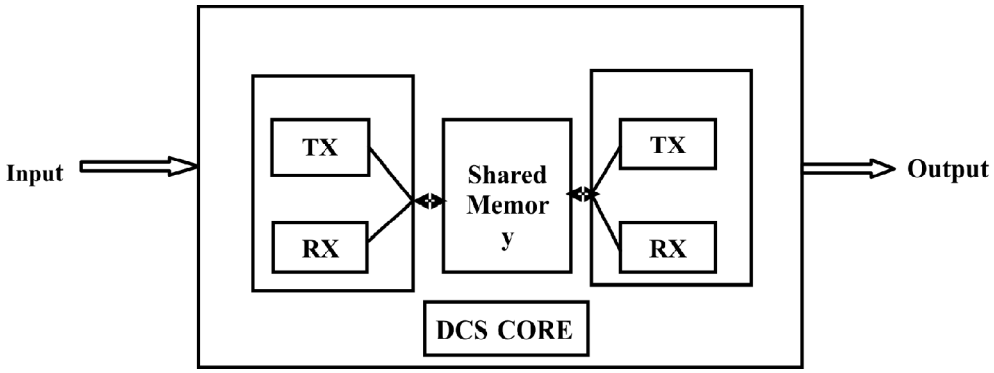

Figure 1: A Buffered switching mode

In this mode a shared memory/buffer is present in between the devices (transmitter and receiver) which will store the data that is being transmitted or received. The shared memory can be a FIFO or DRAM to make the application high speed. Shared memory is required for real time switching in order to provide synchronization for slower devices, such as video conferencing or online video streaming. It provides more reliability by storing the data for retransmission if required.

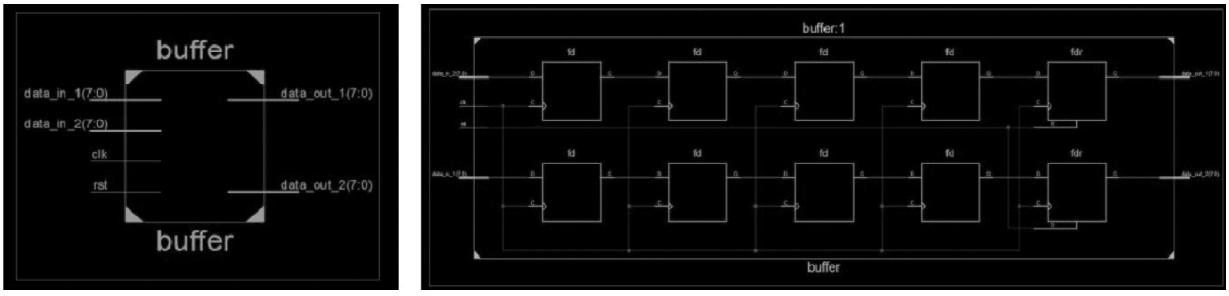

Figure 2: RTL schematic of Buffered switching mode

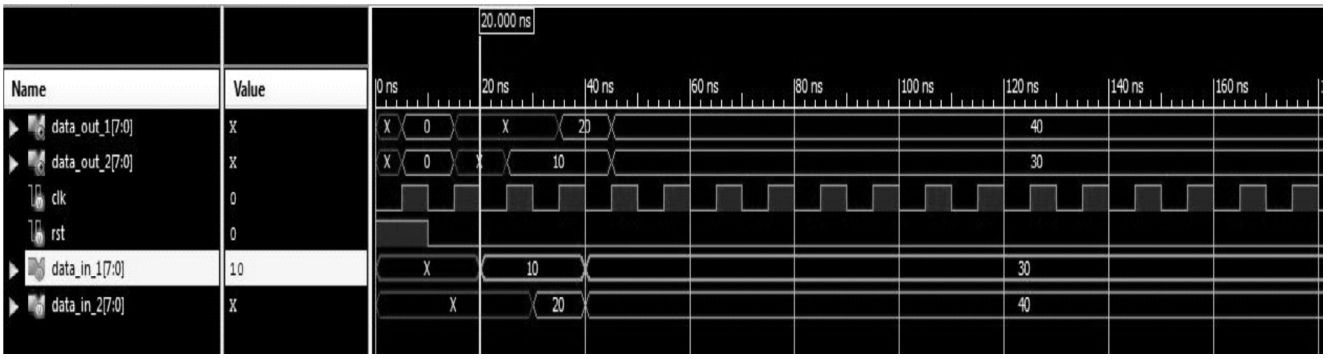

Figure 3: Simulation Waveform for buffered switching mode

Figure 2 and 3 shows the RTL schematic and simulation waveform for shared memory respectively. The waveform shows that the data is stored in memory during the transfer through the cross connect switch which helps to impart the data successfully.

3.2. Non – Buffered Switching Mode

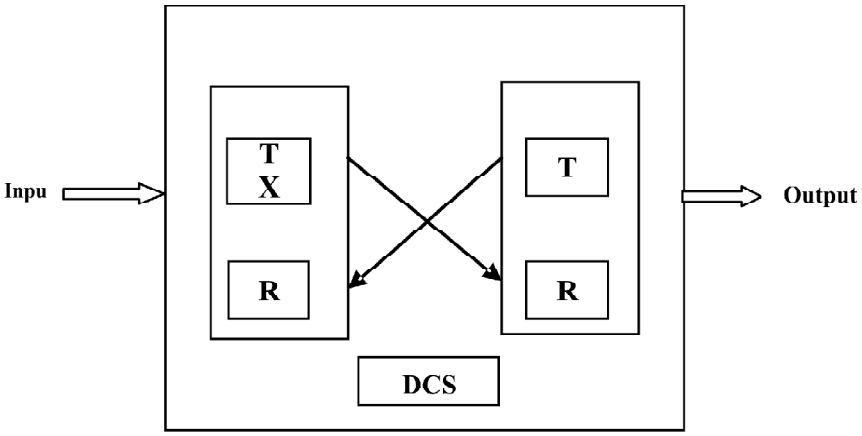

Figure 4: A Non-Buffered Switching mode

The above figure shows a duplex non-buffered DCS core having no buffers at the transmitters and receivers. The advantage here is data transfer/switching will happen at faster rate and hence reduces the latency. A circuit with no buffers is also simple to implement as it eliminates the need to create, manage and destroy the buffers. But there is no record of any data transmitted or received. Hence, a lost packet would not be able to recover.

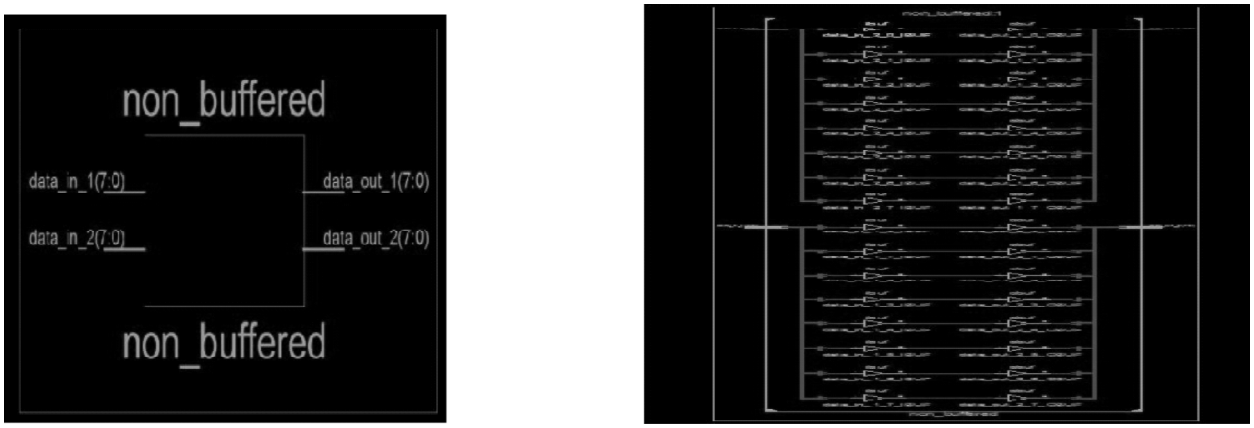

Figure 5: Schematic of Non-Buffered Switch

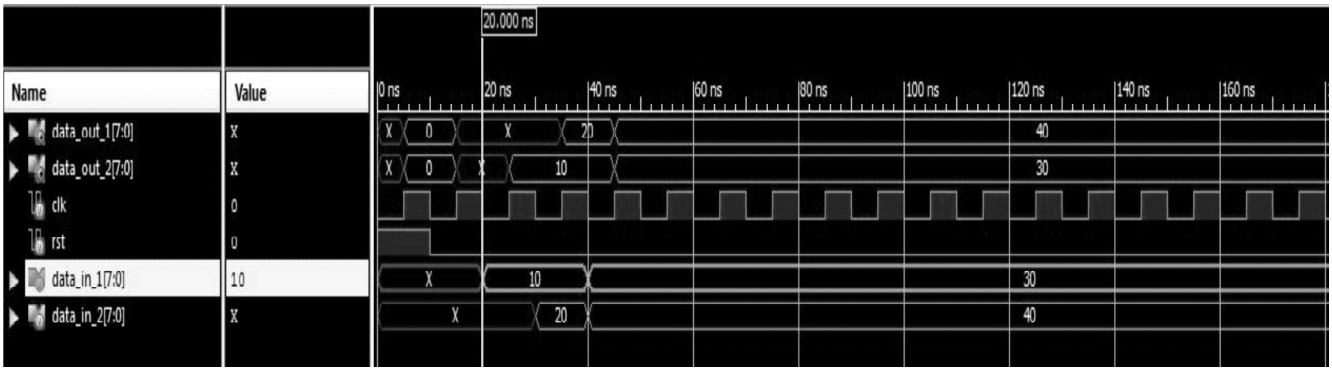

Figure 6: Simulation Waveform of Non-Buffered Switch

Figure 6 and 7 shows the RTL schematic and simulation waveform for Non-Buffered switching mode respectively. In this case the transmitter and receiver does not have any buffer to store the data and the data is lost if the buffers are not free.

3.3. Asynchronous and Synchronous Mode

Synchronous data transmission requires signal timing and both sender and receiver uses the same clock rate to access the data. It has minimum overhead and thus results in higher throughput but possess a complex design. Asynchronous transmission does not require a common clock signal for transfer of data from sender to receiver, rather an agreed pattern of bits are used. Due to requirement of less hardware it is cheap and simple but demands for relatively large overhead.

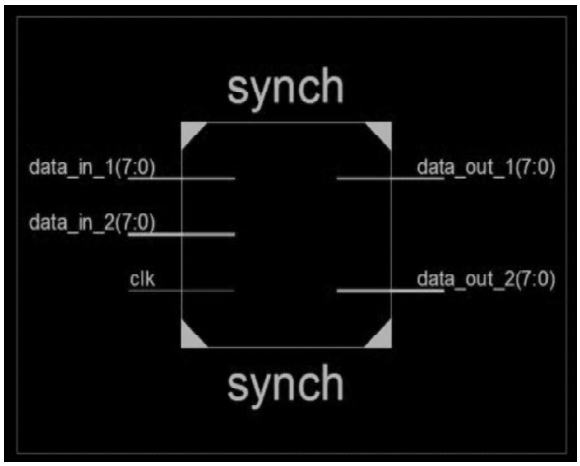

Figure 7: (a) RTLSchematic

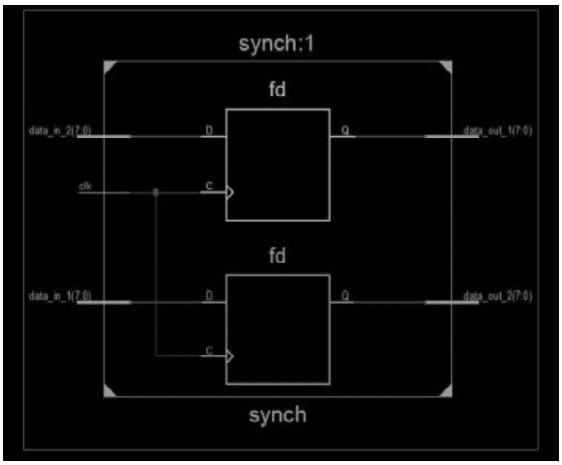

Figure 8: (b) RTL Schematic of Synchronous mode

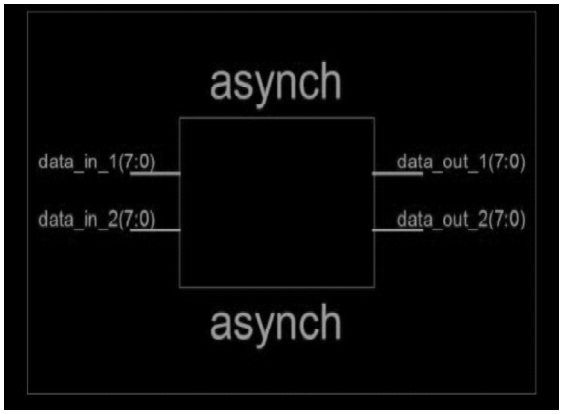

Figure 9: RTL schematic of Asynchronous mode

Figure 8 and 9 shows the RTL schematic for Synchronous and Asynchronous mode. In synchronous all the units work in synchronism with each other and thus the data transfer takes place in correct time slots and chances of glitches are very less.

A Comprehensive Digital Cross Connect (DCS) for Buffered & Unbuffered Switching Application

3.4. Shared Bus

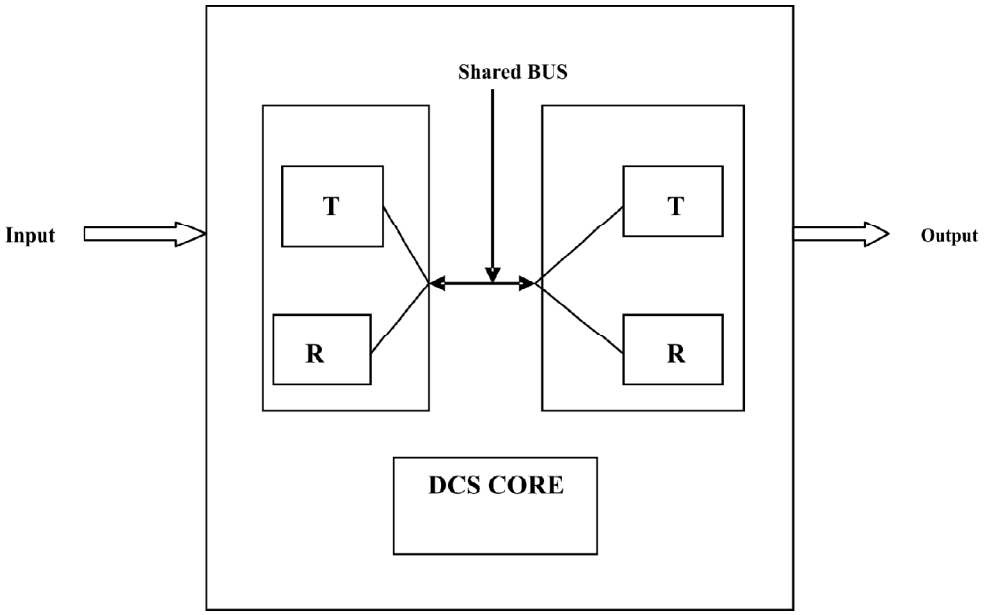

Figure 10: RTL schematic of Asynchronous mode

A switch with a single high-speed bus that is shared by all incoming and outgoing ports on a Time Division Multiplex (TDM) basis. The devices will communicate with each other at various time slots assigned to them before data transfer takes place. It has a simple hardware to implement (no buffers) and performs a non-blocking operation. It is suitable for all Interchip and Intrachip communication within the same equipment.

Figure 11 and 12 shows the RTL schematic and waveform for shared bus respectively. In this case all the line modules in the switch share one data path on the basis of time.

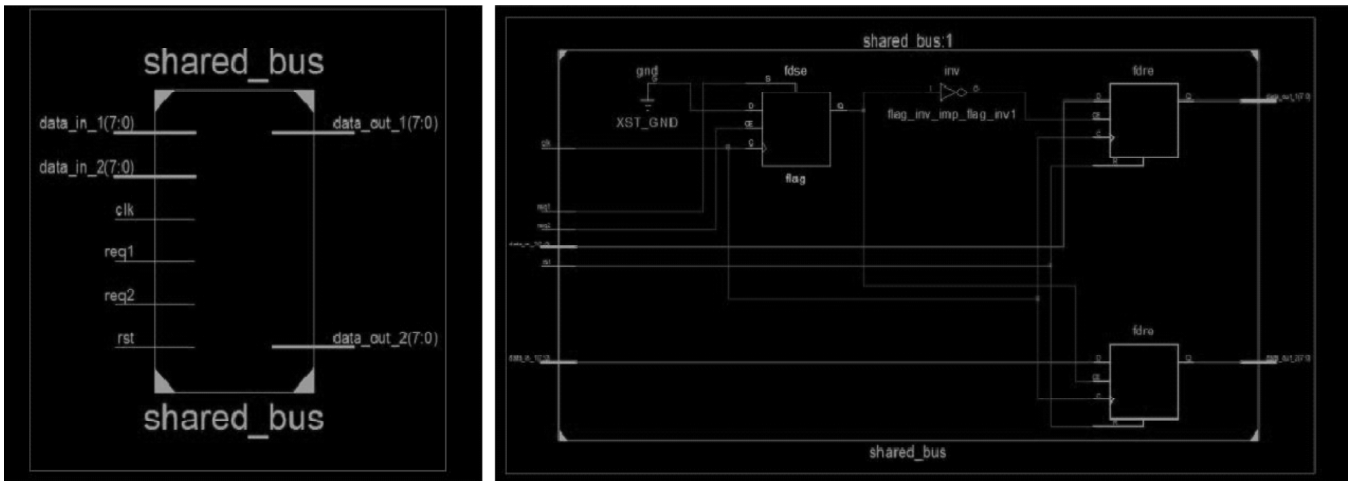

Figure 11: RTL Schematic of Shared Bus

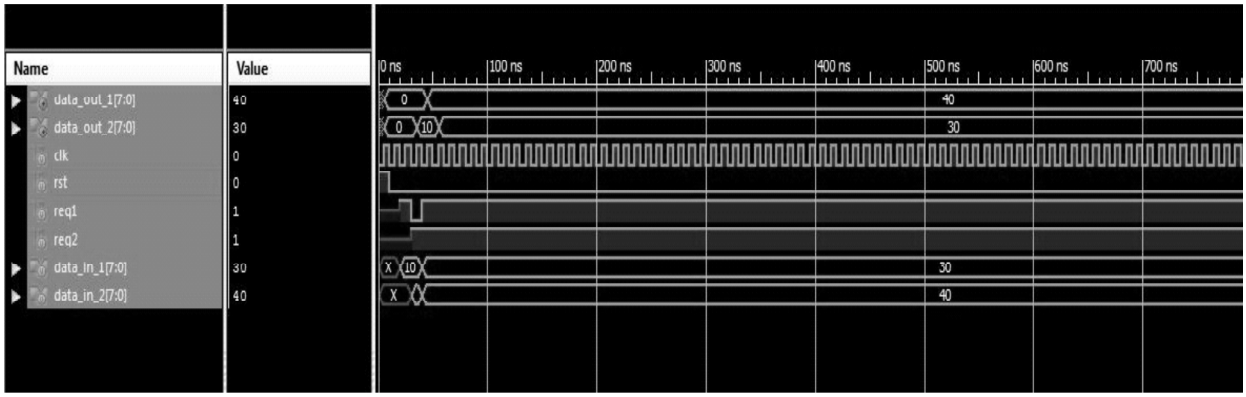

Figure 12: Simulation Waveform of Shared Bus

IV. CONCLUSION

Switch architecture greatly influences the performance and efficiency of the overall network. It must support the various characteristics for different types of traffic. Buffered switch architecture provides synchronization and permits retransmission of data whereas unbuffered switch reduces latency and provides high operating speed. In this work we have proposed and simulated a comprehensive DCS which supports both the modes (buffered and unbuffered) on a single platform. The simulation results shows that the proposed DCS can perform switching of buffering as well as unbuffered data and requires minimum customization to make it suitable for a particular application. The proposed switch can support synchronous & asynchronous mode of operation. In future, the proposed DCS can be implemented on FPGA board to provide a hardware based approach.

REFERENCES

[1] Kai Liu, Jian Yan, Jianhua Lu, Xiang Chen, "Predictive Unicast and Multicast Scheduling in Onboard Buffered Crossbar Switches", *IEEE Communications Society*, Volume.20, Issue.03, March 2016.

[2] Nebojsa Maletic, Soko Divanovic, Milutin Radonjic, "Performance Analysis of Dual Crosspoint Queued Crossbar Packet Switch", *IEEE TELSIKS, Oct 16-19, 2013.*

[3] Yanming Shen, Shivendra S. Panwar, H. Jonathon Chao, " Providing 100% Throughput in a Buffered Crossbar Switch", *IEEE Center for Advanced Technology in Telecommunication, 2007.*

[4] Mingje Lin, Nick McKewon, "The Throughput of a buffered crossbar switch", *IEEE Communications Letter, May 2005.*

[5] Dr. Kannan Balasubramaniam, C.Sindhu, "ISA Independent Scheduling Algorithm for Buffered Crossbar Switch", *International Journal of Advanced Networking and Applications, Volume 3, 2011.*

[6] Leo Hao, Cao Feifei, Lio Dongsbeng, "Design of an Unbuffered switch for Network on Chip", *High Technology Letters, Volume 19, March 2013.*

[7] S. Pongyupinpaniach, S. Singhaniyom, Manfred Glesner, "An Interleaving Switch-based Crossbar Architecture for MP SoC on FPGA", *Institute of Microelectronic System, Germany 2010.*

[8] Jong Arm Jun, Sung Hyuk Byun, Seong Yeob Nam, "A Two Dimensional Scalable Crossbar Matrix Switch Architecture", *IEEE Department of Internet Technology, Korea, 2003.*

[9] Chin Hau Hoo and Akash Kumar, "An Area Efficient Partially Reconfigurable Crossbar Switch with Low Reconfiguration Delay", *IEEE Department of Electrical and Computer Engineering, 2012.*