

Implementation of Optimized router pipeline Stages used for Network on Chip

Yudhishtir U. Bhangale¹, Prof. Neetesh Raghuwanshi², Prof. Jayant P. Bhoge³

^{1,2}Department of Electronics & Communication Engineering

³Department of Electrical Engineering

^{1,2}RKDF Institute of Science and Technology (RKDFIST), Bhopal (MP), India.

³NMKC COET, Jalgaon, Maharashtra, India

Abstract- As the feature size is continuously decreasing and integration density is increasing, interconnections have become a dominating factor in determining the overall quality of a chip. Due to the limited scalability of system bus, it cannot meet the requirement of current System-on-Chip (SoC) implementations where only a limited number of functional units can be supported. Long global wires also cause many design problems, such as routing congestion, noise coupling, and difficult timing closure. Network-on-Chip (NoC) architectures have been proposed to be an alternative to solve the above problems by using a packet-based communication network. In this paper, the Circuit-Switched (CS) Router was designed and analysed the various parameters such as power, timing and area. The CS router has taken more number of cycles to transfer the data from source to destination. So the pipelining concept was implemented by adding registers in the CS router architecture. The proposed architecture increases the speed of operation and reduces the critical path of the circuit. The router has been implemented using VHDL. The parameters area, power and timing were calculated in Xilinx 9.2i simulator and Modelsim tool with default operating voltage of 3V and packet size is 39 bits. Finally power, area and time of these two routers have been analysed and compared.

Keywords- Network-on-chip, XY routing algorithm, Field Programmable Gate Array(FPGA), VHDL, system-on-chip (SoC), Latency.

I. INTRODUCTION

Recently Networks on Chip (NoC) is playing vital role in development in VLSI. Increasing levels of integration resulted in systems with different types of applications, each having its own I/O traffic characteristics. Since the early days of VLSI, communication within the chip dominated the die area and dictated clock speed and power consumption. Using buses is becoming less desirable, especially with the ever growing complexity of single-die multiprocessor systems. As a consequence, the main feature of NoC is the use of networking technology to establish data exchange within the

chip. All links in NoC can be simultaneously used for data transmission, which provides a high level of parallelism and makes it attractive to replace the typical communication architectures like shared buses or point-to-point dedicated wires [1]. Apart from throughput, NoC platform is scalable and has the potential to keep up with the pace of technology advances. NoC network can be modelled as a graph where in nodes, processing elements and edges are the connective links of the processing elements. Figure 1 shows the basic NoC architecture, it basically includes processing element (PE), router. Each PE is attached to NI which connects the PE to a local router. When a packet was sent from a source PE to

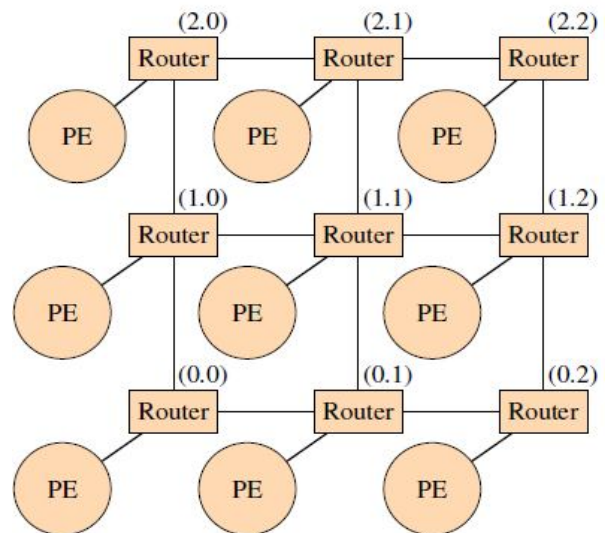


Figure 1: Basic Router architecture with mesh topology

a destination PE, the packet is forwarded hop by hop on the network via the decision made by each router. Like in any other network, router is the most important component for the design of communication back bone of a NoC system. In a packet switched network, the functionality of the router is to forward an incoming packet to the destination resource if it is directly connected to it, or to forward the packet to another router connected to it. It is very important that design of a NoC router should be as simple as possible because

implementation cost increases with an increase in the design complexity of a router [2].

The remaining section of the paper is organized as follows: In section II describe the Literature overview of NOC implementation and related work using xilinx 9.2i. Section III describe Proposed mechanism of NOC implementation. Section IV describe Implemented algorithm and Results, Last Vth section gives overall conclusion of the paper.

II. LITERATURE SURVEY

The author illustrated the impact of repeater insertion on inter-router links with adaptive control and eliminating some of the buffers in the router. The approach saved appreciable amount of power and area without significant degradation in the throughput and latency, though there is still some scope to increase the buffer utilization inside the router [3]. Reference [4] shows work on router for NoC to increase throughput of the network and they introduced the architecture which shows a significant improvement in throughput at the expense of area and power due to extra crossbar and complex arbitration scheme. The throughput increased up to 94% but power consumption is increased by the factor of 1.28 [4].

By utilizing the buffer with bidirectional channels indicated significant improvement in system performance, though in this case, each channel controller will have two additional tasks: dynamically configuring the channel direction and to allocate the channel to one of the routers, sharing the channel. Also, there is a 40% area overhead over the typical NoC router architecture due to double crossbar design and control logic [5]. The author developed an algorithm to optimize size of decoupling buffers in network interfaces. The buffer size is proportional to the maximum difference between the number of words produced and the number of words consumed at any point in time. This approach showed significant improvement in power dissipation and silicon area. The buffer size can be further optimized by considering the idle time of buffer. If some buffer is idle at some time instant, it can share the load of neighbouring input channel and thus increase the utilization of existing resources with small control logic [6].

The author proposed the router architecture with Reliability Aware Virtual Channel (RAVC). In this approach, more memory is allocated to the busy channels and less to the idle channels. This dynamic allocation of storage shows 7.1% and 3.1% latency decrease under uniform and transposes traffic patterns respectively at the expense of complex memory control logic, though this solution is latency efficient

but not area and power efficient [7]. In NoC different switching techniques are used for forwarding information through the network and these techniques have significant effect on the design of router architecture. Switching techniques are broadly categorized into circuit switching and packet switching. Today's NoC designs are based on Packet switching [8].

Packet switching is further categorized into Store and Forward (SAF), Wormhole (WH) and Virtual Cut Through (VCT). But all these techniques face Head-on-Line (HoL) blocking problem, which results from input buffering contention in routers. To overcome the problems in router switching techniques, researchers have proposed various buffering allocation techniques, micro architectural buffer structures, and effective buffer arbitration algorithms. In [9] J. Dally introduced the idea of virtual channel for deadlock-free routing for networks. The most important improvement in switching technique is the introduction of virtual channels (VCs) [10]. Dally and Towels illustrate the basic virtual-channel router architecture [8] and showed that virtual-channel router works in a pipeline to decrease router delay. In [9] authors introduced low latency virtual-channel router in which a single flit can travel through VC router within only one cycle. In [11], authors introduced a low latency router which uses adaptive routing. Virtual channel router is used to solve network deadlock by adopting adaptive routing, and provide guaranteed service and best-effort service in [12]. As Network-on-Chip is strict resource constrained, so a good virtual-channel router should make a better tradeoff between performance and implementation cost.

III. NOC ROUTER ARCHITECTURE

A NoC router consists of number of input ports, a number of output ports, a crossbar switch which connects the input ports to the output ports, and a local port for accessing the Processing Element (PE) connected to this router. In addition to this, router contains a logic block that decides the overall routing strategy for moving data through the NoC. When the data in the form of packet is moved from source to its destination, it is sent on the network based on the routing decision taken by each router. At each router the packet is first collected and stored in buffer then the routing decisions are taken and channel arbitration is made by the control logic. Finally the granted packet crosses through the crossbar and reach to the next router. This process repeats until the packet reaches to its destination. The routing units control logic is a finite state machine (FSM). It processes the packet header to compute an appropriate output channel and generates requests for that output channel accordingly. This NoC router architecture mainly consists of three parts:

- 1) **Virtual Channel:** When a physical channel is divided into a multiple number of logic channels, these logic channels are called as virtual channels. A virtual channel has its own queue, but it shares the bandwidth of the physical channel in a time multiplexed fashion. Virtual channels offer flexibility, better channel utilization and improve network throughput and reduce the effect of blocking shown in figure 2.

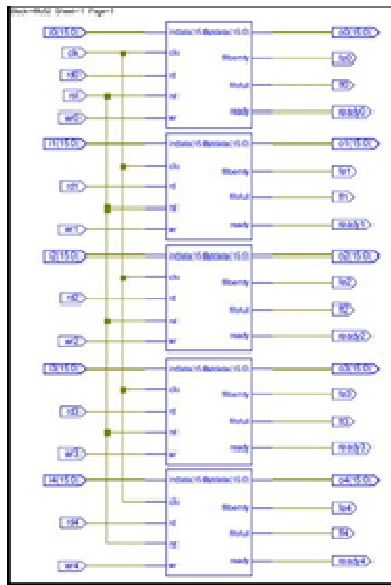


Figure 2. RTL view First in First out.

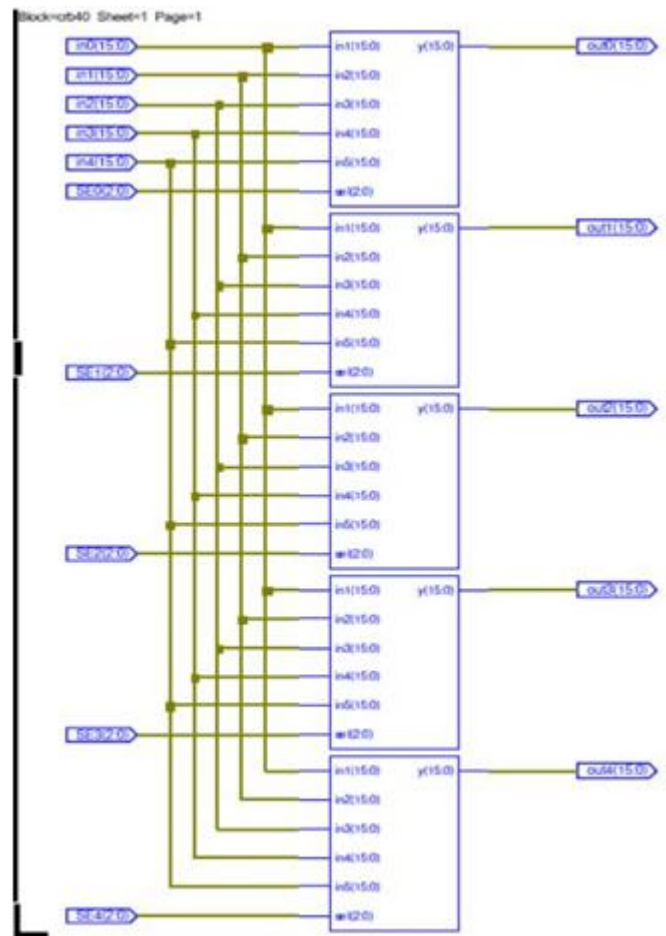


Figure 3. Implimented Crossbar switches

- 2) **Arbiter:** An arbiter is required to determine how the physical channel can be shared amongst many requestors. Here fixed priority arbiter is used. In fixed priority arbiter, each input port has its own fixed priority level. Depending on this priority level, an arbiter grants an active request signal with the highest priority.
- 3) **Crossbar Switch:** The crossbar module in the design is responsible for physically connecting an input port to its destined output port, based on the grant issued by the arbiter as shown in figure 3.

IV. IMPLEMENTATION AND RESULTS

The router architecture has five input ports, five output ports and each input port has four virtual channels with each VC having four flit buffers. The data coming to each input port is stored in virtual channels temporarily. Each input port sends a request to the arbiter to grant access to the crossbar. So, based on the priority level of each input port, arbiter grants access to the crossbar. Then the data traverse through the crossbar and reached to the destination port. The design is implemented in VHDL on structural Register Transfer Level (RTL) as shown in figure 2 and 3 and it is synthesized and simulated using Xilinx ISE Design Suite 9.2i. The router was prototyped in Vertex 5 Device. The simulation result for NoC router is shown in fig. 4 and fig. 5. Here table I shows the comparison of proposed router with Reference Router. The operating frequency of this router is 411.372MHz. Minimum input arrival time before clock and Maximum output required time after clock is estimated as 1.661ns and 3.630ns respectively. The minimum clock period required is 4.748ns.

Table 1. Result Comparison Table

Sr. No.	Design	Conventional	This work
1	Topology	Mesh	Mesh
2	Number of ports	5	5
3	Routing algorithm	Minimal routing	XY
4	Frequency (MHz)	219	411
5	Area (Slice)	1029	393
6	Power estimation (mW)	29	267
7	Performance per port	66ns	2.431ns
8	FPGA device	vertex 5	vertex 5

Device Utilization Summary				
Slice Logic Utilization	Used	Available	Utilization	Notes
Number of Slice Registers	303	19,200	2%	
Number used as Flip Flops	303			
Number of Slice LUTs	920	19,200	4%	
Number used as logic	800	19,200	4%	
Number using O6 output only	800			
Number used as Memory	40	5,120	1%	
Number used as Single Port RAM	40			
Number using O5 and O6	40			
Number used as exclusive route-thru	80			
Number of route-thrus	80	38,400	1%	
Number using O6 output only	77			
Number using O5 and O6	3			
Slice Logic Distribution				
Number of occupied Slices	392	4,800	8%	
Number of LUT Flip Flop pairs used	991			
Number with an unused Flip Flop	598	991	60%	
Number with an unused LUT	42	991	6%	
Number of fully used LUT-FF pairs	331	991	33%	
Number of unique control sets	24			
ID Utilization				
Number of bonded I/Os	182	220	82%	
Specific Feature Utilization				
Number of BUFIO/BUFGCTRLs	1	32	3%	
Number used as BUFIOs	1			
Total equivalent gate count for design	19,047			
Additional JTAG gate count for IOBs	8,736			

Figure 4. Device Utilization Summary

Xilinx XPower - [noc7]					
File	Edit	View	Tools	Window	Help
Ambient Temperature (°C)					25
Junction Temperature (°C)					30.08
Case Temperature (°C)					30.02
Part Type				Commercial	
Airflow (LFM)				0	
Package				ff324	
Total Power (mW)					267.18

Figure 5. Total power Consumption

V. CONCLUSION

The various challenges faced by researchers in SoC design forced them to look for new alternatives which paved the way for Network-on-chip technology. The NoC is a vast and emerging research area that is still in its initial stages. The NoC area has a significant influence in the design of next generation SoC or multicore architectures. In our project we went through the various research aspects of NoC and details

of network topology and routing algorithms were explored. We tried to contribute in the research of NoC by exploring the design space of NoC routers which is a dominant component of the network. The main focus of our current research was aimed at an efficient design of a router for NoC applications. The router is the most important component since it determines various network parameters like latency, throughput and delay. In this project we went through three different router architectures. All the three router designs were of five input and five output port architecture. The designing has been done using the hardware description language VHDL in XILINX ISE tool. Its FPGA implementation is done and its functional model is also verified. After analyzing proposed router architecture performs better than the other. It has constant delay, constant latency, high throughput. Moreover it has concurrent transmission which gives it more flexibility over the other two architectures and it is less error prone.

REFERENCES

- [1] Bouraoui Chemli and Abdelkrim Zitouni, “ Design and Evaluation of Optimized router pipeline stages for Network on Chip, IEEE IPAS16: International Image Processing Applications And Systems Conference 2016, IEEE, pages 1-5, 2016.
- [2] Maha Beheiry, Hassan Mostafa2, Yehea Ismail and Ahmed M. Soliman, “3D-NOCET: A Tool for Implementing 3D-NoCs based on the Direct-Elevator Algorithm, 18th Int’l Symposium on Quality Electronic Design, pages 144-149, 2017.
- [3] Atef Dorai, Virginie Fresse, Abdellatif Mtibaa, El Bay Bourennane, “Backoff Hardware Architecture for Inter FPGA Traffic Management, 2017 International Conference on Advanced Systems and Electric Technologies (ICASET),IEEE, pages 104-019, 2007.
- [4] Adewale Adetomi, Godwin Enemali, and Tughrul Arslan,“Clock Buffers, Nets, and Trees for On Chip Communication: A Novel Network Access Technique in FPGAs, 2017 IEEE International Parallel and Distributed Processing Symposium Workshops, IEEE computer Society, IEEE, Pages 219-222, 2017.
- [5] Jems Rottekowski, Dinna Goheringer,“Data Stream Processing in Net-work on Chip, 2017 IEEE Computer Society Annual Symposium on VLSI, IEEE. Pages 633-638, 2017.
- [6] Nachiket Kapre, “Implementing FPGA overlay NoCs using the Xilinx Ultra Scale memory cascades, IEEE 25th

- Annual International Symposium on Field-Programmable Custom Computing Machines, IEEE, pages 40-47, 2017.
- [7] Nachiket Kapre, “one bit serial noc for FPGA, IEEE 25th Annual International Symposium on Field-Programmable Custom Computing Machines, IEEE, pages 32-39, 2017.
- [8] Khaled E. Ahmed, Mohamed R. Rizk and Mohammed M. Farag, “ Overloaded CDMA Crossbar for Network-On-Chip, IEEE Transactions On Very Large Scale Integration (Vlsi) Systems, IEEE, pages 1-14, 2017.
- [9] Jinglei Huang, Xiaodong Xu and Lan Yao, Song Chen, “ Reconfigurable Topology Synthesis for Application-Specific NoC on Partially Dynamically Reconfigurable FPGAs, SLIP17 June 17, 2017, Austin, TX, USA, IEEE, 2017.
- [10] Hadi Mardani Kamali and Shahin Hessabi, “ AdapNoC: A Fast and Flexible FPGA-based NoC Simulator, IEEE, 2016.
- [11] Huan-Yuan Chen, Shu-Hao Hsu, Wen-Jyi Hwang and Chau-Jern Cheng, “ An Efficient FPGA-Based Parallel Phase Unwrapping Hardware Architecture, IEEE Trans. Computational Imaging, pages 1-13, 2016.
- [12] Kavyashree G S, Saroja S. Bhusare and Sunita Shirahatti, “Architectural Based Congestion Management for Network on Chip implemented on FPGA, IEEE, 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), IEEE, pages 258-263, 2016.
- [13] Chethan Kumar H B, Shubham Agarwal and Nachiket Kapre, “ Deflection Routing for Multi-Level FPGA Overlay NoCs, IEEE, 2016.
- [14] Hung K. Nguyen and Xuan-Tu Tran, “ Design and Implementation of a Hybrid Switching Router for the Reconfigurable Network-on-Chip, 2016 International Conference on Advanced Technologies for Communications (ATC), IEEE, pages 328-333, 2016.
- [15] Yu-Kun Song, Qing-Song Qian and Duo-Li Zhang, “ Design and Implementation of Dual-Port Network on Chip Based on Multi-core System, IEEE, 2016.
- [16] Mayank Kumar, Kishore kumar, Sanjiv kumar gupta and Yogendera Kumar, “ FPGA Based Design of Area Efficient Router Architecture for Network on Chip (NoC), International Conference on Computing, Communication and Automation (ICCCA2016), IEEE, 1600-1605, 2016.
- [17] Serif Yesil, Suleyman Tosun and Ozcan Ozturk, “ FPGA Implementation of a Fault-Tolerant Application-Specific NoC Design, 11th International conference on design and technology for Nano-scale era(DTIS), IEEE, 2016.
- [18] Geethu Jayan and Pavitha P. P., “ FPGA Implementation of an Efficient Router Architecture Based on DMC, 2016 International Conference on Emerging Technological Trends [ICETT], IEEE, 2016.
- [19] Sukhbani Kaur viridi, Shushant Shekhar, gaurav Varma Shekhar Mahesh-wari, “Implimentation of crossbar switch for NOC on FPGA, International Conference on Computing for Sustainable Global Development (INDIA-Com), IEEE, pages 2087-2091, 2016.
- [20] Siddhartha, Nachiket Kapre, “ eBSP: Managing NoC traffic for BSP workloads on the 16-core Adapteva Epiphany-III Processor, 2017 Design, Automation and Test in Europe (DATE), IEEE, pages 73-78, 2017.
- [21] Khaled E. Ahmed, Mohamed R. Rizk and Mohammed M. Farag, “ Overloaded CDMA Interconnect for Network-on-Chip (OCNoC), 2016 IEEE.
- [22] Taimour Wehbe and Xiaofang Wang, “ Secure and Dependable NoC-Connected Systems on an FPGA Chip, IEEE Transactions On Reliability, IEEE, Pages 1-12, 2016.
- [23] Otavio A. de Lima Jr., Otavio A. de Lima Jr. and Frederic Rousseau, “ A survey of NoC evaluation platforms on FPGAs, IEEE, pages 1-6, 2016.
- [24] Helio Fernandes da Cunha Junior, Bruno de Abreu Silva and Van-derlei Bonato, “ Parameterizable Ethernet Network-on-Chip Architecture on FPGA, Euromicro Conference on Digital System Design, conference publishing services, IEEE, pages 263-266, 2015.
- [25] Yao Chen, Swathi T. Gurumani, Yun Liang, Guofeng Li, Donghui Guo, Kyle Rupnow and Deming Chen, “ FCUDA-NoC: A Scalable and Efficient Network-on-Chip Implementation for the CUDA-to-FPGA Flow, IEEE Transactions On Very Large Scale Integration (Vlsi) Systems, IEEE, pages 1-14, 2015.
- [26] Amit Bhanwala, Mayank Kumar and Mayank Kumar, “ FPGA based Design of Low Power Reconfigurable

Router for Network on Chip (NoC), International Conference on Computing, Communication and Automation (ICCCA2015), IEEE, pages 1320-1326, 2015.

[27] Manel Langar, Riad Bourguiba, Jaouhar Mouine, “ Design and Implementation of Enhanced on chip Router, 12th International Multi Conference on Systems, Signals & Devices, IEEE, pages 1-4, 2015.

[28] Zalak Dave, Shivank Dhote, Jonathan Joshi, Abhay Tambe and Sachin Gengaje. “ Network on Chip Based Multi-function Image Processing System using FPGA, 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, Pages 488-492, 2015.