ISSN 2395-1621

FPGA based Efficient Routing Implementation of programmable Network on chip

[#1]Prof A.A Trikolikar, [#2]Shantanu Khandke, [#3]Abhishek Dandge, [#4]Tejas Pandit

[2]shantanuk8@gmail.com

[#1]Prof. Department of Electronics and Telecommunication
[#234]Department of Electronics and Telecommunication

JSPM's Imperial College Of Engineering. And Research, Pune.

ABSTRACT

More and more complex and larger system on chips are getting developed as a result of increase in chip density following Moore's law. Advanced So-Cs have in their shelf significantly noticeable communication mechanisms. No-C has solved the scalability problems to a larger extent compared to bus based interconnect. No-C has been providing a back bone infrastructure for System-on-chips since long. Among the available communication matrices No-C has helped tremendously in the communication performance enhancement. No-C research majorly involves work on packet switching, though at the same time circuit switching assures high communication rates and predictable communication latencies. The current research is targeted towards implementation of No-C architecture for FPGA based designs using circuit switching approach. Proposed implementation is termed as enhancement of light weight circuit switched architecture. The programmable No-C architecture is being implemented using VHDL and synthesized on the Virtex-5 XC5VLX20T package FF323 device at 139 M-Hz. It provides sufficient customization on the number of ports, nodes and amount of data. Performance improvement over the existing implementations has been validated by the experimental synthesized results.

Keywords :

ARTICLE INFO

Article History

Received 21st March 2016

Received in revised form :

23rd March 2016

Accepted : 25th March 2016

Published online :

28th March 2016

I. INTRODUCTION

The implementation of very large systems on a single chip is termed as System on chip (SoC). These architectures generally consist of combination of CPUs, memories and custom hardware models. SoCs could also be put into practice on Programmable Logic Devices (PLDs), e.g. Field Programmable Gate Arrays (FPGA) or Complex Programmable Logic Device (CPLD).The emergence of programmable SoCs have made significant contribution in on chip system world because of two major reasons mentioned under: At first bus based communication creates scalability issues with increasing system complexity and could prove to be an important bottleneck, thus giving rise of the adoption of Network on chip architecture. NoCs have successfully addressed the problem of SoC scalability. The concept of programmability among NoC has been taken into consideration by architecting them for the varied applications taking different shapes for different applications at various times.Second issue deals with design and verification, which again becomes very important to deal with at the level of higher complexities. FPGA based communication could become a very viable solution in this case, as design and verification could be repeated n number of times for any design idea within FPGA. A computing architecture for FPGAs has greatly simplified application development. It should abstract away the differences between FPGA devices, while supporting communications with external devices at full interface speed, consuming as few resources as possible, and allowing the application to determine how data should be handled. FPGAs have since long been shown to be effective and efficient for performing a variety of computations . But at the same time the high non recurring engineering costs and long time to market for ASICs, clearly indicates more better use of FPGAs for designing different hardware applications .

In the proposed work using another HDL, that is, VHSIC Hardware Description Language (VHDL), the implementation of Programmable NoC architecture is being carried out. There is not much support from the EDA

(Electronic Design Automation) industry for asynchronous systems. Thus, researchers have combined the ideas of synchronous and asynchronous designs. One such strategy is GALS (globally asynchronous and locally synchronous) solution. GALS divides a system into smaller, locally decoupled synchronous regions and then composes a few of them to yield a localized subsystem. These synchronous regions and subsystems would be easier to integrate into a global solution and verify. There will be an asynchronous way in which all the local synchronous regions will communicate at the system level.

Therefore, these different synchronous regions need not have to be synchronized to a single global clock. This approach will reduce the requirement for chip-wide clock trees; the designers could focus on local synchronous regions only, which would be far less complex than thecomplete system. Since one has the flexibility to reduce the clock speed of a given synchronous region (or node) independent of other such regions, the amount of power consumption in a system can be managed better and reduced. One GALS solution is NOC (Network-on-Chip) . NOC can improve design productivity by supporting modularity and reuse of complex cores. Thus, it enables a higher level of abstraction in the architectural modelling of future systems.

II. LITERATURE SURVEY

[1] A DfT Architecture for Asynchronous Networks-on-Chip. This paper is published by Xuan Tu Tran, Jean Durupt, Fran¸cois Bertrand in MAY 2006, they studied on the bases of The Networks-on-Chip (NoCs) paradigm is emerging as a solution for the communication of SoCs. Many NoC architecture propositions are presented but few works on testing these network architectures.

[2] Efficient Routing Implementation of Programmable Network on Chip on FPGA using Circuit Switching Approach , IJETAE This paper is published by Parag Parandkar , Purnima Khandelwal , Geetesh Kwatra in Volume 5, Issue 1, January 2015, the study made by them was Advanced SoCs have in their shelf significantly noticeable communication mechanisms. NoC has solved the scalability problems to a larger extent compared to bus based interconnect. NoC has been providing a back bone infrastructure for System-on-chips since long.

[3] P-NoC: a flexible circuit-switched No-C for FPGA-based systems. This paper was published by C. Hilton and B. Nelson in 31st October 2005 and revised in 8th February 2006 they studied on the bases of Such So-Cs can also be implemented on FPGA substrates, something we will refer to as programmable So-Cs (P-SoCs), in this paper.

[4] Journal Of Engineering , Computing and Architecture ISSN 1934-7179 Survey of Network on Chip (NoC) Architectures & Contributions. This paper was published by Ankur Agarwa, Boca Raton in Volume 3, Issue 1, 2009,they studied on the bases of In this paper, we have summarized over sixty research papers and contributions in NOC area.

[5] Efficient FPGA Based Bidirectional Network on Chip Router through Virtual Channel Regulator Special Issue: Proceedings of 2nd International Conference on Emerging Trends in Engineering and Management, ICETEM 2013. This paper was published by Mr. Ashish Khodwe, Prof. C. N. Bhoyar, in 2013 they studied on the bases of In this study, we analyse the move towards Networks-on-Chips router from an area and power perspective by accurately modeling a Bidirectional Network-on-chip router through VirtualChannel Regulator in FPGA. Accurate speed, area and power metrics are also reported for the networks router, which will allow a more complete comparison to be made across the NoC architectural router space considered. The proposed architecture of BiNoC router is simulated in Xilinx ISE 9.1i software.

[6] (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 4, No.9 A Survey of Network-On-Chip ToolsThis paper was published by Ahmed Ben Achballah Slim Ben Saoud in 2013 they studied on the bases of The literature contains many relevant studies and surveys discussing NoC proposals and contributions. However, few of them have discussed or proposed a comparative study of NoC tools. The objective of this work is to establish a reliable survey about available design, simulation or implementation NoC tools. We collected an important amount of information and characteristics about NoC dedicated tools that we will present throughout this survey. This study is built around a respectable amount of references and we hope it will help scientists.

III. PROBLEM STATEMENT

Implement a router for No-C in VHDL based upon the references found. Design and implement interface unit between No-C routers and process units. Design and implement a No-C based system connected to several processing units. No-C architecture for FPGA based design using circuit switching approach.

IV. PROPOSED SYSTEM

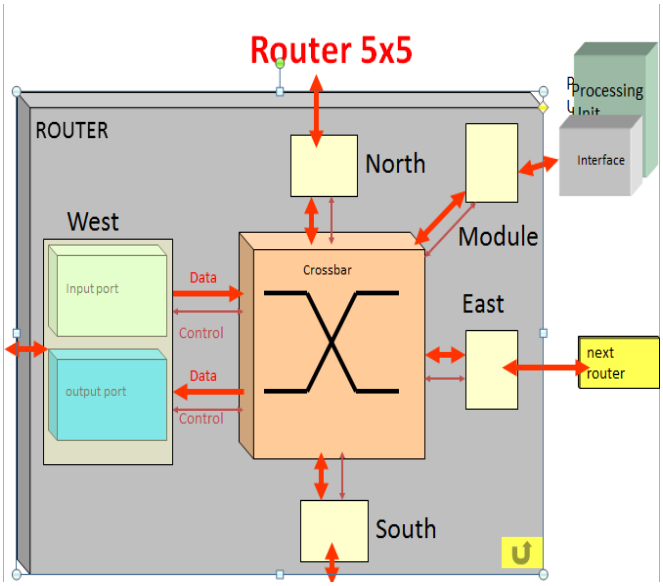

Fig 1. Block diagram

Description:

The router has five bidirectional interfaces where four of them are connected to the neighbor routers through the physical links and the fifth one is connected to the IP core through the Network Adapter (NA).The proposed router is designed such that when a flit enters the router it can only be routed to the output ports of the others four interfaces and

cannot be routed back in the same direction. The routing direction is encoded in the header flits using the two MSB as listed in Table I. The connection between the input and output ports is performed through a non-blocking crossbar and hence the router can accept any number of simultaneous inputs.

Direction	North	East	South	West	Local
Direction encoding	00	01	10	11	Route back

Table 1 Direction Bit

- **Proposed Router Input Port**

The input port of the proposed router design comprises of four designs stages and these include,

FIFO Buffer Stage

First in First out (FIFO) buffer which provides the router interface to the other NoC components. When there is more than one input request to the router simultaneously then FIFO buffer used to store information. To ensure that the latch will not accept new flits before the end of the handshake cycle, the four control signals of the latch are generated using the acknowledge signal from the output handshake channels together with the output of the AND gate. These four signals form inputs to an OR gate whose output control the latch circuit. The buffer will only latch new data after the complete handshake cycle is finished which is indicated by the falling edge of the acknowledge signal of the FIFO output port. Fig. 3.1. shows the final design of the FIFO stage.

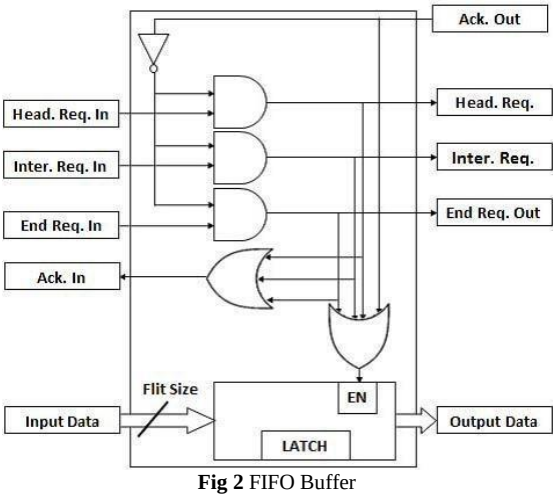

Fig 2 FIFO Buffer

The Address Latching Stage

Address latching stage is used to control data routing for that four output De-Multiplexer are used and to select output two select lines are required which are taken from the two MSB of the header flit and required route directions are extracted as shown in figure IV.

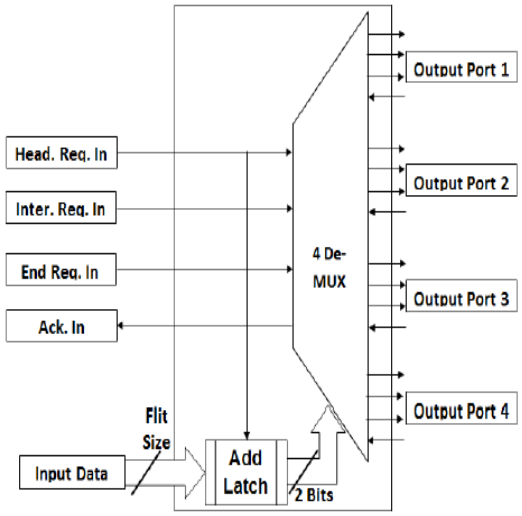

Fig.3 The input port address latching and De-MUX stage

Input Data Manipulation Stage

Depending on the type of the received flits, this stage decides whether to do some manipulation to the received data or passing the flits directly to their destination. The input data manipulator is designed such that it is activated at the arriving of new flit to the FIFO buffer output. The arriving flits are manipulated according to their types. In case a header flit is detected by this stage, then according to the next router address is stored in its two MSB.

Output De-Multiplexing Stage

Three de-multiplexers are designed to handle the handshake request signals which need to be directed to the appropriate output port along with its corresponding flit. Each de-multiplexer is connected to the output port of the other four channels of the router. The same design is used for the acknowledge signals where all the four coming signals from the output interfaces of the output port are connected to the input of a 4-1 multiplexer. The output of this multiplexer is connected to the acknowledge signal at the input interface of the input port. Figure IV shows De-Multiplexing stage.

- **Proposed Router Output Port**

The output port of the proposed router design comprises of three design stages.

Arbitration Stage

This stage provides priority if more than one signals are detected at the same time. In the case of two or more header request signals are asserted simultaneously, the proposed designed arbitrated between them and give access to one signal only.

4-Inputs 1-Output Handshake Channel Component

As the previous stage grants access to only one channel, this component is designed to have four mutual exclusive input channels and one output channel. A multiplexer is used to select the output flit from the four input data channels. The multiplexer control signal is designed in such a way to certify that the selected flit data will be valid for the complete handshake cycle. Figure V shows the design of the component.

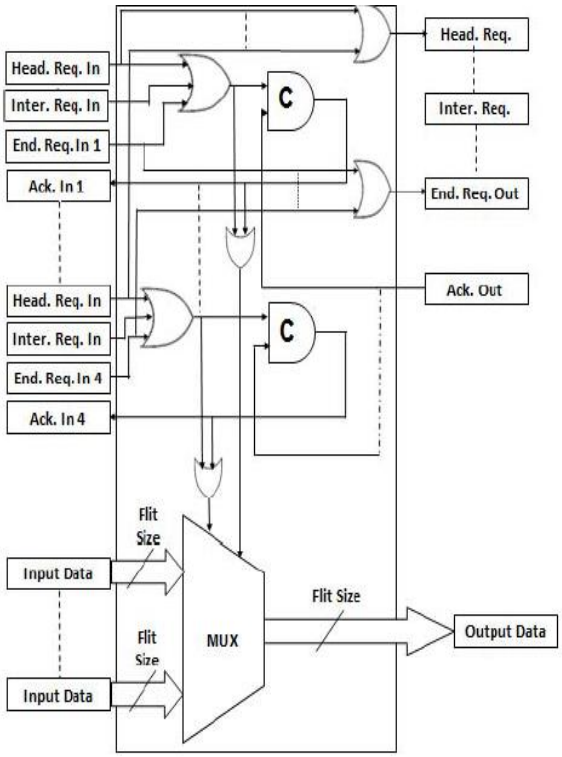

Fig. 4 Output Port

4-Phase Bundled Data Handshake Protocol Merge

This is the last design of the proposed asynchronous router output port that has four mutual exclusive input channels and one output channel. Here, output flit is selected by using four input multiplexer.

V. CONCLUSION

Flexible light weight circuit switched approach is depicted for FPGA based systems. The work has been shown to be delivering better results in terms of clock frequency improvisation compared to the earlier implementations. The Programmable NoC architecture is implemented using VHDL on the Virtex-5 XC5VLX20T package FF323 device at 139 MHz. It provides sufficient customization on the number of ports, nodes and amount of data. Performance improvement over the existing implementations has been validated by the experimental synthesis results. This encourages current design to be further modified and enhanced using higher level Hardware languages working on system level, like that of SystemC and System Verilog.

REFERENCES

[1] Mushtaq U., Hasan, O. ; Awwad, F.; PNOC: Implementation on Verilog for FPG, 9th IEEE International Conference on Innovations in Information Technology (IIT), 17-19 March, 2013, Abudhabi, pp. 148-151 10.1109/Innovations.2013.6544409.

[2] Shuai Che, Jie Li, Jeremy W. Sheaffer, Kevin Skadron, and John Lach. Compute-Intensive Applications with GPUs and FPGAs in Proceedings of the 2008 Symposium on Application Specific Processors, SASP ‘08, pages 101 – 107. IEEE Computer Society, 2008.

[3] Eric S. Chung, Peter A. Milder, James C. Hoe, and Ken Mai. Single- Chip Heterogeneous Computing : Does the Future Include Custom Logic , FPGAs , and GPGPUs?, International Symposium on Microarchitecture (MICRO-43), Atlanta, GA, 2010, pages 225 – 236, 2010.

[4]David Barrie Thomas, Lee Howes, and Wayne Luk. A Comparison of CPUs, GPUs, FPGAs, and Massively Parallel Processor Arrays or Random Number Generation in Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays, FPGA ‘09, pages 63–72. ACM, 2009.

[5] Jeremy Fowers, Greg Brown, Patrick Cooke, and Greg Stitt. A Performance and Energy Comparison of FPGAs, GPUs, and Multicores for Sliding-Window Applications in Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays, FPGA ‘12, pages 47–56. ACM, 2012.

[6] Altera. Altera Annual Report (2012 Form 10-K). http://www:sec:gov/Archives/edgar/data/768251/000076825 113 000008/altera10k12312012:htm.

[7] A.Agarwal and R. Shankar, A layered architecture for NOC design methodology, IASTED International Conference on Parallel and Distributed Computing and Systems, pp. 659-666, 2005.

[8] L. Benini and G. De Micheli, Networks on Chip: a New SoC Paradigm, IEEE Computer, volume1, pp. 70 – 78, 2002.

[9] C.Hilton and B. Nelson, PNoC: A flexible circuit switched NoC for FPGA-based systems, IEE proceedings computers and digital techniques, volume 153 Issue 3, 2006.

[10] E. Salminen, V. Lahtinen, K. Kuusilinna, and T. Hamalainen, Overview of bus-based system-on-chip interconnections, in Proceedings of the IEEE International Symposium on Circuits and Systems.ISCAS 02, , pp. 372 – 375 vol.2, 2002

