International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-1S3, December 2019

Architectural Enhancement of Network on Chip

Senthil Athiban M, Mehazin Shaju, M M Sravani, S Ananiah Durai

Abstract: ***This paper gives a new architectural design suggestion of NoC, with efficient way of communication. Firstly, to create a serial data communication architecture in competence with the existing widely used parallel form of data transmission and reception [1]. Secondly to enable simultaneous transmission and reception between more than one module at the same time. Thirdly to create the architecture that is modifiable as per the need of user. The theoretical data rate calculated was 300 MBps. The throughput we achieved after the completion is 250MBps.***

Keywords : Architecture, NoC, SoC, Topology.

I. INTRODUCTION

In today's world there is an ever-increasing needed for system on chip with good power efficiency and high processing power. Many of the current technology's inbuilt modules interact using various bus topology. A bus might be a solution for smaller SoCs, but for more significant modules, we need to come up with a new way of communication between the modules which are operating at different speeds. Also, at high frequency parallel communication is difficult as the cross talk between wires increases increasing signal loss [2]. Our architecture can support multiple modules communicating in different frequencies and data rates with less hardware requirements.

II. EXISTING TECHNOLOGY

We found a few research papers that validated our idea for the project and also helped to modify and bring about the changes from our initial plan. From the paper few conclusions were drawn like [3], firstly application will not be singled out and there should be more provision for multiple applications to run at the same time. Secondly devices will be multifunctional. They should be independent modules/devices that need not wait or hold for other resource and finally the NoC should be able to handle multiple data rates and frequency of data transmission.

The available architecture of NoC mostly followed parallel mode of data communication and the topology of NoC architecture was mesh and bus topology.

Manuscript published on 30 December 2019.
* Correspondence Author (s)
Senthil Athiban, Pursuing Master of Science, Engineering Electrical Engineering Specializing, Electronics and Mixed Signal Circuits In Arizona State University,
Mehazin Shaju Research and Development Engineer, Continental, Bengaluru, India.
Sravini Pursuing Ph.D Under Dr.S Ananiah Durai, Vellore Institute of Technology, India.
Dr.S Ananiah Durai, Associate Professor, School of Electronics Engineering, Vellore Institute of Technology, Chennai Campus, India.

The papers suggested that there was a need of much user-friendly platform was multiple devices of different data rates could transfer easily. Another paper by Zhibin Yu Institute of computer, Xidian University, China was profoundly discussing the switching architecture in SoCKit [5]. Most of the NoC architecture was inhibited with crossbar switching methodology.

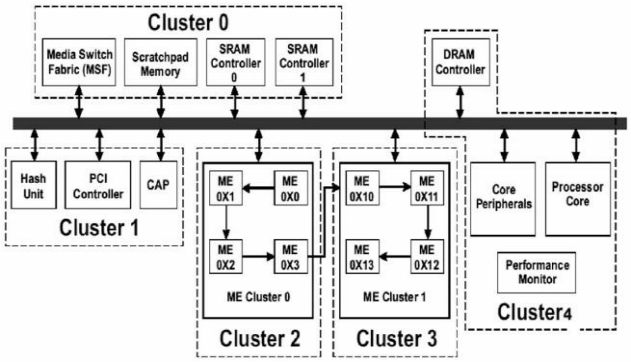

Fig.1. Functional Block Diagram of Typical Network Processor [4]

Crossbar switches are synthesizable at the time of manufacturing for the number of modules that we need to attach to then NoC [6]. After the synthesis these switches are not re-scalable from an outer level. Also, these switches are more complex and matrix matched switching that will not allow simultaneous switching.

This was an issue that we wanted to modify and change. Crossbar switch increases quadratically in complexity as the number modules it is connecting increases [5], [7], [8] & [9]. After comparing the possible methodology used in this paper, we understood the few drawbacks of crossbar switch and its complex architecture to design ourselves. Fig.1 Shows the traditional circuit switching name it as crossbar switch.

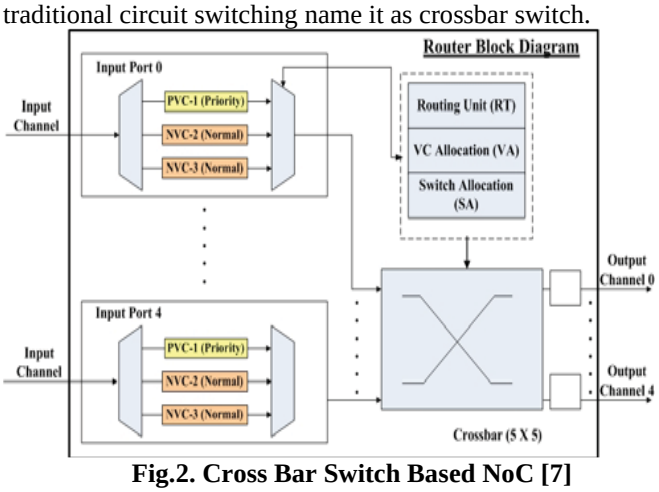

Fig.2. Cross Bar Switch Based NoC [7]

Another factor in regard with crossbar was the slow transfer of data as only one channel for communication will be active. Moreover, bus topology with linear resource and data communication sometimes get lagged due to non-empty channels associated with the modules.

III. CURRENT TECHNOLOGY

The current on-chip communication is mostly shared bus architecture or simple bus architecture. Though bus topology is fast for processing data, but for communication bus topology is resource intensive. For example, ARM based processors use APB and ASB/AHB bus architecture for their communication in conjunction with a peripheral bridge. Though ARM processors [10] are powerful and efficient the bus topology demands a huge of resource and doesn't support simultaneous communication. Another example is in the Intel processors use bus extensively for communication with other modules. But for co-processors like graphics processing unit or network cards are connected to PCI express links, which communicate serially. This communication is often converted to parallel communication using the Memory Controller Bus (MCH). MCH was not able to manage the high data rates of PCIe, so Intel developed a new serial architecture for its communication known as Intel Quick Path interconnect. This made the communication between the core and its peripheral mode faster thereby increasing the processing power.

IV. METHDOLOGY

In the proposed methodology packet switching algorithm are used instead of traditional circuit switching. Fig 3 shows the packet switching algorithm. So, each module can communicate at any time without bothering to place a request for communication. Each module will send the data packets to the network controller which then forwards it to the necessary module.

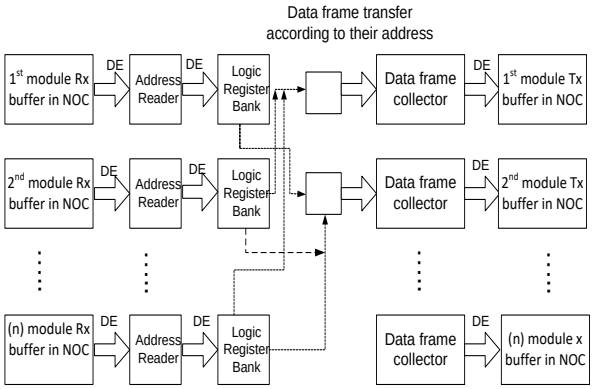

Fig.3. Packet Switching Algorithm

This architecture is highly scalable since the designer can add any number of modules to the NoC, or create different architecture as we have suggested below. Since the communication is happening between a finite number of modules, we don't use dynamic addressing as at the end of a chip design the number of modules is constant. Our NoC architecture can allow a 0.2 GB/s data transfer rate between any 2 modules with ideal conditions.

V. DESIGN OF NOC ARCHITECTURE

The design of NoC and its sub modules are customizable that is the data flow rate and the number of modules can be modified. Cross clock domains during communication is handled by the NoC itself. The design of transmitter module has flags and dedicated up and down channels.Fig.4 shows the transmitter module of the design.

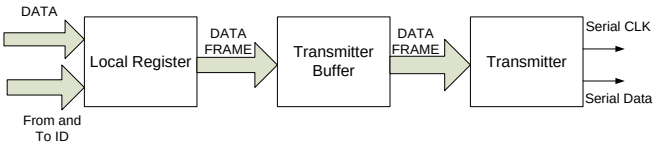

Fig.4. The Transmitter Module

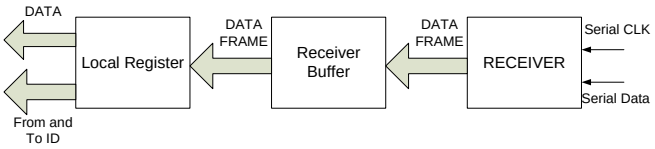

Fig.5. The Receiver Module

The dedicated links acts as two-way communication channels so that there is no need for waiting for resources or any delay caused due to the completion of one process. This increases the efficiency of data packet transfer. The entire communication can be more specifically described as hand shake and acknowledgement authorized or dependent. The request signals to and fro from the modules and the acknowledgement send to and fro through the NoC and modules will be the communication initiation controls.

Flags with the module will act as an indicator for giving the pre-process initiation signals about the channel status and ongoing communication process if any with respect to the channel and its links. Similarly, the receiver module will be having the same transmission (up) and reception (down) links for communication and flags as well. Fig.5 shows the receiver module of the design. The complete network control and logic circuit for selectively choosing the channels for communication is controlled by the NoC itself.

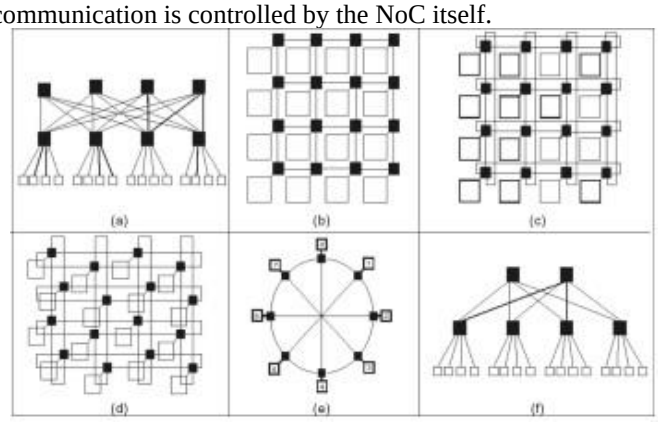

Fig.6. Network Topologies [4]

The Fig.6 depicts the hybrid layer of the NoC architecture that the prototype can handle for simultaneous inter module and intra module communication using NoC.

VI. IMPLEMENTATION

Implementation of the project was ideally divided into modules and NoC. NoC behaves as the center for control logic and all the data will be send across it for communication. The transmitter module is where the data packet transfer begins. We have implemented the mode of communication in serial mode. The data bit frame will be sent across up and down by packet switching. To begin with the transmitter section of the module, every transmitter has four links in its structure that will enable and decide the status of communication.

The serial data line and serial clock line in the module will synchronize and will initiate the beginning and termination of the data transfer process. The flags corresponding to the module will give the status of the channel for data transfer. Once a module for example module A wishes to send a packet of data to module B, then the transmitter channel of this module will be first requested for and then the data will be sent and stored before the actual communication begins. The buffer available in each links will perform store and forward paradigm of data transfer.

Data packet will be transferred serially in each clock cycles. We had used 16-bit frame so in 16 cycles all the data will be transferred serially. The transferred data will be stored in buffer and then pushed forward for reception. The buffer will have provision to display the current status of the buffer and decide if the next packet of data should be enabled for transferring. The NoC will take over the control of the switching logic without the use of any switching matrixes. The NoC is designed with fault tolerant measures such that the logic controller will compare the transmission and reception ID before the data is transmitted. This will prevent self-loop communication that might happen due to discrepancy of the bits or sequence number error. The module ID will be compared before assisting the data for further transfer. The data frame will have reception module ID which will decide the channel to be chosen for transferring the data.

Name	Theoretical	Practical
Throughput	300MBps	250MBps

Fig.7. Four Module design Implementation of NoC

Throughput calculations:
This formula was used to calculate the throughput values of NoC [4].

$$TP = \frac{(TOTAL\ MESSAGES\ COMPLETED) \times (MESSAGE\ LENGTH)}{(NUMBER\ OF\ IP\ CORES) \times (TOTAL\ TIME)} \quad \text{----------(1)}$$

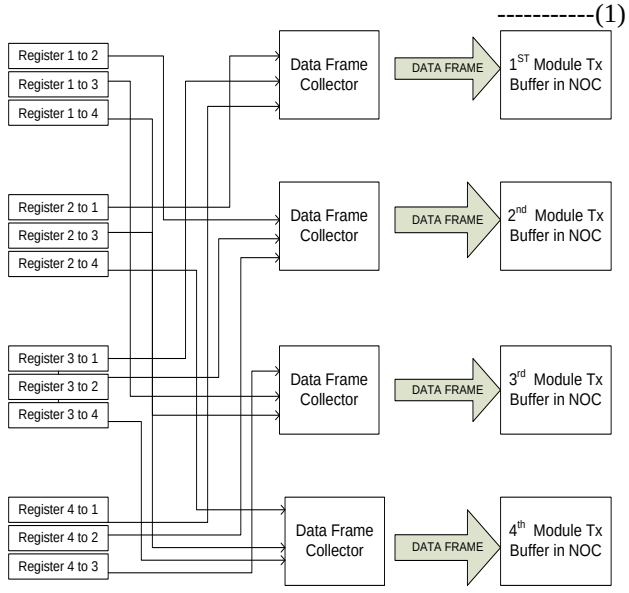

Fig.8. Serial communication simulation

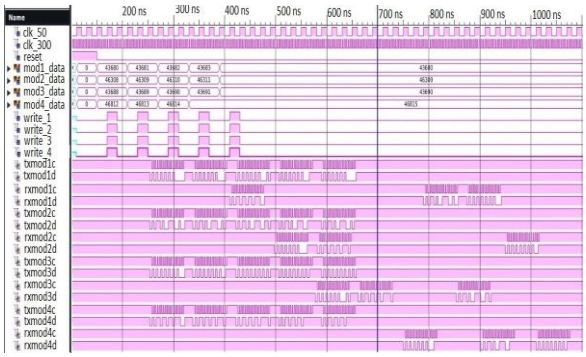

Fig.9. Simultaneous communication with network congestion

VII. SIMULATION RESULTS

The reception module again starts from taking the data that was written into the buffer, the read clock will initiate the signal for the data to be read into the reception channel. The bits will then be sent over the channel again in 16 clock pulse serially. The flags indicate the status of medium available for the data to be read before the reception channel is initiated. Once the flag gives clear signal, the data will be sent forward to the module and the data frame with reception ID will be discarded before transmission starts towards the receptor.

The type of modules that can be connected to the NoC is not dependent on the architecture of NoC design and the transfer of data will also be independent with respect to individual modules.

Fig.7 depicts four module design implementations in Spartan 6 mojo V3 FPGA board.

VII. CONCLUSION

In this project, a method to communicate across different modules that can be connected to the NoC board was discussed. The topology of the NoC for communication is like linear hybrid structure such that the network congestion and the transfer of data packet is completely controlled by the NoC. The use of switching matrix is completely avoided and buffer-based store and forward transmission is implemented. The complete process is dependent on acknowledgement signal for the transmission or reception to happen. These signals convey if the channels are empty for transmission or they are being processed already. After the simulation of the software we could achieve the complete data transfer in serial mode without any loss of data. The design and protocol of the project is hence satisfactorily working and completed.

We chose packet switching instead of traditional circuit switching used in the past, in a switch based NoC architecture. So, each module can communicate at any time.

REFERENCES

1. W. J. Dally and B. Towles, “Route packets, not wires: on-chip interconnection networks,” Proceedings of the 38th annual Design Automation Conference. Acm, pp. 684–689, 2001.
2. R. R. Dobkin, A. Morgenshtein, A. Kolodny, and R. Ginosar, “Parallel vs. serial on-chip communication,” in Proceedings of the 2008 international workshop on System level interconnect prediction. ACM,pp. 43–50, 2008.
3. S. Kumar *et al*., “A network on chip architecture and design methodology”,*Proceedings IEEE Computer Society Annual Symposium on VLSI. New Paradigms for VLSI Systems Design. ISVLSI 2002*, Pittsburgh, PA, USA, pp. 117-124, 2002.
4. P. P. Pande, C. Grecu, M. Jones, A. Ivanov, and R. Saleh, “Performance evaluation and design trade-offs for network-on-chip interconnect architectures,”IEEE transactions on Computers, vol. 54, no. 8, pp. 1025–1040, 2005.
5. H. Ye, Q. Wang, P. Yang, W. Li and Z. Yu, "The Design and Implementation of a NoC System Based on SoCKit," *2016 International Conference on Networking and Network Applications (NaNA)*, Hakodate, pp. 145-148, 2016.
6. F. Karim, A. Nguyen, and S. Dey, “An interconnect architecture fornetworking systems on chips,” IEEE micro, vol. 22, no. 5, pp. 36–45,2002.
7. K.-C. Chang, M. Liao, and B.-Y. Shiu, “Design and implementation of a noc supporting priority-based communications for many-core socs,”International Computer Symposium (ICS2010). IEEE, pp.483–488, 2010
8. H. Luo, S. Wei, D. Chen, and D. Guo, “Hybrid circuit-switched noc for low cost on-chip communication,” Anti-counterfeiting, Security, and Identification. IEEE,pp. 1–5,2012.
9. P. P. Pande, C. Grecu, A. Ivanov, and R. Saleh, “Design of a switch fornetwork on chip applications,” Proceedings of the 2003 International Symposium on Circuits and Systems, 2003. ISCAS’03., vol. 5. IEEE, pp. V–V,2003.
10. X.-T. Tran, T. Nguyen, H.-P. Phan, and D.-H. Bui, “Axi-noc: Highperformance adaptation unit for arm processors in network-on-chip architectures,” IEICE Transactions on Fundamentals of Electronics,Communications and Computer Sciences, vol. 100, no. 8, pp. 1650–1660, 2017.
11. X.-T. Tran, T. Nguyen, H.-P. Phan, and D.-H. Bui, “Axi-noc: Highperformance adaptation unit for arm processors in network-on-chip architectures,” IEICE Transactions on Fundamentals of Electronics,Communications and Computer Sciences, vol. 100, no. 8, pp. 1650–1660, 2017.
12. H.-K. Hsin, E.-J. Chang, C.-H. Chao, and A.-Y. Wu, “Regional aco based routing for load-balancing in noc systems,” Second World Congress on Nature and Biologically Inspired Computing (NaBIC).IEEE, pp. 370–376, 2010.

AUTHORS PROFILE

Senthil Athiban is currently pursuing his Master of Science in Engineering in Electrical Engineering specializing in Electronics and Mixed signal Circuits in Arizona State University, United States of America. He received his Bachelor of Technology degree in Electronics and Communication Engineering from Vellore Institute of Technology, India. His research analog and digital circuit design and radiation hardening.

Mehazin Shaju is currently working as a research and development engineer at Continental, Bengaluru, India. She received her Bachelor of Technology degree in Electronics and communications Engineering from Vellore Institute of Technology, India.

Sravini is currently pursuing her PhD under Dr.S Ananiah Durai at Vellore Institute of Technology, India. She received her Bachelor’s and Master’s degrees from Indian Universities.

Dr.S Ananiah Durai is an Associate Professor in School of Electronics Engineering, Vellore Institute of Technology, Chennai Campus, India. He received his Doctoral degree in Integrated Circuit Design from Massey University, Auckland, New Zealand. His research interests are analog CMOS, CMOS MEMs, and on-chip signal conditioning. He has authored over 12 research papers published in various journals.