

Performance realization of Bridge Model using Ethernet-MAC for NoC based system with FPGA Prototyping

S.P. Guruprasad¹, B.S. Chandrasekar²

¹Research Scholar, Dept. of ECE, Jain University, India

²Director, CDEVL, Jain University, India

Article Info

Article history:

Received Apr 9, 2019

Revised Mar 13, 2020

Accepted Mar 24, 2020

Keywords:

Bridge

NoC

FPGA

Ethernet-MAC

SoC Interconnect

ABSTRACT

The System on Chip (SoC) integrates the number of processing elements (PE) with different application requirements on a single chip. The SoC uses bus-based interconnection with shared memory access. However, buses are not scalable and limited to particular interface protocol. To overcome these problems, The Network on Chip (NoC) is an emerging interconnect solution with a scalable and reliable solution over SoC. The bridge model is essential to communicate the NoC based system on SoC. In this article, a cost-effective and efficient bridge model with ethernet-MAC is designed, and also the placement of the bridge with NoC based system is prototyped on Artix-7 FPGA. The Bridge model mainly contains FIFO modules, Serializer and deserializer, a priority-based arbiter with credit counter, packet framer, and packet parser with Ethernet-MAC transceiver Module. The Bridge with a single router and different sizes of the NoC based systems with mesh topology are designed using adaptive-XY routing. The performance metrics are evaluated for Bridge with NoC in terms of average Latency and maximum throughput for different Packet Injection Rate (PIR).

Copyright © 2019 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

S.P Guruprasad,
Research Scholar, Department of ECE,
Jain University,
#1/1-1, Atria Towers, Palace Road, Bangalore, India 560001.
Email: sppg1306@gmail.com

1. INTRODUCTION

The present and future embedded systems communicate with external devices through regular communications like the internet or Local Area Network (LAN). Hence, the internet is the prime source of the network which offers the accessing and sharing the information resources. The Network on Chip (NoC) develops gradually as a future communication system for SoC. The communication between processing elements (PE's) is by packets through network routers [1]. The NoC interconnection provides better solutions over current bus-based systems concerning performance metrics and overhead issues in networks. The NoC allows PE's communications parallelly for system performance improvements, and power utilization will be reduced by using the shorter links rather than long shared buses [2-3].

Various mechanisms are incorporated to solve the on-chip interconnection in NoC based systems. The wishbone compactible interface protocol is connected to NoC via bridge module for multiple master and slave communications. The FPGA (Field Programmable Gate Array) based NoC systems support dynamically reconfigurable systems than existing Application-Specific Integrated Circuits (ASIC) based NoC systems [4]. The FPGA's are mainly utilized to implement different kinds of digital systems. FPGA has many advantages which include, easy to use, flexibility or reprogramming, verification, and validation with device programming. The FPGA prototyping is common to step to verify the hardware design of SoC. The interconnect protocol stack is used based on the Open Systems Interconnect (OSI) layers. The point to point communication as a network stack, point to point memory-mapped communication as a streaming stack, and distributed shared memory-mapped communications a memory-mapped stack. This protocol stack uses the first five layers except for application and presentation layers. This protocol stack uses the programming models like Network

Interface (NI), router, Links, Shell, Bus, and Intellectual property (IP) for interconnections in multiple communication paradigms [5-6]. The Peripheral Component Interconnect (PCI) express is a system interface that provides the stability requirements for future applications.

Most of the advanced FPGA can be used to support most of the serial Interface or off-board communication protocols like USB, SPI, Gigabit Ethernet, PCI Express, Serial RapidIO, and others in a single FPGA platform for different designs. The USB and SPI are serial master-slave protocols that are not used in the proposed design. The implementation of different Bridges or multiple configurations of bridges with PCI Express in a single targeted FPGA which reduces the area complexity on the chip, but cannot provides long-distance wired connection [7]. Therefore, the proposed bridge architecture uses the Ethernet-MAC protocol, and it provides long-distance wired communications with scalable connections. It is suitable for on-chip and off-chip Bridging. In the future, the security features are incorporated in Network interfaces using cryptography algorithms [8] for Bridge based NoC designs and adopted for secured Network elements design.

In this research article, a cost-effective bridge model using Ethernet-MAC also bridge placement with NoC based systems are designed. The bridge results are FPGA resource-efficient, low Latency, and throughput concerning input traffic is evaluated For NoC based systems. Section 2 explains about related work on Bridge and interface protocols and research gap. Section 3 describes the Bridge model using Ethernet-MAC with a block diagram. The Bridge with NoC model is explained in section 4. Section 5 analyzes the Hardware synthesis results and performance evaluation. Finally, conclude the proposed work and future scope of the work.

2. RELATED WORK

In this section, the review of the related works is considered, which contains Bridging approaches with NoC, different SoC interface protocols, and its application usages. Bruce et al. [9] explain the evolution of the SoC interconnection and how NoC fits into the SoC system. NoC provides the interconnection solution with different vital features, which includes quality of service (QoS), asynchronous clock domains, close connection links, dynamic routing capability, and structure of the network.

Beyranvand et al. [10-11] described the bridging schemes in NoC systems to improve the QoS. The bridging scheme is introduced between the multiple chips like FPGA, ASIC, DSP, etc. and each device chip contains NoC systems. The flow of the protocol stack layers based on the OSI with on-chip interconnection is elaborated. The bridge scheme in each OSI layer is investigated for FPGA board prototyping. The design space and design criteria are explained for the bridging scheme. The bridge architecture is designed with the help of the Time Division Multiplexing (TDM) scheme and Xilinx Ethernet wrapper in both hardware and software approaches. The QoS provided as Best Effort (BE) and Guaranteed Throughput (GT) traffic to improve the Latency and bandwidth of the multiple connections BY concerning Input traffic.

Manu et al. [12] explained the implementation of AMBA-based advanced system bus (ASB) to APB Bridge. The APB Bridge is modeled using Finite State Machine with memory mapping. The ASB acts as a master module with priority-based arbitration, which can communicate data directly peripheral devices over APB Bridge. The Rachata et al. [13] presents the Hierarchical Rings with deflection (HiRD) routing to improve the scalability of the hierarchical ring interconnects. The bridge routers are designed as a hierarchical ring based NoC with different levels. Schoeberl et al. [14] present the Time predictable memory module with arbitration and access for on-Chip processors. The worst-case execution time is calculated for each TDM slot to improve the NoC latency, and it is used for Bridging arbitration schemes. The Monemi et al. [15] explains the NoC Router Micro-architecture with Low Latency using request masking technique and evaluate the NoC design with Open source Network simulators on Targeted FPGA.

Anala et al. [16] presented mesh-based NoC systems using different routing algorithms. The NOXIM simulator is used for testing the NoC designs. The results are analyzed for different traffic patterns includes random and transpose-2, and switching methods include buffer level and neighbors on a path, but these designs failed to work hardware environments. Fabio et al. [17] explained about AXI interface with Xilinx Zynq-7000, all programmable (AP) SoC, and analyze the reliability between the programmable logic like accelerator IP Block, IP block interfacing and Embedded ARM Processor using AXI-communication interface. Tidala et al. [18] described the performance of the NoC using the AXI-4 interface protocol on FPGA systems. The AXI provides the Interface between master (Programmable systems) and slave (memory controller). However, these systems are failed to maintain the excellent transmission rate, routing interface complexity, and QoS for different burst lengths.

Research gap: It is being noticed from the review of the recent works on the Bridge-based designs, different interface protocols interconnections for NoC or SoC-based systems, and the following research gaps are identified below.

- Minimal research on Bridge-based architecture and adoption in NoC based MPSoC system. Very few of the hardware-based Bridge model is designed and adopted on NoC systems.

- Most of the existing approaches are used as bus-based protocols for interconnection with NoC systems and lack a cost-effective solution and hardware complexities.
- The work done on On-chip FPGA Prototyping for Bridge architecture with the inclusion of Ethernet-MAC is very less.
- Most of the Hardware-based Bridge architecture uses off-chip Xilinx Ethernet-MAC Wrapper, and these systems are failed to maintain the synchronization issues.

Thus a cost-effective standalone bridge architecture with the inclusion of Ethernet-MAC for NoC based systems needs to fulfill the above gaps with better outcomes. The next section explains the detailed architecture of the Bridge Model to overcome the above gaps.

3. BRIDGE MODEL

The hardware architecture of the Bridge Model is represented in Figure 1. The Bridge model has four First in First out (FIFO) modules, multiplexer and demultiplexers, Serializer and deserializer, packet framer and packet parser, priority-based arbiter, credit counter, and Ethernet-MAC transceiver.

The Network Interface (NI) receives data signals via different request ports externally and sends it to the bridge model. The bridge models store the individual port data in different FIFO's. The synchronous FIFO used to write the data sequentially into corresponding memory locations. Each memory location is holding 32-bit data. Once data values are filled in memory locations and full signal set to high, Read the data from the same memory locations till the last data and empty signal indicate the memory is location is empty and set to high. The width of the FIFO depends upon the number of data transferred on the Bridge model. The multiplexer receives the two or more FIFO's data parallely and establishes the connection based on the priority arbitration. The priority-based arbiter receives the requests from the different ports like Data transaction level (DTL) with memory-mapped access and generates the grants based on the priority. The priority-based arbiter operates as a scheduler, to improve the QoS parameters in every bridge connections.

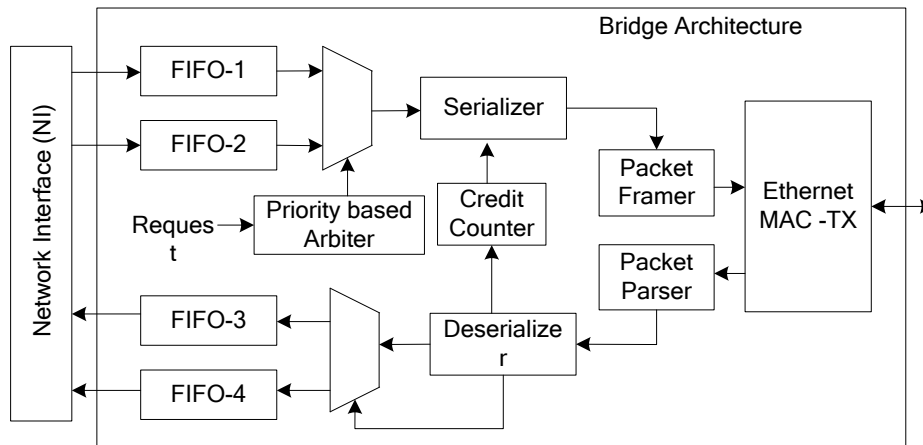


Figure 1. The hardware architecture of the bridge model

The serializer process the multiplexed 32-bit data concurrently and generates the 8-bit sequential data. The Serializer acts as a Parallel In Serial Out (PISO) for data conversion, and Serializer works based on the credit counter. The deserializer provides the input credits to the credit counter. The credit counter is a 2-bit counter, and for every successive count, parallel to serial data conversion is performing in the Serializer. These serial data used as payload data in Ethernet-framer. The ethernet frame and packet format are represented in Figure 2.

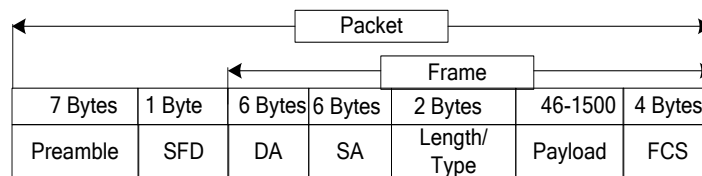


Figure 2. Ethernet frame and Packet format

The Ethernet frame is defined as per IEEE 802.3 [19] format and used in the bridge model. The Ethernet packet consists of an Ethernet frame with preamble and Start of delimiter (SFD). The Ethernet frame

has 56-bit (7-byte) preamble used to synchronize the packet data with proper timing. The 8-bit SFD is used to initiate the frame data. The 6-byte of the destination address in the receiver and 6-byte of the source address in the transmitter is used to configure the network. The Medium access control (MAC) packet information is accessed by using 2-byte type or length field. A user can send up to 1500 bytes of data and store it in the payload. The Cyclic Redundancy check (CRC) is used for detection and correction error bits in the 4-bytes of Frame Check Sequence (FCS).

The Hardware architecture of the Ethernet MAC transceiver architecture [20] is represented in Figure 3, which is having Receiver and transmitter models. The receiver model contains receiver- Finite state machine (FSM), Frame Length Counter (FLC), CRC, Receiver-FIFO. The transmitter model has transmitter-FIFO, padding with zeros followed by CRC and transmitter-FSM. The carrier and collision signals are used to check the Ethernet-MAC is operate as a half or full-duplex mode. The receiver-FSM perform the data processing operation. If the data is a valid packet and it will continue the process, otherwise it will discard the packet. The data packets are stored in Receiver-FIFO, while the excess packets are dropped. The FLC will check the received packet is valid or error or rejected.

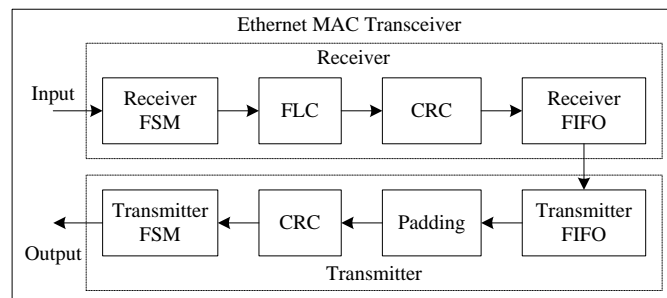


Figure 3. The hardware architecture of the Ethernet-MAC Transceiver

In transmitter-FIFO, hold the packets till the packet timer expires. The transmitter-FSM receives the data packets and processes by using transmitter FIFO till the end of the frame. If the frame is short, the data packets are padded by zeros and applied to the FCS using CRC to validate the packets. If the packet collision happens in the data processing, the data enter into jam state in FSM. After successive retries of data transmission and move to the next stage, and it is ready to transmit the valid data packets to the next stage of the bridge model.

The packet parser receives the data packets from the payload of the Ethernet-MAC Transceiver and divides into 8-bits of serial packet sequentially and which is inputs to deserializer module. The deserializer works based on the Serial in parallel out (SIPO) manner and receives the 8-bit and convert to 32-bit parallel data using shifting operation with the counter. These counter values are input to a credit counter. The 32-bit parallel data received as an input to the demultiplexer, based on the counter values, generates the two or more data values and inputs to FIFO's. The transmitted FIFO's data must be the same as received FIFO's to validate the bridge model. In next section describes the bridge placement in NoC.

4. BRIDGE WITH NOC USING ADAPTIVE-XY ROUTING

The bridge model is interconnected to NoC based systems that offer the on-chip and off-chip data flow control, off-chip data transaction, off-chip with various interconnections, and arbitration between multiple connections of the NoC based Multiprocessing SoC (MPSoC). The MPSoC chips are considered as an ASIC or FPGA devices for prototyping of Bridge interconnection. An example of the Bridge interconnection with 2x2 NoC architecture is represented in Figure 4. In this example, The Bridge model receives the data signals from the external resources like LAN or internet cable to Ethernet-MAC port. The Ethernet-MAC receives the valid data packets and transmits to the bridge model. The bridge model receives the data packets and performs the bridge interconnection operation and pass to the router model via NI. The Mesh topology is used in this design for building the different 2X2, 3X3, and 4X4 NoC architectures. The 2x2 NoC has four routers (R1, R2, R3, and R4), and all the routers are interconnected using link wires. The R1 receives the Bridge interconnected data and perform the data transaction based on the destination address of the Routers. Each router model has the five –port input registers, followed by packet framing with arbitration and adaptive-XY routing algorithm [21].

The 5-port input register have east, west, north, south, and local port and receives the bridge data packets in local port and store it temporarily, and passed to the priority encoder. The priority encoder works based on an arbitration request. The arbiter receives the request from the input registers and generates the grants to process the encoded data Future—the encoded data used in the packet formation.

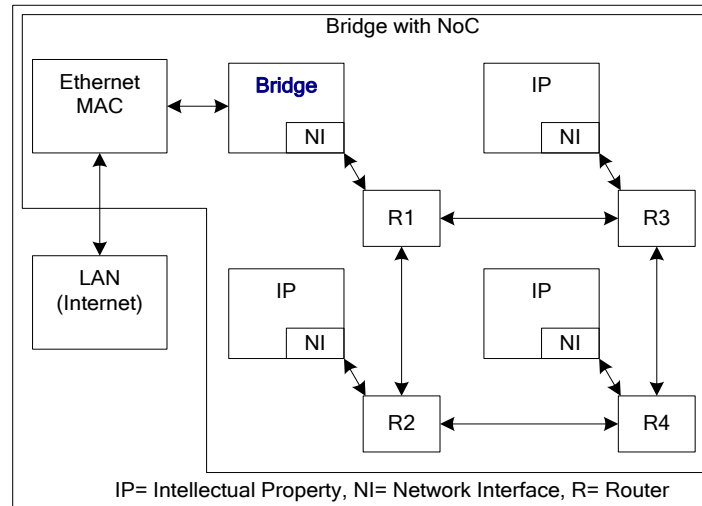


Figure 4. An example of Bridge Interconnection with 2x2 NoC

The router packet formation for Bridge NoC is represented in Figure 5. The packet formation is framed based on the request, destination XY address, and priority output. The 32-bit data acts as flit information, 2-bit destination X, and 2-bit destination Y address and 1-bit request form the router packet, which is having 37-bits, and it is considered as single 'Phit.' In this design, NoC is having 37-bits (1-Phit), which forms the 5-bytes.



Figure 5. Packet formation for Bridge –NoC Router

The current XY address is fixed as an identity for each router. The router R1 has 4-bit XY current address "0000". Similarly, for R2- 0001, For R3- 0100 and R4-0101. The routing computation in NoC is performed based on the routing algorithm. In this work, the adaptive-XY routing algorithm is considered. An adaptive-XY routing algorithm is an adaptive form of the Normal –XY Routing algorithm. The routing algorithm operates the 1st dimension in X-direction, and 2nd dimension in Y-direction with less number of routing path is confirmed, and the packet is assisted to destination dimension with less congestion. The shortest routing path is best, which finds the alternative congestion-free routing to avoid the input traffic based on the congestion parameter configuration. In this example, The Router R4 is considered as a destination location, and the bridge data packets will be reached to the destination based on the routing algorithm. In the next section, The Hardware synthesis results and performance analysis in terms of Latency and throughput are evaluated for Bridge with NoC Architecture.

5. RESULTS AND DISCUSSION

This section gives the outcome of hardware synthesis and performance analysis of the Bridge –NoC architecture using mesh topology with different network sizes. The proposed work is synthesized on Xilinx Environment using Verilog-HDL, and the implementation with Hardware prototyping is considered using the Artix-7 FPGA Platform.

5.1. Hardware synthesis Results

The Standalone Bridge with and without the inclusion of the Ethernet-MAC packet transceiver is incorporated in the designs. In advanced FPGA development boards, the Ethernet-MAC wrapper is available as an IP Core by Xilinx, and it is not considered in the proposed bridge architecture. The synthesis results of a bridge with and without the inclusion of the Ethernet-MAC are presented and tabulated in Table 1. The Resources of the synthesis results includes area utilization in terms of the slice registers, slice LUT's, and LUT-FF Pairs usage along with the maximum operating frequency and power utilization is presented. The FPGA resource of the bridge area cost is < 1%, excluding the Ethernet-MAC and other physical layer modules. With the inclusion of the Ethernet-MAC, bridge area cost is > 1% of the FPGA resources. The Bridge with Ethernet-MAC consumed less power utilization of 0.134W and operated at a 212.365MHz frequency on FPGA.

Table 1. Resource Utilization of Bridge Model

Resource Utilization	The Bridge without Ethernet MAC	Bridge with Ethernet MAC
Slice Registers	104	281
Slice LUTs	220	963
LUT-FF pairs	77	233
Maximum Frequency (MHz)	211.955	212.365
Total Power (W)	0.121	0.134

The resource utilization of the Bridge with NoC includes the single router, 2x2, 3x3, and 4x4 using Mesh topology-based architecture are tabulated in table 2. and represented in Figure 6.

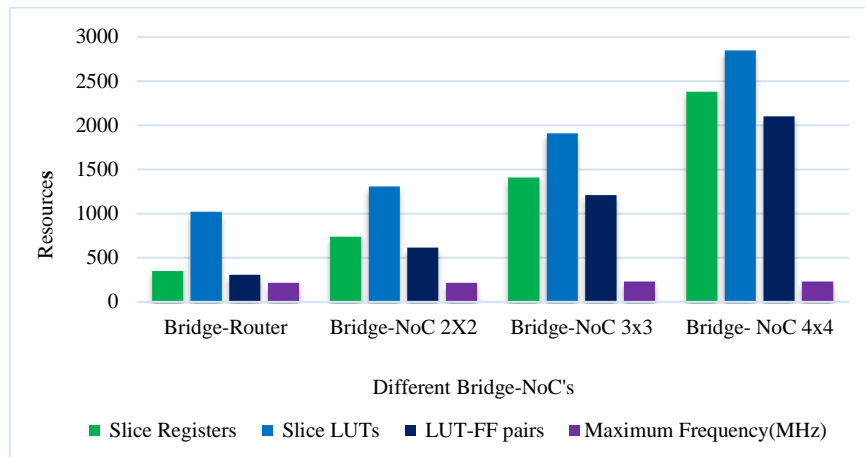


Figure 6. Synthesis Results of the Bridge-NoC with different Network Sizes

The bridge with a single router, 2x2, 3x3 and 4x4 area cost is < 2%, >2% , >3 % , > 4% of FPGA resources respectively. If the size of the network increases, the Bridge-NoC Architecture area resource utilization will also increase.

Table 2. Resource Utilization of bridge-NoC using mesh topology.

Resource Utilization	Bridge-Router	Bridge-NoC 2X2	Bridge-NoC 3x3	Bridge- NoC 4x4
Slice Registers	349	739	1408	2379
Slice LUTs	1020	1305	1907	2846
LUT-FF pairs	305	616	1208	2100
Maximum Frequency(MHz)	214.243	215.243	230.535	231.535

The Bridge with 4x4 NoC requires 2379 slice Registers, 2846 slice LUT's, 2100 LUT-FF pairs. The Bridge with 4x4 NoC operated up to a 214.24MHz frequency on Artix-7 FPGA, which speeds up the Bridge with NoC architecture, and it is suitable for real-time scenarios. These synthesis results show that the proposed Bridge –with NoC architecture is effectively implemented and prototyped on the FPGA platform. The proposed work also stated with better speed, and less FPGA resource utilization has been achieved.

5.2. Performance Analysis

The performance analysis of the Bridge with NoC architecture for different sizes are analyzed, which includes average Latency and Maximum throughput for input load in terms of PIR. The uniform traffic pattern is considered as an input load for analysis purpose. The total number of input data packets that can be sent in a single clock cycle is called a PIR. For example, if the Bridge is having a PIR of 0.5 means, Bridge can send 50 input packets in 100 clock cycles. The average Latency per flit of the Bridge is represented in terms of clock cycles (ns). The flit is a 32-bit input data packet used in the NoC systems.

The minimum Latency of the Bridge with Ethernet-MAC is calculated using the number of flits used and bridge architecture latency. The Bridge with Ethernet-MAC consumes 1232 clock cycles to complete the bridge packet transceiver operation. For a bridge with a single router or NoC latency is calculated based on the number of IP's used, minimum bridge latency along with router data flow logic. The router data flow logic takes two clock cycles to perform the routing operating in NoC from source to destination based on the adaptive –XY algorithm. The average Latency of a bridge with a single router and 2X2, 3x3, and 4x4 NoC architectures are analyzed for PIR are presented in figure 7. The Latency of the Bridge with a single router uses 740 clock cycles, and Bridge with 4x4 NoC uses 11846 clock cycles at 0.6 PIR.

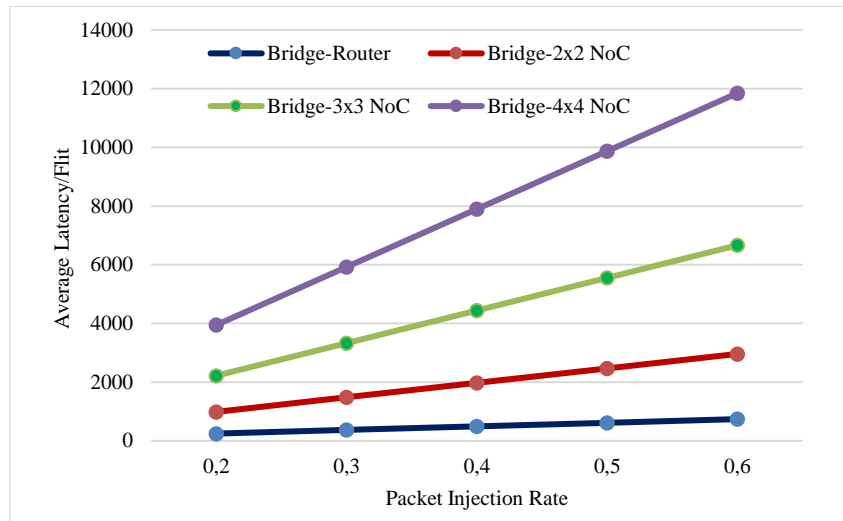


Figure 7. Average latency/flit v/s PIR of Bridge with NoC

If the PIR value is increasing, The Bridge with NoC utilizes more number of clock cycles. The FIFO module used in the bridge model has to perform the read and write operation based on a number of the packets sent, and it operates sequentially.

The maximum throughput of the Bridge with NoC architecture is calculated by the number of IPs used and followed by data packet size (flit), PIR, and Maximum operating frequency (MHz) obtained by FPGA design. An assumption is made that the IPs are interconnected to NoC boundary based on the mesh topology via network interfaces. The maximum throughput of the Bridge with a single router is obtained around 4.615 Gbps, and Bridge with 2x2, 3x3, 4x4 NoC is 17.132, 40.261, 71.576 Gbps respectively at 0.6 PIR. The maximum throughput of the Bridge with a single router, 2X2, 3x3, and 4x4 NoC are obtained by concerning different PIR is represented in figure 8.

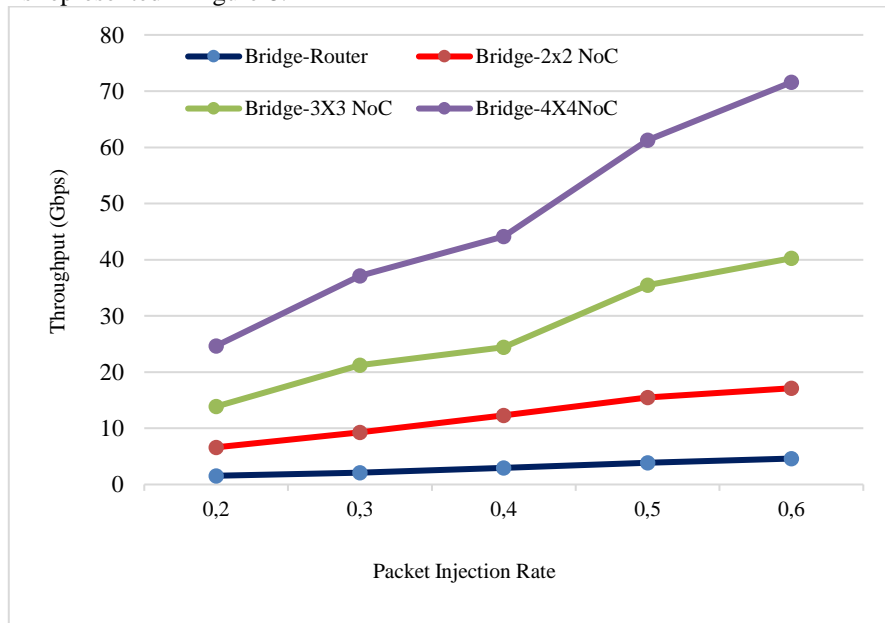


Figure 8. Maximum throughput v/s PIR of Bridge with NoC

The comparative study of the proposed Bridge with Ethernet –MAC with existing Bridge with AXI [22] and Ethernet-MAC [11] are tabulated in Table 3 by concerning the area resources and Frequency on Different FPGA's. The resource constraints include Slices, Flip-flops (FFs), and Maximum operating frequency (Fmax). The 32-bit Datawidth is fixed for all the designs. The proposed Bridge with Ethernet –MAC improves the chip area and Fmax than existing bridge approaches. The proposed Bridge model improves 52.89% and 41.12% overhead of Fmax by concerning Bridge with AXI [22] and Ethernet-MAC [11], respectively.

Table 3. Comparison of Bridge Model and Bridge-NoC with existing Approaches.

Designs	FPGA Device	Data Width	Slices	FF's	Fmax (MHz)
Only Bridge					
Bridge with AXI [22]	Stratix-II	32	402	249	100
Bridge with MAC [11]	Virtex-5	32	722	389	125
Bridge with MAC [11]	Virtex-6	32	2142	559	125
Bridge with MAC (This work)	Artix-7	32	281	233	212.3
Bridge based NoC					
Wishbone Bridge Based NoC [4]	Virtex-II,	36	431	452	151
Bridge based NoC (This work)	Artix-7	32	379	305	214.2

The Bridge based NoC Router is compared with existing wishbone-Bridge based NoC Router [4] with constraint improvements by concerning Slices, Flip-flops, and Fmax.

6. CONCLUSION

This research article presents an efficient and cost-effective bridge model with the inclusion of the Ethernet-MAC and also adopted the bridge model with NoC based systems. The bridge model offers the robustness by the inclusion of the Ethernet-MAC and which prototype quickly on on-chip FPGA Devices. The Bridge with a single router and different sizes of Mesh topology-based NoC is designed using congestion-free adaptive-XY routing. The synthesis results of the bridge architecture without and with the inclusion of the Ethernet-MAC utilizes < 1% and > 1% of FPGA resources and also Bridge with a single router and 4X4 NoC use < 2% and >4% of FPGA resources. The Bridge with 4X4 NoC operates at 231.535MHz on FPGA. The performance metrics of the Bridge with NoC consists of average Latency and maximum throughput for different PIR. The Bridge with single router works at 4.615 Gbps, and Bridge with 4x4 NoC operates at 71.576 Gbps at 0.6 PIR. The Bridge with Ethernet-MAC and Bridge-based NoC Router are compared with existing approaches with better improvements in hardware constraints. This model can be incorporated in futuristic researches with the security features to bridge and NoC based systems to strengthen the data packets from attacks.

REFERENCES

- [1] Kommineni, Bhavani Prasad, Rajkumar Srinivasan, Rickard Holsmark, Alf Johansson, and Shashi Kumar. "Modeling and Evaluation of a Network on Chip (NoC)–Internet Interface," 2005.
- [2] Smith, Burton J. "Interconnection networks for shared memory parallel computers." In mppoi, p. 255. IEEE, 1995.
- [3] Khan, Gul N., and Anita Tino. "Synthesis of NoC interconnects for custom MPSoC architectures." In 2012 IEEE/ACM Sixth International Symposium on Networks-on-Chip, pp. 75-82. IEEE, 2012.
- [4] Ehliar, Andreas, and Dake Liu. "An FPGA based open-source network-on-chip architecture." In 2007 International Conference on Field Programmable Logic and Applications, pp. 800-803. IEEE, 2007.
- [5] Billington, Jonathan, and Smit Saboo. "An investigation of credit-based flow control protocols." In Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops, p. 34. ICST, 2008.
- [6] Andreas Hansson and Kees Goossens. 2009. An on-chip interconnect and protocol stack for multiple communication paradigms and programming models. In *Proceedings of the 7th IEEE/ACM international conference on Hardware/software code sign and system synthesis (CODES+ISSS '09)*. ACM, New York, NY, USA, 99-108.
- [7] Implementing PCI Express Bridging Solutions in an FPGA, Lattice Semiconductor White Paper, and July 2010.
- [8] S. P. Guruprasad and B. S. Chandrasekar, "An evaluation framework for security algorithms performance realization on FPGA," 2018 IEEE International Conference on Current Trends in Advanced Computing (ICCTAC), Bangalore, India, 2018, pp. 1-6.
- [9] Mathewson, Bruce. "The evolution of soc interconnect and how noc fits within it." In Design Automation Conference, pp. 312-313. IEEE, 2010.
- [10] Nejad, Ashkan Beyranvand, Matias Escudero Martinez, and Kees Goossens. "An FPGA bridge preserving traffic quality of service for on-chip network-based systems." In Design, Automation & Test in Europe, pp. 1-6. IEEE, 2011.
- [11] Nejad, Ashkan Beyranvand, Anca Molnos, Matias Escudero Martinez, and Kees Goossens. "A hardware/software platform for QoS bridging over multi-chip NoC-based systems." *Parallel Computing* 39, no. 9 (2013): 424-441.
- [12] Manu, B. N., and P. Prabhavathi. "Design and implementation of AMBA ASB APB Bridge." In 2013 International Conference on Fuzzy Theory and Its Applications (iFUZZY), pp. 234-238. IEEE, 2013.
- [13] Ausavarungnirun, Rachata, Chris Fallin, Xiangyao Yu, Kevin Kai-Wei Chang, Greg Nazario, Reetuparna Das, Gabriel H. Loh, and Onur Mutlu. "Design and evaluation of hierarchical rings with deflection routing." In 2014 IEEE 26th International Symposium on Computer Architecture and High-Performance Computing, pp. 230-237. IEEE, 2014.
- [14] Schoeberl, Martin, David Vh Chong, Wolfgang Puffitsch, and Jens Sparsø. "A time-predictable memory network-on-chip." In 14th International Workshop on Worst-Case Execution Time Analysis. OASICS, 2014.

- [15] Monemi, Alireza, Chia Yee Ooi, and Muhammad Nadzir Marsono. "Low latency network-on-chip router microarchitecture using request masking technique." *International Journal of Reconfigurable Computing* 2015.
- [16] Anala, M. R., Amit N. Subrahmanya, and Allbright D'souza. "Performance Analysis of Mesh-based NoC's on Routing Algorithms." *International Journal of Electrical and Computer Engineering* 8, no. 5 (2018): 3368.
- [17] Benevenuti, Fabio, and Fernanda Lima Kastensmidt. "Reliability evaluation on interfacing with AXI and AXI-S on Xilinx Zynq-7000 AP-SoC." In *2018 IEEE 19th Latin-American Test Symposium (LATS)*, pp. 1-6. IEEE, 2018.
- [18] Tidala, Neethika. "High-Performance Network on Chip using AXI4 protocol interface on an FPGA." In *2018 Second International Conference on Electronics, Communication, and Aerospace Technology (ICECA)*, pp. 1647-1651. IEEE, 2018.
- [19] IEEE 802 LAN/MAN Standards Committee. *IEEE Std. 802.3 Edition*, 2012.
- [20] Guruprasad, S. P., and B. S. Chandrasekar. "An Optimized Packet Transceiver Design for Ethernet-MAC Layer Based on FPGA." In *International Conference on Intelligent Data Communication Technologies and Internet of Things*, pp. 725-732. Springer, Cham, 2018.
- [21] Guruprasad, S. P., and B. S. Chandrasekar, "Design and Performance analysis of Adaptive-XY over N-XY and DO Routing on FPGA Platform", *IJRECE*, pp. 2166-2171, Vol. 6 Issue 3, July-September, 2018.
- [22] Xiao, Fu-ming, Dong-sheng Li, Gao-ming Du, Yu-kun Song, Duo-li Zhang, and Ming-lun Gao. "Design of AXI bus based MPSoC on FPGA." In *2009 3rd International Conference on Anti-counterfeiting, Security, and Identification in Communication*, pp. 560-564. IEEE, 2009.

BIOGRAPHY OF AUTHORS



Mr. Guruprasad S.P received his B.E Degree in Electronics and communication from VTU, Karnataka, India in 2007, Master of Technology in VLSI Design and Embedded System from VTU in 2011. He is working in Private Organization as a Sr. Design Engineer and has 7 years of experience in VLSI Field. He is pursuing his Ph.D. from Jain University, India. His research area includes VLSI Design, Network Elements Design, Computer Network Security, MPSoC, and FPGA Design.



Dr. Chandrasekar B.S was born in 1965 secured state ranks in his 10th and 12th classes. He acquired B.E (E&C) from BMSCE, Bangalore, and proceeded to IISc for his Master's and Doctoral Degrees. He was a Professor / HoD from 1996 to 2000 at BMSCE for both day and evening colleges. During this period, he was also Chairman, BoE (BU), and BoS (VTU). Then on he proceeded to work with Infosys Technologies for 14 years in the 'Telecom Software' domain. Presently, he is Director for Distance Education & Virtual Learning @ Jain Deemed to be University.