

PAPER • OPEN ACCESS

Design of Network on Chip (NoC) Computing Node for Mesh Topology using Soft-core NIOS-II Processor

To cite this article: Udaysing V Rane *et al* 2021 *J. Phys.: Conf. Ser.* **1921** 012075

View the [article online](#) for updates and enhancements.

You may also like

- [Review—Challenges and Innovations in Nano-Computing and Reliable Interconnect Technologies](#)
Shilpa Gupta
- [Design and development of low power NoC architecture using clock gating technique for SoC applications](#)
Nagarjuna Malladhi and Girish V Attimarad
- [A congestion-aware OE router employing fair arbitration for network-on-chip](#)
Lu Liu, Yadong Sun, Zhangming Zhu *et al.*



The Electrochemical Society
Advancing solid state & electrochemical science & technology

ECS UNITED

247th ECS Meeting
Montréal, Canada
May 18-22, 2025
Palais des Congrès de Montréal

Early registration deadline: April 21, 2025

Unite with the ECS Community

Design of Network on Chip (NoC) Computing Node for Mesh Topology using Soft-core NIOS-II Processor

Udaysing V Rane¹, Rajendra S Gad² and Charanarur Panem³

¹Dnyanprassarak Mandal's College and Research Centre, Assagao, Bardez, Goa, India
udaysingvrane@gmail.com

²Department of Electronics, Goa University, Goa, India

³Research Student, Department of Electronics, Goa University, Goa, India.

Abstract. Paper discusses the design of computing node for Mesh topology-based Network on Chip (NoC) platform. The computing node has a capability of processing and routing data over four I/O port switch. Mesh topology is popular due to scalability and its simplest form. The platform is design for (N x N) computing nodes, where each node has soft-core based 'NIOS-II' processor. Each computing node has four-way switch East, West, North, South and one local port. The Mesh topology uses static look-up table optimized using proper routing algorithms. The look-up table information is used by Virtual Circuit Interface (VCI) of the switch node to forward the packets using combination of priority based round robin diagonal propagation scheduling algorithm. The design of computing node is generated using –Quartus-II ALTERA software and simulated for functional simulation. Such platforms are popular for data parallelism having applications in Smart sensors.

Keywords: NoC, Computing Node, NIOS-II, Mesh topology

1. Introduction

The advance in semiconductor processing technology to sub-nano has brought into several side effects. One such side effect is a wiring delay and is a critical issue now a days. This wiring delay grows exponentially as speed of gate delay becomes much faster as shown in Figure 1 and is mainly because of the capacitance increase caused by narrow channel width and due to increased crosstalk. Hence, to sustain this trend, the wiring is one of the critical issues to be concerned.

The resent applications mainly require lot of computations along with low power and high performance, thus computing resources number on single chip has also increased and is possible due to VLSI technology which supports an extensive integration of transistors. The entire system can be built on single chip called as System-on-Chip (SoC) which consist of several computing resources like Central Processing Units (CPU), Intellectual Properties (IP) Cores, Digital Signal Processors (DSP) etc. the interconnection between these computational resources is another challenging issue. To communicate with each computational resource within the system, a shared based bus interconnection is used in majority of System-on-Chip applications, due to its low-cost and simple control characteristics. However, Scalability is a major drawback of such shared bus interconnection since at a time only one master can utilize the bus thus all the bus accesses should be serialized by the arbitrator. Therefore, a new interconnection method is



required in an environment where the request for bus is large and the required bandwidth for interconnection is more than the current bus.

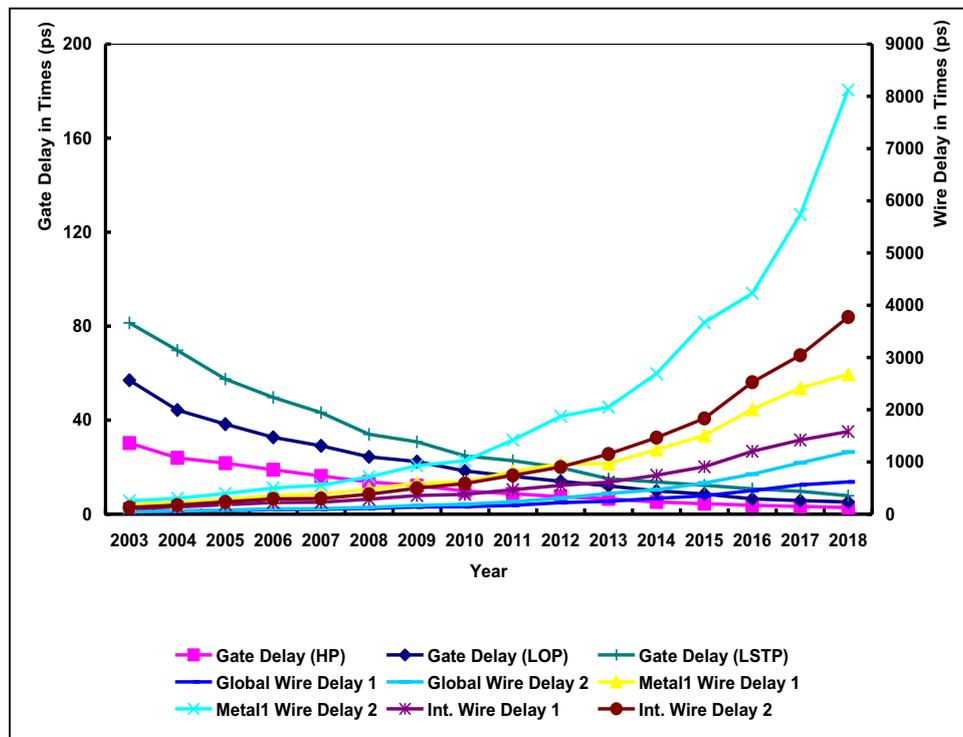


Figure 1. Technical Roadmap in Semiconductor Industry [1]

To overcome the issue of scalability in shared bus base architecture some advance bus-based architectures were adopted like ARM™ AMBA [2], OpenCore's WISHBONE SoC interconnection [3], and IBM CoreConnect™ [4-5].

Alternative solution to overcome this drawback of communication and wiring delays in upcoming technology is to accept interconnections which is like network called NoC architecture. This concept is taken from the modern computer networks. Which is achieved by using network-like communication where routers are inserted intermediate between communication objects thereby shortening required wiring. This router-based interconnection mechanism gives very good scalability and even the wiring is not that complex. The economics has already proved that the NoC minimizes cost of SoC manufacturing, market time, volume time, and risk of design or increase in SoC performance. It is already proved that NoC approach has outperform traditional busses and has improved the system throughput notably. The crossbars or multilayered busses have characteristics between NoC and traditional busses but still NoC outperform when comes to performance and complexity.

The design of NoC depends on the interfaces between interconnection fabric and processing elements of NoC. This established procedure has few weak points with respects to slow bus response time, problems in scalability, limitation of bandwidth, and also energy limitation. Also since this bus interconnection are connected to several component in a network, leads to slow interface and high consumption of power. Moreover, Increasing the number of connections of the elements infinitely is impossible due to bandwidth limitation in a bus. Therefore, NoC design performance greatly relies on the interconnection paradigm [6-10]

In Computer Network the network technology, though well developed, without any modification or reduction, we cannot apply to a chip-level intercommunication environment directly. Thus, several researchers are working on for developing network architectures suitable for on-chip communication. The basic functionality needs to be very simple and light-weighted and should be small enough to be a component constructing a SoC for NoC architecture. This paper discusses the design of a computing node for the NxN node mesh topology having four way switch and a local port interface with processing element.

2. NoC system architecture details:

2.1 Topologies

Many different criteria must be taken into consideration when designing NoC schemes such as routing algorithms, network topologies, latency, efficiency, complexity, etc. With respect to network topologies there are several different kinds proposed by many researchers. One such topology called SPIN proposed by Guerrier and Greiner [11], In order to Interconnect Ip blocks they have used a fat-tree architecture. Each node has four children in this fat tree, and the parent is repeated on every level of the tree four times. A mesh-based architecture of interconnecting components, called CLICHÉ, as suggested by Kumar et al. [12]. This architecture is made up of a M x N mesh of switches that connect the design of machine resources using IPs that are positioned with switches. Dally and Towles et al [13] are suggesting the 2D Torus NoC design. In theory this architecture is the same as a normal mesh. The only distinction is that the edges switches are connected by wrap-around channels with the switches on the other side. This Architecture consists of eight nodes and 12 bidirectional connections. Karim et al. [14] suggests the OCTAGON MP-SoC architecture. A processing element and a switch are included in each node. Connection between nodes in a simple octagonal unit takes up to two hops. The Butterfly Fat-Tree (BFT) architecture was suggested by Pande et al [15]. The IPs are positioned in this architecture at the leaves and the buttons on the vertices.

As a feasible topology of the NoC structure, the mesh topology becomes common because of its modularity. Without any modifications to the existing structure new nodes and links can be used to topology and quickly expanded. On-chip connectivity uses mesh nodes as fundamental components and shapes theoretically essential components in the NoC setting to achieve a scalable communication model [16]. The basic Mesh topology design consists of an IP core-based processing element embedded with the Switch/Router using Ethernet/Bus-based Network Interface (NI) as depicted in (Figure 2).

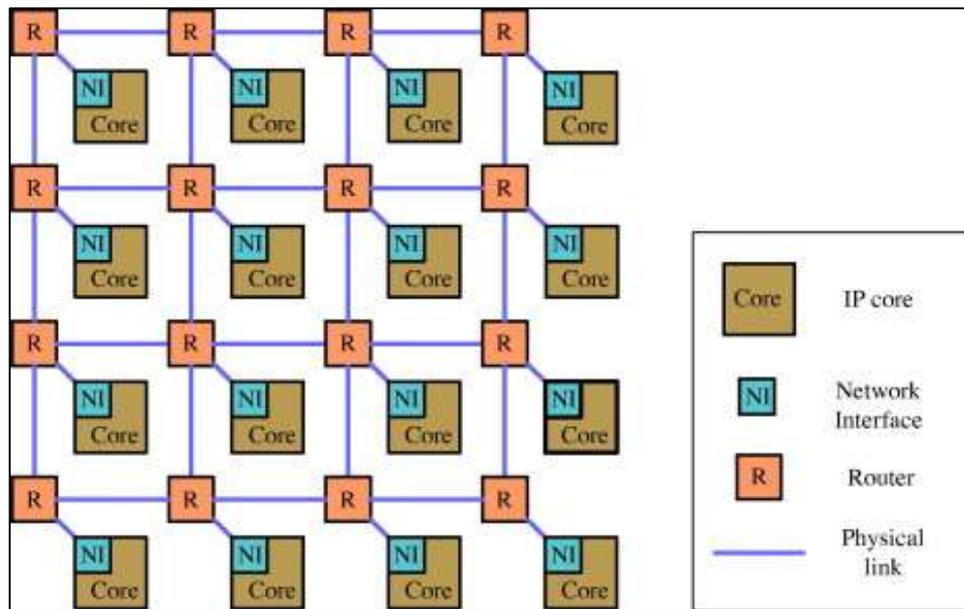


Figure 2. Architecture of Mesh topology NoC

2.2 Routing algorithms

Another parameter in the NoC setting is the routing algorithm that decides the best path from source to destination. There can be multiple paths from source to destination. The switching technique is used to connect the systems for making one to one communication and decides the best route for packet transfer to the destination. This approach is generally known as switching circuits, switching packets, and switching wormholes [17]. It determines whether and how their inputs are connected by internal switches with outputs and when messages can be transferred along these paths. The value of this adjustment is to evaluate the track from source to destination over the full duration of the data delivered such that the bandwidth of the network is maintained. The advantage is that during data transfer and the initialization phases valuable resources are added to it causing needless delays. Packet switching, though does not connect services, involves large architectural buffers that increase overhead hardware. When switching wormholes, large packets further split into a flux control unit or flits called a fixed volume, resulting in fewer buffer space on switches. In this paper we proposed to design the ‘N x N’ switch Mesh NoC platform as described in Figure 2. The local node consists of processing element embedded with the Switch called computing node using soft core NIOS-II processor.

2.3 Switch Design Details

The switch consists of First In First Out (FIFO) buffer for each connection i.e. local and four directional to store the incoming and outgoing Ethernet/ATM/IP packets to be sent to various nodes. The arbitration control logic i.e. *Virtual Circuit Interface (VCI)* will guide the cross bar array to guide the packet to the respective FIFO destination or to the local computing node.

Figure 3 displays the description of the 4x4 parameter switch architecture. 4 data lines, 4 frame pulse inputs, 1 clock input, a reset input, and a global reset input are the input lines for the switch. 4 data output lines, 4 true data lines, 4 output frame pulse lines, 1 clock output, and 4 outputs are the output lines of the switch that show the origin of the data coming to each port of the data output. With small modifications, this move can be scaled up or down. The four data inputs (data-in1 to data-in4) are 8-bit wide buses each and hold packets of fixed size Asynchronous Transfer Mode (ATM).

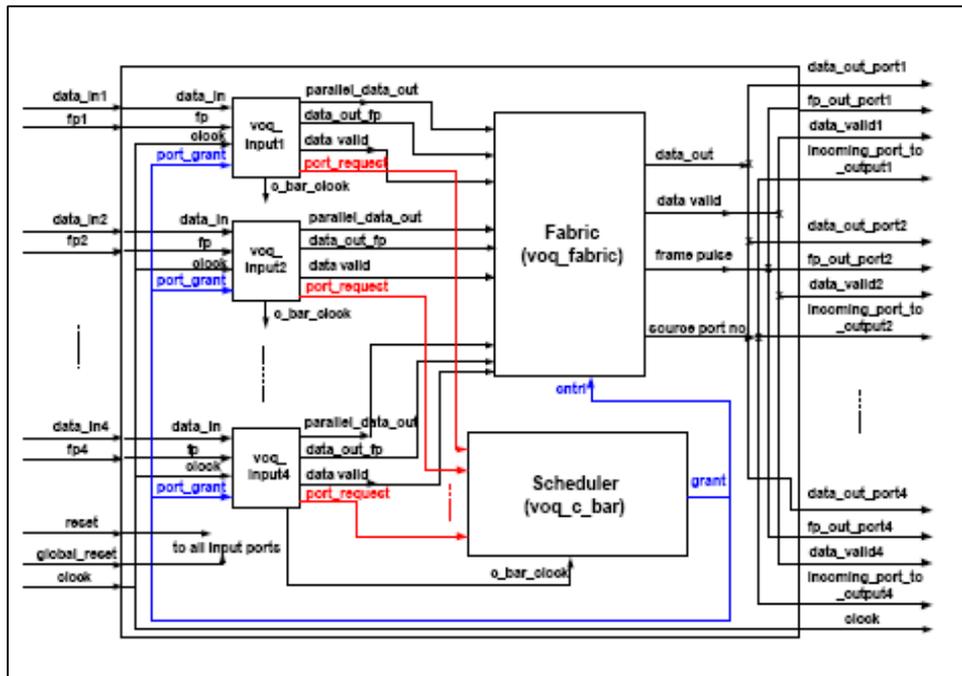


Figure 3. Block diagram of switch

Additional packet formats like IP packets must first be broken into ATM cells before switching. Each clock loop has one byte for the turn. Data are on the rising edge of the clock in our commutation. For all switch components that are used to clock input and output data sources, the input 'clock' is global. Internally, the input port modules create a new clock called "c bar clock." The length of this clock is the time of the packet that is required for an output packet from the switch. The length of the packet depends on the clock input frequency. Our architecture is based on the law that the "c bar clock" is 59 times lighter than the "clock" input. An ATM packet measures 53 bytes, thus taking 53 clock cycles, and 6 more clock cycles to handle internal delays and buffer changes. The 1-bit large signal called input frame pulse (fp1 to fp 4) is used to signify packets beginning. At least one clock cycle should have a pulse on the 'fp' input side. This signal is inspected at the downside of the input of the clock. The first data byte, which arrives on the second edge, is called the first byte of the packet after the frame pulse is identified. The transition resets all counters in the design and initializes all counters into their initial values. Input buffers in which the ATM packets are stored are often reset by using the "global reset" signal. If a mistake happens during a packet swapping, you will use the reset signal to adjust the packet again. However, 'Global reset' can be used if you want to restart the whole switch and erase the contents of the buffers. The output ports in our switch are just chip pins and have little to do with processing or storage. The packets are re-assembled and deposited in an output module on the output ports so it can reach the network. The output lines viz 'clock' and 'data out port' as well as 'fp out port' and 'data valid' are used. In 'data out port1' output data bytes are sent to the output ports 'data out port4.' The pulse signals produced by the output frames (fp out port1 to fp out port4) mark a start for the outgoing packets for their respective line data. The ratio of the start of the packet to the frame pulse in the escape ports is the same as input.

When it comes to choosing the path from source to destination between potential routes, the routing algorithms are called deterministic and adaptive [18]. Without regard to the Network's state, the ignorant/deterministic routing algorithms select a path, which makes the architecture complex relatively simple. Adaptive routing algorithms consider the network resource buffer state, node or connection status

or channel load history. Depending on the degree of adaptiveness, they are graded as minimum or entirely adaptive routing. Since the simplicity of routing tracks is used for adaptive routing algorithms, Sophistication of architecture is also improved. For eg, the DOR (organized routing dimension [19], ROMM [20] and the Near-Optimal Words of the Routings in Two Dimensions [21] are examples of deterministic or undisturbed routing algorithms. Some researchers have developed algorithms that use adaptive routing algorithms in addition to this improved performance routing. We have used static look-up table optimized using above-described routing algorithms in our conceptualized mesh topology. In order to direct the packets by integrating the priority round ribbon diagonal propagation preparation, Virtual Circuit Interface (VCI) information is used for the lookup table information.

2.4 Implementation of the computing node

A mesh topology measurement node is introduced with the 8-bit Avalon-Quarter-II bus interface using the NIOS-II soft core processor (Figure 4). The computer node is an optimized switch with a computing capability in the soft-core IP NIOS-II processor.

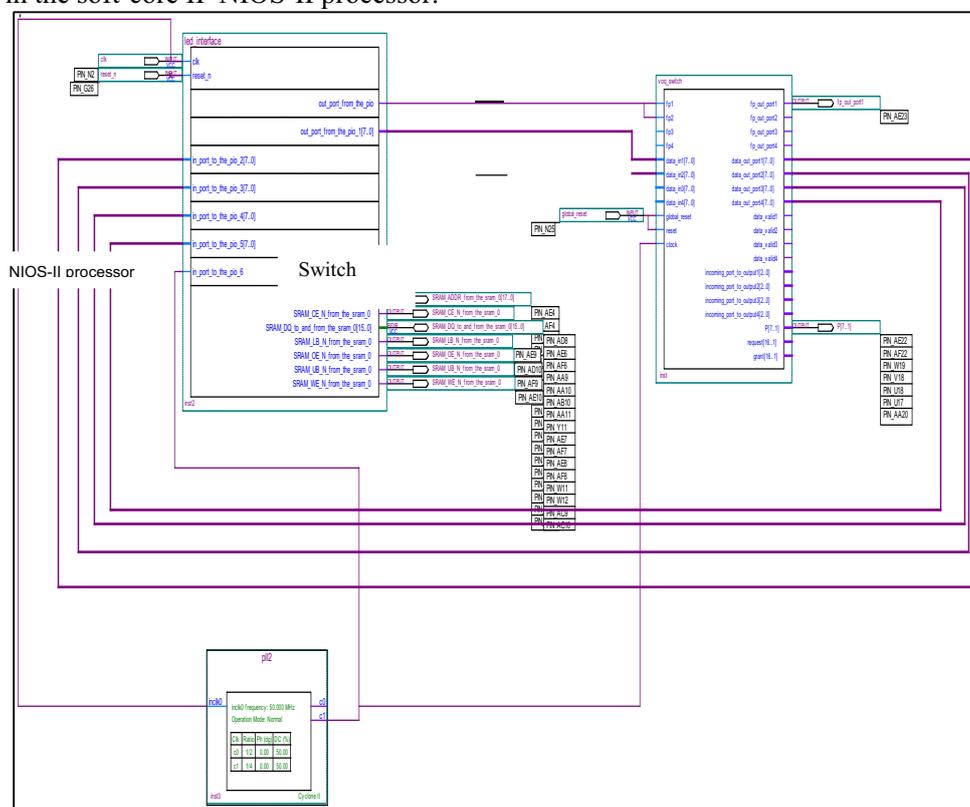


Figure 4. Computing Node embedded with switch.

3. Results and Discussion

Using QUARTUS II program, the practical simulation was carried out. Input to switch was supplied through vector waveform file as shown in Figure 5. Four different packets of 53 bytes were injected to four different input ports of switch. Every input port of switch has fixed look up table based on this information the routing is performed. The input data packets, and expected output data packets are shown in Table 1&2 and the same output is seen in the output waveform as shown in Figure 6.

Computing Node embedded with switch was also dumped on cyclone II DE2 board and using NIOS II 8.0 IDE we wrote a c program to input the data packets and output was analyzed using SignalTap II Logic Analyzer. It was found that data synchronization with the clock was difficult, further we also tried to input data using Real time operating system.

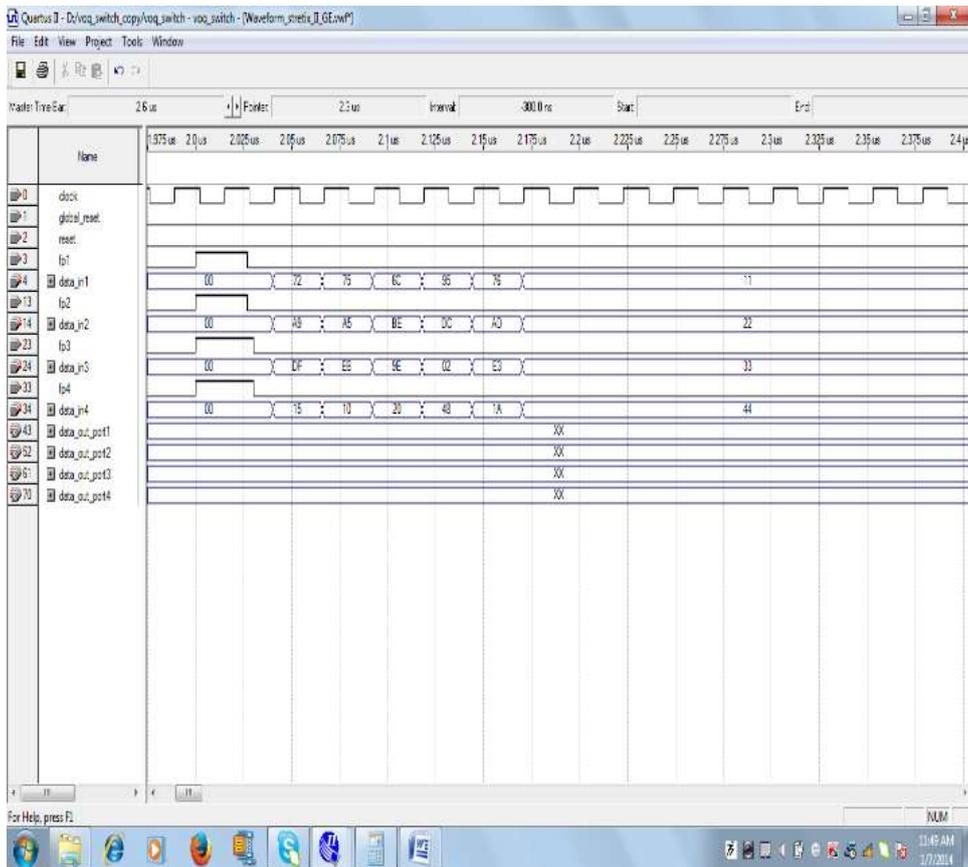


Figure 5. Input vector waveform file

Table 1. Details of Input packets

Input data byte stream					
1 st byte	2 nd byte	3 rd byte	4 th byte	5 th byte	6-53 bytes
72	75	6C	95	76	11
A9	A5	BE	DC	AD	22
DF	EB	9E	02	E3	33
15	10	20	48	1A	44

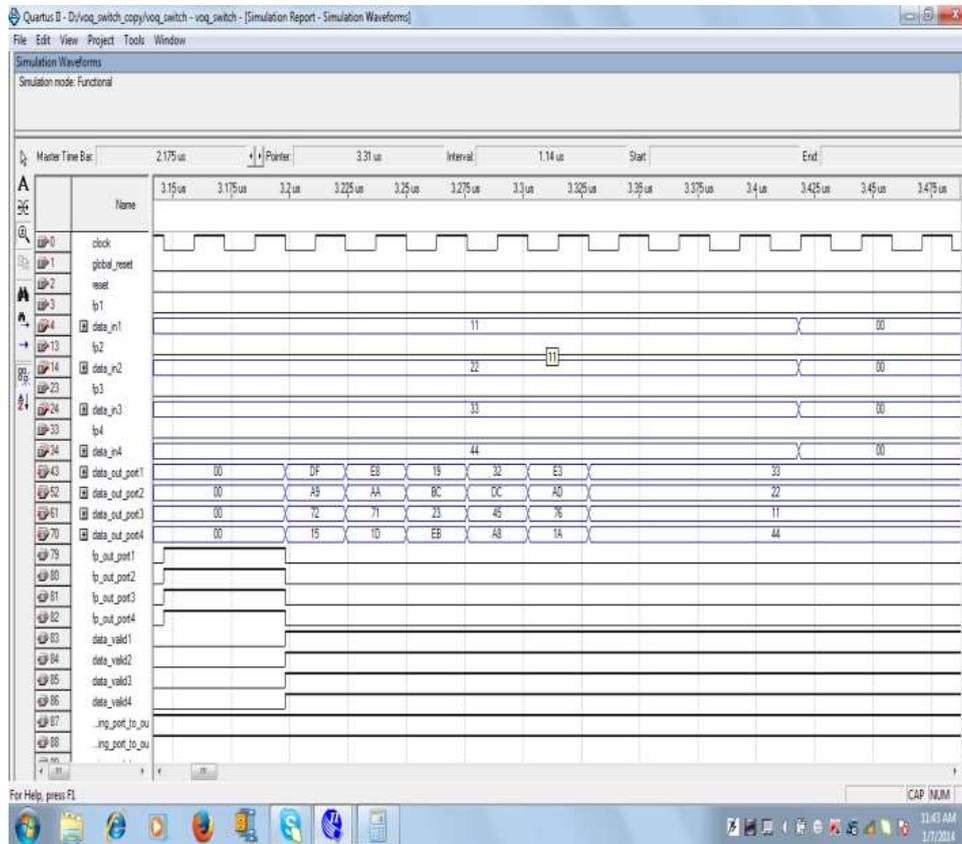


Figure 6. Output functional simulation results.

Table 2. Details of Output packets

Output data byte stream					
1 st byte	2 nd byte	3 rd byte	4 th byte	5 th byte	6-53 bytes
72	71	23	45	76	11
A9	AA	BC	DC	AD	22
DF	E8	19	32	E3	33
15	1D	EB	A8	1A	44

References

- [1] ITRS, International Technology Roadmap for Semiconductors 2004 Update, 2004.
- [2] AMBA™ Specification Rev. 2.0, <http://www.arm.com>, 1999.
- [3] Specification for the: WISHBONE System-on-Chip (SoC) Interconnection Architecture for Portable IP Cores, OpenCore, 2002.
- [4] The CoreConnect Bus Architecture, <http://www-03.ibm.com/chips/products/coreconnect/>, 1999.

- [5] A Comparison of Network-on-Chip and Busses, http://www.artemis.com/noc_whitepaper.pdf, 2005.
- [6] Charanarur Panem, R. S. Gad, U. V. Rane and V. R. Gad, "Evaluation of 3D & 4D Two Layer Mesh NoC for Fault Tolerance over Combinations of Vertical Channel," 2018 International Conference on Current Trends towards Converging Technologies (ICCTCT), Coimbatore, 2018, pp. 1-7, doi: 10.1109/ICCTCT.2018.8551081.
- [7] Charanarur Panem, U. V. Rane, V. R. Gad, R. S. Gad and G. M. Naik, "Evaluation of 2,4,8 Layer 3D Mesh NoC with Virtual Channel for Capacity Enhancement," 2018 International Conference on Current Trends towards Converging Technologies (ICCTCT), Coimbatore, 2018, pp. 1-8, doi: 10.1109/ICCTCT.2018.8550919.
- [8] C. Panem et al., "Performance analysis of 16×16 , 32×32 , 64×64 2-D mesh topologies for network on chip," 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, 2017, pp. 1757-1764, doi: 10.1109/WiSPNET.2017.8300063.
- [9] Charanarur Panem, Udaysingh V. Rane, Rajendra S. Gad, "Network On Chip Performance Analysis Of 2-D Mesh ,2D Torus,3D Mesh Topologies For CBR and FTP applications" , 10th Annual Symposium on VLSI and Embedded System VSI-2017, ,GEC, Goa, April 28th -2017.
- [10] Charanarur Panem*, U A Rane, Rajendra S. Gad, "3D & 4D 2 Layer Mesh NoC Evaluation for Fault Tolerant over Combinations of Vertical Channel" , 11th Annual Symposium on VLSI and Embedded System VSI-2018, Goa University, Goa, March 22-23,2018.
- [11] P. Guerrier and A. Greiner, "A Generic Architecture for On-Chip Packet-Switched Interconnections," Proc. Design and Test in Europe (DATE), pp. 250-256, Mar. 2000.
- [12] S. Kumar et al., "A Network on Chip Architecture and Design Methodology," Proc. Int'l Symp. VLSI (ISVLSI), pp. 117-124, 2002.
- [13] W. J. Dally and B. Towles, "Route Packets, Not Wires: On-Chip Interconnection Networks," Proc. Design Automation Conf. (DAC), pp. 683-689, 2001.
- [14] F. Karim et al., "An interconnect Architecture for Networking Systems on Chips," IEEE Micro, vol. 22, no. 5, pp. 36-45, Sept./Oct. 2002.
- [15] P. P. Pande et al., "Design of a Switch for Network on Chip Applications," Proc. Int'l Symp. Circuits and Systems (ISCAS), vol. 5, pp. 217-220, May 2003.
- [16] J. Duato et al., "Interconnection Networks: An Engineering Approach," IEEE Computer Society Press, 2003.
- [17] P. P. Pande et al., "Performance Evaluation and Design Trade-Offs for Network-on-Chip Interconnect Architectures," IEEE Trans. Computers, vol. 54, no. 8, pp. 1025-1040, Aug. 2005.
- [18] Principles and Practices of Interconnection Networks, W. J. Dally and B. Towles, Morgan Kaufmann Publishers, San Francisco, CA, 2004.
- [19] H. Sullivan and T. R. Bashkow, "A Large Scale, Homogeneous, Fully Distributed Parallel Machine," Proc. Symp. Computer Architecture, pp. 105-117, ACM Press, 1977.
- [20] T. Nesson and S. L. Johnsson, "ROMM Routing on Mesh and Torus Networks," Proc. ACM Symp. Parallel Algorithms and Architectures, pp. 275-287, ACM Press, 1995.
- [21] D. Seo et al., "Near-Optimal Worst-case Throughput Routing for Two-Dimensional Mesh Networks," Proc. Int'l Symp. Computer Architecture (ISCA), pp. 432-443, June 2005.
- [22] J. Hu and R. Marculescu, "DyAD – Smart Routing for Network-on-Chip," Proc. Design and Automation, pp. 260-263, ACM Press, 2004.
- [23] W. J. Dally et al., "Deadlock-free Message Routing in Multiprocessor Interconnection Networks," IEEE Trans. Computer, vol. C-36, no. 5, pp. 547-553, May 1987.
- [24] J. Duato, "A New Theory of Deadlock-free Adaptive Routing in Wormhole Networks," IEEE Trans. Parallel and Distributed Systems, vol. 4, no. 12, pp. 1320-1331, Dec. 1993.
- [25] G. Chiu, "The Odd-Even Turn Model for Adaptive Routing," IEEE Trans. Parallel and Distributed Systems, vol. 11, no. 7, pp. 729-738, Jul. 2000.

- [26] C. J. Glass and L. M. Ni, "The Turn Model for Adaptive Routing," *Journal of ACM*, vol. 31, no. 5, pp. 874-902, Sep. 1994.
- [27] C. J. Glass and L. M. Ni, "Maximally Fully Adaptive Routing in 2D Meshes," *Proc. Int'l Conf. Parallel Processing*, I: 101-104, 1992.