

On Chip Packet Routing: An Algorithm for Packet Routing in a Network-on-Chip

Naren V Tikare

Department of Electronics and Communication
Bapuji Institute of Electronics and Communication
Visvesvaraya Technological University
Shamnur Road, Davangere, Karnataka – 577004, INDIA

ABSTRACT

The paper proposes a novel algorithm for a *Network-on-Chip*, which is based on packet switching. Unlike datagram protocol used in large communication network, which can be one of the algorithms in NoC; this paper concentrates on static method of routing of packets. The algorithm described here, uses three different types of packets to achieve communication between various intellectual properties connected to the chip. A packet named route establisher commences the start of packet transfer establishing a fixed route from the source node to the destination node. The packet following route establisher is the data packet, which hops through the same nodes as fixed by the route establisher. When all the data is sent or received, source IP has an option of destroying the link with the destination IP, using route destroyer packet, or might keep it for future communication. A simple prototype using sixteen nodes is used in the design to prove the working of NoC, which further can be expanded to any number of nodes as per the requirement in the design. The proposed algorithm is applicable for a network which is a square mesh topology. The number of nodes in a row should be exactly equal to the number of nodes in a column. The paper also enhances a unique internal architecture of a node in NoC, focusing on the uses of Content Addressable Memory (CAM) as a routing table in a node.

General Terms

Network-on-Chip, Algorithms, Content Addressable Memory.

Keywords

Network-on-Chip, Static routing, Parallelism, Content Addressable memory (CAM) routing table.

1. INTRODUCTION

Due to the huge advancement in the technology, it is estimated that more than a 100 billion transistors will be integrated on a 300mm² die [1]. In order to overcome the ever increasing use of transistors on a chip, a more efficient way of placing them in a die is necessary. When considering power as a main constraint in today's technology, the design paradigm is migrating to many core architectures. The means of communication in NoC can be either circuit switched or packet switched. Packet switching provides efficient and low power means of communication [1, 2]. Network-on-Chips are aimed to provide enhanced performance, modularity, design productivity when compared to traditional communication architectures such as dedicated signal wires and bus architectures. It is seen that Network-on-Chip architectures have overcome a large number of System-on-Chip disadvantages and are also scalable.

Traditionally, communication between processing elements

in SoCs was based on bus architectures [14] with an increased complexity of wires, lead to the main cause for product long time to market. However, for large multiprocessor system on chip's, that too with many different computational as well as Processing elements it is obvious that the bus architectures will soon become a bottleneck from the point of performance, scalability of wires and also power dissipation; in order to overcome this, NoCs have been introduced [6]. We know that the wires can be scaled only to some extent and no attempts can be made in the future to further scale the number of wires [2], [13]. This problem is overcome by NoCs where the problem of interconnection of wires from one IP to another can be scaled to any limit by just reducing the number of nodes in the network. NoC is a new era of communication within a system [5], where the communication between different cores is undergone not by the help of wires but with a special routing technique. It provides efficient communication [16] between all the cores in a SoC [3], [4], [5]. The communication takes place by the help of packets where every packet takes a specific route within NoC for efficient communication between two IP's. As per the design described in this paper, NoC will make use of three different types of packets in order to make or break a connection between different IP's or system cores in a SoC [9]. A new concept of routing technique is been introduced which will make use of a CAM routing table, consisting of the addresses of all the IP's connected to the NoC. It is noticed that, by this technique there is subsequently, less power consumption when compared with the traditional wire architectures [11]. This is interesting because most of the today's NoC designers are struggling for a way to come up with a design of NoC which gives better performance along with reduced power consumption.

There is a huge advancement in the field of NoC; and there are many research centers and private owned companies working for the advancement of the technology in NoC. One of the world's best semiconductor companies working on NoC is ARTERIS [15]. It provides NoC IP interconnects and tools [15] which are used to improve the performance of SoC systems. The NoC IP developed by this company is implemented in large scale in mobiles, multimedia and many other applications. ARTERIS aims at fastest, smallest and very low power interconnects which helps them in simplifying the SoC design. At a press release held on November 9, 2009 ARTERIS announced two new products on NoC IP interconnects namely, Flex way [15] and Flexner [15] packages. These two products are designed so that they address the complete range of different styles on SoC, its complexity and size. The new products are designed to deal with a broad array of interconnection

issues that the present IC designers face. Flexway is designed to specify the needs of simple SoC designs and FlexNoC is designed for higher end complex SoC designs for implementing NoCs. The two IP products designed by *ARTERIS* are built with enhanced design environment and FlexArtist design tool set. The key innovation of the two new products is that they have the ability to use the solutions for traditional bus interconnections as well as NoC's. Another semiconductor company named *SILISTIX* [16] makes use of a technique called GALS (Globally Asynchronous Locally Synchronous) [8], [16] network on chip interconnects. By using this technique in SoC, it can handle deep submicron design closure issues as well as complicated system on chip communication needs simultaneously. By careful design and characterization of NoC components the company assures to give NoC interconnects low power dissipation, area and better performance as predicted. These predicted solutions by *SILISTIX* [16] are used by many of the companies which produce ultra mobile computing, DVR, DTV, smart phones [16] and many other phones for better solutions. In the year 2005, *ST MICROELECTRONICS* [17] introduced a topology called *SPIDERGON* [17]. Company designed this type of topology to provide industry's best price tradeoffs and performance for SoC future devices. In this type of topology all the IP blocks were arranged in a ring fashion and each of the IP block is connected to its counter clock wise or clock wise neighbor in a polygonal ring topology [10]. And also each of the IP block is also connected directly to its diagonal node in the network. This type of topology increases the performance and reduces area size. There are many more industries working on this topic. Here in this paper, a novel algorithm for routing packets is discussed and not prioritizing only the hardware designs. The paper starts by describing the architecture of a simple node and the working of the NoC when all the nodes are connected together. A brief explanation on the methodology is provided to describe the concept more elegantly; followed by the tools and implementation techniques used in the design and then the conclusion.

2. ARCHITECTURE

There is a variety of designs used in maintaining the flow of packets in the NoC. These designs are called as topologies. There are many different ways of connecting nodes which combine together to form a NoC. These topologies have their unique and advantageous style of packet transfer which has been proposed by J. Duato et al, W. J. Dally and many others in a mean time. Few of them are: Mesh, Octagon, Star, Torus, Spin and many more. They are very well explained in [14]. In all these styles, the method used for communication is packet switching and is proved to be more efficient and low power means of communication. In this paper, the design of NoC is carried by the simple Mesh topology and a prototype is a 4x4 Mesh, with constitutes a total of 16 nodes. Attempt has been made to achieve packet switching via various nodes using static methods. Simple node architecture is described in the following.

A node, in its exterior consists of 5 sets of input and output busses, 5 sets of control lines such as strobe and enable, for communication between neighboring nodes and an information provider which is usually a black box which controls the mechanism and internal circuitry of the node. In the interior, a node is equipped with a CAM routing table, an associated RAM memory and a register which is used to split the bits of the packets and perform actions wisely. A total of

four input and output busses, and control lines of a node are connected to its neighboring nodes for subsequent packet transfer and information exchange. Busses are involved in carrying packets and control lines are one bit in size and carry hand shake signals. The left over input and output bus and one set of control lines are connected to an intellectual property. For example it means that a NoC built with sixteen nodes (4x4 mesh) can handle a total of sixteen intellectual properties connected to it.

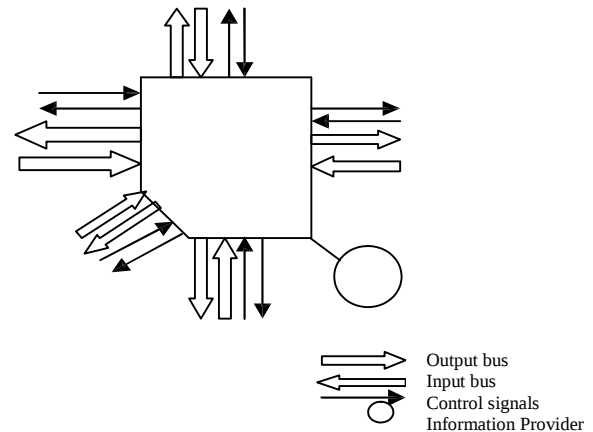


Fig. 1 Architecture of a node

The internal architecture of a node is still more complex which consists of information provider. It has a CAM routing table and a small RAM associated with the routing table. An internal register receives the packet in the node. According to the designed prototype (ex. 4x4 mesh), a NoC is a square mesh with four rows and four columns. Every node is assigned a unique address starting from 0000 to 1111 as shown in figure 2. The addresses are assigned chronologically, and not disturbing the order of sequence. The size of CAM in a node is number of nodes in a row (4) x number of nodes in a column (4). When the system is offline, CAM is updated with the respective node addresses which fall in a row, i.e. The CAM in the first node of the first column has addresses 0000, 0001, 0010, and 0011. The second, third and the forth node in the same row has the same addresses with the same order in their CAM. Likewise, the nodes in the second column has their CAM filed with address 0100, 0101, 0110, 0111 and third column nodes with addresses 1000, 1001, 1010, and 1011 and the nodes in the forth column with addresses 1100, 1101, 1110, 1111 respectively.

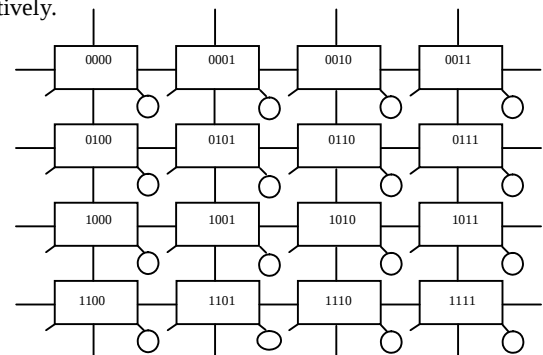


Fig. 2 Architecture of NoC (Square mesh topology) showing the addresses of all the nodes

The three types of packets used for communication have their own unique structure, which is described in the following. The transfer of packets in the network for communication

between two intellectual properties is initialized by a packet called route establisher from the source node. This packet has its header and payload filled with its source node address and the destination address to which the packet is destined. The structure of the packet is shown in figure 3. The type of packet signifies the packet to be a route establisher or a data packet or a route destructor. It is of two bits in size. If type of packet is 01 = route establisher, 10 = route destructor, 11 = data packet. This particular packet links the source intellectual property and the destination intellectual property with a specific route for static transfer of packets.

Source address	Destination address	Type of packet
----------------	---------------------	----------------

Fig. 3 Structure of a Route establisher packet

The data packet follows a route establisher packet. It consists of the source address in its header followed by the data information in its header. The structure of the packet is shown in figure 4. The type of packet = 11.

Data	Source address	Type of packet
------	----------------	----------------

Fig. 4 Structure of a Data packet

The route destructor packet is the last type of packet used by an intellectual property, if the transfer of packets between the intellectual properties to which the route establisher has previously traversed is not necessary. It destructs the route established by the route establisher and delinks the two intellectual properties by carrying zero bits in its payload. The structure of the packet is shown in figure 5. The type of packet = 10.

Zeros	Source address	Type of packet
-------	----------------	----------------

Fig. 5: Structure of a Route Destroyer packet

In every node of a NoC, every move of packet from present node to the next node corresponds to two direction bits. The bits are allocated in the RAM depending on the direction the establisher traverses. If the destination address of the establisher is greater than all the addresses in the CAM then allocated direction bits is 10. If the destination address is smaller than all the addresses in the CAM then direction bits = 00. If the destination address is in the same column, and is greater than the present node address than the direction bits = 11 and if the destination address is in the same column of nodes and is smaller than the present node address than the destination bits = 01. These bits are used along with the source address present in the establisher packet for static routing of the data packets. These bits are stored in the RAM which is present in every node. The source address will act as the address of the RAM and the direction bits will provide information about the next node for the data packet following the establisher. The structure of direction bits is shown in figure 6.

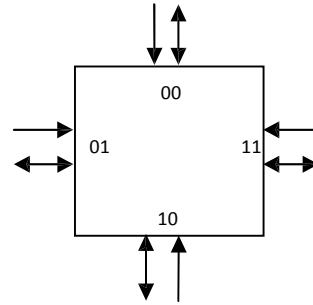


Fig. 6 Assignment of Direction bits in the node

3. METHODOLOGY

NoC is made up of a collection of several nodes connected by various busses and control signals. These nodes have their own internal architecture as described in section I. The operation of the NoC is as follows. Every node in the NoC is assigned a unique address starting from 0000....0 to 1111....1 and the network is assumed to be in square mesh topology. Likewise, every packet in the network is identified by the first two binary bits, which describes the type of packet being a route establisher or a data or a route destroyer packet. During the initiation of packet transfer, source IP sends a route establisher to the node connected to it. Cognition of the packet takes place in the first step where the node identifies it to be a route establisher. The next few bits following the first two bits are the source and destination addresses of the nodes. Every node has a CAM routing table which during offline stores the addresses of the nodes which fall in a row. For example, if the NoC is assumed to be of 4x4 mesh structure, then the routing table of the first node may have addresses 0000,0001,0010,0011 as shown in figure 7. The corresponding nodes in the same row may contain the same addresses in their routing table in chronological order starting from 0000 to 0011. The establisher which enters the node compares its destination address with that of the addresses in the routing table. If the address is found in the particular table, then the destination of the establisher lies in the same row and by just adding 0001 to the present address it can hop from the present node to the next forth coming node. Rather, if the address is not present in the current routing table, the packet assumes the destination to be not present in the particular row, and in turn adds 0100 to the present address to traverse to the next column of nodes in the NoC and the same procedure of comparison follows. As the addresses are assigned chronologically, if the present node address of the packet is greater than the destination address it adds 0001 to the present address, else uses 2's complement subtraction to subtract 0001 to the present address to traverse to the prior node and so on. If the address is not in the same column, an addition of 0100 is added to the present address if the destination address is greater than the highest address in the routing table to move the packet to the next column of nodes; else 0100 is subtracted in 2's complement to move the packet to the previous column. If the node address is equal to the destination address, the packet is received by the intellectual property connected to the node. The process of addition and subtraction of bits with the present address to find the destination address is carried out just by the route establisher packet and not by the data and route destroyer packet. Every time an establisher traverses to a new node, the source address present in the packet is stored in the RAM along with the direction bits. This information is used by the data packets which follow the establisher. The

data packet carries the concerned data information and nothing rather than the source address for routing from source to the destination Intellectual property. The information in the RAM is retained only till the arrival of route destroyer packet, which is sent by the same source IP when all the data packets are sent and the communication between these two IP's is no longer necessary. Route destructor goes on the name of deleting the information that is previously stored by the route establisher in the RAM of all the nodes that the packet has previously traversed. There is a possibility that the route destructor may carry the last flit of data to be delivered to the destination. A clear idea of the routing is understood by the help of the figure 7 which is a NoC prototype built with (4x4) 16 nodes in square mesh topology. The figure shows the internal structure of all the nodes in a NoC. It also shows the complex connection of nodes to one another connected by busses and control signals. In the figure 7, paths of two different packets named Packet 1 and Packet 2 are marked. Considering the case of Packet 1, this enters the NoC through node 0000 whose destination is 1111. Firstly, in the node 0000, it is recognized to be as a route establisher packet by comparing the first two bits of the packet. Then, the destination address present in the packet is compared with all the addresses in the CAM routing table. The destination address of the packet is seen to be not present in the CAM routing table as well as the destination address is greater than all the addresses in the CAM. So, 0010 is added with the present address to move to the next column of nodes. In turn, the RAM in the node 0000 is filed with the source address of the packet along with direction bits 10 (binary) which is recognized by the flow of packet travelling downwards in the network.

Now the establisher is in node 0010. The same procedure is followed with the packet in this node till the packet reaches node 1100. When compared the destination address of the packet with the CAM routing table in the node 1100, the address is seen to be present in the routing table, but the present node is not the destination of the packet. The present address of the node is smaller than the destination address, so 0001 is added to the present node address and the packet is moved to node 1101, storing the direction bits as 11 along with the source address of the packet in the RAM of node 1100. The same procedure is carried out till the packet reaches node 1111 where the packet is retrieved by the intellectual property connected to the node.

Now that the route establisher has stored the information regarding the path from source to destination nodes of the packet, it is an ease to flow any number of packets through the same path by just retrieving the information from the RAM. Data packets follow the establisher and these packets carry the necessary data to be transferred to the destination along with its source address which helps to retrieve the information in the nodes. When the data packets enters node 0000, the packet is identified to be a data packet by the first two bits in the node. Next the source address in the packet is used to find the direction bits in the RAM. The RAM in turn provides information about the direction of the packet and without any calculation and comparison of the bits with the CAM; the data packet is moved to the next node. The same procedure follows in all the nodes, till the packet

reaches node 1111.

Suppose the connection between the two intellectual properties connected to node 0000 and node 1111 is no longer necessary, then the route destroyer packet is sent from the source intellectual property to the destination node which carries zero bits along with the source address. As and on the packet moves through the nodes 0000 till 1111, it clears the RAM allocation contained with the source address and the direction bits and reaches the destination node. Thus delinks the path between node 0000 and node 1111.

The same procedure follows with Packet 2 and all other packets maintaining connections in the NoC. Refer the diagram of the NoC in Fig 7.

4. PARALLELISM

The wire in the nodes of a NoC is shared by many signals. In this design, a high level of parallelism is achieved, as all the nodes in the NoC can operate simultaneously on different types of packets sent by different intellectual properties to a node at the same time. The node processes each packet separately and identifies the packet to be a route establisher, or a data packet or a route destroyer packet and takes decision accordingly. If the packet is identified to be a route establisher, according to the information in the packet two addresses are linked in the CAM and its corresponding information is noted in the memory with a designated address allocating to the particular packet. If the forth coming packet is identified to be another Route establisher packet, the same procedure is followed. At a time, a node can handle any number of connections (not to be mistaken by the number of packets in the node; here is designed to one at a time). Subsequent data and route destroyer packet are handled in the same manner in the network.

5. TOOLS/IMPLEMENTATION

The design of NoC is carried out in a software tool named Xilinx version 13.2. The simulation results have all been observed in simulation software called ISIM. Firstly, the design is tested with various test benches, with at least more than thousand sets of input data. Input data corresponds to the three types of packets where the bits in the packet have been manually entered in the test bench as inputs, then the program is compiled and the corresponding simulation result is been observed. Once the design is working efficiently, it is then implemented in an FPGA, named Spartan 3. The design is well sophisticated with 980000 gates approximately. The specifications of the FPGA are given in [17].

6. RESULTS

A few screen shots have been included from the software to show the flow of packets to and from the node in figure 8 and 9. The labeling of the buses and control signals is indicated. The timing diagram proves the flow of packets from the source to the destination node. Lastly, screen shots of a node and the entire NoC are included in figure 10 and 11.

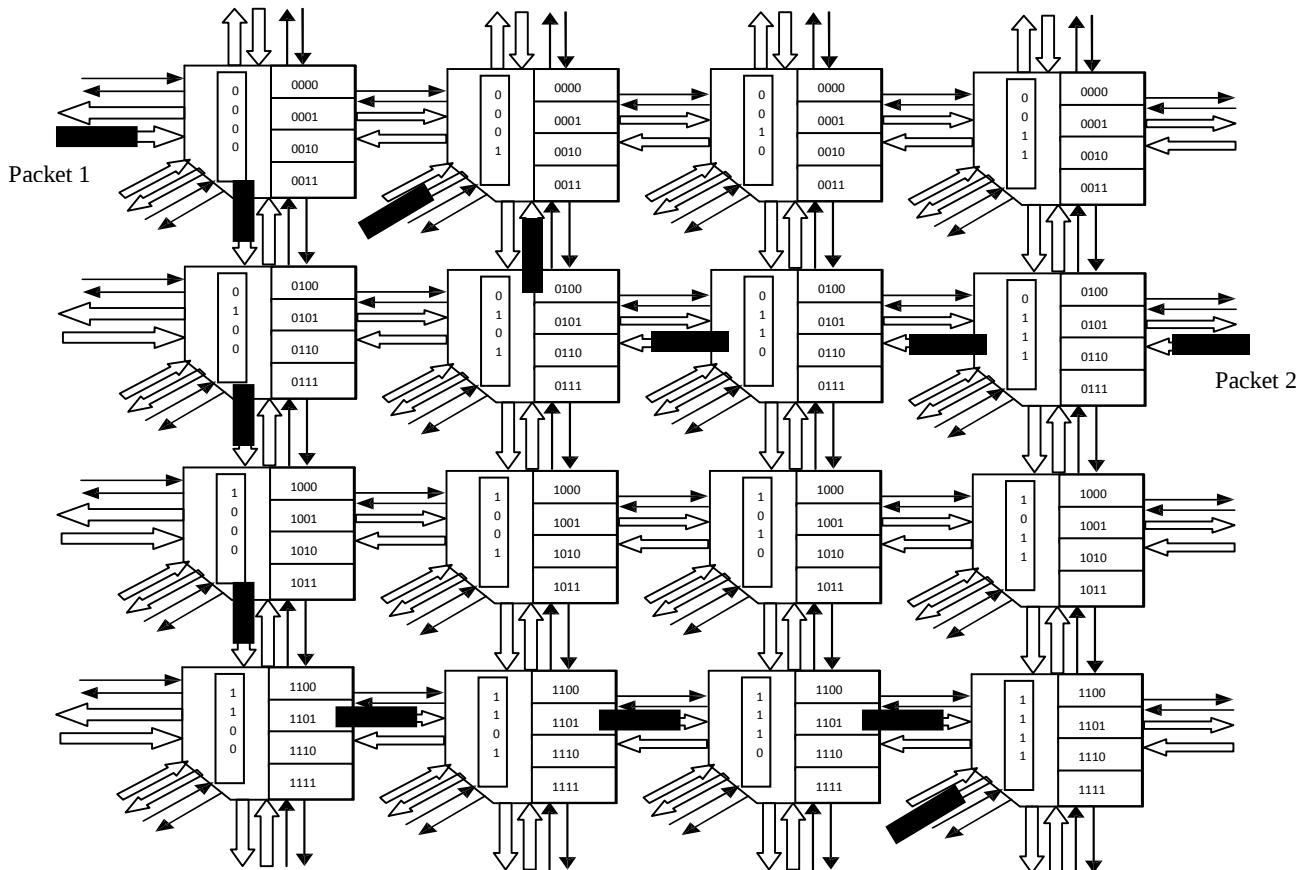


Fig 7: Nodes connected together to form a NoC. Example shows the flow of various packets in NoC. Packet 1 and packet 2 are considered in this figure.

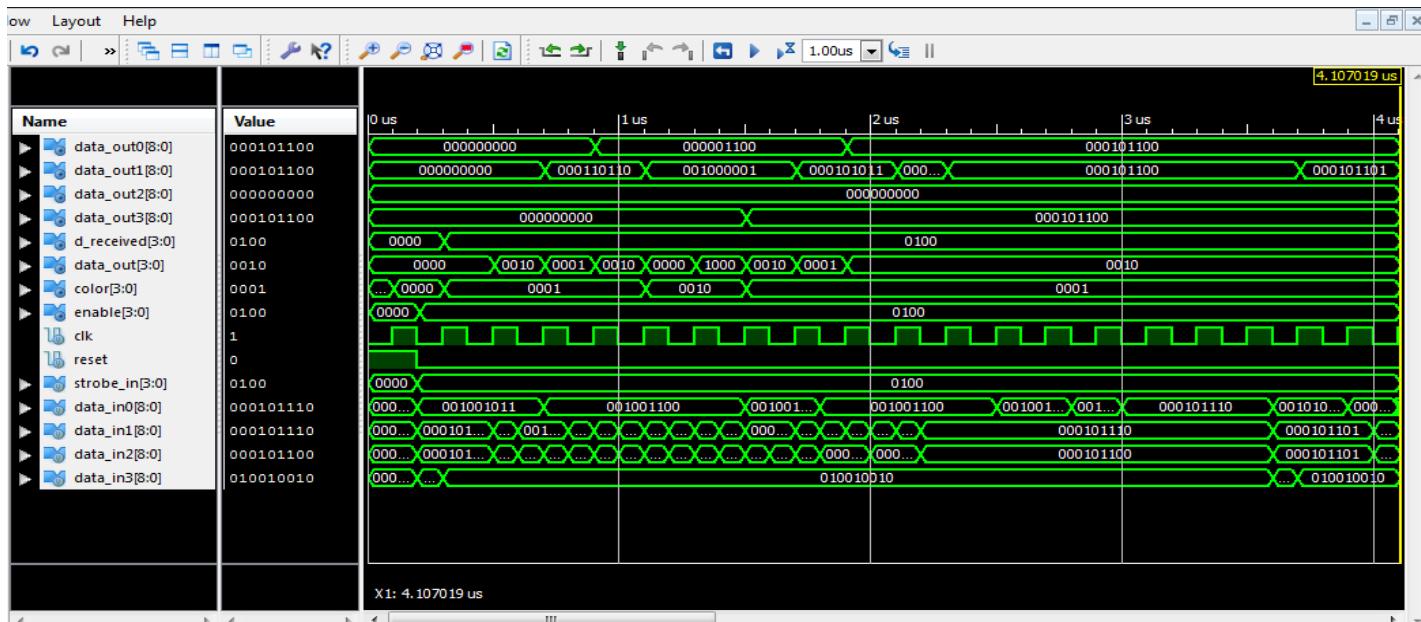


Fig. 8 TEST 1 Timing diagram output of the NoC tested with first set of input values

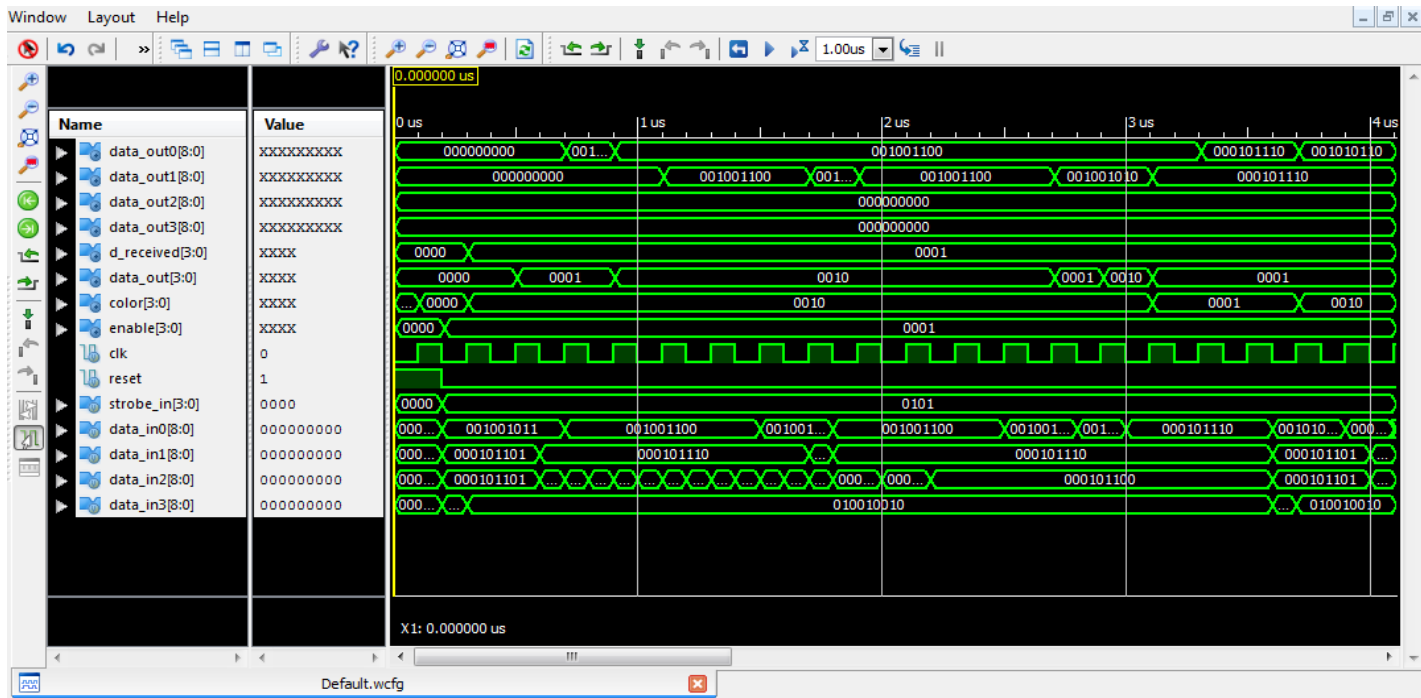


Fig.9: Test 2 Timing diagram output of the NoC tested with first set of input values

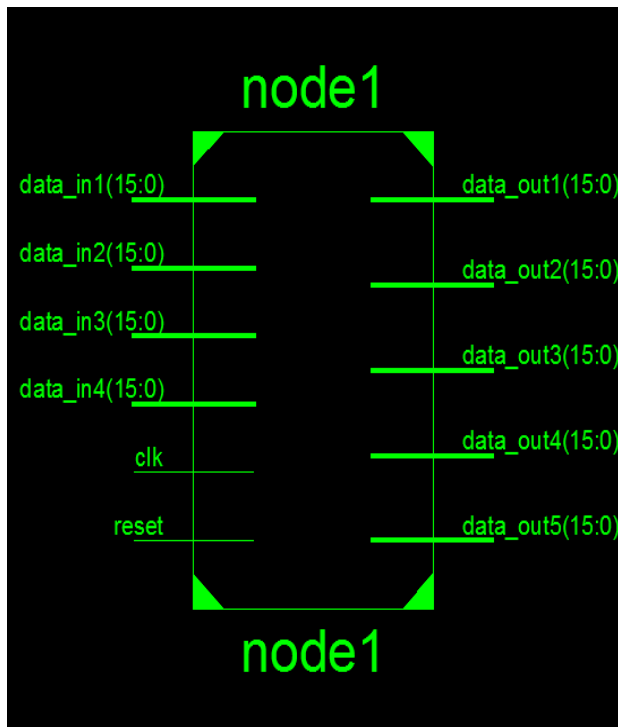


Fig 10: Screen shot of a single node in a NoC

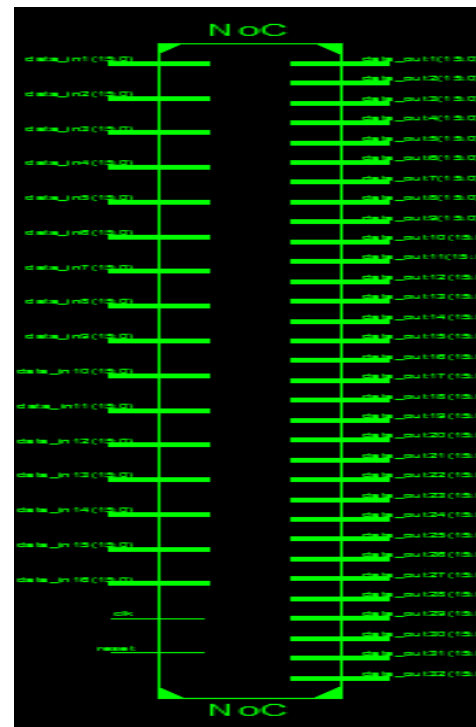


Fig 11: Screen shot of all nodes connected together to form a NoC.

7. CONCLUSION

NoCs are the emerging network solutions for SoC designs and have also tackled many disadvantages of SoCs. The design described in this paper focuses on the algorithm to route packets in the network by static means. It deeply discusses static methods to transfer packets between different intellectual properties. The concept is well described by using 3 different types of packets namely route establisher packet, data packet and route destroyer packet. Every intellectual property connected to the NoC is given a unique address, which is saved in CAM routing table of all the nodes in NoC. When there is a need for data transfer from an intellectual property to another intellectual property connected to the same NoC, the source node introduces route establisher in to the network, followed by data packet; when the data is fully transmitted, source intellectual property sends route destroyer and the connection between source intellectual property and destination intellectual property is erased.

There is a possibility that one NoC can be connected to another NoC; only if the design considerations such as bus width, control signals match each other.

8. REFERENCES

- [1] L. Benini (Ed.), Integration. The VLSI journal, special issue on Network on Chip, volume 38, Issue 1, pages 1-13, 2004.
- [2] W. J. Dally and B. Towles, "Route Packets, Not Wires: On-Chip Interconnection networks" Proc. Design Automation Conf. (DAC), pp. 683- 689, 2001.
- [3] Journal of System Architecture; Volume 50, Issue 2-3, February 2004, Pages 61-63, Special issue on networks on chip.
- [4] The 2nd IEEE International Symposium on Networks-on-Chip, April 2008.
- [5] W.J. Dally and C.L. Seitz, "The Torus Routing Chip," Technical Report 5208: TR: 86, Computer Science Dept., California Inst. of Technology, pp. 1-19, 1986.
- [6] 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.
- [7] S.Kumar *et al.*, "A Network on Chip Architecture and Design Methodology," Proc. Int'l symposium.VLSI (ISVLSI), pp. 117-124, 2002.
- [8] S.Kumar *et al.*, "A Network on Chip Architecture and Design Methodology," Proc. Int'l symposium. VLSI (ISVLSI), pp. 117- 24, 2002.
- [9] F. Karim *et al.*, "An interconnect Architecture for Networking Systems on Chips," IEEE Micro, vol. 22, no. 5, pp. 36-45, Sept. /Oct.2002.
- [10] H. H. Najaf-abadi *et al.*, "Performance Modeling of Fully Adaptive Wormhole Routing in 2D Mesh-Connected Multiprocessors," Proc. Int'l Symp. Modeling, Analysis, and Simulation of Computer and Telecommunications Systems (MASCOTS), pp. 528-534, Oct. 2004.
- [11] Computer Communications, Volume 30, Issue 18, 10 December 2007, Pages 3823-3831 Optical Networking: Systems and Protocols.
- [12] Communication networks; fundamental concepts and key architectures-Alberto Leon and Indira Widjaja, McGraw hill companies, 2001.
- [13] http://www.asic-world.com/verilog/verilog_one_day2.html as read on 18th may 2010.
- [14] Ran Ginosar, "Fourteen Ways to Fool Your Synchronizer," *async*, pp.89, Ninth IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC'03), 2003
- [15] Biberman, A.; Lee, B.; Sherwood Droz, N.; Lipson, M.; Bergman, K.; "Broadband Operation of Nanophotonic Router for Silicon Photonic Networks-on-Chip, "IEEE Photonics Technology Letters, 2010.
- [16] IEEE Computer Society Annual Symposium on VLSI (ISVLSI'02), A Network on Chip architecture and Design Methodology, Pittsburgh, Pennsylvania, April 25-April 26 for NoC architecture.
- [17] Journal of Systems Architecture, Volume 50, Issues 2-3, February 2004, Pages 105-128 Special issue on networks on chip.