International Journal of Modern Engineering Research (IJMER)
www.ijmer.com Vol. 3, Issue. 5, Sep - Oct. 2013 pp-3188-3195 ISSN: 2249-6645

Noise Tolerant and Faster On Chip Communication Using Binoc Model

Arun Arjunan.[1], Karthika Manilal.[2]
[1]PG Scholar, Dept.of Electronics and Communication Engg, TKM Institute of Technology, Kollam, Kerala, India.
[2]Asst.Professor, Dept. of Electronics and Communication Engg, TKM Institute of Technology, Kollam, Kerala, India.

ABSTRACT: *Network on chip (NoC) has become the most promising and reasonable solution for connecting many cores in system on chips (SoC). In conventional NoC architectures neighbouring routers are connected via hard wired unidirectional communication channels. Due to unpredictable and uneven traffic patterns in NoC one of the channels may be overflowed due to heavy traffic in one direction while the other unidirectional channel is idling and thus causing inefficient resource utilization, data loss and degradation in performance. So as a remedy for this situation a bidirectional NoC (BiNoC) can be used, which uses bidirectional channels to connect adjacent routers and it also supports runtime reconfiguration of channel direction according to traffic demand by using channel direction control (CDC) protocol. Since data communication through Network on Chip is susceptible to noise due to the presence of various noise sources, the incorporation of a hybrid error control scheme in which combined approach of error correction and retransmission is used which increases the reliability of the system. This architecture will allow the NoC structure to handle massive data transmission by effectively increasing the communication bandwidth, resource utilization capability and speed of NoC communication together with the increased reliability. The architecture is modelled using VHDL.*

I. INTRODUCTION

With vigorous advancement in semiconductor processing technologies, the chip integration has reached a stage where a complete system can be placed in a single chip. A system on chip (SoC) is an integrated circuit (IC) that integrates all components of an electronic system into a single chip. Applications of these systems are in the area of telecommunications, multimedia, and consumer electronics where it has to satisfy real time requirements. As technology scales toward deep sub-micron, mare and more number of computational units will be integrated onto the same silicon die, causing tight communication requirements on the communication architecture. Due to this fact the traditional solution for inter core communication in SoCs such as shared bus systems and point to point links were not able to keep up with the scalability and performance requirements. As a result "Network on Chip" (NoCs) emerged which has some reasonable and promising features for application to giga-scale system on chips such as modularity, scalability, high band width availability, despite the increased design complexity. NoC consists of components such as IP Cores, Network Interface (NI), and Routers or switch which routes the packets of data through the Network according to a routing algorithm and interconnecting channels or wires. Packets of data to be communicated through the NoC are transmitted through the NoC via routers and channels to reach the destination IP core from the source IP core of the SoC.

The city block style, tiled NoC architecture is the most popular type of NoC and is considered in most of designs due to its flexibility, simplicity, scalability and performance advantages. In this type of architecture, the wires and routers are arranged like street grids of a city, while the resources (logic processor cores) are placed on city blocks separated by wires. Here neighbouring routers are connected together using a pair of unidirectional communication channels where each channel is hard-wired to handle either outgoing or incoming traffic only. At run time quite often one channel may be overflowed with heavy traffic in one direction, while the channel in the opposite direction remains idling. This leads to performance loss, inefficient resource utilization, reduced throughput of the system and wastage of bandwidth in City Block style Network on Chip architectures. As a solution for these problems the concept of reversible lanes in city traffic can be implemented in Network on chip (NoC). A counter flow lane is one in which the driving directions are changed using some electronics signs in-order to provide high capacity to the direction with heavier traffic volume. Such a Network on chip (NoC) which implements the idea of reversible lanes in city traffic to configure the direction of channel according to the traffic inside the system is termed as Bidirectional Network on Chip (BiNoC) [1]. The channels are made dynamically self reconfigurable at real time by using a protocol called as Channel direction control protocol (CDC). This project targeted the Bidirectional NoC having dynamically self reconfigurable channels to reduce the limitations of conventional NoC architecture such as inefficient resource utilization, reduced channel bandwidth availability for massive unidirectional communication and it also increases the effective throughput of the network on chip without using additional channels for interconnection.

II. RELATED WORKS

By borrowing the ideas from real world computer networks in the chip level communication issue the concept of Network on chip evolved. Network-on-Chip (NoC) provides scalable bandwidth requirement where number of simultaneous bus requesters is large and their required bandwidth for interconnection is more than the bus based design [7][8]. When we are moving to the era of nano scale technology one of critical issues is a wiring delay. While the speed of basic elements such as gate delay becomes much faster, the wiring delay is growing exponentially because of the increased capacitance caused by narrow channel width and increased crosstalk [10]. In designing NoC systems, there are several issues to be concerned with, such as topologies, switching techniques, routing algorithms, performance, latency, complexity and so on. Mesh Topology is a Feasible topology and is easily expandable by adding new nodes [1][6][9][3]. A routing algorithm

determines the direction of Packet Transmission in NoCs [2][4]. The algorithm determines to what direction packets are routed during every stage of the routing. XY routing is a dimension order routing which routes packets first in x- or horizontal direction to the correct column and then in y- or vertical direction to the receiver [4][6].The performance of NoC communication architecture is dictated by its flow-control mechanism. Wormhole flow control has advantages such as less memory requirement and less latency. In wormhole routing packets are divided to small and equal sized flits (flow control digit or flow control unit). . After first flit the route is reserved to route the remaining flits of the packet. This route is called wormhole.

In conventional tiled NoC structures neighboring routers are connected using hardwired unidirectional communication channels which can handle data only in a single direction i.e. either output data or input data [1]. While considering the case of basic NoC structure one channel may be overflowed with heavy unidirectional traffic, while the channel in the opposite direction remains idling since the channels or links that connect the neighboring routers together are hardwired in such a way to handle traffic in only one particular direction [2]. This causes performance degradation and inefficient resource utilization in traditional city block style NoC architectures. Thus Traditional tiled NoC structures are not up to the mark in handling a heavy flow of traffic in single direction although it is equipped by resources to handle the situation.

As we move to consider Deep Submicron NoCs (DSM NoCs), communication becomes unreliable because of the increased sensitivity of interconnects to on-chip noise sources, such as crosstalk and power-supply noise [5].In DSM SoCs; low swing signaling reduces signal-to-noise ratio thus making interconnects more sensitive to on-chip noise sources such as cross-talk, power supply noise, electromagnetic interferences, soft errors, etc [5]. A common practice to increase the reliability of NoC is to incorporate error detection and correction schemes into the design. . Hamming codes [5] are the first class of linear codes devised for error correction and have been widely employed for error control in digital communication and data storage systems. When the transmitted codeword is received, an error detecting stage checks the parity bits. If a correction stage is applied, the exact location of the error can be identified so that the corrupted bit can be restored. A distance- 3 Hamming code can be easily modified to increase its minimum distance to 4, adding one more check bit, chosen so that the parity of all of the bits, including the new one, is even to form Single Error Correction and Double Error Detecting hamming Code (SECDED)[5]. This version of the Hamming code is traditionally used for single error correction and double error detection.

In this Paper a noise tolerant and faster on chip communication is proposed using bidirectional Network on Chip (BiNoC) having dynamically self reconfigurable channels. The noise toleration capability of the system is to be increased by using a hybrid scheme of error detection/correction and retransmission schemes. The packet switched Bidirectional NoC prototype design considers the design constraints such as two dimensional mesh topology, XY routing algorithm, and wormhole flow control mechanisms.

III SYSTEM ARCHITECTURE

Network on Chip provides the infrastructure for the communication in multicore single chip systems. A NoC consists of resources and switches that are connected using channels so that they are able to communicate with each other by sending messages. Here the topology used is mesh, which is a simplest layout; also routing in a two-dimensional mesh is easy. [2]. Figure: 1 shows the basic structure of a tiled, city block style network on chip. The inter router communication in conventional NoC design is accomplished by using two unidirectional communication channels that are hardwired for handling traffic only in a single direction. This causes problems such as performance degradation and ineffective resource utilization in conventional tiled NoC architectures. In a BiNoC, each communication channel allows itself to be dynamically reconfigured to transmit flits in either direction [2]. This promises better bandwidth utilization, lower packet delivery latency, and higher packet consumption rate for the network. The flow direction at each channel is controlled by a channel-direction-control protocol (CDC protocol).Figure: 2 shows the basic block diagram of a 3*3 grid array structure of Bidirectional NoC in which routers are interconnected together using bidirectional channels.

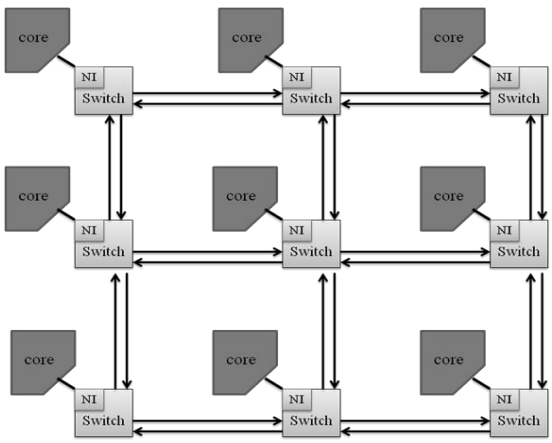


Figure:1 Basic structure of 4*4 array of NoC.

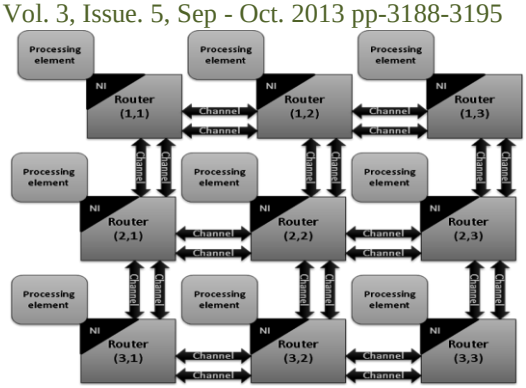


Figure: 2 Block diagram of 3*3 2D-Mesh BiNoC

A. Existing Conventional NoC router Architecture

The network on chip routers which are connected together using communication links are the most vital part in a NoC design. These routers are responsible for the correct switching of data to the destination resource by using a routing protocol. The architecture of a basic tiled network on chip is shown in Figure: 3. The Router used in basic city block style NoC will have five ports corresponding to the directions North, East, South, west and a local PE(Processing Element). Each port is having its input and output channel, and each input and output channel is having its control and decoding logic, which supports five parallel connections at the same time simultaneously. The input channel consists of three parts i.e. FIFO, FSM, and XY logic. The FIFO is used as input buffer to store the data temporarily. XY Logic is used for comparing the coordinates stored in header flit with the locally stored coordinates and thus finds out the direction to which the packet has to be switched. There is a switch present inside the router which switches the data from the input port of a particular direction of router to the output port of the desired location calculated by the XY routing algorithm. An Arbiter is used in output channel to overcome the problem of multiple input requests coming at single output port. Arbiter is based on rotating priority scheme in which each port get reduced its priority once it has been served.

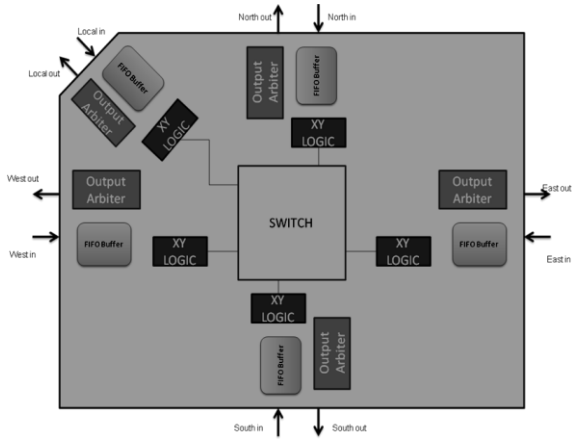


Figure: 3 Five Port Router in Basic Tiled NoC

B. BiNoC Router Architecture

Modifications to the five port router structure when used in Bidirectional network on chip are:

1. All the ports of router will be bidirectional ports.
2. Buffers should be placed in every port to store data temporarily.
3. FIFO buffer capacity should be made higher to accommodate heavy traffic.
4. A high priority finite state machine (HP FSM) and a low priority finite state machine (LP FSM) will be connected at each port to dynamically configure the channel direction according to the traffic needs.

Architecture of five port router used in BiNoC is shown in Figure: 4.Which has the capability of dynamically self reconfiguring the directions of its bidirectional port according to the traffic needs. The direction of data transmission of a bidirectional channel needs to be self-configured, at run time based on local traffic demands. To achieve this goal a distributed CDC (Channel direction Control) protocol is used. Configuration of a bidirectional channel direction is controlled by a pair of FSMs in the channel control blocks of the routers at both ends. Opposite priorities are assigned to FSMs on the other channel of same pair of adjacent routers. The operations of adjacent FSMs synchronized against a common clock.

The two FSMs exchange control signals through a pair hand-shaking signals: input−req (input request) and output−req (output request). When the sending end router has data packet to transmit the output−req is made '1'. The output−req signal from one router becomes the input−req signal to the FSM of the other router. Each FSM also receives a channel−req (channel request) signal from the internal routing computation module as there is a grant given to a particular port to output data. channel−req = 1 when a data packet in the local router is requesting the current channel.

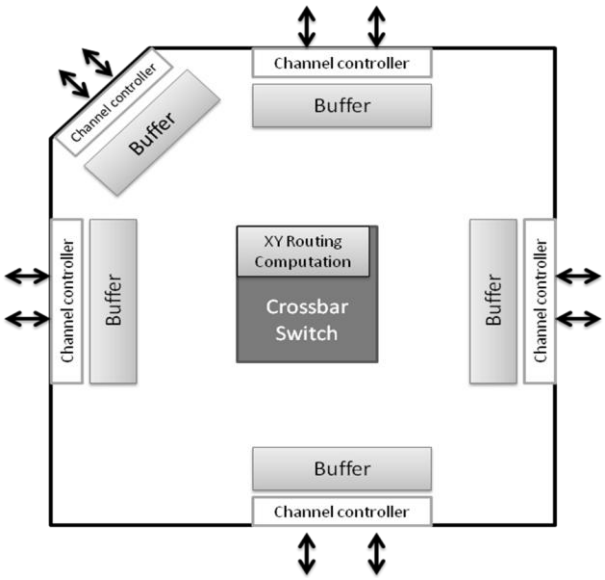
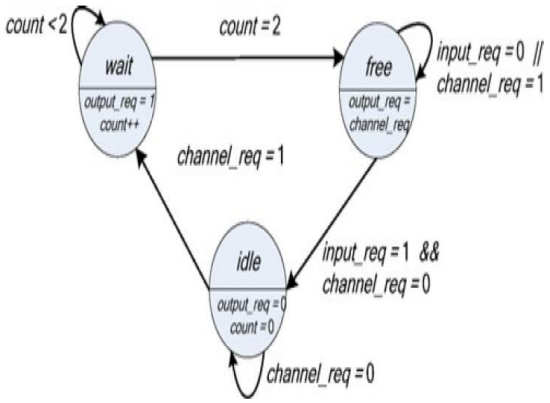


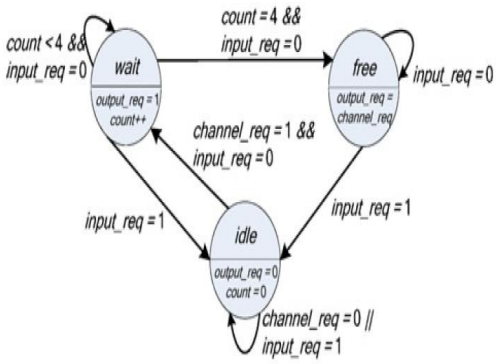
Figure: 4 Architecture of BiNoC Router

The state transition diagram of high priority FSM and low priority FSM are shown in Figure: 5. Each FSM consists of three states: Free, Wait, and Idle, defined as:

1. Free State: the channel is available for data output to the adjacent router.
2. Idle State: the channel is ready to input data from the adjacent router.
3. Wait State: an intermediate state preparing the transition from the idle state with an input channel direction to the Free State with an output channel direction.



(a)



(b)

Figure: 5 state transition diagram of HP and LP FSMs
The inter router channel direction control scheme is shown in figure: 6.

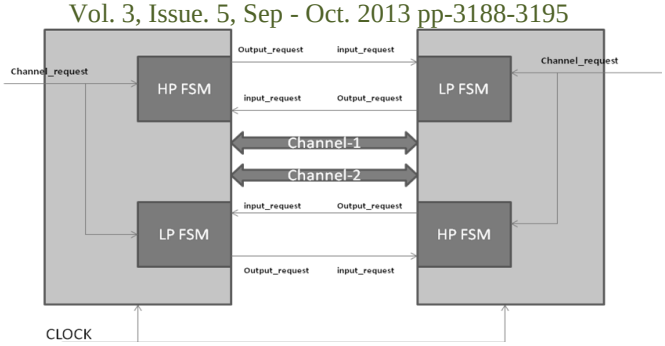


Figure: 6 channel Direction control scheme(CDC)

C. Flit Structure

As this project uses worm hole flow control mechanism, the data packet will be divided into equal sized flits (flow control units). A packet of data will contain a header flit having the information of source, destination PEs and if needed it may contain length of packets [9] followed by the data flits which carry the actual data to be communicated and finally there will be a tail flit which indicates the completion of packet. In this proposed system the flit structure is considered in the way that the first bit shows the flit to be the header-trailer or the data. When the first bit equals one, this flit is a header or trailer. In this case, the 2nd bit determines which one is the header and which one is the trailer. The data flit will have first bit as zero. The sizes of flits considered in this project are of 16 bits each.

(a)

(b)

(c)

Figure 7 (a) Header, (b) Tail and (c) Data Flit structures

D. Error Control

Hamming codes are the widely used error detection and correction code in network on chip due to its simplicity, short timing delay and less area overhead. SEC-DED Hamming code is used here for error detection and correction of errors. A retransmission scheme in which the receiver acknowledges the sender to transmit the data again is also implemented in NoC designs to increase the reliability. For Retransmission scheme the Network Interface (NI) of sender will be having additional Retransmission Buffers which will store the packets that have been transmitted.

When considering the switch to switch and end to end error control policy [7], end to end error control policy has less latency and area overhead. So in this project an end to end and hybrid scheme of error detection, correction and retransmission are considered for increasing the reliability of the NoC communication. Since wormhole flow control mechanism is been used in this project error detection and correction will be occurring in flit by flit basis so it assures more reliability than packet by packet error control mechanism.

Flit size considered is 16 bit so for a SEC-DED hamming code will have 5 check bits and one overall parity bit. he codeword generated by the SEC-DED Hamming encoder (22, 16) will be of 22 bits containing 16 data bits 5 check bits and one overall parity bit.Here p1,p2,p4,p8,p16 are the check bits occupying the 2^n positions of codeword while p is the overall parity bit of the data to be transmitted. Structure of codeword generated by SEC-DED hamming encoder is shown in Figure 8.

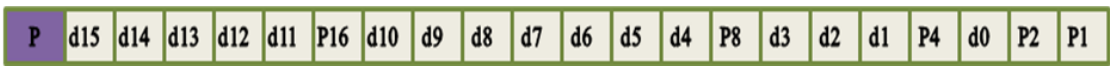


Figure: 8 structure of (22, 16) SEC-DED hamming codeword

Hamming decoder is more complex than encoder. The Hamming Decoder block recovers a binary message vector from a binary Hamming codeword vector. (22, 16) SEC_DED Hamming decoder takes 22 bit codeword and it calculates the syndrome and overall parity and decides whether an error has occurred or not in a transmitted code word. Since the error control scheme used in this work is a hybrid scheme of error correction and retransmission in which if certain number of uncorrectable errors are found to be occurred in the received data packet then the hybrid SEC-DED decoder has to generate the retransmission request.

IV. SIMULATION RESULTS

The modules are modeled using VHDL in Xilinx ISE Design Suite 12.1 and the simulation of the design is performed using Modelsim SE 6.2c to verify the functionality of the design. Analysis of the results obtained is discussed in this section.

In Figure 9, five port router used in conventional NoC designs is shown. The five port router designed here is the

one used in basic tiled NoC architecture. The router consists of five input and five output ports. The structure of router is shown in Figure: 3.6. The routing decision is taken by the router according to the XY routing logic present in the router module. The 8 bits from 3rd bit of the header flit is analyzed to find out the address of destination router and it is compared to find the direction of switching inside the router. Switching is made into effect by a crossbar switch inside the NoC router. There may be multiple requests for one particular output port at a time, but when using an arbiter only one request is served once. Since Wormhole flow control is used in this router once a request is granted then the data from that port is switched till a tail flit is encountered.

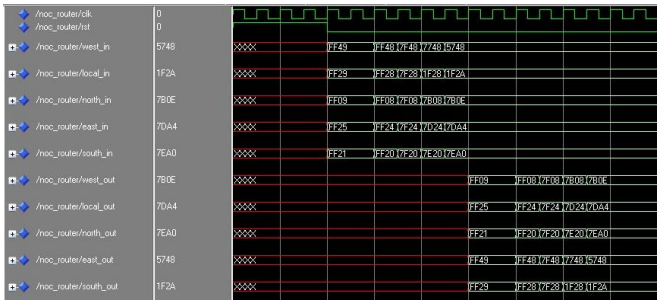


Figure:9 Five port Router Simulation

In Figure:10 the simulation results of high priority and Low priority FSM are shown which dynamically configures the direction of bidirectional channels that interconnects the adjacsent routers in BiNoC structure.

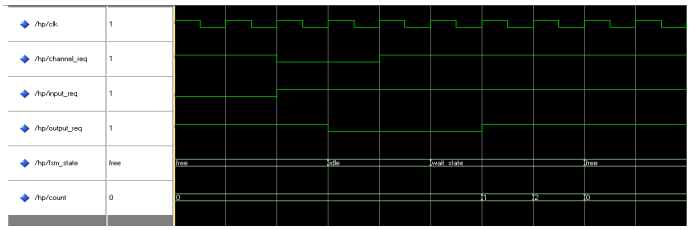


Figure:10a Simulation result of HP FSM

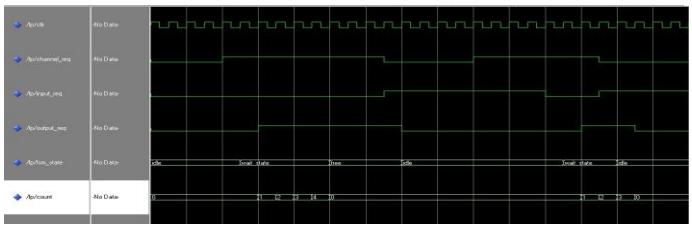


Figure:10b Simulation result of LP FSM

Simulation result of BiNoC router is shown in figure: 9 which show the switching of data flits through bidirectional ports of the router. The direction of bidirectional ports are configured according to the distributed CDC protocol consisting of high priority and low priority FSMs at run time.

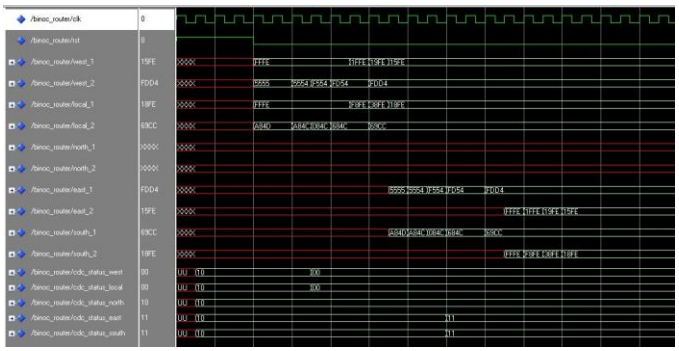


Figure: 11 Simulation result of BiNoC Router

Figure:12 shows the simulation result of Single Error Correction-Double Error Detecting (SEC-DED)Hamming encoder. The hamming encoder designed here is a (22, 16) SEC-DED hamming encoder. Encoder inputs are 16 bit flits and its output is 22 bit codeword consisting of check bits, data bits and an overall parity bit.

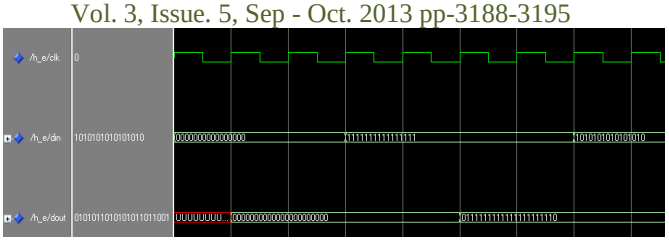


Figure: 12 Simulation result of Hamming Encoder

Decoder designed here is a SEC-DED hamming decoder with forward error correction and retransmission. Decoder module takes 22 bit codeword and calculates the syndrome as well as overall parity and makes decisions.

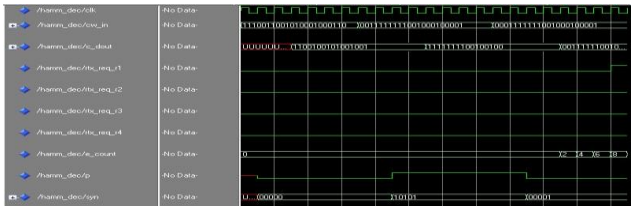


Figure: 12 Simulation result of Hamming Encoder

Figure 13 shows the simulation result of 2*2 BiNoC with hybrid error control scheme. Here communication is established between router having address (1, 1) and router having address (2, 2) via router having address (1, 2). Here the channels and buffers will be having more width to accommodate the hamming encoded flits than normal BiNoC. The Header and data flits are injected in the router R1 (1, 1) and the flits are switched through the network by using both the channels in the path. At the receiver end i.e. R4 (2, 2) the encoded flits are decoded by the hybrid decoder and if single bit errors are detected then it is corrected and if two bit errors are occurring then errors are counted and a Retransmission Request signal is made HIGH which reaches NI of the sender router.

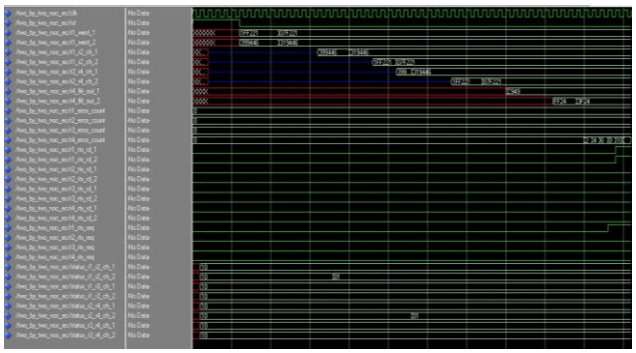


Figure: 5.13 Simulation result of 2*2 BiNoC with Hybrid Error Control Scheme

Table: 1 shows the comparison of flit switching speed and resource utilization of Basic NoC router and BiNoC router. Flits where injected into the west port of routers and switching was done to the ports in east direction for both the routers.

Type of Router	No of Clocks	No of flits switched	No o f channels configured
Basic NoC Router	50	44	1
BiNoC Router	50	84	2

Table: 1 Flit Switching Speed Comparison of Basic NoC Router Vs BiNoC Router.

Table: 2 shows the details of clock cycles, channels used and number of flits switched by the 2*2 Bidirectional NoC when a communication was established form Router R1 (1, 1) to R4 (2, 2).

2*2 BiNoC	No of clocks	No of flits reaching destination	No of channels used
Communication from R1 to R4	100	146	4

Table: 2 BiNoC (2*2) flit switching from R1 (1, 1) to R4 (2, 2).

V. CONCLUSION

This paper focuses on the development of Bidirectional Network on Chip prototype for faster on chip communication than conventional city block style NoC. The system also incorporates a noise toleration scheme by using SEC-DED hamming code and for increasing the reliability, a hybrid scheme of error detection/correction and retransmission of packets is employed in the system. Wormhole routing mode based basic 5 port NoC router and 5 port BiNoC router having bidirectional ports which are dynamically configurable to either direction by using CDC protocol are designed and simulated. Routing algorithm considered is XY routing algorithm. Comparative analysis of Basic NoC router with BiNoC reveals that the flit switching speed and resource utilization capability of BiNoC router is almost double for massive unidirectional communications. 2*2 BiNoC was developed by interconnecting the routers together in mesh topology and communication through network based on XY routing algorithm and wormhole flow control was established. The error control policy considered in this work is End to End Error Control Policy since its area overhead and delay overhead for packet transfer are less. For error correction and detection SEC-DED hamming code is used which can correct single bit errors and detect two bit errors. By using BiNoC having hybrid scheme of error control a faster and reliable on chip communication is achieved. The whole architecture is designed using VHDL, synthesized using Xilinx ISE 12.1 and simulated using ModelSim 6.2c simulator.

REFERENCES

[1] Y. C. Lan, S. H. Lo, Y. C. Lin, Y. H. Hu, and S. J. Chen, "BiNoC: A bidirectional NoC architecture with dynamic self-reconfigurable channel," *IEEE Trans. Comput.-Aided Des,*vol. 30, No. 3, Pp. 421–434, March 2011.

[2] Wen-Chung Tsai, Ying-Cherng Lan, Yu-Hen Hu, and Sao-Jie Chen "Networks on Chips:Structure and Design Methodologies" Hindawi Publishing Corporation Journal of Electrical and Computer Engineering, Volume 2012, Article ID 509465

[3] Ankur Agarwal, Cyril Iskander, Ravi Shankar, "Survey of Network on Chip (NoC) Architectures & Contributions" journal in engineering, computing and architecture, volume 3, issue 1, 2009

[4] Ville Rantala, Teijo Lehtonen, Juha Plosila, "Network on Chip Routing Algorithms "TUCS Technical Report No 779, August 2006

[5] Davide Bertozzi, Luca Benini, Giovanni De Micheli, "Error Control Schemes for On-Chip Communication Links: The Energy–Reliability Tradeoff", IEEE Transactions on Computer-Aided Design Of Integrated Circuits and Systems, Vol. 24, No. 6, June 2005.

[6] M.Mirza Aghatabar, S.Koohi, S. Hessabi, M.Pedram, "An Empirical Investigation of Mesh and Torus NoC Topologies Under Different Routing Algorithms and Traffic Models" Sharif University of Technology, Tehran, Iran, University of Southern California, CA, USA

[7] Erno Salminen, Ari Kulmala, and Timo D. H¨am¨al¨ainen "On network-on-chip comparison" Erno Salminen, Ari Kulmala, and Timo D. H¨am¨al¨ainen Tampere University of Technology, Finland.

[8] Dan Marconett "A Survey of Architectural Design and Implementation Tradeoffs in Network on Chip Systems", University of California, Davis, CA 95616, USA

[9] Tobias Bjerregaard and Shankar Mahadevan "A Survey of Research and Practices of Network-On-Chip" Technical University of Denmark

[10] Jun Ho Bahn "Overview of Network-on-Chip"