SCALABLE AND LOW POWER NETWORK ON CHIP DESIGN FOR RECONFIGURABLE SYSTEMS

A

THESIS

SUBMITTED TO

GALGOTIAS UNIVERSITY

GREATER NOIDA

IN FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN

ELECTRONICS ENGINEERING

By

Himani Mittal

12SET503002

School of Electrical, Electronics & Communication Engineering

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis, entitled **Scalable and low power network on chip design for reconfigurable systems**, in fulfilment of the requirements for the award of the degree of Doctor of Philosophy in Faculty and submitted in Galgotias University, Greater Noida is an authentic record of my own work carried out during a period from January, 2013 to December, 2017 under the supervision of Dr**. Yogendera Kumar**.

The matter embodied in this thesis has not been submitted by me for the award of any other degree of this or any other University/Institute.

Himani Mittal

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Dr. Yogendera Kumar
Supervisor

The Ph.D. Viva-Voice examination of _____ Research Scholar, has been held on _____.

Sign. of Supervisor Sign. of External Examiner

ACKNOWLEDGMENT

I would like to give my thanks to Chancellor of the Galgotias University, Mr. Sunil Galgotia, for providing me such a good academic environment, labs, libraries etc, that helped a lot directly/indirectly, throughout my thesis work.

I would like to express my thanks to Dean Dr. Siba Ram Khara for his support and corporation.

I am very grateful and deeply indebted to my honourable mentor and guide Dr. Yogendera Kumar, VLSI Division, School of Electrical, Electronics and Communication Engineering, Galgotias University. It is my great pleasure to express my deepest regards and whole hearted indebtedness to him for his inspiring encouragement, continuous guidance, active cooperation, constant supervision, valuable suggestions, constructive criticism and help in carrying out this work successfully. I consider it my good fortune to have got an opportunity to work with such a wonderful person.

I express my gratitude to my parents and family, who always inspired me for higher studies and are my constant source of inspiration and encouragement in every step of my life.

I also express my heartiest gratitude to my friends for their invaluable advice and encouragement.

(Himani Mittal)

12SET503002

ABSTRACT

In the recent times reconfigurable systems have emerged as an alternative to the customized IC solutions. As a result the electronics industry is shifting towards using reconfigurable devices such as Field Programmable Gate Arrays (FPGAs) as the computing platforms. Reconfigurable computing can also be viewed as a trade off between general purpose computing and application specific designs. Since reconfigurable systems provide the architecture and design flexibility, they have increased the progress in the hardware software co design technology and as a result finding applications in various areas such as scientific and biological computing, artificial intelligence, signal processing, security computing and control oriented design etc.

As the number of components in reconfigurable systems increases, interconnecting all these components is becoming more and more challenging. Therefore, a more structured high level interconnect able to reduce the design effort and improve hardware efficiency is becoming necessity for these systems. To overcome theses interconnect challenges, various intra-chip communication techniques such as point to point and bus based systems were applied. But these techniques showed the lack of scalability. As a result, Network on Chip (NoC) has arisen as a more efficient and scalable solution to this problem. In a NoC-based system, packets are transmitted from a source to a destination PE (processing element) via routers (switches) arranged by links. The arrangement of interconnections between the PEs determines the topology of the NoC based system.

The reconfigurable hardware solutions are based on computing architectures able to adapt their behaviour in response to several different inputs. Moreover, efforts are made to apply the reconfiguration techniques to all level of computing systems, for intrachip communication structures (buffers, topologies, memories, routers, etc.) as well as processor's micro architecture (instruction set, data path, cache hierarchy etc.). Keeping in mind the above mentioned issues current thesis work has been carried out. A scalable and low power NoC for reconfigurable systems is designed in this thesis. Stress has been put in designing the reconfigurable and power efficient components of the NoC like routers and links. For this purpose Verilog hardware

description language is used for the design entry and Modelsim 10.3 for simulation and Xilinx ISE design suit is used for the synthesis of routers and links. Spartan-6 FPGAs are used for the implementation of these routers and links. Topology selection has been done with the help of various performance metrics like maximum end-to-end latency, dropping probability and throughput etc. 2-D Mesh is found suitable for this work from these analyses. Four reconfigurable and power efficient routers namely Reconfigurable Router to Improve Efficiency (RRIE), Heterogeneous Reconfigurable Router for Low Power and High Performance (HRRLPHP), Eight Directional Reconfigurable Router for High Throughput (EDRRHT), Smart Reconfigurable Router for Online Detection of Faulty Blocks (SRRODFB) and three high performance, low power links namely High Speed Low Power Link for reconfigurable systems (HSLPL), Link to Minimize Switching Transitions for Reducing Dynamic Power for reconfigurable system (LMSTRDP and Link by Multi-encoding to Reduce Transition Activities in reconfigurable systems (LMRTA) have been designed and tested in the present thesis.

The first designed router is RRIE which has a FIFO, channel flow/control logic and a crossbar as a switch. Compared to the earlier designed reconfigurable routers in the literature, this router has low area, consumes less power and provides high performance. It dissipates only around 15mW of power and occupies 64% smaller area compared to the earlier designed homogeneous routers. The problem with RRIE is that it does not support the heterogeneous data. This problem is removed in the second router HRRLPHP. To reduce the area further two tag bits have been added in this router which are able to indicate the direction of the output from the crossbar. The control circuitry of this router has been improved by including features like acknowledge for writing, acknowledge for reading, request and grant signals for writing in another channel's FIFO. The modified channels, FIFO and crossbar are designed and tested and it shows the significant improvement in area and power dissipation. Area of around 53 mm^2 and power dissipation of only 0.020 w is estimated in this case. The drawback of this router is that the throughput is degraded marginally (3%). This drawback is removed in our third designed router namely EDRRHT. Also, semi-buffer concept is introduced in EDRRHT. Four more directions (NE, NW, SE & SW) are added so that router is now able to support eight directions. It also supports buffer-less storage concept to store the data of four newly added directions. Data of newly added directions share FIFOs with their neighbours to store the data. This way, this router requires less area for implementation. The newly

designed architecture is able to provide more routes to the router for data routing. Thus it is helpful in decreasing critical path length and helps in increasing the performance. Also, rectilinear stennier tree path has been used to shorten the data path of the router. Furthermore, a circular FIFO is used in place of the normal fall-through FIFO. Although power dissipation of 0.045 w and area of 85 mm^2 are slightly higher than the previous router, it is still better as compared to other routers in terms of other features and performance.

In addition, a router which is able to recognize and differentiate between permanent and transient errors and the way to handle them and to handle the spotting of faulty blocks of a NoC is also designed in the present thesis. It is given the name SRRODFB and is a smart reconfigurable router for online detection of faulty blocks. It is also able to enhance the throughput, the network load and the data packet latency. In place of buffers, FIFO is used to achieve better performance. Hence it is able to have low area, low power dissipation and high performance compared to the other designed routers in this work as well in literature.

In the proposed HSLPL link, MUX gating technique is used which is able to reduce the power dissipation as well as latency. In the second link namely LMSTRDP, we have reduced the number of transitions with the help of encoder and decoder. Power dissipation has been further reduced in this case. In the third link namely LMRTA, multi-coding technique has been used to reduce the transition activity. The rationale behind these proposed schemes for links is to minimize not only the switching activity but also coupling switching activity which is mainly responsible for link power dissipation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

List	Full Form
NoC	Network-on-Chip
SoC	System-on-Chip
SIP	System in package
SOB	System on Board
PE	Processing Element
IC	Integrated Circuit
RTL	Register Transfer Level
SAF	Store and Forward
VCT	Virtual cut through
WH	Wormhole Switching
CMOS	Complementary Metal Oxide Semiconductor
XPA	Xilinx Power Analyser
PCI	Peripheral Component Interconnect
FPGA	Field Programmable Gate Array
VLSI	Very Large Scale Integration
IP	Intellectual Property
ASIC	Application Specific Integrated Circuit
NI	Network Interface
FIFO	First In First Out
RR	Round robin
RRIE	Reconfigurable router to improve efficiency
HRRLPHP	Heterogeneous reconfigurable router for low power high performance.
EDRRHT	Eight directional reconfigurable router high throughput

SRRODFB	Smart reconfigurable router for online detection of faulty blocks
HSLPL	High speed low power link design for reconfigurable systems.
LMSTRDP	Link to minimize switching transitions for reducing dynamic power.
LMRTA	Link by multi encoding to reduce transition activities in reconfigurable system.

LIST OF PUBLICATIONS

JOURNALS:

1. Himani Mittal and Yogendera Kumar, **"Design of 8 direction Reconfigurable Router Supporting Heterogeneous Data for Network on Chip (NoC)",** Journal of Elsevier Materials Today, ICMAT Jan.2017.

2. Himani Mittal and Yogendera Kumar**, "Innovative Heterogeneous Design of Low Power Reconfigurable Router for Network on Chip (NoC) ",** IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE) e-ISSN: 2278-1676, p-ISSN: 2320-3331, vol. 11, pp. 01-09, Issue 5 Ver. II (Sep - Oct 2016).

3. Himani Mittal and Yogendera Kumar , "**High Performance Smart routing for Network On Chip ",** IJARCCE , International Journal of Advanced Research in Computer and Communication Engineering, ISO 3297:2007 Certified, vol. 5, Issue 9, Sep. 2016.

4. Himani Mittal and Yogendera Kumar**, "Comparative analysis of performance driven link designs for network on Chip",** Elsevier AKCE International Journal of Graphs and Combinatorics, CRNT, April 2017.

5. Neha Sharma, Himani Mittal, Reena Pathak and Yogendera Kumar, "**Design of power efficient router with low latency and high throughput for Network-on-Chip (NoC) ",** under communication.

CONFERENCES:

1. Himani Mittal and Yogendera Kumar**, "Simulation Analysis and Performance Evaluation of Four Innovative Routers for Network on Chip"**, WISPNET, March 2017.

2. Reena Pathak, Neha Sharma, Himani Mittal and Yogendera Kumar," **Design of high throughput and low power reconfigurable router with 4S buffers for NoCs**", under Communication.

CHAPTER 1

INTRODUCTION

1.1 Introduction:

Advancements in VLSI technology have driven the integration of more and more functionality into a single chip Integrated Circuits (ICs). It has allowed the manufacturing of complex electronic systems in a single chip with more than 10 billion transistors [1]. These single chip ICs are becoming increasingly complex. As a result, the development cost of these cutting-edge ICs has been continuously increasing and has become a threat to the continuation of the semiconductor roadmap [2]. It has been estimated that the cost of developing a chip at 28 nm technology node could reach over 170 million USD [3]. Furthermore, the significant investment and engineering effort do not reduce the risk of project failure. The development cycle of chips ranges from months to years with high uncertainty [2]. Therefore, the costs and risks associated with these customized IC solutions can only be justified for a limited number of ultra-high volume electronic systems. In the mean time, the microprocessor industry has moved from single-core to multi-core and later on to many-core architectures. These are having hundreds of similar types of cores and are known as Chip Multi-Processors (CMPs). Another important direction is toward System on Chip (SoCs), composed of many types of processing and other components on a single chip [4]. With the advent of SoCs, the design of electronic systems initiated an era of digital convergence, leading to integration of numerous applications onto a single chip. The mobile phone is a typical example of it. Furthermore, the increase in the embedded processing power led to the creation of new applications which must be executed concurrently. These include some modern portable equipment present in different markets like medicine, consumer electronics, precision agriculture, nano-satellites, Internet of Things, etc. [5]. Although these systems on chip have several advantages over the single chip ICs it has led to high power consumption, high design cost, large form factor, low mean-time between failure, low reliability, increased security risks, and a lot of side effects and conflicts.

Many of these issues can be resolved if reconfigurable logic is introduced into the system. Hence, the electronics industry has begun shifting towards using reconfigurable systems as an alternative platform. Compared with single chip ICs, the programmability of reconfigurable systems has increased the flexibility of designs while introducing acceptable overheads in power, area and performance. Hardware/software systems implemented on reconfigurable devices can achieve shorter time-to-market and are more amenable to upgrades or bug fixes over the product life-cycle [6]. It is predicted that by 2024, on average 70% of the chip functionality will be reconfigurable in various forms [2]. Reconfigurable systems present a good solution for applications consisting of a large number of processing units. In addition, through reconfigurability, the reusability of the resources is achieved. Field Programmable Gate Arrays (FPGAs) is the most common technology for implementing these Reconfigurable systems [7, 8].

As the number of components in Reconfigurable computing systems increases, the interconnect schemes based on Network-on-Chips (NoCs) approach are increasingly used. Earlier, different intra chip communication techniques were applied in their development, such as point to point and bus-based systems. NoCs provide scalable performance and parallelism in communication, meeting the requirements identified for future Reconfigurable electronic systems [9, 10, 11, 12, 13, 14, and 15]. A NoC can be described by its topology and by the method used for buffering, switching, routing, flow control and arbitration. It has several components which include routers (Rs), processing elements (PE) and network interfaces (NI). Router (R) is a vital component of the NoC. It is responsible to route incoming data packets to its destination. Router is used to forward packet from source to destination according to a prescribed routing algorithm. A processing element is connected to a router with the help of a network interface. The function of network interface is to packetize the data to make it travel in NoC through routers. The data is sent from a source to destination hop by hop with the help of routers. Every router, after receiving this data in form of packets or flits, first stores it in input buffers and then the control logic is making the routing decition and ultimately sending it to the next router or processing element. The channel or links are the physical media through which data is transferred from one router to the other router or processing element. In addition, the arbitration logic

is also used to streamline the data flow and selecting the priority among the various processing elements. Fig.1 shows the block diagram of a general NoC architecture [16].

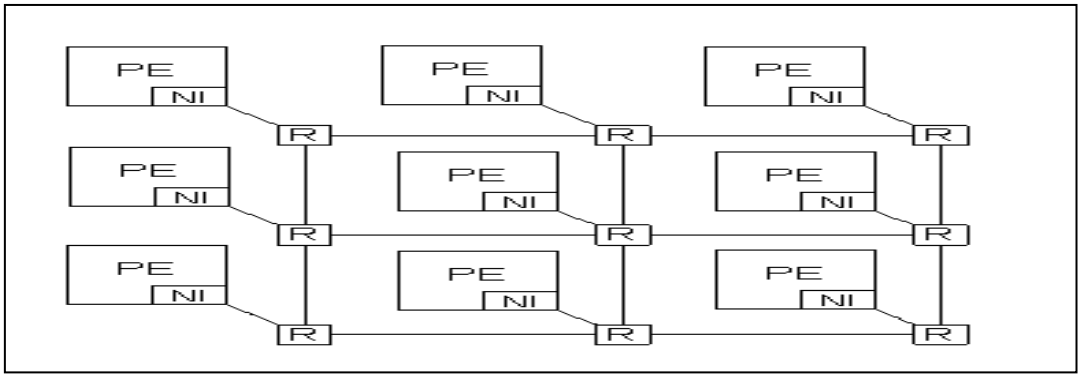


Fig.1 A general NoC Architecture

Since, in a NoC, a router forms the link that is used for maintaining the scalability and parallelism ordering in transfer and reception of data. Hence, routers play a vital role in functioning of a NoC system. In the earlier designed homogeneous routers, at static time resources are allocated which cannot be changed at the run time, this causes high data rejection and thus not able to handle the flowing data. Designing the same NoC router to cover the whole spectra of applications would mean an oversized and expensive router, in terms of area and power. At the same time, designing specific routers for different markets would mean that many important decisions would have to be taken at design time, hence precluding scalability and online optimizations targeting different behaviours with different application demands.

1.2 Statement of the Problem:

Now days, there is vast research going on in the field of reconfigurable routers that can be leveraged for NoCs. Design and implementation of these NoCs brings new challenges.

1. If a NoC's router has a larger FIFO buffer, the throughput will be larger and the latency in the network is smaller, since it will have fewer flits stagnant on the network. Nevertheless, there is a limit on the increase of the FIFO depth. Since each communication will have its peculiarities, sizing the FIFO for the worst case communication scenario will compromise not only the routing area, but power as

well. However, if the router has a small FIFO depth, the latency will be larger, and quality of service (QoS) can be compromised. The proposed solution is to have a heterogeneous router, in which each channel can have a different buffer size. In this situation, if a channel has a communication rate smaller than its neighbour, it may lend some of its buffer slots that are not being used. In a different communication pattern, the roles may be reversed or changed at run time, without a redesign step.

2. A NoC contains large number of IP cores on a single chip, the communication among these is a major issue. Communication among these IP cores should be hassle free. An organized structure is needed to perform this communication. The NoC usually have three basic components namely routers, links, and NIs (also known as wrappers). Routers and links are the most important components of a NoC design. Various types of routers are used in NoC. The performance improvement, low area, and low power are basic requirements of the router design. Similarly, various types of links designed for reduction of power while transmitting the data is the need of the hour along with the router designs.

1.3 Objectives:

Embedded Network on Chip (NoC) for reconfigurable systems is an active research field. Many aspects of NoCs still need further exploration and understanding [7, 17, and 18]. Efforts are made to apply the reconfiguration techniques to all level of computing systems, for intra-chip communication structures (buffers, topologies, memories, routers, etc.) as well as processor's micro-architecture (instruction set, data path, cache hierarchy etc.). In this thesis our main concentration is to design the efficient heterogeneous routers for NoC applications and do their performance analyses. The router is the main component that determines the latency, throughput, reliability and efficiency of the entire NoC design. Also, designing efficient links for reduced power dissipation, area and increased efficiency is important. Furthermore, comparison of various designed routers and links and their performance analyses compared to the previously designed routers and links is necessary.

Following are the objectives in broad spectrum of the present thesis:

1) The first objective of the present work is to select the topology from various topologies on the basis of performance metrics.
2) The second objective is to model and analyze a router architecture which is able to allow maximum utilization of buffers by allowing sharing of buffer units from its neighbouring buffers. The performance of the router in terms of high throughput, low latency, less area and low power consumption is to be analyzed. Various router architectures have been designed and compared for this purpose.
3) The third objective is to design and use of interconnection links to save power. Various types of links are designed and compared for this purpose.

1.4 Organization of the Thesis:

The thesis is organized in six chapters:

Chapter One gives an introduction and the evolution of the Network on Chips (NoCs). The motivation for the research and how the contributions will make a difference in the field of NoC chip at large are discussed in this chapter.

Chapter Two presents an overview of the necessary background on network -on-chip routers, interconnection or links etc. Networks coherency is provided as a foundation for understanding the rest of the dissertation. Furthermore, the detailed literature survey of network on chip routers and links is presented in this chapter.

The preliminaries of network on chips are discussed in **Chapter Three**.

Chapter Four provides the designing of the various reconfiguration system routers along with their performance comparison. Various routers designed are as:

a. RRIE (Reconfigurable router to improve efficiency).

b. HRRLPHP (Heterogeneous reconfigurable router for low power and high performance).

c. EDRRHT (Eight directional reconfigurable router for high throughput).

d. SRRODFB (Smart reconfigurable router for online detection of faulty blocks)

RRIE presents a reconfigurable router in which the buffer width can be reconfigured according to the need of the channel at run time. It improves efficiency and thus

increases the performance. HRRLPHP is a reconfigurable router for NoC applications. In this proposed router channel can take data anytime from its own channel as well as from the neighbouring channels. In EDRRHT four more directions (NE, NW, SE, and SW) are added. It now supports Eight Directions. It uses Buffer less storage concept to store the data of four newly added directions. Data of newly added directions share FIFO from their neighbour to store its data. Thus, the architecture requires less area. SRRODFB is the smart reliable router based on a new reliable NoC-based communication approach called RKT-NoC, along with its simulation results.

Chapter Five presents various links designed using MUX gating and various encoding/decoding techniques for reducing power along with their simulation results. Various links designed are:

a. HSLPL (High speed low power link for reconfigurable systems).

b. LMSTRDP (Link to minimize switching transitions for reducing dynamic power for reconfigurable system).

c. LMRTP (Link by multi encoding to reduce transition activities in reconfigurable system).

Finally the conclusions are drawn from the results of the work presented in **Chapters Three** through **Five.** These conclusions are discussed in **Chapter Six.** This Chapter also suggests the scope for future work in the area covered by this thesis.

CHAPTER 2

LITERATURE SURVEY

An extensive survey of the research conducted by various researchers in the field of NoC design, Routers, links, routing algorithm and embedded NoCs for FPGAs has been carried out in the present Chapter.

2.1 NoC Design:

Scientists came up with the idea of System on chip (SoC) with buses as the interconnections [19]. This improved the performance of the embedded systems but later, the bus between multi components of embedded systems was not able to cope the heterogeneous and challenging communication requirements of SoCs. This led to the idea of Network on Chip (NoC). This survey paper [19] is also highlighting and gathering the key aspects of NoC i.e. routers and links from various sources which act as catalyst for further research in this area.

Network on chip (NoC) is a modern day communication structure where various IP'S are connected with each other in some topology. These topologies can be 2D mesh, butterfly, torus, tree, or a mixture of these called a hybrid topology [6]. NoC is made up of routers, processing elements (PE's), network interface (NI) and interconnects (links). Routers are the most important elements of NoC and are connected with each other by links. PE's are connected to routers through NI's. NI separate the data communication of PE's from inter router communication. NI transforms the message generated by PE's to the packet format which router understands. Routers send these packets to the neighbour routers so that it can reach at the destination. PE's can be homogeneous or heterogeneous devices. The PE's can be a cache, reconfigurable devices, memory, or any other processing element [10]. As NoC resources are well structured and arranged in a particular topology [13], it makes the NoC scalable and modular.

In 2001 W.J Dally proposed [16] the basic principle of an interconnection network. This put the research foundation for on chip communication. It describes about the communication between processing elements, and present SoCs designs, provide integrated solutions to challenging design problems in the communications between different IPs.

L. Benini & G.D Micheli [10] named the on chip interconnection network as Network on Chip (NoC). In this paper, various NoC architectures have been reviewed based on the different parameters [3-4]. The NoC [20] is based on new error detection mechanisms for smart NoCs, here during runtime position and number of faulty blocks and PE's can vary. A new online detection method for data packet and algorithms for error correction is presented in [93]. It is able to differentiate between permanent and transient errors and is able to accurately spot the place of faulty blocks [95], while preserving the network load, the throughput and the data packet latency.

The purpose of this survey is to review the existing NoC architectures from different perspectives (parameters). Most of the papers in the literature review bit about the architectures in their papers and the explanation are specific to their application or domain of the paper. In order to fill the gap and help the research community this survey study is conducted in which more than 100 different NoC architectures are reviewed. To the best of our knowledge, this is the most comprehensive literature review of NoC architectures. There are few other survey studies on NoC architectures [39-40] but they cover the basic trends, principles and working mechanism of network on chip. These survey studies which are published in 2008 are not covering the broader spectrum of NoC architectures using various parameters and only cover eight NoC architectures based on few parameters. The survey study [42] is very well written but explains the general trends in NoC research and practices. The comparison is more qualitative then quantitative. The survey study was published in 2012 and it covers seventy seven different NoC architectures [54]. These architectures are compared based on only four parameters. The parameters include year, switching, topologies and implementation. This paper explains the basic principles of NoC which include routing algorithm, switching, flow control techniques and other mechanisms. The NoC has brought the tremendous change in the on-chip communication mechanism. It has replaced the traditional wires with the routers and interconnects. The bus between multiple components was not able to cope with communication requirements of them. To overcome this problem O. Sharma [106] describes a quadro coding technique for design space exploration using a two-steps scheme to optimize the number of virtual channel buffers used to implement logical channels multiplexed

across the physical channel in a router output port for QOS supported on-chip communication.

D. Matos [67] presented a Network-on-Chip (NoC) suitable for Dynamically Reconfigurable systems. The router in this NoC detects online errors of routing algorithm and is also able to find errors in the data packet. Algorithms here are adaptive in nature which permits bypassing of faulty components and dynamically implemented in the network. The architecture is based on additional state indications as well as particular logic blocks. The basic idea is that faulty blocks which are permanent are disconnected from the routers. The originality in the proposed NoC is that only the permanently faulty parts of the routers are disconnected. This guarantees a high throughput during run time in the NoC without any data packet loss. It is also having a self-loopback mechanism inside each router. High throughput along with low power, low area maintaining high performance is the calling demand.

D. D. Tomaso [50] discussed that power, area, and performance of the NoC architecture are tightly integrated with the design and optimization of the link, router (buffer and crossbar), and topology. Recent work has shown that adaptive channel buffers (on-link storage) can considerably reduce power consumption and area overhead by reducing or replacing the power-hungry router buffers [6]. However, channel buffer design can lead to head-of-line (HoL) blocking, which eventually reduces the throughput of the network. In this paper, it discusses about design channel buffers and router crossbars to improve the performance (latency, throughput) while reducing the power consumption.

R. Marculescu discusses about the key research problems in NoC design. Through dynamic and creative research into questions ranging from integrating reconfigurable computing techniques, to task assigning, scheduling and arrival, to designing an operating system to take advantage of the computing and communication flexibilities brought about by run-time reconfiguration and network-on-chip, this represents a complete source of the techniques and applications for reconfigurable network-on-chip necessary for understanding of future of this field [18]. There's a crisis of global interconnection with existing bus architectures in such SoC Designs. In response to this crisis, Network-on-Chip (NoC) is an upcoming paradigm, and is becoming the

leading contender to replace the conventional bus architectures. Many Network-on-Chip topologies [50] have been proposed in an attempt to tackle various chip architecture needs and routing techniques. In this paper, some of the topologies such as Mesh, Torus, Binary Tree and Butterfly Fat Tree (BFT) have been simulated using a Network Simulator (NS2) and their performances have been assessed and compared taking throughput, maximum end-to-end latency and dropping probability as assessment parameters. Till now we focus on high performance features but how to control errors should also be taken into consideration.

T. Boraten, in 2016 proposed the impact of various runtime techniques to control soft errors against noise, high performance, and low energy consumption which are key objectives in the design of on-chip networks and on the trade off between them [28]. In all these works performance and reliability are measured separately. It shows the use of error-control schemes in on-chip networks results in degradable systems, hence, performance and reliability must be measured jointly using a unified measure, i.e., perform ability. A detailed comparative analysis of the error-control schemes using the perform ability analytical models and SPICE simulations is provided taking into consideration voltage swing variations (used to reduce interconnect energy consumption) and variations in wire length. Furthermore, the impact of noise power and time constraint on the effectiveness of error-control schemes are analyzed.

2.2 Routers:

Analytical Design and implementation of on-chip router architecture for NoCs, which is the concept of high efficiency architectural design is present in Design and Performance Evaluation of a NoCs- Based Router Architecture for SoCs [10]. After online detection of chip there arises the need for varying buffer depth so emerge the concept of reconfigurable routers. In this section of literature survey we did a study where several of these ideas demonstrate that in the same application there is the need to use routers with different features and communication needs. A. Bouhraoua [87] observed the need to have heterogeneous links in a network-on-chip, where for each application a link with appropriate size is used. A heterogeneous router was presented based on heterogeneous links with two wrappers for each router. These wrappers control the traffic between the heterogeneous channels and each channel works with the message compatible with its width. Following this path, Kreutz et al. [43]

analyzed the architecture of three routers, each one having different characteristics. Ahmad et al. [58] proposed a network router designed with a bus like interface in 2008. An in-built wrapper is used, and thus any component compatible with a bus can be integrated into the NoC architecture. The interface of this router becomes a simple bus. The objective of this proposal was to reduce the design time and to ease integration, since it is not necessary to known to the NoC architecture. When the network has a channel that requires high bandwidth, the NoC changes the switching, obtaining a dedicated path between the IPs.

C.K Hsu presents low power detection method that supports buffer less concept. However, buffer less NoC [79] has some limitations that include lower network throughput caused by deflection routing at high load. It discusses the drawbacks of buffers and hence introduce buffer less Network-on-Chip. The major component of a Network-on-Chip architecture is the router, which affects the data transmission latency, chip area and power consumption. Inside the router, buffers occupy a significant amount of power and a large partition of chip area. Therefore buffer less NoC, which discards the buffers in the routers, has been proposed for solving the power and area problem. In this paper, a low power deflection routing method is proposed for the buffer less on-chip network dealing with the routing problem and achieving the low power goal. It uses routing matrix for constructing the possible routing path, and then selects the best route for each data packet. Only few calculations are used in this method therefore lowering power consumption the low power goal. In addition, the longer router critical path impacts the router frequency, which reduces the amount of bandwidth provided by the network router. These limitations reduce any benefit of buffer less NoC - especially for high-throughput workloads such as GPGPU. It provides a simple analysis into how the benefit of buffer less NoC can be reduced for high throughput workloads, especially in terms of energy-efficiency. Then propose clumsy flow control (CFC) - a congestion control mechanism that can reduce the amount of deflection and improve the efficiency of buffer less NoC. The clumsy flow control enables the allocation to be simplified and propose a novel switch allocation (randomized-deterministic allocation) which significantly reduces the router critical path. The combination of these two techniques result in our buffer less NoC to exceed the system performance of buffered network

by approximately 7% (up to 22%) while reducing network area by 53% and energy by 52%.

A novel system-level buffer planning algorithm that can be used to customize the router design in Networks-on-chip (NoCs) is presented in System-Level Buffer Allocation for Application-Specific Networks-on-Chip Router Design. M.A. Faruque discusses a novel methodology [54] for design space exploration using a two-steps scheme to optimize the number of virtual channel buffers used to implement logical channels multiplexed across the physical channel in a router output port for QoS supported on-chip communication.

P.P. Pande discusses about broad perspective on the design, synthesis and Integration issues associated with NoC design [82]. It also discusses about types of IPs, compact switch architecture and its fast path setup scheme for circuit-switched on chip network.

2.3 Links:

The links have become main element in dynamic power dissipation in a Network on Chip (NoC) design. D.N Sama in paper [40] entitled "A Novel Encoding Scheme for Low Power in Network on Chip Links" proposed schemes on the power reduction in between the links. Dynamic power dissipation in interconnects is a major contributor to power consumption in Network on Chips (NOCs). This is mainly due to two factors, self switching activity of the particular link and coupling switching activity among adjacent links. Two novel techniques are proposed to reduce power consumption due to switching transition and crosstalk. First technique reorders the data in such a way that switching transition is brought down. In the second technique, it is ensured that power consumption due to cross coupling activity is reduced. Encoder and Decoder exhibiting the proposed scheme have been described in RTL level in Verilog HDL, synthesized and mapped into UMC180 nm technology library. It has been observed that the proposed technique (TSC) offers an average reduction in dynamic power consumption of 17.34%. Proposed scheme was compared with existing techniques and observations concluded that there was not much degradation in area, speed and static power dissipation. Power reduction when subjected to different kinds of data streams was analyzed and results indicate that proposed

scheme offers uniform power reduction irrespective of the nature of data stream unlike the existing techniques.

C.S. Behere [101] in 2014 proposed the data encoding scheme for the power reduction in network on chip links. This data encoding scheme exploits the wormhole switching techniques and works on an end-to-end basis. It means, flits are encoded by the network interface (NI) before they are injected in the network and are decoded by the destination NI. This makes the scheme transparent to the underlying network since the encoder and decoder logic is integrated in the NI and no modification of the router architecture is required. Data encoding schemes are other techniques to reduce switching power in NoC links by reducing self transition and coupling transition in links. This paper proposed data encoding scheme called self and coupling driven bus invert (SCDBI). By using this method we can reduce the switching power by 34.64% without any significant degradation in area and performance. The SCDBI Encoding Scheme gives better power reduction in between the links. The power dissipated by the links of a NoC contribute significant fraction of the total power budget. This paper proposed the use of data encoding techniques as a viable way to reduce power dissipation in NoC links. The SCDBI scheme is transparent because they operate on an end-to-end basis and no need to modify the router architecture, only the NI is to be augmented with the encoder and decoder. Although, it represents an overhead, does not introduce a significant penalty both in terms of cost (i.e., silicon area) and latency. The simulation results and synthesis results have shown that, by using the SCDBI encoding scheme power dissipation of the coupling switching activity and self switching activities are reduced in links that connects routers inside the NoC. Precisely, as compared to a baseline implementation in which no data encoding techniques are used, a reduction of up to 34.64% of power dissipation has been observed without any compromise in area as well as performance.

M. Palesi et.al in paper [110] entitled “Data Encoding Schemes in Networks on Chip” proposed the use of data encoding techniques as a viable way to reduce both power dissipation and energy consumption of NoC links.

The proposed encoding schemes have been compared with several encoding schemes proposed in literature on data set from various sources. N. Jafarzadeh et.al in paper [108] entitled “Data Encoding Techniques for Reducing Energy Consumption in Network-on-Chip” proposed efficient power reduction than the previous paper. This

paper presented various encoding methods to reduce dissipation of power in various links of NoC. Simulation results carried out on various set of data validitate the proposed methods effectively, which shows saving up to 54% of power dissipation and 17% of energy consumption without degradation in area and performance. reducing the test time with small area overhead, K. P. Sridhar demonstrated a method using Hamming Encoder and Decoder [99] aimed to reduce the test time with small area on NoC. It reduces the switching activity between the modified test inputs which lead us to reduce the test power dissipation during testing. The area is estimated by adding hardware component in the Field Programmable Gated Array (FPGA) SPARTAN III kit using XILINX ISE Design Suite 14.7. Hardware Description Language (HDL) Verilog is used to describe the additional hardware component. The simulation result verifies that 6% area is required additionally. So far various single technique targeted specific parameter has been surveyed. Now there comes hybrid of techniques taking advantage of each technique is also surveyed.

S. Chetan compared the power ratio between the network on chip links [101] and find router links are more power hungry than router. The dynamic power dissipation in links is major contributor to the power consumption in network on chip. This is due to the self switching and the second factor is cross coupling capacitance. In the proposed encoding technique the first self switching is reduce by checking the switching transition and then the coupling between the links is checked and ensured that the power consumption is reduced. The encoder and decoder of the proposed scheme are described in the RTL level in Verilog HDL.

2.4 Routing Algorithms:

For any N.O.C architecture algorithm plays a vital role for communicating packets. S.D Chawade describes a modified XY routing algorithm [58] combined with a scheduler to be used on NoCs. This method is a fast way to transferring data via a specific path between two nodes in the network and the scheduler further helps to avoid collision.

XY and OE algorithms based on the 4x4 mesh topology by using NIRGAM emulator presented in Comparison of 2D MESH Routing Algorithm in NoCs [59] .It describes about most suitable algorithm for 2D mesh topology.

The NoC online fault detection problem has been widely explored in literature for non reconfigurable architectures. Basically the associated methodologies and algorithms

have focused on minimizing different metrics, using different algorithmic solutions. J. Borecky [94] discusses about various fault models for on line testing on a reliable reconfigurable real-time operating system (R3TOS) that was developed for reconfigurable systems without NoC. The R3TOS executes scheduling and placement of tasks using several different metrics and methods. The placement algorithms focus on preserving the maximum empty rectangle (MER) for the future use for as long as possible. The algorithm analyses the timeline, for preventing future fragmentations of modules, keeping the modules packed, and, consequently, achieving higher computation densities. [29] Compares various algorithms for minimizing energy consumption: exhaustive search, simulated annealing, greedy incremental, and largest communication, among others. Several works also proposed a multi objective approach like in, where an immune-adaptive algorithm was presented for reducing both power consumption and latency. Along with online detection problems associated with N.O.C mapping has also been reviewed. N.Karimi [95] presents on-line fault detection and location for NoC interconnects where a placement and mapping methodology is studied for simple PRS-NoCs, with a direct and regular NoC topology. The space of the FPGA is divided in big reconfigurable slots connected to routers, in which modules are placed. The placement and mapping are proposed for both application design and operating time and for multi applications. In the pre processing step, the solution is divided in basic mapping and specific configurations. The basic mapping is composed of the modules which are used by the most of the applications, and the specific configurations are a set of configurations that cannot be reused from the based mapping. Firstly, a genetic algorithm is used for placing modules of the base mapping. A second mapping is executed for placing the modules that belong to specific configurations. Other algorithms are proposed for real-time mapping: a greedy algorithm for configurations reuse or a SAT solver. For reducing latency a new pipelined router design focused on reducing the router latency and identifies the router components that take most of the critical path, and thus limit the router frequency.

XY routing algorithm[58], which is a kind of distributed deterministic routing algorithms present in Comparison Research between XY and Odd-Even Routing Algorithm of a 2-Dimension 3X3 Mesh Topology Network-on-Chip it demonstrate the two routing algorithms XY routing algorithm and OE routing algorithm for 3X3

mesh topology. Literature survey for high performance based on different topologies has also been done. A.Barzinmehr [63] discusses about recent multi core processors have leveraged a ring topology and hierarchical ring can increase scalability but presents different challenges, including higher hop count and global ring bottleneck. In particular, we propose a novel hybrid flow control for hierarchical ring topology to scale the topology efficiently. The flow control is hybrid in that the channels are allocated on flit granularity while the buffers are allocated on packet granularity. The hybrid flow control enables a simplified router micro architecture (to minimize per-hop latency) as router input buffers are minimized and buffers are pushed to the edges, either at the output ports or at the hub routers that interconnect the local rings to the global ring-while still supporting virtual channels to avoid protocol deadlock. We describe a packet-quota-system (PQS) and a separate credit network that provide congestion management, support prioritized arbitration in the network, and provide support for multi flit packets. We also provide alternative designs for the credit network and PQS architectures. A detailed evaluation of a 64-core CMP shows that the tNoC improves performance by up to 21 percent compared with a baseline, buffered hierarchical ring topology while reducing NoC energy by 51 percent. If in the above discussed router buffer comes in sharing mode than it can lead to high performance.

A.Tran in 2014 presents a new shared buffer based router architecture employing dual buffers [79]. It is shown that the proposed architecture has an advantage in its increased buffer utilization leading to an improved throughput and reduced latency. The experiments show that the proposed architecture improves the NoC's throughput and latency compared with a conventional architecture with little overhead in the control logic. Using buffers in various ways for saving power is already discussed above.

2.5 EMBEDDED NoCs on FPGA:

This section reviews embedded NoCs on FPGA research conducted by various academic and industrial groups. According to L.Benini and D.Micheli, NoCs are forecast to become the main interconnection substrate for large systems on chip (SoCs), which is described in “the only path to mastering the complexity of SoC designs in the years to come. “ As FPGAs become ever larger reconfigurable computing platforms, researchers started believing the huge benefits from NoCs in

integrating the FPGAs with an embedded NoC and so they focused on how to use the NoC for transporting data in FPGA applications of different design styles. The Fabric Port is a flexible interface between the embedded NoC and the FPGA's core; it can bridge any fabric frequency and data width up to 600 bits to the faster but narrower NoC at 1.2 GHz and 150 bits.

Field-programmable gate-arrays (FPGAs) have evolved to include embedded memory, high-speed I/O interfaces and processors, making them both more efficient and easier-to-use for compute acceleration and networking applications. However, implementing on-chip communication is still a designer's burden wherein custom system-level buses are implemented using the fine-grained FPGA logic and interconnect fabric. The design of highly scalable NoCs for reconfigurable systems was reported by Bartic et al [111]. Their decision allowed to instantiate arbitrary network topology and having low latency and high throughput. To reduce the effort to design the communication channels and to improve their hardware efficiency was done by [112]. Altera's Qsys[113] and Xilinx's Vivado [114] IP integrator automated the task of converting a desired communication style and specification into an implementation using the FPGAs soft logic and interconnect fabric to make the appropriate point to point links and arbitrated logical buses.

CoRAM[115], LiPar [116], NoCem [117] and Conmet [118] describe the work in which the use of packet switched NoC in FPGA design were investigated. R.Giddin et al [119] presented a novel NoC based architecture for FPGAs. The architecture and implementation of a configurable router for embedded NoC support within FPGAs was proposed by Pan and Manjijian [110].

M. S. Abdelfattah [121,122,123,124] explored the addition of a fast embedded network-on-chip (NoC) to augment the FPGA's with an embedded NoC and focused on how to use the NoC for transporting data in FPGA applications of different design styles. A flexible interface between the FPGA fabric and the embedded NoC allows modules of varying widths and frequencies to transport data over the NoC. To implement embedded NoC on system level communication in 2017 M. S. Abdelfattah et al describes FPGAs with an embedded network-on-chip (NoC). They have designed custom interfaces to connect a packet-switched NoC to the FPGA fabric and I/Os in a configurable and efficient way and then define the necessary conditions to

implement common FPGA design styles with an embedded NoC. Four application case studies highlight the advantages of using an embedded NoC.

2.6 CONCLUSION:

The developments in the area of NoC design and reconfigurable systems are reviewed in this chapter. This literature survey shows that there has been a great amount of work carried out in this area. Based on these works, the NoC design has advanced to the extent of being useful in the reconfigurable systems. Various components constituting NoCs, like routers, links and routing algorithm etc. have also been reviewed in this chapter. Based on these studies, it is concluded that this area needs more designs for further development of embedded NoCs for reconfigurable systems.

CHAPTER -3

PRELIMINARY CONCEPTS OF RECONFIGURABLE SYSTEMS, SoCs & NoCs

In this chapter preliminary concepts of reconfigurable system and network on chip are discussed briefly:

3.1 RECONFIGURABLE SYSTEM:

A reconfigurable system is a architecture that has good features of both hardware and software by combining flexibility of software and the high performance of hardware. It functions by processing high speed computing fabrics which is very flexible.

The best known example of reconfigurability in electronics is the field programmable gate arrays (FPGAs), made popular originally in its role as a replacement for application-specific integrated circuits (ASICs).

A simplified reconfigurable system [10], has six basic features as shown in Fig. 3.1.

i) soft-defined switches.

ii) composable building block (circuit) elements (fixed and configurable).

iii) input/output structures.

iv) configurable wiring to connect these elements together.

v) configuration management engine to control the states of switches and configurable settings and

vi) programming model defined for users that provides a conceptual picture of the reconfigurable system to aid in its effective use to implement end designs.

Detailed descriptions of these features are:

i) Switches are the foundation of reconfigurable systems, since they allow for the insertion of a configurable state. A switch can be thought of as software wire, where

the existence or absence of connection is defined by changing state. They are usually implemented in real-world systems with MOSFET devices.

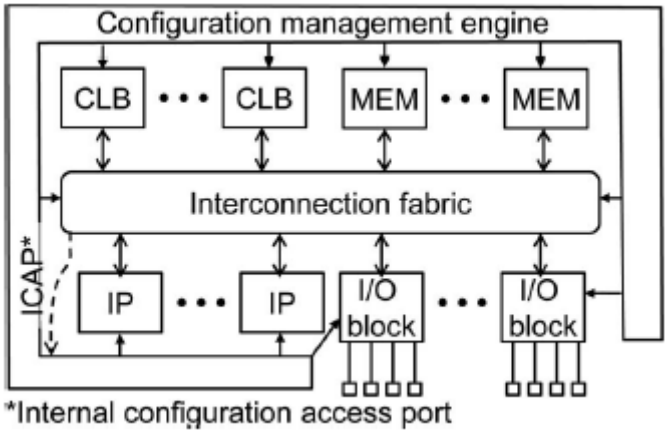


Fig. 3.1 A typical reconfigurable system [8]

ii) Building Block Elements In reconfigurable systems, as in the cells of the standard library in VLSI design, modular composition of primitive elements and hierarchy become key design principles. For FPGAs, the most important building block primitive is the lookup table (LUT), which is simply a programmable truth table capable of implementing different logic functions (AND, OR, etc.). LUTs are themselves embedded within more complex building blocks referred to as configurable logic blocks (CLBs). They are included in Fig. 3.1, as they represent the key workhorse in modern FPGAs for implementing complex digital functions. In principle, almost all digital functions could be implemented only using CLBs. As FPGAs evolved, other types of building blocks emerged, referred to as hard-core intellectual property (IP) blocks (which may either be fixed or also have embedded ‘‘knobs’’ and configurable features). These IP blocks contain dedicated circuitry corresponding to frequently recurring design patterns. While at least some of these IP blocks could be implemented as ‘‘soft cores’’ (literally shaped from within the FPGA fabric itself using CLBs), using combinations of hard-core IP and fabric-based (soft) IP has resulted in the most efficient implementations.

iii) Input / Output Blocks Input/output blocks (IOBs) are a specialized form building block element intended to provide external communication. In modern FPGAs, IOBs are almost always configurable, usually capable of supporting more than a dozen

distinct physical layer interface concepts, ranging from general-purpose I/O to high-speed serial links and often dedicated hard-core IP blocks supporting popular, and interfaces (e.g., Ethernet).

iv) Configurable Wiring defines detailed terminal connections between building blocks within a system providing the essential glue that expresses a system design, and it is important for these connections to be made flexibly. A considerable fraction of the real estate of modern FPGAs is dedicated to wiring, and a number of methods have evolved in an attempt to combat wiring manifold growth.

v) Configuration Management Engine As shown in Fig. 3.1, reconfigurable systems require some configuration management engine to manage the state of (and transfer user designs into) reconfigurable system fabrics.

vi) A programming model, representation of all user-accessible switches within a reconfigurable system (including the switch states of every CLB, IP block, and configurable wire) is often referred to as the configuration bit stream. Since the bit stream is in effect an informational string uniquely defining the personality of the reconfigurable system or ‘digital DNA’’.

3.2 SYSTEM ON CHIP (SoCs) & NETWORK ON CHIP (NoCs):

A SoC stands for System on Chip [19]. It is a microchip on which the entire system resides, i.e., all components such as transistors; peripherals etc are integrated on a single chip. Before the evolution of SoCs different components were designed and integrated or connected with external wire (buses) as shown in Fig. 3.2. Also, the software was not integrated in these systems. As time spent, feature size decreased which gave rise to the more packing density. In addition, requirement of low power devices or simply need of battery operated devices increased. Time to market is also expected to be low. All these factors gave rise to the evolution of SoCs, where multiple IP cores are integrated on a single chip.

Advantage of SoCs [19]:

Less consumption of power is seen in SoC. More than 85% power is consumed in data and bus address cabling. Since all the components are on the same chip and internally connected, and their size is also very small, the power consumption is significantly decreased. A SoC provides better design security at firmware and

hardware levels. A SoC features faster execution rate due to its high speed processor and memory.

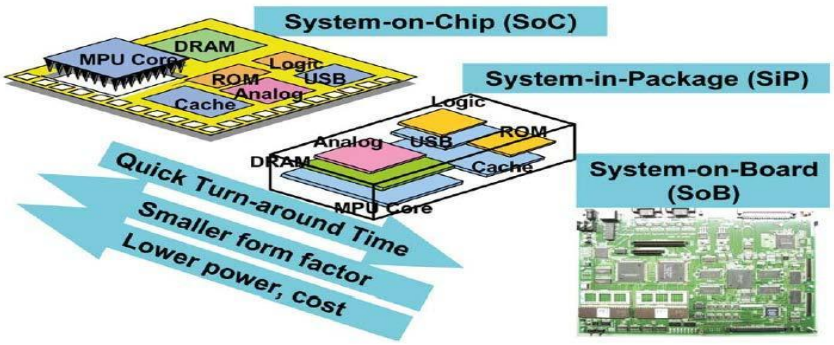


Fig. 3.2 SoCs versus SIP versus SOB [19]

Disadvantage of SoCs:

The initial cost of designing SoC is high, even a single transistor or system damage is very costly so the complete board cost and its servicing is very expensive. Due to cost consideration it is not recommended for power-intensive applications.

Network on chip or network on a chip (NoC) is a communication system on chip (SoC) between various intellectual property (IP) cores.

NoCs route data from the source to the destination components, by a network fabric that consists of switches (routers) and interconnection links (wires). An example of a NoC system is illustrated in Fig. 3.3 It shows a NoC architecture showing interconnections with mesh topology, that consist of processing elements (PEs) joined by routers and links.

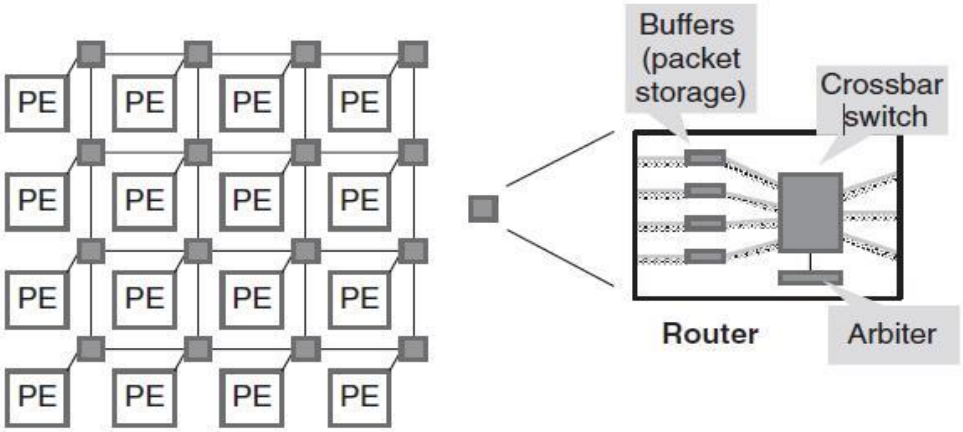


Fig. 3.3.a A NoC system [27]

Network interface is used to packetize data generated by processing element which is further connected to a router, which has buffers at its input to accept data packets

from processing element or neighbouring routers connected. Inside the router is placed a crossbar switch which is used to route the data packets from the input buffers to output, based on the address in the packet header.

To determine which packet should be given priority when multiple packets are received to travel on same link a component named arbiter is used.

Networks-on-chip (NoC) is an evolutionary process as shown in Fig.3.4, the time frame for which will be determined by several factors such as the rate of increasing design complexity and how rapidly process technology will continue to advance, in the coming years.

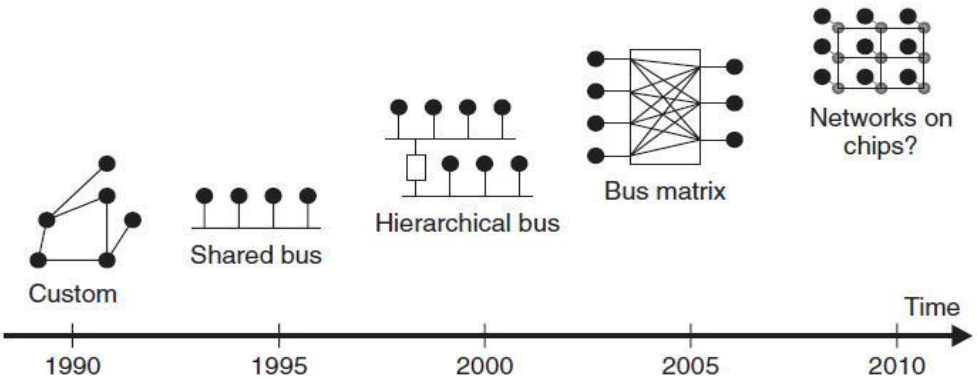


Fig. 3.3.b Evolution of on-chip communication architectures [27]

The communication design space for NoC type architectures is much larger and more complex than for bus-based communication architectures, the design problems. Remain the same, requiring a trade-off between performance, power, cost, area, and reliability to create a communication fabric that can meet the requirements of a given application.

3.2.1 NETWORK TOPOLOGIES:

The topology of a NoC specifies the physical organization of the interconnection network. It defines how nodes, switches, and links are connected to each other. Topologies for NoC can be classified into three broad categories — direct networks, indirect networks, and irregular networks. These are described below.

In direct network topologies as shown in Fig. 3.4, each node is directly connected to neighbouring nodes. The nodes consist of memories and computational blocks along with a network interface block or router. Links are used to connect the routers of the neighbouring nodes .

A fully connected direct network topology, as shown in Fig. 3.4 (a), where every node

is directly connected to all the other nodes is quite prohibitive. More the connectivity more is the probability of better performance, but at the cost of more energy and area. Examples of popular orthogonal direct networks include the *n*-dimensional mesh Fig. 3.4(b), ring Fig. 3.4(c), torus Fig. 3.4(d), folded torus Fig. 3.4(e), and octagon Fig. 3.4(f) topologies.

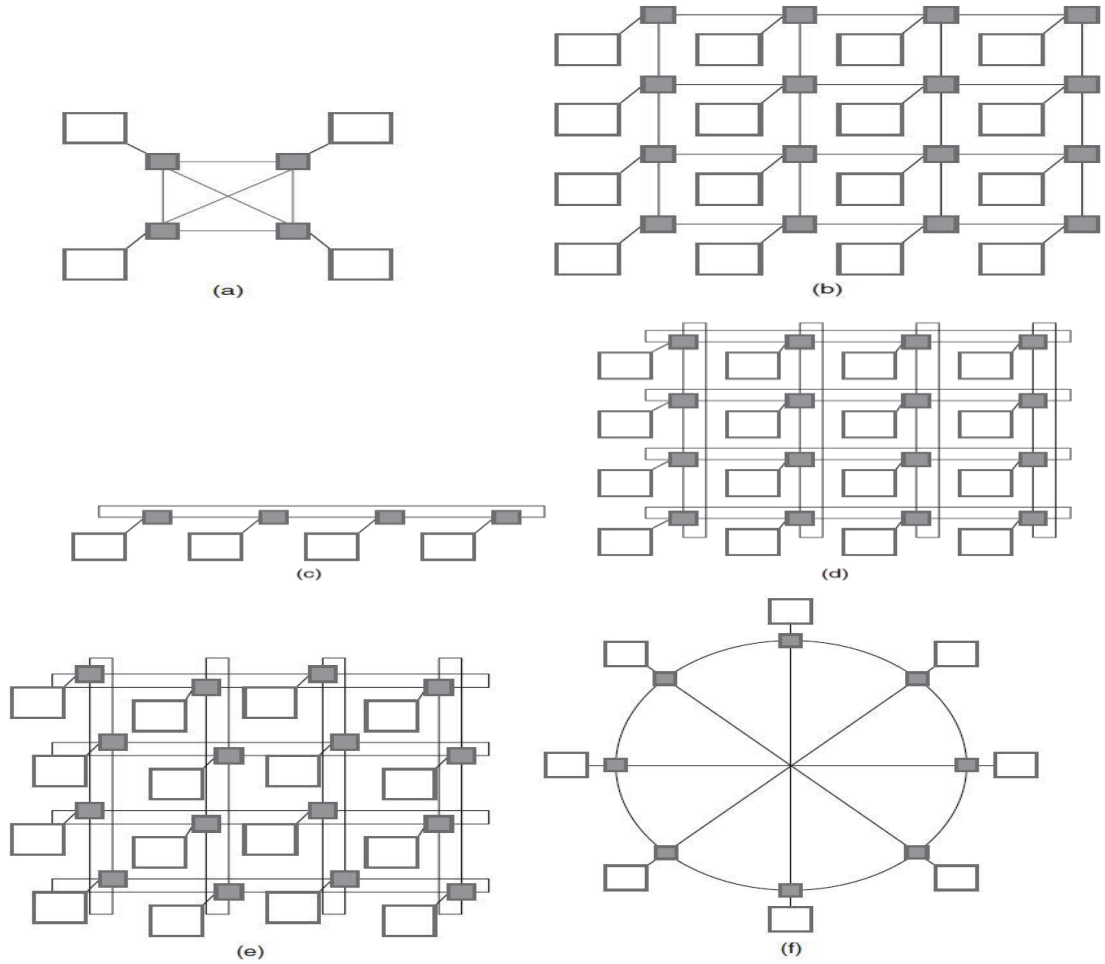


Fig. 3.4 Direct network topologies (a) Point to point (b) Mesh (c) Ring (d) Torus (e) Folded Torus. (f) Octagon [27]

In indirect network topologies, each node is connected to an external switch, and switches have point to-point links to other switches. The NI associated with each node connects to a port of a switch. Switches do not perform any information processing, and correspondingly nodes do not perform any packet switching. Indirect or switch-based networks are another major class of interconnection networks. Instead of providing a direct connection among some nodes, the communication between any two nodes has to be carried through some switches. Each node has a network adapter that connects to a network switch. Each switch can have a set of ports. Each port

consists of one input and one output link.

Fig. 3.5 shows examples of popular multi-stage indirect networks. [27] Fig.3.5 (a) shows an example of the fat-tree topology. In this tree topology nodes are connected only to the leaves of the tree. For a simple-tree topology, it was observed that the root and its neighbours have higher traffic. This problem can be alleviated by using fat trees, where links among adjacent switches are increased as they get closer to the root of the tree. Figure 3.5 (b) shows a 2-ary, 3-fl y butterfly topology.

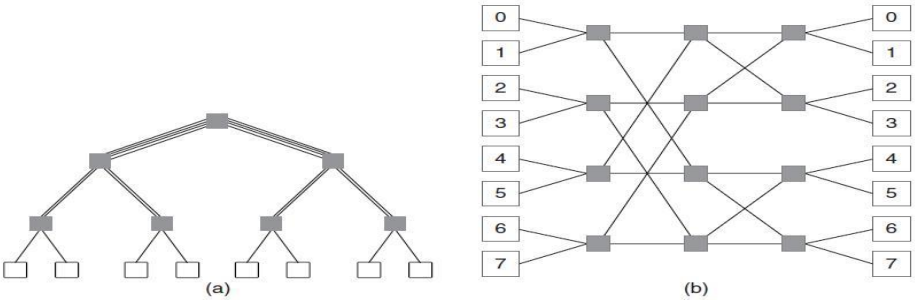


Fig.3.5.a Indirect network topologies (a) fat tree (b) butterfly [27]

Irregular or ad hoc network topologies are usually a mix of shared bus, direct, and indirect network topologies. The goal of these topologies is to increase available bandwidth as compared to traditional shared buses, and reduce the distance between nodes as compared to direct and indirect networks. Irregular topologies are typically customized for an application.

An example of an irregular network topology is a reduced mesh in which unnecessary routers and links have been removed as shown in Fig. 3.5(a). A cluster-based hybrid network— where each cluster has any combination of a shared bus-based, direct, or indirect network topology—is another example of an irregular topology. Fig.3.5 (b) shows an example of a cluster-based hybrid topology which combines a mesh and a ring topology.

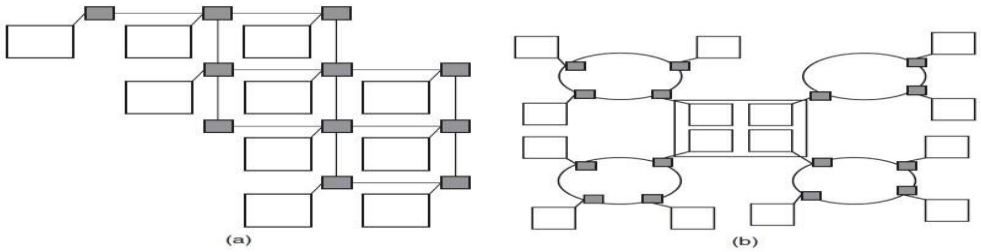


Fig. 3.5.b Irregular network topologies (a) optimized (reduced) mesh, (b) cluster-based hybrid (mesh + ring) topology. [27]

3.2.2 Switching Strategies:

The NoC switching strategy determines how data flows through the routers in the network. The architecture of a generic router is shown in Fig.3.6 and is comprised of the following major components.

i) Buffer: These are first-in first-out (FIFO) buffers for storing messages in transit. In the model shown in Fig. 3.6, a buffer is associated with each input physical channel and each output physical channel. High-speed routers will utilize crossbar networks with full connectivity, while lower-speed implementations may utilize networks that do not provide full connectivity between input buffers and output buffers.

ii) Routing and arbitration unit: This component implements the routing algorithms, selects the output link for an incoming message, and accordingly sets the switch. It will be routed again after the link is freed and if it successfully arbitrates for the link.

iii) Link controllers (LCs): The flow of messages across the physical channel between adjacent routers is implemented by the link controller. The link controllers on either side of a channel coordinate to transfer units of flow control.

iv)Processor interface: This component simply implements a physical channel interface to the processor rather than to an adjacent router.

It consists of one or more injection channels from the processor and one or more ejection channels to the processor. Ejection channels are also referred to as delivery channels or consumption PEs (nodes) generates messages that are partitioned into possibly several data packets. A packet is further divided into multiple **flits** (flow control unit).

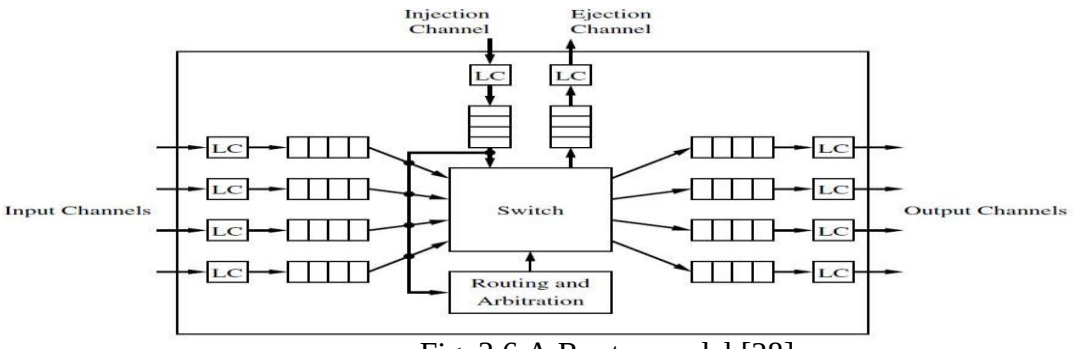


Fig. 3.6 A Router model [28]

A flit is an elementary packet on which link flow control operations are performed, and is essentially a synchronization unit between routers [28]. Each flit is made up of one or more phits (physical units). A phit is a unit of data that is transferred on a link

in a single cycle. The size of a phit is typically the width, in bits, of the communication link. Fig. 3.7 shows the structure of phits, flits, packets, and messages. Different NoCs architectures use different phit, flit, packet, and message sizes.

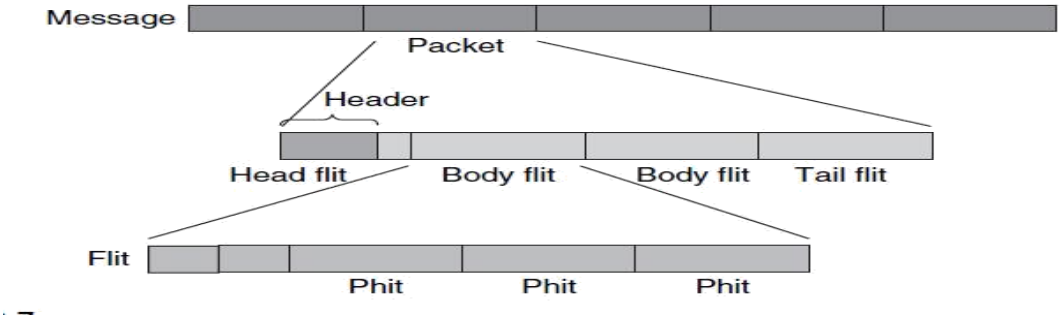


Fig.3.7 Structure of messages, packets, flits, and phits [27]

The choice of size can have a significant impact on cost, performance, and power for NoCs fabrics. The two main modes of transporting flits in a NoCs are circuit switching and packet switching. These are described below.

In circuit switching, as shown in Fig. 3.8, a physical path between the source and the destination is reserved prior to the transmission of data. The physical path is made up of a series of links and routers, and the messages from the sender are sent in their entirety to the receiver once a path (circuit) is reserved. A message header flit traverses the network from the source to the destination, reserving links along the way. If the header fl it reaches the destination without any conflicts, all links in the path are available and an acknowledgement is sent back to the sender. Data transfer then commences for the sender upon receipt of the acknowledgement. If a link has been reserved by another circuit, however, a negative acknowledgement is sent back to the sender. The path is held until all the data has been transmitted, after which the path is torn down as part of the tail fl it. The advantage of this approach is that the full link bandwidth is available to the circuit once it has been set up

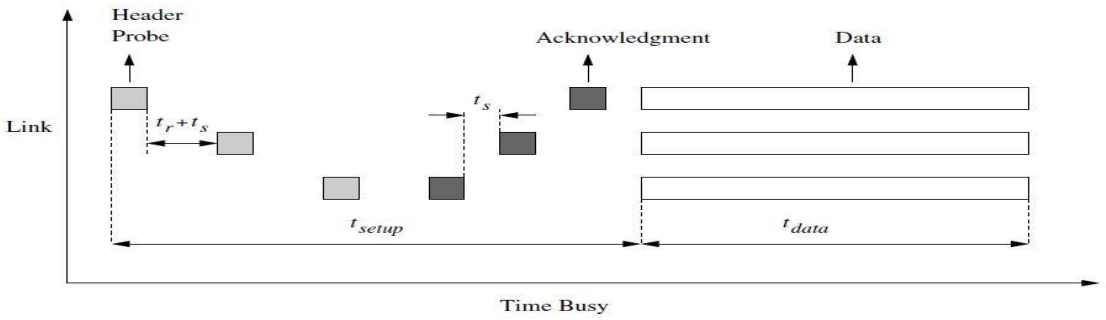


Fig. 3.8.a Circuit switching time space diagram [28]

In packet switching as shown in Fig. 3.8, instead of establishing a path before sending any data as is done in circuit switching, the packets are transmitted from the source and make their way independently to the receiver, possibly along different routes and with different delays.

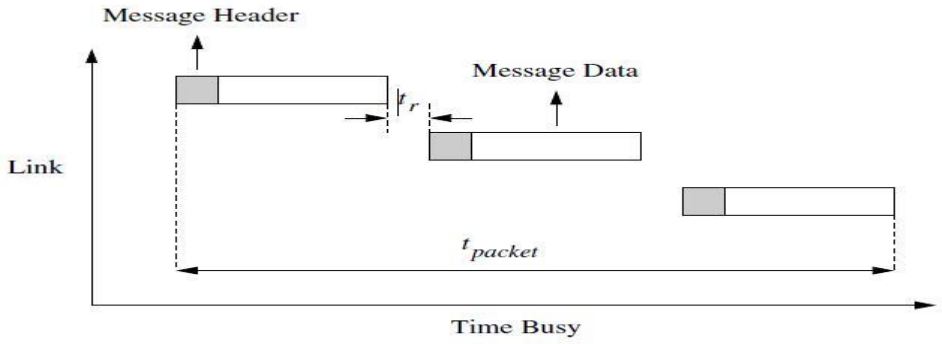


Fig. 3.8.b Packet-switching time space diagram [28]

While in circuit switching there is a start up waiting time, followed by a fixed minimal latency in the routers, packet switching involves a zero start up time, followed by a variable delay due to contention in routers along a packet's path. Without link reservation, multiple packets from different sources can arrive at a router and attempt to use a link at the same time. There are three popular packet switching schemes: (i) store and forward (SAF), (ii) virtual cut through (VCT), and (iii) wormhole (WH) switching. These are described below.

i) SAF Switching:

In this simple switching technique, a packet is sent from one router to the next only if the receiving router has buffer space for the entire packet. Routers forward a packet only when it has been received in its entirety. The buffer size in the router is at least equal to the size of a packet. Because of large buffer size requirements for this technique, it is not commonly used in NoC.

ii)VCT switching:

This technique as shown in Fig. 3.9, reduces the router latency over SAF switching by forwarding the first flit of a packet as soon as space for the entire packet is available in the next router (instead of first waiting for the entire packet to be received and then ensuring that sufficient buffer space is available in the next router before initiating packet transfer).

Cut-through routing is assumed to occur at the flit level with the routing information contained in 1 flit. This model assumes that there is no time penalty for cutting through a router if the output buffer and output channel are free. Depending on the

speed of operation of the routers, this may not be realistic. Note that only the header experiences routing delay, as well as the switching delay and wire delay at each router. The other flits follow the first fl it without any delay, if no space is available in the receiving buffer, no flits are sent, and the entire packet is buffered. Since the buffering requirements of this scheme are the same as that for SAF, it is also not frequently used in NoC.

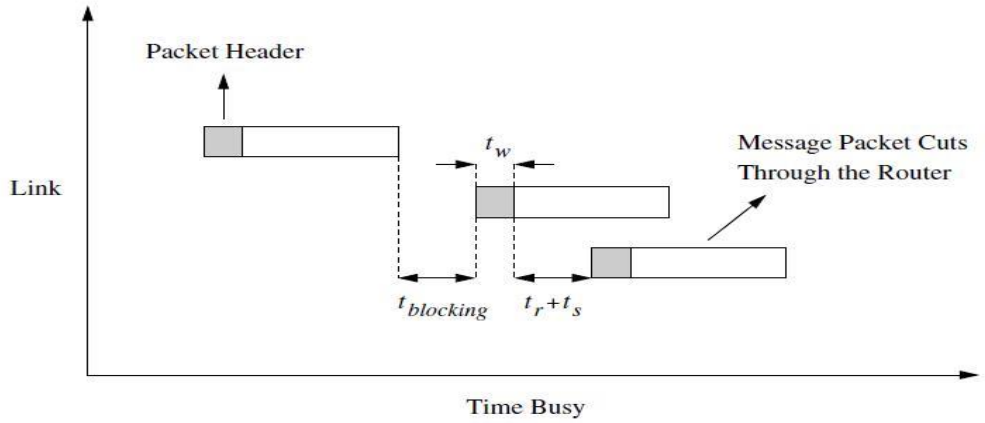


Fig. 3.9 Virtual cut-through switching time space diagram [28]

iii) Wormhole switching:

In this technique, buffer requirements are reduced to one flit, instead of an entire packet. A flit from packet is forwarded to the receiving router if space for that flit is available in the router. In wormhole switching, message packets are also pipelined through the network. However, the buffer requirements within the routers are substantially reduced over the requirements for VCT switching. A message packet is broken up into flits. The flit is the unit of message flow control, and input and output buffers at a router are typically large enough to store a few flits. For example, the message buffers in the Cray T3D are 1 flit deep, and each flit is comprised of eight 16-bit phits. The message is pipelined through the network at the flit level and is typically too large to be completely buffered within a router. Thus, at any instant in time a blocked message occupies buffers in several routers. The time-space diagram of a wormhole-switched message is shown in Fig.3.10

If there is insufficient space in the next router to store the entire packet, parts of the packet are distributed among two or more routers. Such a distribution of packets among multiple routers can result in blocking of links, which leads to higher congestion than in SAF and VCT. WH switching is also more susceptible to deadlocks than both SAF and VCT switching, due to usage dependencies between

links.

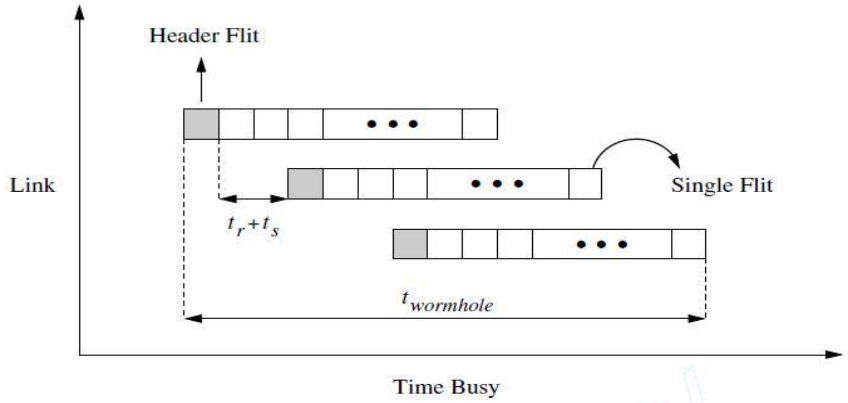


Fig.3.10 Wormhole-switching time space diagram [28]

3.2.3 ROUTING ALGORITHMS:

Routing algorithms [58] are responsible for correctly and efficiently routing packets or circuits from the source to the destination. Routing schemes can broadly be classified into several categories such as static or dynamic routing, distributed or source routing, and minimal or non-minimal routing. These are described below:

i) Static and dynamic routing:

Routing decisions in a NoC router can be either static (also called deterministic or oblivious) or dynamic (also called adaptive). In static routing, fixed paths are used to transfer data between a particular source and destination. This routing scheme does not take into account the current state of the network, and is unaware of the load on the routers and links when making routing decisions. One of the many advantages of static routing is that it is easy to implement, since very little additional router logic is required. Static routing also permits packets to be split among multiple paths between a source and destination, in a predetermined manner. If only a single path is used, static routing usually guarantees in-order delivery of data packets. This eliminates the need for adding bits to packets at the NI, in order to correctly identify and reorder them at the destination. In dynamic routing, routing decisions are made according to the current state of the network, considering factors such as availability and load on links. As such it is possible that the path between the source and destination changes over time, as traffic conditions and requirements of the application change. This also allows support for more traffic on the same NoCs topology. Static routing is used for cases where traffic requirements are steady and known ahead of time, whereas dynamic routing is more desirable when traffic conditions are more irregular and unpredictable. Examples of static (or oblivious) routing algorithms include dimension

order routing (DOR), XY, pseudo-adaptive XY, surrounding XY.

ii) Distributed and source routing:

Static and dynamic routing schemes can be further classified depending on where the routing information is stored, and where routing decisions are made. In distributed routing, each packet carries the destination address (e.g., *XY* co-ordinates or number identifying the destination node or router), and routing decisions are made in each router by looking up the destination addresses in a routing table or by executing a hardware function. Thus every network router can be considered to implement a function that takes the destination address of a packet as an input and generates a routing decision as an output. When a packet arrives at the input port of a router, the routing table is consulted (or routing logic is executed) to determine the packet's output port based on its destination address. In source routing, pre-computed routing tables are stored at a node's (or PE's) NI. When a source node transmits a data packet, the routing information is looked up at the source router (or NI) based on the destination address, and this information is added to the header of the packet. Each packet thus carries the routing choices in its header for each hop in its path. When a packet arrives at a router, the routing information is extracted from the routing field in the packet header. Unlike distributed routing, source routing does not require a destination address in a packet, any intermediate routing tables, or functions needed to calculate the route. Source routing requires additional routing information in a packet header, and the number of bits increases for longer paths. Additional routing tables may also be needed with specific entries for each source.

iii) Minimal and non-minimal routing:

Another way to distinguish between routing schemes is to classify them as minimal or non-minimal distance routing. A routing is minimal if the length of the routing path from the source to the destination is the shortest possible length between the two nodes. A non-minimal routing scheme does not have such constraints, and can use longer paths if a minimal path is not available. By allowing non-minimal paths, the number of alternative paths is increased, which can be useful for avoiding congestion. Non-minimal routing can, however, have an undesirable overhead of additional power consumption in NoC. A routing algorithm is typically responsible for ensuring freedom from deadlocks in packet switched NoC. A deadlock occurs when one or more packets in a network become blocked, and remain blocked for an indefinite

amount of time waiting for an event that cannot occur. When a flit is transferred between neighbouring routers, it releases a buffer in the first router and occupies the buffer in the second router. If the buffer in the second router is occupied, the flit is held until the buffer is available.

CHAPTER 4

RECONFIGURABLE ROUTER DESIGNS

The NoCs usually have three basics components-router, links, and wrappers. Router is the most important component of NoCs design. Various types of router are used in NoCs. The performance improvement, low area, high throughput, low latency and low power are basic requirements of the router design. In this work, we lay emphasis on why reconfigurable routers are required and how with optimum size buffer depths it can work to improve performance and efficiency. The basic principle is that channel can borrow or lend unit of its buffer from or to of its neighbouring channel.

In this chapter after selecting an appropriate topology, we have designed four reconfigurable routers, to improve overall efficiency and performance by reconfiguring buffer size according to the need of channel at run time.

Methodology & Techniques:

Methodology of design process for proposed routers of reconfigurable system:

1. DESIGN ENTRY (CODING)
2. SIMULATION
3. SYNTHESIS
4. IMPLEMENTATION

For the design purpose hardware description language VERILOG is used**.** For the simulation purpose MODELSIM Edition10.3 TOOL is used and test bench is written in Verilog by which working of routers designs is tested and simulated in Modelsim tool.

For the Synthesis purpose Xilinx ISE Design Suite 13.4 Tool is used. The Xilinx design tool gives a low level design. It shows the RTL VIEW, total design summary of circuit, and total power consumption by the circuit.

4.1 SELECTION OF TOPOLOGY:

In NoC selecting a topology plays vital role as flow control, routing strategy etc. are based on it. Of the various topologies briefed in Section 3.2.1, the Mesh is quite easier to design and it can also easily integrate regular-sized IP cores on a single chip.

Among the various mesh topologies, 2D Mesh is a very popular topology used for NoCs due to its less complex implementation, simplicity of the XY routing strategy and the network scalability.

Assessing various performance metrics [62] of NoC like Maximum End-to-End Latency, Dropping Probability and Throughput, 2D mesh has better throughput and dropping probability as compared to others. So for this work we have selected 2D mesh because of its flat configuration.

4.2 RECONFIGURABLE ROUTERS:

1. RRIE (Reconfigurable router to improve efficiency):

In this proposed reconfigurable router buffer size is reconfigured as per the need of the channel during run time without burdening routers and thus to increase overall performance.

2. HRRLPHP (Heterogeneous reconfigurable router for low power and high performance):

In this proposed heterogeneous reconfigurable router channel will take data either from its own channel or from neighbouring channel thus supporting heterogeneous data.

3. EDRRHT (Eight directional reconfigurable router for high throughput):

In this proposed eight directional reconfigurable router, four more directions (NE, NW, SE, and SW) are added. It now supports Eight Directions. It uses Buffer less storage concept to store the data of four newly added directions. Data of newly added directions share FIFO from their neighbour to store its data. Thus, the architecture requires less area.

4. SRRODFB (Smart reconfigurable router for online detection of faulty blocks):

In this proposed smart reconfigurable router with the help of loopback technique faulty blocks can be localised and removed, it is based on a new reliable NoC-based

communication approach called RKT-NoC.

4.3 DESIGN OF RRIE:

In this proposed reconfigurable router, efficiency has been improved by reconfiguring buffers according to the requirement, for use in NoCs. The router has low area, consume less power and provide high performance hence overall increase in efficiency. In this architecture it is possible to dynamically configure different buffer depth for the channel. Block diagram of RRIE is shown in Fig. 4.1. Broadly it is divided in the following parts:

- FIFO.
- Channel flow /Control logic.
- Crossbar.

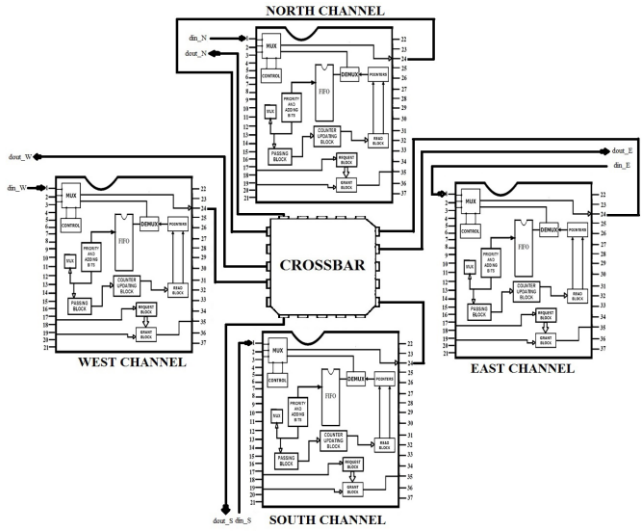


Fig. 4.1 Block diagram of RRIE

Detailed description is given below:

4.3.1 FIFO:

FIFO (First In First Out) buffer as shown in Fig.4.2 is used as input buffer to store the data temporarily. Whether the communication can be started or not is decided by the status of the FIFO. If it (FIFO) is empty then the data can be written in it and communication can start. When FIFO becomes full the data is forwarded to the next router (destination). The control logic controls the read and write operations of the

FIFO. In our design, the word length considered for FIFO is 8-bits but it can be changed as per the requirement of the design. The number of read and write operations in the FIFO buffers are stored by Read counter (rd_en) and write counter (wr_en) variables. These variables tell whether the FIFO is empty or full. The variable data_in_s corresponds to data that is an input to south FIFO while data_in_e and data_in_w are the data coming from east and west buffers, respectively, and in reference with the south buffer. The data output from the south buffer is represented by Data_out_s and it flows via channel to crossbar and is available as output with the logic applied. East and west buffers' data headed to their own buffers are represented by data_out_e and data_out_w, respectively. Acknowledgement signals are ack_s, ack_e and ack_w. For accepting data I/P of south FIFO south_enable, for accepting data I/P of east FIFO east_enable and for accepting data I/P of west FIFO, west_enable is used. For reset, rst is used. It reckons buffer into its initial state. For clock supplied to the entire design clk is used.

4.3.2 Channel Flow/Control Logic:

Channel Flow or Control Logic is shown in Fig. 4.3. It controls the arbitration of the ports and resolves contention related issues. Updated status of all the ports is kept by it and it has the information about the ports which are free and which ports are under communication process with each other. Packets having the similar priority and destinations as the same output port are scheduled. Channel Flow/Control Logic usually play the role of in-charge for processing the priorities among many different request inputs. This type of situation arises when, in a given period of time, many input ports request for the same output port or resource. The Channel Flow/Control Logic releases the output port connected to the crossbar once the last packet has finished transmission, so that other waiting packets are able to use the output. In case of an input channel connected to an output channel, the flits are sent one by one, and the pointers are updated as each flit is sent.

Each channel has the information about its number of buffer slots allocation. When the pointers shows an address which belongs to a neighbour buffer slot, the control logic allows the sending of the respective flits to the output of its channel. As we do not change the routing policy, there is no possibility of entering a deadlock situation. When, one channel is asking buffers from another channel which is also asking for

buffers can be a matter of concern. As, only the neighbours are asked about lending/borrowing, no cycle can be made, hence at the circuit level there is also no issue of deadlock.

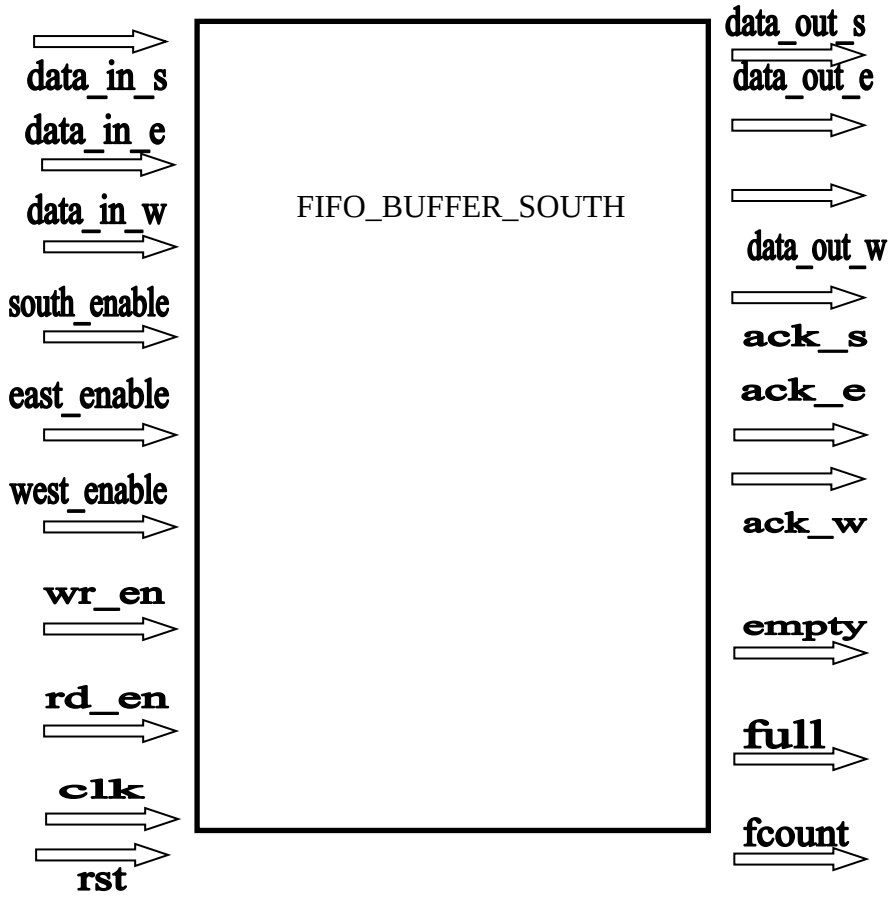


Fig.4.2 RRIE FIFO Buffer

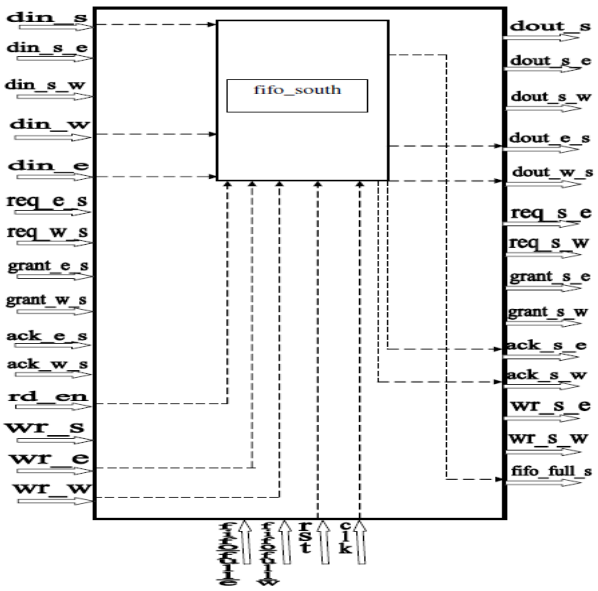


Fig. 4.3 South channel flow of RRIE

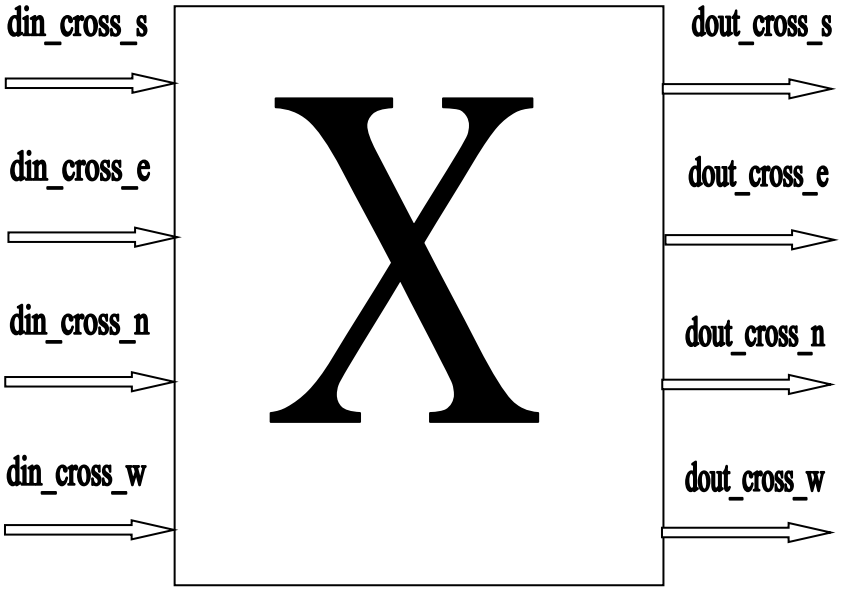


Fig. 4.4 Crossbar logic

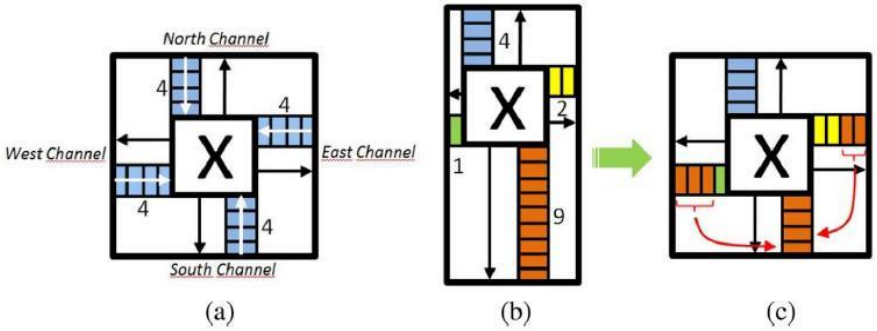


Fig. 4.5 (a) Router designed with FIFO depth of 4 (b) need of configuration of the router according to the need of data (c) Distribution of buffer words among the neighbour channel [67].

Fig. 4.5 shows the router designed with fixed buffer depth of 4, the need of reconfiguration of data and finally reconfigured router. By this reconfiguration process, proposed architecture uses less number of FIFOs, for storing large data. In this way proposed architecture has high performance.

4.3.4 Algorithm

For the reconfigurable router design discussed in this work, the FIFO resource was allocated dynamically so that the adjacent neighbouring FIFO could store their data in it as in the case when higher word length arrives that is at communication pattern change.

The Algorithm followed for the same has been brought up into working by designing via HDL and is discussed below:-

```
1.  for router=0 to number_of_routers
2.  begin: for channel=0 to channel=3
3.      begin: reset=0;current_max=0; right_max =0; left_max =0;
4.          if current_not_full && write_enable
5.              if current_enable=1
6.              begin: buf_mem[write_ptr]=data_in_current;
7.              current_max=current_max+1;
8.  right_max=right_max+1; left_max=left_max+1;end
9.              else if right_enable=1
10.             begin: buf_mem[write_ptr]=data_in_right;
11.             right_max=right_max+1;
        left_max=left_max+1;end
12.             else if left_enable=1
13.             begin: buf_mem[write_ptr]=data_in_left;
14.             left_max=left_max+1;end
15.     end
16. end
```

This Algorithm was used only to present a study of buffer distribution, but actually, this control can be realized in each router. An external control signal can be provided in each router to define the buffer depth of each channel. The routers are autonomous among themselves; the dependency just exists among the channels belonging to the same router. Using a traffic monitor it is possible to define at runtime the appropriated buffer depth to each channel, just indicating this size to the input ports responsible for this control. This Algorithm provides an alternative to calculate the ideal buffer depth according to the traffic in each channel as discussed in the objective.

4.3.5 Simulation Results:

Fig. 4.6 shows the simulation result of reconfiguration-al router. This simulation is performed using Model Sim PE 10.4a.

Fig. 4.6 shows how the data enters from N, S, and W to east FIFO then comes out from east buffer one by one.

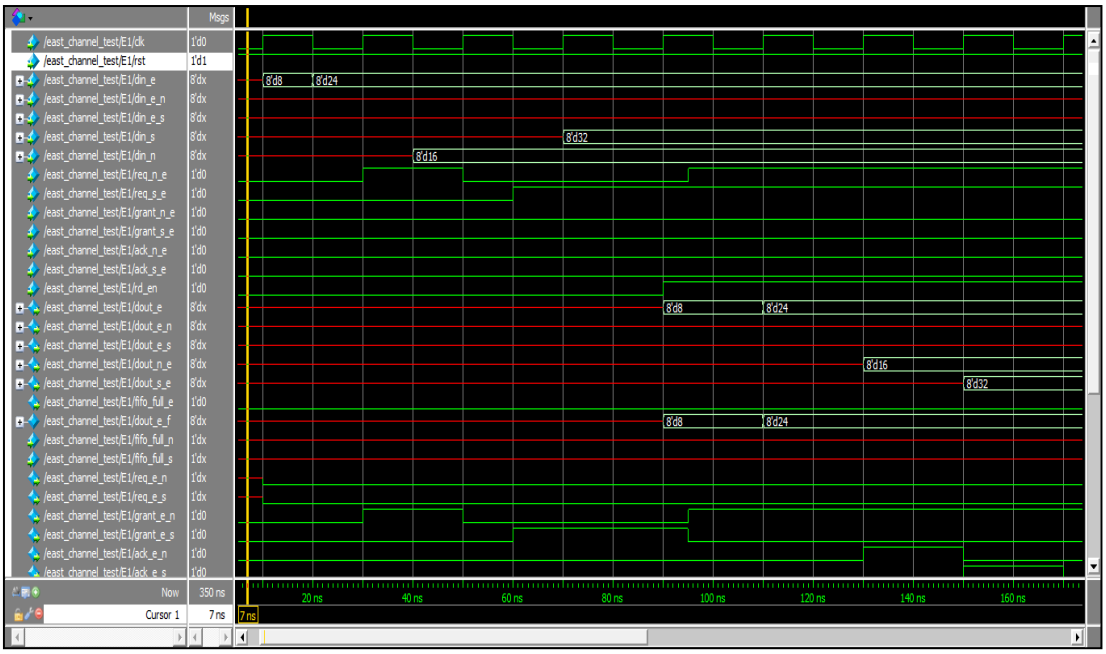


Fig.4.6 East channel of RRIE

In Fig.4.7 also as in Fig. 4.6 data from other channels enter into North Channel then stores in north FIFO and then comes out one by one out from north FIFO to respective destination.

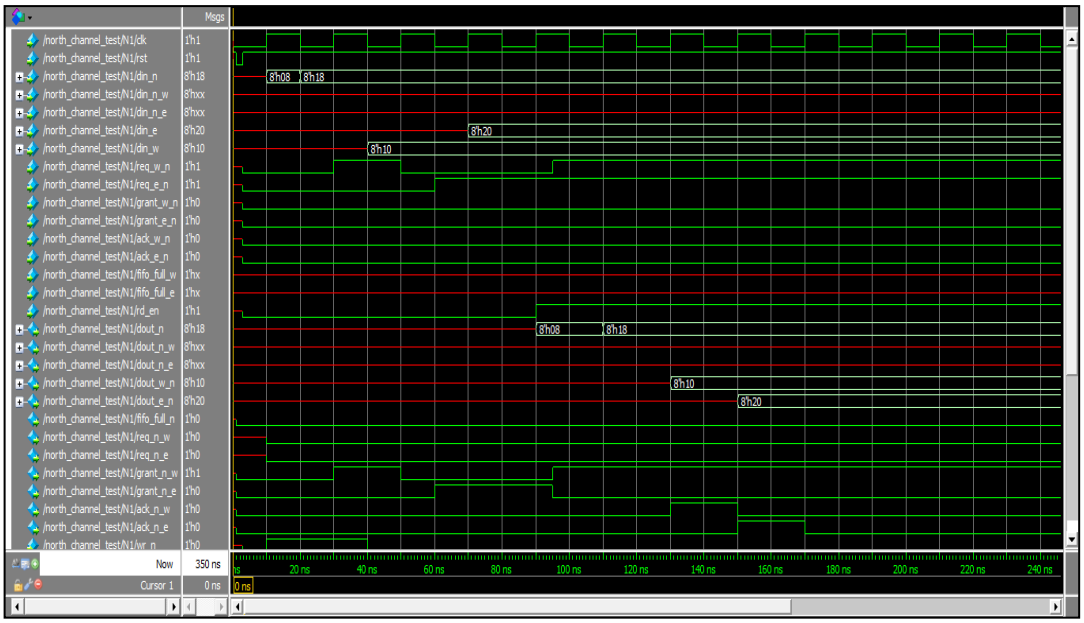


Fig.4.7 North Channel of RRIE

In Fig.4.8 also as in Fig. 4.6 data from other channels enter into south channel then stores in south FIFO and then comes out one by one out from south FIFO to respective destination

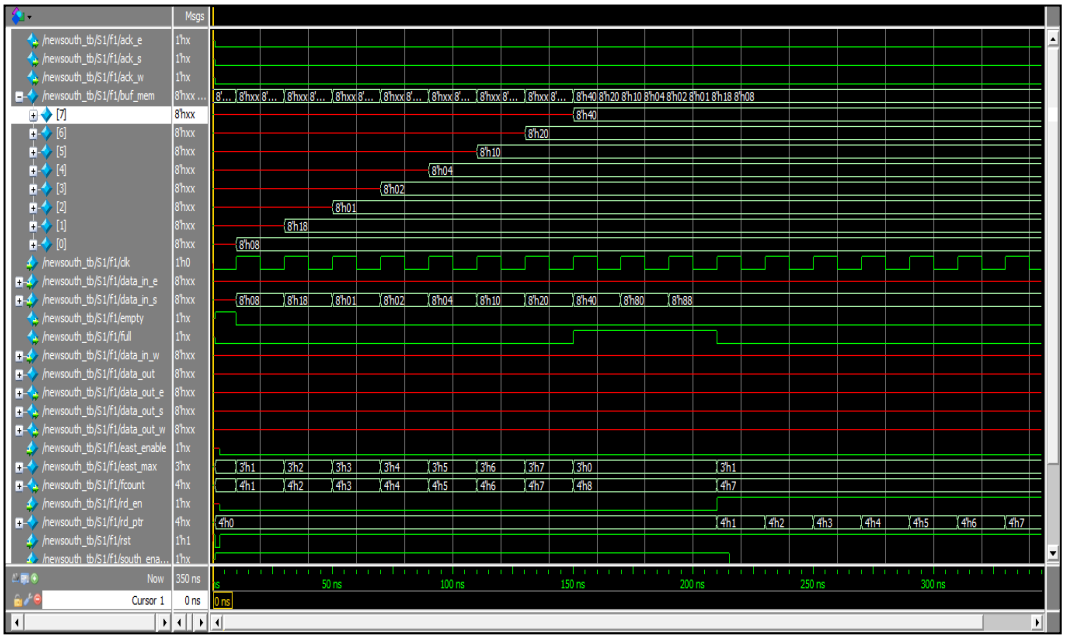


Fig.4.8 South FIFO buffer of RRIE

Fig. 4.9 shows how the data from all four channels is routed towards crossbar and then goes back to destined routers.

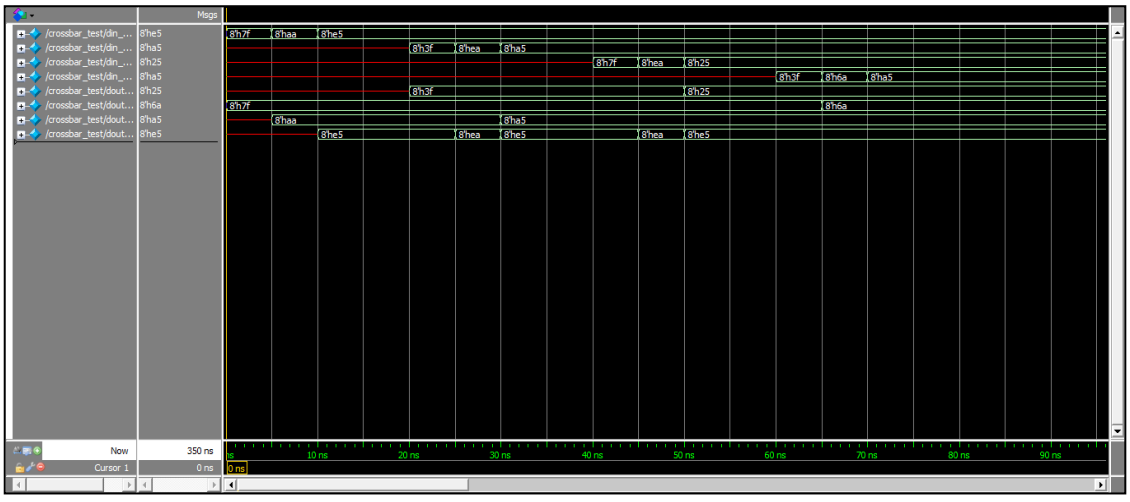


Fig. 4.9 Crossbar of RRIE.

Top module instantiates each and every internal module. The simulation results of this top module of RRIE are shown in Fig. 4.10

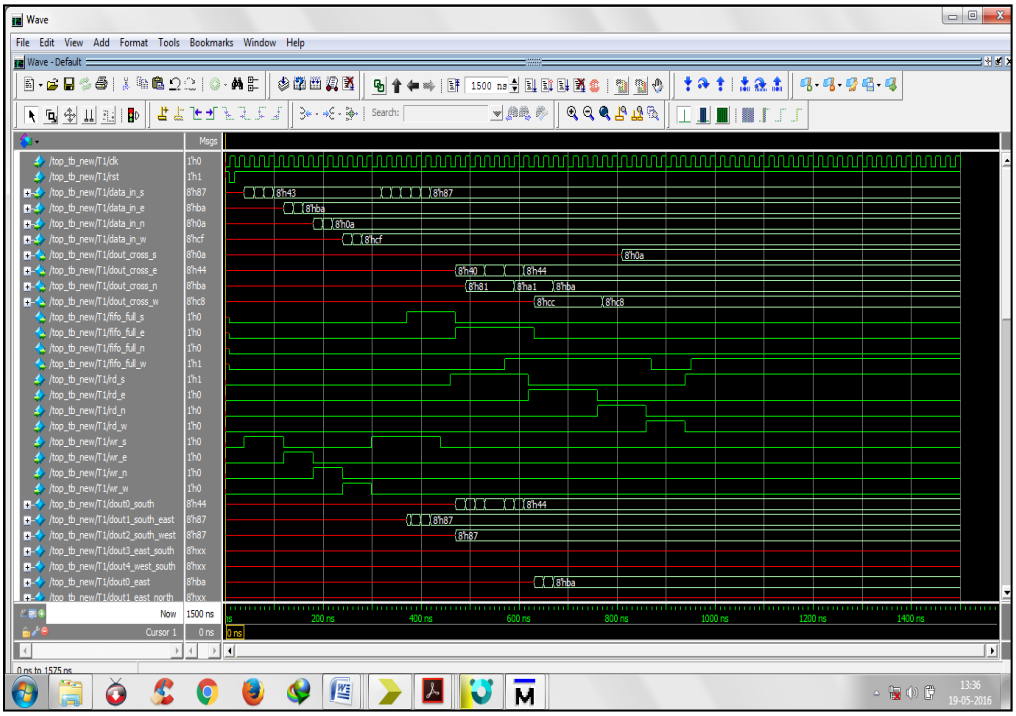


Fig. 4.10 Top Module of RRIE.

With the proposed router it is possible to have one single NoC connecting different applications that might change their communicating patterns at run time. In the same way, this architecture allows application updates without compromising the performance of the system. Meanwhile, if a homogeneous router had been used in these situations, design modifications at design time would have had to be made to achieve the optimum case. In such case, one would need to redesign the homogeneous NoC to set buffer sizes and position of the cores in the network. The technique proposed here avoids costly redesigns and new manufacturing.

4.3.6 Synthesis Results:

Fig. 4.11 shows RTL view of complete architecture after synthesis and it shows all four channels – south, west, east, north and the crossbar switch.

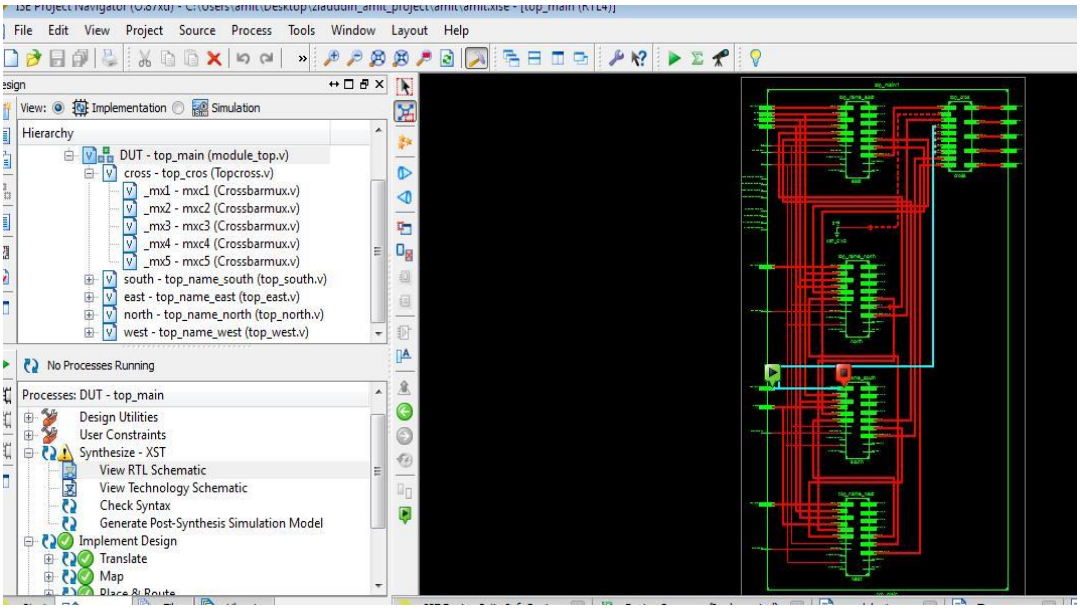


Fig.4.11 RTL view of RRIE

Fig. 4.12 shows RTL view after synthesis of the complete south channel. It shows FIFO, five multiplexers, and crossbar switch.

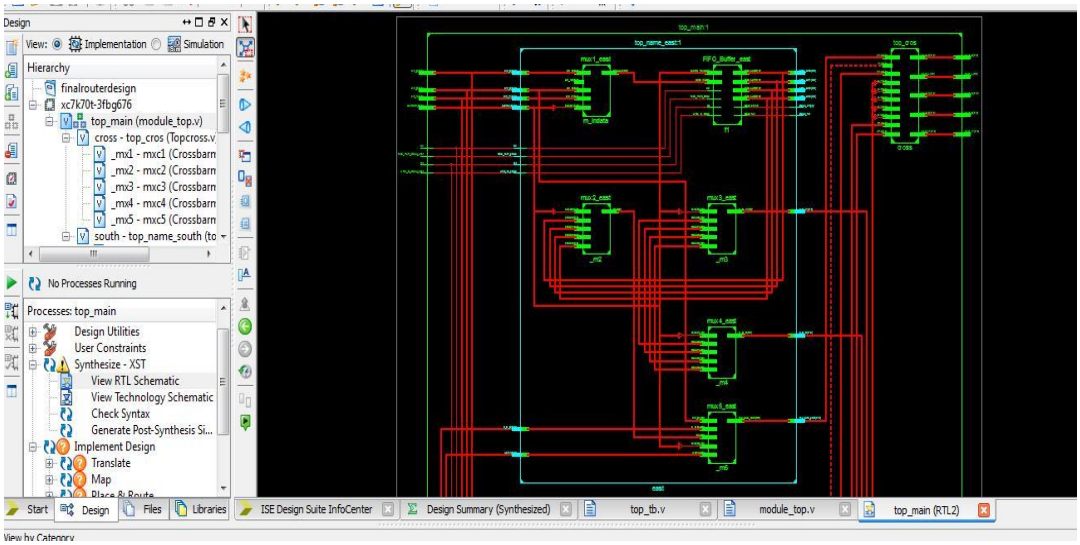


Fig.4.12 RTL View of south channel of RRIE

Complete router architecture has been designed and total power dissipation of the complete router architecture is estimated with the help of XPower Analyzer tool in Xilinx ISE. SPARTAN-6 FPGA is used for the implementation purpose. This total power dissipation calculated at a frequency of 100 MHZ is shown in Fig.4.13 it is

clear shown in Figure that the total power dissipated by the router architecture is around 15 milliwatt.

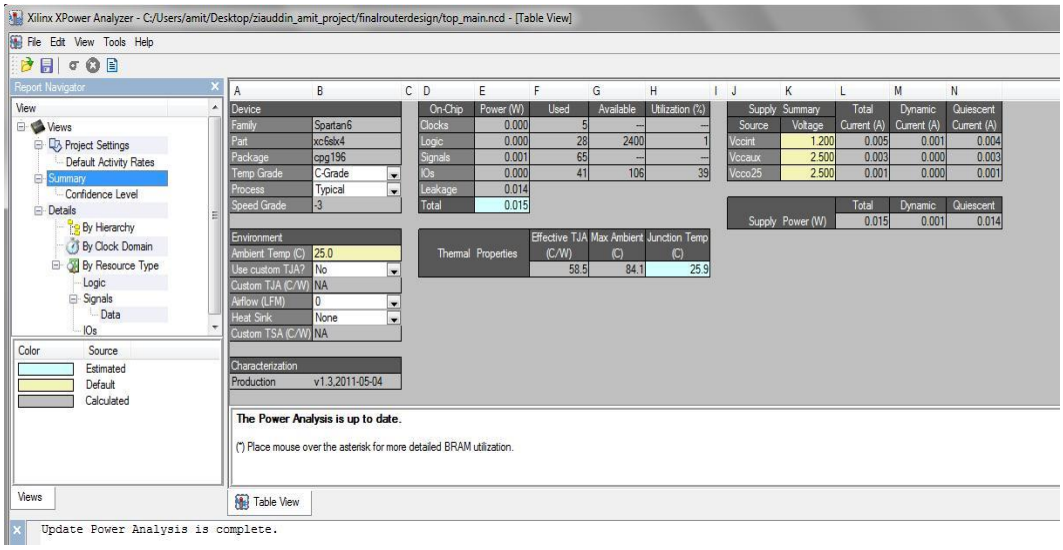


Fig. 4.13 Total power dissipation of RRIE.

Fig.4.14 gives the design summary of the reconfigurable router in terms of number of slice registers, number of slice LUTs, memory used etc.

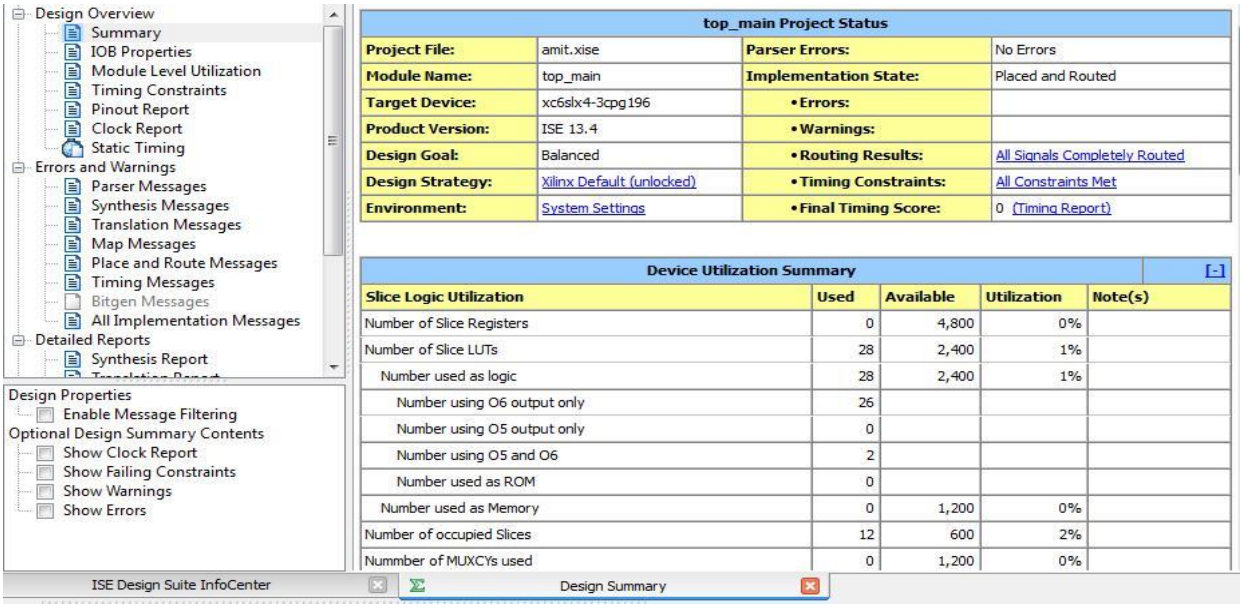


Fig.4.14 Design summary

To validitate the concept of proposed reconfigurable router and to prove the advantage of NoC with reconfigurable routers instead of homogeneous routers, various simulation results have been shown. It is possible to dynamically change the buffer depth of each channel in reconfigurable routers while maintaining the same

performance level. Also by simulation results it is verified that to obtain the same performance level with the reconfigurable router, the original architecture, i.e. homogeneous routers requires more buffers.

The RRIE, while obtaining the same performance as that of the original architecture, shows a significant reduction in power dissipation. It obtains the same performance as that of the homogeneous router but with a buffer depth which is 63% smaller. Also, RRIE shows that it is possible to reconfigure the router in accordance with the application while obtaining similar performances.

4.4 Design of HRRLPHP:

The problem with the RRIE is that in it when a channel starting to accept packets from its neighbour (left or right) it does not take any packet further from that channel. It means it is not supporting heterogeneous data. From the simulation results area is found to be slightly high. Both of these drawbacks are improved in the next proposed reconfigurable router, HRRLPHP.

In this proposed heterogeneous reconfigurable router it continues to take the packet input from its own channel even if a channel is taking data from its neighbouring channel. So, it will support heterogeneous data. To reduce the area, two tag bits to the packets in packet switching or flits (in wormhole switching) have been added. These bits are added before storing the packets in the FIFO to indicate the channel to which this packet or flit was input. Tag bits are '00' when the packets are directly coming to the concerned channel, '01' when packets coming from the right neighbour and '10' when the packets are coming from the left neighbour. These tag bits are also used for indicating the direction of output from the crossbar. For an 8 bit packet, the first two bits are used to indicate the address of the next router. Control circuitry is implemented in the present router by including features like acknowledge for writing, acknowledge for reading, request and grant signals for writing in another channel's FIFO.

Fig.4.15 shows the proposed router architecture. Proposed architecture has 4 channels, namely south, east, west and north. These channels are connected with each other and through the crossbar switch.

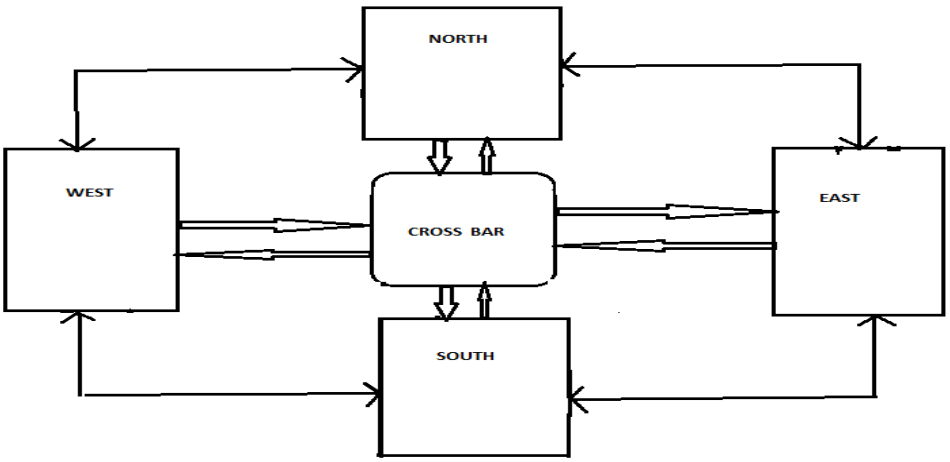


Fig.4.15 Block diagram of HRRLPHP

4.4.1 FIFO Buffer:

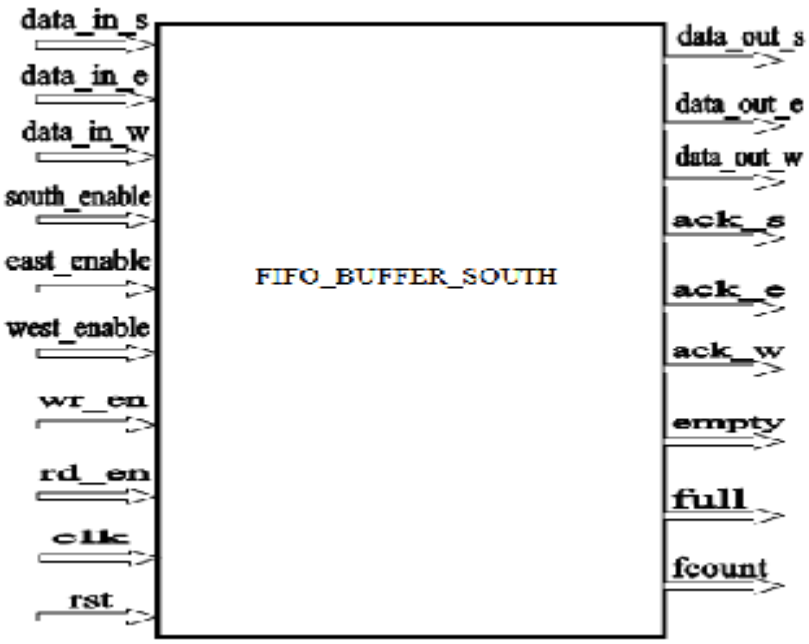


Fig.4.16 FIFO buffer of HRRLPHP

4.4.2 Modified South Channel:

Fig. 4.17 shows how data flow in South FIFO is out then depending upon control logic it is transferred to east, west, north FIFO. Similarly data from other three FIFO's are out and send to corresponding destination depending upon control logic.

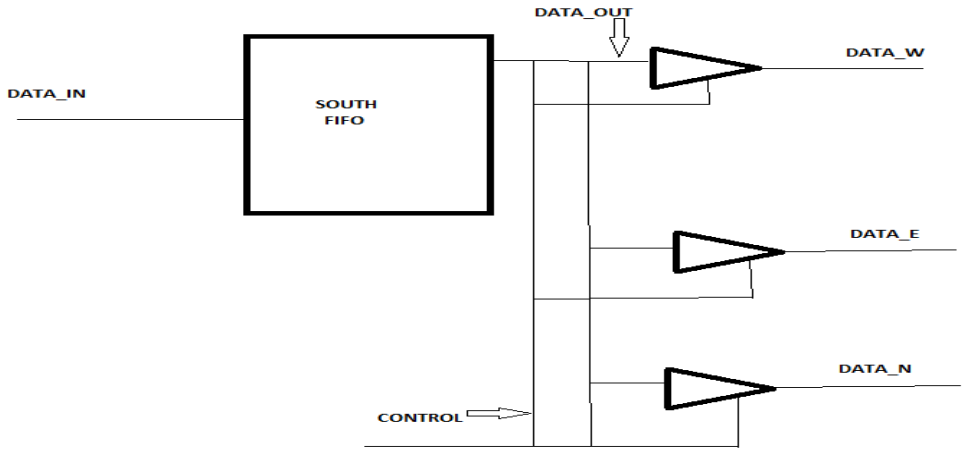


Fig.4.17 South Channel of HHRLPHP

In the following Fig. 4.18, shows the input output pin diagram of RRIE.

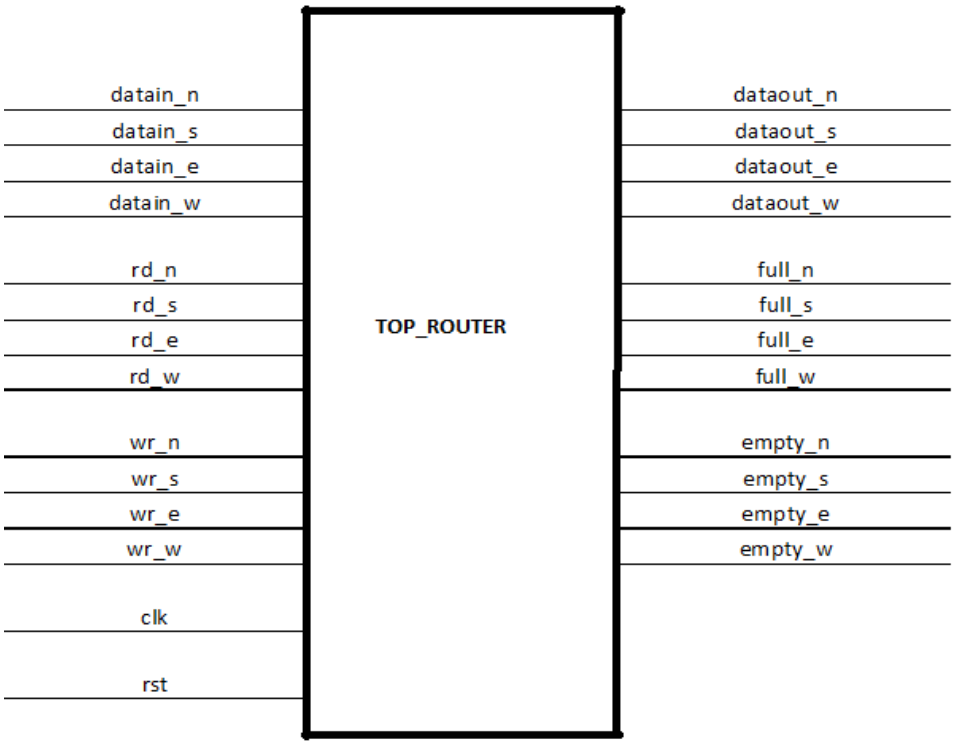


Fig.4.18 Pin Diagram of HRRLPHP

4.4.3 Simulation Results:

By giving some data into FIFO, stack memory locations could be checked. Now simulation waveforms of FIFO are shown in Fig.4.19. Data, 001, 111, 110, and 101

enter in FIFO through the input. These data values are store in F.I.F.O locations. In the simulated waveform we can check that FIFO_ buffer locations or memory locations are occupied by this series of data or not. It can be seen in the Fig. that all four memory locations are occupied by a series of data.

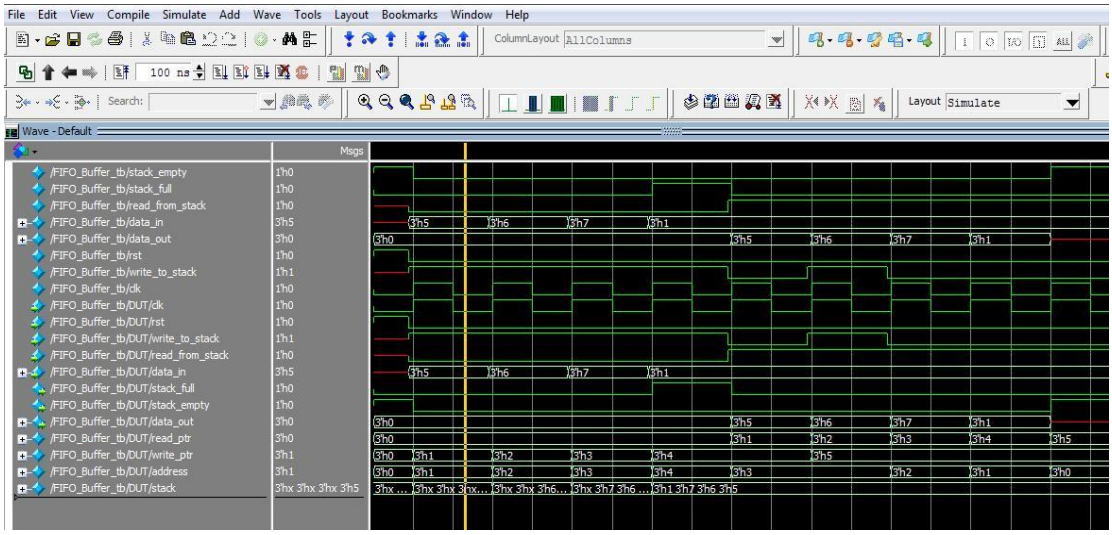


Fig.4.19 Waveform of the FIFO of HHRLPHP

Fig. 4.20 shows how data is entered from all directions as an input to multiplexer and then depending on selection line data entered from west is finally out.

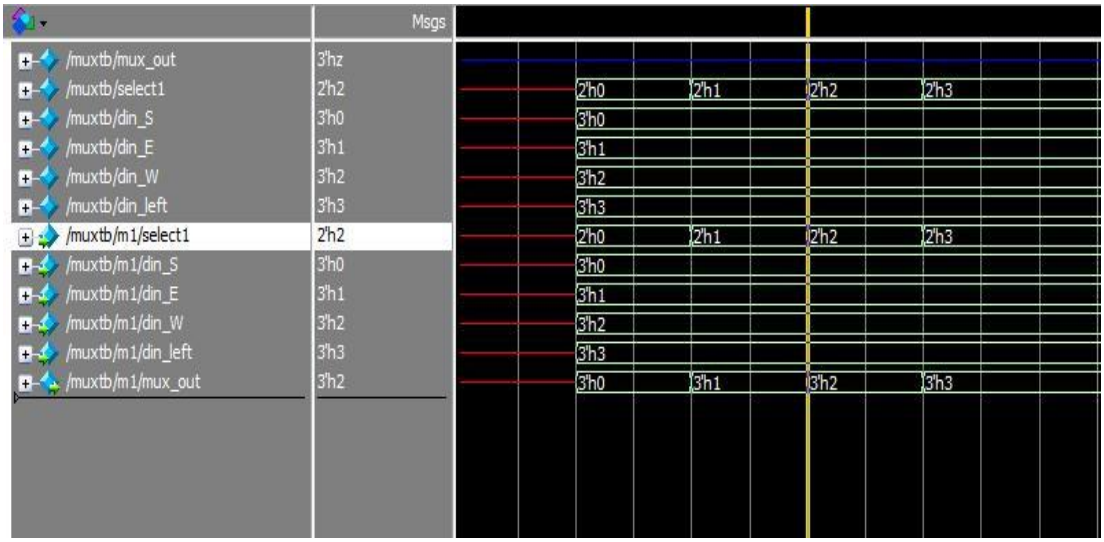


Fig. 4.20 Waveform of multiplexer of HRRLPHP

Fig.4.21 shows the complete south channel waveform. It shows that all four channels are connected with crossbar switch because final output is taken from the

crossbar. Waveform of One value stored in West FIFO Stack and two values stored in East Stack.

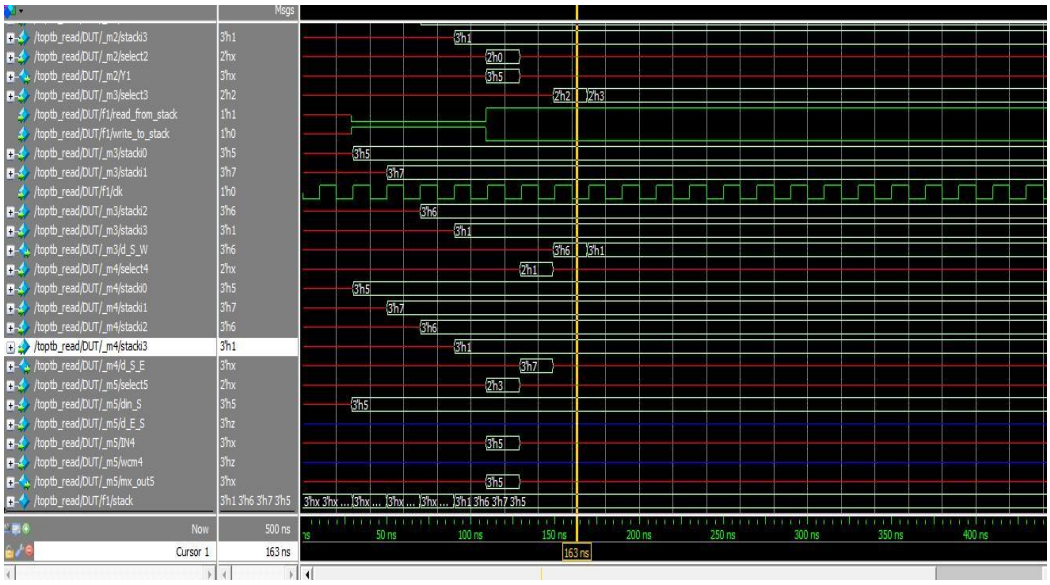


Fig.4.21 Complete south channel waveform.

When 9 values are entering through south channel, 4 values store in south, 3 values store in east & 2 values store in west. Fig. 4.22 shows the waveform of west stack and east FIFO stack. Fig.4.22.a shows, how FIFO buffer of south, west & east is occupied by data value and also the waveform of stored value in South, East, West channels.

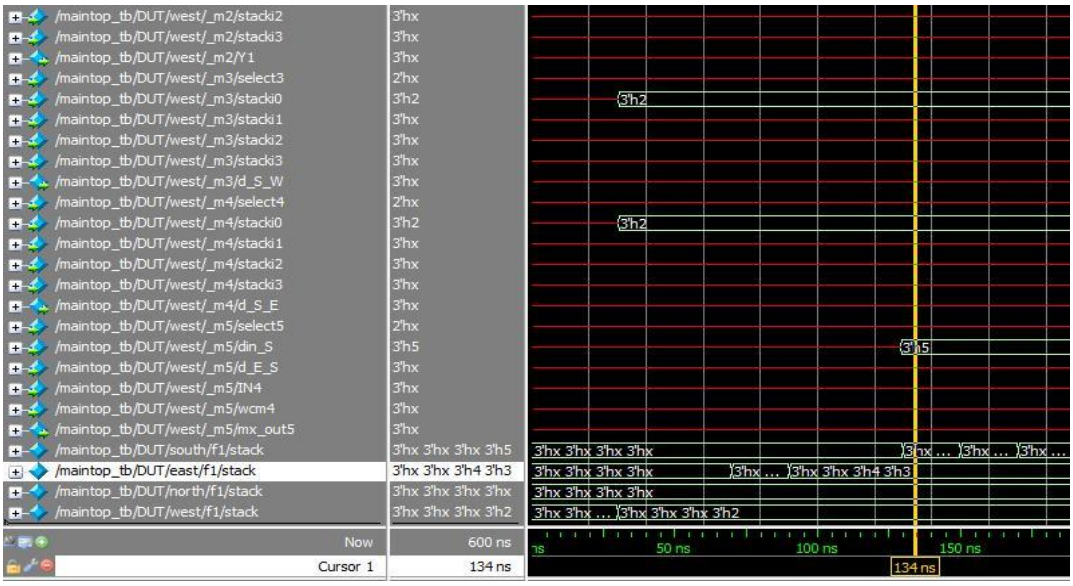


Fig.4.22.a West FIFO Stack and East FIFO Stack

Fig.4.22.b shows the values stored in south, east, and west FIFO Stack. Final output of south channel is coming out from the mux5 of south channel. It is connected to

crossbar switch.

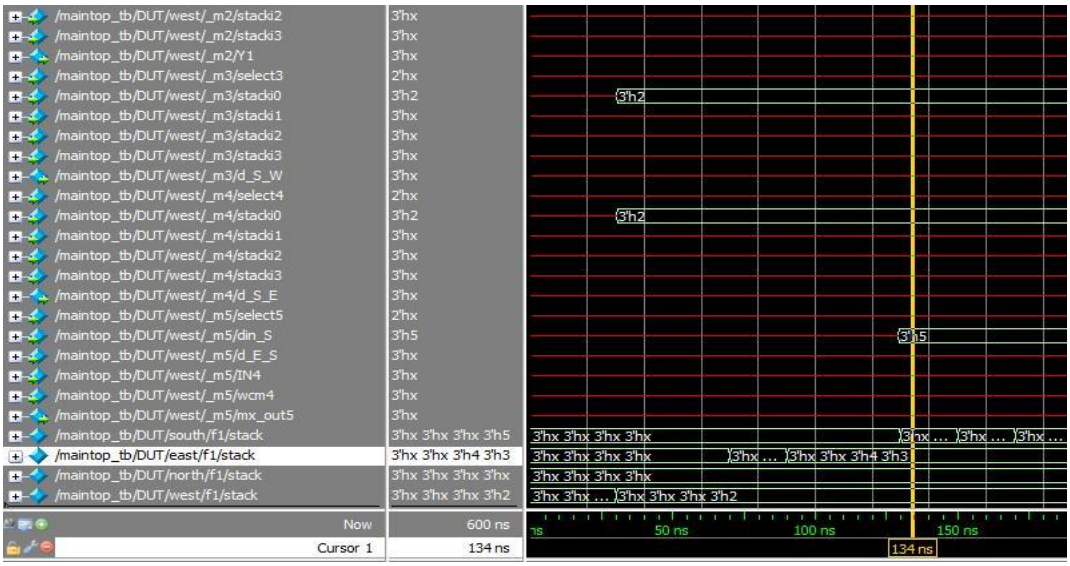


Fig.4.22.b stored value of south, east, west FIFO Stack

Fig. 4.23 shows the waveform of final output dout_S after simulating the code in Modelsim software.

From the simulation waveform it can be seen that final output of dout_S is same that is entered in input of south channel. Its synthesis process is done by using Xilinx ISE tool and the resultant RTL view is shown in the proceeding figures.

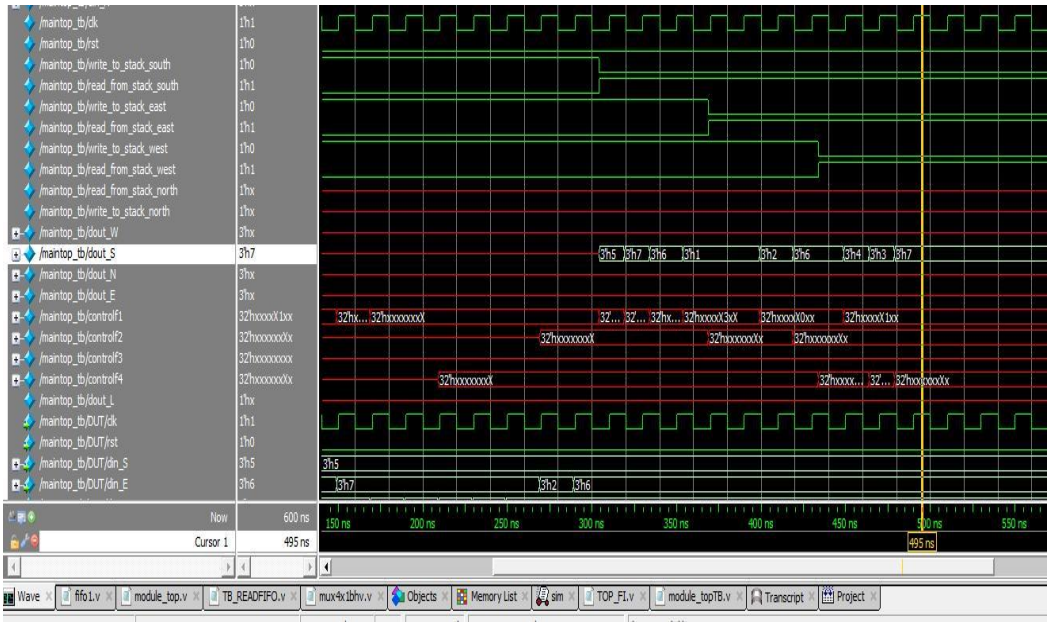


Fig. 4.23 Waveform of final output dout_S of HRRLPHP

4.4.4 Synthesis Results:

Verilog HDL Code is successfully synthesized using Xilinx tool. The output as the RTL view of FIFO is shown in Fig.4.24.

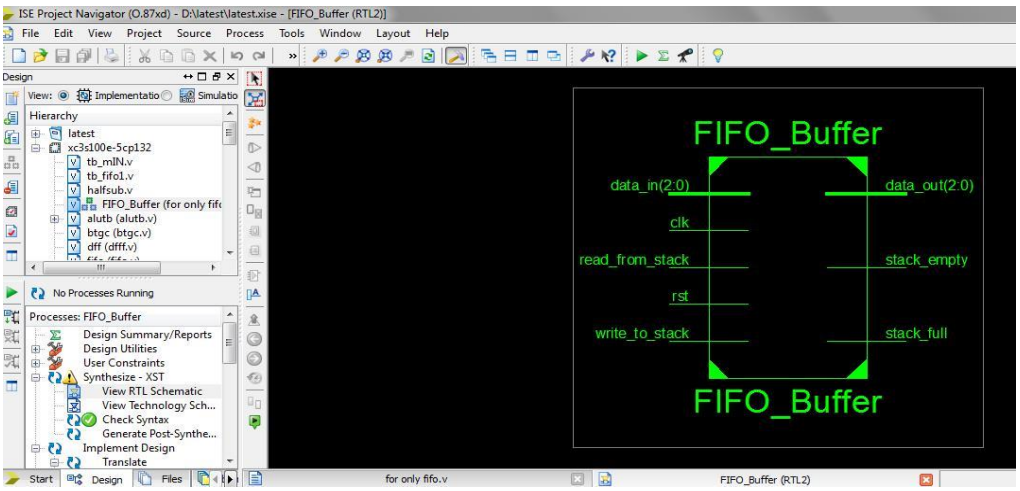


Fig.4.24 RTL View of FIFO of HRRLPHP

A detailed gate level view of FIFO by post synthesis is shown in Fig. 4.25.

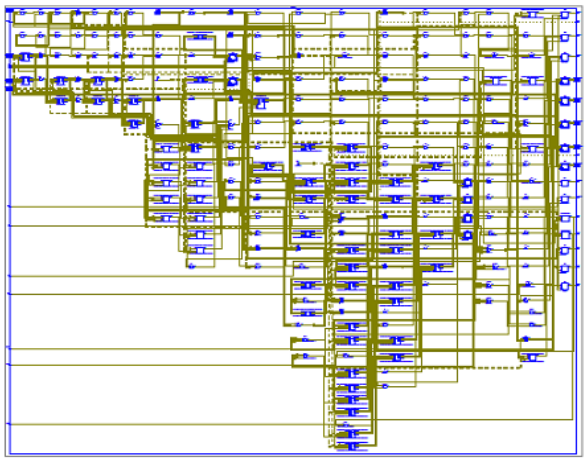


Fig.4.25 Detailed gate level view of FIFO of HRRLPHP

In Fig. 4.26 post synthesis result of south channel of HRRLPHP is shown.

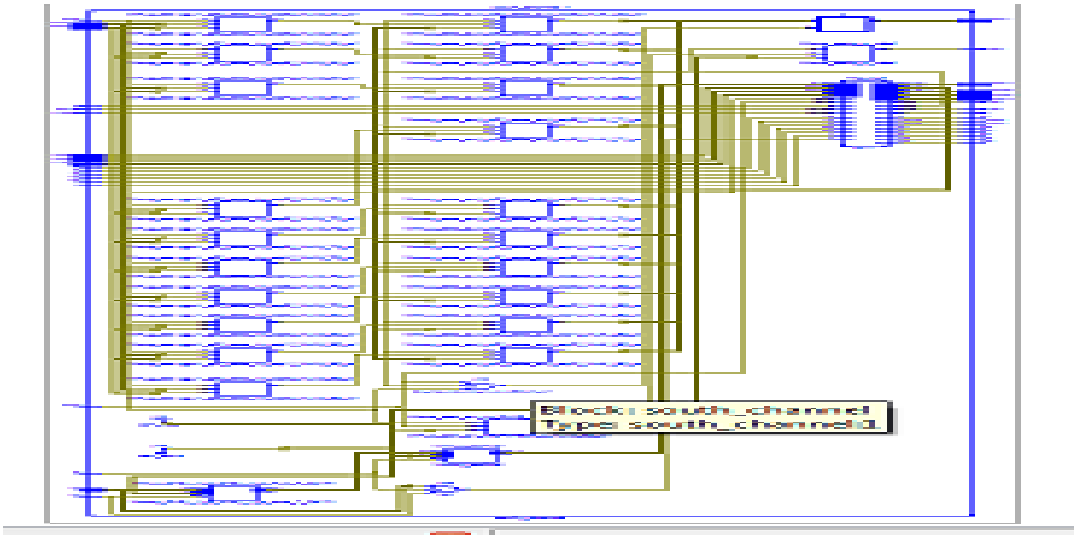


Fig.4.26 RTL view of south channel of HRRLPHP

Fig. 4.27 shows the RTL schematic of crossbar of HRRLPHP

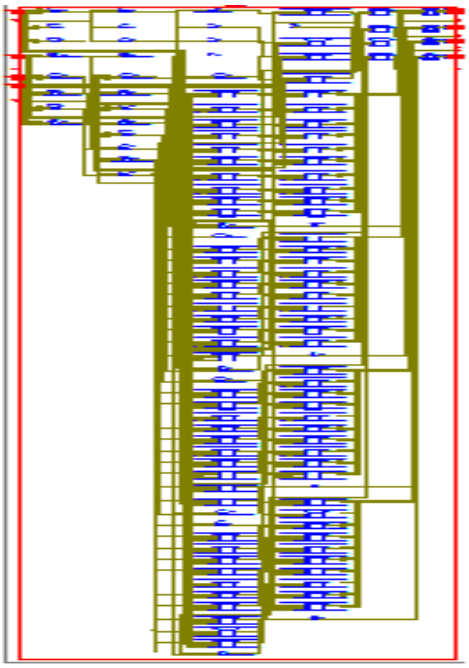


Fig.4.27 RTL view of Crossbar of HRRLPHP

Fig. 4.28 shows the area estimation utilized in the designing of HRRLPHP.

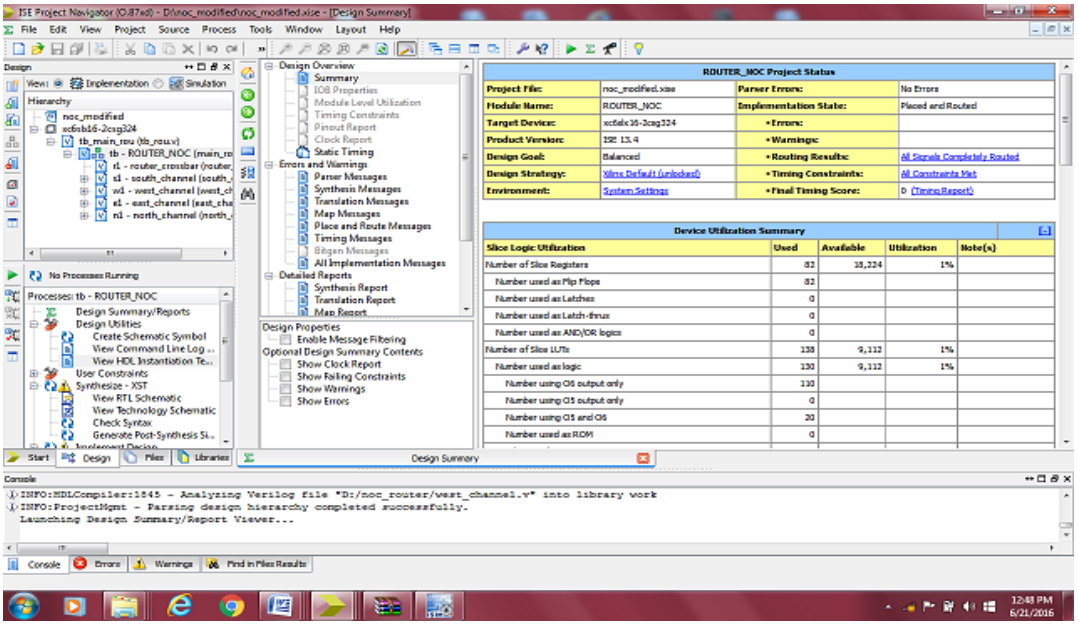


Fig.4.28 Area of HRRLPHP

In Fig.4.29 total power dissipation is shown and it is that the total power dissipated by HHRLPHP is 0.020 W.

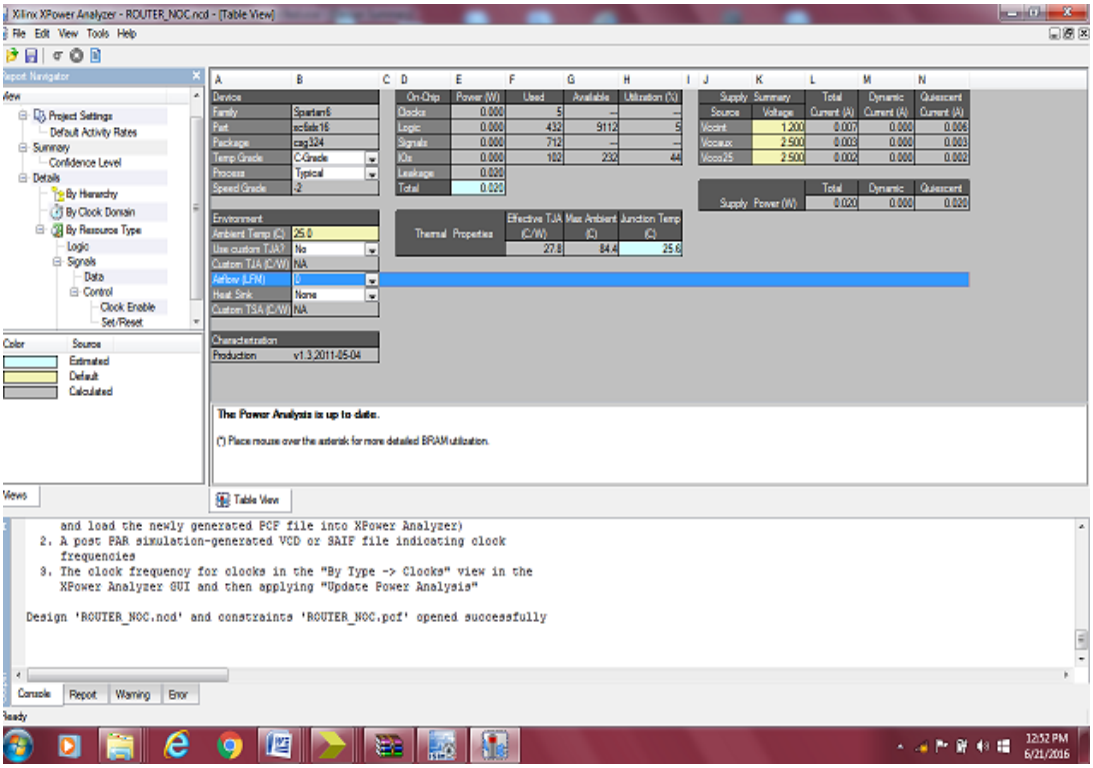


Fig.4.29 Power of HRRLPHP

The proposed heterogeneous reconfigurable router design for low power and high performance architecture is far more efficient technique for integrated circuits on chip than proposed reconfigurable router. This architectural technique saves power, area and decreases the latency by a significant margin.

Using the homogeneous router, in order to sustain the same performance that the proposed heterogeneous buffer scheme can achieve, a buffer of fixed size would have to be much larger, with many more flip-flops. Thus, the proposed router reduces power dissipation considering the same performance, once it uses much smaller buffers, hence, less flip-flops and less power dissipation. The latency is minimized by replacing normal crossbar switch with multiplexer based crossbar switch. The area is minimized for heterogeneous reconfigurable router by addition of two tag bits for direction. The implementation of technique and its simulated test results show that the overall network performance of the proposed technique is better compared to the conventional method. The improvement in performance is significant. The power consumption is reduced to .020 W and on chip area utilization is only 53 mm^2. The throughput, however, is degraded marginally at 3% due to elimination of buffer (which can be improved in next proposed reconfigurable router), which is tolerable and insignificant for most of the applications. The power and area improvement is huge and thus overall good improvement has been achieved.

4.5 DESIGN OF EDRRHT:

In HRRLPHP, low throughput is realised because data is routed in four directions only. Also it does not have semi buffer concept to increase efficiency.

To improve the above drawbacks we propose EDRRHT where, four more directions (NE, NW, SE, and SW) are added. It is now able to support Eight Directions. It also support buffer less storage concept to store the data of four newly added directions. Data of newly added directions share FIFO from their neighbour to store its data. Thus, the architecture requires less area. Data from one direction can now be routed to seven directions in comparison of previous one in which data from one direction can be routed to three directions only. The Newly designed architecture provides more routes to router to route the data thus it is helpful in decreasing critical path length of a design and help in increasing the speed of the NoC. With this newly designed

architecture, we have also used rectilinear stennier tree path to shorten the data path to router.

To obtain high throughput we have added new features in EDRRHT, like for reading and writing acknowledgement signals are introduced, for writing in neighbouring FIFO signal grant and request is added. Instead of using a normal fall-through FIFO, we have used a circular FIFO. Two tag bits are added to indicate the channel to which this packet or flit was input.'00' for directly input packets and '10' for packets coming from the right neighbour and '01' for packets coming from the left neighbour. The direction of output from the crossbar are decoded by tag bits.

Basic components of proposed architecture in Fig .4.30 is same as that of previous one but with the addition of four arbiters NW, NE, SW, SE which supports buffer less concept and used to store data from neighbours depending on priority which is further dependent on priority table .

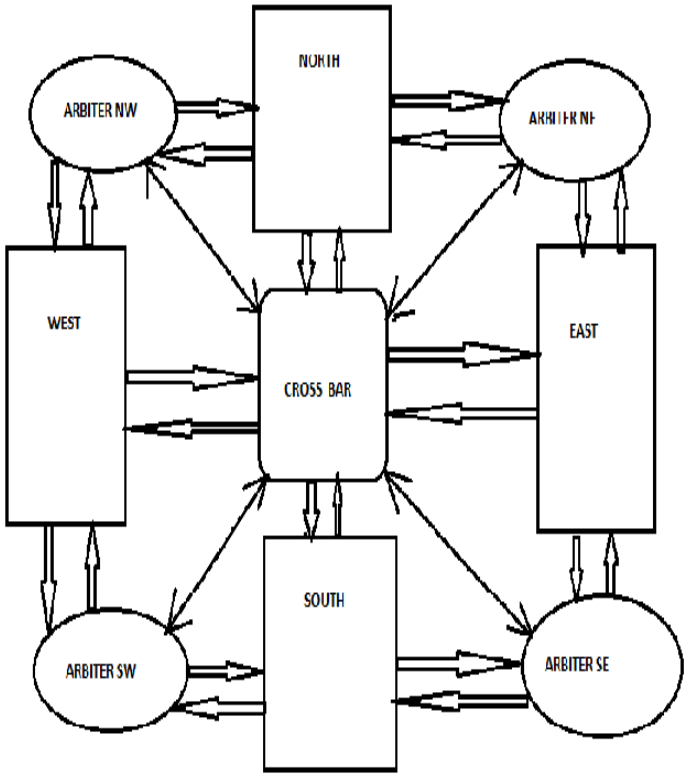


Fig.4.30 Block diagram of EDRRHT.

Detailed block diagram of South West arbiter is shown in Fig.4.31 along with the in / out signals. For the same data input and output signals red colour signals have more priority than green input / output signals. Similar block diagram is applicable for other arbiters with analogous change in signals.

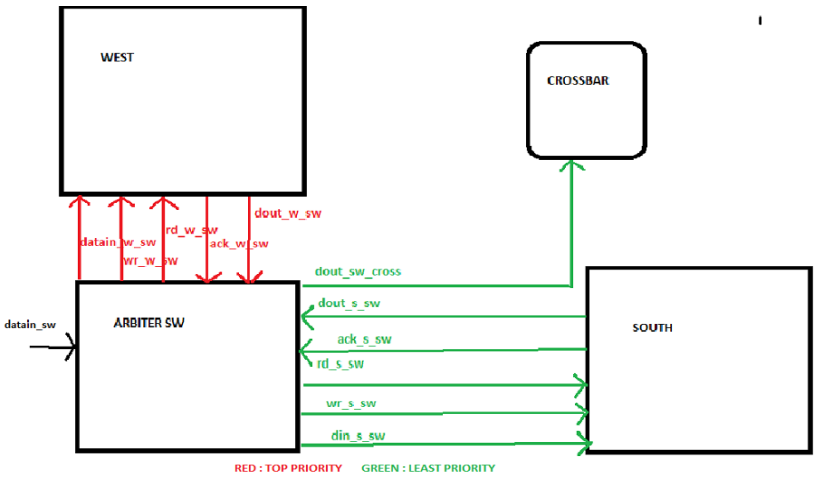


Fig.4.31 Detailed Block diagram of South West module.

4.5.1 Buffer Selection Criterion:

Table 4.32 shows which buffer is selected depending on the position of arbiter. If Bottom_right arbiter is selected then right buffer is selected. Similarly left buffer is selected if the position of arbiter is Bottom_left. Likewise, right and left buffer is selected with arbiter position as Top_left and Top_right.

Position of Arbiter	Buffer Selection
Bottom_right	Right
Bottom_left	Left
Top_left	Right
Top_right	Left

Table 4.32 Buffer selection

4.5.2 Flowchart:

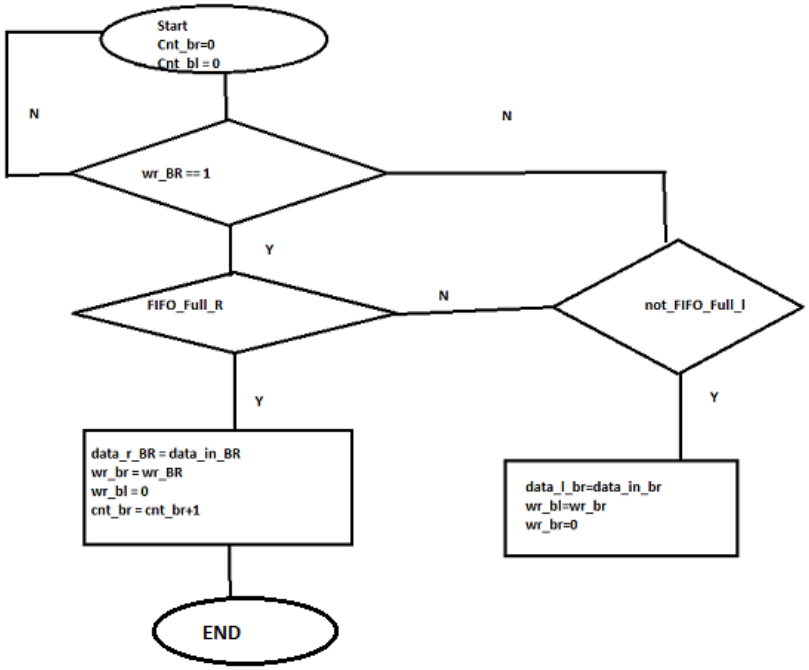


Fig. 4.33 Flowchart for Bottom Right writing

4.5.3 Algorithm:

1. Data from another router to S, E and W will get stored in S, N, and E&W channel respectively.

2. In case, when data in any direction exceeds its F.I.F.O size in its respective channel then it will request its neighbouring channels for storing its data in neighbour.

3. As soon as any channel receive sharing request from its neighbour, it will first check whether its F.I.F.O is full or not.

3.a If it is not full & it has received sharing request from both the neighbour then it will give grant to right neighbour on priority basis .

3. b If it has received sharing request from any one of the neighbour then it will grant to that neighbour.

3. c If F.I.F.O of the respective channel is full then it will grant to any neighbour.

4. As soon as the requesting channel receives grant from one of its neighbouring channels, it passes its data to neighbouring channel (left channel on priority) with write signal.

5. Routers' channel is designed to make it store heterogeneous data which is not possible in RRIE as it stores homogeneous layered data.

6.

S
S
W
W
E
E
W
W
~~S~~
~~S~~

Original Data

S
S
W
E
S
E
W
W
S
W

Proposed data

It is clear that in original data south channel cannot read twice but in proposed data supporting heterogeneous data any channel can be read any number of times.

7. On read request, the channel first reads its own data from its own channel. When it finds that all stored data have been read and it has stored its data in neighbour also then it will generate command to read its data from neighbour.

It is seen from the data format in Fig. 4.34 that for supporting heterogeneous data 3 bit source address is given along with destination address and data packet.

Data Packet:

Destination_address	Data_Packet

9 7

Source_address	Destination_address	Data_Packet

11 9 7 0

Fig.4.34 Data format in F.I.F.O supporting heterogeneous data

4.5.4 **Simulation Results:**

Simulation results in Fig.4.35 shows that the south F.I.F.O is accepting the data from south and south east arbiter.

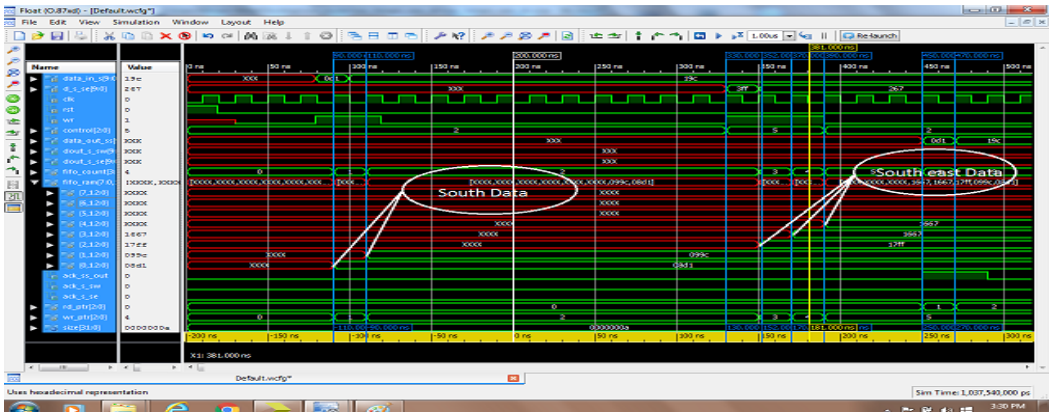


Fig.4.35 South FIFO OF EDRRHT

Below simulation results in Fig.4.36 shows that with Ack signal high south data is out from south and SE arbiter data is also stored and out from south F.I.F.O respectively.

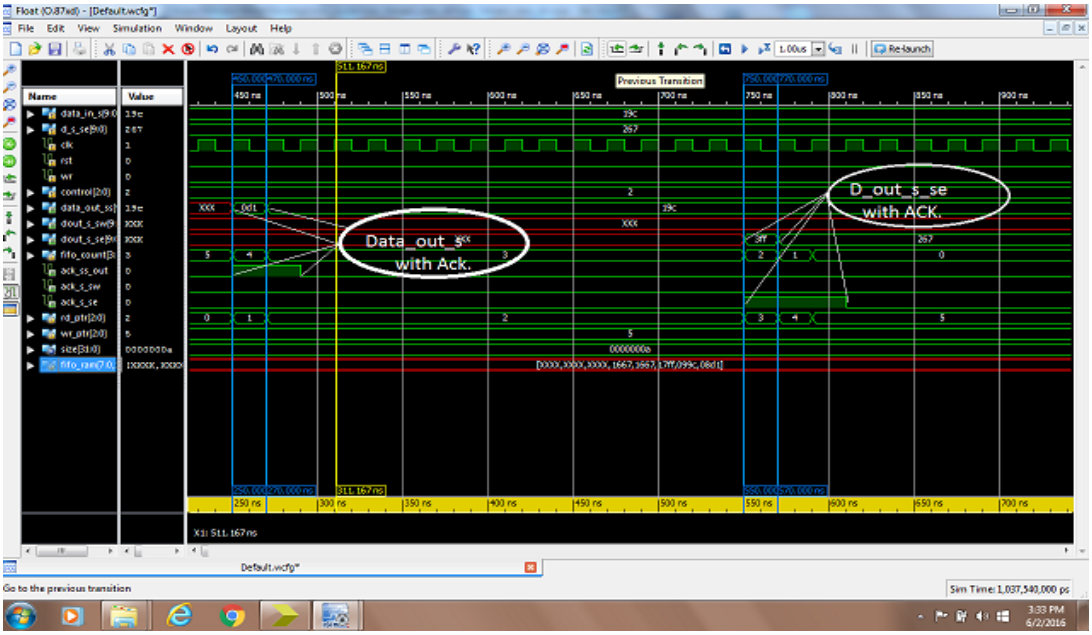


Fig.4.36 South channel output of EDRRHT

Fig. 4.37 shows that simulation depending upon the priority various data input from one out of eight locations are out from another one out of eight locations for e.g. data input in south and north is out from west FIFO.

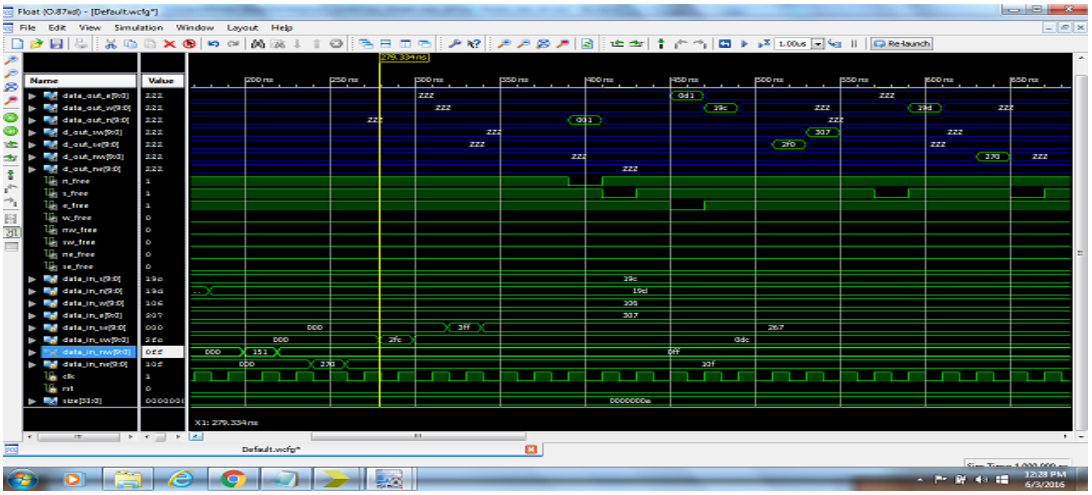


Fig.4.37 Top module of EDRRHT

It is obvious from Fig4.38 that data in west FIFO get routed to North and South respectively.

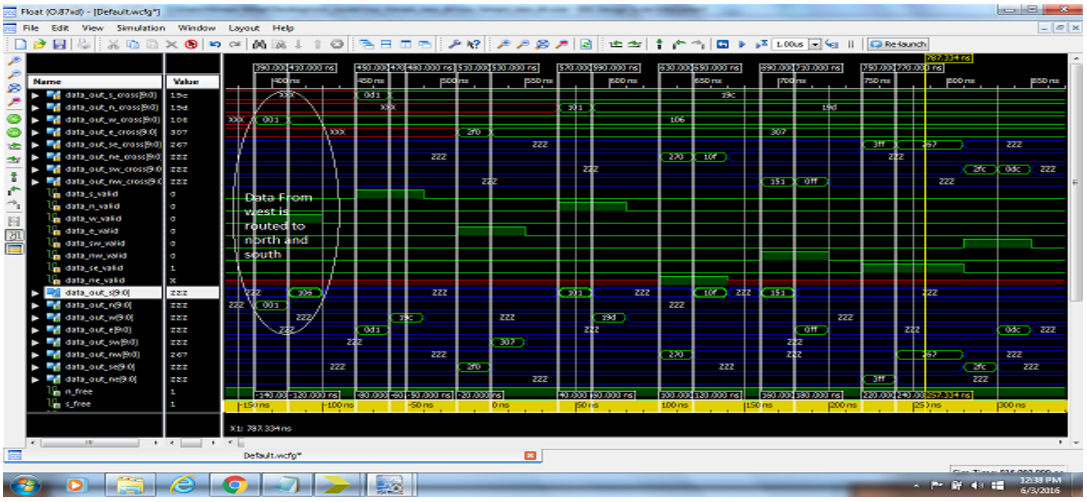


Fig.4.38 Crossbar of EDRRHT

4.5.5 Synthesis Results:

Fig. 4.39 shows the RTL schematic of main module of EDRRHT.

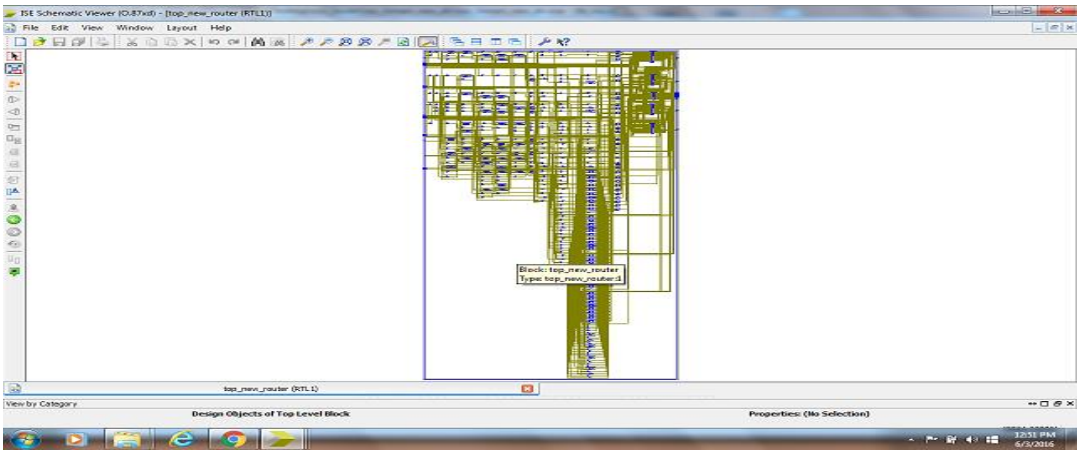


Fig.4.39 RTL view of main module of EDRRHT

RTL schematic of south channel of reconfigurable router EDRRHT can be seen in Fig. 4.40.

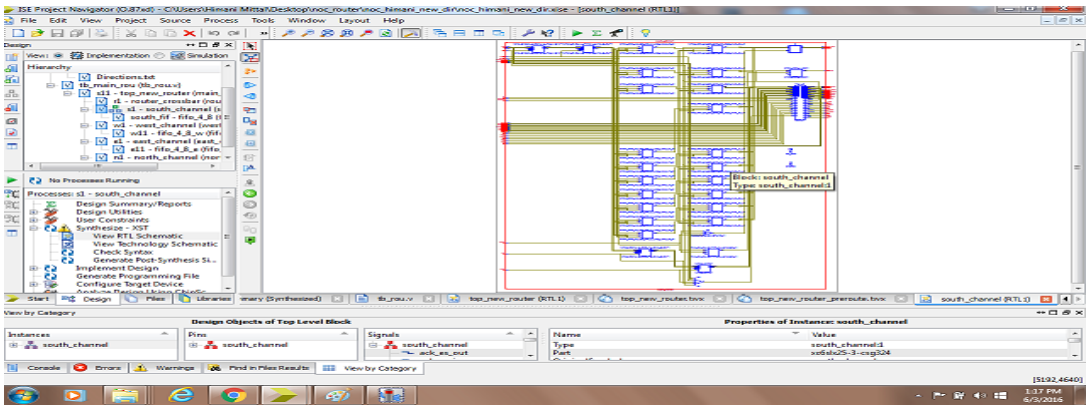


Fig.4.40 RTL view of south channel of EDRRHT

Synthesis of south channel of reconfigurable router EDRRHT generated pin diagram shown in Fig. 4.41

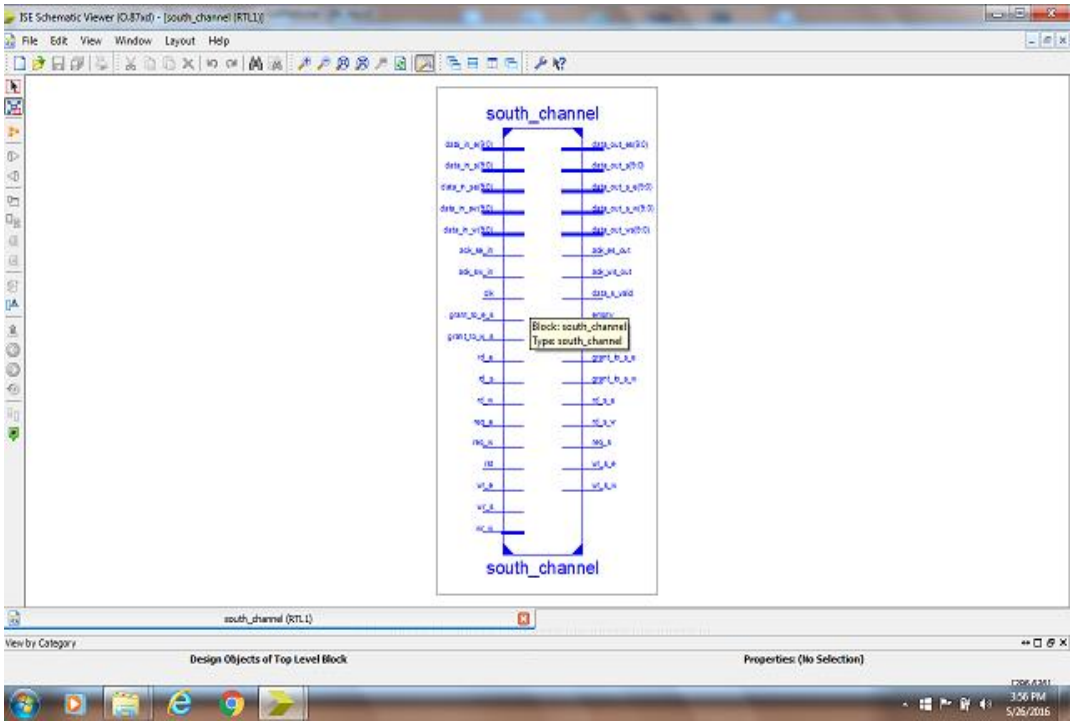


Fig.4.41 Pin diagram of south channel of EDRRHT

By synthesis, pin diagram of main module of EDRRHT is generated and shown in Fig. 4.42

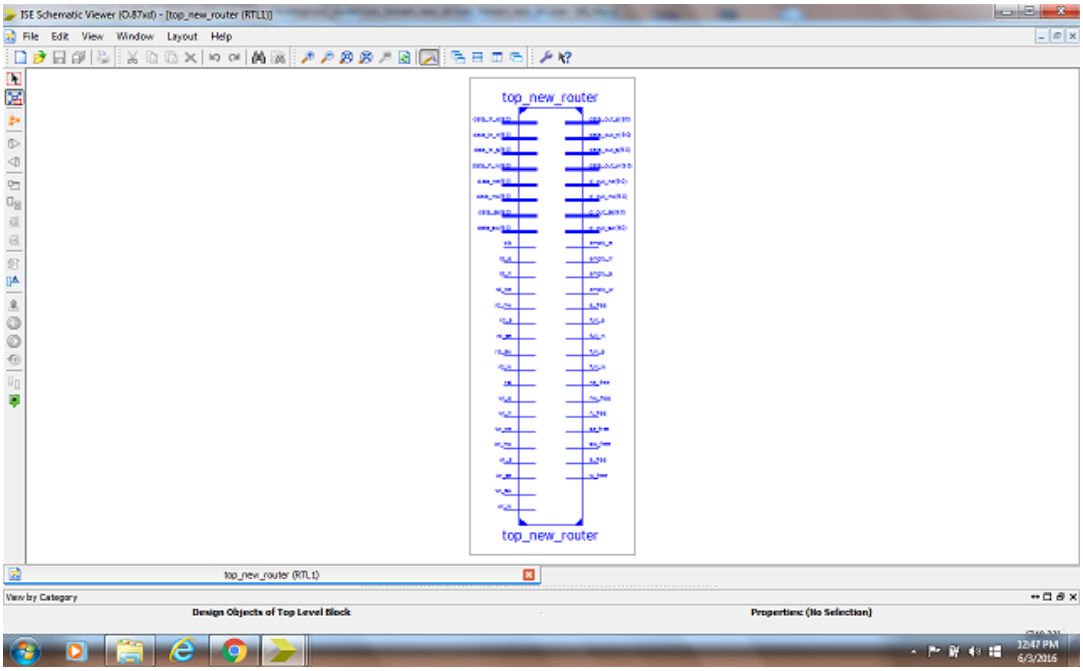


Fig.4.42 Pin diagram of top module of EDRRHT

Fig. 4.43 gives the area estimated for the design of reconfigurable router EDRRHT. It is seen that compared to other routers it takes more area as four more directions are added.

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	549	30064	1%
Number of Slice LUTs	1198	15032	7%
Number of fully used LUT-FF pairs	343	1404	24%
Number of bonded IOBs	194	226	85%
Number of BUFG/BUFGCTRLs	1	16	6%

Fig.4.43 Area of EDRRHT

Xilinx xpower analyzer estimates the total power dissipation of 0.045 mW of EDRRHT in Fig.4.44 .Power dissipation is more because of the addition of four more semi buffers.

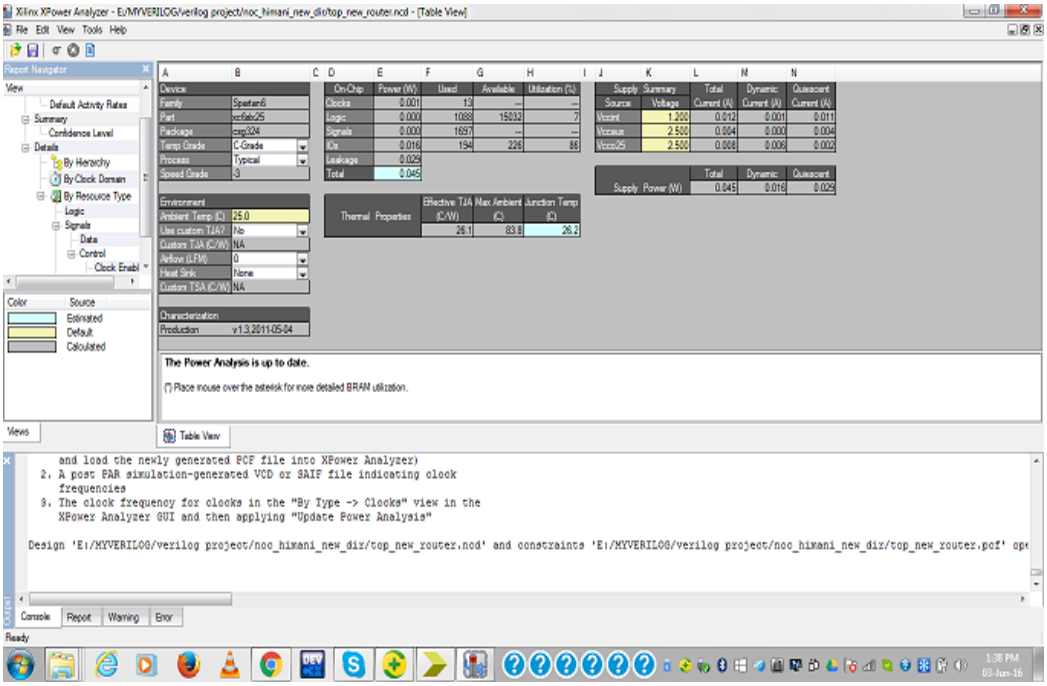


Fig.4.44 Power dissipation of EDRRHT

To validate the proposed approach, we investigated several router architectures. We have implemented buffer less concept by adding four more directions. As we illustrated earlier, throughput is increased as we follow rectilinear path, this can be verified by our simulation results. The Proposed architecture, the combination of a buffer less router along with buffered one introduces an efficient communication and storage mechanism which will shoot up throughput in NoC. This mechanism makes it possible to hide the heterogeneous nature of different reconfigurable engines from the end user. Considering that estimated area is about 85 mm^2 (Slice number 549) and the average power consumption is in the range of 0.045 mW.

With the applications simulated in this work, we confirmed that the original homogeneous NoC presents a large under utilization of the router, since not all of its channels are used. In such cases, the extra buffer words on channels not used in the original router would be unnecessarily consuming power. Besides, for the same performance results, the eight directional reconfigurable routers present a great reduction in power dissipation, reaching up to 35% of power reduction. The larger the link size, the larger the power savings allowed by the reconfigurable router, since in this case the impact of the extra circuits required to allow reconfiguration are

amortized. With respect to area results, the proposed reconfigurable router presents gain due to addition of four more directions when compared with the original router. Besides, another advantage in the use of the NoC with the eight directional reconfigurable router instead of the homogeneous router is that one can dynamically change the buffer depth to each channel, in accordance with the necessity of the application.

Thus, we can conclude that the obtained results emphasize the fact that the proposed NoC router does not degrade the system performance, and can save power.

4.6 Design of SRRODFB:

From the simulation results of EDRRHT it is realised that, it does not handle spotting of the faulty blocks of NoC. Also it does not recognize and differentiate between permanent and transient errors and how to handle them. Major drawback is that it lacks suitable algorithm that helps in online detection of faults while preserving its throughput, data packet latency and network load.

Taking into considerations above drawbacks we proposed SRRODFB .In proposed Smart reconfigurable router design for online detection of faulty blocks, online detection of faulty blocks is possible. It can distinguish between permanent and transient errors and is able to accurately localize the position of the faulty blocks (data bus, input port, output port) in the NoCs, while preserving the throughput, the network load, and the data packet latency. FSM, Routing Error detection, Logic, centralized Journal of data Packet - Removed, as our loop back module itself does the same functionality. Buffers are removed, instead of buffers; FIFO is used to achieve better performance.

Hence in proposed optimized architecture shown in Fig. 4.45 is having comparatively Low Area, Low Power; and due to the FIFO logic used in the routing logic, which makes router to achieve higher performance than the existing RKT switches well as low overhead.

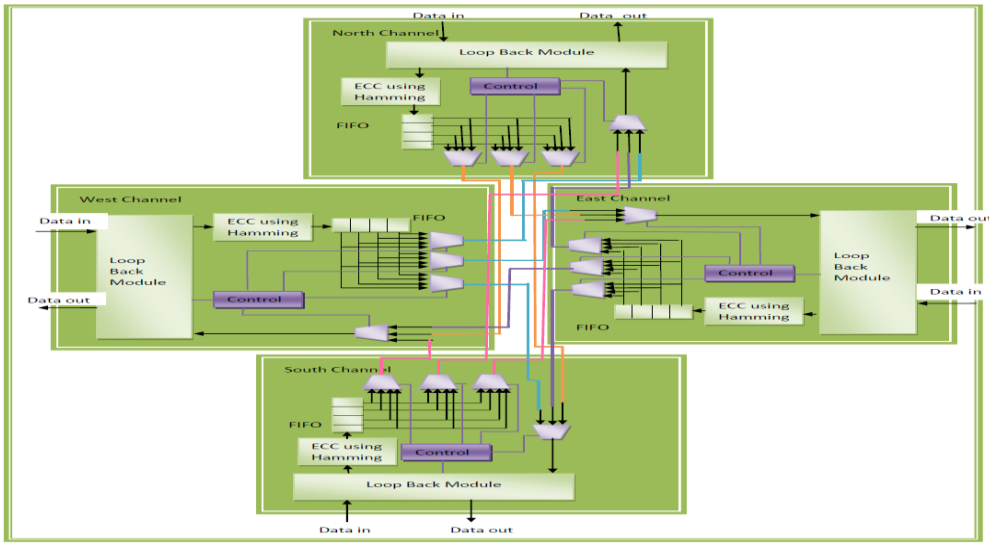


Fig.4.45 Block diagram of SRRODFB.

4.6.1 Simulation Results:

Fig.4.46 shows the simulations of proposed router, It is seen that all then, N, S, W, E modules send a data packet simultaneously and similarly receive the data packets simultaneously. It is also observed that all the data packets do not make any loopback and bypass as there is no fault present.

Here it is showing that the data is input in four directions and after routing data is out from four directions N, S, E, and W according to XY algorithm. Also input is given and deliberately error is introduced which is corrected by ECC hamming code.

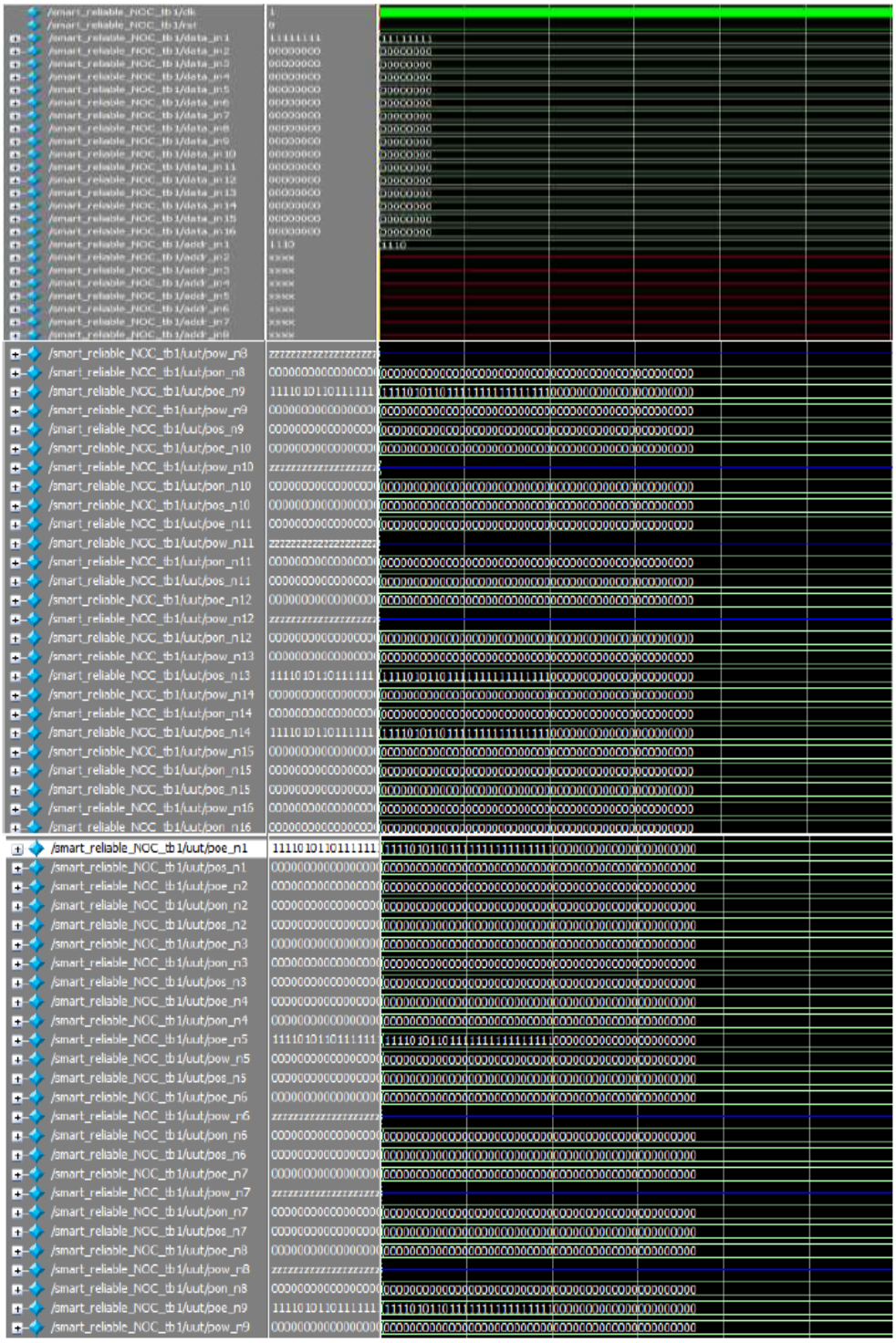


Fig.4.46 Simulation results of SRRODFB

4.6.2 Synthesis Results:

Top router pin diagram is generated by synthesis in Xilinx and is shown in Fig. 4.47.

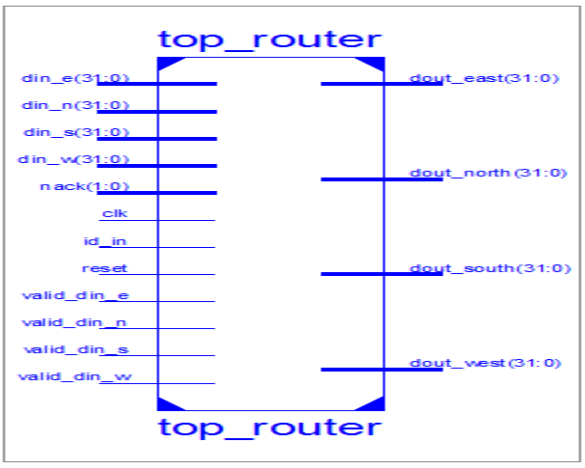


Fig.4.47 Pin diagram of SRRODFB

Fig. 4.48 shows the RTL schematic of main module of reconfigurable router SRRODFB. In Fig 4.48 loop back module, hamming correction block and FIFO are separately shown.

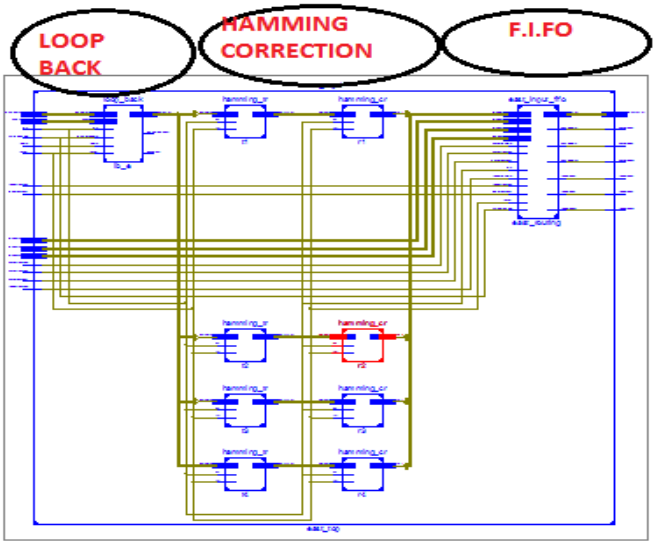


Fig. 4.48 RTL view of top module of SRRODFB

Fig. 4.49 shows the RTL view of SRRODFB, showing separately north top, south top, east top, west top.

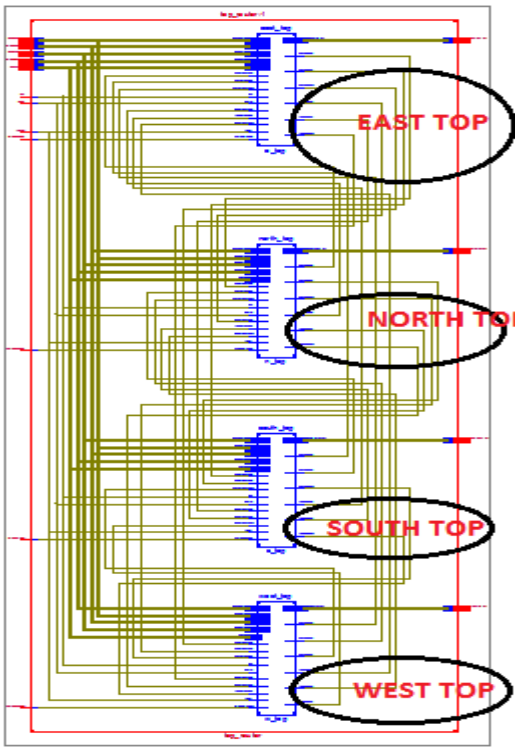


Fig.4.49 RTL view of SRRODFB

Detailed view of loopback is shown in Fig. 4.50.

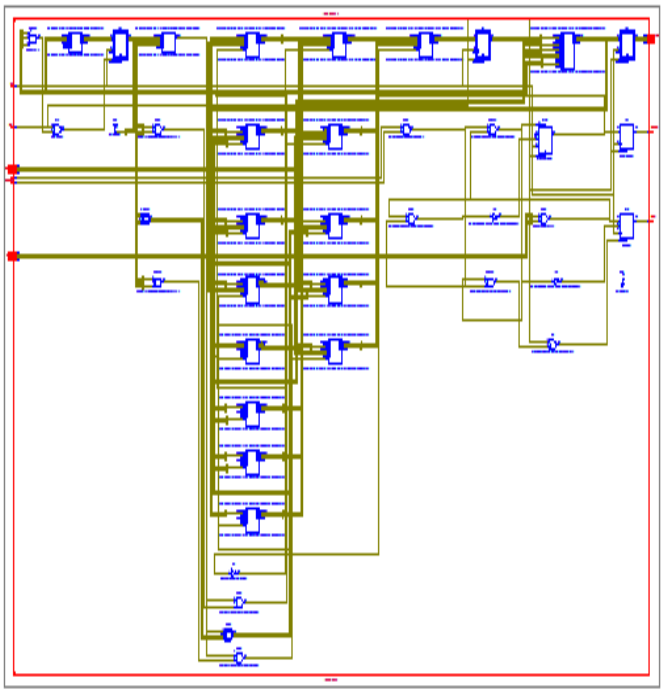


Fig.4.50 RTL view of loopback of east top of SRRODFB

Fig. 4.51 shows the detailed RTL view of hamming coder which is one of the important parts of SRRODFB.

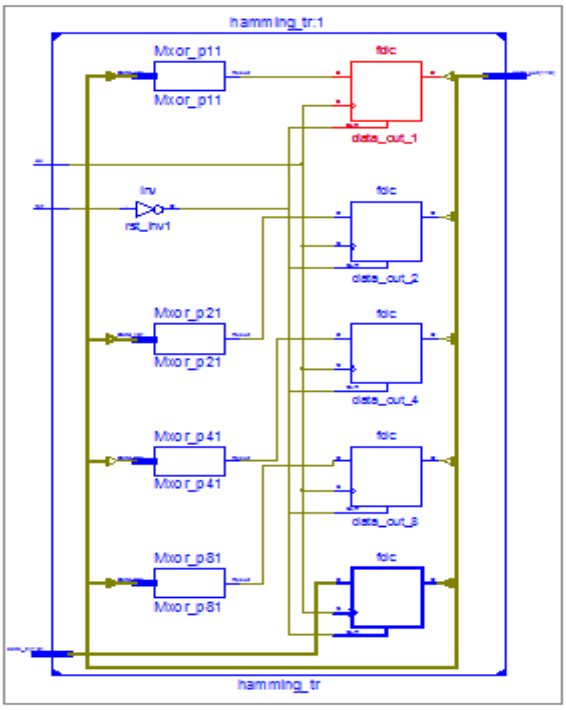


Fig. 4.51 RTL view of hamming coder of east top of SRRODFB

Fig. 4.52 shows the detailed RTL view of hamming decoder which is one of the important parts of SRRODFB.

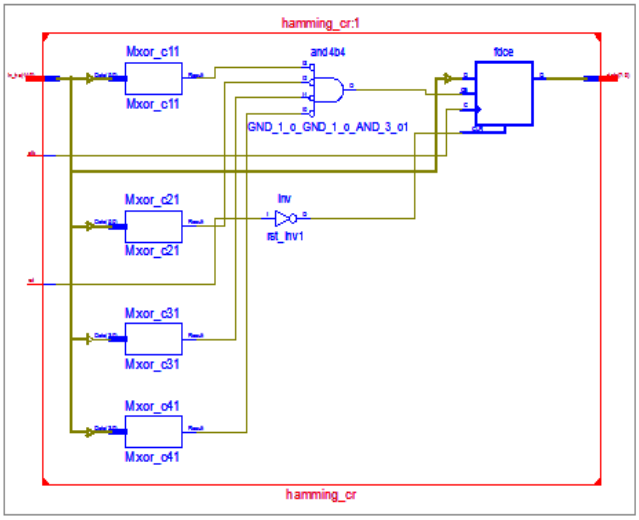


Fig.4.52 RTL view of hamming decoder of east top of SRRODFB

RTL view of east top module is shown in Fig. 4.53. All modules are similar in architecture as observed from their RTL schematic.

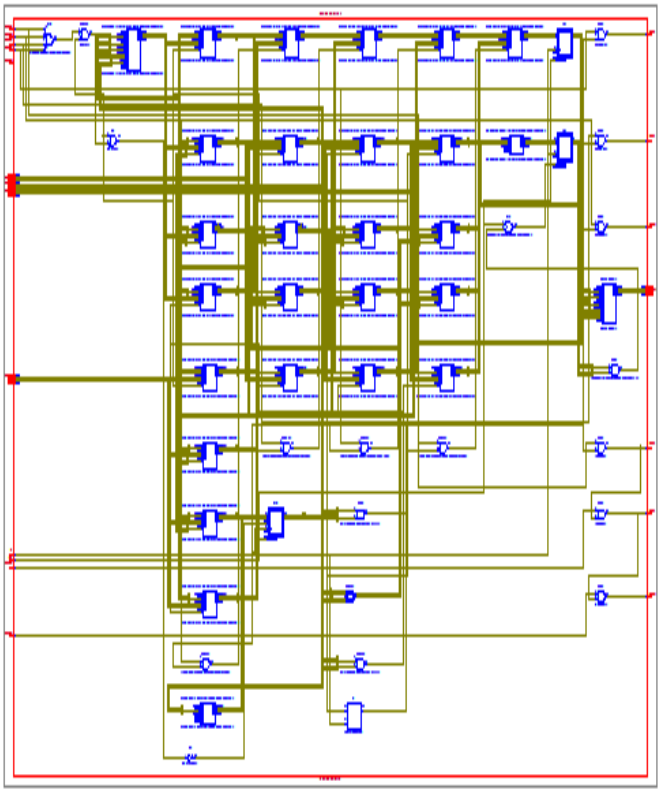


Fig .4.53 RTL view of east top of SRRODFB

Pin diagram of top module is generated by synthesis in Xilinx tool and is shown in Fig.4.54.

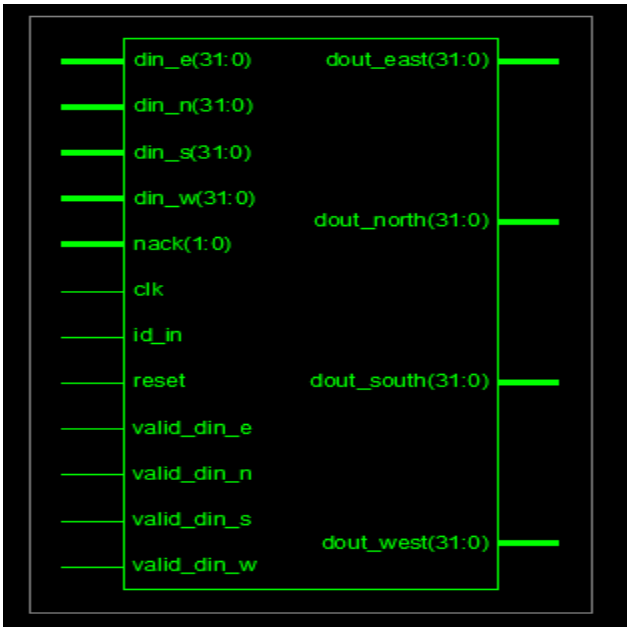


Fig.4.54 Pin diagram of top module of SRRODFB

Detailed RTL view of east top module is generated by synthesis and is shown in Fig.4.55.

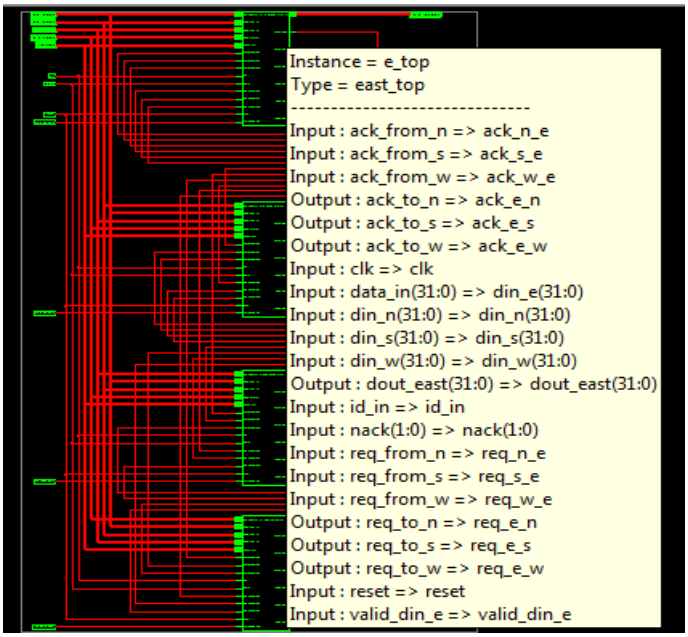


Fig.4.55 Detailed view of east top of SRRODFB

Detailed RTL view of north top module is generated by synthesis and is shown in Fig.4.56.

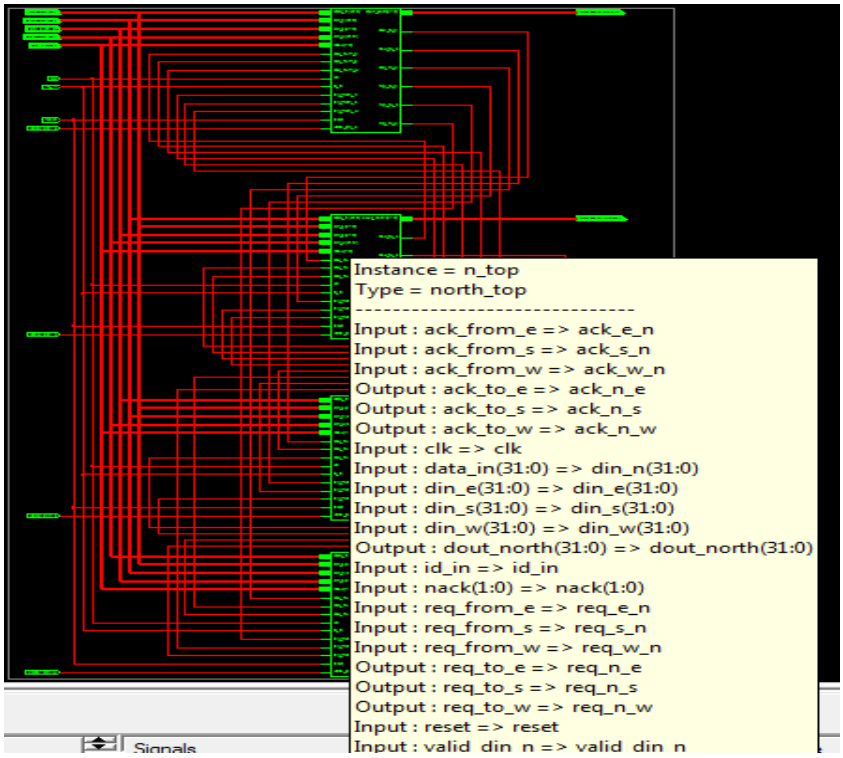


Fig .4.56 Detailed view of north top of SRRODFB

Detailed RTL view of south top module is generated by synthesis and is shown in Fig.4.57.

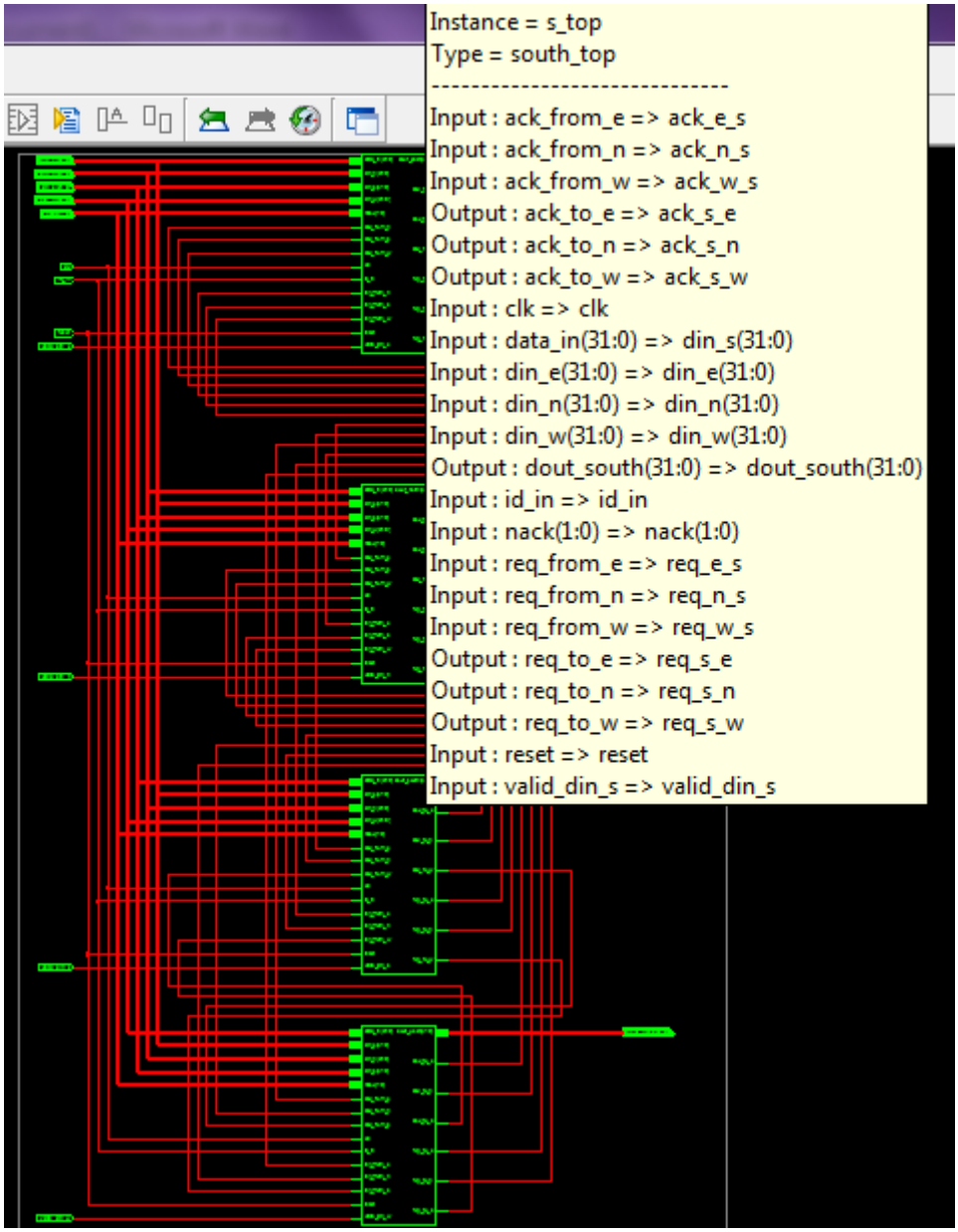


Fig.4.57 Detailed view of south top of SRRODFB

Detailed RTL view of west top module is generated by synthesis and is shown in Fig.4.58.

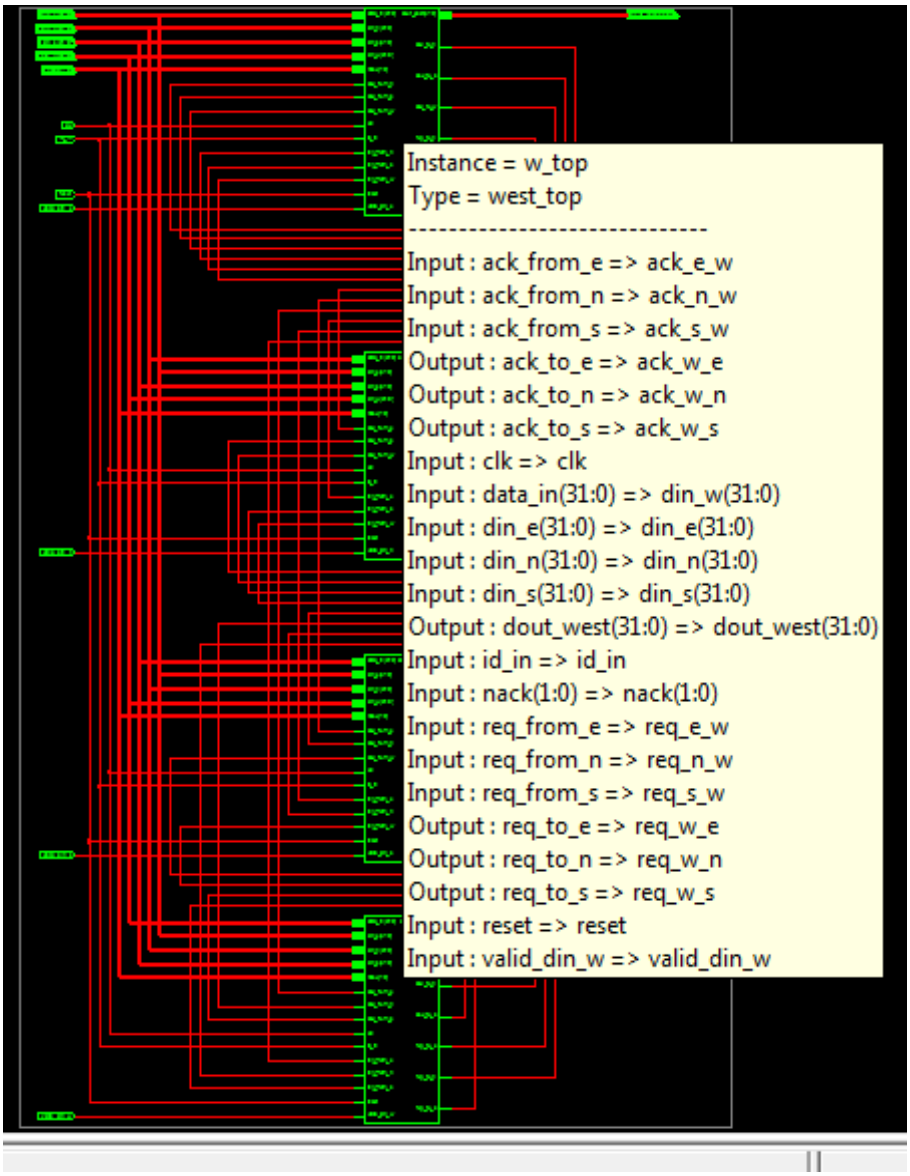


Fig.4.58 Detailed view of west top of SRRODFB

Detailed diagram of the internal architecture of east module of reconfigurable router SRRODFB is shown in Fig.4.59. Following Figures show all the major components required for east module i.e. loopback, hamming coder, hamming decoder, FIFO buffer.

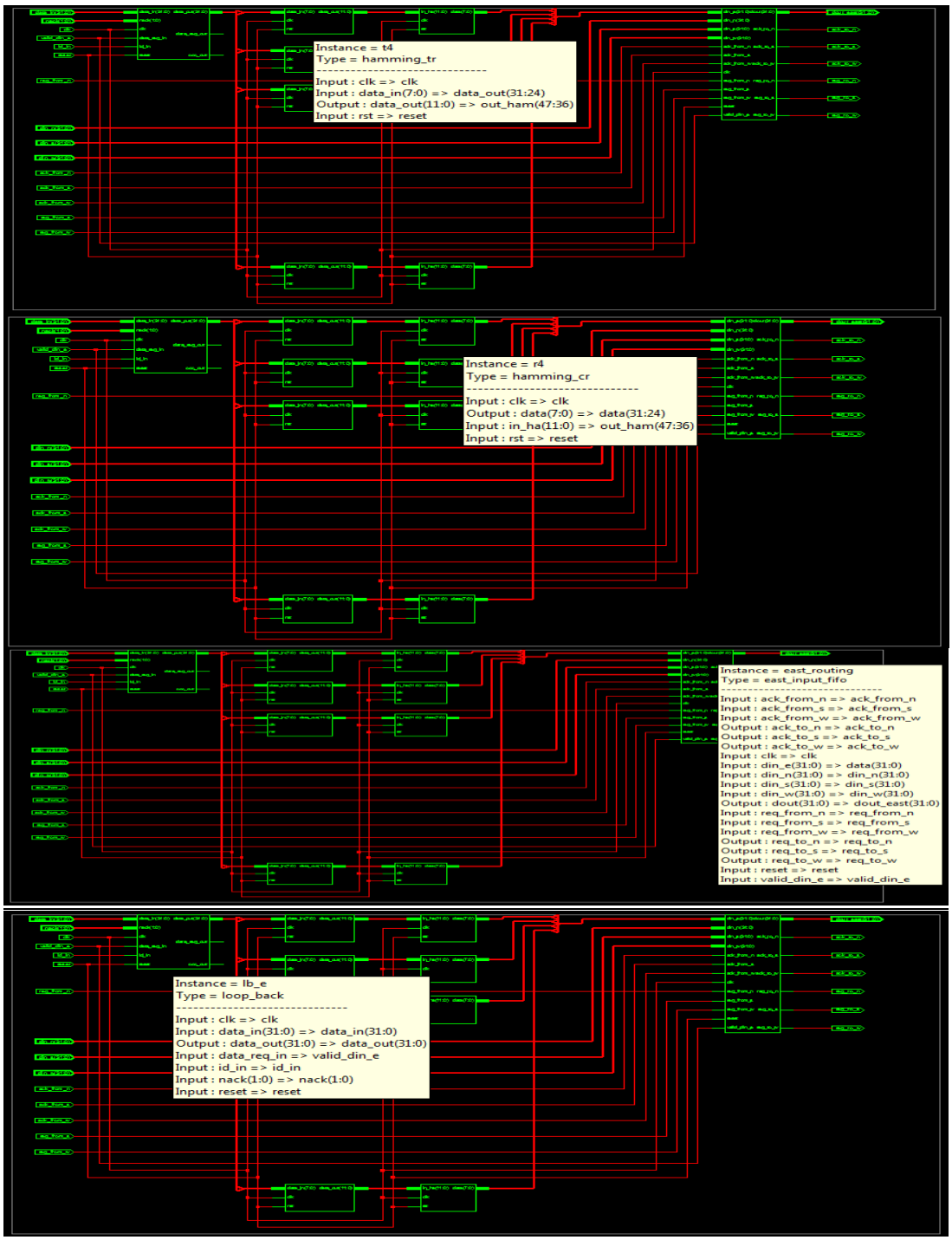


Fig.4.59 East top internal architecture of SRRODFB

Detailed diagram of internal architecture of north module of reconfigurable router SRRODFB is shown in Fig.4.60. Following Figures show all the major components required for north module i.e. loopback, hamming coder, hamming decoder, FIFO buffer.

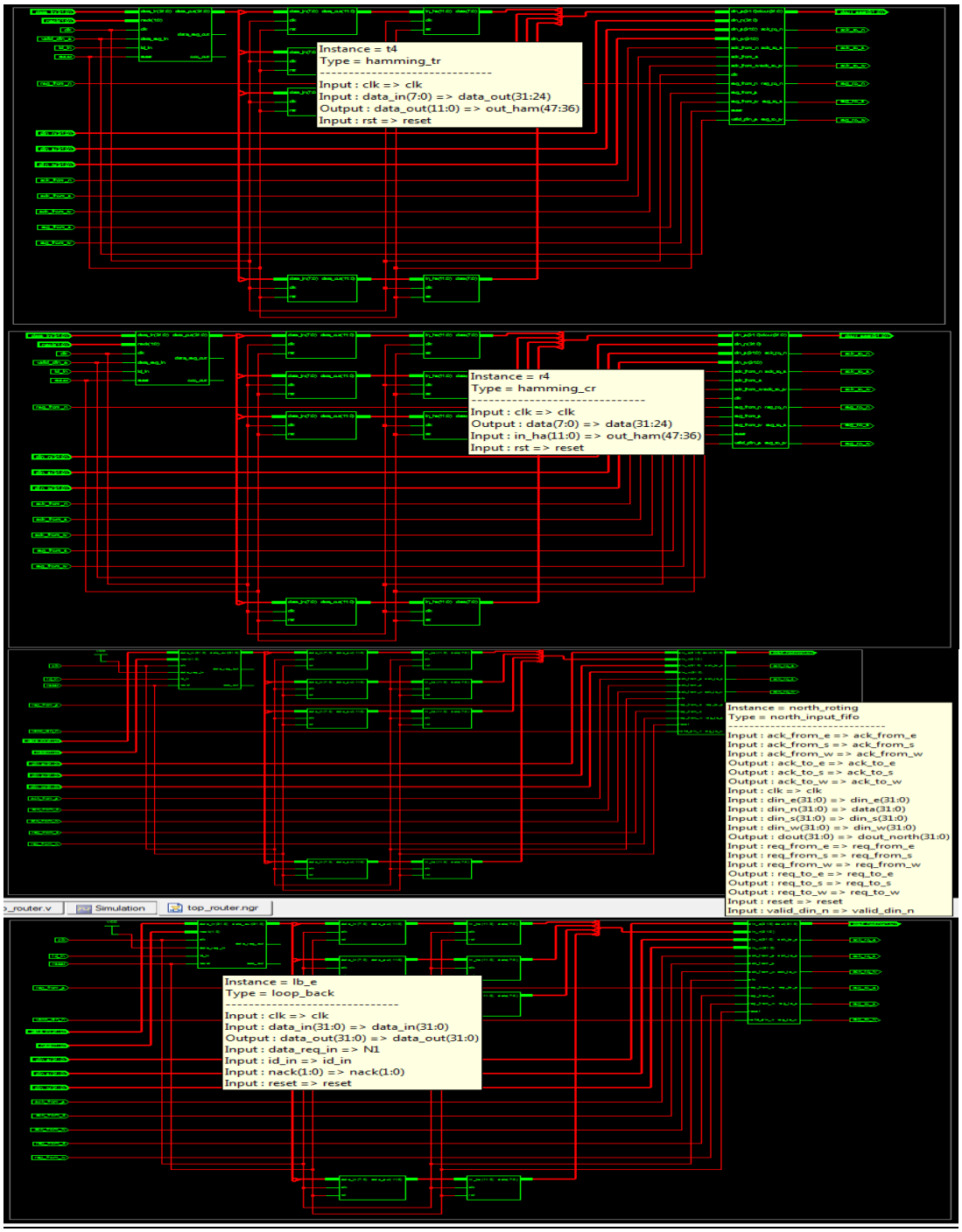


Fig 4.60 North top internal architecture of SRRODFB

Detailed diagram of internal architecture of south module of reconfigurable router SRRODFB is shown in Fig.4.61. Following Figures show all the major components required for south module i.e. loopback, hamming coder, hamming decoder, FIFO buffer.

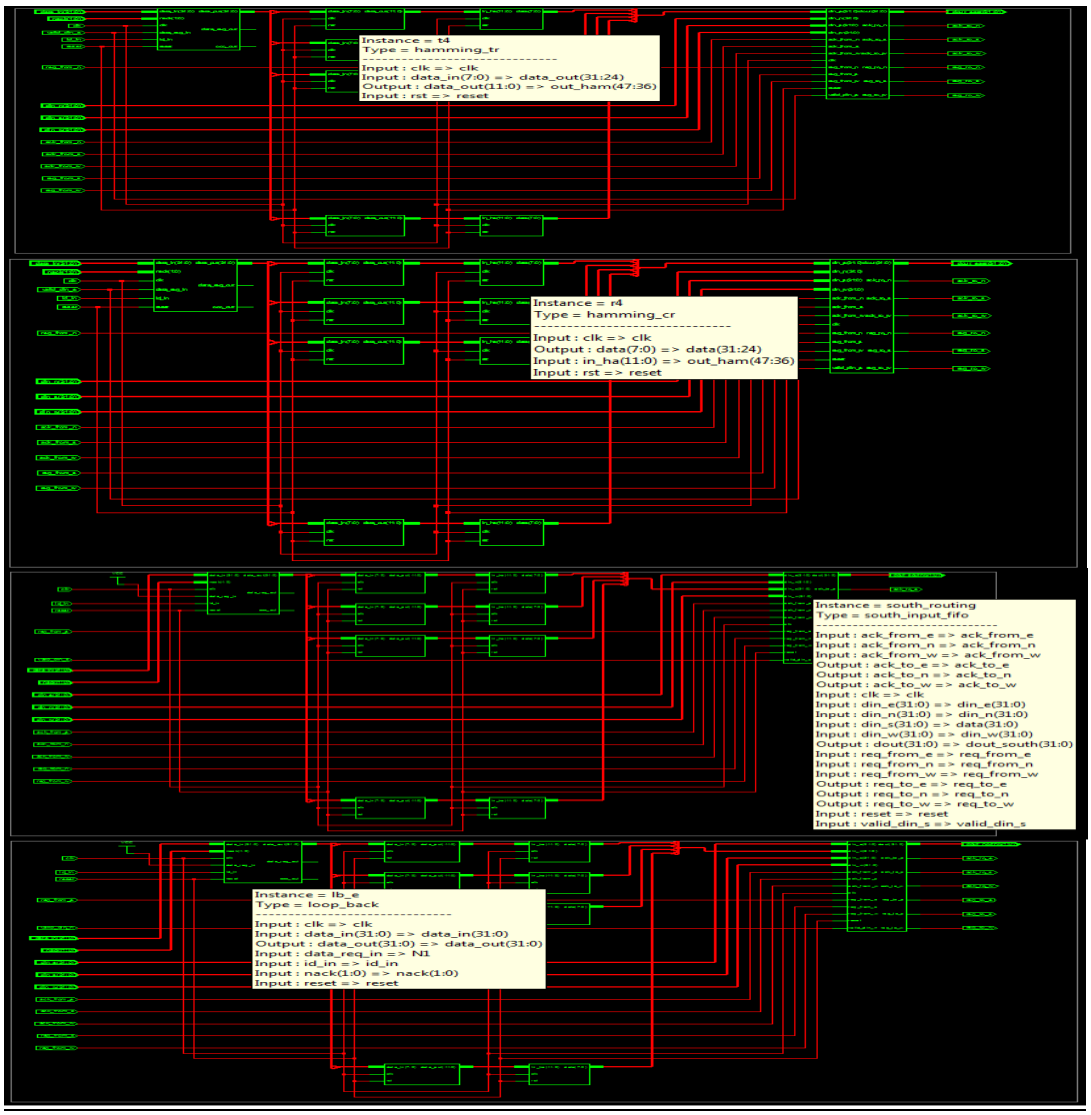


Fig.4.61South top internal architecture of SRRODFB

Detailed diagram of internal architecture of west module of reconfigurable router SRRODFB is shown in Fig.4.59. Following Figures show all the major components required for west module i.e. loopback, hamming coder, hamming decoder, FIFO buffer.

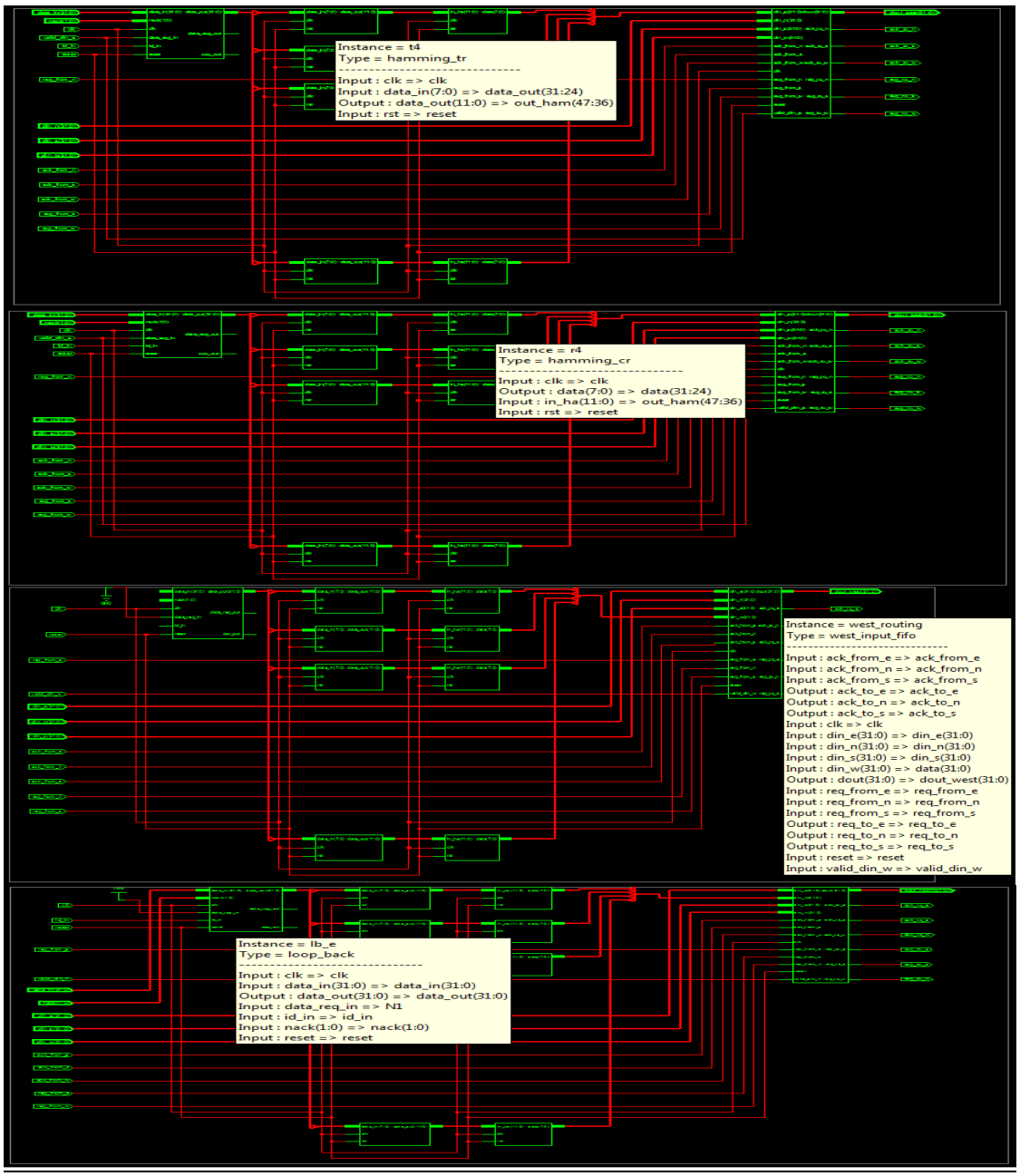


Fig.4.62 West top internal architecture of SRRODFB

Synthesis report generated by Xilinx tool shows the delay of reconfigurable router SRRODFB as shown in Fig.4.63.

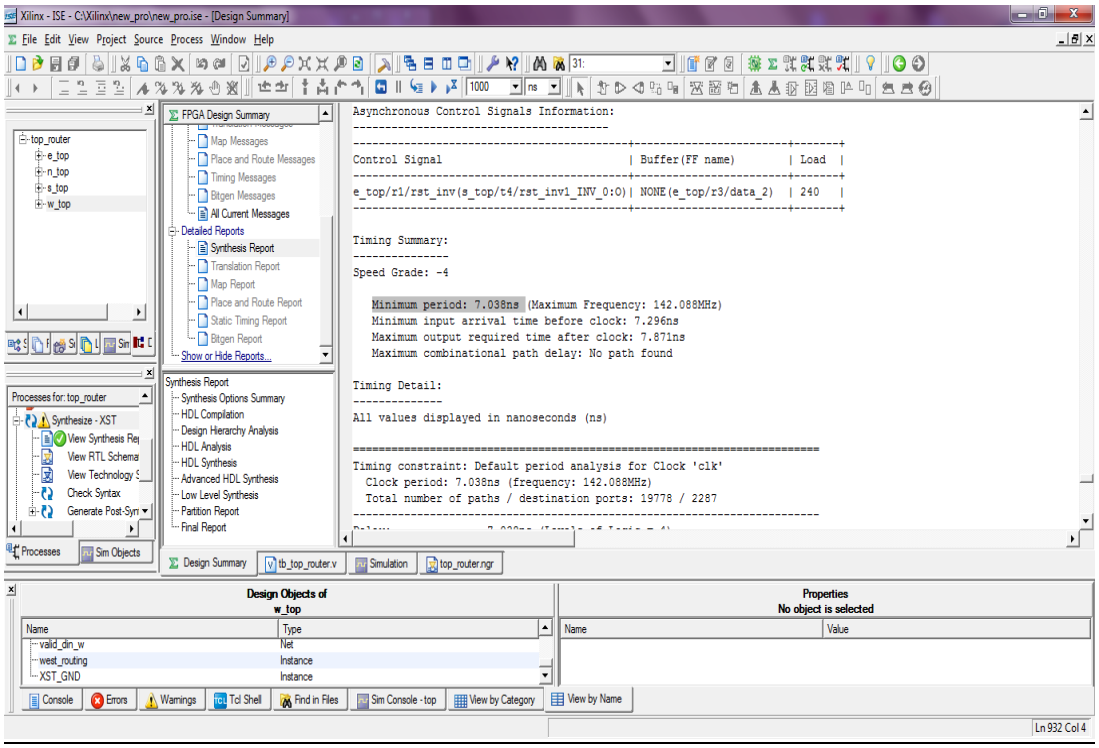


Fig.4.63 Delay of SRRODFB

Synthesis report generated by Xilinx tool shows the frequency of reconfigurable router SRRODFB as shown in Fig.4.64.

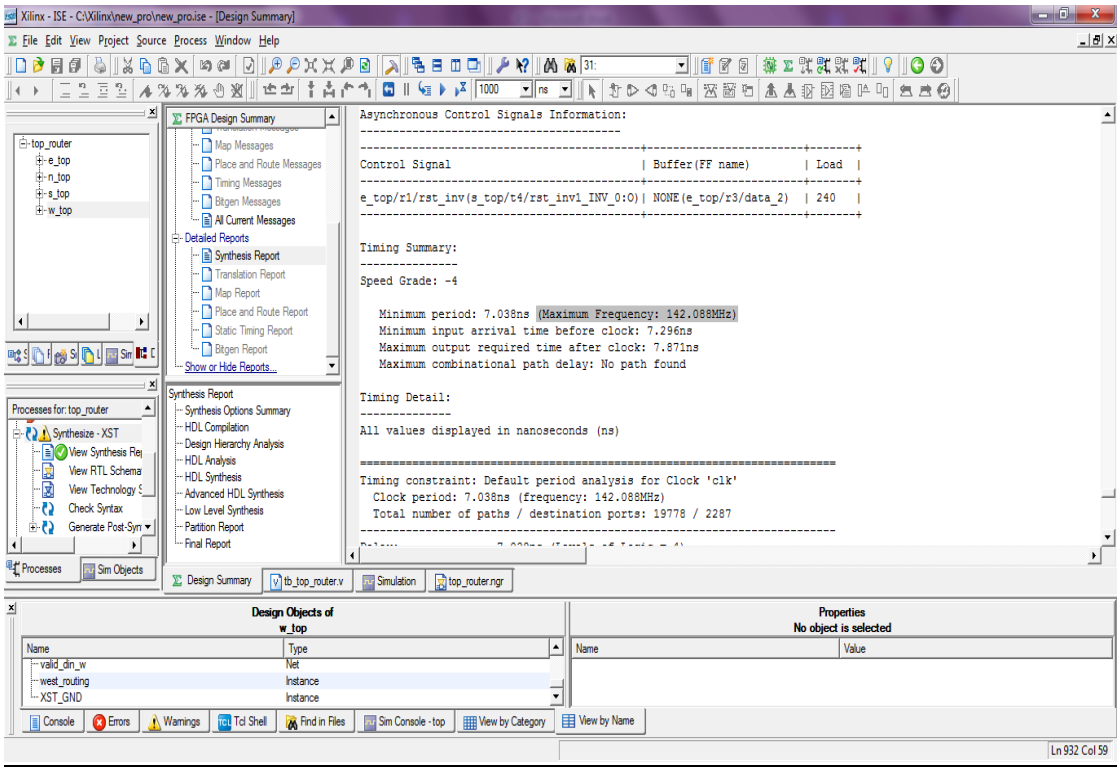


Fig.4.64 Frequency of SRRODFB

Synthesis report generated by Xilinx tool shows the area of reconfigurable router SRRODFB as shown in Fig.4.65.

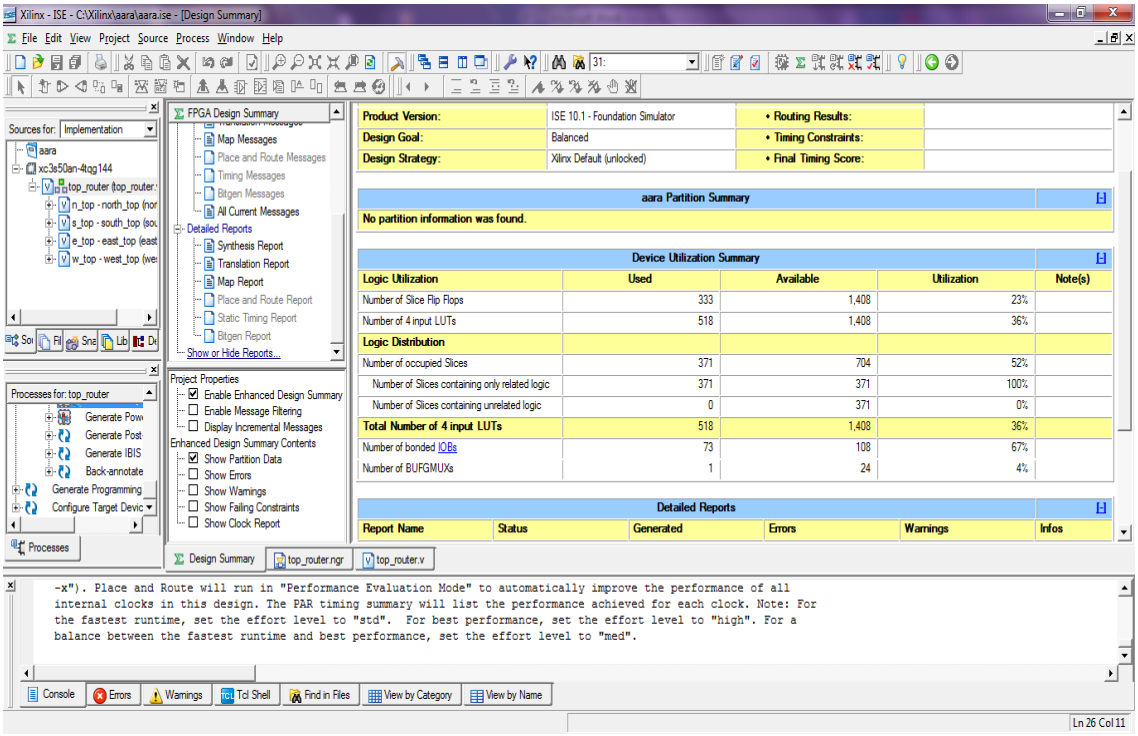


Fig. 4.65 Area of SRRODFB

Synthesis report generated by Xilinx tool shows the power of reconfigurable router SRRODFB as shown in Fig.4.66.

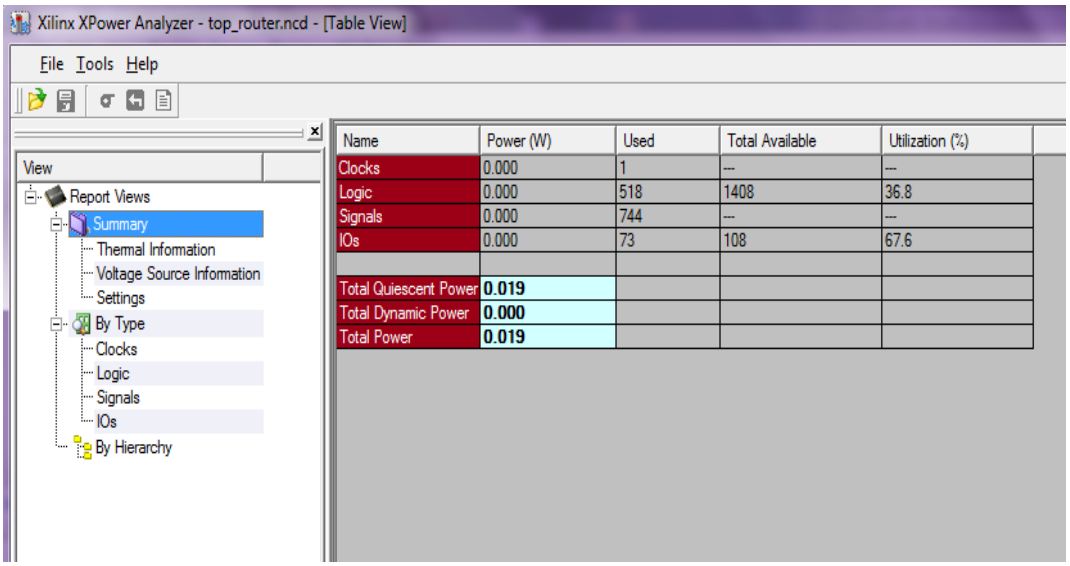


Fig.4.66 Power of SRRODFB

Validation of proposed concept:

The proposed architecture of smart reconfigurable router is highly reliable when compared with ordinary NoC due to the addition of error correcting code by hamming distance in the design and it avoids the dead lock and live lock problem. Routing algorithm based on XY logic allows the bypassing of unavailable components. The presented technique can also distinguish between permanent and transient errors and localize error sources more accurately than switch-to-switch and code-disjoint techniques. It also has the advantage of disconnecting only the faulty part like bus, input port or output port. Further the use of loopback module and arbiter serves the data from loss.

There is significant decrease in no. slice registers, LUT's as compared with previous routers because of elimination of F.S.M, routing logic. Power consumption is also reduced to a great extent to .019mW which is a remarkable result.

4.7 Comparison of four reconfigurable routers RRIE, HRRLPHP, EDRRHT, SRRODFB:

Based on simulation results obtained from the comparison of various parameters of reconfigurable routers like buffer depth, no. of slices, power, delay and frequency Table 4.2 is drawn. It is obvious from Table 4.2 that EDRRHT consumes highest number of slices; dissipate moderate power, while having lowest delay. SRRODFB on the other hand dissipitates minimum power while having delay almost equal to EDRRHT and moderate number of slices. The results obtained are further elaborated in this section through various Fig.

Parameters	Buffer depth	No. of Slices	Power dissipation (W)	Delay (ns)	Frequency (MHz)
RRIE	Varying	28	0.085	15.0	100
HRRLPHP	Varying	82	0.021	8.3	120
EDRRHT	Semi Buffer less	549	0.045	7.1	140
SRRODFB	4 bits	333	0.019	7.03	142

Table 4.2 Comparison of various parameters of RRIE, HRRLPHP, EDRRHT, SRRODFB

Fig. 4.68 shows column chart graph of the power comparison of four proposed reconfigurable routers. It is seen that as we move from RRIE to next modified reconfigurable router power dissipation reduces though it increases for EDRHT as four more directions are added.

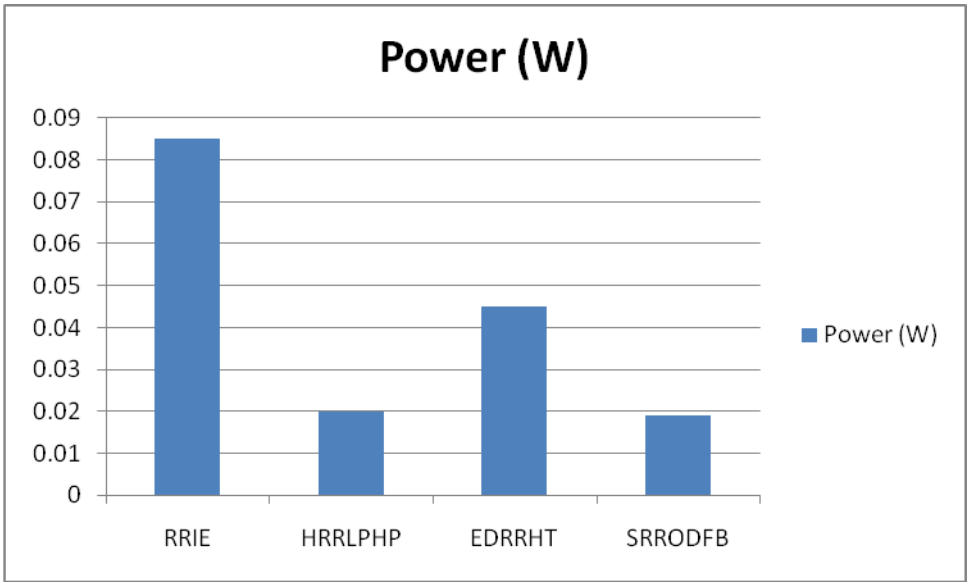


Fig.4.68 Power dissipation comparison of RRIE, HRRLPHP, EDRRHT, SRRODFB

Fig. 4.69 shows column chart of delay comparison of four reconfigurable routers. As we head to next router delay decreases and it is least in EDRRHT because data travels in four more directions thus reducing latency.

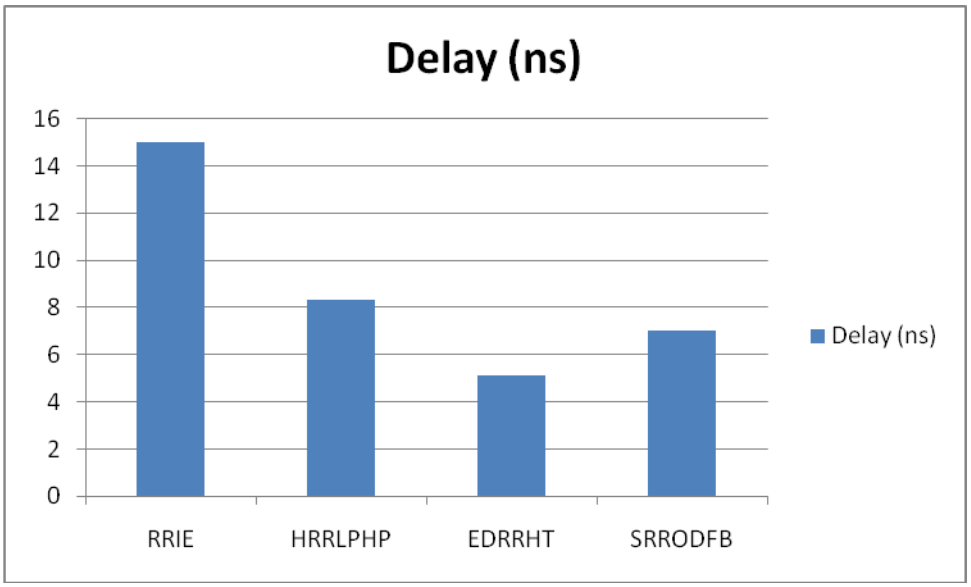


Fig.4.69 Delay comparison of RRIE, HRRLPHP, EDRRHT, SRRODFB

Fig. 4.70 shows column chart graph showing frequency comparison of four reconfigurable routers. Frequency increases as we move from RRIE to SRRODFB.

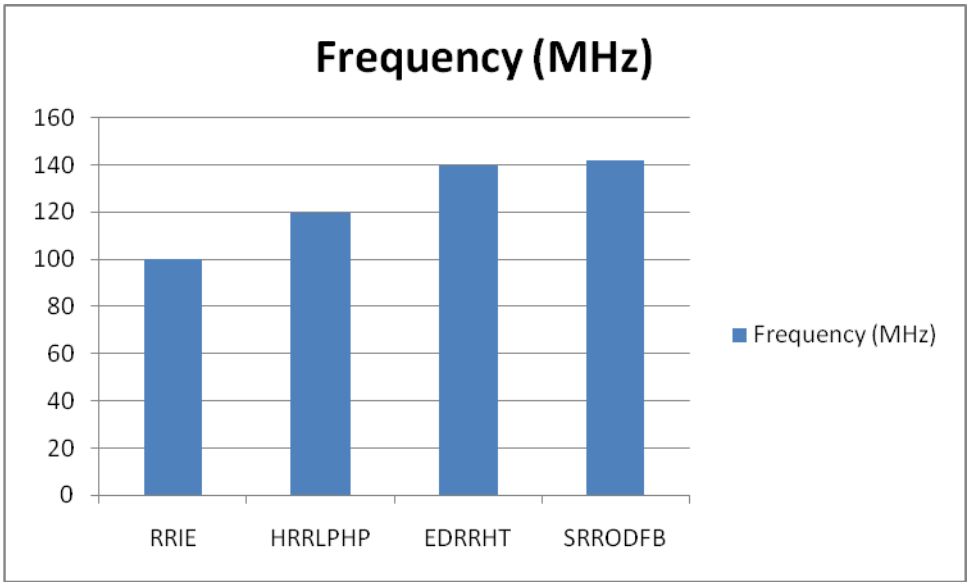


Fig.4.70 Frequency comparison of RRIE, HRRLPHP, EDRRHT, SRRODFB

Fig. 4.71 shows column chart graph showing no. of slices comparison of four reconfigurable routers. Frequency increases as we move from RRIE to SRRODFB.

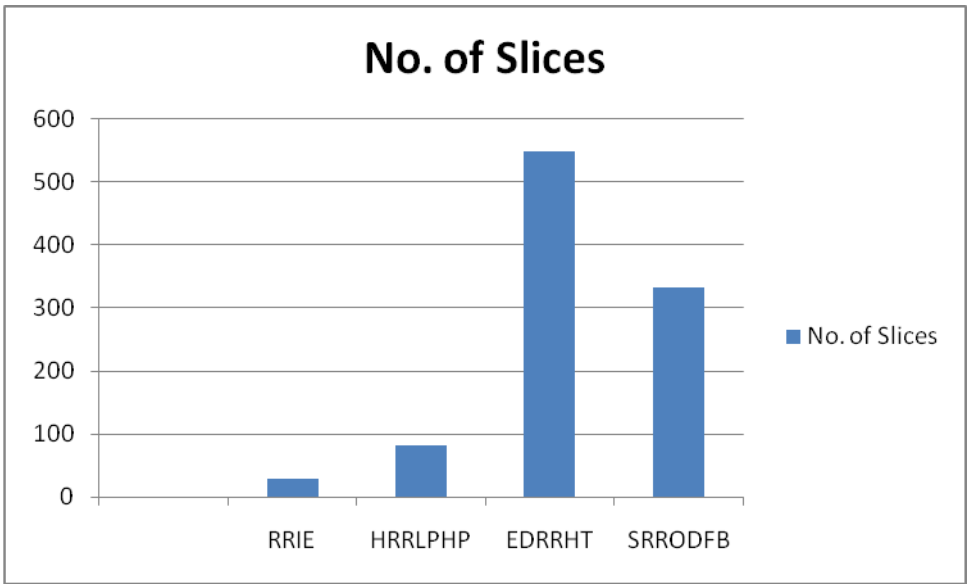


Fig.4.71 No. of slices comparison of RRIE, HRRLPHP, EDRRHT, SRRODFB

Fig. 4.72 shows graph chart showing performance analysis comparison of four reconfigurable routers. Frequency increases as we move from RRIE to SRRODFB.

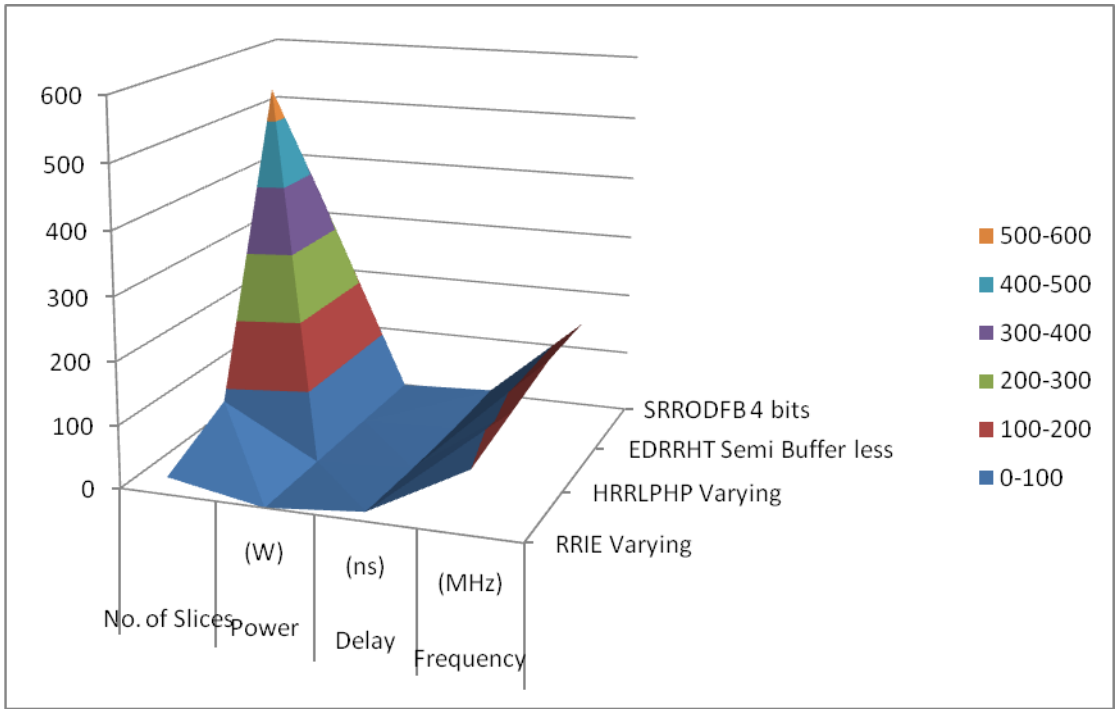


Fig.4.72 Performance Analysis of RRIE, HRRLPHP, EDRRHT, SRRODFB

CHAPTER FIVE

LINK DESIGN FOR NoC USED IN RECONFIGURABLE SYSTEM

As per the conclusion of chapter 4, eight directional reconfigurable router designed for high throughput is concluded as the best router for the reconfigurable systems. As the routers following a certain topology in NoCs transmit data/information to other routers/IPs through links, the design of power efficient links is a major research challenge. This chapter presents various links designed for NoCs used in the reconfigurable systems along with their performance comparisons.

Also, links or Interconnects consume the paramount share of dynamic power in circuits. Researchers have shown that interconnect links consume up to 60% of the dynamic power in NoC. In this chapter, it has been shown that by using three different ways i.e. Mux gating, encoding and decoding and multi coding technique power consumption can be reduced in NoC links.

5.1 Design Methodology:

For the design purpose hardware description language VERILOG is used**.** For the simulation purpose MODELSIM Edition10.3 TOOL is used and test bench is written in Verilog by which working of Links Designs is tested and simulated in Model sim tool.

For the Synthesis purpose Xilinx ISE Design Suite 13.4 Tool is used. The Xilinx design tool gives a low level design. It shows the RTL VIEW, total design summary of the circuit, and total power consumption by the circuit.

Three different types of links have been designed for the reconfigurable systems in the present work. These are summarized as

1. HSLPL (High speed low power link design for reconfigurable systems):

In this proposed link HSLPL, MUX gating technique is used to lower power dissipation and latency.

2. LMSTRDP (Link to minimize switching transitions for reducing dynamic power for reconfigurable system):

In this proposed approach LMSTRDP, we have reduced no. of transitions with the help of encoder and decoder. The encoder will encoded the data before placing it onto data bus in such a way that the number of transitions is reduced. Hence it saves the switching power dissipation on the other side decoder decodes the data. These encoders and decoders can be used for off chip data transfer also.

3. LMRTP (Link by multi encoding to reduce transition activities in reconfigurable system):

In the third approach LMRTP, multi coding technique is used to reduce transition activity in logic circuits.

This chapter provides the designing of the links for reconfigurable system routers along with their performance comparison:

5.2 DESIGN OF HSLPL:

In the proposed link design of high speed low power link for reconfigurable system focus is on adjacent links using MUX gating technique where path is reduced from source to destination by introducing diagonal path and the data once received is disconnected from the previous source.

Addition of diagonal path avoids deadlock by removing cyclic dependencies, hence messages can transverse through diagonal path i.e. minimal path, it reduces latency and increases throughput.

The objective of HSLPL is to reduce power dissipation in links and to increase throughput by considerable factor thus reducing latency .

In the proposed link design Fig. 5.1 shows the path without Mux gating where there is no diagonal path and it takes 5x distance to travel from source to destination.

In Fig. 5.2 which shows HSLPL which uses MUX gating, with the introduction of diagonal path distance from source to destination is reduced from 5x to 3y , where 5x is the distance in traditional routers and 2y+x is the distance in proposed one and x is

direct distance from one router to another while y is diagonal distance between two routers.

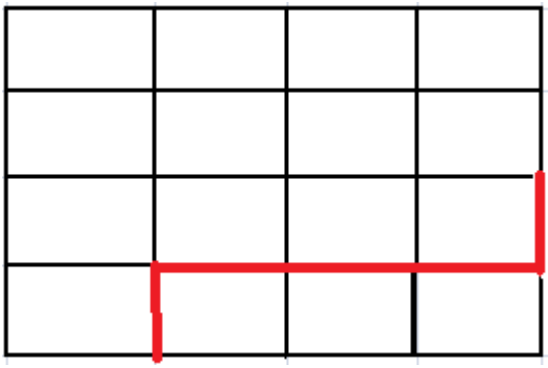


Fig.5.1 Links without MUX gating with path 5x.

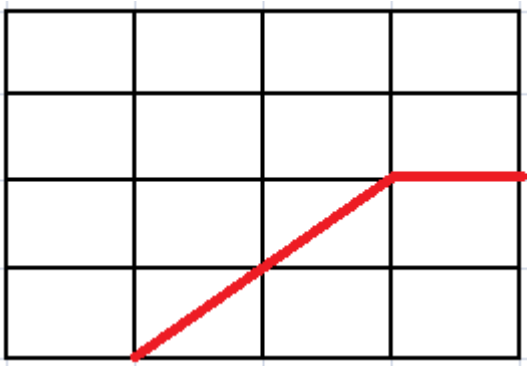


Fig.5.2 Proposed HSLPL with path 2y+x.

5.2.1 Simulation Results :

In Fig.5.3 when data is transmitted through Mux from west router to north router it is seen that once received completely in destination router i.e. north , data is disconnected from source router i.e. west thus justified the proposed concept.

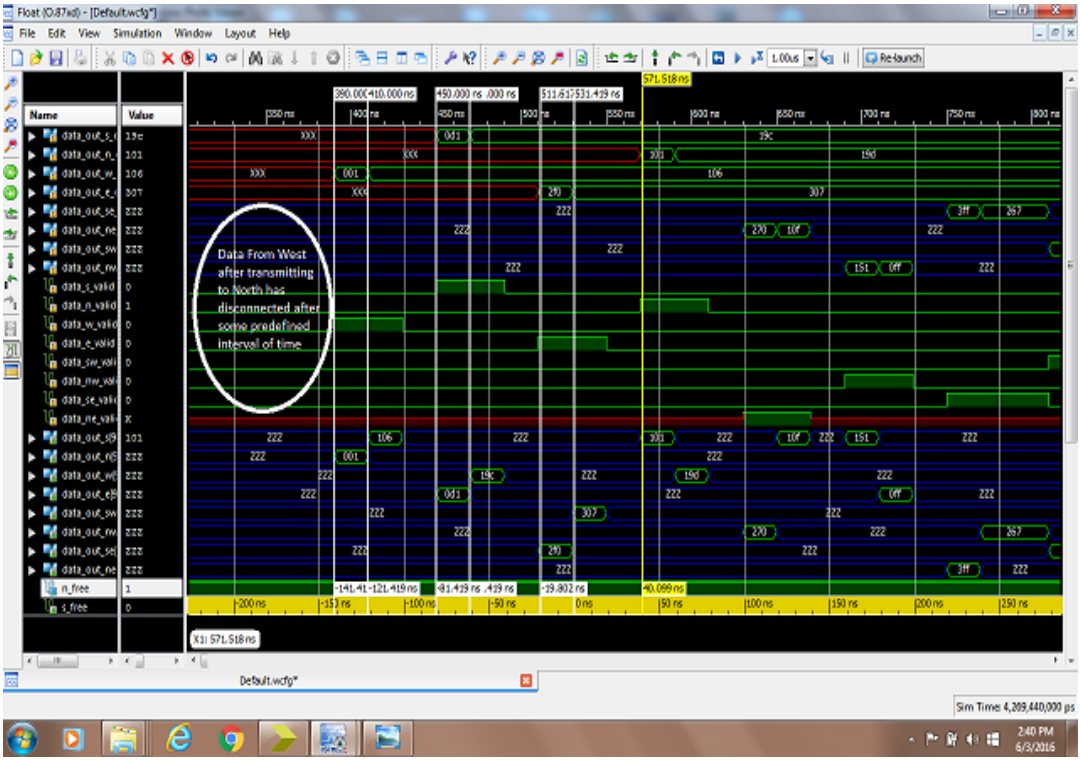


Fig.5.3 Simulation of MUX data transmission.

Fig.5.4 shows the Mux gating from all four directions S, N, W, and E. Data entering from these directions are routed to destined routers through respective Mux and then disconnected from their source routers.

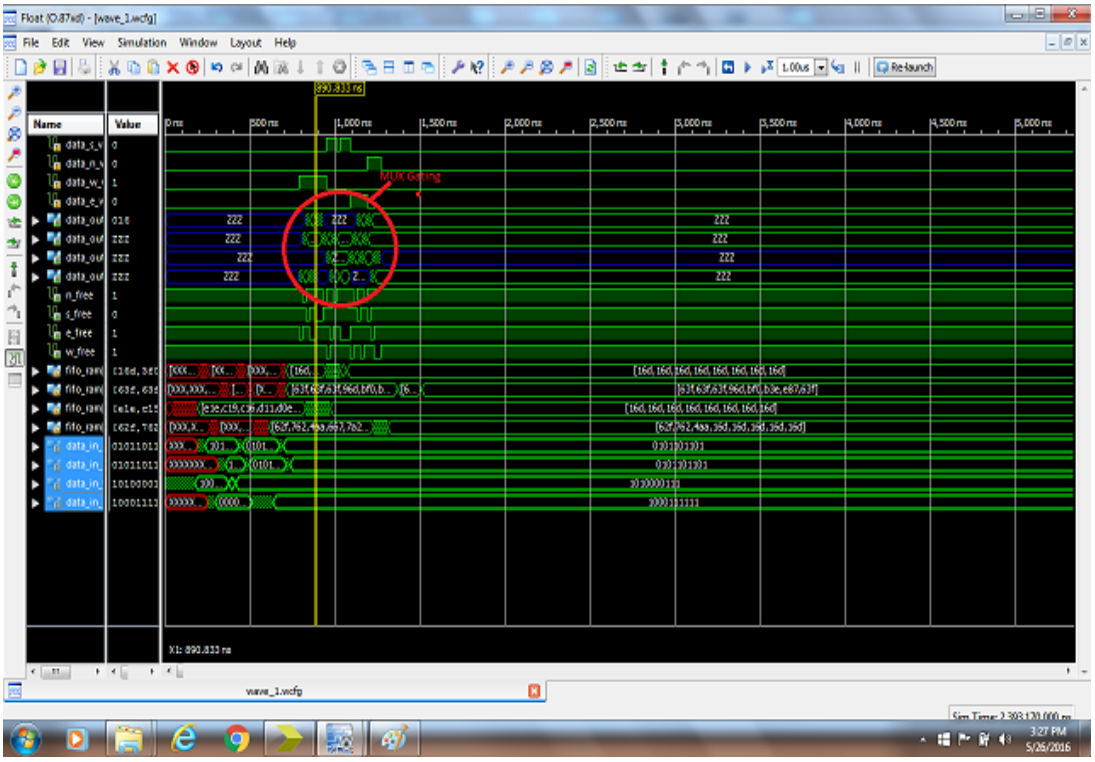


Fig.5.4 Simulation of MUX gating

Power dissipation of HSLPL is shown in Fig. 5.5 where total power dissipation is 0.045 W and dynamic power dissipation is 0.016 W.

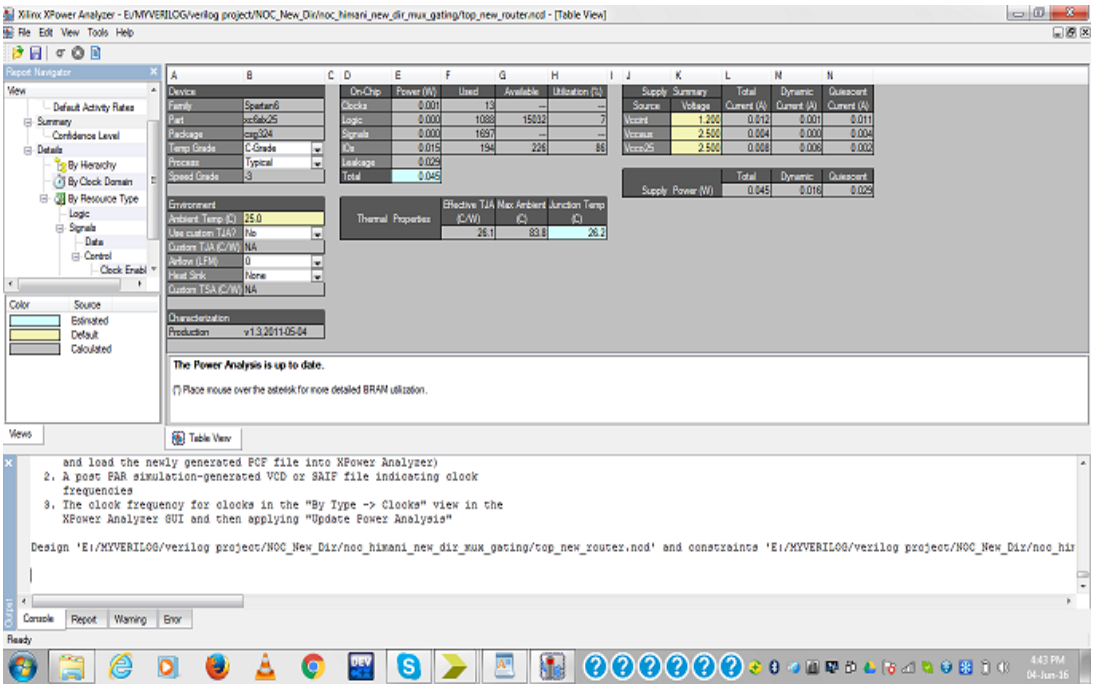


Fig 5.5 Power dissipation of HSLPL

Device utilization summary of HSLPL is given in Fig. 5.6.

Area :

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	574	30064	1%
Number of Slice LUTs	1347	15032	8%
Number of fully used LUT-FF pairs	393	1528	25%
Number of bonded IOBs	170	226	75%
Number of BUFG/BUFGCTRLs	1	16	6%

Fig.5.6 Area of HSLPL

Simulation results go well by showing that the proposed architecture is able to increase throughput by a factor of more than twice for random traffic, also reducing latency.

5.3 DESIGN OF LMSTRDP:

With continuous shrinking technology the power dissipation ratio of NoC links and routers increases making links more power hungry than routers. Dynamic power consumption is the major source of power consumption in NoCs and it occurs due to self switching and cross coupling capacitance. The large intrinsic capacitance associated with buses is responsible for a substantial fraction (approx 40%) of total power dissipated, because the bus power dissipation is proportional to switching activity.

The main drawback with the existing encoding schemes such as Bus Invert and Bus Invert Transition Signalling is the extra bus line used to indicate the receiver that the data is encoded.

In our proposed link design for low power consumption a methodology has been proposed to get rid of extra bit line. Proposed technique reduces power consumption by taking into consideration the contribution of both self switching and cross talk. In the proposed encoder data are rearranged in such a way that the transition in each link is minimised thus reducing power dissipation due to self switching. Proposed decoder is designed to reduce cross couple activity as the inverting of data depends upon the

contribution of the cross couple activity. Fig. 5.7 shows a technique where no. of transitions is reduced to a great extent so good reduction of power can be easily seen. The input binary data is converted into gray code and then the data is sent as it is, through the data bus. From the next gray-coded data, AND operation is performed between 4th and 5th bits for an 8-bit data. If this operation results in '0', then XOR operation is performed between the remaining six bits of gray-coded data (i.e. Lower three bits and upper three bits) and the six bits (lower three bits and upper three bits) of the previous encoded data. The output of the XOR combined with the 4th and 5th bits is sent through the data bus. If the AND operation results in '1', then XNOR operation is performed in the same way.

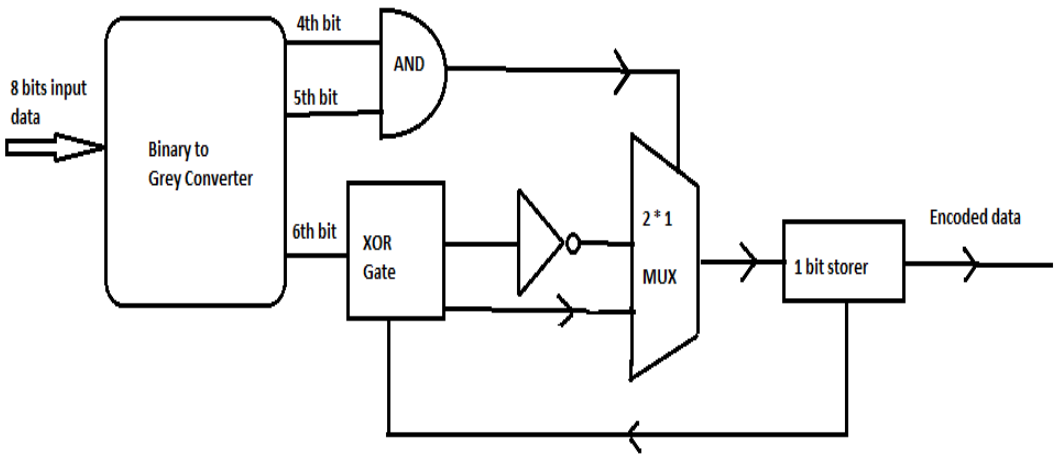


Fig 5.7 Encoder of LMSTRDP

At the decoder side in Fig. 5.8, again AND operation is done between 4th and 5th bits of the data received through the data bus. If it results in '0', then XOR operation is done between the six bits of the received data and the six bits of the previously received data. The output so received is then converted into binary.

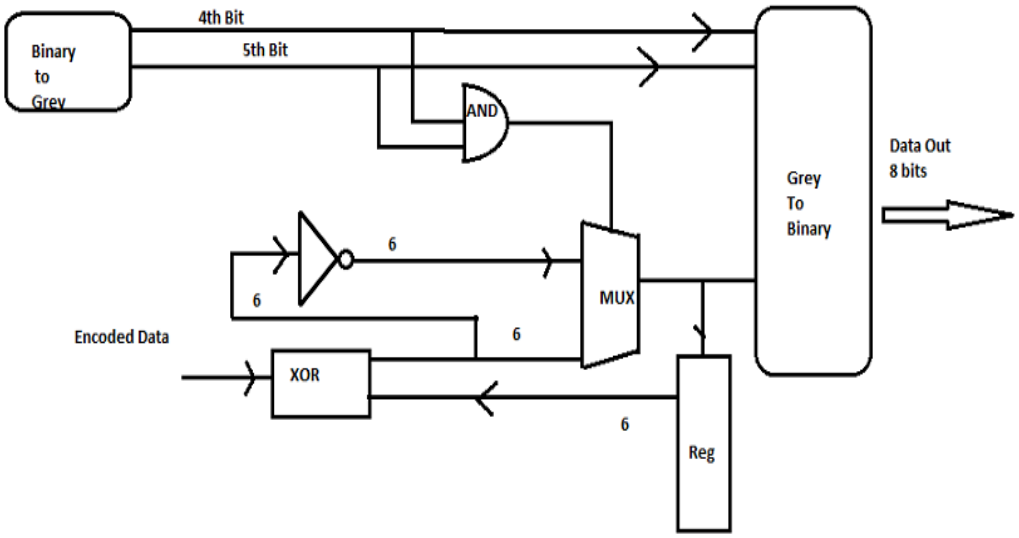


Fig .5.8 Decoder of LMSTRDP

5.3.1Simulation results:

In Fig. 5.9 data coming out from router is encoded thereby reducing transition and then received at decoder of destined router.

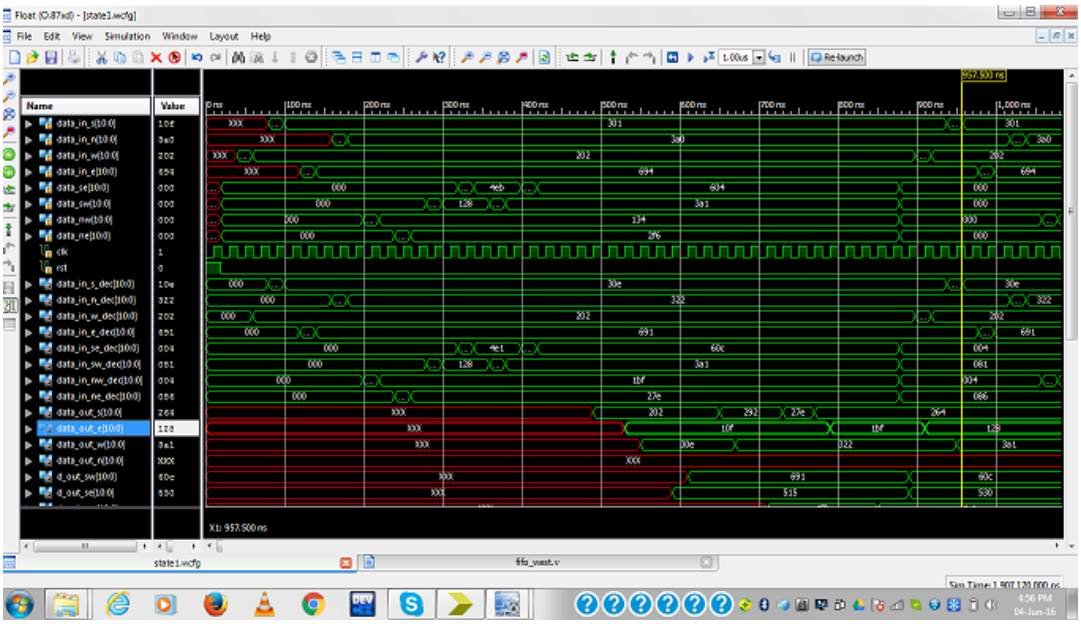


Fig.5.9 Simulation results of LMSTRDP

Fig. 5.10 shows the RTL schematic view of encoders and decoders attached with South Router. It can be seen that decoders are fixed in the links to decode the data coming from previous router.

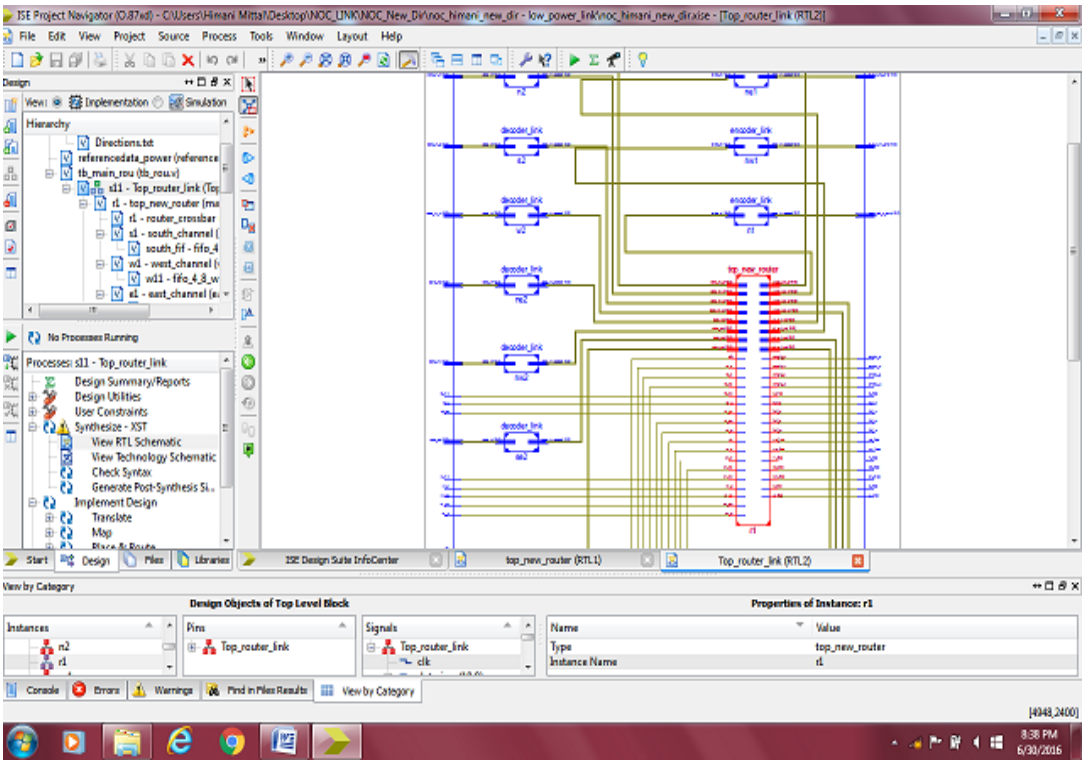


Fig.5.10 RTL Schematic view of LMSTRDP

Fig. 5.11 shows the RTL view of south decoder as an individual entity. It shows various basic components that make up a decoder.

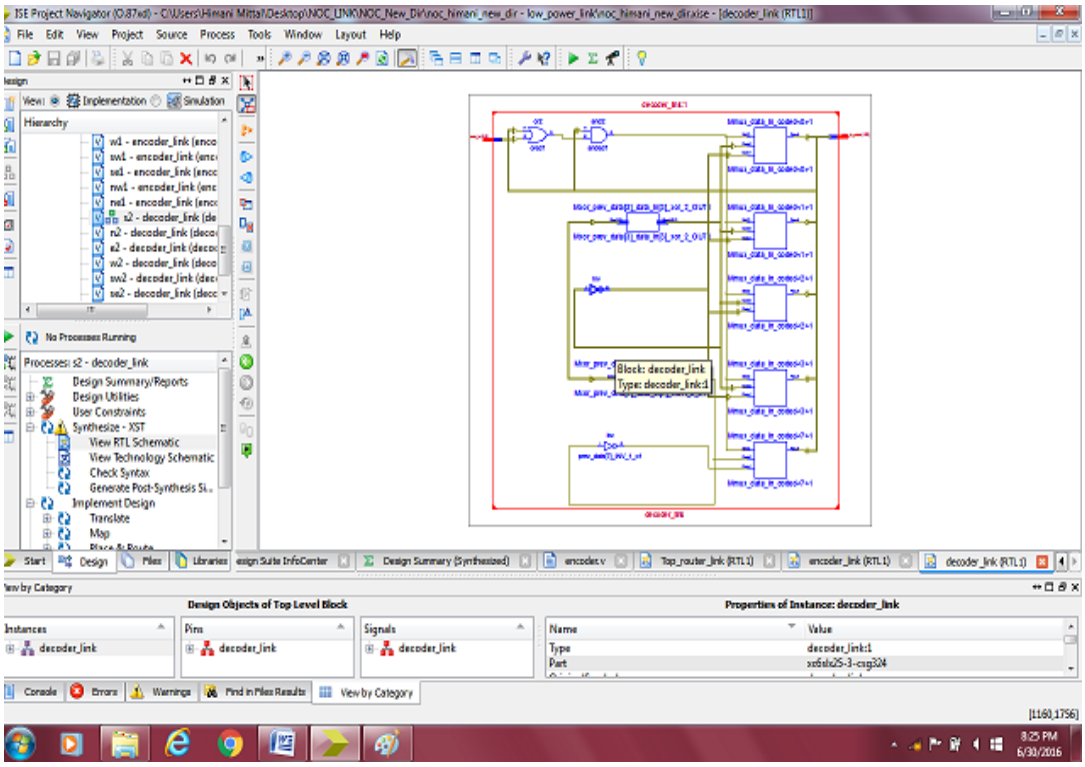


Fig. 5.11RTL view of south decoder of LMSTRDP

Fig. 5.12 shows RTL view of south encoder again as a separate entity and also various basic components that make up an encoder.

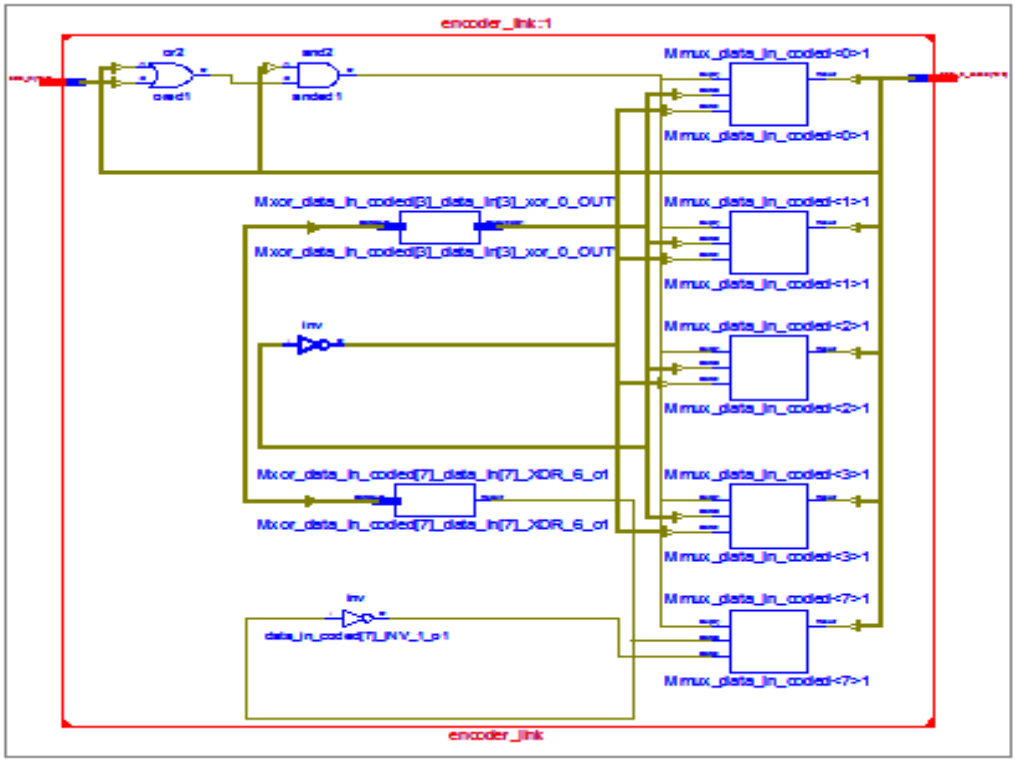


Fig. 5.12 RTL View of South Encoder of LMSTRDP

Fig. 5.13 shows the RTL view of top router module along with LMSTRDP encoder and decoder. It shows how the data coming out is encoded and how the data entered in top router is decoded to reduce switching transitions.

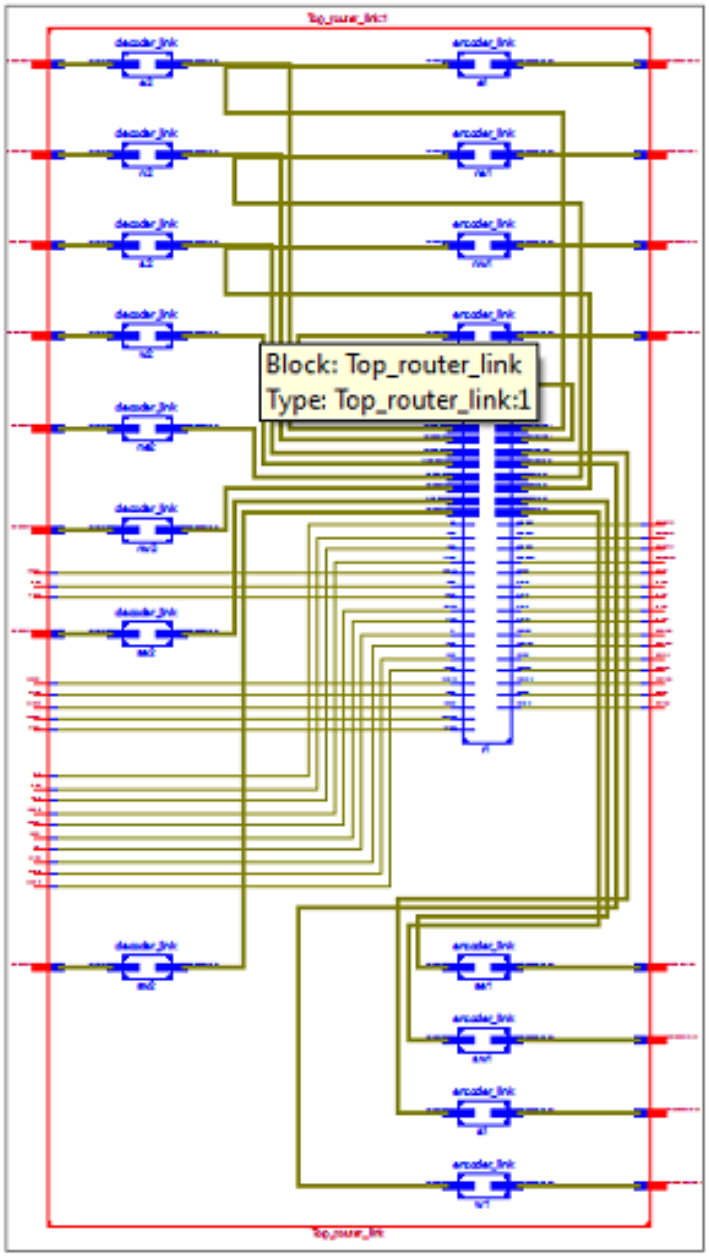


Fig.5.13 RTL view of top module of LMSTRDP

Fig.5.14 shows the power consumption in LMSTRDP. The dynamic power consumption is found to be 0.031 w.

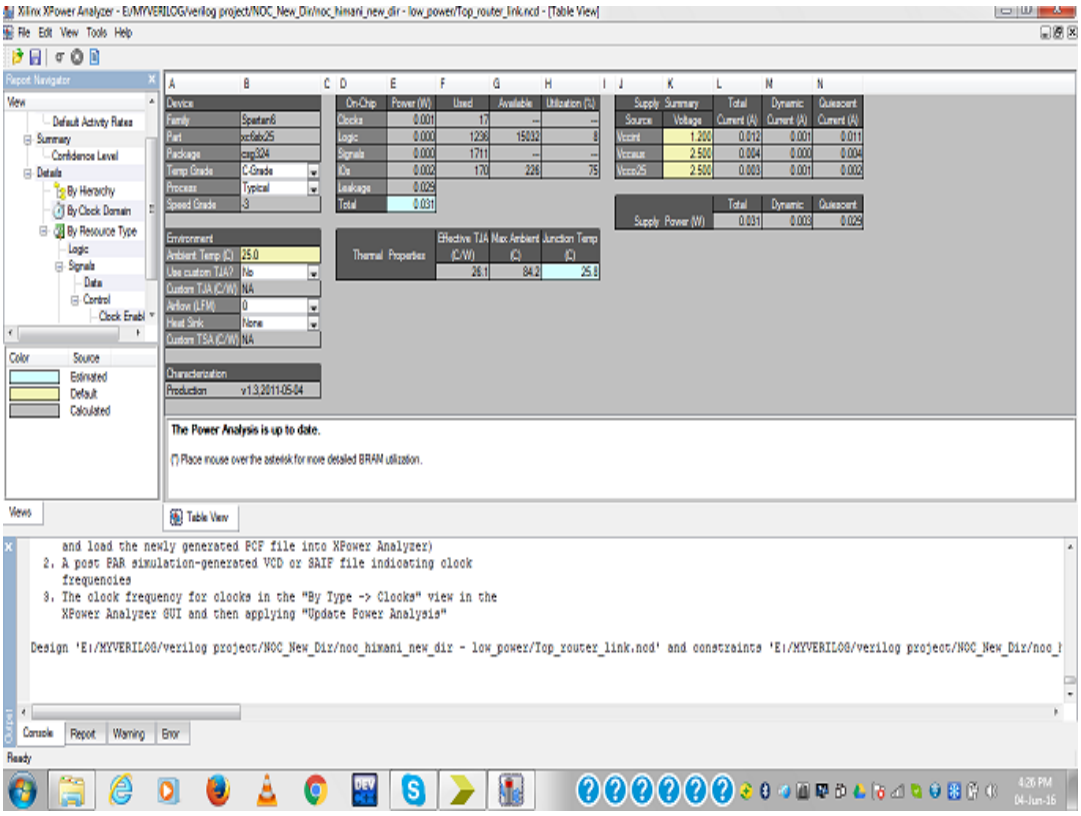


Fig.5.14 Power Dissipation of LMSTRDP

Simulation results are well explicit showing that proposed techniques is better than the existing techniques because they do not need the extra bus line from encoder to decoder to indicate the decoder that data is encoded or decoded. Fig. 5.14 shows saving of 32% of power, which is a remarkable achievement. It has very less area overhead as compared to the previous techniques. In previous techniques, most of data is unable to save power. One more significant advantage of this technique is that out of every data only 18% of total data combinations dissipate power which is great achievement that we can't achieve with previous techniques.

5.4 DESIGN OF LMRTA:

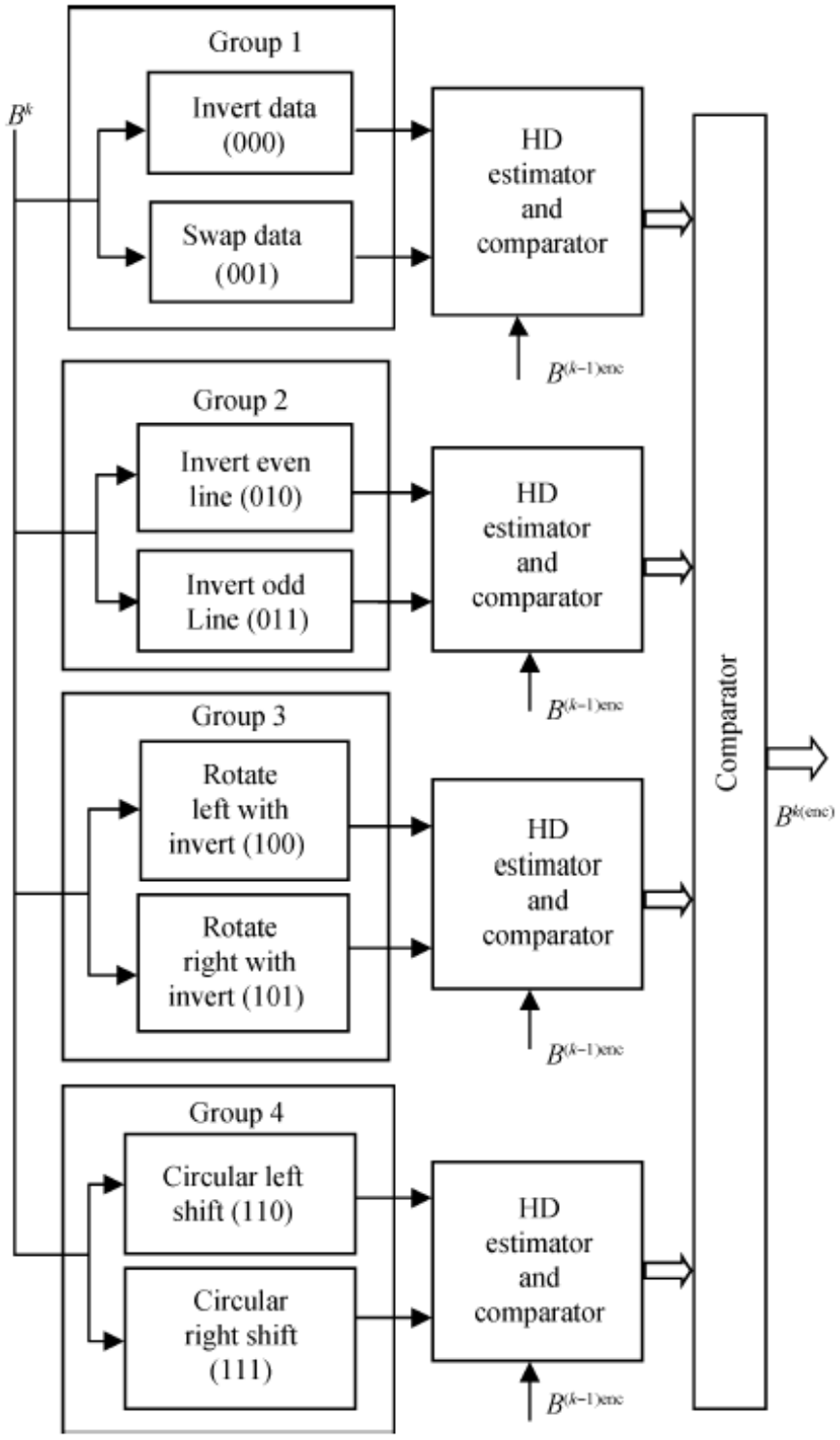
In our earlier proposed router, we did take into account the energy dissipation due to charging and discharging of internal node capacitance. Transition activities can be reduced further by doing changes in algorithm and at logic circuit level which is not done in earlier proposed router.

In proposed link design, transition activities are reduced using multi encoding algorithm and logic circuit level. At algorithm level transition activities are reduced

by different coding styles. At logic circuit level, one bit full adder is added for estimation of hamming distance which helps in reducing device count.
In this work of multi coding techniques as shown in Fig.5.15, the input data are coded in eight different ways and grouped into four. Each group has two coding techniques. Control bits are added to each coding technique to recover the original data at the decoder.

Here a 4-bit comparator is used to compare the Hamming distance of the different methods and find out the least among them. Since we are conditionally computing the Hamming distance, for any case the comparator has to compare only three values. The inputs to this block are the encoded data along with its bit representation and its corresponding Hamming distance. The output of this block is the encoded data with the least number of transitions and its 3-bit representation. This encoded data
is sent over the bus along with its 3-bit representation.

HD comparator module separately calculates the Hamming distance between the coded data and the reference data for each coding method then compares which coding method has the minimum Hamming distance. By using this concept, among the eight, four methods are discarded and the remaining four methods are again compared and the code having the least Hamming distance is transmitted via the bus. The Hamming distance estimator shown in Fig.4, includes Bus Invert (BI), adaptive coding, gray coding and transition method. Of these, we elaborate only on BI, which is shown to be the most effective in NoCs. BI compares the data to be transmitted with the current data on link. If the Hamming distance (the number of bits in which the data patterns differ) between the new information and the link state is larger than half the number of bits (wires) on the link, then the data pattern is inverted before transmission.



:

Fig.5.15 Block Diagram of LMRTA

5.4.1 Simulation Results:

In Fig. 5.16 it can be seen how data is encoded by eight different coding schemes and depending upon the least hamming distance it travels through link and then decoded to original one in destined router link.

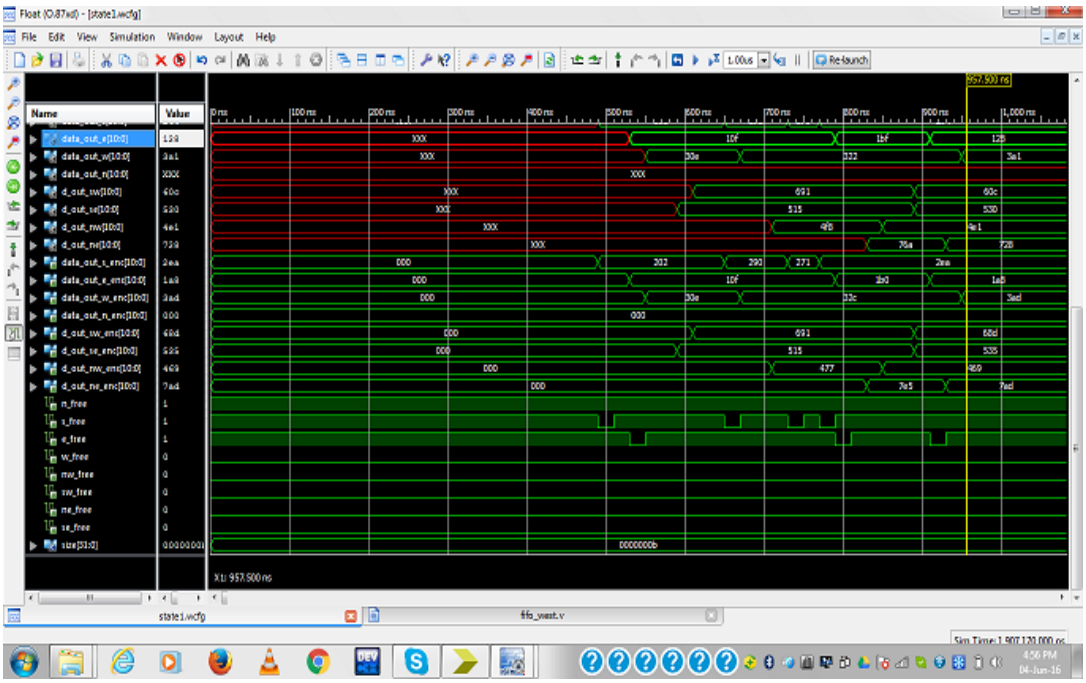


Fig.5.16 Simulation of LMRTA

RTL view of complete proposed LMRTA can be seen in Fig. 5.17.

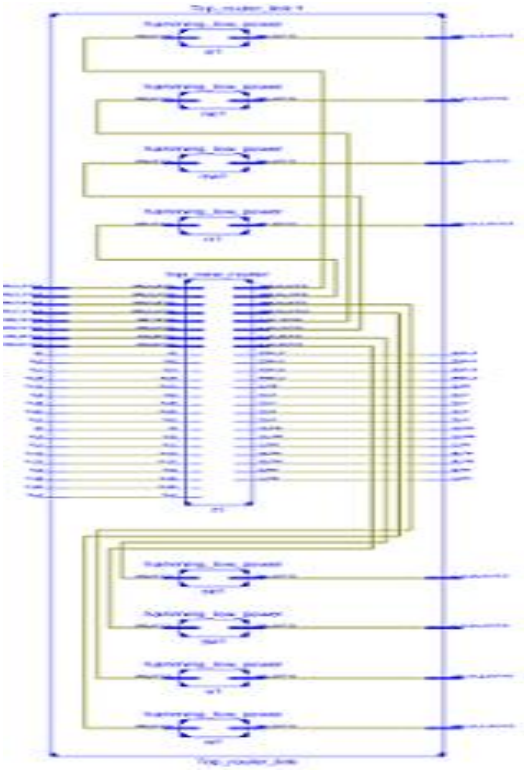


Fig.5.17 RTL views of LMRTA.

Fig. 5.18 shows that by using multi coding technique power dissipation has been reduced to .02W.

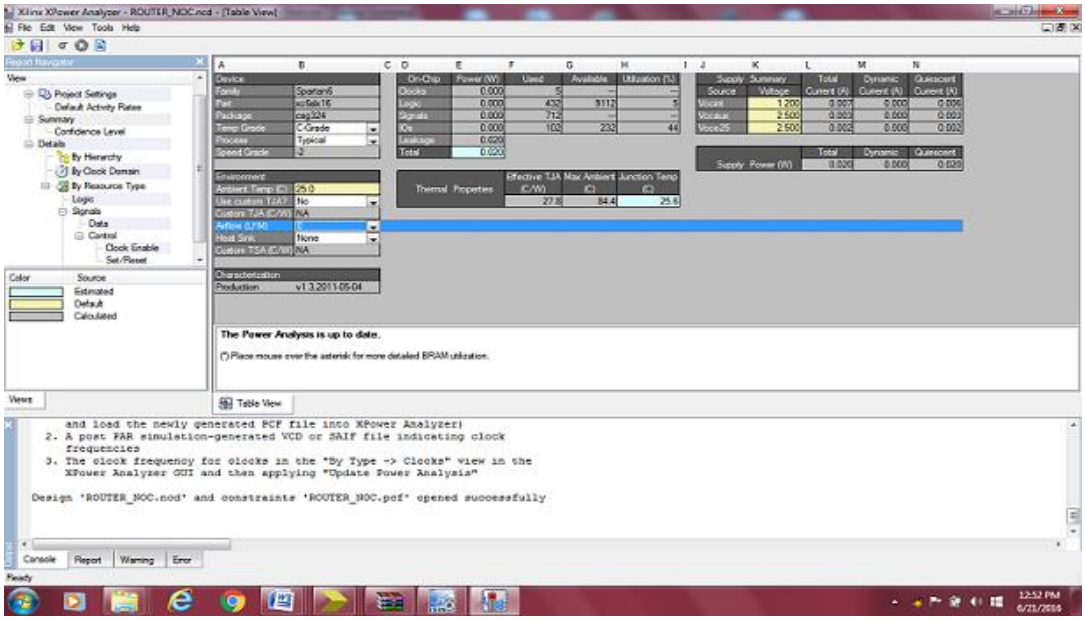


Fig.5.18 Power dissipation of LMRTA

We have successfully designed and simulated multi encoding technique to achieve energy reduction. The main goal of the proposed method is to reduce the overall transitions in reconfigurable systems. The simulation result shows the overall reduction of up to 39 % in power dissipation and also 1 % to 10% of energy reduction compared to other existing encoding methods. In this method the delay and glitches are not concentrated effectively.

5.5 COMPARISON OF HSLPL, LMSTRDP, LMRTA:

Table 5.1 shows the comparison of results obtained by simulation of power dissipation and area while designing HSLPL, LMSTRDP, and LMRTA.

Table 5.1 Chart comparison of power and area of HSLPL, LMSTRDP, LMRTA

Name of Technique	**Power dissipation(W)**	**Area Overhead(micrometer square)**	**Extra bus line needed**
Link without technique	0.054	2016	No
HSLPL	0.045	1921	No
LMSTRDP	0.031	2175	No
LMRTP	0.02	1480	No

Fig. 5.19 shows column chart of various power results obtained through simulation. Here we see that power dissipation decreases at a good pace as we modified link design from HSLPL TO LMRTP.

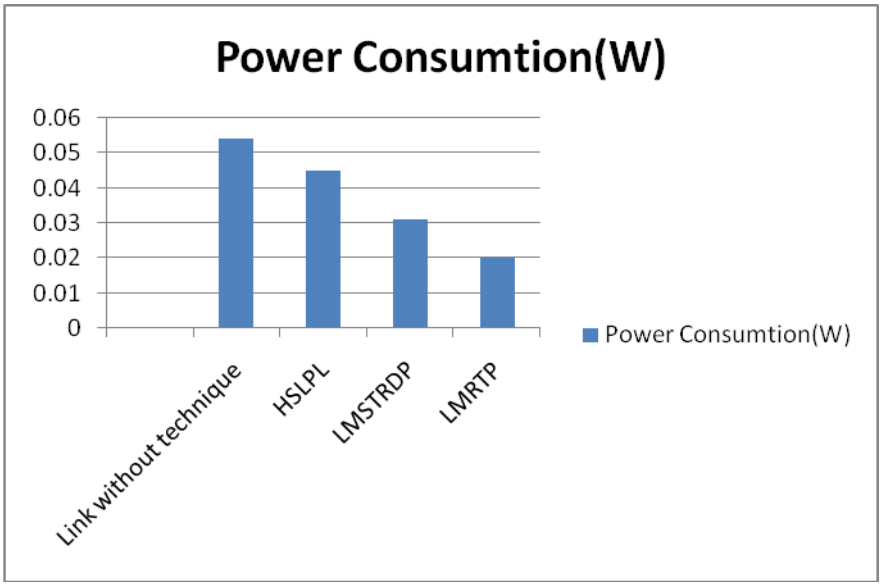


Fig. 5.19 Power consumption of HSLPL, LMSTRDP, LMRTP

Area results of link designs HSLPL, LMSTRDP, LMRTP are shown in column cart of Fig. 5.20, where in LMSTRDP there is slight increase in area at the cost of reduction in power.

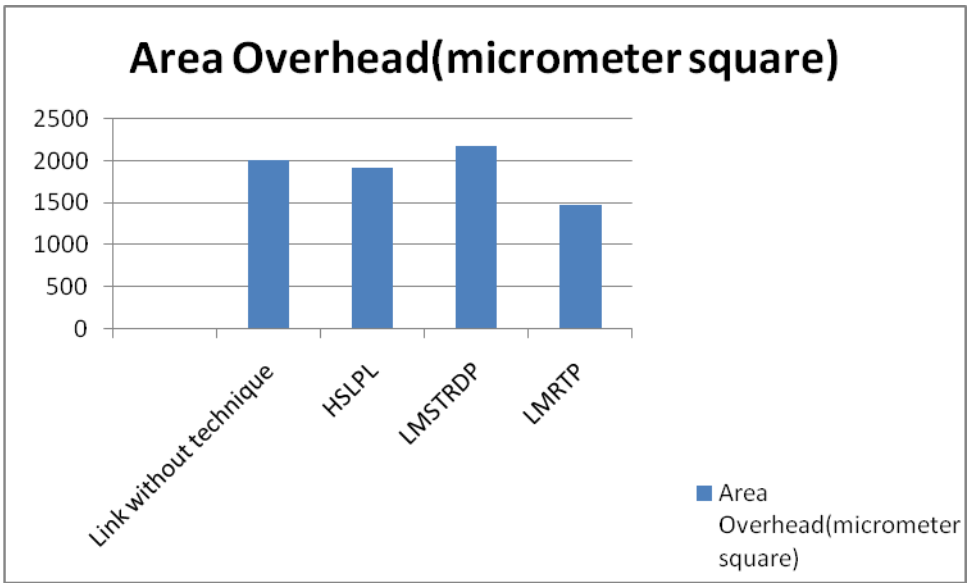


Fig. 5.20 Area comparison of HSLPL, LMSTRDP, LMRTP

Analyzing the results we find that average no. of bit transitions are reduced to more than 35% as we move from first to third link design.

Simulations demonstrate a reduction of up to 55% in the number of bit transitions and up to40% savings in power consumed on the link.

CHAPTER 6

CONCLUSION AND SUGGESTIONS FOR FUTURE WORK

The conclusions drawn from the results of the work presented in **Chapters Three** through **Five,** are presented in the current chapter. This Chapter also suggests the scope for future work in the area covered by this thesis.

6.1 CONCLUSION:

Design of scalable and low power network on chip for reconfigurable systems has been done in this thesis. The thesis begins with a brief introduction to the subject matter of the work carried out and the existing problems, which are treated in this thesis work. A survey of the existing literature is carried out to build a background and bring out the motivation for the work. Various researchers have designed NoCs applicable for reconfigurable systems. Various components like routers; links etc. have also been designed and tested by the researchers. A review of the studies and designs carried out in the area of network on chip for reconfigurable systems have been included in this thesis. Afterwards, fundamental concepts related to the reconfigurable systems, SoCs & NoCs are presented in Chapter Three.

Taking the support of the designs found in the literature, the problems at hand were tackled. The resulting studies and designs are presented in Chapters Four and Five. For this purpose, firstly topology selection has been done. Various topologies are listed and compared in Chapter Three. Various performance metrics like maximum end-to-end Latency, dropping probability and throughput etc. are used for the comparison purpose. It has been found that 2D mesh is suitable for the present work.

After selecting the appropriate topology, next task is to design various components of the NoC. In the present work stress has been put on the design of the routers and links. To enhance the efficiency of the NoCs, the power efficient and reconfigurable routers without the need of oversized buffers and power efficient and high performance links are designed in the present work.

To design the router and links, various software tools have been utilized. Hardware description language Verilog is used for writing the code for all components at register transfer level. All these codes have been written in synthesizable form. Model Sim 10.3 is used for the simulation purpose and Xilinx ISE design suite 13.4 is used for the synthesis at the logic gate level. RTL schematic view, gate level view, design summary, power dissipation and performance have been obtained using this software tool. Obtained results are presented in Chapters Four and Five.

To design the power efficient and smart reconfigurable routers, several versions of it are designed and tested. These versions are

a. RRIE (Reconfigurable Router to Improve Efficiency).

b. HRRLPHP (Heterogeneous Reconfigurable Router for Low Power and High Performance).

c. EDRRHT (Eight Directional Reconfigurable Router for High Throughput).

d. SRRODFB (Smart Reconfigurable Router for Online Detection of Faulty Blocks).

In the proposed RRIE, NoC efficiency has been compared as a function of the possibility to reconfigure the buffer size according to the requirements of each channel of the router at run time. It is done without the need to oversize the buffers for the guaranteed performance. It has low area, consumes less power and provides high performance. In this approach it is possible to dynamically configure different buffer depth for the channel. This router has FIFOs, channel flow/control logic and a cross bar. It has been found that the total power dissipated by this router architecture is around 15 milliwatt which is less compared to the previous designs available in the literature. Moreover, this router obtains the same performance as that of homogeneous router but with a buffer depth which is 64% smaller. Also, it is possible to reconfigurable the router in accordance with the application in the modified architecture. It obtains similar performance even when the application changes radically.

The problem with the RRIE is that, in it when a channel starting to accept packets from its neighbour (left or right) it does not take any packet further from that channel. In other words we can say that it is not supporting heterogeneous data. Also, its area is found to be slightly high. These drawbacks are improved in the second reconfigurable

router namely HRRLPHP. In this router even if a channel is taking data from its neighbouring channel it will continue to take data from its neighbouring channel also. Hence, it supports heterogeneous data. To reduce the area, two tag bits to the packets in packet switching or flits (in wormhole switching) has been added. These bits are added before storing the packets in the FIFO to indicate the channel to which this packet or flit was input. Tag bits are '00' when the packets are directly coming to the concerned channel, '01' when packets coming from the right neighbour and '10' when the packets are coming from the left neighbour. These tag bits are also used for indicating the direction of output from the crossbar. For an 8 bit packet, the first two bits are used to indicate the address of the next router. Control circuitry is implemented in the present router by including features like acknowledge for writing, acknowledge for reading, request and grant signals for writing in another channel's FIFO. The modified channels, crossbar and FIFO are simulated and synthesized. Obtained results show that area is significantly reduced and is only 53 mm^2 while the power dissipation is found to only 0.020 W. The drawback of this router is that throughput is degraded marginally (3%). Although it is tolerable and insignificant for many applications, we have tried to remove this drawback in our next router namely EDRRHT. Another problem with HRRLPHP is that it does not use the semi-buffer concept to increase efficiency. These two things are added in our third router (EDRRHT). Four more directions (NE, NW, SE & SW) are added so that router is now able to support eight directions. It also supports buffer less storage concept to store the data of four newly added directions. Data of newly added directions share FIFOs from their neighbours to store its data. This way, this architecture requires less area for implementation. The newly designed architecture provides more routes to router to route the data thus it is helpful in decreasing critical path length and help in increasing the performance of the NoC. Also, we have used rectilinear Stennier tree path to shorten the data path of the router. We have used a circular FIFO in place of normal fall through FIFO. Also, several changes have been made in control circuitry to improve the throughout. Compared to the second router it has four additional arbiters (NW, NE, SW and SE) which support buffer less concept and is used to share data from neighbours depending on the priority which is further dependent on the priority table. Obtained results show a power dissipation of 0.045 W, which is higher

than the second router because of the addition of four more semi buffers. Area estimated is found to be 85 mm^2.

The EDRRHT is not able to handle the spotting of the faulty blocks of NoC. Also, it is not able to recognize and differentiate between permanent and transient errors and the way to handle them. Another drawback is that it lacks the suitable algorithm to help in online detection of faults while preserving its throughput, data packet latency and the network load. These drawbacks have been removed in our fourth router namely SRRODFD. This is a smart reconfigurable router for online detection of faulty blocks. It can distinguish between permanent and transient errors and is able to accurately localise the position of the faulty blocks (data bus, input port, output port) in the NoCs. It is also able to preserve the throughput, the network load and the data packet latency. Buffers are removed. In place of buffers FIFO is used to achieve better performance. Hence it is able to have low area, low power dissipation and high performance compared to the other designed routers.

A set of MUX gating and data encoding and decoding schemes aimed at reducing the power dissipated by the links of a NoC has also been designed and simulated in this work. The purpose behind the proposed schemes is to minimize the switching activity as well as the coupling switching activity. These activities are mainly responsible for link power dissipation in the deep sub-micrometer technology regime. The proposed encoding schemes are agnostic with respect to the underlying NoC architecture in the sense that their application does not require any modification neither in the routers nor in the links. The encoders implementing the proposed schemes have been assessed in terms of power dissipation and silicon area. The decoders are implemented to perform the reverse operation of encoders and hence to recover the data.

It is obvious that the reconfigurable hardware solutions are based on computer architectures able to adapt their behaviour in response to several different inputs. Also, reconfigurable techniques are applied to all levels of the systems like buffers, routers, and links. The present thesis is an effort to implement this principle. Furthermore, scalability, power efficiency, low area and high performance are the other objectives to be fulfilled by modern embedded NoCs for reconfigurable

systems. By applying all the above mentioned concepts, the present thesis is able to fulfil its objective.

6.2 SUGGESTIONS FOR FUTURE WORK:

The solutions presented and architecture provided in this work is neither optimal nor complete. So there is always lot of scope for modifications.

1. Eun Lee and Bagherzadeh [107] presented the use of various clocks while sending flits in the NoC. Body flits operate faster than head flits. So accordingly, as FIFOs work faster than the route decision, it is possible to use various clocks to feed body flits or head flits. While the head flit is analyzed to tell the route path, body flits can continue advancing along the reserved path already established, improving the performance of the router.

Thus, with the help of different clocks it could be subjugated in the current work so as to obtain the better delays and latency factor.

2. Ahmad et al. [66] discussed a network router designing with a bus like interface. A built-in wrapper is utilized, and because of this any bus compatible component could be integrated into the NoC architecture. The interface of this router would be a simple bus. The aim of this discussion was to reduce the design time and to ease integration. When the network has a channel that requires high bandwidth, the NoC changes the switching, obtaining a dedicated path between the IPs. This discussion could also be done as advancement to the current project module. In turn it will reduce the paradigm shift totally but up to certain level only, hence making the model to easily associate with the present SoC models.

3. The disadvantage of the previously discussed SoC proposals was that all of them must be used at design time, generally with a static approach, hence causing problems of scalability, whenever the NoC is used for a new application in the same platform.

Thus, updating of product, customization of an MPSOC or a different market, using the same components, can be considered paramount for futuristic approach; although it seems that it results up being costlier than projected.

4. Link: Data packets travels among various cores in a NoC by the use of either a serial or a parallel link. Parallel links use a buffer-based architecture and can be operated at a relatively lower clock rate in order to reduce power dissipation. These parallel links come up with high silicon cost due to inter-wire spacing, shielding and repeaters. It can be optimised to a certain limit by employing multiple metal layers. If

seeing the other side, serial links allow savings in wire area, minimization in signal interference and noise, and limits the need for having buffers. Serial links have the advantages of a simpler layout and simpler timing verification. Serial links faces SI (Intersymbol Interference) between successive signals while operating at high clock rates. Encoding along with asynchronous communication protocols can take care of these drawbacks.

6. Optimization through Links Interconnect: Network on chip communication is based on modules connected via a network of routers with links between the routers that comprise of long interconnects. So minimizing interconnects in order to achieve the required system performance is required. By inserting repeaters global wires timing optimization is obtained. This will result in a makeable increase in cost, area, and power consumption. Present research states that in the near future, inverters operating as repeaters will occupy a large portion of chip resources. Thus, there is a need for optimizing power on the NoC. Encoding is an effective way of reducing dynamic power consumption. Innovative ways will have to be explored to optimize the power consumed by the on-chip repeaters.

REFERENCES

1. T. Morgan, "**Oracle cranks up the cores to 32 with sparc m7 chip**," technical representation, Enterprise Technical System Edition, 2014.

2. International Roadmap Committee, **The International Technology Roadmap for Semiconductors** *(2012) [online]*. *Ava*ilable: http://www.itrs.net/reports.html.

3. ***Xilinx Corporate Overview*** (Xilinx Inc., San Jose, 2013) [online]. Available: http://www.xilinx.com/aboutus/corporate_overview.pdf.

4. J. D. Owens, W. J. Dally, R. Ho, D. N. Jayasimha, S. W. Keckler and L. S. Peh, "**Research Challenges for On-Chip Interconnection Networks**," IEEE Micro, vol. 27, no. 5, pp. 96-108, Sept.-Oct. 2007.

5. P.E. Gaillardon,"**Reconfigurable logic: Architecture, tools and applications**", CRC Press, Inc., Boca Raton, Florida, USA, Oct. 2015.

6. P.A. Hsiung, M. D. Santambrogio, and C.H. Huang," **Reconfigurable System Design and Verification**", CRC Press, Inc., Boca Raton, Florida, USA, 2009.

7. S. M. Trimberger, "**Three Ages of FPGAs: A Retrospective on the First Thirty Years of FPGA Technology**," Proceedings of the IEEE, *vol*. 103, no. 3, pp. 318-331, March 2015.

8. J. C. Lyke, C. G. Christodoulou, G. A. Vera and A. H. Edwards, "**An Introduction to Reconfigurable Systems,**" Proceedings of the IEEE, vol. 103, no. 3, pp. 291-317, March 2015.

9. W. J. Dally and B. Towles, "**Route packets, not wires: on-chip interconnection networks**," Proceedings of the IEEE 38th Design Automation Conference, pp. 684-689, March 2001.

10. L. Benini and G. De Micheli, "**Networks on chips: a new SoC paradigm**," Computer, *vol.* 35, no. 1, pp. 70-78, Jan 2002.

11. F. Moraes, N. Calazans, A. Mello ,"**HERMES: an Infrastructure for Low Area Overhead Packet-switching Networks on Chip,"** Integration, the VLSI Journal, vol. 38, no. 1, pp 69-93, Oct. 2004.

12. “**A Comparison of Network-on-chip and Buses,**” White Paper, Arteris, SA, 2005.

13. A. Ben Abdallah, M. Sowa, "**Basic network-on-chip interconnection for future gigascale MCSoCs applications: communication and computation orthogonalization",** Proceedings of Tunisia-Japan Symposium on Society, Science and Technology (TJASSST), pp 4-9, Dec. 2006 .

14. **Xilinx. Inc. Zynq-7000**, All Programmable SoC Technical Reference Manual (UG585), March 2013[online].Available: http://www.xilinx.com/support/documentation/user_guides/ug585-Zynq-7000-TRM.pdf.

15. F.A. Samman, T. Hollstein, M. Glesner,"**Multicast parallel pipeline router architecture for network-on-chip**", Proceedings of the Conference on Design, Automation and Test in Europe, pp. 1396–1401, Munich, Germany, Mar 2008.

16. W.J. Dally and B.Towles, “**Principles and Practices of Interconnection Networks,”** Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 2003.

17. R. Marculescu, J. Hu and U. Y. Ogras, "**Key research problems in NoC design: a holistic perspective**," Third IEEE/ACM/IFIP International Conference on Hardware/Software Co design and System Synthesis , pp. 69-74 , New Jersey, USA, 2005.

18. R. Marculescu, U. Y. Ogras, L. S. Peh, N. E. Jerger and Y. Hoskote, "**Outstanding Research Problems in NoC Design: System, Microarchitecture, and Circuit Perspectives,"** IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 28, no. 1, pp. 3-21, Jan. 2009.

19. R. Saleh, S. Wilton, S. Mirabbasi, A. Huan, M. Greenstreet, G. Lemieux, P.P. Pande, C. Grecu and A. Ivanov**, “System-on-Chip: Reuse and Integration, ”** Proceedings of the IEEE , vol. 94, no. 6, pp.1051-1069, June 2006 .

20. C. Bobda, A. Ahmadinia, M. Majer, J. Teich, S. Fekete, and J. van der Veen, “**DyNoC: a dynamic infrastructure for communication in dynamically**

reconfigurable devices," Proceedings of the IEEE International Conference on Field Programmable Logic and Applications, pp. 153–158, August 2005.

21. L. Benini and G. De Micheli, "**System-Level Power Optimization: Techniques and Tools**," ACM Trans. Design Automation of Electronic Systems, pp. 115-192,April 2000.

22. A.B. Achballah and S.B. Saoud**, "A Survey of Network-On-Chip Tools," I**nternational Journal of Advanced Computer Science and Applications, pp. 1-7, Dec. 2013.

23. H.Chang, L. Cooke, M.Hunt, G. Martin, A. McNelly and L. Todd**,"Surviving the SoCs Revolution a Guide to Platform-Based Design,"** Kluwer Academic Publishers, pp. 3-4**,** Edition 2002.

24. M. Keating and P. Bricaud, Reuse MethodologyManual: **, For System-on-a-Chip Designs,** 3rd ed. Boston, MA: Kluwer, 2002.

25. P. Guerrier and A. Greiner**, "A generic architecture for on-chip packet-switched interconnections,"** Proceedings of the Design, Automation and Test in Europe Conference and Exhibition, pp. 250–256, March 2000.

26. S.Jovanovic,C. Tanougast and S. Weber**,"A New High- Performance Scalable Dynamic Interconnection for FPGA-based Reconfigurable System," I**EEE International Conference on Application Specific System, Architecture and Processor, pp. 61-66, July 2008.

27. S. Pasricha and N. Dutt, **"On-Chip Communication Architectures System on Chip Interconnect. ,** pp. 439-455, Edition 2008.

28. J.Duato, S.Yalamanchili, L. Ni**., "Interconnection networks: an Engineering Approach,"** Morgan Kauffman, 2003.

29. C. E. Leiserson**, "Fat-trees: Universal networks for hardware-efficient supercomputing,"** IEEE Transactions on Computers, vol. 34, No. 10, pp. 892 – 901, October 1985.

30. A. Naeemi, R. Sarvari and J. D. Meindl**, "On-chip interconnect networks at the end of the roadmap: limits and nanotechnology opportunities,"** International Interconnect Technology Conference, pp.201-203, 2006.

31. Sweta Sahu and Harish M. Kittur, "**Area and Power Efficient Network on Chip Router Architecture,"** IEEE Conference On Information and Communication Technology, pp. 855-859, April 2009.
32. M. Meribout, "**A new scalable and reconfigurable architecture**," IEEE Potentials, vol. 22, no. 3, pp. 26-32, Aug.-Sept. 2003.

33. R. S. Cardoso, M. E. Kreutz, L. Carro and A. A. Susin, "Design **space exploration on heterogeneous network-on-chip**," IEEE International Symposium on Circuits and Systems, vol.1, pp. 428-431, July 2005.

34. S. Jovanovic, C. Tanougast and S. Weber, "**A new high-performance scalable dynamic interconnection for FPGA-based reconfigurable systems**," International Conference on Application-Specific Systems, Architectures and Processors, pp. 61-66, July 2008.

35. B. Ahmad, A. Ahmadinia and T. Arslan, "**Dynamically Reconfigurable NoC with Bus Based Interface for Ease of Integration and Reduced Design Time**," NASA/ESA Conference on Adaptive Hardware and Systems, pp. 309-314 , June 2008.

36. S. V. Rana, M. D. Santambrogio and D. Sciuto, "**A light-weight Network-on-Chip architecture for dynamically reconfigurable systems**," International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation, pp. 49-56, Jun. 2011.

37. L. Sterpone, M. Porrmann and J. Hagemeyer, "**A Novel Fault Tolerant and Runtime Reconfigurable Platform for Satellite Payload Processing**," IEEE Transactions on Computers, vol. 62, no. 8, pp. 1508-1525, Aug. 2013.

38. Z. J. Yang, A. Kumar and Y. Ha, "**An area-efficient dynamically reconfigurable Spatial Division Multiplexing network-on-chip with static throughput**

guarantee," International Conference on Field-Programmable Technology, pp. 389-392, Dec. 2010.

39. T. Pionteck, R. Koch, and C. Albrecht, "**Applying partial reconfiguration to networks-on-chips**," Proceedings of the International Conference on Field Programmable Logic and Applications, pp. 1–6, August 2006.

40. D.N. Sarma, G. Lakshminarayanan, K.V.R. Suryakiran Chavali, "**A Novel Encoding Scheme for Low Power in Network on Chip Links**", VLSI Design 25th International Conference on, pp. *257-261, 2012.*

41. L. Sterpone, M. Porrmann and J. Hagemeyer, "**A Novel Fault Tolerant and Runtime Reconfigurable Platform for Satellite Payload Processing,"** IEEE Transactions on Computers, vol. 62, no. 8, pp. 1508-1525, Aug. 2013.

42. D. Park, C. Nicopoulos, J. Kim, N. Vijaykrishnan, and C. R. Das, "**Exploring fault-tolerant network-on-chip architectures**," Proceedings of the International Conference on Dependable Systems and Networks , pp. 93–104, IEEE, June 2006.

43. P. h. Pham, Y. Kumar and C. Kim, "**A Compact and High Performance Switch for Circuit-Switched Network-On-Chip,**" IEEE International SOC Conference, pp. 53-56, Sep. 2006.

44. H. Zimmer and A. Jantsch, "**A fault model notation and error-control scheme for switch-to-switch buses in a network-on-chip,"** Proceedings of the 1st IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis , pp. 188–193, Oct. 2003.

45. B. Kia, K. Mobley and W. L. Ditto, "**An Integrated Circuit Design for a Dynamics-Based Reconfigurable Logic Block**," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 64, no. 6, pp. 715-719, June 2017.

46. E. Kakoulli, V. Soteriou, C. Koutsides and K. Kalli, "**Silica-Embedded Silicon Nanophotonic On-Chip Networks**," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 36, no. 6, pp. 978-991, June 2017.

47. M. P. Vestias and H. C. Neto, "**A Generic Network-on-Chip Architecture for Reconfigurable Systems: Implementation and Evaluation**," International Conference on Field Programmable Logic and Applications, pp. 1-4,April 2007.

48. A. Ivanov and G. De Micheli, "**Guest Editors' Introduction: The Network-on-Chip Paradigm in Practice and Research,**" IEEE Design & Test of Computers, vol. 22, no. 5, pp. 399-403, Sept.-Oct. 2005.

49. J. L. Manferdelli, N. K. Govindaraju and C. Crall, "**Challenges and Opportunities in Many-Core Computing**," Proceedings of the IEEE, vol. 96, no. 5, pp. 808-815, May 2008.

50. D. DiTomaso, R. Morris, A. K. Kodi, A. Sarathy and A. Louri, "**Extending the Energy Efficiency and Performance With Channel Buffers, Crossbars, and Topology Analysis for Network-on-Chips**," IEEE Transactions on Very Large Scale Integration Systems, vol. 21, no. 11, pp. 2141-2154, Nov. 2013.

51. D. Bertozzi et al., "**NoC synthesis flow for customized domain specific multiprocessor systems-on-chip**," IEEE Transactions on Parallel and Distributed Systems, vol. 16, no. 2, pp. 113-129, Feb. 2005.

52. D. Bertozzi and L. Benini, "**Xpipes: a network-on-chip architecture for gigascale systems-on-chip**," IEEE Circuits and Systems Magazine, vol. 4, no. 2, pp. 18-31, 2004.

53. K. Srinivasan and K. S. Chatha, "**A Low Complexity Heuristic for Design of Custom Network-on-Chip Architectures,**" Proceedings of the Design Automation & Test in Europe Conference, pp. 1-6 , Mar. 2006.

54. M. A. Al Faruque, T. Ebi and J. Henkel, "**AdNoC: Runtime Adaptive Network-on-Chip Architecture**," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 20, no. 2, pp. 257-269, Feb. 2012.

55. W. Wolf, A.A. Jerraya, and G.Martin, "**Multiprocessor System-on-Chip Technology**," IEEE Transaction on Computer-Aided Design of Integrated Circuit and System, Vol. 27, no. 10, pp.1701-1713, Oct. 2008.

56. Phi-Hung Pham, Phuong Mau and Chulwoo Kim, "**A 64-PE Folded-Torus Intra-chip Communication Fabric for Guaranteed Throughput in Network-on-Chip Based Applications**," IEEE Custom Integrated Circuit Conference, pp. 645-648, Sept 2009.

57. K. Reddy, T.N Swain, A.K Singh, J.K. Mahapatra, "**Performance assessment of different Network-on-Chip topologies**," Devices, Circuits and Systems, 2nd International Conference on, vol., no., pp.1-5, 6-8 March 2014.

58. S.D. Chawade, M.A Gaikwad and R.M Patrikar, "**Design XY Routing Algorithm for Network-On-Chip Architecture,**" International Journal of Computer application, Vol. – 43, No.-21, pp. 1-5, May 2012.

59. P. Hao, H. QiI, D. Jiaqin, and P. Pan, "**Comparison of 2D MESH Routing Algorithm in NoCs**," IEEE International Conference on ASIC, pp. 791-795, Oct. 2011.

60. A. Roca, J. Flich, F. Silla, and J. Duato, "**A Latency-Efficient Router Architecture for CMP Systems,**" IEEE Euromicro conference on digital system designs architecture, method, tool, pp. 165-172, Sept 2010.

61. B. Ahmad, A. Ahmadinia, T. Arslan, T., "**Dynamically Reconfigurable NOC with Bus Based Interface for Ease of Integration and Reduced Designed Time**", NASA/ESA Conference on Adaptive Hardware and Systems, pp. 309-314, 2008.

62. T. Krishna, C. H. O. Chen, W. C. Kwon and L. S. Peh, "**Smart: Single-Cycle Multihop Traversals over a Shared Network on Chip**," IEEE Micro, vol. 34, no. 3, pp. 43-56, May-June 2014.

63. A. Barzinmehr and S. Tosun, "**Energy-aware application-specific topology generation for 3D Network-on-Chips"**, IEEE 20th International Symposium on Design and Diagnostics of Electronic Circuits & Systems, pp. 84-87, April 2017.

64. S. Abadal et al., "**Scalability of Broadcast Performance in Wireless Network-on-Chip**," IEEE Transactions on Parallel and Distributed Systems, vol. 27, no. 12, pp. 3631-3645, Dec. 1 2016.

65. H. K. Mondal, S. H. Gade, R. Kishore, S. Kaushik and S. Deb, "**Power efficient router architecture for wireless Network-on-Chip**," 17th International Symposium on Quality Electronic Design, pp. 227-233, Mar. 2016.

66. S. F. Fard and B. F. Cockburn, "**Reconfigurable performance measurement system-on-a-chip for baseband wireless algorithm design and verification,**" IEEE Wireless Communications, vol. 19, no. 6, pp. 84-91, Dec.2012.

67. D. Matos, C. Concatto, M. Kreutz, F. Kastensmidt, L. Carro and A. Susin, " **Reconfigurable Routers for Low Power and High Performance,**" IEEE Transactions on Very Large Scale Integration Systems, vol. 19, no. 11, pp. 2045-2057, Nov. 2011.

68. R. Hegde and N. Shanbhag, "**Toward Achieving Energy Efficiency in Presence of Deep Submicron Noise**," IEEE Transaction VLSI Systems, pp. 379-391 Aug. 2000.

69. M. P. Vestias and H. C. Neto, "**Router Design for Application Specific Networks-on-Chip on Reconfigurable Systems,**" International Conference on Field Programmable Logic and Applications, pp. 389-394, Aug. 2007.

70. B. S. Mohammad, H. Saleh and M. Ismail, "**Design Methodologies for Yield Enhancement and Power Efficiency in SRAM-Based SoCs**," IEEE Transactions on Very Large Scale Integration Systems, vol. 23, no. 10, pp. 2054-2064, Oct. 2015.

71. R. S. Cardoso, M. E. Kreutz, L. Carro and A. A. Susin, " **Design space exploration on heterogeneous network-on-chip**," IEEE International Symposium on Circuits and Systems, Vol.1 , pp. 428-431 ,July 2005

72. G. Angelopoulos, M. Médard and A. P. Chandrakasan, "**Harnessing Partial Packets in Wireless Networks: Throughput and Energy Benefits**," IEEE Transactions on Wireless Communications, vol. 16, no. 2, pp. 694-704, Feb. 2017.

73. D. K. Panda, S. Singal and R. Kesavan, "**Multidestination message passing in wormhole k-ary n-cube networks with base routing conformed paths**," IEEE Transactions on Parallel and Distributed Systems, vol. 10, no. 1, pp. 76-96, Jan 1999.

74. A. Tran and B. Baas, "**Achieving High-Performance On-Chip Networks With Shared-Buffer Routers**" , IEEE Transactions on Very Large Scale Integration Systems , vol. 22 , no. 6 , pp.1391 -1403 , March 2014 .

75. M. Hosseinabady, M. R. Kakoee, J. Mathew and D. K. Pradhan, "**Low Latency and Energy Efficient Scalable Architecture for Massive NoCs Using Generalized de Bruijn Graph**," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 19, no. 8, pp. 1469-1480, Aug. 2011.

76. M. Modarressi, A. Tavakkol and H. Sarbazi-Azad, "**Virtual Point-to-Point Connections for NoCs**," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 29, no. 6, pp. 855-868, June 2010.

77. A. Mineo, M. Palesi, G. Ascia, P. P. Pande and V. Catania, "**On-Chip Communication Energy Reduction Through Reliability Aware Adaptive Voltage Swing Scaling**," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 35, no. 11, pp. 1769-1782, Nov. 2016.

78. C.A. Zeferino, M.E. Kreutz and A.A. Susin, "**RASOCSS: A Router Soft-Core for Networks-on-Chip**," IEEE Proceedings of the Design, Automation and Test in Europe Conference and Exhibition, Vol.3, pp.198-203, Feb 2004.

79. C.K. Hsu, K.L.Tsai, J.F. Jheng, S.J. Ruan, C.A. Shen, "**A low power detection routing method for bufferless NoC**," International Symposium on Quality Electronic Design (ISQED), vol. 16, no .1, pp.364-367, 4-6 March 2013.

80. E. Vansteenkiste, B. A. Farisi, K. Bruneel and D. Stroobandt, "**TPaR: Place and Route Tools for the Dynamic Reconfiguration of the FPGA's Interconnect Network**," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 33, no. 3, pp. 370-383, March 2014.

81. A. Shrivastava and A.K. Pandit, “**Design and Performance Evaluation of a NOCS-Based Router Architecture for MPSOC**,” IEEE International conference on computational intelligence and communication network, pp. 468-472, Nov 2012.

82. P. P. Pande, C. Grecu, A. Ivanov, R. Saleh and G. De Micheli, "**Design, synthesis, and test of networks on chips**," IEEE Design & Test of Computers, vol. 22, no. 5, pp. 404-413, Sept.-Oct. 2005.

83. S. Yue, L. Chen, D. Zhu, T. M. Pinkston and M. Pedram, "**Smart Butterfly: Reducing static power dissipation of network-on-chip with core-state-awareness**," IEEE/ACM International Symposium on Low Power Electronics and Design, pp. 311-314, 11th-12th Aug. 2014.

84. Gowthaman T V and Mugilan D, "**Elastic buffer for Virtual Channels in heterogenous switching Network on Chip**," International Conference on Wireless Communications, Signal Processing and Networking, pp. 2045-2048, March 2016.

85. M.Lakshmi swethlana, V.Lavanya, and Dr.R.Ramana reddy,“ **Design and Verification Eight Port Router for Network on Chip**,” International Journal of Engineering Research & Technology , Vol. 1 , pp.1-5, July 2012.

86. R. Kumar and A. Gordon-Ross, "**MACS: A Highly Customizable Low-Latency Communication Architecture**," IEEE Transactions on Parallel and Distributed Systems, vol. 27, no. 1, pp. 237-249, Jan. 1 2016.

87. A. Bouhraoua and M. E. Elrabaa, "**Addressing Heterogeneous Bandwidth Requirements in Modified Fat-Tree Networks-on-Chips**," 4th IEEE International Symposium on Electronic Design, Test and Applications, pp. 486-490, Jan 2008.

88. H. Jingcao, U. Y. Ogras, and R. Marculescu, “**System-Level Buffer Allocation for Application-Specific Networks-on-Chip Router Design,"** IEEE Transactions on Computer Aided Design Integrated Circuits System, vol. 25, no. 12, pp. 2919–2933, Dec. 2006.

89. T. Boraten; A. Kodi, "**Runtime Techniques to Mitigate Soft Errors in Network-on-Chip (NoC) Architectures**," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol.PP, no.99, pp.1-1, Oct. 2016.

90. C. Killian, C. Tanougast, F. Monteiro and A. Dandache, " **Smart Reliable Network-on-Chip**," IEEE Transactions on Very Large Scale Integration Systems, vol. 22, no. 2, pp. 242-255, Feb. 2014.

91. C. Feng, Z. Lu, A. Jantsch, M. Zhang and Z. Xing, "**Addressing Transient and Permanent Faults in NoC With Efficient Fault-Tolerant Deflection Router,**" IEEE Transactions on Very Large Scale Integration Systems, vol. 21, no. 6, pp. 1053-1066, June 2013.

92. C. Grecu, L. Anghel, P. P. Pande, A. Ivanov and R. Saleh, "**Essential Fault-Tolerance Metrics for NoC Infrastructures**," 13th IEEE International On-Line Testing Symposium , pp. 37-42 ,July 2007.

93. C. Grecu, A. Ivanov, R. Saleh, E. S. Sogomonyan and Partha Pratim Pande, "**On-line fault detection and location for NoC interconnects**," 12th IEEE International On-Line Testing Symposium , pp. 6-7,July 2006.

94. J. Borecky, M. Kohlik, P. Kubalik and H. Kub´tov´, "**Fault Models Usability Study for On-line Tested FPGA**," 14th Euromicro Conference on Digital System Design, pp. 287-290 , Oct. 2011.

95. N. Karimi, A. Alaghi, M. Sedghi, and Z. Navabi, "**Online Network-on-Chip switch fault detection and diagnosis using functional switch faults,**" Journal of Universal Computer Science, vol. 14, no. 22, pp. 3716–3736, July 2008.

96. M. Palesi, A. Khademzadeh and A. Afzali-Kusha, "**Data Encoding Techniques for Reducing Energy Consumption in Network-on-Chip,**" IEEE Transactions on Very Large Scale Integration Systems, vol. 22, no. 3, pp. 675-685, March 2014.

97. J. Zeng, J.Y. Zhou, R.B. Lin, "**Transition inversion coding with parity check for off-chip serial transmission**", Electronics Circuits and Systems 21st IEEE International Conference , pp. 634-637, 2014.

98. D.J.r Pagliari, E. Macii, M. Poncino, "**Zero-Transition Serial Encoding for Image Sensors**", Sensors Journal IEEE, vol. 17, pp. 2563-2571, June 2017.

99. K. P. Sridhar, R. Agalya, D. Narmatha, B. Vignesh and S. Saravanan, "**Test data compression using Hamming Encoder and Decoder for system on chip (SOC) testing**," International Conference on Circuits, Power and Computing Technologies , pp. 1094-1098, Mar. 2014.

100. S. Ghosh, P. Ghosal, N. Das, S. P. Mohanty and O. Okobiah, "**Data Correlation Aware Serial Encoding for Low Switching Power On-Chip Communication**," IEEE Computer Society Annual Symposium on VLSI, pp. 124-129, Sep. 2014.

101. C. S. Behere and S. Gugulothu, "**Power reduction in network on chip links**," International Conference on Green Computing Communication and Electrical Engineering , pp. 1-4, Mar. 2014.

102. D. C. Suresh, B. Agrawal, J. Yang and W. A. Najjar, "**Tunable and Energy Efficient Bus Encoding Techniques**," IEEE Transactions on Computers, vol. 58, no. 8, pp. 1049-1062, Aug. 2009.

103. S.Wang, E. Ipek, "**Reducing data movement energy via online data clustering and encoding**", Microarchitecture 49th Annual IEEE/ACM International Symposium on, pp. 1-13, 2016.

104. H. Seol, W. Shin, J. Jang, J. Choi, J. Suh, L.S. Kim, "**Energy Efficient Data Encoding in DRAM Channels Exploiting Data Value Similarity**", Computer Architecture ACM/IEEE 43rd Annual International Symposium on, pp. 719-730, 2016.

105. X. Chen, Z. Lu, Y. Lei, Y. Wang and S. Chen, "**Multi-bit transient fault control for NoC links using 2D fault coding method**," Tenth IEEE/ACM International Symposium on Networks-on-Chip , pp. 1-8, Aug. 2016.

106. O. Sharma, A. Saini, S. Saini and A. Sharma, "**A Quadro Coding Technique to Reduce Self Transitions in VLSI Interconnects,"** IEEE International Symposium on Nanoelectronic and Information Systems pp. 98-101, 2016.

107. N. Bagherzadeh and S. E. Lee, "**Increasing the throughput of an adaptive router in network-on-chip (NoC),**" Proceedings of the 4th International Conference on Hardware/Software Codesign and System Synthesis, pp. 82-87,2006.

108. N. Jafarzadeh, M. Palesi, A. Khademzadeh and A. Afzali-Kusha, "**Data Encoding Techniques for Reducing Energy Consumption in Network-on-Chip,"** IEEE Transactions on Very Large Scale Integration Systems, vol. 22, no. 3, pp. 675-685, March 2014.

109. N. Jafarzadeh, M. Palesi, S. Eskandari, S. Hessabi and A. Afzali-Kusha, "**Low Energy yet Reliable Data Communication Scheme for Network-on-Chip,"** IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 34, no. 12, pp. 1892-1904, Dec. 2015.

110. M. Palesi, G. Ascia, F. Fazzino, V. Catania, "**Data Encoding Schemes in Networks on Chip**", Computer-Aided Design of Integrated Circuits and Systems IEEE Transactions on, vol. 30, pp. 774-786, 2011.

111. T. A. Bartic et al., "**Highly scalable network on chip for reconfigurable systems**," Proceedings of International Symposium on System-on-Chip (IEEE Cat. No.03EX748), pp. 79-82, 2003.

112. T. S. T. Mak, P. Sedcole, P. Y. K. Cheung and W. Luk, "**On-FPGA Communication Architectures and Design Factors**," International Conference on Field Programmable Logic and Applications, pp. 1-8, 2006.

113. Altera Corporate, "**Creating a system with Qsys**", (2014) [online]. Available: https://www.altera.com.

114. Xilinx,"**Vivado IP integrator user guide**,"(2013)[online].Available: https://www.xilinx.com.

115. E.S. Chung, J. C. Hoe, and K. Mai.," **CoRAM: an in-fabric memory architecture for FPGA-based computing,**" Proceedings of the 19th ACM/SIGDA international symposium on Field programmable gate arrays, pp. 97-106, 2011.

116. B. Sethuraman and R. Vemuri, "**Multi2 Router: A Novel Multi Local Port Router Architecture with Broadcast Facility for FPGA-Based Networks-on-**

Chip," International Conference on Field Programmable Logic and Applications, Madrid, pp. 1-4, 2006.

117. G. Schelle and D. Grunwald, "**Exploring FPGA network on chip implementations across various application and network loads**," *International* Conference on Field Programmable Logic and Applications, pp. 41-46, 2008.

118. M. K. Papamichael and J. C. Hoe," **CONNECT: re-examining conventional wisdom for designing nocs in the context of FPGAs**", Proceedings of the ACM/SIGDA international symposium on Field Programmable Gate Arrays, ACM, pp.-37-46, 2012.

119. R. Gindin, I. Cidon and I. Keidar, "**NoC-Based FPGA: Architecture and Routing,**" First International Symposium on Networks-on-Chip, pp. 253-264, 2007.

120. R. Pau and N. Manjikian, "**Implementation of a configurable router for embedded network-on-chip support in FPGAs**," Joint 6th International IEEE Northeast Workshop on Circuits and Systems and TAISA Conference, pp. 25-28,2008.

121. M. S. Abdelfattah and V. Betz, "**The Case for Embedded Networks on Chip on Field-Programmable Gate Arrays**," IEEE Micro, vol. 34, no. 1, pp. 80-89, Jan.-Feb. 2014.

122. M. S. Abdelfattah, A. Bitar, and V. Betz , " **Take the Highway: Design for Embedded NoCs on FPGAs**, " Proceedings of the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA '15),pp. 98 -107, ACM, New York, USA, 2015.

123. M. S. Abdelfattah and V. Betz, "**Power Analysis of Embedded NoCs on FPGAs and Comparison With Custom Buses**," IEEE Transactions on Very Large Scale Integration Systems, vol. 24, no. 1, pp. 165-177, Jan. 2016.

124. M. S. Abdelfattah, A. Bitar and V. Betz, "**Design and Applications for Embedded Networks-on-Chip on FPGAs**," IEEE Transactions on Computers, vol. 66, no. 6, pp. 1008-1021, June 1 2017.

SCALABLE AND LOW POWER NETWORK ON CHIP DESIGN FOR RECONFIGURABLE SYSTEMS

A

THESIS

SUBMITTED TO

GALGOTIAS UNIVERSITY

GREATER NOIDA

IN FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN

ELECTRONICS ENGINEERING

By

Himani Mittal

12SET503002

School of Electrical, Electronics & Communication Engineering

CHAPTER 6

CONCLUSION AND SUGGESTIONS FOR FUTURE WORK

The conclusions drawn from the results of the work presented in **Chapters Three** through **Five,** are presented in the current chapter. This Chapter also suggests the scope for future work in the area covered by this thesis.

6.1 CONCLUSION:

Design of scalable and low power network on chip for reconfigurable systems has been done in this thesis. The thesis begins with a brief introduction to the subject matter of the work carried out and the existing problems, which are treated in this thesis work. A survey of the existing literature is carried out to build a background and bring out the motivation for the work. Various researchers have designed NoCs applicable for reconfigurable systems. Various components like routers; links etc. have also been designed and tested by the researchers. A review of the studies and designs carried out in the area of network on chip for reconfigurable systems have been included in this thesis. Afterwards, fundamental concepts related to the reconfigurable systems, SoCs & NoCs are presented in Chapter Three.

Taking the support of the designs found in the literature, the problems at hand were tackled. The resulting studies and designs are presented in Chapters Four and Five. For this purpose, firstly topology selection has been done. Various topologies are listed and compared in Chapter Three. Various performance metrics like maximum end-to-end Latency, dropping probability and throughput etc. are used for the comparison purpose. It has been found that 2D mesh is suitable for the present work.

After selecting the appropriate topology, next task is to design various components of the NoC. In the present work stress has been put on the design of the routers and links. To enhance the efficiency of the NoCs, the power efficient and reconfigurable routers without the need of oversized buffers and power efficient and high performance links are designed in the present work.

To design the router and links, various software tools have been utilized. Hardware description language Verilog is used for writing the code for all components at register transfer level. All these codes have been written in synthesizable form. Model Sim 10.3 is used for the simulation purpose and Xilinx ISE design suite 13.4 is used for the synthesis at the logic gate level. RTL schematic view, gate level view, design summary, power dissipation and performance have been obtained using this software tool. Obtained results are presented in Chapters Four and Five.

To design the power efficient and smart reconfigurable routers, several versions of it are designed and tested. These versions are

a. RRIE (Reconfigurable Router to Improve Efficiency).

b. HRRLPHP (Heterogeneous Reconfigurable Router for Low Power and High Performance).

c. EDRRHT (Eight Directional Reconfigurable Router for High Throughput).

d. SRRODFB (Smart Reconfigurable Router for Online Detection of Faulty Blocks).

In the proposed RRIE, NoC efficiency has been compared as a function of the possibility to reconfigure the buffer size according to the requirements of each channel of the router at run time. It is done without the need to oversize the buffers for the guaranteed performance. It has low area, consumes less power and provides high performance. In this approach it is possible to dynamically configure different buffer depth for the channel. This router has FIFOs, channel flow/control logic and a cross bar. It has been found that the total power dissipated by this router architecture is around 15 milliwatt which is less compared to the previous designs available in the literature. Moreover, this router obtains the same performance as that of homogeneous router but with a buffer depth which is 64% smaller. Also, it is possible to reconfigurable the router in accordance with the application in the modified architecture. It obtains similar performance even when the application changes radically.

The problem with the RRIE is that, in it when a channel starting to accept packets from its neighbour (left or right) it does not take any packet further from that channel. In other words we can say that it is not supporting heterogeneous data. Also, its area is found to be slightly high. These drawbacks are improved in the second reconfigurable

router namely HRRLPHP. In this router even if a channel is taking data from its neighbouring channel it will continue to take data from its neighbouring channel also. Hence, it supports heterogeneous data. To reduce the area, two tag bits to the packets in packet switching or flits (in wormhole switching) has been added. These bits are added before storing the packets in the FIFO to indicate the channel to which this packet or flit was input. Tag bits are '00' when the packets are directly coming to the concerned channel, '01' when packets coming from the right neighbour and '10' when the packets are coming from the left neighbour. These tag bits are also used for indicating the direction of output from the crossbar. For an 8 bit packet, the first two bits are used to indicate the address of the next router. Control circuitry is implemented in the present router by including features like acknowledge for writing, acknowledge for reading, request and grant signals for writing in another channel's FIFO. The modified channels, crossbar and FIFO are simulated and synthesized. Obtained results show that area is significantly reduced and is only 53 mm^2 while the power dissipation is found to only 0.020 W. The drawback of this router is that throughput is degraded marginally (3%). Although it is tolerable and insignificant for many applications, we have tried to remove this drawback in our next router namely EDRRHT. Another problem with HRRLPHP is that it does not use the semi-buffer concept to increase efficiency. These two things are added in our third router (EDRRHT). Four more directions (NE, NW, SE & SW) are added so that router is now able to support eight directions. It also supports buffer less storage concept to store the data of four newly added directions. Data of newly added directions share FIFOs from their neighbours to store its data. This way, this architecture requires less area for implementation. The newly designed architecture provides more routes to router to route the data thus it is helpful in decreasing critical path length and help in increasing the performance of the NoC. Also, we have used rectilinear Stennier tree path to shorten the data path of the router. We have used a circular FIFO in place of normal fall through FIFO. Also, several changes have been made in control circuitry to improve the throughout. Compared to the second router it has four additional arbiters (NW, NE, SW and SE) which support buffer less concept and is used to share data from neighbours depending on the priority which is further dependent on the priority table. Obtained results show a power dissipation of 0.045 W, which is higher

than the second router because of the addition of four more semi buffers. Area estimated is found to be 85 mm^2.

The EDRRHT is not able to handle the spotting of the faulty blocks of NoC. Also, it is not able to recognize and differentiate between permanent and transient errors and the way to handle them. Another drawback is that it lacks the suitable algorithm to help in online detection of faults while preserving its throughput, data packet latency and the network load. These drawbacks have been removed in our fourth router namely SRRODFD. This is a smart reconfigurable router for online detection of faulty blocks. It can distinguish between permanent and transient errors and is able to accurately localise the position of the faulty blocks (data bus, input port, output port) in the NoCs. It is also able to preserve the throughput, the network load and the data packet latency. Buffers are removed. In place of buffers FIFO is used to achieve better performance. Hence it is able to have low area, low power dissipation and high performance compared to the other designed routers.

A set of MUX gating and data encoding and decoding schemes aimed at reducing the power dissipated by the links of a NoC has also been designed and simulated in this work. The purpose behind the proposed schemes is to minimize the switching activity as well as the coupling switching activity. These activities are mainly responsible for link power dissipation in the deep sub-micrometer technology regime. The proposed encoding schemes are agnostic with respect to the underlying NoC architecture in the sense that their application does not require any modification neither in the routers nor in the links. The encoders implementing the proposed schemes have been assessed in terms of power dissipation and silicon area. The decoders are implemented to perform the reverse operation of encoders and hence to recover the data.

It is obvious that the reconfigurable hardware solutions are based on computer architectures able to adapt their behaviour in response to several different inputs. Also, reconfigurable techniques are applied to all levels of the systems like buffers, routers, and links. The present thesis is an effort to implement this principle. Furthermore, scalability, power efficiency, low area and high performance are the other objectives to be fulfilled by modern embedded NoCs for reconfigurable

systems. By applying all the above mentioned concepts, the present thesis is able to fulfil its objective.

6.2 SUGGESTIONS FOR FUTURE WORK:

The solutions presented and architecture provided in this work is neither optimal nor complete. So there is always lot of scope for modifications.

1. Eun Lee and Bagherzadeh [107] presented the use of various clocks while sending flits in the NoC. Body flits operate faster than head flits. So accordingly, as FIFOs work faster than the route decision, it is possible to use various clocks to feed body flits or head flits. While the head flit is analyzed to tell the route path, body flits can continue advancing along the reserved path already established, improving the performance of the router.

Thus, with the help of different clocks it could be subjugated in the current work so as to obtain the better delays and latency factor.

2. Ahmad et al. [66] discussed a network router designing with a bus like interface. A built-in wrapper is utilized, and because of this any bus compatible component could be integrated into the NoC architecture. The interface of this router would be a simple bus. The aim of this discussion was to reduce the design time and to ease integration. When the network has a channel that requires high bandwidth, the NoC changes the switching, obtaining a dedicated path between the IPs. This discussion could also be done as advancement to the current project module. In turn it will reduce the paradigm shift totally but up to certain level only, hence making the model to easily associate with the present SoC models.

3. The disadvantage of the previously discussed SoC proposals was that all of them must be used at design time, generally with a static approach, hence causing problems of scalability, whenever the NoC is used for a new application in the same platform.

Thus, updating of product, customization of an MPSOC or a different market, using the same components, can be considered paramount for futuristic approach; although it seems that it results up being costlier than projected.

4. Link: Data packets travels among various cores in a NoC by the use of either a serial or a parallel link. Parallel links use a buffer-based architecture and can be operated at a relatively lower clock rate in order to reduce power dissipation. These parallel links come up with high silicon cost due to inter-wire spacing, shielding and repeaters. It can be optimised to a certain limit by employing multiple metal layers. If

seeing the other side, serial links allow savings in wire area, minimization in signal interference and noise, and limits the need for having buffers. Serial links have the advantages of a simpler layout and simpler timing verification. Serial links faces SI (Intersymbol Interference) between successive signals while operating at high clock rates. Encoding along with asynchronous communication protocols can take care of these drawbacks.

6. Optimization through Links Interconnect: Network on chip communication is based on modules connected via a network of routers with links between the routers that comprise of long interconnects. So minimizing interconnects in order to achieve the required system performance is required. By inserting repeaters global wires timing optimization is obtained. This will result in a makeable increase in cost, area, and power consumption. Present research states that in the near future, inverters operating as repeaters will occupy a large portion of chip resources. Thus, there is a need for optimizing power on the NoC. Encoding is an effective way of reducing dynamic power consumption. Innovative ways will have to be explored to optimize the power consumed by the on-chip repeaters.