2012

Design Space Exploration of FPGA-Based NoC Routers

Abdelrazag Imbewa
University of Windsor

Design Space Exploration of FPGA-Based NoC Routers

By

Abdelrazag Imbewa

A Thesis

Submitted to the Faculty of Graduate Studies

through Electrical and Computer Engineering

in Partial Fulfillment of the Requirements for the

Degree of Master of Applied Science at the

University of Windsor

Windsor, Ontario, Canada

2012

Copyright

Design Space Exploration of FPGA-Based NoC Routers

by

Abdelrazag Imbewa

APPROVED BY:

_____

Dr. Z. Kobti, Outside Department Reader
School of Computer Science

_____

Dr. K. Tepe, Departmental Reader
Department of Electrical & Computer Engineering

_____

Dr. M. Khalid, Advisor
Department of Electrical & Computer Engineering

_____

Dr. E. Abdel-Raheem, Chair of Defense
Department of Electrical & Computer Engineering

April 17, 2012

Author's Declaration of Originality

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

Abstract

Currently, FPGAs serve as Field-Programmable-Systems-on-Chip (FPSoCs) and are widely used to implement computationally intensive applications. As the number of components in FPSoCs increases, the interconnect schemes based on Network-on-Chip (NoC) approach are increasingly used. Routers greatly impact the performance and cost of NoCs. In this thesis, we explore the design space of FPGA-based NoC routers. We implement three types of packet switched NoC routers on a Stratix II FPGA using parameterized VHDL models. To reduce the area and increase the speed, we use novel techniques. Buffer size is decreased by minimizing the number of control fields in a packet. Both edges of the clock are utilized, and credit based flow control is used to accelerate the router. The proposed routers were evaluated based on area, frequency, and zero load latency. Synthesis results and zero load latency evaluations show that they are significantly superior to widely referenced, previously proposed routers.

Acknowledgements

After thanking Allah, my almighty god, I would like to express my gratitude and appreciation to my supervisor Dr. M. Khalid for his guidance and support, which helped me complete this work. He is the one who introduced me to this fantastic field that was very suitable to my interests and encouraged my creativeness. The courses I took with him, working with him as a graduate assistant throughout the period of my Masters program at the University of Windsor, as well as his thesis advice, helped me develop the skills needed to conduct my research.

My appreciation also goes to the respected thesis committee members Dr. Kemal Tepe and Dr. Ziad Kobti for their advice and help. I would like to thank my friends and colleagues for their company and interesting discussions, especially Mohan, Matt, Mike, Ali, Krunal, Shoaleh and Vinod. My gratitude is also extended to Mrs. Louise Pillon and all the ELIP members as well as all the staff in the ECE department, especially Andria Ballo.

Even though she is away from me, my mother's prayers and love acted as a shield that eased the hard times I experienced. I am grateful to my wife, Hana, for all the sacrifices she made. Without her support I would not have reached this milestone in my life. I would like to thank my brothers, who live far away, for their support and encouragement, especially Shaker, who has been very patient and cooperative.

Lastly, my thanks to my children: Qabas, Noufl, Abdulazeez, and Mohamad - sorry for not spending as much time with you as I should have.

Table of Contents

Author's Declaration of Originality iv
Abstract v
Acknowledgements vi
List of Figures x
List of Tables xii
List of Abbreviations xiii
1 Chapter 1 1
Introduction 1
1.1 Thesis Goals 2
1.1.1 General Objectives 3
1.1.2 Specific Issues and Research Approach 3
1.2 Thesis Outlines 5
2 Chapter 2 6
Background and Related Research Work 6
2.1 On-Chip Interconnect Architecture 6
2.2 Network on Chip (NoC) 8
 Deadlock: 9
 Livelock: 9
 Starvation 10
2.2.1 Main NoCs' Parameters 10
2.2.1.1 NoC Topology 10
2.2.1.2 Switching Techniques 11
2.2.1.3 Flow Control 12
2.2.1.4 Buffer Size 13
2.2.1.5 Link Width 13
2.2.1.6 Routing Algorithm 14

2.2.1.7 Arbitration 14
2.2.2 NoC Evaluation Criterions 15
2.2.2.1 Cost Metrics 15
2.2.2.2 Performance Metrics 15
2.3 Field Programmable Gate Arrays (FPGAs) 16
2.3.1 FPGA Advantages and Disadvantages 17
 Advantages: 17
 Disadvantages: 17
2.3.2 FPGA-based NoCs 18
2.3.3 Capacity of a Logic Unit in FPGAs 18
2.3.4 Choosing FPGA Target Device 19
2.3.4.1 Stratix II FPGA Family Logic Structure 19
2.3.4.2 Virtex-based FPGA Families Logic Structure 20
2.3.4.3 Stratix II VS. Virtex-based Comparison 20
2.4 Related Research Work 24
2.5 Summary 28
3 Chapter 3 29
NoC-Router Design and Architecture 29
3.1 Setting Router Design Parameters 29
3.2 Router General Overview 30
3.3 Router Micro-Architecture 32
3.3.1 Router Versions Variances 32
 Dual Clock Mechanism 33
3.3.2 Packet Format 33
3.3.2.1 VCTR Version Packet Format 35
3.3.2.2 WHR Versions Packet Format 35
3.3.3 Input Port (FIFO Buffer) 36
3.3.4 Direction Decoder 38
3.3.5 Routing & Arbitration Unit (Arbiter) 38
3.3.6 Switching Box (Switch) 40
3.4 Design Improvement Techniques 42

3.5 Functionality Verification 42
3.5.1 VCTR Simulation 43
3.5.2 WHR-1clk Simulation 44
3.5.3 WHR-2clk Simulation 44
3.6 Summary 45
4 Chapter 4 46
Evaluation and Results 46
4.1 Assessment Approach 46
4.2 Results and Analysis 47
4.2.1 Synthesis Results 47
4.2.1.1 Area 48
4.2.1.2 Frequency 50
4.2.1.3 Power Consumption 53
4.2.2 Performance Calculation and Analysis 54
4.2.3 Comparison with Previous Work 56
4.3 Summary 59
5 Chapter 5 60
Conclusion and Future Research Directions 60
Appendix 64
Full Reports of Synthesis Results 64
References 78
VITA AUCTORIS 81

List of Figures

Figure 2.1 On-Chip Interconnect (a) Based on Single Bus, (b) Based on Hierarchy of Buses [7] 7

Figure 2.2 Point-to-Point Interconnect (a) With Few Nodes, (b) With Increased Number of Nodes 7

Figure 2.3 a Sample 4X4 Mesh Network on Chip [10] 8

Figure 2.4 Typical NoC Router Design [11] 9

Figure 2.5 Examples of Network Topologies [13] 10

Figure 2.6 Comparison between switching techniques: (1) Store-and-Forward Switching, (2) Circuit Switching, and (3) Wormhole Switching [15]. 13

Figure 2.7 FPGA General Architecture [17]..... 17

Figure 2.8 Stratix II ALM [20] 19

Figure 2.9 (Virtex-II Pro / Virtex-4) Slice [20] 20

Figure 2.10 Stratix II VS. Virtex-4 Power Consumption Example from Xilinx Source [21]..... 21

Figure 3.1 Router Block Diagram..... 31

Figure 3.2 Router Design Hierarchy..... 32

Figure 3.3 Time-space diagram for sending a 5-flit packet over 4 channels [5]. 34

Figure 3.4 Packet Format..... 36

Figure 3.5 The Architecture of FIFO Buffer 37

Figure 3.6 Direction Decoder Architecture 38

Figure 3.7 Arbiter Architecture (a) for WHR versions (b) for VCTR version 39

Figure 3.8 Switch Architecture 41

Figure 3.9 Simultaneous Communications in NoC Router 43

Figure 3.10 VCTR Functional Simulation 44

Figure 3.11 WHR-1clk Functional Simulation 44

Figure 3.12 WHR-2clk Functional Simulation 45

Figure 4.1 Stand-Alone Router Area for the Three Versions 48

Figure 4.2 3X3 Mesh NoC Area for the Three Versions 49

Figure 4.3 Average Router Area in 3X3 Mesh NoC for the Three Versions 50

Figure 4.4 Stand-Alone Router Frequencies for the Three Versions 51

Figure 4.5 3X3 Mesh NoC Frequency for the Three Versions 52

Figure 4.6 Stand-Alone Router Power Consumption for the Three Versions 52

Figure 4.7 3X3 Mesh NoC Power Consumption for the Three Versions 53

Figure 4.8 Zero Load Latency Case Study 56

Figure 4.9 Performance and Area Comparison between VCTR and MoCReS Summary 57

Figure 4.10 Performance and Area Comparison with some Previous Work Summary ... 59

List of Tables

Table 2.1 Stratix II VS. Virtex-4 Device Utilization Example based on Xilinx Source [21] 21

Table 2.2 Stratix II vs. Virtex-4 Implementation Comparison for Various Functions [20]. 22

Table 2.3 Design Building Block Performance & Logic Utilization Comparison between Stratix II &Virtex-4 FPGAs according to Altera [20]. 22

Table 2.4 Normalized Relative Logic Capacity based on Altera White Paper [18] 23

Table 2.5 Stratix II vs. Virtex-II Pro Equivalent Device Matchup from Altera [18] 23

Table 4.1 a Performance and Area Comparison with MoCReS Stand-Alone Routers 57

Table 4.2 a Performance and Area Comparison with some Previous Work 58

List of Abbreviations

Abbreviation	**Definition**
ALM	Adaptive Logic Module
ALUT	Adaptive Look-up Tables
ASIC	Application Specific Integrated Chip
B	Body flit
BW	Bandwidth
CAD	Computer Aided Design
cc	common clock
CS	Circuit Switched
DSP	Digital Signal Processing
FIFO	First In / First Out
Flit	Flow Control Unit
FPGA	Field Programmable Gate Array
FPSoC	Field Programmable System on Chip
FSM	Finite State Machine
H	Head flit
HDL	Hardware Description Language
IC	Integrated Circuit
IP	Intellectual Property
LC	Logic Cell
LE	Logic Element
LSB	Least Significant Bit
LUT	Look-Up Table
Mbps	Megabit per second
mc	multiple clocks
MHz	Megahertz

MLPR	Multi-Local Port Router
MSB	Most Significant Bit
MW	Milli-Watt
NoC	Network-on-Chip
ns	nanosecond
Op_A	Optimized for Area
Op_S	Optimized for Speed
P_s	Packet size
PS	Packet Switched
ps	picosecond
QoS	Quality of Service
RAM	Random Access Memory
R_c	Router count
R_d	Router delay
SAF	Store and Forward
SoC	System-on-Chip
SR	Shift Register
T_{bc}	Best Case Latency
TDM	Time Division Multiplexing
TP	Throughput
TSMC™	Taiwan Semiconductor Manufacturing Company Trademark
VCT	Virtual Cut Through
VCTR	Virtual Cut Through Router
VHDL	Very large scale integrated circuit Hardware Description Language
WH	Wormhole
WHR-1clk	Wormhole Router with One Clock
WHR-2clk	Wormhole Router with Two Clocks

1 Chapter 1 Introduction

It would not be surprising nowadays if someone bought a new electronic product and he or she knew, after a short time, that the next developed version of that item had been released by the same manufacturer. Also, a competitive product from different producer could be available for lower cost and/or better performance. People are demanding devices embedded with powerful digital equipment; yet, the size and cost have to be put at minimum, if customer satisfaction is desired. Thus, explorations and developments of digital systems are endless while the technological aspects permit.

To take advantage of the fast growth of chip density, and their capabilities of configuring more logic, Systems on Chip (SoCs) emerged as an attractive approach for many modern applications that need extensive processing and/or storage requirements, such as multimedia applications, while maintaining the least possible dimensions. The implementation of larger and more complex embedded systems in a single chip is now feasible - hence, the necessity of integrating a group of Intellectual Properties (IPs), to increase the productivity of the design and decrease the cost through the reuse of the predesigned systems. SoCs should be able to hold numerous hardware and/or software modules, such as processors, memories, peripherals, controllers, Digital Signal Processors (DSPs), and other custom logic blocks. Achieving the design goals of multicore FPSoCs is deeply influenced by the communication infrastructure needed to establish the information exchange between IPs, which in turn is affected by the efficiency of the design as well as the implementation environment.

The nature of the on-Chip interconnect plays a vital role in the global performance and cost of the SoC. Network-on-Chip (NoC) architecture has been proposed as a high performance, scalable and power efficient alternative to the bus based architecture [1],

[2]. It solves the scalability problem by supporting multiple concurrent connections between IP cores, and allows reuse of pre-tested IP cores to minimize design and verification times, all the while maintaining a low area-overhead [3]. NoC separates the concerns of communication and computing, and is expected to be ideally suited to simplify system complexity and deal with declining system productivity.

However, with the acceleration of today's applications and their associated SoCs, NoCs still suffer from bottlenecks like: latency, bandwidth violation, increased area, power consumption, and congestion. Performance and cost requirements will vary (usually they compete each other) depending on the application. All in all, adjusting the parameters of the NoC router (as a dominant component in the communication architecture) will facilitate the exploration of design space to come up with the optimum NoC characteristics, in addition to providing a wider implementation range, based on the necessities of the application. The other important factor that impacts the configuration of SoCs and their interconnect mechanisms, in terms of cost and performance, is the implementation medium. Field Programmable Gate Arrays (FPGAs) are increasingly replacing the Application Specific Integrated Circuits (ASICs) in many contemporary applications because of FPGAs' advantages such as: 1) low development cost and short time to market, 2) ease of upgrading, 3) suitability for research purposes, given that they provide fast design cycle and immediate results, and 4) lack of manufacturing operations (no IC fabrication is involved). On the other hand, they are generally slower than ASIC, have limited area, and need more power. Even though recent FPGAs are being designed such as to increase their capabilities and enhance their shortcomings, it is still challenging to satisfy some applications' specifications.

1.1 Thesis Goals

The FPGA-based NoC is an active research field. Many design aspects still need more exploration, and the existing proposed solutions to the problems associated with implementing real systems require additional enhancements.

1.1.1 General Objectives

- Developing a deep understanding of FPGA-based NoCs systems' architectures and parameters to provide further exploration of their design space.
- Helping designers in determining the appropriate tradeoffs between the design parameters, based on the desired application with as much flexibility as possible.
- Introducing some solutions to existing NoC bottlenecks that cause huge limitations, especially with the growing demand of applications.

1.1.2 Specific Issues and Research Approach

An FPGA based on-chip network has a unique set of design goals that includes satisfying bandwidth requirements with minimum (limited) resource availability [4]. Logic utilization (area) in FPGAs is the main obstacle before designers who aim to increase the speed of their system - not only because it is limited and most of FPGA's space has to be kept for IPs, which will be configured along with the NoC architecture, but also because power consumption rises with increasing the area.

Utilizing the CAD tools and taking advantage of the fast prototyping nature of FPGAs, we will try to implement and evaluate some strategies for minimizing the area of NoC's routers. As well, other tactics during the design will be examined to boost the speed and shrink the latency of the NoC.

The size of the buffer is proportional to flit's width and to the required buffering positions during packets flow. Adding extra control fields to the flits is one of the disadvantages of network interconnections. As well as the extra buffer size requirements, they increase the data transported over the network wasting more time and wires. Thus, the area and latency increase. Further, most researchers, in their performance evaluations, consider the delivering of these control data as a part of the actual throughput. In contrast, minimizing the number and the size of the control fields will give a more accurate picture of the performance of the NoC and reduce the required size of the NoC to handle the same amount of actual data. Similarly, making the buffer depth as small as possible will dramatically reduce the router area; however, there is a borderline to that reduction, if we

want to guarantee the minimum latency of delivering the packet. Our objective is to find a less area-consuming logic to replace those control fields that consume a larger area, and to compact the buffer depth without adding extra latency.

Operating frequency and routing delay play a vital role in the overall speed of the NoC system. Achieving a high clock rate will obviously speed up the operations performed within a specific time; however, the slowest part of the router determines the frequency of the whole router. In the packet switched routers, which are mostly used in NoC routers, routing the head flit requires more time than forwarding the body flit. Thus, it is possible to forward the body flits with a different faster clock rate than the head flit. To the best of our knowledge, the idea of a dual-clock flow control mechanism has not been used for FPGA-based NoCs. Lee et al. [5] are the only researchers who proposed this idea for their fully adaptive router in TSMC^{TM} 90nm technology and the results were obtained by simulation. The other important factor that affects the speed is the number of clock cycles needed to complete the routing of the head flit and forwarding the body flit. Decreasing these numbers substantially accelerates the routing decision and the delivery of packet's flits, resulting in shortening the lower bound of the average latency. In short, we are working on diminishing the clock's period time where it is possible to do so, but more crucially, lessening the number of these periods that is obligatory for each router operation.

In order to achieve the specified goals of this research, we introduce the implementation of a parameterized VHDL models of a NoC-Routers on Stratix II (one of the families of FPGA devices from Altera [6]) to address the design space exploration for FPGA-based NoCs' Routers. We focused in our approach on minimizing the area of the router because area is at premium on an FPGA, without sacrificing the speed. Our router in general is a 5-ports packet switched router with deterministic XY-Routing algorithm and dynamic arbitration scheme (Round-Robin). The depth of input buffers, located at the input ports, and the flit size (channel width), are parameterized by means of VHDL generics. Three versions of our router were developed: Virtual Cut Through Router (VCTR), Wormhole Router with one clock (WHR-1clk), and dual clock Wormhole Router (WHR-2clk). We presented the general architecture of the router as well as the architectural and functional

differences between the three versions. The strategies that followed to decrease the area and increasing the speed are explained. To demonstrate the feasibility and functionality of our design, it was verified via Altera Quartus II Simulator Tool, and synthesized, placed, and routed targeting a widely held Stratix II FPGA family, device EP2S15F672I4 from Altera, by means of the Quartus II Synthesis Tool. Also, a 3X3 Mesh NoCs using 9 models of each router were configured on the same device and using the same software. Comparisons of the synthesis results before and after NoC configuration, between the three versions, and with number of widely cited previously proposed FPGA-based NoC routers, were conducted. The results show the substantial gain for our routers, denoted by their smaller area and mostly faster frequency. Furthermore, the calculations of best case (zero load) latency, that represents the lower bound of the average latency, prove that our design is much faster than others, at least when the NoC works without contention.

1.2 Thesis Outlines

The general approach of this thesis is to explore the design space of NoC routers, with more focus on FPGA-based NoCs, through the design, enhancements, and evaluation. The thesis is organized as follows: Chapter 2 introduces a background on NoC and FPGA aspects as well as a review of related research work. In Chapter 3, the designed routers' architectures are described and functionally verified along with the techniques used to augment the performance and reduce the area. The evaluation methodology and attained results are presented in Chapter 4 including the comparison with some previous work results. Finally, Chapter 5 concludes all the contributions made in this thesis, and outlines future research directions.

2 Chapter 2
Background and Related Research Work

This chapter broaches a background about some aspects of Networks on Chip (NoCs) and Field Programmable Gate Arrays (FPGAs) that are important to understand the key factors of exploring the design space for FPGA-based NoCs. Starting with an idea about On-Chip communication development, the following sections explain the NoC building components, parameters, which can be used also to classify NoC's routers. Then, a discussion about NoCs' evaluation metrics is provided. This is followed by FPGAs overview that contains the advantages and the disadvantages, and choosing the appropriate FPGA device including a comparison between two popular FPGA families. The chapter concludes with an ample review of available previously proposed FPGA-based NoCs research.

2.1 On-Chip Interconnect Architecture

The communication between SoC's cores can be established traditionally via shared buses, dedicated point to point, or a mix of them. The first can be as easy as a shared single bus such as the one shown in Figure 2.1 (a), which is suitable for small systems, or more complex hierarchal multi buses, using sophisticated protocols and bridges, to serve larger systems as it is displayed in Figure 2.1 (b). Yet, many disadvantages are associated with bus interconnections. Noticeably, the bandwidth is limited, concurrent communications are not possible, and the scalability is restricted and causes speed degradation. Although point to point approach provides the optimum bandwidth and latency, it suffers from some drawbacks like; routing difficulties and the rapid escalation in the number of links with increasing the number of IPs. Figure 2.2 (a) and (b) illustrate the difference between using a smaller or greater number of nodes with point to point protocol.

As the number of (IPs) grows, the use of these techniques becomes a bottleneck because of scalability complications and efficiency. The alternative proposed approach was the Network on Chip (NoC).

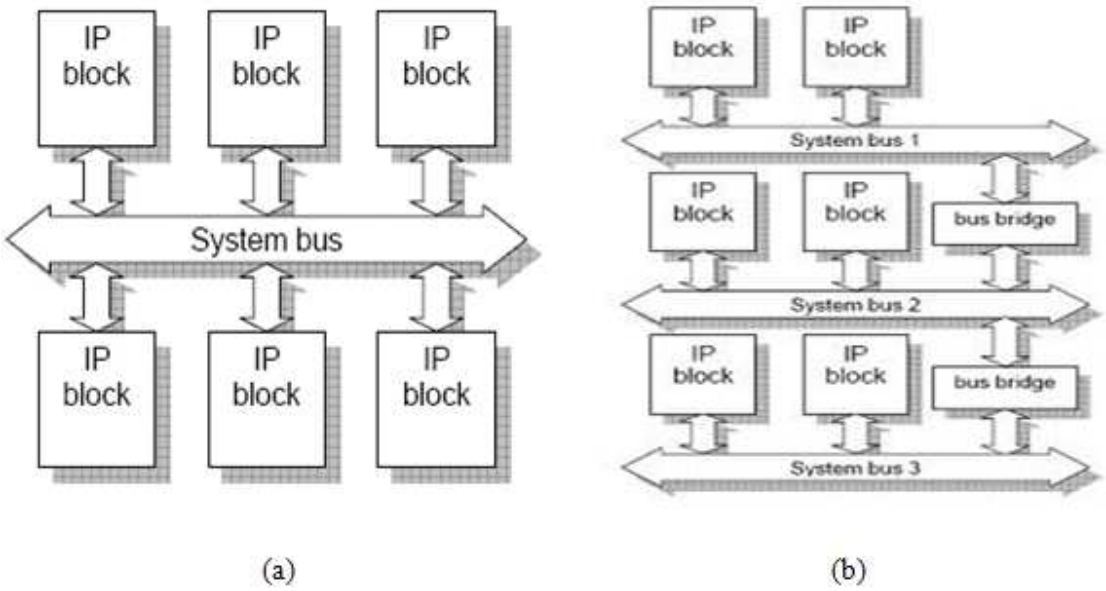


Figure 2.1 On-Chip Interconnect (a) Based on Single Bus, (b) Based on Hierarchy of Buses [7]

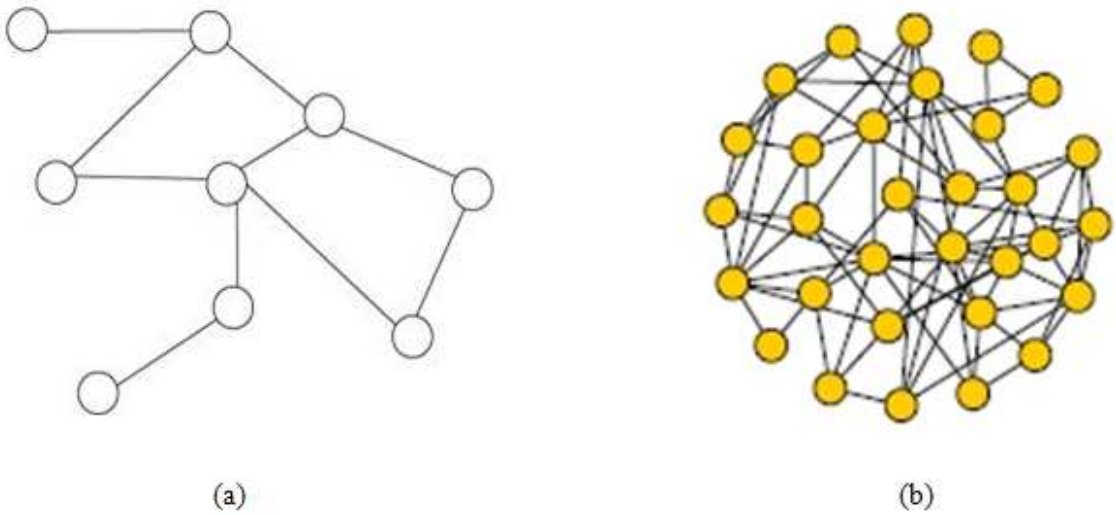


Figure 2.2 Point-to-Point Interconnect (a) With Few Nodes, (b) With Increased Number of Nodes

2.2 Network on Chip (NoC)

In NoCs, the communications among IPs inside the chip are mainly formed through micro routers that receive and forward the messages from and to adjacent IPs/routers, as well as other required components; such as the Network Adapters that regulate the interface (usually packetizing and de-packetizing of the message) between the core and the router and vice versa, and links that are used to connect routers to adapters or other routers. Fig 2.3 shows an example of NoC. Also, a typical NoC router design is displayed in Figure 2.4. The idea of data-routing network is principally replicated from the conventional off-chip computer networks; however, some differences can be mentioned here. For instance, NoC links are very short and inexpensive compared to the ones used in off-chip networks. Yet, the area and power consumption of NoCs are limited and latency is critical, whereas computer networks allow more flexibility to these aspects. The advantages of using NoC approach over the use of bus based on-chip communications, especially with FPGA-based SoCs, are widely studied and verified by many researchers such as in [1], [8], [9], [10]. NoCs solve the scalability problem, increase the available bandwidth, lessen latency, support multiple concurrent connections between IP cores, allow for pre-tested design reuse as a way for designers to keep pace with the technological development, and maintain a low area-overhead and power outlay.

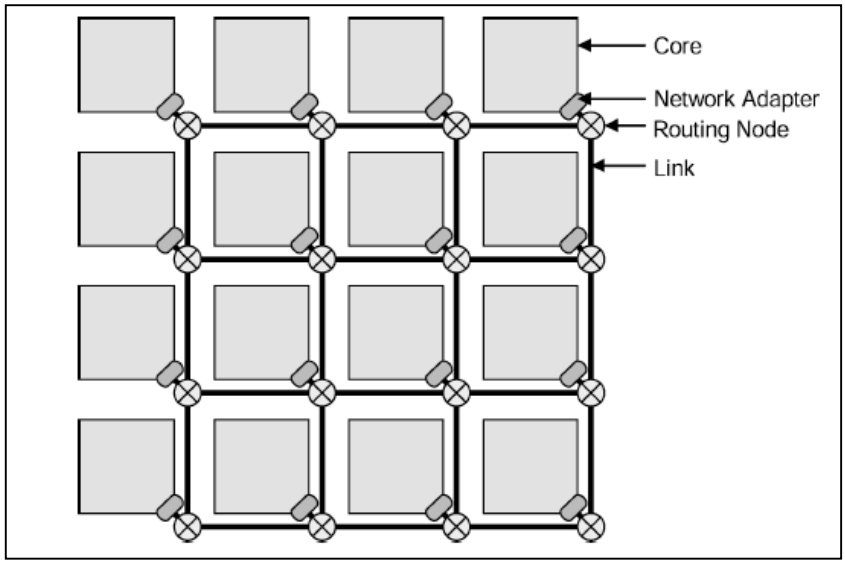


Figure 2.3 a Sample 4X4 Mesh Network on Chip [10]

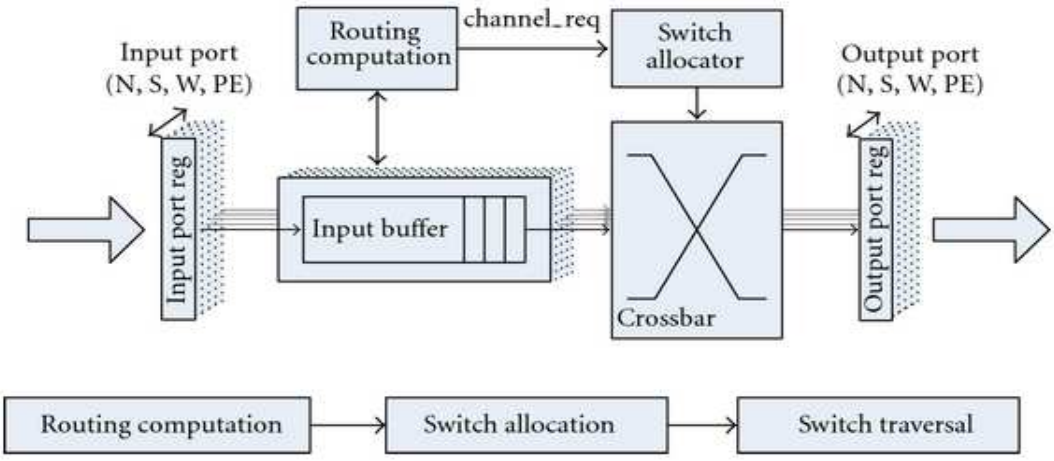


Figure 2.4 Typical NoC Router Design [11]

However, there are some NoCs issues that researchers are still trying to enhance, especially with increasing SoCs requirements and specifications. These issues include latency, consumed area and power, quality of services, and flexibility. A good NoC has to be able to meet the application requirements.

Before discussing the NoCs parameters and metrics, some terminologies that will be mentioned in the following context are defined here. They represent some of NoC's glitches and are occasionally considered as quality of service criterions.

- **Deadlock:** Deadlock occurs when network resources (e.g., link bandwidth or buffer space) are suspended waiting for each other to be released, that is, where one path is blocked leading to another being blocked in a cyclic fashion [12]. Virtual Channels, which are the sharing of a physical channel by several logically separate channels by means of Time Division Multiplexing (TDM) with individual and independent buffer queues, can break the loop, as can proper routing and placement to avoid a deadlock situation [10].
- **Livelock:** It arises when packets are not progressing toward their destination because they are routed in a cyclic track around their destination. Using the shortest route (minimal path routing) can prevent this problem.

- **Starvation**: It is the case when packets with lower routing priority are prevented from using the network resources to reach their destination because higher priority packets reserve these resources continuously. A fair routing scheme that guarantees resource allocation to all packets, can avoid this scenario.

2.2.1 Main NoCs' Parameters

The design of a network on a chip to produce the desired features of the system is guided by many parameters. These parameters are correlated with the architecture and communication protocols of the NoC. They not only characterize the performance and cost of the communication aspect of the SoC and the overall appearance of the system, but also, are used in many occasions to classify the NoCs and their central components (the Routers). We tried in this section to cover the main NoC parameters and design options.

2.2.1.1 NoC Topology

The distribution of network nodes and the schemes of the links between these nodes are known as the topology. It is one of the parameters that affects the overall area, power consumption, and speed of the NoC. In terms of regular and scalable topologies, there are many types with different dimensions. The most popular topologies used with NoCs are n-dimensional Mesh, Tours, and Ring. Figure 2.5 depicts some examples of practical topologies.

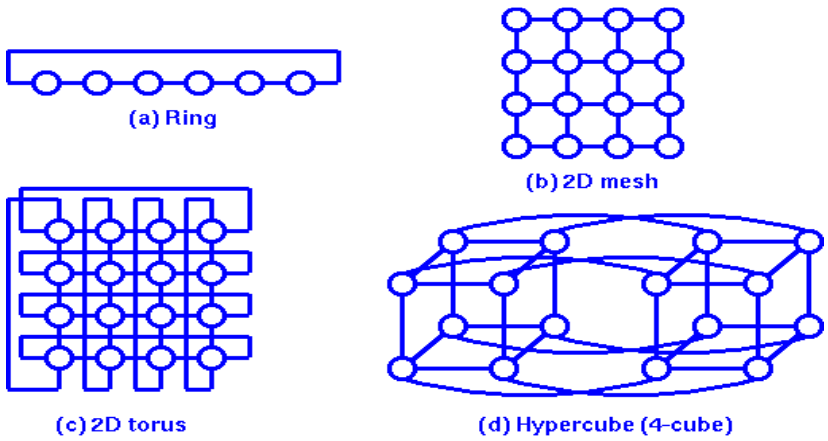


Figure 2.5 Examples of Network Topologies [13]

Not much research has been done to compare the effects of implementing FPGA-based NoCs with different topologies; however, 2-dimentional Mesh is the most dominant approach among the proposed work, for its physical mapping simplicity and routing patterns straightforwardness, as well as cost predictability.

2.2.1.2 Switching Techniques

The way that messages pass through the network from upstream to downstream is known as switching technique. Network latency is affected by the switching technique used. Circuit switching (CS) and packet switching (PS) are the recognized methods of data transmitting among NoC researchers. Circuit switched networks reserve a physical path before transmitting the data packets, while packet switched networks transmit the packets without reserving the entire path [14]. CS technique requires a setup time to build and tear down connections, and its channel reservation nature often leads to idle times and causes unreliable blocking. The only upside to this method is its ability to provide guaranteed bandwidth during connection times [3]. On the other hand, PS routers can send the message packet by packet any time as smaller fractions called flits. The packet will be directed to its destination based on the routing information included in it, interleaving with other packets. PS kind can be further classified to three modes or sub-techniques as follows:

- **Store-And-Forward (SAF)**

This scheme buffers the entire packet inside the router and forwards it only when there is enough space available in the receiving node to accommodate the whole packet. While this simplifies the design, it makes the latency proportional to packet size and requires a large amount of buffer space at each node. Hence, it increases the overall area.

- **Wormhole (WH)**

Wormhole mode mediates between packet switched flow control and circuit switched concepts. Buffering requirements in WH is very low; hence it supports the design of NoC routers with smaller area. The first flit (head flit) carries the routing information of packet. It is routed to the next hop and all remaining body flits will follow the same path

in a pipeline fashion. As soon as the flit is received at a router, it can be transmitted. Therefore, latency is proportional to the flit size. However, blocking the head flit leads to channel reservation along the path because the whole packet will be blocked in the places where its flits span. This makes WH mode more prone to deadlock than other techniques.

- **Virtual Cut Through (VCT)**

Virtual Cut Through (VCT) strategy is a mix between store-and-forward and wormhole strategies. It supports higher throughput than wormhole routing by efficiently releasing the upstream buffers during blockages [4]. Buffering requirements is the same as SAF mode, but the packet flows as in a WH mode while there is no blocking. The action taken in the case of blocking, however, is unlike WH. While the head flit of the packet stops, the body flits continue to move in the path toward the blocked node until they eventually aggregate in its buffer if the head is still blocked. This approach solves the problem of stalling the channels existing in WH; however, nodes must be able to buffer the entire packet. Thus, the router area is much bigger than WH and also the packet size should be limited and predetermined.

Figure 2.6 explains the differences between switching techniques in terms of latency in contention-free network. VCT is not shown because it is similar to WH in this case. The transmission events over the time from the source “S” through the intermediate nodes (I1, I2, and I3) toward the destination are shown in the Figure.

2.2.1.3 Flow Control

Flow control describes the behavior of the network to allocate its resources to message packets. These resources include buffers if any, channels, ports, and control logic. Depending on the employed switching technique, the amount of allocated resources and timing will vary. For example, circuit switched technique does not require buffers while packet switched does. The most popular scenarios of flow control are handshaking and credit based protocols.

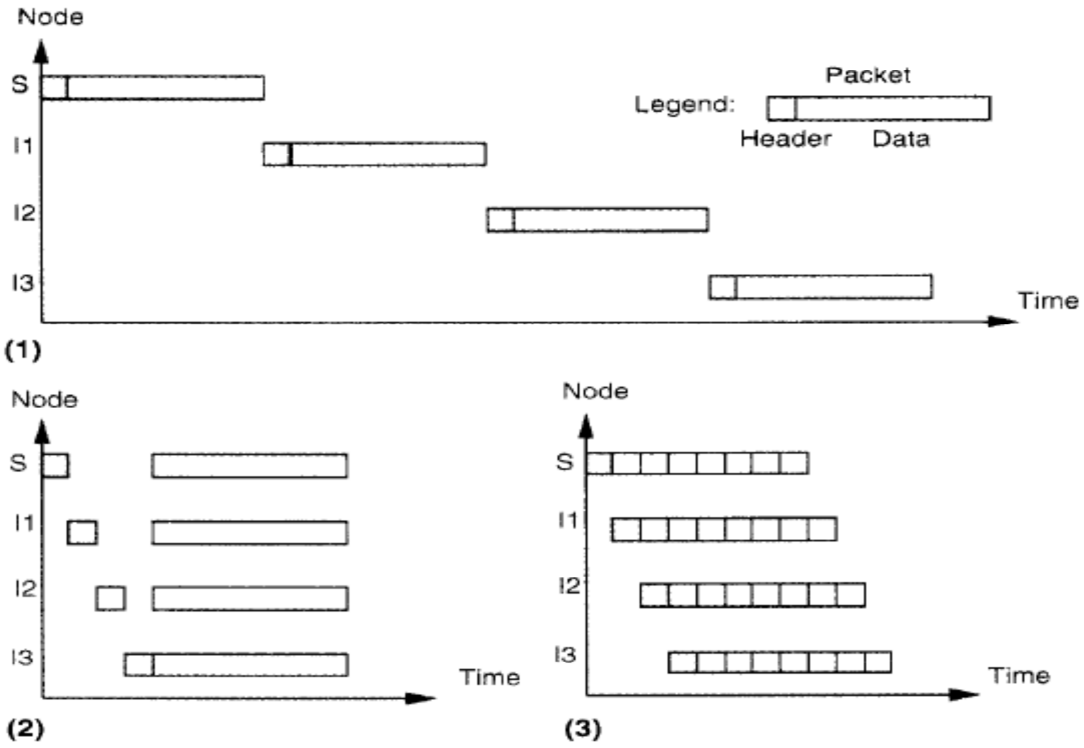


Figure 2.6 Comparison between switching techniques: (1) Store-and-Forward Switching, (2) Circuit Switching, and (3) Wormhole Switching [15].

2.2.1.4 Buffer Size

Buffers are needed to control the flow of messages traversing the network based on packet-switched techniques, and also advocated to handle network's contention and decrease latency. The minimum required buffer dimensions (width and depth) are functions of switching modes, packet size, flit size, and expected traffics. Buffers are approved to be the most dominating consumer of chip area among the routers' components. One of the vital goals in designing NoCs is to minimize the buffer size to save a considerable amount of area without or at least with reasonable performance (latency/throughput) degradation.

2.2.1.5 Link Width

The links between nodes represent communication channels that will be used to forward messages from one router to another through the path between source and distention. The

width of these links not only bounds flit size and bandwidth of the network, but affects the size of buffers at ports as well. Also, area and power consumption are influenced by tuning this parameter because of its direct impact on the number of wires in the network.

2.2.1.6 Routing Algorithm

The routing algorithm defines the path taken by a packet between the source router and the target router. Designer's goal is to make the routing scheme as efficient as possible to maintain high speed performance and low area and power consumption, while avoiding network ties such as deadlock, livelock, and starvation as well as the possibility of offering recovery mechanisms and congestion control. Routing algorithms can be classified in several ways. The surveys presented in [10] and [15] provide ample backgrounds about these classifications. For NoCs, XY routing is very popular for its simplicity and low area overhead. It's a deterministic minimal algorithm that is usually implemented using a distributed routing, where each node in the path will direct the packet based on the routing information included in its head. The packet is directed to the network X axis first until it reaches the Y axis of the destination node. Then, it is directed to that node through the Y axis. The XY-routing is one of the cheapest approaches to obtain a deadlock free network [4] and [16], and also prevents livelock [3]. One of the important factors that affects the latency of the network is the routing decision time, which is affected by the routing algorithm and how it is implemented. Minimizing this time decreases the latency.

2.2.1.7 Arbitration

Since NoC's routers usually receive simultaneous messages from surrounding nodes, internal scheduling is required to regulate the priority of granting output ports to incoming packets. This scheme is known as arbitration. It can be classified to static and dynamic. Static arbitrations are fixed to specific order, which makes them simple to implement; however, they are prone to starvation problem. On the other hand, dynamic schemes adapt with network conditions at run-time. Although this approach is more complicated to implement and might consume more area, it is more efficient, flexible, and provides a starvation-free NoC. There are various arbitration schemes such as; Round

Robin, First Come First Serve, Priority Based, and Priority Based Round Robin. Usually, the first two are used for best effort data delivery and the last two are used for guaranteed traffic.

2.2.2 NoC Evaluation Criterions

2.2.2.1 Cost Metrics

Area and power consumption are the most leading criterions of the NoC cost. The designer's goal is to minimize them especially for small and mobile applications, where these resources are limited. More concern about these costs arises with the utilization of FPGAs to accommodate contemporary SoCs that are known as FPSoCs. Because FPGAs have fixed logic units and routing paths, their area and power consumption are correlated. Getting small area usually depends on NoC design parameters and implementation efficiency. Buffers are approved to be the most area hungry components among other router's parts, by many researchers. However, buffers are important to reduce the latency and to handle data flow obstacles. Wormhole switching technique, for instance, is preferable and appropriate for low cost NoC design because its buffering requirements are low, although some buffers still needed to overcome its downsides like the case of adding virtual channels to prevent the deadlock situation or increasing the buffer depth to decrease channel reservation (making it close to VCT mode). Thus, buffer dimensions should be further lessened, as well as reducing the area of other router's components, by cost-effective design.

2.2.2.2 Performance Metrics

There are many metrics to evaluate the speed of NoCs and they mainly observe data transaction times. Overall operating frequency is one of the important factors that influences the speed of message delivery; however, it is not the only factor. The most universally applicable metrics of NoC are throughput and latency. Throughput is the amount of data transferred over a period of time. Ideally, it can be referred to as a bandwidth, which represents the data processing rate under the best possible conditions. Throughput can be decomposed into several intervals such as overall application, packet or flit throughput, measured per system, IP core, router, or port, calculated as an average,

along with other creative possibilities [3]. Latency, on the other hand, represents the wasted time between sending the data at the source node and receiving it at the destination node. Usually it is calculated as the average delay of packet/flit traversing the network. The lower bound of average latency is called the best case latency or zero load latency. It is the latency of the packet/flit when there is no congestion in the network. The other approach used to evaluate the latency with the present of congestion, is the simulation that injects artificial traffic and measures the delivery time. In all cases, internal router delay and serialization delay play a vital role in the overall latency. These delays can be reduced through tuning different parameters such as switching techniques and flow control, and minimizing the routing decision time.

2.3 Field Programmable Gate Arrays (FPGAs)

Field Programmable Gate Array (FPGA) is a special kind of integrated circuits (ICs) that constructed of arrays of pre-fabricated logic components, routing paths, I/O's, and reconfigurable switches, and have the ability to be programmed repeatedly to any desired digital design that does not exceed the IC capacity. Figure 2.7 illustrates FPGA general architecture. The design usually is implemented using a Hardware Description Language (HDL), such as VHDL or Verilog and downloaded (programmed) to the chip after a compilation procedure that includes synthesizing, placement, routing, and floor planning, by means of special software such as Altera Quartus II.

The logic units (blocks) are programmable components (typically in the form of look up tables) and their logic capacities vary from one FPGA product to another. An FPGA is programmed by loading data bits in memory cells which control transistor switches to establish non-permanent connections. An FPGA can support hundreds of thousands of gates of logic operating at speeds of tens of megahertz [3]. With growing chip density in terms of the number of transistors and gates, more capable logic units that can be used to implement more sophisticated functions are being offered by some FPGA manufacturers. Furthermore, modern FPGAs even contain bigger and advanced modules, such as memory blocks, DSPs, and processors, with some margin of configurability.

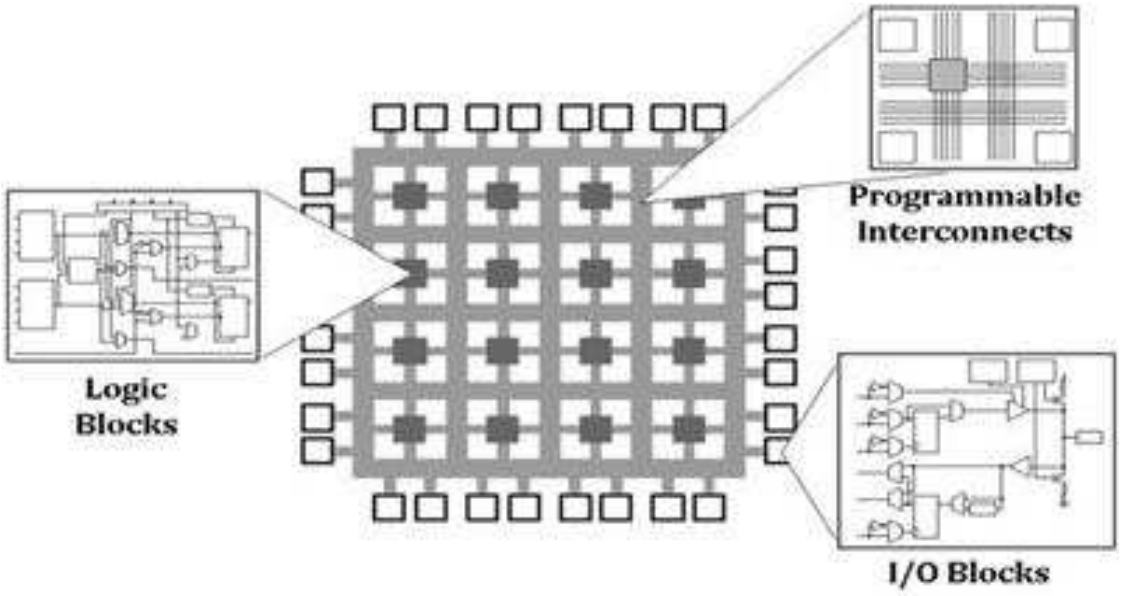


Figure 2.7 FPGA General Architecture [17]

2.3.1 FPGA Advantages and Disadvantages

- **Advantages:**
  - Can be programmed and re-programmed many times to implement any desired digital circuit that fits.
  - Compared to Application Specific Integrated Circuit (ASIC), FPGA provides flexible and fast prototyping implementation medium for embedded systems.
  - Low development cost and short time to market.
  - Easy to upgrade.
  - Suitable for research purposes because of:
    - Fast design cycle.
    - Immediate results.
    - No manufacturing operations involved.

- **Disadvantages:**
  - Generally slower than ASICs.
  - Limited area.
  - Need more power.

2.3.2 FPGA-based NoCs

As FPGAs capacity and capability grow, they are increasingly being used to build wider range of SoC appliances. The increasing heterogeneity coupled with higher operating frequencies enable FPGAs to replace ASICs in several high performance applications [4]. The concept of dynamically reconfiguring FPGAs applies well to micro-network design [8]. Because recent FPGAs have various hardware and/or software blocks embedded within them, such as DSPs, memory, and even processors, and because NoCs are approved to be promising solutions to the communication challenges of on-chip interconnections, these blocks, along with customizable logic blocks, make FPGAs perfect candidates for NoC designs. FPGAs limitations, however, such as area constrains that bound the size of the design and in some cases, conflict with obtaining the desired performance, add the challenge of choosing the wright set of NoC parameters that work best with FPGAs.

2.3.3 Capacity of a Logic Unit in FPGAs

Because FPGAs can be configured in a variety of patterns, and because their fundamental logic architectures are different among different FPGA products, there is no direct one-to-one theoretical mapping to compare the logic unit of one FPGA's architecture to another without performing full place and route compilations using the appropriate software tools [18].

The efficiency of logic utilization of an FPGA architecture depends on the following factors:

- The logic capacity of a single logic unit.
- The embedded functions that are present in the FPGA, such as a DSP block or embedded RAM.
- The structure of the design, such as whether the design includes multiplexing, wide functions, or arithmetic functions.
- The effectiveness of the synthesis tool.
- The quality of the place and route software.

2.3.4 Choosing FPGA Target Device

The most popular FPGA families of devices used in NoC research are Xilinx [19] Virtex-II and Virtex-4, and Altera Stratix and Stratix II. Before making the decision to choose the appropriate family and device that will be used for accommodating the application design, studying their structures and characteristics are needed for the following two reasons:

1. To meet the application requirements such as area, speed, and power.
2. To be able to make a comparison between the design on hand and previous work fairly.

2.3.4.1 Stratix II FPGA Family Logic Structure

The Stratix II basic logic unit is called the adaptive logic module (ALM). A single ALM contains two adaptive look-up tables (ALUTs), which provide up to two independent combinational outputs, two adder logic blocks, and two registers. [18]. Figure 2.8 shows Stratix II ALM.

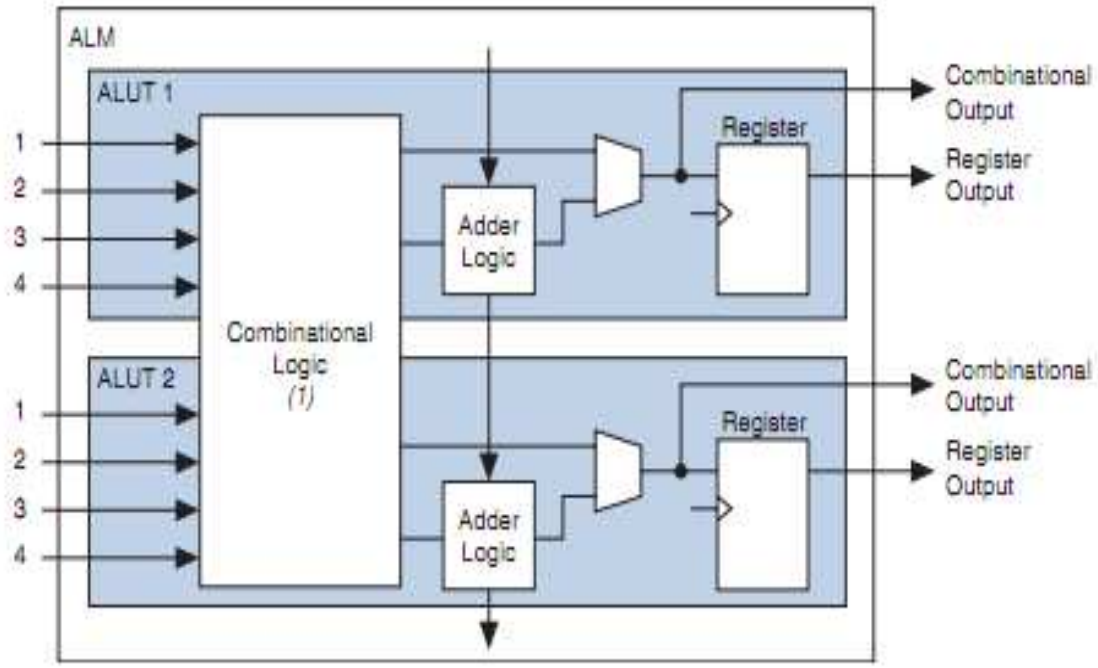


Figure 2.8 Stratix II ALM [20]

The combinational logic block can be adaptively divided into two LUTs that are either the same or different sizes. For higher performance and reduced area, one can configure an ALM as a large LUT with up to seven inputs. This reduces the logic level and routing delay that impact performance when cascading smaller four-input LUTs. For better logic resource efficiency, two LUTs of different sizes into a single ALM can be implemented to reduce the waste of logic resources [20].

2.3.4.2 Virtex-based FPGA Families Logic Structure

On the other hand, the Virtex-II Pro / Virtex-4 device logic unit is the Slice. A Virtex-based slice can be divided into two half-slices. Each slice consists of two fixed four-input LUTs, embedded multiplexers, carry logic, and two registers. See Figure 2.8. To implement functions with greater than four inputs, four-input LUTs are either cascaded together by general routing or combined together using the embedded multiplexers in the slices [20].

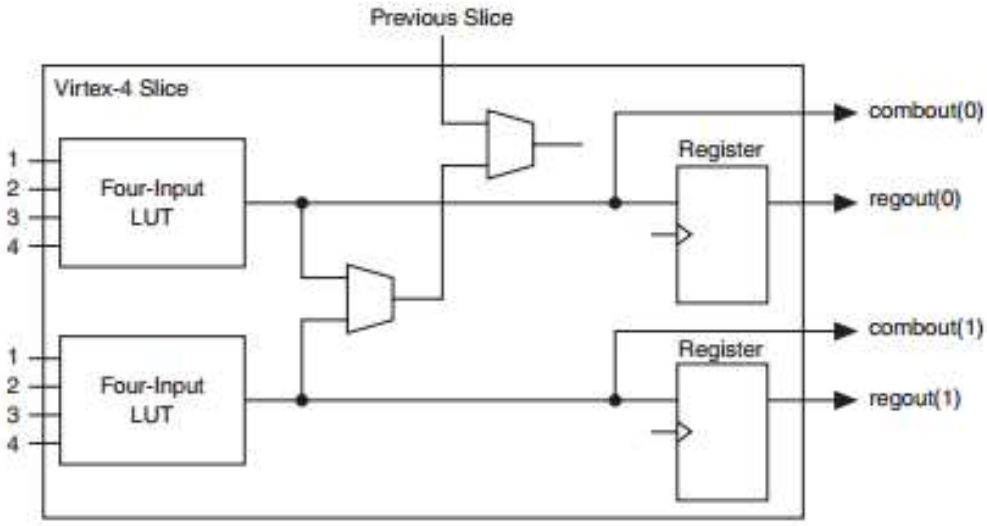


Figure 2.9 (Virtex-II Pro / Virtex-4) Slice [20]

2.3.4.3 Stratix II VS. Virtex-based Comparison

Based on Steve Sharp, Xilinx Corporate Solutions Marketing [21], the device utilization shown in Table 2.1 below for the test design, was taken from the respective map/fitting

reports generated by the implementation tools. It can be understood that Altera Stratix II device slightly consumes fewer logic resources than Xilinx Virtex-4 device. However, they performed this test to compare the power consumption between Stratix II and Virtex-4. The diagram in Figure 2.9 shows the power measurement according to their hardware-based power consumption evaluations. They verified that Virtex-4 FPGAs consume less power than Stratix II FPGAs.

Table 2.1 Stratix II VS. Virtex-4 Device Utilization Example based on Xilinx Source [21]

	Xilinx Virtex-4 LX60	Altera Stratix II 2S60
Logic Usage	21,731 LUTs* (40%)	19,849 ALUTs (41%)
Register Usage	17,315 FFs (32%)	16,175 FFs
Memory Block Usage	36 blocks (22%)	58 M4K blocks (23%) 6 M512 blocks (2%) 0 M-RAM blocks (0%)
DSP Block Usage	64 blocks (100%)	166 blocks (58%)

*Includes 2,062 LUTs used as shift registers (inferred in synthesis phase)

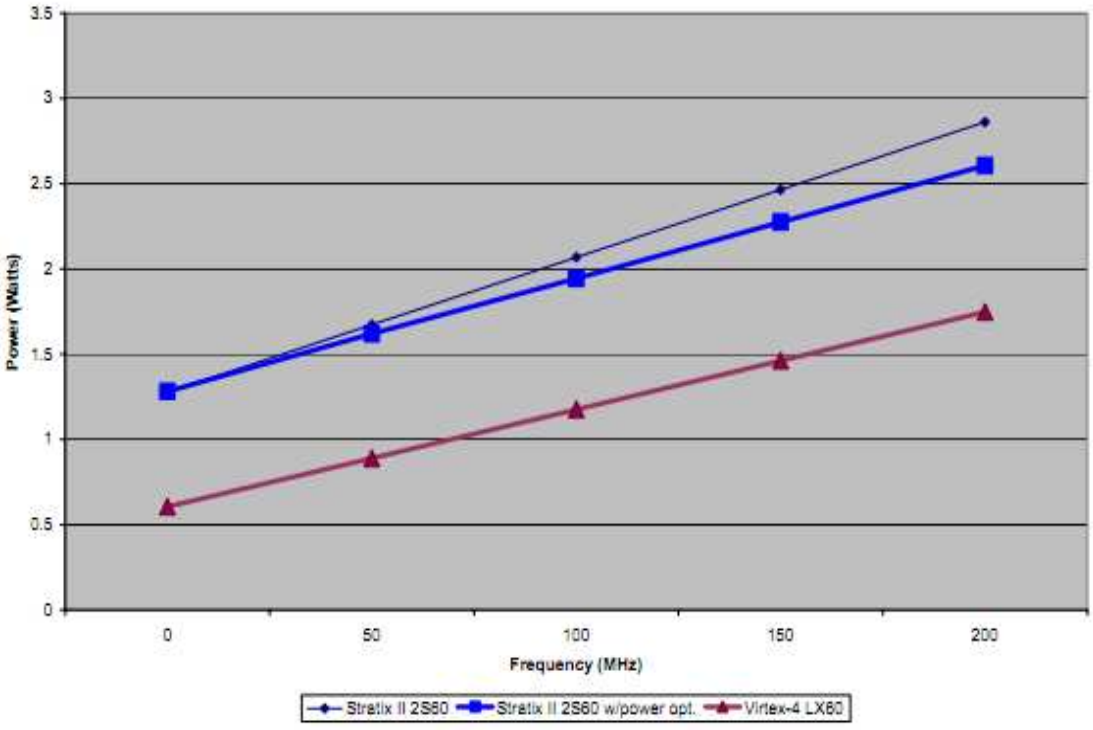
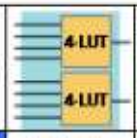
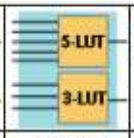
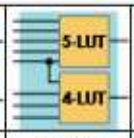
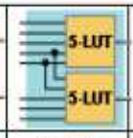
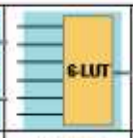
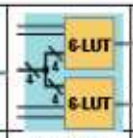
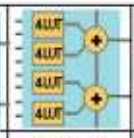
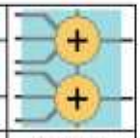


Figure 2.10 Stratix II VS. Virtex-4 Power Consumption Example from Xilinx Source [21]

On the other hand, Altera white papers [18], [20], claim the superiority of Altera Stratix II FPGA family in terms of resource utilization and performance. Table 2.2 compares Stratix II and Virtex-4 implementations of various LUT constructs and arithmetic functions.

Table 2.2 Stratix II vs. Virtex-4 Implementation Comparison for Various Functions [20].

Function								
Stratix II	1 ALM	1 ALM	1 ALM	1 ALM	1 ALM	1 ALM	1 ALM	1 ALM
Virtex-4	1 Slice	1.5 Slices	1.5 Slices	1.5-2Slices	2 Slices	2-4 Slices	1-2 Slices	2 Slices

Note: This comparison applies to all Virtex-based devices

Table 2.3 shows the performance and logic utilization summary for different simple design building blocks that commonly appear in real designs that was benchmarked and compared between the Stratix II and Virtex-4 devices by Altera.

Table 2.3 Design Building Block Performance & Logic Utilization Comparison between Stratix II &Virtex-4 FPGAs according to Altera [20].

Design	Description	Propagation Delay (ps)		Logic Utilization	
		Stratix II	Virtex-4	Stratix II (ALM)	Virtex-4 (Slice)
Five-input function	Two generic five-input logic functions.	53 to 378	334	1	2
Six-input function	Generic six-input logic function.	53 to 378	629	1	2
Seven-input function	Nested if-then-else function.	53 to 378	623	1	1.5
16-bit barrel shifter	16-bit single directional barrel shifter.	1,621	3,050	19	34
16-bit, 128-input adder	128-input adder tree without pipeline. Each input is 16-bit wide.	8,992	11,050	605	1,080

Note: Logic utilization excludes the ALM or slice's input and output register (flip flop) usage

Based on extensive benchmark analysis done by Altera [18], using real customer designs, the ratio of Stratix II ALUT to other architecture building blocks is specified in Table 2.4.

It shows the relative logic capacity of a single Stratix II ALUT compared to a Stratix logic element (LE) and a Virtex-based logic cell (LC), normalized to Stratix II.

One Stratix II ALUT is comparable to that of 1.25 Stratix Les and 1.54 Virtex LCs.

Table 2.4 Normalized Relative Logic Capacity based on Altera White Paper [18]

Stratix II	Stratix (LE)	Virtex (LC) Architecture
1 ALUT	0.8 ALUT	0.65 ALUT

Using the previous ratio as a conversion factor, when comparing the Stratix II and Virtex-II Pro devices, the approximate equivalent device density can be comprehended. Table 2.5 lists some devices in the Stratix-II and Virtex-II Pro families. Nevertheless, because of variations in design structures and logic utilization, the design must still be compiled in design software such as Quartus II to get the actual device utilization.

Table 2.5 Stratix II vs. Virtex-II Pro Equivalent Device Matchup from Altera [18]

Stratix II		Virtex-II Pro *(1)*		
Device	Equivalent ALUT	Device	Logic Cells *(2)*	Equivalent ALUT *(3)*
EP2S15	12,480	XC2VP20	20,880	13,559
		XC2VP30	30,816	20,011
EP2S30	27,104	XC2VP40	43,632	28,333
		XC2VP50	53,136	34,504
EP2S60	48,352	XC2VP70	74,448	48,343
EP2S90	72,768	XC2VP100	99,216	64,426
		XC2VP125	125,136	81,258
EP2S130	106,032			
EP2S180	143,520			

Note:

(1) Virtex-II Pro lower density devices are omitted.

(2) Logic cell count is quoted from Xilinx Virtex-II Pro data sheet.

(3) Virtex-II Pro equivalent ALUT count is based on the 1.54 conversion factor, which is from empirical data.

Xilinx white paper WP161 (Comparing Virtex-II and Stratix Logic Utilization) states that one slice is equivalent to 2.25 logic cells (LC) [3].

It can be concluded that:

$$Number\ of\ Startix\ II\ ALUTs = Number\ of\ Virtex\text{-}based\ Slices\ X\ 2.25\ X\ 0.65$$

We have chosen to use Altera Stratix II family because of the advantages mentioned above. This information, such as the conversion, factors will be useful in obtaining a comparison between our proposed design and some other designs that target different FPGAs products, in chapter 4.

2.4 Related Research Work

Network on Chip and especially FPGA-based NoC is a relatively recent topic of research, even though the operating theory is inherited from well-known computer networks. In addition to exploring the design space to enhance the performance of the NoC via tuning their wide range of parameters, researchers are also concerned about the FPGA's tight restrictions in terms of area, as well as the increase of power consumption. In this section, we give an overview of available FPGA-based NoCs' previous work achievements and shortcomings. A quantitative comparison and analysis of most relevant work to our design is explained in Chapter 4.

The authors of ICN [22] introduced a fully-pipelined 2X2 2D-torus NoC, and they use a wormhole routing. As well, they presented an IP interface model. The design requires 2 head flits and 1 tail flit to be included in a packet. The archived frequency is 40 MHz and the area was characterized and estimated, but no latency results were reported.

The work, presented by Bartic et al. [23], provides a scaled Virtual Cut Through (VCT) NoC. They optimized their design for speed by sacrificing very large area consumption. Their frequency is 50 MHz and the best case latency of their design was analyzed as a

number of cycles using a formula similar to the one we used to calculate and compare the latency.

RASoC [24] proposed by Zeferino et al., discusses the area of a wormhole router by varying the channel width considering FIFO depth = 2 and 4 flits, synthesized in Altera FPGA FLEX family. The achieved area is relatively high with this small size of buffering. Although they reported the average frequency, which is 56.7 MHz, they did not discuss the latency, or the effect of using such small buffers on it.

In Hermes [25], the authors tried to enhance the area and latency of their router and configured NoC; however, their design suffers from two drawbacks that prevent these enhancements. First, is the length of time required for the routing decision, which is 10 clock cycles, and second, is the use of handshake protocol that needs at least 2 clock cycles to forward a flit. This will increase the lower bound of the average latency and constrains the minimum needed buffer size. A third drawback is the slow frequency, which is only 25 MHz. However, they provided a formula to calculate the best case latency and proved the feasibility of this formula by simulation. A framework for networks on chip generation and verification is later presented by the same authors in MAIA [26].

LiPaR, presented by Sethuraman et al. in [27] is designed to establish simultaneous connections between input and output ports through decentralizing the arbitration as an attempt to reduce the latency. This along with the use of 5X5 crossbar, costs a lot of area; yet, the latency was still unimproved because of the use of store and forward (SAF) switching mode that makes the latency proportional to the packet size. The other issue with the use of SAF mode is the need to provide a large buffering especially when the packet size is big. They assumed that the packet size will not exceed 16; thus, their buffer depth is designed to be 16 as well; however, the area of 3X3 Mesh NoC wastes 3934 (28.72%) slices of Xilinx XC2VP30 FPGA with only (32.25 MHz) frequency. The latency was studied as the number of cycles needed for different scenarios of communication. They use a formula to calculate this latency. Later, Sethuraman, introduced a Multi-Local Port Router (MLPR) in [28] as an architectural modification to

LiPaR to enable the use of multi local cores. The stated results pointed out the percentage area saving and performance enhancement, compared to the single local core LiPaR, without reporting explicit synthesis results. Then, the authors extended their work by proposing Multi^2 router, a Multi-cast Multi Local Port Router in [29], intending to decrease the latency and traffic. They proved by simulation the ability to broadcast to any combination (non-contiguous random scheme) of local ports simultaneously. However, the synthesis results showed an increased area and timing as a result of incrementing the number of local ports. This happens because the width of data packet is increased to 16 in order to accommodate the extra bit requirements for implementing the multi-cast feature. In [30] they presented OptiMap: a tool for automated generation of NoC architectures using multiport routers for FPGAs.

Hilton et al., who published PNoC [9], compared their circuit switched (CS) router with bus-based networks and proved the performance advantage of a network-based approach in terms of scalability and simultaneous communications between the system cores. Some strategies have been proposed to overcome the major CS problems such as the idle time on communication links and required setup latency. They also compared their 8-ports router to a packet switched 8-routers Mesh NoC proposed in [23]. Yet, the area still large let alone the growth of the routing complexity with the increasing number of router ports that will limit the scalability. Further, the comparison does not include the latency although they reported a good frequency.

GNoC [31] proposed by Vistias et al., uses a generic router whose area can be traded-off for performance in many different ways. They characterized the area by varying the number of ports, bandwidth, traffic injection rates, and routing algorithm; by constraining the performance metrics such as: throughput and latency, for the execution of dedicated applications where heterogeneous NoC may be the most appropriate solution. They found that sharing some of the router's resources could enhance the area, but increases the latency.

Janarthanan et al. launched MoCReS [1], an NoC that enables the routers to function at independent operating frequencies. Their stand-alone VCT router can operate at high

frequency; however, the latency is bounded by the long setup time needed for accomplishing the routing decision. The area was obtained based on a buffer depth of only 8 assuming that the packet size will be 8 flits at maximum. Their router area will dramatically increase if more realistic packet size is used because VCT mode requires buffer size that is greater than or equal to the packet size.

The design, devolved by Brugge et al. in [3], intended to provide a parameterized NoC by proposing a component library and conducting some experiments to explore the effect of a variety of router parameters on area and latency. They include the effect of multi local cores and IP-core-to-router mapping strategy, which are comparable to the work in [28] and [30]. The area is competitive to the previous work, but still high. In spite of their router's ability to operate at a quite good frequency (100 MHz), the latency results are not convincing due to some design deficiencies including the use of the Store and Forward (SAF) switching mode that make the latency proportional to the packet size, and the use of handshake protocol as a flow control that requires more clock cycles.

Lu at al. [32] recently tackled the idea of reducing the zero-load-latency (minimum latency) and they attained 2 clock cycles. Nevertheless, because of wasted clocks in delivering the flits and credits, their router's buffer should be at least 5 locations depth to achieve full utilization of the available bandwidth. They also studied the effect of the number of ports (radix) and channel width (bandwidth) on router area, frequency, and power consumption. The FPGA resources utilization is relatively large for some configurations taking into account the recent FPGA device they used, which provides 270,400 LUTs.

We tried in our research to avoid most of the technical hitches noticed in reviewed literature to enhance the area without neglecting the performance. This overview has been attempted to some extent to be comprehensive; yet, because NoCs are a vast research field, not all the aspects are covered here. We agreed with the use of a formula similar to the one presented in some works to calculate the zero load latency and we extend that to utilize it to create a quantitative comparison with other work based on a case study.

2.5 Summary

A background about NoCs and FPGAs concepts and related work is covered in this chapter. An idea about On-Chip communication development, NoC building components, parameters and classifications of NoCs' routers, and NoCs' evaluation criterions are provided. Then, FPGAs' technology aspects that contain the advantages and the disadvantages, and choosing the appropriate FPGA device including a comparison between Altera Stratix II and Xilinx Virtex-based, are discussed. The Chapter ends with an ample review of available previously proposed FPGA-based NoC research. This background is important to presume the key factors of exploring the design space for FPGA-based NoCs and comparison. Chapter 3 presents a description of the proposed NoC router structural design, versions, design efficiency strategies, and functionality.

3 Chapter 3
NoC-Router Design and Architecture

The router represents the heart of the NoC, and its design efficiency determines the performance and cost of the Network. Our designed router has three versions that have a common general architecture, but have some differences in terms of design parameters and added enhancements. These versions are VCTR, WHR-1clk, and WHR-2clk. Starting with the chosen parameters for designing our router, the router's architecture, is then described along with the techniques used to augment the performance and reduce the area. After that, the functionally of the router is verified by simulation.

3.1 Setting Router Design Parameters

Our router is designed to support 2D-Mesh topology although it can be easily modified to work with other topologies, such as torus. Mesh topology is selected because of its scalability, physical mapping simplicity, and routing patterns straightforwardness, as well as cost predictability. This will be suitable for FPGA implementation.

Two modes of packet switching techniques are utilized to form the proposed router versions. The first mode is the Virtual Cut Through (VCT) that is used with VCTR version, and the second mode is the Wormhole (WH) that is used with WHR-1clk and WHR-2clk versions. Architectural and functional differences between these three versions will be discussed in the following sections.

Credit based flow control is employed in our design to allocate the network resources to the packets. The way we implement this protocol is very efficient. It consumes only one clock cycle to forward a flit from one node to another and also, requires only one wire or signal, which is called *Credit_in* or C*redit_out* in our design, for each input or output

channel. Hand shake protocol, in contrast, would require more clock cycles (at least two) and more wires (Request/Acknowledge) for each channel.

Buffer size is made parametarizable using VHDL generics. However, it is chosen to be eight locations depth when obtaining the synthesis results to make a fair comparison among the three versions and with previous work. It is referred to as *buffer_size*.

Link width, which is interrelated to flit size, is referred to as *flit_size* in this design and is also, made parametarizable. For prototyping purposes, it is also chosen to be eight bits.

Our design uses XY routing algorithm for its simplicity and efficiency. The direction that packets should take is decided by each node based on stored codes after comparing the destination address carried in the head flit of the packet. Direction decoding time is part of the routing decision time. We minimize this time by including a separate *Direction Decoder* unit that is responsible for providing this direction to the routing and arbitration unit, *Arbiter,* instantly after the arrival of the head flit to the first location of the buffer and activating the selection signal *LUT_sel* via the *Arbiter*, which is used to select the right address from the corresponding input port.

A central dynamic Round Robin arbitration scheme, which provides a starvation-free network, is implemented to control the priority regulation between input ports competing for an output port.

3.2 Router General Overview

Fig 3.1 shows the block diagram of the designed router and its external links that connects it to the adjacent (North, East, South, and West) IPs/routers, as well as the local IP. VHDL is used for the router implementation. There are one input port and one output port corresponding to each direction. Thus, the router has 5 input ports and 5 output ports in general. This number of ports is sufficient to deal with the worst cases of 2D-Mesh topology connections, which are the intermediate nodes, even though the network's external (sides) nodes require only 4 links, while the angle nodes need only 3. Further, not all routers have to be connected to local IP cores. Routers without local cores are used

to complete the network connections, and the number of required links is reduced by 1 in these routers.

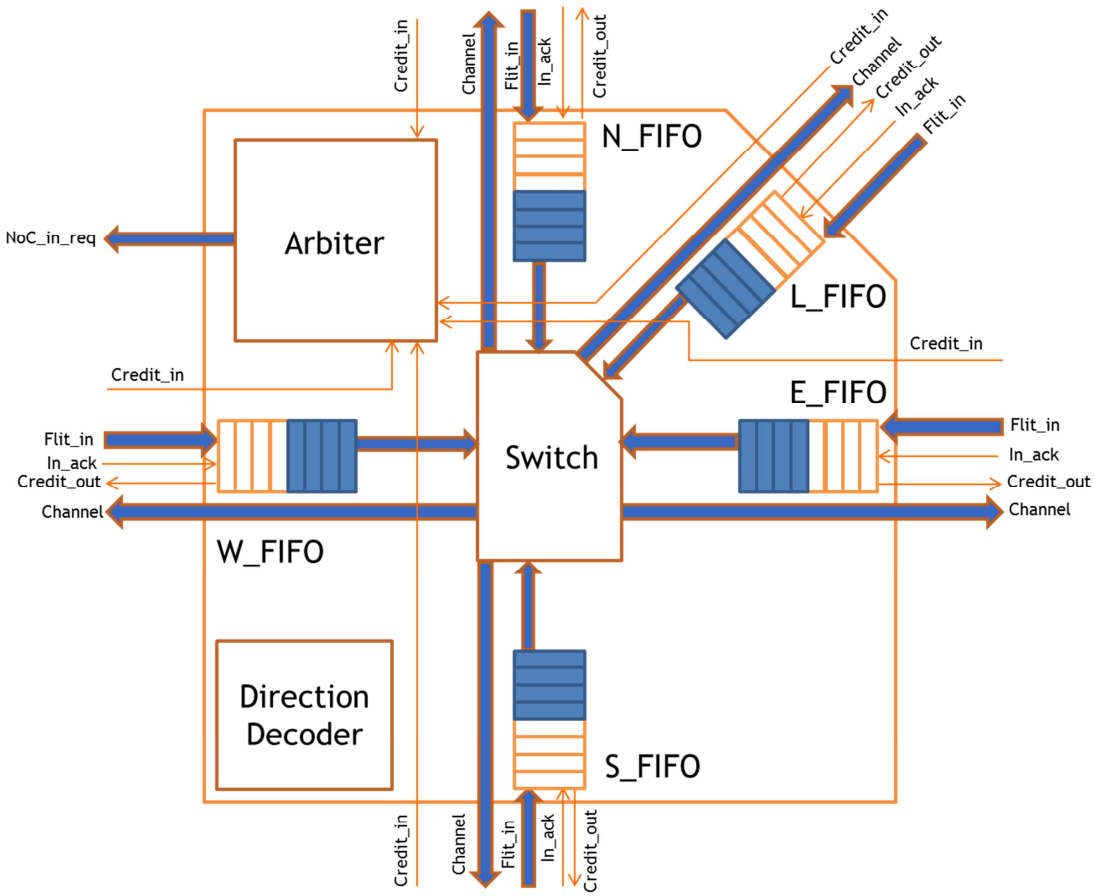


Figure 3.1 Router Block Diagram

The input ports receive the signals *Flit_in*, *Credit_in*, and *In_ack*, from the neighboring nodes or local IP, while the output ports send the signals *Channel*, *Credit_out*, and corresponding of *NoC_in_req*. The router is internally constructed of four components. These components are; *FIFO Buffer*, *Direction Decoder*, *Switch,* and *Arbiter*. Each component is either behaviorally described or structurally composed of sub-components that are also designed in the same manner. The total number of built components and sub-components is 13 including the main module, the *Router*, while the hierarchy of the router design utilizes 41 instantiated modules as it can be explained by Figure 3.2. This

hierarchical design enables us to instantiate some repeated components such as *FIFO Buffers*, and also facilitates the verification and modification stages.

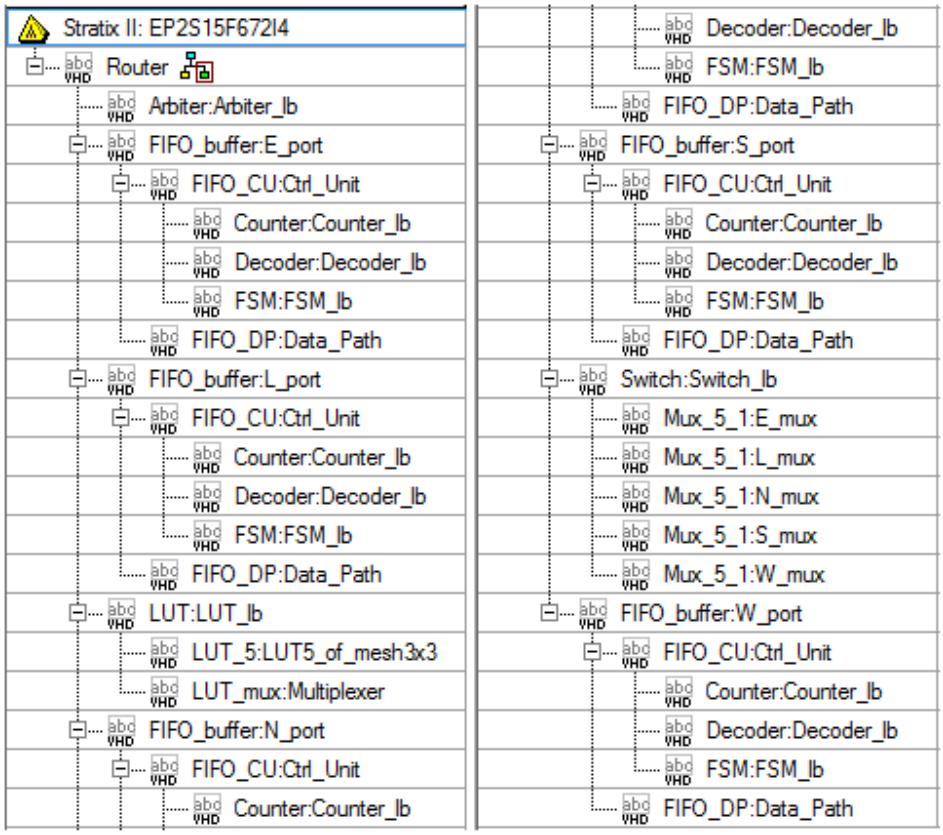


Figure 3.2 Router Design Hierarchy

3.3 Router Micro-Architecture

3.3.1 Router Versions Variances

Although the general architecture of the router is the same in the three proposed versions, there are some essential variances inside some components, which result from the use of different protocols to deliver the messages between the nodes. These variances change parts of the implementation logic, and hence, affect the physical characteristics and performance of the router.

VCTR version uses the virtual cut through switching technique that is characterized by the ability to buffer the whole packet. Since the buffer depth is limited and fixed during the runtime, the packet size should be pre-designed to be less than or equal to the designed buffer depth. In our router, buffer depth is parametarisable and can be chosen at design time using VHDL generics. It is chosen to be equal to the selected packet size, which is 8 flits, for prototyping purposes. There is no need to include the size of the packet in the head flit and no processing is involved to obtain this size at the routing and arbitration unit because it is already known. In contrast, WHR-1clk and WHR-2clk versions of our router use the wormhole switching technique that can deal with different packet sizes. The size of the packet will be included in the head flit and will be obtained by the routing and arbitration unit to use it to count the number of forwarded flits at runtime.

The difference between WHR-1clk and WHR-2clk, however, is the use of dual clock mechanism [5] with the proposed WHR-2clk version, which will be innovatively implemented with our router in FPGA-based NoCs. This mechanism is explained below:

- **Dual Clock Mechanism**

The concept of this mechanism is to use different clocks to forward the head flit and the body flits of the packet. The reason is the possibility of forwarding the body flits immediately without any computations. Hence, they can be forwarded with much faster clock than the clock that will be used for the head flit, which must follow the routing decision logic timing. The theoretical result is: reducing the average latency and the contention, via compressing the effective length of body flits in the time domain. Figure 3.3 shows the difference between using the same clock for head flit “H” and body flits “B”, and using different clocks, to control the flow.

3.3.2 Packet Format

The main role of a packet-switched network is to transport packets from source to destination. A packet is the smallest logical unit of data that an IP can inject into the network. A packet consists of multiple flits (packets can also be only one flit). A flit is the smallest physical unit of data that is routed by the network. The first few flits of the

packet are called header-flits. They contain the destination address (i.e. routing information) of the packet. The actual payload of the packet is contained in the remaining flits [22].

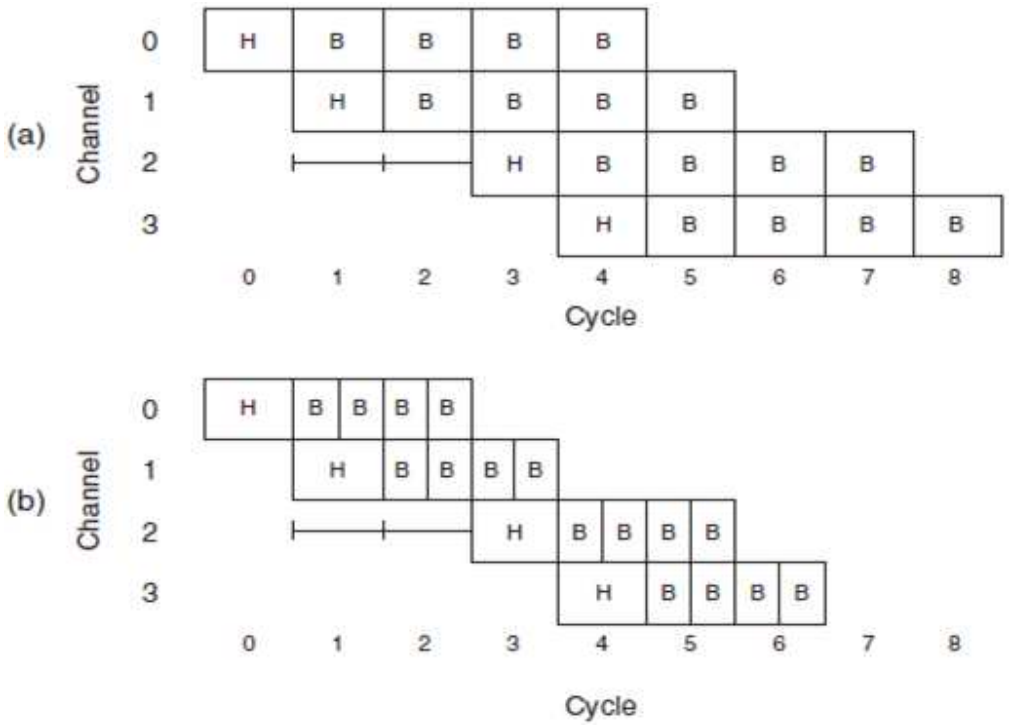


Figure 3.3 Time-space diagram for sending a 5-flit packet over 4 channels [5].
(a) WHR-1clk version. (b) WHR-2clk version.

The number of flits in the packet must be fixed and predefined in VCTR version. It is important to note that the maximum packet size will not exceed the designed buffer size (see section 3.3.2.1). On the other hand, using the wormhole (WH) switching mode in WHR-1clk and WHR-2clk versions, gives more flexibility to the packetizing unit to form the packets with desired size. The wormhole router will be able to deal with different packet sizes without predefinition. However, the flit size, in bits, affects the upper boundary of the packet size (see section 3.3.2.2).

A novel idea is introduced in our design regarding the minimizing of the control fields included in the flit. As it can be seen in Figure 3.4, no control fields are included in the body flits. Also, the packet does not have a tail flit. In order for the router to sense the

head flit and the end of the pay-load, some logic has been added in arbitration and routing unit instead of using the traditional flit type field(s), (see section 3.2.5). This idea works toward reducing FIFO buffer width by some bit(s), consequently, saving a considerable amount of consumed buffer area, as well as the power consumption, and providing more realistic evaluation in terms of the actual delivered data (Throughput). The feasibility of this approach has been approved by comparing the synthesis results with and without using control fields.

3.3.2.1 VCTR Version Packet Format

In our design, the packet size for VCTR version can be predefined anywhere from 1 flit to the designed buffer size number of flits. As shown in Figure 3.4, the first flit is the head flit that contains the routing information. It is followed by the body flits that carry the data. Looking closely at the head flit, it can be seen that the most significant 4 bits are particularized to store the destination address of the packet. The least significant bit will be set high by the packetizing unit to be used as the arriving notification signal. The remaining least significant bits (flit size – 4) are not used for this mode because, as mentioned before, the packet size is already known by the router as a built-in parameter; it is predetermined during the design time as a function of the buffer size.

$$(Packet\_size)_{flits} \leq (buffer\_size)_{locations}$$

3.3.2.2 WHR Versions Packet Format

In the wormhole mode case, the size of the packet sent by the packetizing unit at the source node is unknown by the routers. This size must be included in the head flit (Packet Size field in Figure 3.4). Our wormhole router is designed to obtain this size and use it to transfer the right number of flits after establishing the connection between the current router and the next router in the path that is already set by the routing and arbitration logic. The relationship between packet size and flit size is defined by the following equation:

$$(Packet\_size)_{flits} \leq 2^{(flit\_size\ -\ 4)}{}_{bits}$$

Where flit_size $\geq$ 5 bits.

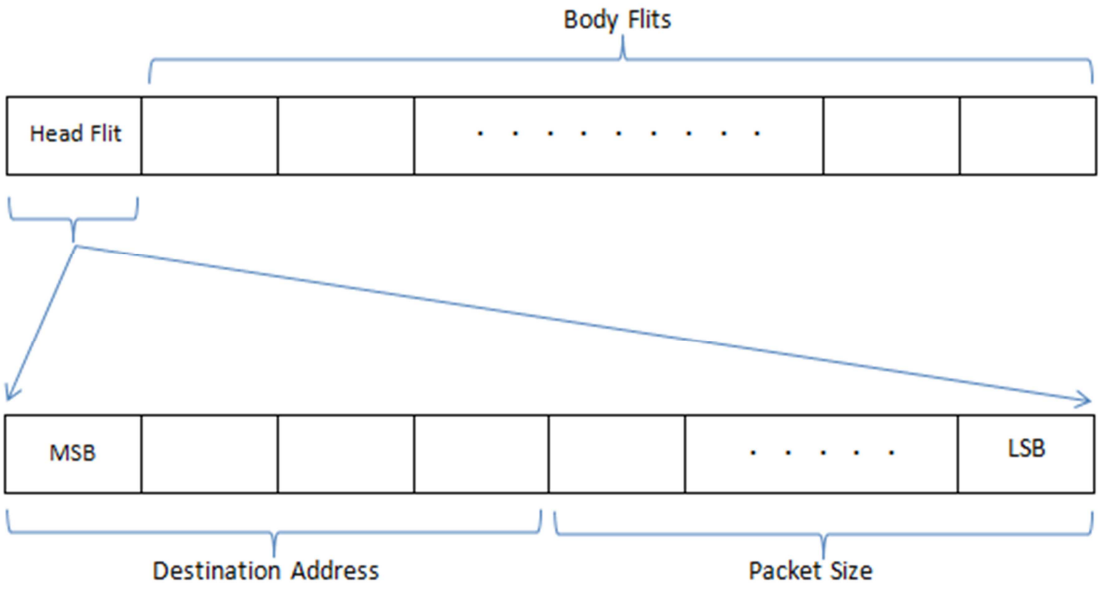


Figure 3.4 Packet Format

3.3.3 Input Port (FIFO Buffer)

Associated with each input port, we use an input FIFO buffer for temporarily storing the coming flits until they find their way to the final destination. Our buffer works on the basis of first in first out (FIFO) mechanism. Figure 3.5 sketches the internal design of our *FIFO Buffer* module. It consists of two main units: control unit *FIFO_CU* and data path *FIFO_DP*. *FIFO_CU* receives the control signals *In_req* and *Out_req*, and transmits the signal *Credit_out*, and it is responsible for flits loading and shifting, as well as tracking the buffer usage through its internal sub-modules: Finite State Machine *FSM*, *Counter*, and *Decoder*. *FIFO_DP*, on the other hand, receives the coming flits, stores them temporarily, and transmits them to the next stages in the router under the control of *FIFO_CU* unit. It is designed as a number of shift registers *SR*. Each *SR* is used to store one flit, which with other *SRs*, represents the depth of the buffer. The number of locations (FIFO depth) must be greater than or equal to the number of possible flits that can be stored during the routing decision time in order to get the best case latency. In our router design the decision time is only a one clock cycle; and the time needed to write a flit is only one clock cycle as well (Credit-based protocol has been used). Therefore, only one

buffer location, as well as the head flit location, is enough to get the minimum latency with the absence of blocking. However, the buffer size has been made parameterizable and chosen to be 8 to make a fair comparison with other designs. Also, these extra buffer locations will be useful to handle the contention if needed.

Upon the arrival of the packet at the input port, it is stored in the *FIFO Buffer* flit by flit every clock cycle; given that free space is available. At the same time and with the same speed, *FIFO Buffer* will be evacuated as long as the assigned output port finds free space at the adjacent destination's input port. The evacuation stops when the control signal "*Credit_in"* coming to *Arbiter* from that adjacent node is not valid, indicating that no more space is available. However, the storage in the current buffer will continue. The control signal "*Credit_out"* representing this port will be sent as a back pressure to the adjacent source node, when *FIFO Buffer* becomes nearly full.

As soon as the first flit (*head_flit*) reaches the first location of the FIFO, which is connected directly to the routing stage as well as the *Switch*, the packet information included in this header flit will notify the *Arbiter* about the arrival of the packet in this port, and provide the destination address to the *Direction Decoder* to calculate the direction.

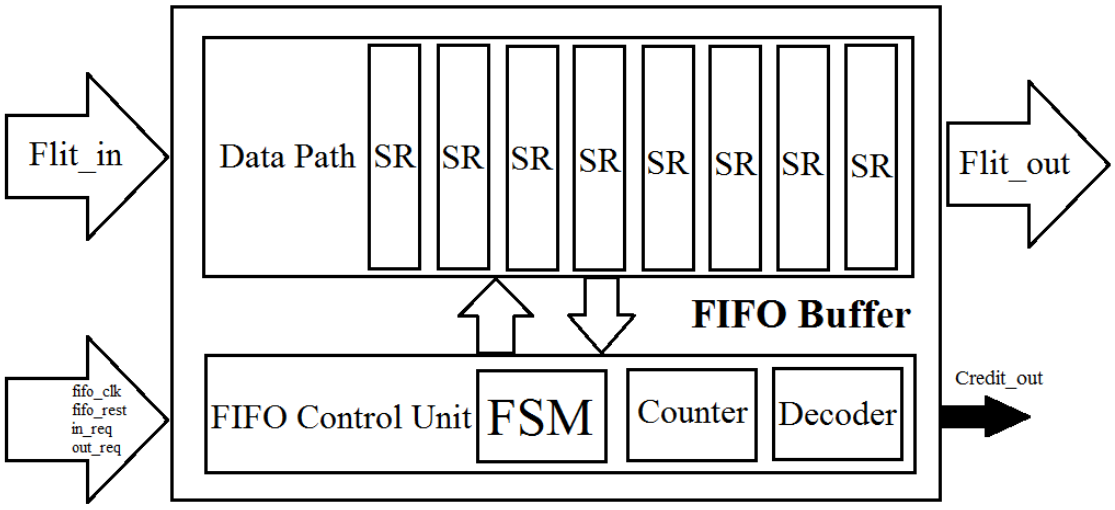


Figure 3.5 The Architecture of FIFO Buffer

3.3.4 Direction Decoder

The *Direction Decoder* unit is responsible for obtaining the direction that packet should take based on the address provided by the head flit. An architectural diagram of this unit is shown in Figure 3.6. It receives the address from the head flit that just arrived to the first location of the *FIFO Buffer* unit. The addresses coming through the input signals *N_addr*, *E_addr*, *S_addr*, *W_addr*, and *L_addr* are multiplexed via a *LUT_mux* component, which is controlled by a signal called *LUT_mux_sel* derived by the *Arbiter*. The direction will be forwarded through the signal *Dest_out* to the input signal *Direction* at Routing & Arbitration unit *Arbiter* after decoding in the form of a 3-bit code (001, 010, 011, 100, or 101) representing the directions (North, East, South, West, or Local) respectively. It is important to note that the design of this unit depends on the location of the node in the NoC taking into account XY-Routing algorithm. Our design of this table was done based on 3X3 mesh topology for prototyping purposes; however, it can be easily modified to support 4X4 mesh or even bigger and different topologies by modifying the other parts of the router such as the packet format.

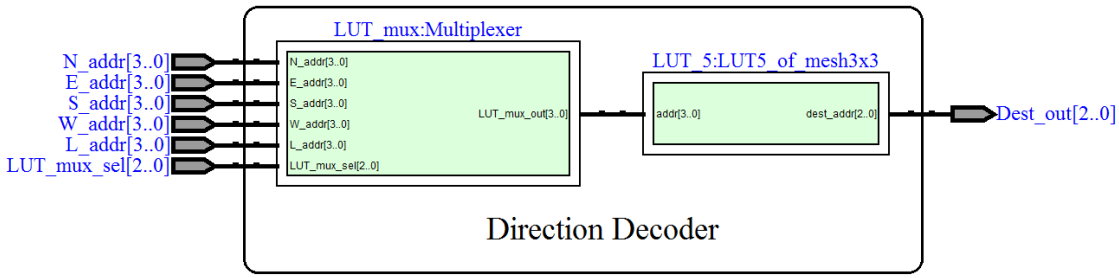


Figure 3.6 Direction Decoder Architecture

3.3.5 Routing & Arbitration Unit (Arbiter)

Although several messages (up to 5 in our design) can arrive and be buffered at the router's input ports concurrently, the routing decisions are serialized by traversing the output channels in the order decided by the *Arbiter*. Arbitration is done by a round-robin method implemented with a dynamic priority scheme. The notifications arriving from the input ports will be served in the following order: North, East, South, West, and Local. The last port served will be given the least priority in the following transmission.

Because of the variances between our router versions mentioned in section 3.3.1, there is a difference in some input signals entering the *Arbiter* as is explained in Figure 3.7. VCTR version uses *N_arrived, E_arrived, S_arrived, W_arrived,* and *L_arrived* single wire signals, coming from the least significant bit of head flits, as notifications of the north, east, south, west, and local input ports respectively. On the other hand, both WHR-1clk and WHR-2clk use *N_packet_size, E_packet_size, S_packet_size, W_packet_size,* and *L_packet_size* multiple wire signals as notifications of the corresponding input ports as well as carrying the packet size information. The packet size is obtained and considered as the start value of a counter that will be decremented by one each time a flit is transported.

Upon granting an input port, the *Arbiter* configures the *Direction Decoder* unit using the output signal *LUT_sel* asking it to provide the direction based on the information provided by the current processed packet from the current corresponding port. Then, the availability of free buffer locations in the neighboring destination hop is checked through examining the validity of the signal *Credit_in* connected to input signals *N_credit, E_credit, S_credit, W_credit,* and *L_credit* before opening the channel. As well, by using the output signal *in_out_Ctrl,* the *Arbiter* will acknowledge the *FIFO Buffer* unit and enable the multiplexers in the cross point matrix *Switch* to establish a connection.

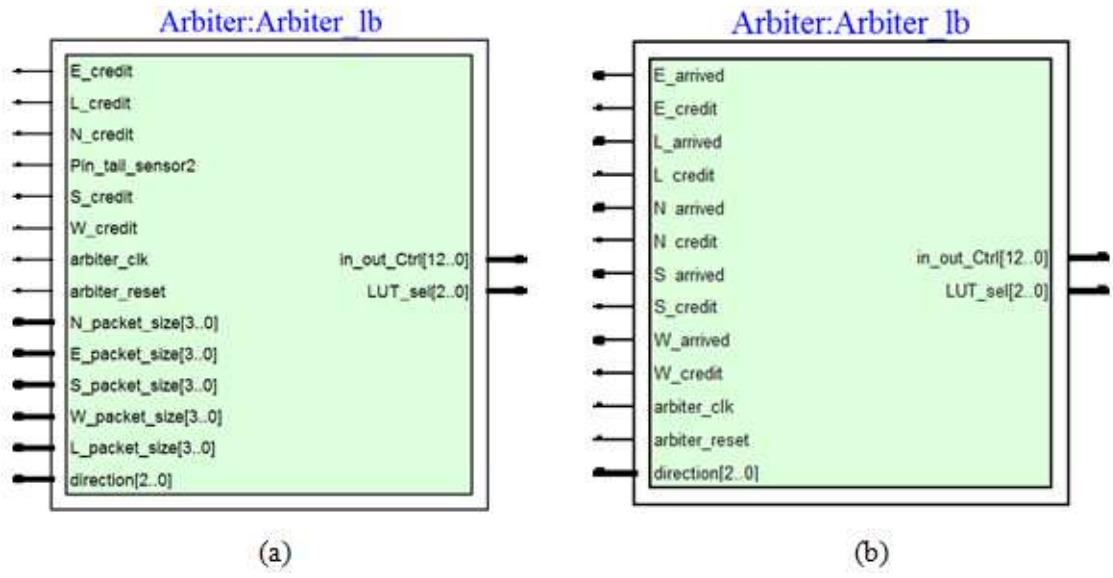


Figure 3.7 Arbiter Architecture (a) for WHR versions (b) for VCTR version

3.3.6 Switching Box (Switch)

The switching box is the final stage of the router. It maps the packets coming from the input ports to the assigned channels. The *switch,* as shown in Figure 3.8, is composed of five (5–to-1) multiplexers supporting all possible connections between input and output ports. Enabling the multiplexers and selecting the appropriate inputs is accomplished via the control signals *N_en, E_en, S_en, W_en,* and *L_en,* and s*el* respectively, which are derived by the *Arbiter*. The input signals *N_in, E_in, S_in, W_in,* and *L_in* are linked to corresponding flits coming from input buffers, while *N_out, E_out, S_out, W_out,* and *L_out* are joined to router channels. Smaller multiplexers (2-to-1 and/or 4-to-1) could have been used to establish the connections according to the node position in the network. The possibility of that is due to the fact that packets are not supposed to be routed back to the node they come from. Also, based on XY routing algorithm, packets arriving from vertical directions are only routed to the other direction (North or South) of the same axis (Y), (not to (X) directions East and West), as well as the routing to the local port. Furthermore, the routers located at the sides and the angles (corners) of the mesh network, have fewer numbers of adjacent nodes, and consequently, have fewer number of input and output ports. In order to evaluate the effect of using the general full (5-to-1) multiplexers versus the use of customized (2-to-1 and/or 4-to-1) ones, and come up with the optimal design in terms of area, speed, and power consumption, some experiments were performed on our router on the basis of it being put in the middle of the NoC and surrounded by other nodes from all the directions (worst case). The characteristics of our single clock wormhole router were gaged using *Altera Quartus II* synthesis tool. Interestingly and surprisingly, it showed that resource utilization (area) had been increased and the frequency had been decreased, when the smaller customized (2-to-1 and/or 4-to-1) multiplexers are used to construct the switch instead of using the full bigger (5-to-1) multiplexers. FPGA device architecture might be behind this unexpected result. As a result of this experiment, and for simplicity, flexibility, and reusability of general (5-to-1) multiplexer in our router design, it is chosen to build the *switch* model. The effects of customizing the multiplexers all over the NoC have not been evaluated because it is out of the focus of this research; however, it will be interesting to consider

that as part of future work taking into account NoC sides and corners and exploring the FPGA aspects of utilizing the internal logic elements to configure NoC functional design.

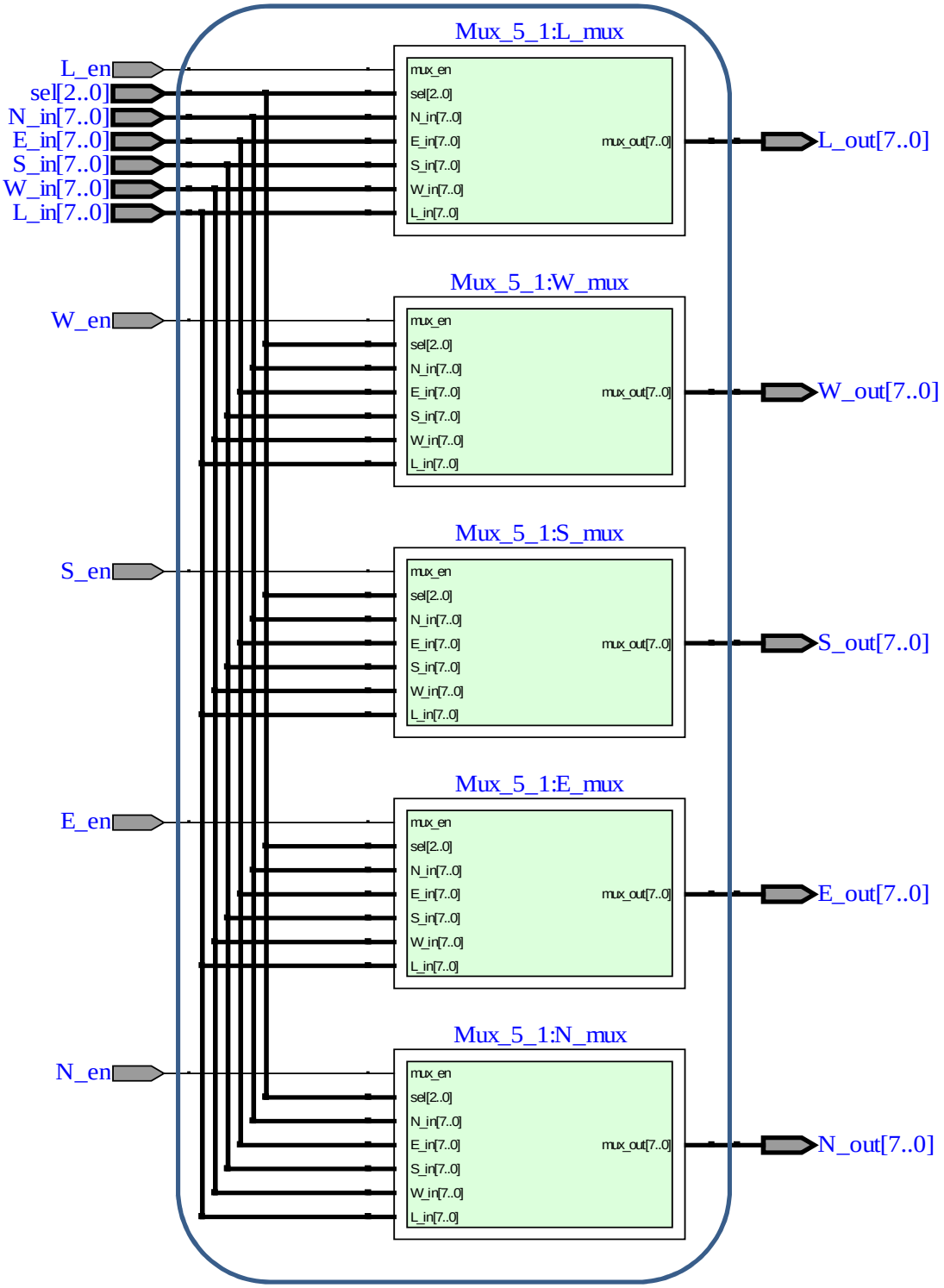


Figure 3.8 Switch Architecture

3.4 Design Improvement Techniques

In order to refine the area and the speed of our router, some strategies during the design are applied. Although some of these techniques are mentioned previously, the most effective and innovative ideas are summarized here:

- Minimize the number of control fields in the packet as well as eliminating the tail flit, while replacing them with some logic and counters in the routing unit to infer the start and end of the packets, and consequently, reduce the FIFO size.
- Use the credit based flow control, so we are able to transmit the flit in only one clock cycle, instead of using the hand shaking protocol that needs at least two clock cycles, such as the one used in Hermes [25]. Also, we reduced the number of wires to one wire (Credit signal) as an alternative to the two wires (Request/Acknowledge Signals) in the case of hand shaking flow control.
- Utilize the rising edge of the clock to accomplish and synchronize some operations while employing the falling edge to undertake other operations. This technique enables us to squeeze a number of dependable operations in only one clock cycle either in the FIFO Finite State Machine (FSM) Read/Write operations, or arbitration and routing handling.
- Implementing and exploring the idea of augmenting the speed of the wormhole router using the dual-clock mechanism. This mechanism forwards the body flits of the packet with faster clock instead of waiting for the slower head flit, resulting in the compact of the effective length of body flits in the time domain.

3.5 Functionality Verification

Since our router is designed in a hierarchal structure of sub-modules, the functionality is verified from the base of the pyramid to the top using Altera Quartus II waveform editor and simulator tool. The main task of the router is to switch incoming packets to the appropriate channels, which will continue their way until they reach the final destination. This role of the router is demonstrated by simulation. The following sub sections discuss the functional simulation of the top module *Router* for the three proposed router versions. This simulation represents the router behavior of the node R5 in the center of a 3X3 Mesh

network as it can be explained in Figure 3.9. The worst case is considered in this simulation, which is receiving concurrent packets from all surrounding 4 nodes as well as the local IP.

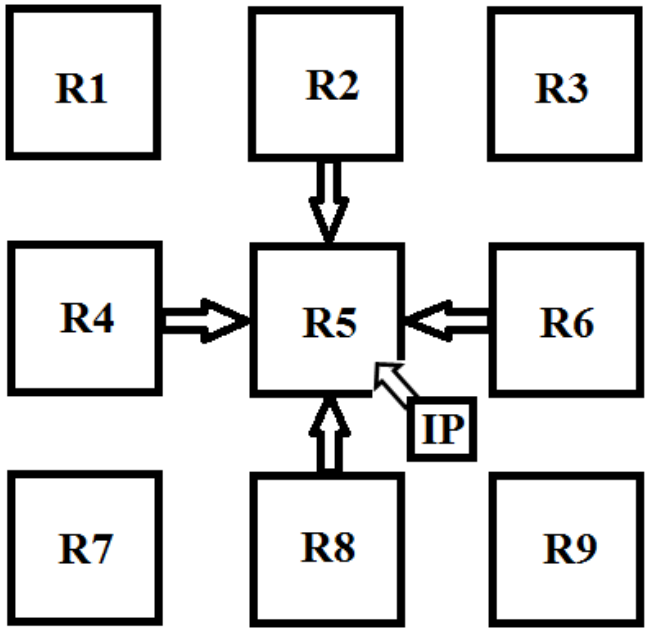


Figure 3.9 Simultaneous Communications in NoC Router

3.5.1 VCTR Simulation

Figure 3.10 shows the simulation of 5 packets coming simultaneously to the input ports of the router through its signals *N_flit_in, E_flit_in, S_flit_in, W_flit_in,* and *L_flit_in.* Each packet consists of 8 flits. The first flit is the head flit that contains the destination address. All packets are routed to the appropriate channels based on these addresses. For example, the packet received from R8 (Figure 3.9) through the southern input *S_flit_in* (Figure 3.10) is directed to be sent to the node R2 since it has the address 2 in its MSB of the head flit. It can be noticed that this packet is routed to the northern output *N_channel* toward its destination. Routing arbitration also can be observed through the timing of output channels as well as *Arbiter_state* signal. Because VCT switching is used in this version, packet size is fixed and the timing of changing the arbitration state is equal.

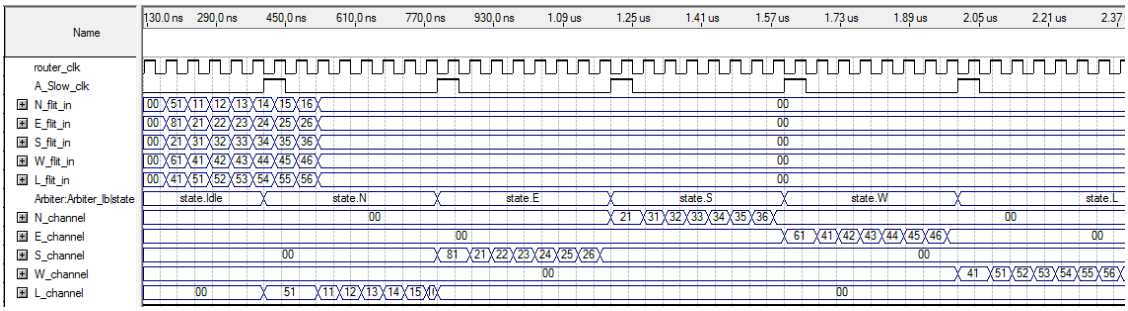


Figure 3.10 VCTR Functional Simulation

3.5.2 WHR-1clk Simulation

The behavior of this version of the router that uses WH switching is shown in Figure 3.11. The coming packets have different sizes. This size is included in the LSB of the head flit. For example, it is 6 in the packet entering through *E_flit_in* signal. That means forwarding 6 flits to the appropriate channel, which is *S_channel* in this case, because the destination address is the node 8. The timing of changing the arbitration state as we can see follows the size of the routed packet.

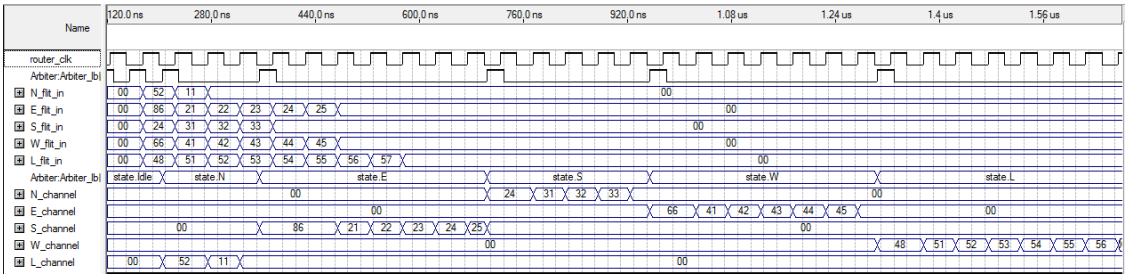


Figure 3.11 WHR-1clk Functional Simulation

3.5.3 WHR-2clk Simulation

The difference between WHR-1clk version and WHR-2clk version can be noticed in Figure 3.12. Instead of forwarding the head flit and body flits at the same speed as the first, the later uses faster clock *body_clk* to forward the body flits. One can see that the router clock *Pin_router_clk* is mixed between the slow *head_clk* and the fast *body_clk*. The result of that is compressing the time needed to route the body flits.

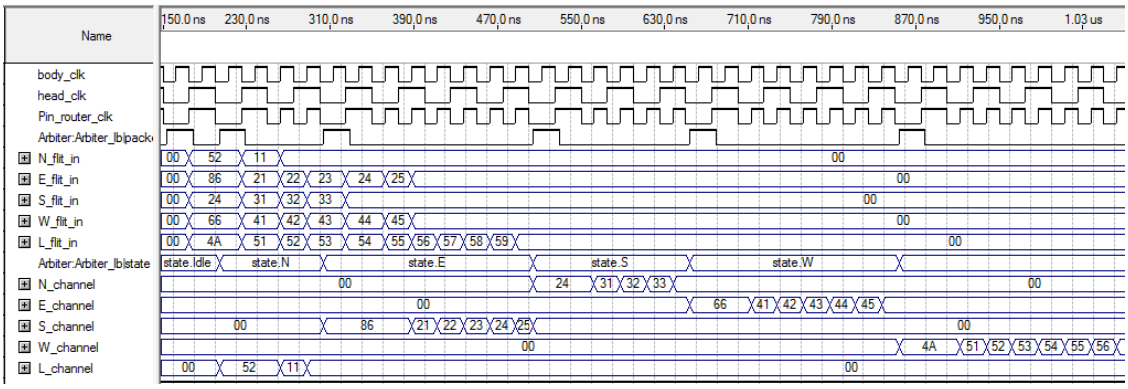


Figure 3.12 WHR-2clk Functional Simulation

3.6 Summary

This chapter presents the architecture of our router, setting the design parameters, and the differences between the three proposed versions of the router. It also talks about the techniques and strategies utilized to achieve high performance and low area, and it concludes with the verification of functionality via simulation. The evaluation process, implementation results, and comparison are discussed in chapter 4.

4 Chapter 4
Evaluation and Results

4.1 Assessment Approach

As it is previously discussed in section 2.2.2, there are many metrics to evaluate the cost and performance of the NoC routers, which in turn are affected by tuning the NoC router's parameters. In this section, we set the specific methodology that was followed to evaluate our design, and compare its versions among each other and to some other previous routers.

The first step is choosing the implementation and testing environment. For the reasons addressed in section 2.3.4, the device EP2S15F672I4 from a very popular Altera Stratix II FPGA family is chosen to be the target of our design. VHDL was used to implement the router design and its internal models. Altera Quartus II CAD tool software [4] is the environment used for synthesizing and compilation (including placement, routing, floor planning) of the design code to determine the actual physical characteristics in the FPGA device like Area, Frequency, and power consumption. Associated with this software, there are built-in tools such as Resource Optimization Advisor, Timing Optimization Advisor, and Power Optimization Advisor. These tools were used in our design to guide the synthesis tool to create the design net-list based on the desired outcome in terms of cost or performance. Hence, the full range of design space can be explored. The other tool accompanying Quartus II software is the Simulator Tool. This tool was used to simulate the router and its internal models to test the functionality of the design and the verification.

The second step is choosing the stages as well as the criterions and metrics that will be used to explore, gauge, and compare our design cost and performance.

These measurements are explained by the following stages:

1. Obtain the physical characteristics (Area, Frequency, and Power Consumption) of the standalone router for our three router versions (VCTR, WHR-1clk, and WHR-2clk) and compare them among each other.
2. Connect 9 modules of each router version to configure 3X3 Mesh NoC. Then, track down the physical characteristics for the whole NoC, as well as notice the area of each individual router module after configuration.
3. Calculate the zero load latency for our stand alone VCTR and for the stand-alone router of MoCReS [1] that uses VCT switching, and compare them in terms of area and performance.
4. Compare our WHR-1clk with three widely cited FPGA-based NoC routers in terms of area and performance including the zero load latency based on case study.

4.2 Results and Analysis

In this section, we report and explore the results attained based on the evaluation methodology that was set.

4.2.1 Synthesis Results

To provide a wider exploration of the router's design space, the area, frequency, and power consumption of our router and the prototyped 3X3 Mesh NoC are reported with different design optimization. Each characteristic is recounted when the router design is being optimized for speed, area, and consumed power; one at a time. These results are presented, analyzed, and compared below. Full synthesis results can be found in the Appendix. Although the flit size and the packet size are parametarizable in our design, we reported our results based on a fixed size for both of them to explore the effect of other parameters such as: flow control and switching techniques through the comparison between the three versions of the router. Fixing the flit and packet sizes also facilitates the comparison to previous work. Therefore, flit size and packet size are chosen to be 8 bits and 8 flits respectively.

4.2.1.1 Area

Our router consumes only a small percentage of the available FPGA device area leaving the remaining logic to efficiently implement the IPs. This percentage varies from version to version of our router, and is affected by the sittings of the optimization advisors. Guiding the synthesis tool by these advisors provides a wide range of design space to choose the appropriate area of the NoC based on the desired application. The logic utilization of the FPGA resources, to the router module, also changes when different configurations of the same module are synthesized. For example, the area of the stand-alone router different from the area of the individual router module that is configured as a part of 3X3 Mesh NoC. Similarly, each router module of the 3X3 Mesh NoC has a different area from others based on its position and connections.

Figure 4.1 shows the logic utilization (area), which is represented by the number of (ALUTs), for the stand-alone router. The area of the three versions of the router is compared when the design is optimized for speed, area, and power consumption.

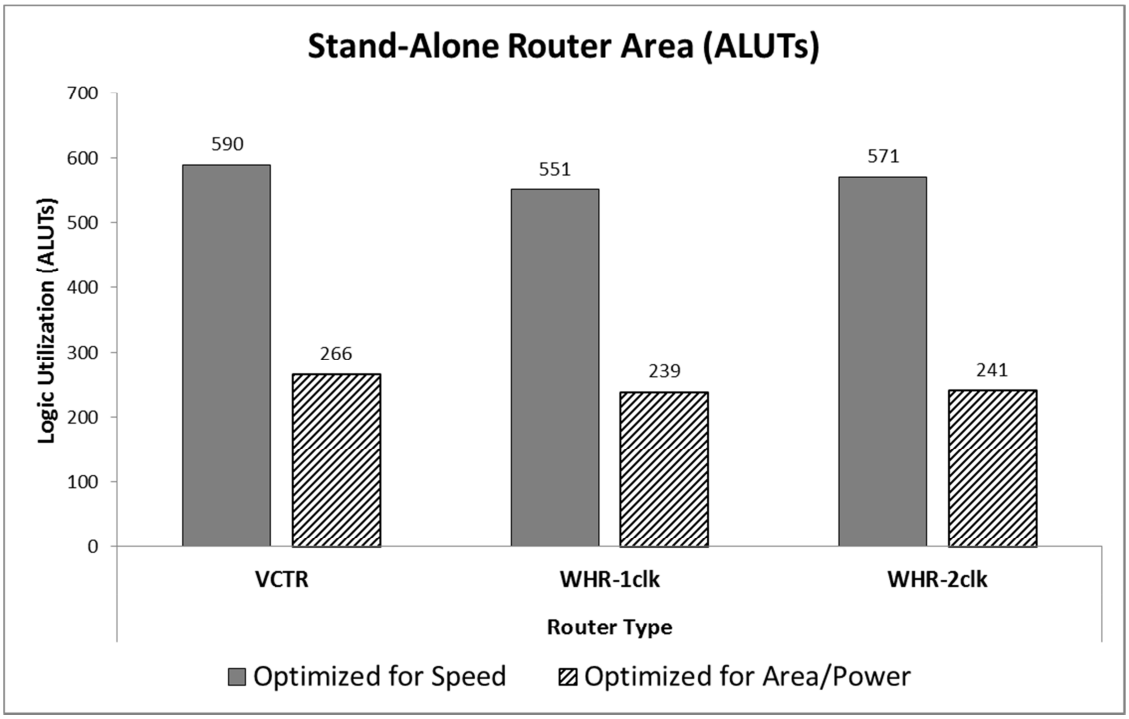


Figure 4.1 Stand-Alone Router Area for the Three Versions

We noticed that area and power consumption are closely correlated. However, optimizing the design for the speed increases the area and power consumption.

The 3X3 Mesh NoC area is given in Figure 4.2 for the three router versions, while the average of the router area in this NoC is shown in Figure 4.3. They explain the change of router area as a result of configuring the routers in the 3X3 Mesh NoC. It is important to mention here that the area of each individual router model is reduced after synthesizing the whole 3X3 Mesh NoC, and each model has a different area from others. Also, the variances between the areas of the three versions are changed after NoC configuration. For instance, we can notice that the area of VCTR version is the biggest among the stand-alone routers, whereas it is the smallest after NoC configuration. When we optimized the design for the area, the average area of VCTR needs merely 1% (131 ALUTs) of Stratix II's EP2S15F672I4 device logic units. WHR-1clk and WHR-2clk need 150 ALUTs and 151 ALUTs respectively, which are only 1.2% of the same device's logic.

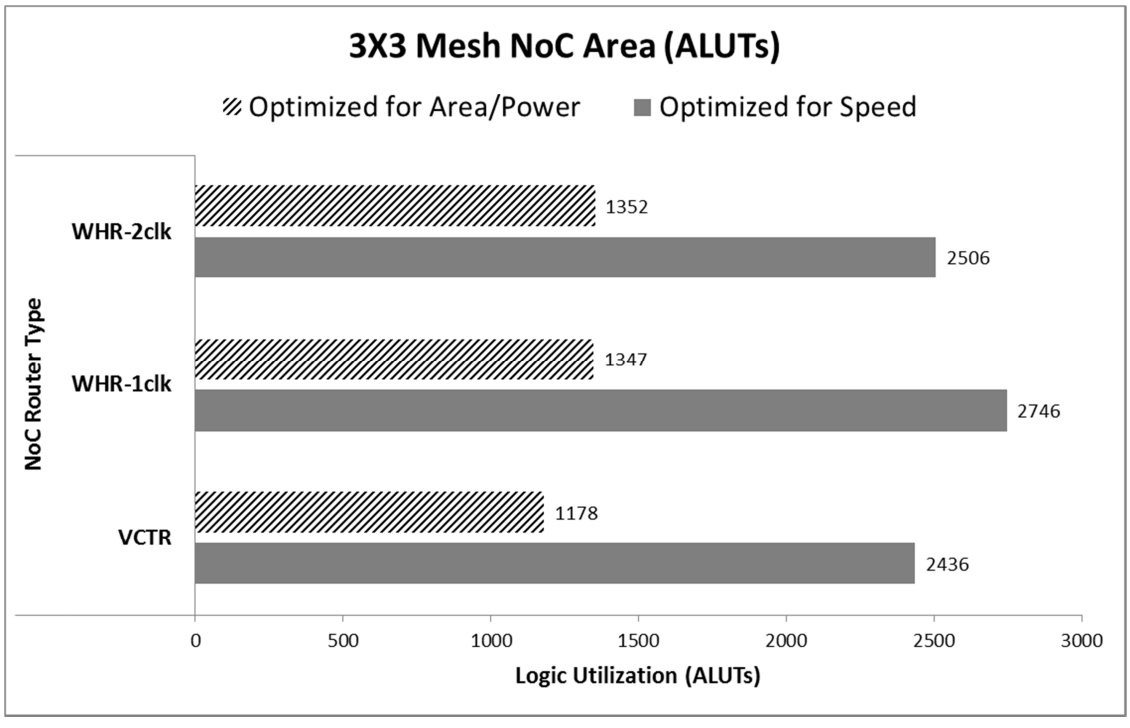


Figure 4.2 3X3 Mesh NoC Area for the Three Versions

The other important result of this area comparison is that augmenting the wormhole router with the dual-clock mechanism does not cost extra area. The stand-alone WHR-2clk version consumes marginally (0.8% and 3%) more area than WHR-1clk when the designs are optimized for area and speed respectively. However, the area of WHR-2clk NoC is (8%) less than the area of WHR-1clk NoC when the design is optimized for speed. When the NoC is optimized for area, there is only 0.3% difference between the two versions (WHR-2clk NoC consumes 1352 ALUTs and WHR-1clk NoC consumes 1347 ALUTs).

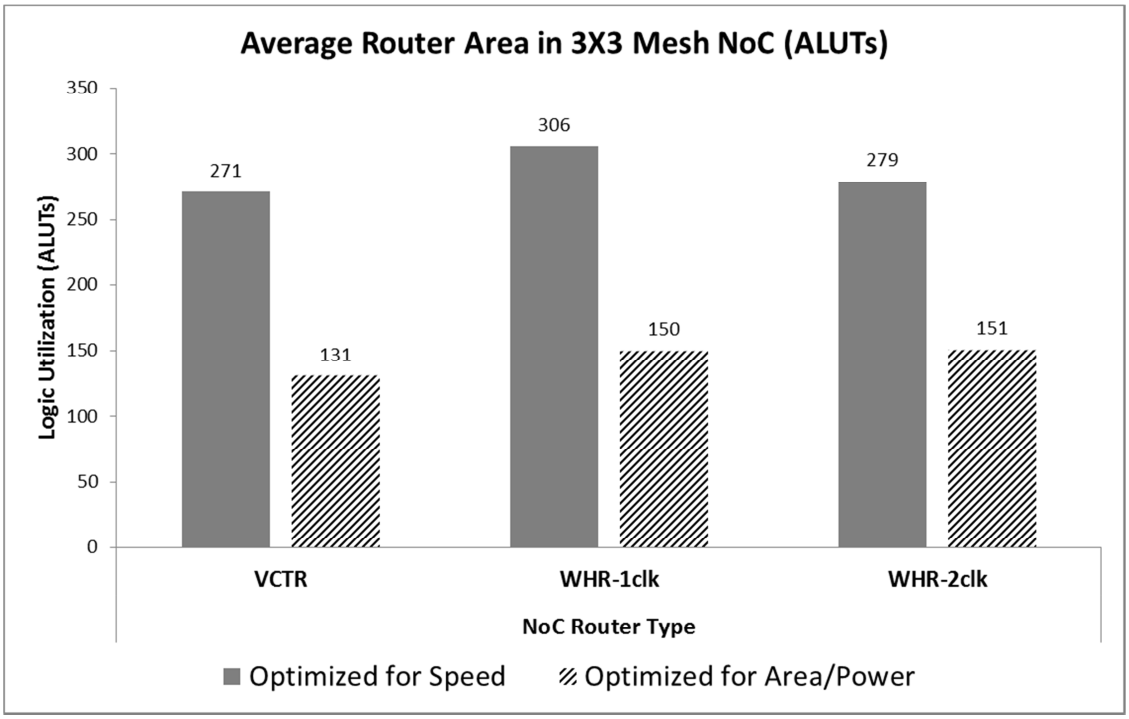


Figure 4.3 Average Router Area in 3X3 Mesh NoC for the Three Versions

4.2.1.2 Frequency

The attained maximum operating frequencies are very competitive and superior to most of the previous published FPGA-based NoC designs. Figure 4.4 compares the maximum operating frequencies of the three versions of the stand-alone router when the design is optimized for speed and area. The stand-alone VCTR can operate at 127.26 MHz by

optimizing the router for the speed. While the stand-alone WHR-2clk is slower than VCTR, it is faster than WHR-1clk version. Optimizing the routers for area degrades the frequency.

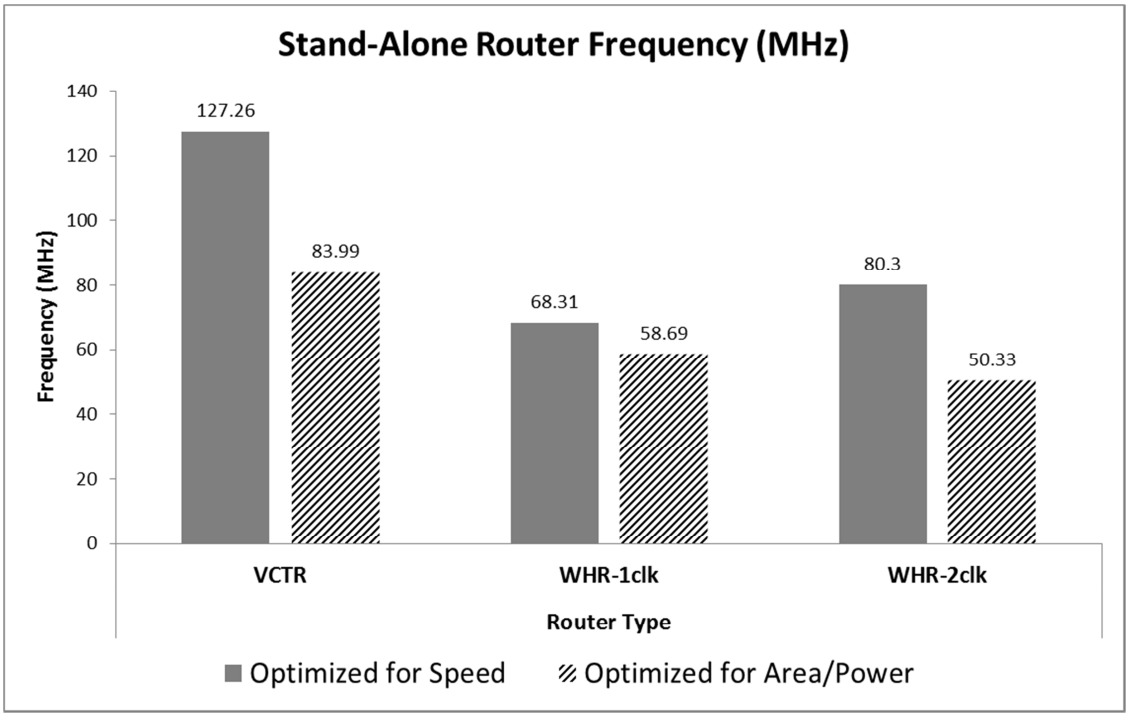


Figure 4.4 Stand-Alone Router Frequencies for the Three Versions

The frequency of the synthesized 3X3 Mesh NoC, for the three versions of the router, is shown in Figure 4.5. Mostly, there is a degradation in the overall maximum frequency, when we synthesize the whole network compared to the frequency of stand-alone routers. This degradation is considerable (40% to 50%) in the case of VCTR version, whereas it is only (7% - 10%) in the case of WHR-1clk version. For the dual-clock version (WHR-2clk) on the other hand, there is (11%) increase of the frequency when it is optimized for area; and (12%) degradation when the speed optimizer is guiding the synthesis tool. The 3X3 Mesh NoC of WHR-2clk can operate at about 70 MHz, which is faster than the NoC composed of the other two versions.

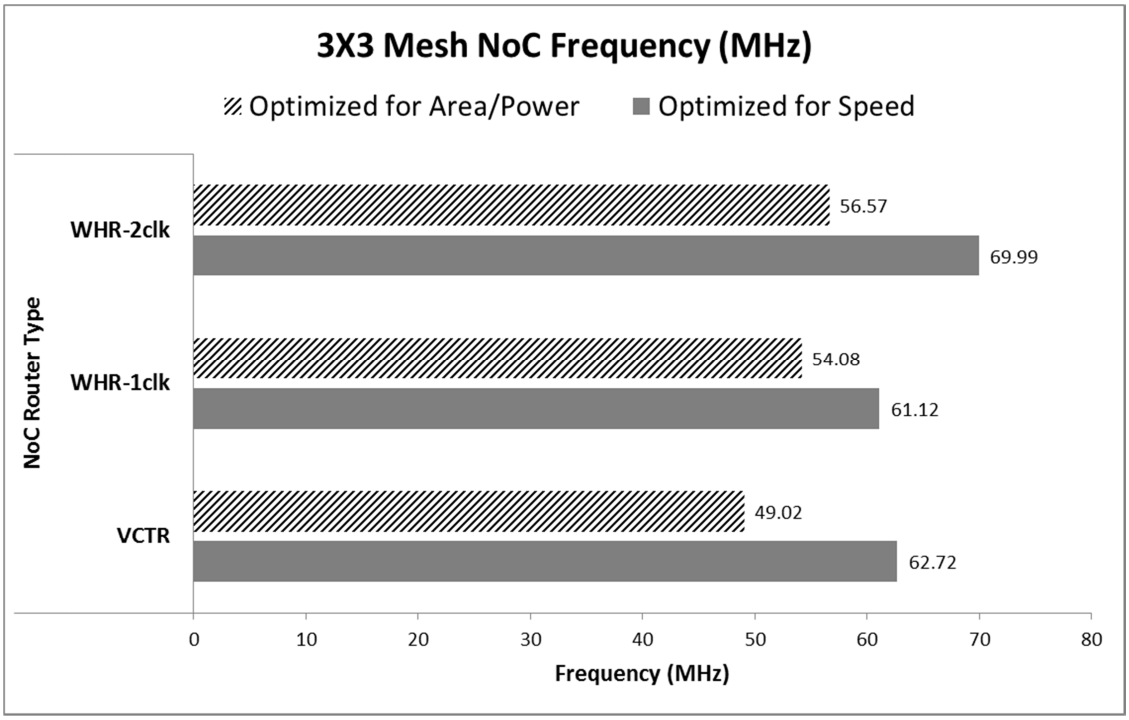


Figure 4.5 3X3 Mesh NoC Frequency for the Three Versions

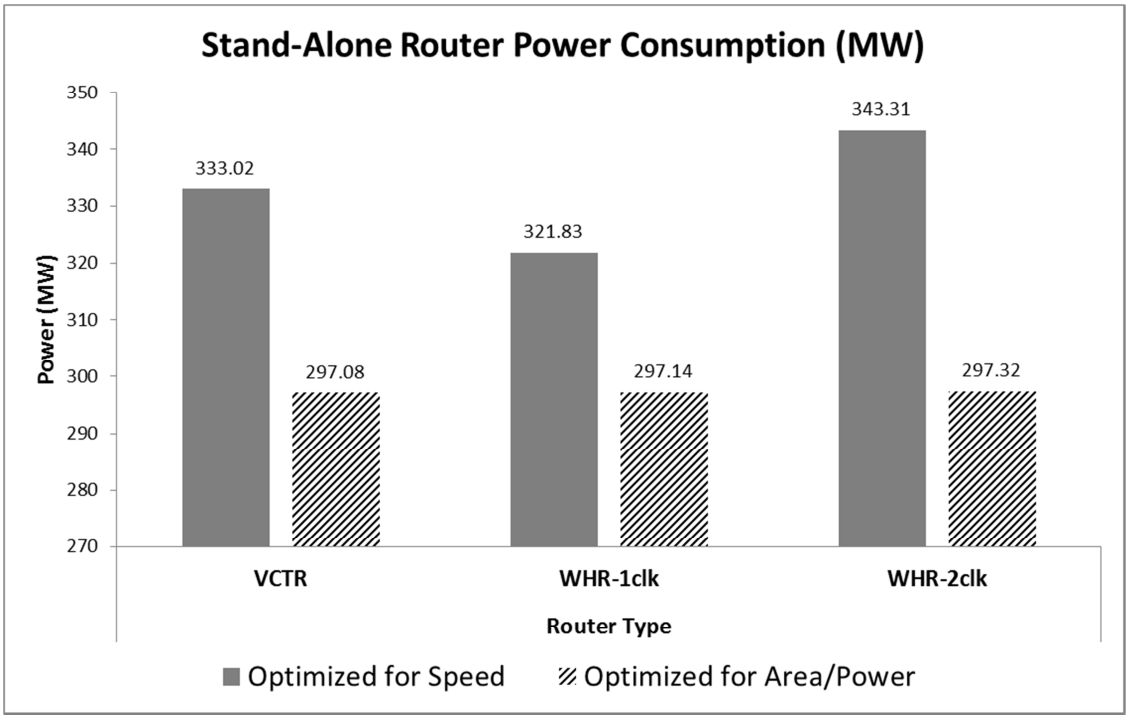


Figure 4.6 Stand-Alone Router Power Consumption for the Three Versions

4.2.1.3 Power Consumption

One of the optimization recommendations, provided by Altera Quartus II Power Advisor, is to optimize the design area, which is contrasted with frequency, in order to reduce the power consumption. Therefore, as it is displayed in Figure 4.6, power results are consistent with area results. For the same reason, NoC power consumption is a bit higher than the stand alone router because more logic is utilized. Figure 4.7 shows the power results of the 3X3 Mesh NoC for the three versions of the router.

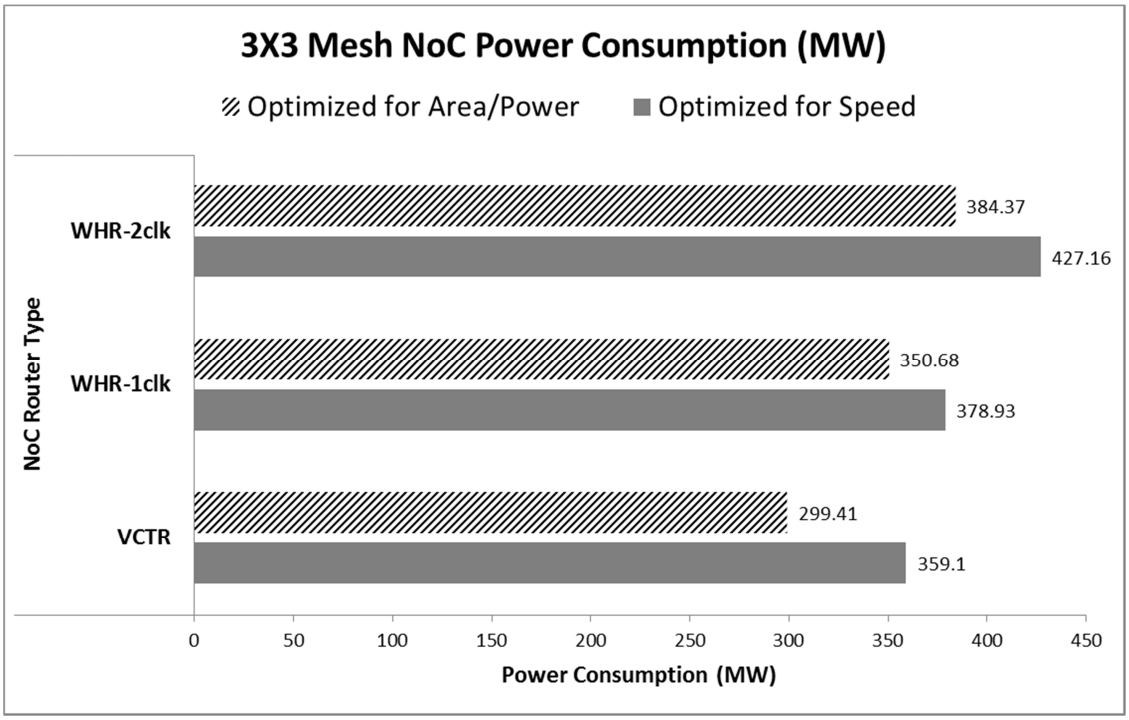


Figure 4.7 3X3 Mesh NoC Power Consumption for the Three Versions

From the previous synthesis results, we can summarize the results in the following points:

1. Optimizing the design for the speed increases the area and vice versa.
2. The router area decreased when it is configured into the NoC.
3. The area overhead of dual-clock augmentation is very marginal compared to the single clock router. Thus, the approach is feasible.

4. For the stand-alone router, VCTR has better frequency than WHRs, but consumes more area. On the other hand, after NoC configuration, WHR-NoCs have better frequency and bigger router area.
5. WHR-2clk can operate at a faster frequency than WHR-1clk in both cases of configuration (stand-alone and 3X3 Mesh NoC).
6. The power consumption is proportional to area for all designs.

4.2.2 Performance Calculation and Analysis

Although the simulation results can give some insight about the behavior of the NoC, they still cannot provide a fair comparison form for NoCs because different designs need different benchmarks. Also, they mostly depend on artificial traffic generators, not real applications. Throughout the literature review, we have not found a unified method to judge the performance of different NoC designs except the formulas used to obtain the maximum throughput and/or the zero load latency; such as the formulas presented in [1], [5], [22], [23], and [25].

As a performance metric, we chose to employ the zero load latency or the best case latency. It represents the lower bound on the average latency of a packet traversing the NoC. This latency is affected by path length, between the source and the destination, and the routing time inside the router of each node in the path.

The number of nodes in the path is determined by the topology and routing algorithm. Decreasing this number diminishes the latency. Also, and more importantly, minimizing the time spent inside the node for completing the routing decision not only reduces the latency, but also reduces the required buffer size; hence improves the area.

Theoretically, the router cannot accept new flits from the upstream router while waiting for the downstream port to be granted unless it has a sufficient number of buffer locations. This number must be greater than or equal to the number of possible flits that can be stored during the routing decision time, in order to achieve the best case latency. In our routers design the decision time is only one clock cycle; and the time needed to write a flit is only one clock cycle as well (Credit-based protocol has been used). Therefore, for the two designs of our wormhole router (WHR-1clk and WHR-2clk), only

one buffer location as well as the head flit location will be sufficient to achieve the minimum latency. However, the buffer size was made parameterized and chosen to be 8 to make a fair comparison with the VCTR version of the router and some other previous designs. Also, these extra buffer locations will be useful to handle the contention if needed. On the other hand, the VCTR's buffer size is already restricted to be big enough to accommodate the whole number of flits in the packet and it is also generically parameterized and chosen to be 8.

Generally, the time (*T*) needed to transport a packet from the source hop to the destination hop over the NoC can be calculated as follows:

$$T = R_d R_c + (P_s - 1) / TP \quad \ldots\ldots\ldots\ldots (4.1)$$

Where: R_d is the router delay, R_c is the routers count along the path from source to destination, P_s is the packet size, *-1* term is used to exclude the head flit which is counted with R_d, and TP is the port throughput.

If the NoC works without contention, the peak performance or the full speed can be reached. In this case the data processing rate utilizes the full capacity of the bandwidth. Thus, to calculate the best case latency T_{bc}, the throughput in equation 4.1 will be replaced by port bandwidth *BW* as following:

$$T_{bc} = R_d R_c + (P_s - 1) / BW \quad \ldots\ldots\ldots\ldots (4.2)$$

In order to evaluate the performance of our designs, formula (4.2) is used to calculate the best case latency T_{bc} of the NoC. T_{bc} can be expressed in nanoseconds if the operating frequency of the NoC is known. As an assessment case study for the NoC, R_c and P_s are assumed to be 3 hops and 8 flits respectively. This case study is visualized in Figure 4.8. The comparison with others is established by calculating the port's bandwidth (The maximum throughput) for each design; then calculating the best case latency based on the same case study.

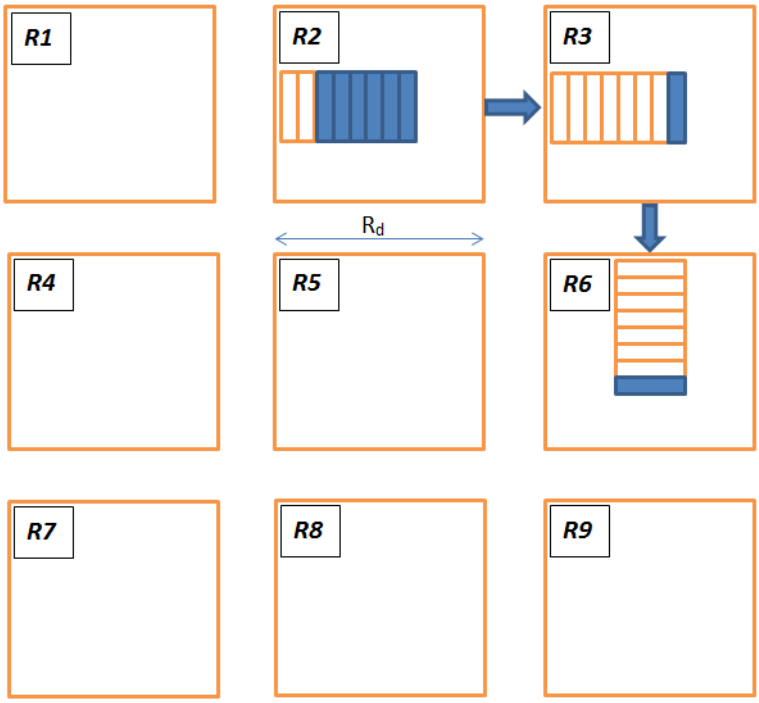


Figure 4.8 Zero Load Latency Case Study

4.2.3 Comparison with Previous Work

In addition to comparing the area and frequency, we calculated the zero load latency for our design and the compared designs based on the same assessment case study assumptions for R_c and P_s. The comparison is performed in two stages. First, we started our contrast by presenting a detailed comparison between our stand-alone VCTR and the results for the stand-alone router provided in MoCReS [1] because it uses the same switching technique. Table 4.1 explains the details of this comparison, while Figure 4.9 summarizes it. Second, by comparing the results of our WHR-1clk and the results obtained from ICN [5], Bartic [6] and Hermes [8]. These results are detailed in Table 4.2 and summarized in Fig 4.10.

From Table 4.1, one can infer that our router delay R_d consumes only one clock cycle, whereas, the competing design MoCReS, wastes seven clock cycles for the routing decision time R_d. Thus, even though the authors of MoCReS have reported the fastest

frequency throughout the reviewed literature for FPGA-based routers, which is (357 MHz), our VCTR still has a shorter zero load latency (T_{bc}) and smaller area.

Table 4.1 a Performance and Area Comparison with MoCReS Stand-Alone Routers

Design	**Flit Size**	**Flit/Cycle**	**F(MHz)**	**Port BW (Mbps)**	**R_d (Cycles)**	**T_{bc} (ns)**	**Area (ALUTs)**
MoCReS(mc)	8	1	357	2856	7	22	441
MoCReS(cc)	8	1	286	2288	7	27.53	412
VCTR(Op_S)	8	1	127	1018	1	14.70	590
VCTR(Op_A)	8	1	83	671	1	22.3	266

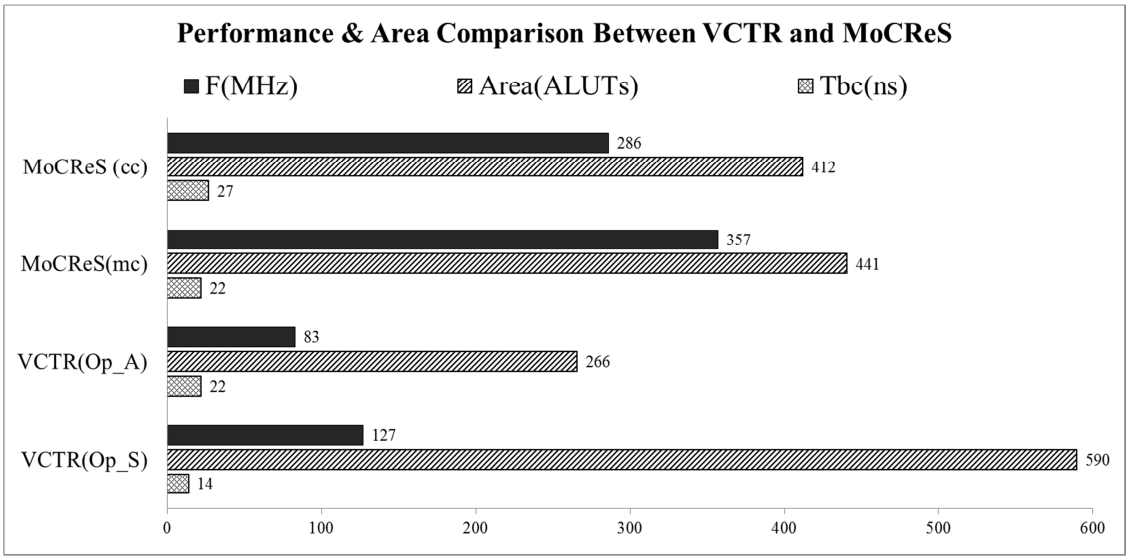


Figure 4.9 Performance and Area Comparison between VCTR and MoCReS Summary

MoCReS design has two versions. They are the common clock (cc), and the multiple clocks (mc). Both of them are included in this comparison along with our VCTR optimized for speed (Op_S), and VCTR optimized for area (Op_A). When we optimized

VCTR for speed to make it operate at its fastest possible frequency (127 MHz), the area increased to (590 ALUTs). This area is 25% higher than MoCReS (mc). However, it can be noticed from Fig 4.9 that VCTR's zero load latency is 36% lower than MoCReS.

On the other hand, optimizing our VCTR for area to get the smallest logic utilization (266 ALUTs), which is 40% less than MoCReS (mc), will operate at the same latency as MoCReS (mc).

MoCReS (cc) gives the worst latency results (27 ns); yet the area is still not much better than MoCReS (mc).

Table 4.2 a Performance and Area Comparison with some Previous Work

Design	**Flit Size**	**Flit/Cycle**	**F(MHz)**	**Port BW (Mbps)**	**R_d (Cycles)**	**T_{bc} (ns)**	**Area (ALUTs)**
HERMES[25]	8	0.5	25	100	10	1270	406
ICN[22]	16	0.5	40	320	2	172	326
Bartic [23]	16	1	50	800	3	189	807
WHR-1clk	8	1	54	435	1	71	150

The results of the second stage are even more positive. As evident from Table 4.2, our WHR-1clk significantly outperforms the other routers with lower area, higher frequency, and significantly reduced zero load latency. The area of WHR-1clk is 53% less than ICN [4], 63% less than Hermes [6], and 81% less than Bartic [5]. Yet, the frequency is 26%, 54%, and 8% correspondingly higher than these three designs. Furthermore, the number of clock cycles consumed to finish the routing decision R_d is only 1 cycle in our router, whereas it is 2, 10, and 3 cycles respectively, in ICN, Hermes, and [5]. This advantage of WHR-1clk along with its faster frequency, contributed to the reduction of the zero load latency. The other contributing factor is the use of the credit based flow control in our design that allows for transmitting each packet's body flit in only one clock cycle instead

of the two cycles as in ICN and Hermes. Therefore, WHR-1clk zero load latency is 59% less than ICN, 62% less than [5], and 94% less than Hermes.

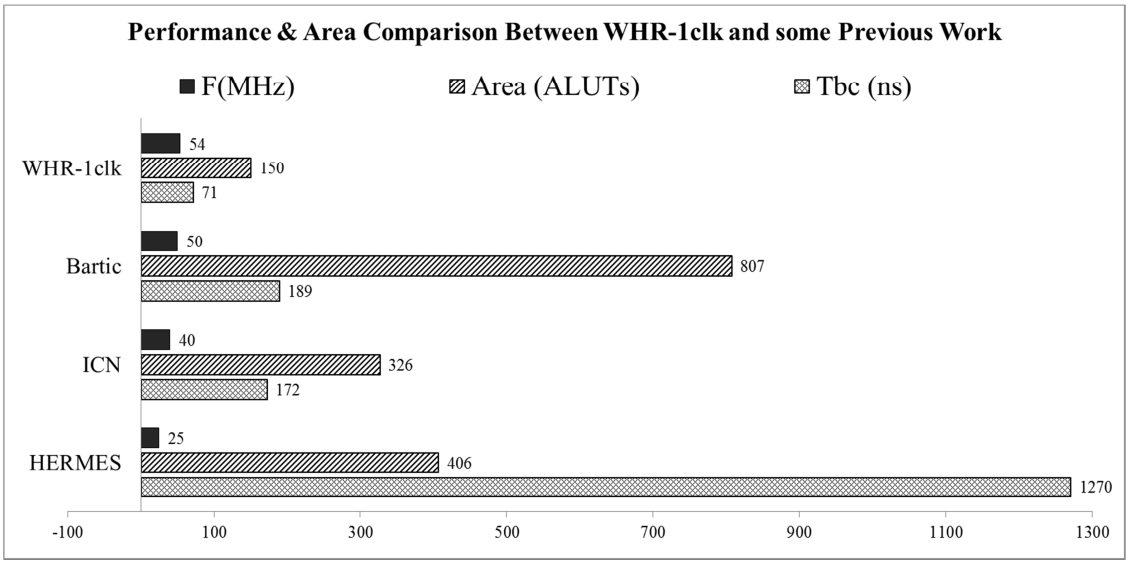


Figure 4.10 Performance and Area Comparison with some Previous Work Summary

4.3 Summary

In this chapter, we started by setting the evaluation methodology of our design, which includes choosing the test environment, and determining the assessment steps. Synthesis results for the three versions of our router and configured NoCs are obtained, compared, and analyzed; providing a wide exploration of NoC routers design space. Further, comparisons with previous work based on cost and performance are discussed and analyzed. These comparisons demonstrate the superiority of our routers, and prove the feasibility of our approach and the employed design strategies.

5 Chapter 5

Conclusion and Future Research Directions

With current FPGAs that deliver high bandwidth, high level of system integration, and convenient flexibility with reduced cost and low total power for high-end applications, FPSoCs are promising to provide reliable and practical digital systems for contemporary applications that require intensive computations and limited dimensions.

Since NoCs are a favorable approach to overcome the limitations of traditional bus-based and point-to-point on-chip communications used with SoCs, more research is needed to explore the design space of FPGA-based NoCs, and to offer more efficient solutions to existing NoC drawbacks.

This thesis tries to contribute to this research by exploring the design space of NoC routers, as a dominant component of the network, and introduces novel techniques that enhance the area of the router and boost its performance.

We implemented parameterized VHDL models of NoC-Routers on Stratix II FPGA. The focus of our approach is to minimize the area of the router because area is at a premium on an FPGA. At the same time, we introduced good techniques to speed up the router. Our router in general is a 5-ports packet switched router with deterministic XY-Routing algorithm and dynamic round robin arbitration scheme. The depth of input buffers, located at the input ports, and the flit size (channel width), are parameterized by means of VHDL generics. Three versions of our router were developed: Virtual Cut Through Router (VCTR), Wormhole Router with one clock (WHR-1clk), and dual-clock Wormhole Router (WHR-2clk). We presented the general architecture of the router as well as the architectural and functional differences between the three versions. The strategies that followed to decrease the area and increase the speed in the general router design are:

- Minimize the number of control fields in the packet as well as eliminate the tail flit, while replacing them with some logic and counters in the routing unit to infer the start and end of the packets, and consequently, reduce the FIFO size.
- Use the credit based flow control, so we are able to transmit the flit in only one clock cycle, instead of using the hand shaking protocol that needs at least two clock cycles. Also, we reduced the number of wires to one wire (Credit signal) as an alternative to the two wires (Request/Acknowledge Signals) in the case of hand shaking flow control.
- Utilize the rising edge of the clock to accomplish and synchronize some operations while employing the falling edge to undertake other operations. This technique enables us to squeeze a number of dependable operations in only one clock cycle either in the FIFO Finite State Machine (FSM) Read/Write operations, or arbitration and routing handling.
- Implement and explore the idea of augmenting the speed of the wormhole router using the dual-clock mechanism. This mechanism forwards the body flits of the packet at a faster clock speed instead of waiting for the slower clock of the head flit, resulting in compacting the effective length of body flits in the time domain.

The functionality of our router design was verified for the three versions using the Altera Quartus II simulator, and it was synthesized for the Altera Stratix II EP2S15F672I4 FPGA using the Quartus II synthesis tool. The simulation also demonstrates the feasibility of the dual-clock mechanism. Likewise, a 3X3 Mesh NoC using each version of our routers was synthesized for the same device. Comparison of the synthesis results before and after NoC configuration, among the three versions of the router, were conducted and discussed. VCTR and WHR-1clk are compared to other published routers based on three metrics: area, frequency and zero load latency. Synthesis results and zero load latency evaluations show that our router is significantly superior to widely referenced, previously proposed routers.

In short, we can summarize our research approach and achievements in the following points:

Approach:

- Design a parameterized NoC router with different switching modes (VCT, WH).
- Employ some novel techniques to reduce the area and speed up the router.
- Apply the idea of the dual-clock to the FPGA-based wormhole router for the first time and explore its effects.
- Prototype a 3X3 Mesh NoC topology for each router version by configuring 9 routers into the target FPGA.
- Explore and evaluate the design space including cost and performance trade-off for the stand-alone routers and after NoC configuration.
- Compare with widely cited previous work.

Achievements:

- Fast and light-weight router architecture for implementing FPGA-based NoC is presented, configured and tested for different modes.
- The feasibility of augmenting the speed of WHR with a dual-clock is demonstrated without area penalty.
- Compared to previous published FPGA-based NoC-Routers, our routers provide the shortest zero load latency and smallest area.

Future work:

- Develop our deterministic router to be an adaptive router that can provide quality of service.
- Use real applications or standard bench marks, if found in the future, to test our routers and compare them to other proposed work.
- Evaluate the reduction of the latency results from compacting the effective length of body flits in the time domain in the designed dual-clock wormhole router experimentally (via simulation), which was already demonstrated theoretically and functionally through the implementation and functional simulation.

- Compare the VCTR and WHR versions of our router using simulation and appropriate benchmarks in the case of tuning some parameters such as the flit size and the buffer size, which are already made parameterizable.
- Explore the accurate effects of our novel idea that was used in this research to minimize the number of control fields in the packet. The achieved low area demonstrates the effectiveness of this approach; however, the goal is to deeply analyze those enhancements from the following aspects:

    1. Calculate the amount of consumed FIFO buffer area, which is saved, especially when big size of buffering is needed.
    2. Assessing the latency and power consumption.
    3. Study the accuracy of the throughput (the actual delivered data).

Appendix

Full Reports of Synthesis Results

Flow Status Analyzed - Sun Mar 25 19:54:04 2012

Quartus II Version 9.0 Build 235 06/17/2009 SP 2 SJ Web Edition

Standalone VCTR Compilation Report

1. Optimized for Speed:

Revision Name Router

Top-level Entity Name Router

Family Stratix II

Device EP2S15F672I4

Timing Models Final

Met timing requirements Yes

Logic utilization 5 %

Combinational ALUTs 590 / 12,480 (5 %)

Dedicated logic registers 441 / 12,480 (4 %)

Total registers 441

Total pins 103 / 367 (28 %)

Total virtual pins 0

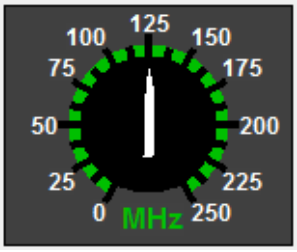
Total block memory bits 0 / 419,328 (0 %)

DSP block 9-bit elements 0 / 96 (0 %)

Total PLLs 0 / 6 (0 %)

Total DLLs 0 / 2 (0 %)

	Value
From	Arbiter:Arbiter_lb|next_state.Idle_741
To	Arbiter:Arbiter_lb|state.Idle
Clock period	7.858 ns
Frequency	127.26 MHz



Power Models Final

Total Thermal Power Dissipation 333.02 MW

Core Dynamic Thermal Power Dissipation 25.34 MW

Core Static Thermal Power Dissipation 272.92 MW

I/O Thermal Power Dissipation 34.77 MW

2. Optimized for Area/ Power Consumption

Revision Name Router

Top-level Entity Name Router

Family Stratix II

Device EP2S15F672I4

Timing Models Final

Met timing requirements Yes

Logic utilization 6 %

Combinational ALUTs 266 / 12,480 (2 %)

Dedicated logic registers 399 / 12,480 (3 %)

Total registers 399

Total pins 103 / 367 (28 %)

Total virtual pins 0

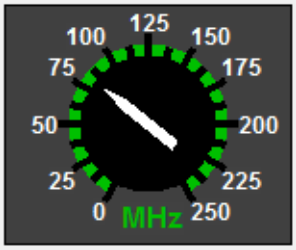
Total block memory bits 0 / 419,328 (0 %)

DSP block 9-bit elements 0 / 96 (0 %)

Total PLLs 0 / 6 (0 %)

Total DLLs 0 / 2 (0 %)

	Value
From	Arbiter:Arbiter_lb|latch[2]
To	FIFO_buffer:L_port|FIFO_CU:Ctrl_Unit|FSM:FSM_lb|state.s1
Clock period	11.906 ns
Frequency	83.99 MHz



Power Models Final

Total Thermal Power Dissipation 297.08 MW

Core Dynamic Thermal Power Dissipation 0.21 MW

Core Static Thermal Power Dissipation 272.59 MW

I/O Thermal Power Dissipation 24.28 MW

Standalone WHR-1clk Compilation Report

1. Optimized for Speed:

Revision Name Router

Top-level Entity Name Router

Family Stratix II

Device EP2S15F672I4

Timing Models Final

Met timing requirements Yes

Logic utilization 5 %

Combinational ALUTs 551 / 12,480 (4 %)

Dedicated logic registers 371 / 12,480 (3 %)

Total registers 371

Total pins 103 / 367 (28 %)

Total virtual pins 0

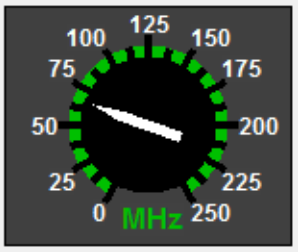
Total block memory bits 0 / 419,328 (0 %)

DSP block 9-bit elements 0 / 96 (0 %)

Total PLLs 0 / 6 (0 %)

Total DLLs 0 / 2 (0 %)

	Value
From	Arbiter:Arbiter_lb|latch[0]
To	Arbiter:Arbiter_lb|\packet_track:counter[3]
Clock period	14.640 ns
Frequency	68.31 MHz



Power Models Final

Total Thermal Power Dissipation 321.83 MW

Core Dynamic Thermal Power Dissipation 18.15 MW

Core Static Thermal Power Dissipation 272.82 MW

I/O Thermal Power Dissipation 30.86 MW

2. Optimized for Area/Power Consumption

Revision Name Router

Top-level Entity Name Router

Family Stratix II

Device EP2S15F672I4

Timing Models Final

Met timing requirements Yes

Logic utilization 5 %

Combinational ALUTs 239 / 12,480 (2 %)

Dedicated logic registers 371 / 12,480 (3 %)

Total registers 371

Total pins 103 / 367 (28 %)

Total virtual pins 0

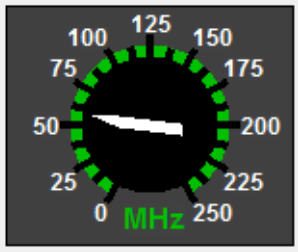
Total block memory bits 0 / 419,328 (0 %)

DSP block 9-bit elements 0 / 96 (0 %)

Total PLLs 0 / 6 (0 %)

Total DLLs 0 / 2 (0 %)

	Value
From	Arbiter:Arbiter_lb|state.E
To	Arbiter:Arbiter_lb|latch[0]
Clock period	17.040 ns
Frequency	58.69 MHz



Power Models Final

Total Thermal Power Dissipation 297.14 MW

Core Dynamic Thermal Power Dissipation 0.24 MW

Core Static Thermal Power Dissipation 272.59 MW

I/O Thermal Power Dissipation 24.31 MW

Standalone WHR-2clk Compilation Report

1. Optimized for Speed:

Revision Name Router

Top-level Entity Name Router

Family Stratix II

Device EP2S15F672I4

Timing Models Final

Met timing requirements Yes

Logic utilization 5 %

Combinational ALUTs 571 / 12,480 (5 %)

Dedicated logic registers 383 / 12,480 (3 %)

Total registers 383

Total pins 106 / 367 (29 %)

Total virtual pins 0

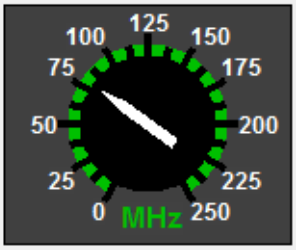
Total block memory bits 0 / 419,328 (0 %)

DSP block 9-bit elements 0 / 96 (0 %)

Total PLLs 0 / 6 (0 %)

Total DLLs 0 / 2 (0 %)

	Value
From	Arbiter:Arbiter_lb|latch[2]
To	Arbiter:Arbiter_lb|\packet_track:counter[3]
Clock period	12.454 ns
Frequency	80.30 MHz



Power Models Final

Total Thermal Power Dissipation 343.31 MW

Core Dynamic Thermal Power Dissipation 25.68 MW

Core Static Thermal Power Dissipation 273.01 MW

I/O Thermal Power Dissipation 44.62 MW

2. Optimized for Area/ Power Consumption

Revision Name Router

Top-level Entity Name Router

Family Stratix II

Device EP2S15F672I4

Timing Models Final

Met timing requirements Yes

Logic utilization 5 %

Combinational ALUTs 241 / 12,480 (2 %)

Dedicated logic registers 372 / 12,480 (3 %)

Total registers 372

Total pins 106 / 367 (29 %)

Total virtual pins 0

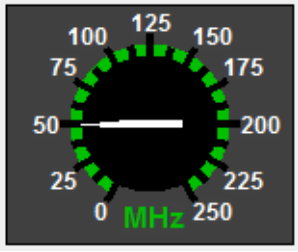
Total block memory bits 0 / 419,328 (0 %)

DSP block 9-bit elements 0 / 96 (0 %)

Total PLLs 0 / 6 (0 %)

Total DLLs 0 / 2 (0 %)

	Value
From	Arbiter:Arbiter_lb|state.S
To	Arbiter:Arbiter_lb|latch[0]
Clock period	19.868 ns
Frequency	50.33 MHz



Power Models Final

Total Thermal Power Dissipation 297.32 MW

Core Dynamic Thermal Power Dissipation 0.29 MW

Core Static Thermal Power Dissipation 272.59 MW

I/O Thermal Power Dissipation 24.44 MW

3X3 Mesh of VCTR Compilation Report

1. Optimized for Speed:

Revision Name NoC_VCT

Top-level Entity Name NoC_VCT

Family Stratix II

Device EP2S15F672I4

Timing Models Final

Met timing requirements Yes

Logic utilization 25 %

Combinational ALUTs 2,436 / 12,480 (20 %)

Dedicated logic registers 1,852 / 12,480 (15 %)

Total registers 1852

Total pins 155 / 367 (42 %)

Total virtual pins 0

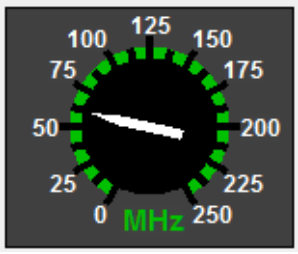
Total block memory bits 0 / 419,328 (0 %)

DSP block 9-bit elements 0 / 96 (0 %)

Total PLLs 0 / 6 (0 %)

Total DLLs 0 / 2 (0 %)

	Value
From	Router6:hop6|Arbiter:Arbiter_lb|latch[0]
To	Router5:hop5|FIFO_buffer:E_port|FIFO_DP:Data_Path|shift_...
Clock period	15.944 ns
Frequency	62.72 MHz



Power Models Final

Total Thermal Power Dissipation 359.10 MW

Core Dynamic Thermal Power Dissipation 58.70 MW

Core Static Thermal Power Dissipation 273.16 MW

I/O Thermal Power Dissipation 27.25 MW

2. Optimized for Area/ Power Consumption

Revision Name NoC_VCT

Top-level Entity Name NoC_VCT

Family Stratix II

Device EP2S15F672I4

Timing Models Final

Met timing requirements Yes

Logic utilization 22 %

Combinational ALUTs 1,178 / 12,480 (9 %)

Dedicated logic registers 1,687 / 12,480 (14 %)

Total registers 1687

Total pins 155 / 367 (42 %)

Total virtual pins 0

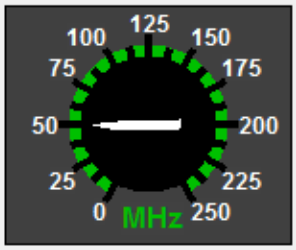
Total block memory bits 0 / 419,328 (0 %)

DSP block 9-bit elements 0 / 96 (0 %)

Total PLLs 0 / 6 (0 %)

Total DLLs 0 / 2 (0 %)

	Value
From	Router6:hop6|Arbiter:Arbiter_lb|latch[0]
To	Router3:hop3|FIFO_buffer:S_port|FIFO_DP:Data_Path|shift_...
Clock period	20.400 ns
Frequency	49.02 MHz



Power Models Final

Total Thermal Power Dissipation 299.41 MW

Core Dynamic Thermal Power Dissipation 0.71 MW

Core Static Thermal Power Dissipation 272.61 MW

I/O Thermal Power Dissipation 26.09 MW

Power Estimation ConfidenceLow: user provided insufficient toggle rate data

3X3 Mesh of WHR-1clk Compilation Report

1. Optimized for Speed:

Revision Name NoC_WH_1clk

Top-level Entity Name NoC_WH_1clk

Family Stratix II

Device EP2S15F672I4

Timing Models Final

Met timing requirements Yes

Logic utilization 28 %

Combinational ALUTs 2,746 / 12,480 (22 %)

Dedicated logic registers 1,927 / 12,480 (15 %)

Total registers 1927

Total pins 165 / 367 (45 %)

Total virtual pins 0

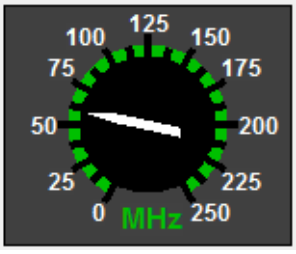
Total block memory bits 0 / 419,328 (0 %)

DSP block 9-bit elements 0 / 96 (0 %)

Total PLLs 0 / 6 (0 %)

Total DLLs 0 / 2 (0 %)

	Value
From	Router4:hop4|Arbiter:Arbiter_lb|latch[2]
To	Router7:hop7|FIFO_buffer:N_port|FIFO_DP:Data_Path|shift_...
Clock period	16.360 ns
Frequency	61.12 MHz

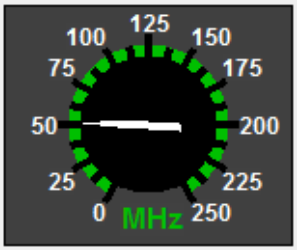


Power Models	Final
Total Thermal Power Dissipation	378.93 MW
Core Dynamic Thermal Power Dissipation	76.69 MW
Core Static Thermal Power Dissipation	273.34 MW
I/O Thermal Power Dissipation	28.90 MW

2. Optimized for Area/ Power Consumption

Revision Name	NoC_WH_1clk
Top-level Entity Name	NoC_WH_1clk
Family	Stratix II
Device	EP2S15F672I4
Timing Models	Final
Met timing requirements	Yes
Logic utilization	24 %
Combinational ALUTs	1,347 / 12,480 (11 %)
Dedicated logic registers	1,770 / 12,480 (14 %)
Total registers	1770
Total pins	165 / 367 (45 %)
Total virtual pins	0
Total block memory bits	0 / 419,328 (0 %)
DSP block 9-bit elements	0 / 96 (0 %)
Total PLLs	0 / 6 (0 %)
Total DLLs	0 / 2 (0 %)

	Value
From	Router5:hop5|Arbiter:Arbiter_lb|latch[2]
To	Router6:hop6|FIFO_buffer:W_port|FIFO_DP:Data_Path|shift...
Clock period	18.490 ns
Frequency	54.08 MHz



Power Models Final

Total Thermal Power Dissipation 350.68 MW

Core Dynamic Thermal Power Dissipation 48.88 MW

Core Static Thermal Power Dissipation 273.08 MW

I/O Thermal Power Dissipation 28.72 MW

3X3 Mesh of WHR-2clk Compilation Report

1. Optimized for Speed:

Revision Name NoC_WH_2clk

Top-level Entity Name NoC_WH_2clk

Family Stratix II

Device EP2S15F672I4

Timing Models Final

Met timing requirements Yes

Logic utilization 26 %

Combinational ALUTs 2,506 / 12,480 (20 %)

Dedicated logic registers 1,580 / 12,480 (13 %)

Total registers 1580

Total pins 158 / 367 (43 %)

Total virtual pins 0

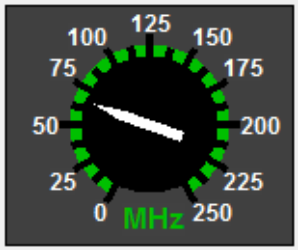
Total block memory bits 0 / 419,328 (0 %)

DSP block 9-bit elements 0 / 96 (0 %)

Total PLLs 0 / 6 (0 %)

Total DLLs 0 / 2 (0 %)

	Value
From	Router6:hop6|Arbiter:Arbiter_lb|next_state.W_500
To	Router6:hop6|Arbiter:Arbiter_lb|state.W
Clock period	14.288 ns
Frequency	69.99 MHz



Power Models Final

Total Thermal Power Dissipation 427.16 MW

Core Dynamic Thermal Power Dissipation 121.43 MW

Core Static Thermal Power Dissipation 273.77 MW

I/O Thermal Power Dissipation 31.95 MW

2. Optimized for Area/ Power Consumption

Revision Name NoC_WH_2clk

Top-level Entity Name NoC_WH_2clk

Family Stratix II

Device EP2S15F672I4

Timing Models Final

Met timing requirements Yes

Logic utilization 23 %

Combinational ALUTs 1,352 / 12,480 (11 %)

Dedicated logic registers 1,708 / 12,480 (14 %)

Total registers 1708

Total pins 158 / 367 (43 %)

Total virtual pins 0

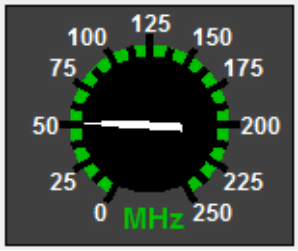
Total block memory bits 0 / 419,328 (0 %)

DSP block 9-bit elements 0 / 96 (0 %)

Total PLLs 0 / 6 (0 %)

Total DLLs 0 / 2 (0 %)

	Value
From	Router4:hop4|Arbiter:Arbiter_lb|latch[1]
To	Router1:hop1|FIFO_buffer:S_port|FIFO_DP:Data_Path|shift_...
Clock period	18.250 ns
Frequency	54.79 MHz



Power Play Power Analyzer Status Successful - Sun Mar 25 16:21:53 2012

Quartus II Version 9.0 Build 235 06/17/2009 SP 2 SJ Web Edition

Revision Name NoC_WH_2clk

Top-level Entity Name NoC_WH_2clk

Family Stratix II

Device EP2S15F672I4

Power Models Final

Total Thermal Power Dissipation 384.37 MW

Core Dynamic Thermal Power Dissipation 80.74 MW

Core Static Thermal Power Dissipation 273.38 MW

I/O Thermal Power Dissipation 30.25 MW

References

[1] A. Janarthanan, V. Swaminathan and K. Tomko, "MoCReS: an Area-Efficient Multi-Clock On-Chip Network for Reconfigurable Systems," *Proc. ISVLSI*, 2007, pp. 455-456.

[2] R. Marculescu et al, "Outstanding Research Problems in NoC Design: System, Microarchitecture, and Circuit Perspectives, " *IEEE Trans. on CAD of ICs and Systems*, vol. 28, no. 1, January 2009.

[3] M. Brugge. "Design and Evaluation of a Parameterizable NoC Router for FPGAs." MSc thesis, University of Windsor, Canada, 2009.

[4] A. Janathanan. "Networks-on-Chip Based High Performance Communication Architectures for FPGAs." Ph.D. dissertation, University of Cincinnati, USA, 2008.

[5] S. Lee and N. Bagherzadeh, "Increasing the Throughput of an Adaptive Router in Network-on-Chip (NoC)," *Proc. CODES+ISSS*, 2006, pp. 82 – 87.

[6] Altera Inc. Internet: http://www.altera.com.

[7] Steve Furber. "Future Trends in SoC Interconnect, Design & Reuse (D&R)." Internet: http://www.design-reuse.com/articles/14612/future-trends-in-soc-interconnect.html, [Jan 23, 2012].

[8] L. Benini, G. Micheli, "Network on chips: a new soc paradigm," *Computer*, Vol. 35, Issue 1, Jan. 2002.

[9] C. Hilton and B. Nelson, "PNoC: a Flexible Circuit Switched NoC for FPGA-Based Systems," *Proc. IEE Computers and Digital Techniques*, 2005, pp. 181-188.

[10] T. Bjerregaard and S. Mahadevan, "A Survey of Research and Practices of Network-on-Chip," *ACM Computing Surveys (CSUR)*, Vol. 38, Issue 1, 2006.

[11] W. Tsai, Y. Lan, Y. Hu, and S. Chen, "Networks on Chips: Structure and Design Methodologies, " *Journal of Electrical and Computer Engineering*, Vol. 2012, Article ID 509465, 15 pages, 2012.

[12] W. Dally and C. Seitz, "Deadlock-Free Message Routing in Multiprocessor Interconnection Networks," *IEEE Trans. Computers*, Vol. C-36, Issue 5, pp. 547-553, May 1987.

[13] "Topologies." http://www.computertipsntricks.com/nt_topologies.htm, Feb. 2009, [Jan. 25, 2012].

[14] J. Suseela. "Parameterizable Network-on-Chip Emulation Framework," MSc thesis, University of Nevada, USA, 2010.

[15] L. Ni and P. McKinley, "A Survey of Wormhole Routing Techniques in Direct Networks," *IEEE Computer Magazine*, Vol. 26 (2), pp. 62-76, Feb. 1993.

[16] C. Zeferino, A. Susin, "SoCIN: A Parametric and Scalable Network-on-Chip," *Proc. of Integrated Circuits and Systems Design*, pp. 169-174, Sep. 2003.

[17] "Understanding Parallel Hardware: Multiprocessors, Hyperthreading, Dual-Core, Multicore and FPGAs." http://zone.ni.com/devzone/cda/tut/p/id/6097, Dec. 2011, [Feb. 15, 2012].

[18] "Logic Structure Comparison Between Stratix II and Virtex-Based Architectures." Altera White Paper, WP-STXIIXLNX-1.0, Internet:http://www.altera.ru/Disks/Altera%20Documentation%20Library/literature/wp/wpstxiixlnx.pdf, Feb. 2004 [Jan. 19, 2012].

[19] Xilinx Inc. Internet: http://www.xilinx.com/.

[20] "Stratix II vs. Virtex-4 Performance Comparison, Altera White Paper." WP-S2052505-2.0, Internet: http://www.altera.co.jp/literature/wp/wp_s2v4perf.pdf, Sep. 2006, [Jan. 19, 2012].

[21] Steve Sharp. "Virtex-4 Dynamic Power Comparison - A Case Study." Xilinx Corporate Solutions Marketing, Internet: http://www.xilinx.com/products/silicon_solutions/fpgas/virtex/virtex4/resources/Virtex-4_Power_Case_Study.pdf, Dec. 2005, [Jan. 19, 2012].

[22] T. Marescaux, T. Bartic, D. Verkest, S. Vernalde, R. Lauwereins," Interconnection Networks Enable Fine-Grain Dynamic Multi-Tasking on FPGAs," *Proc. FPL*, 2002, pp. 795-805.

[23] T. Bartic, J-Y. Mignolet, V. Nollet, T. Marescaux, D. Verkest, S. Vemalde, R. Lauwereins, "Highly Scalable Network on Chip for Reconfigurable Systems," *Proc. IEEE International Symposium on System-on-Chip*, 2003, pp. 79-82.

[24] C. Zeferino, M. Kreutz, A. Susin, "RASoC: a Router Soft-Core for Networks-on-Chip," *Proc. DATE*, 2004, pp. 198-205.

[25] F. Moraes, A. Mello, L. Möller, L. Ost, N. Calazans, "Hermes: an Infrastructure for Low Area Overhead Packet-Switching Networks on Chip," *Integration, the VLSI Journal*, Vol. 38, pp. 69-93, Oct. 2004.

[26] L. Ost, A. Mello, J. Palma, F. Moraes, N. Calazans, "MAIA: a Framework for Networks on Chip Generation and Verification," *Proc. ASP-DAC*, 2005, pp. 49-52.

[27] B. Sethuraman, P. Bhattacharya, J. Khan and R.Vemuri, "LiPaR: a Light-weight Parallel Router for FPGA-Based Networks-on-Chip," *Proc. GLSVLSI*, 2005, pp. 452-457.

[28] B. Sethuraman, "Novel Methodologies for Performance & Power Efficient Reconfigurable Networks-on-Chip," *Proc. FPL*, 2006, pp.1-2.

[29] B. Sethuraman and R. Vemuri, "Multi2 Router: a Novel Multi Local Port Router Architecture with Broadcast Facility for FPGA Based Networks-on-Chip," *Proc. FPL*, 2006, pp. 1-4.

[30] B. Sethuraman and R. Vemuri, "OptiMap: a tool for automated generation of NoC architectures using multiport routers for FPGAs," *Proc. DATE*, 2006, pp. 6-11.

[31] M. Vestias and H. Neto, "Area and Performance Optimization of a Generic Network-on-Chip Architecture," *Proc. SBCCI*, 2006, pp. 68-73.

[32] Y. Lu, J. McCanny and S. Sezer, "Generic Low-Latency NoC Router Architecture for FPGA Computing Systems," *Proc. FPL*, 2011, pp. 82-89.

[33] N. Kapre. "Packet-Switched On-Chip FPGA Overlay Networks," MSc thesis, California Institute of Technology, USA, 2006.

VITA AUCTORIS

Abdelrazag Imbewa was born in 1976 in Al-Khoms, Libya. He graduated from Al-Rifai High School in 1994. From there he went on to the Higher Institute of Industry, Misurata, Libya where he obtained a B.Sc. in Computer Engineering in 1999. He is currently a candidate for the Master's degree in Electrical and Computer Engineering at the University of Windsor and hopes to graduate in Winter 2012.