Investigating The Effect of Virtual Channel on The Performance of Network-on-Chip

by

Adnan Ahmad
B.E., Sarhad University of Science and Information Technology, Peshawar, 2013

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF APPLIED SCIENCE

in the Department of Electrical and Computer Engineering

Investigating The Effect of Virtual Channel on The Performance of Network-on-Chip

by

Adnan Ahmad
B.E., Sarhad University of Science and Information Technology, Peshawar, 2013

Supervisory Committee

Dr. Fayez Gebali,, Supervisor
(Department of Electrical and Computer Engineering)

Dr. Samer Moein, Departmental Member
(Department of Electrical and Computer Engineering)

ABSTRACT

Network-on-Chip(NoC) is the communication backbone in domain of of multi-core processor systems. As the number of cores in conventional bus based architecture is increasing communication techniques are becoming ineffective and complex. Wormhole flow control is the most commonly used flow control algorithm in NoC. However as the work load is increased in the NoC wormhole flow control causes head-of-line blocking which results in contention for the physical channel. This issue can be resolved by using virtual channel flow control. In this work we investigate the effect of input-queued Virtual Channels router model on the performance of NoC by varying differnt parametrs like injection rate and the packet length. We simulate K-ary-n cubes mesh topology with dimension order routing (DOR) under synthetic workloads in order to find the effect of virtual channels on the performance of Mesh network in term of throughput and latency. We show that as the number of virtual channels is increased there is an improvement in the throughput and latency of the network up to a certain number of virtual channels beyond which the network reaches saturated state. Our work can be used as a guidance to find the optimal number of virtual channels for a given NoC configuration and traffic parameters.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

First of all, I would like to thank Allah, the most Gracious and Merciful, for giving me the patience, courage, intelligence, love and strength to make possible this work. I am really indebted to my Creator for being a great teacher and mentor. I would like to express my sincere gratitude and appreciation to my supervisor, Dr. Fayez Gebali, for giving me the opportunity to work under his supervision in the field of Network–on–Chip and for his continuous support, patience, encouragement, always availability even on weekend and self fostering research environment provided during my research work. I would present thanks to Dr. Sudhakar Ganti from UVIC Computer Science department and Dr.Samer Moein from UVIC Electrical department, for their valuable feedback while serving as my thesis advisory committee members.
A very special thanks also goes to my many friends (you know who you are). Without you guys this thesis would have been much less exciting. I would like to thank my parents along with my brothers and sisters. Thank you for all your support and your continued love and encouragement over all these years.

I believe I know the only cure, which is to make one's centre of life inside of one's self, not selfishly or excludingly, but with a kind of unassailable serenity-to decorate one's inner house so richly that one is content there, glad to welcome any one who wants to come and stay, but happy all the same in the hours when one is inevitably alone.

Edith Wharton

DEDICATION

For my parents
Fazal Ghani and Bahrul Haya
especially my brothers and sisters !

Chapter 1

Introduction

1.1 Introduction

Conventional bus-based interconnection architecture failed to provide the required bandwidth, power consumption, latency and an effective communication for message transmission. As the number of processing element(PEs) increases on a single chip an effective communication among different Procesing element (PEs) over network on chip become critical issue. Therefore the researchers presented a new concept of high-performance, scalable and flexible interconnection network named NoC in order to solve such communication bottleneck among processing elements (PEs) on a single chip. With the development of multi core architecture noc provide an excellent scalability, higher throughput and lower network latency. There are several properties that can improve the performance of network such as network topology, flow control strategy, router architecture, resource allocation management and routing algorithm. In this work we focus on flow control strategy. There are two most commonly used flow control algorithms known as wormhole flow control and virtual channel flow control. In wormhole flow control each message is divided into packet and each packet is further divided into header flit, body flit and tail flit. In NoC header flit hold routing information of a packet and size of the message as well [2]. When the header flit advances along the specified path the body flit follows it in a pipeline way. However when the downstream router receives the tail flit of a packet then the path becomes free for next coming packet header flit. In case the incomming header flit found the link is already in use so in that situation rest of the body flits would be blocked along the path until the link becomes free and this phenomenon is known as Head- of-line

blocking. In wormhole flow control as the traffic increases packets may experience large delay to cross the network due to blocked channel as a result of the contention for the physical channel. Due to head-of-line blocking and deadlock challenges wormhole flow control degrades the performance of NoC [3]. In order to resolve this issues virtual channel flow algorithm have been used which brought a significant improvement in the performance of NOC. Virtual channel flow algorithm determines that how different resources such as buffer and channel bandwidth are allocated to a routing packet and how to resolve packet collision issue.

To avoid head-of-line blocking and deadlock challenges there are several approaches were carried out for the performance and evaluation of NoC. System on chip are unable to integrate large scale multi core communication efficiently. Dally proposed an analytical model under the constant buffer storage per node for torus network using Deadline scheduling scheme [6]. Modeli et al. presented an analytical model for the performance of spidergon NoC [5]. Huang et al. In [6] proposed a novel based approach for customizing the virtual channels allocation based on the traffic characteristics of the target application. Minghua Tang and Xiaola et al. [41] proposed a novel flow control approach for the performance evaluation of NoC and conducted simulation based on it accurate and open source System C simulator. Paliwal et al. in [42]. proposed a scheme to compare and evaluate the performance of guaranteed throughput and best effort traffic in noc under different traffic patterns.

1.2 Contributions

Main contributions of this thesis could be summarized as follows.

- Simulate the performance of virtual channel router model based on DOR(Dimension order algorithm).
- We investigate the effect of virtual channel with respect to packet length in 2D mesh network.

The remainder of this dissertation is organized as follows: chapter 2 discusses Overview Network-on-chip, switching techniques, packet format and wormhole router. In chapter 3 we discussed input-queued virtual channel router model, basic concept of arbitration in NoC and comparison of traffic patterns. Similarly in chapter 4 we investigate

the effect of virtual channel model using Booksim2 NoC simulator. In last chapter 5 we conclude our contributions and briefly outlines opportunities for future work.

Chapter 2

Overview Network on Chip

Networks-on-Chip (NoCs) are distributed communication infrastructures, consisting of routers, that are interconnected by point-to-point links. The computation cores, memories and I/O interfaces of system are connected to the routers via network interfaces (also called network adapters). Usually one core is connected to a router, but there are also network topologies or architectures in which multiple cores are connected to one router.

2.1 Bus Based Interconnect Architecture

System-on-chip or system on chip (SoC) refers to implementing all components of a computer or other electronic system into a single integrated circuit (chip). A typical SoC may contain digital, analogue, mixed-signal, and often radio-frequency functions all on a single chip substrate [33]. A typical SoC consists of:

- One micro-controller, microprocessor or DSP core(s). Some SoCs called multiprocessor System-on-Chip (MPSoC) include more than one processor core.
- Memory blocks including a selection of ROM, RAM, EEPROM and flash.
- Timing sources including oscillators and phase-locked loops.
- Peripherals including counter-timers, real-time timers and power-on reset generators.
- External interfaces including industry standards such as USB, FireWire, Ethernet, USART, SPI.

- Analogue interfaces including ADCs and DACs.
- Regulators and power management circuits [33].

These blocks communicate through an industry-standard bus such as the AMBA bus from ARM. DMA controllers were routed data directly between external interfaces and memory register, by-passing the processor core and thereby increasing the data throughput of the SoC [34]. A SoC composed of both the software and hardware, mentioned above, software used for controlling the microcontroller, microprocessor or DSP cores, peripherals and interfaces. The design flow for a SoC aims to develop this hardware and software in parallel[34].

Table 2.1: Comparison of NoC and Bus architecture[36].

NoC	Bus
Aggregate growth in bandwidth	limited Bandwidth, shared
Unaffected link speed by number of nodes	as nodes increases speed goes down
Built in pipeline fashion	Pipelining is tough
Distributed arbitration	Central arbitration per layer
Separate abstraction layers	No abstraction layers
Performance guarantee is complex to assess	Fairly simple implementation

2.2 Network on Chip Building Blocks

A network-on-chip consists of three key building blocks. The first and most important one is the links that physically connect the nodes and actually implement the communication. The second block is the router, which implements the communication protocol (the descentralized logic behind the communication protocol). The last building block is known as the network adapter (NA) or network interface(NI). This block establish the logic connection between the IP cores and the network. Since each IP may have a distinct interface protocol with respect to the network [40]. While these building blocks set the usuall framework for the networks performance as well.

2.2.1 Link

A communication link is a combination of a set of wires which connects two routers in the network. In NoC Links can be composed of one or more logical or physical channels in which each channel contains a set of wires. Particulary a NoC link has two physical channels which creates a full-duplex connection between the routers (two unidirection channels in opposite directions). The number of wires per channel is same across the network which is known as the channel bitwidth.

2.2.2 Router

The basic function of this block has to implements the communication protocol. Communication protocol is the set of policies which is aquired in the design and implementation of router. Typically a NoC router is composed of a number of input ports, output ports(which connected to shared channels), a switching matrix that connect the input ports to the output ports, and a local port (L) to access the IP(intelectual property) core which is connected to router as shown in Fig 2.1. Apart to this the router also contains a logic block which implements the flow control policies like routing, arbiter, etc. and defines the overall strategy for traverse data across the NoC.

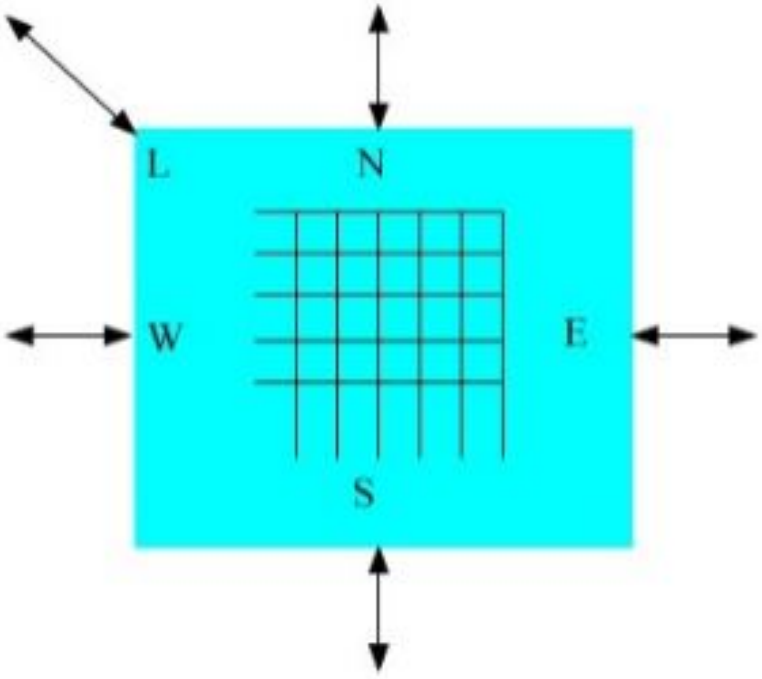

Figure 2.1: Interface View of a Typical Router [44]

2.2.3 Network Interface

The last NoC building block is the network adapter (NA) or network interface (NI). This block makes the logic connection between the IP cores and the network but each IP can have a different interface protocol with respect to the network. This block allows the separation between computation and communication. It also allows the reuse of both core and communication infrastructure of NoC independent of each other [41].

2.3 Topologies of Nocs

There are several factors which have direct influence on the performance of NoC are Topology, routing algorithm, traffic pattern, router architecture or flow control. The network topology determines physical layout and connections between nodes in the Network on chip (NoCs). Length of network have direct impact on the latency and throughput of NoC so also it determine the length in terms of number of hops for transmission of packet from source to destination [7]. In NoC complexity of topology

can determine by number of links at each router and ease of physical design on a chip [4]. There are various topologies to connect core with one another in interconnection network. Appropriate selection of topology for a network may help to improve the performance of network on chip communication [8]. Overview of various topology are given here.

2.3.1 Direct Topolgy

In direct networks each node has a direct connection to other nodes allowing for direct communications among processors. The typical examples of direct networks include hypercube, mesh, torus and star graph, as shown in Figure 2-2 below. These have been widely employed in recent practical multi-computer systems (e.g., Intel iPSC, Intel Cavallino, Intel Delta, and n CUBE [12]. The main advantage of such networks is that the resources of a node are available to each switch).

2.3.2 Indirect Topology

Unlike direct networks which have a direct connection, nodes under indirect networks are connected to other nodes through multiple intermediate stages of switches. Figure 2-2 below shows the common examples of indirect networks which include k-ary n-fly and fat-tree, which are adopted by many experimental and commercial parallel machines, such as DEC GIGA switch, Cray X/Y-MP, Myrinet, Cenju-3, NEC, IBM SP, IBM RP3, Meiko CS-2, Thinking Machine, and Hitachi SR2201 [12]. Fat-tree topology is a popular type of the interconnection network in cluster-based systems in particular [10, 14]. The fat-tree comes from the fact that, the number of links going down to its siblings is equal to the number of links going up to its parent in the upper level. Therefore, the links get fatter towards the root [15]. In fact this type has been the most popular network topology of all over the past fifty years. Commercial machines and many research prototypes have adopted different kinds of fat-tree.

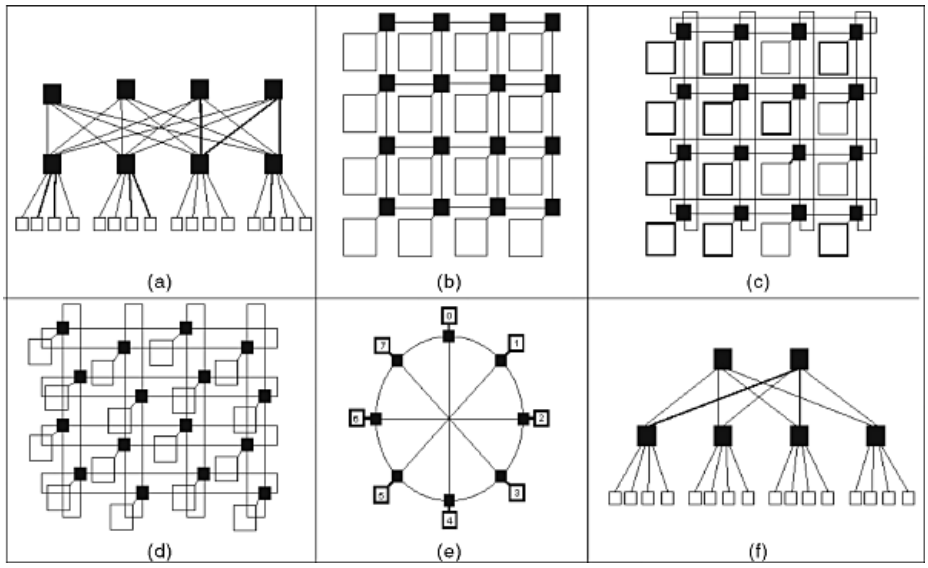

Figure 2.2: NoC Architectures: (a) Spin, (b) 2D Mesh, (c) Torus, (d) Folded Torus, (e) Spidergon /Octagon and (f) Butterfly Fat Tree.

2.3.3 Mesh Topology

Mesh topology is one of the most commonly adopted topology in Network on chip (Nocs). Usually Mesh topology can be describe as a K-ary-n cubes where k is the number of router and n is the number of dimension. In full mesh network at each intersection router exist and is connected to the neighbouring router and this topology can be easily implemented. Several authors have proposed that mesh topology provides better performance and throughput. For our simulation study we have adopted 8x8 mesh topology under synthetic traffic. The k-ary n-cube mesh network consists of N= K^n nodes arranged in n dimensions with k nodes per dimension. Where Each node consists of an intellectual property (IP) which is connected to a router(R) via injection and ejection channels. Figure 2.3 shows a mesh network and its node structure.

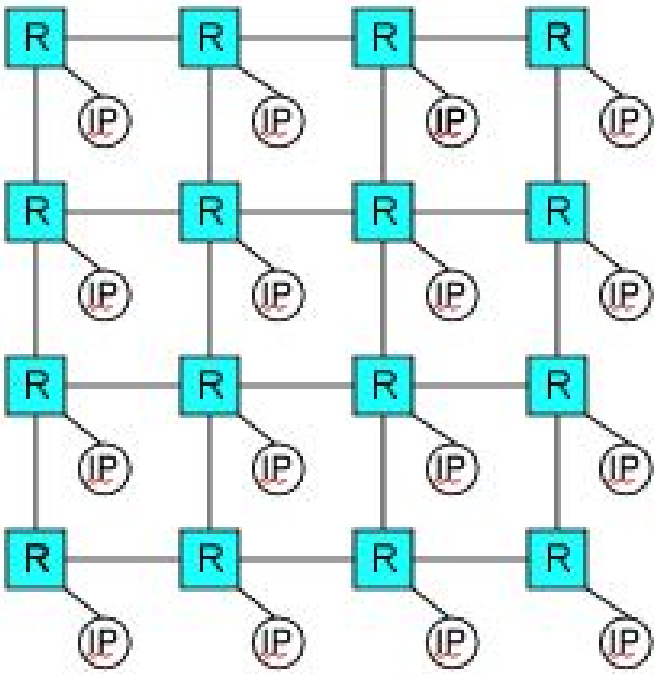

Figure 2.3: A 4-ary 2-cube network

2.4 Packet Format in Noc

Figure 2.4 show the basic units of resource allocation. A Packets is generally the basic unit of routing which decompose into three types of flits as follows Header Flit (HF), Payload Flit (PF) or Body flit, and a Tail Flit (TF). A flit (flow control digit) does not hold any routing information. It is the basic unit of storage allocation. A head flit contains the packet routing information from source to destination and tail flit indicate the end of packet at destination node. In NoC the typical value of flits is between 16 bits to 512 bits. A phit(physical transfer digit) is the basic unit that would be transferred through a link in a single clock cycle. The typical range of phit between 1 bit to 64 bits [37].

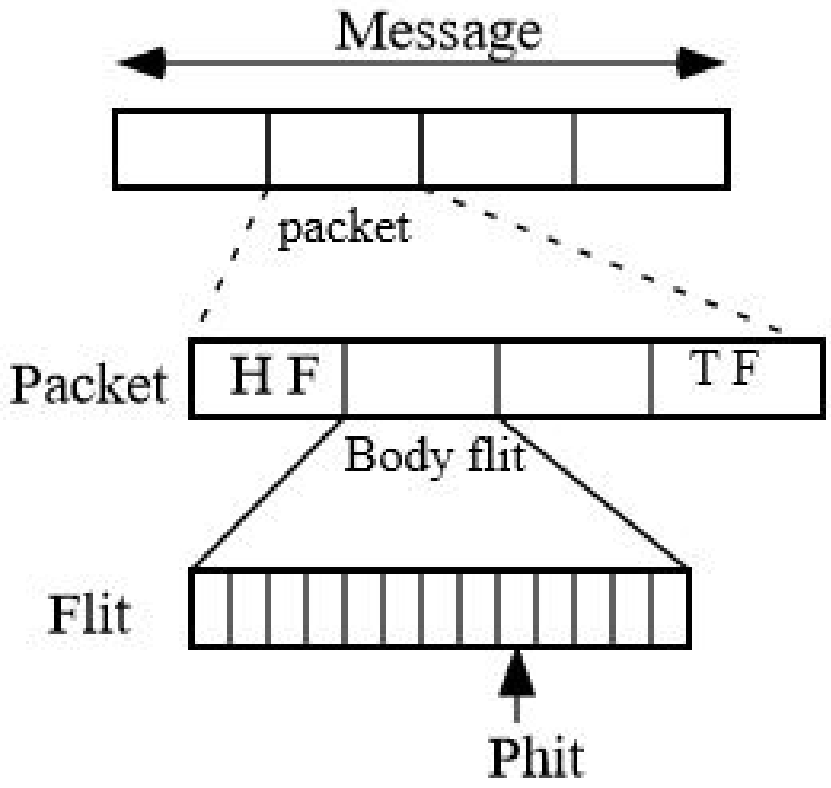

Figure 2.4: Unit of Resource Allocation

2.5 NoC Routing

Routing is the mechanism that defines the route in computer networking along which to send packet or physical traffic [13]. A routing algorithm has a vital role in effective communication of Interconnection network description of the method that determines the route. Routing algorithm have been classified in several ways as follows.

- Source vs. Distributed routing
- Deterministic vs. Adaptive (Static vs. Dynamic) routing
- Minimal vs. Non-minimal routing

2.5.1 Source vs.Distributed Routing

This classification is based on where the routing decisions are made [14]. In source routing the information about the entire path for a packet from source to the destination is pre-defined and provide it in packet header. In source routing, all routing information is defined inside the source core before injecting any packet in network because each source consist on a list or table that provide complete route information to reach all other resources in the network [14]. In source routing the route information that is no longer necessary may be stripped off to save bandwidth [15]. However, the source routing does not scale well because the header size depends on the distance between source and destination [14]. The teraflop chip uses source routing for communication purpose.

In distributed routing, each router computes the next route that will be while used while the packet travels across the network [16]. In distributed routing the packet header contain only the destination node ID. Based on packet destination address, each router decides whether it should be delivered to the local resource or forwarded to one of the neighbouring routers [15]. The packet header size is smaller in DR as compared to SR and it less dependent on network size [16].

2.5.2 Deterministic vs Adaptive Routing

Deterministic routing algorithms does not consider the conditions of the network, like traffic amounts or congestions. A router route the packet and make routing decisions on the based of some algorithm like XY algorithm and only a single fixed path is provide between a source node and a destination node. While adoptive routing provide several paths between a source and a destination target. However the final selection of path is defined at run-time often depending on conditions of the network, like traffic status or congestions. Adaptive routing algorithms are also able to utilize redundant routing resources and adapt their routing decisions to the prevailing traffic conditions and balance the traffic load over the whole network in such a way that the possibility of congestions decreases, which also improves the performance of the network [19]. An adaptive routing algorithm also improves the probability of passing a packet from a congested or malfunction link. While deterministic routing algorithms are the appropriate choices for synthetic traffic patterns, the adaptive algorithms are preferable in the presence of irregular traffic like non-uniform or bursty traffic or in networks with

unreliable routers and links [18]. Deterministic routing algorithms provide typically worse performance than adaptive routing algorithm but the performance of the non-adaptive routing can also be improved by a smart allocation of network resources. A new approach which aims at allocating link capacities to virtual circuits (or routing paths) in such a way that the network throughput would be maximized, is called oblivious routing.

2.5.3 Minimal vs Non Minimal Routing

Route lengths determine if a routing algorithm is minimal or non-minimal. A minimal algorithm only permits paths that are the shortest possible, also known as profitable routes. A non-minimal algorithm can temporarily allow paths that in this sense are non-profitable. Even though non-minimal routes result in a longer distance, the time for a packet transmission can be reduced if the longer route allows for avoiding congested areas. Non minimal routes may also be required for fault-tolerance.

2.6 Problems on Routing

Routing algorithm is responsible for the selection of the effective path in order to transfer the packet from source to destination. Selection of the routing algorithm is mostly important factor to achieve good network performance.So therefore achieving minimum communication average packet latency and high throughput network must be considered free deadlock, livelock and starvation routing algorithm. These are the main issues which degrade the performance of NoC as follow [43].

2.6.1 Deadlock

Deadlock may be define as a situation when one or more packets never reaches their destination node and become blocked and waiting for an indefinite time frame to release the resource (buffers,channels). It can affect both the usability and performance of routing algorithm which leads to degrade the latency and throughput of a packet in NoC and this phenamenon is shown in given figure 2.5 where mutiple packets want to access a single resource like input buffers as shown in Figure 2.5. Actually deadlock is a cyclic chain of packets, that wait for each other to release their buffers.

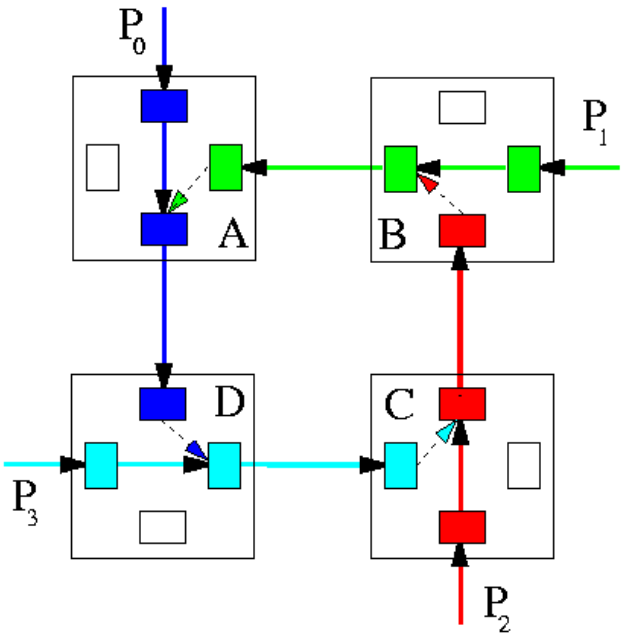

Figure 2.5: Deadlock

2.6.2 Livelock

Livelock occurs in NoC when a packet keeps spinning around its destination node without ever reaching it is known as Livelock. This situation occur in non-minimal routing scheme. Livelock should be cut out to guarantee packets throughput. There are a different types of approaches to avoid the livelock issue. Time to live (TTL) counter counts how long a packet has travelled in the network. When the counter reaches some predetermined value, the packet will be removed from the network. The another resort could be given packets a priority which is based on packets age. The oldest packet always finally get the highest priority and will be routed forward [12].

2.6.3 Starvation

Starvation is a situation where some packets with lower priorities could not advance toward their destination node when the packets with higher priorities reserve the resources all the time. Starvation can be resolved by using a fair routing algorithm or reserving some bandwidth only for low-priority packets.

2.7 Switching Techniques

For the efficient design of communication network the selection of appropriate switching technique is one of the most challenging task apart its switching scheme paly a key role to optimize the performance of NoC [10]. Switching is the actual technique that forwards data from an input channel to an output channel of next hop [11]. Which output channel is to be selected depends on routing algorithm. It defines that how network resources are allocated for packet transmission like channel allocation and the routing algorithm decides that how and when the input channel is connected to the output channel. In switching scheme message having header and data are transmitted to the destination through the nodes in NoC [12]. Particularly there are two main switching techniques as follow circuit switching and packet switching. Packet switching has further three important switching techniques which is used by network as store-and forward, virtual cut-through and wormhole switching [26].

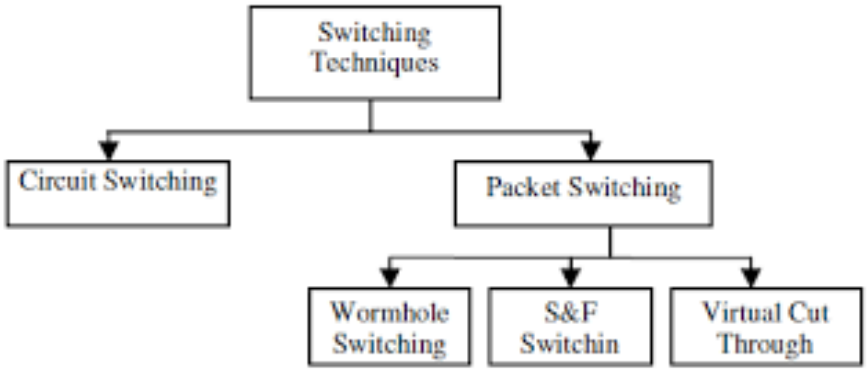

Figure 2.6: Switching Herarchy in NoC

2.7.1 Store and Forward Switching

Store and forward (SAF) have been used in the CLICHE NoC [8]. Based on implementation it is the simplest switching technique. It forwards packets step by step on a per hop basis. When a router receives an entire packet which is stored in the routers input buers and only after the enitre packet has arrived it is forward to the next switching port. As compared to other switching scheme this switching scheme requires relatively large buers that are able to buffer at least a complete packet. This results in high area costs the reception of the complete packet before forwarding it

also increases the network latency as well [7]. The requirement of large buering area can be avoid by using the wormhole switching method [25].

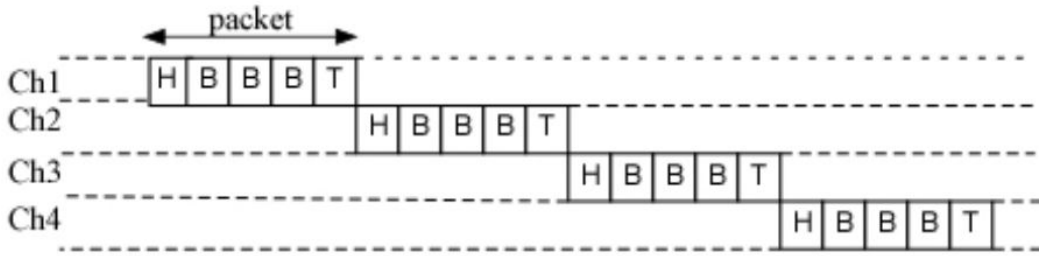

Figure 2.7: Store and Forward Switching

2.7.2 Wormhole Switching

In wormhole switching each packet is split into small pieces called Flits (flow control Digit). Usually the header flit of a packet holds routing and control information of a packet. Actually the body flit of a packet follows the header flit in a piplined fashion. As soon as the node examines the header flit of a packet, it select the next channel on the route and begin flits transmission that channel [4]. As the header flit moved forward the rest flits of a packet folowed it in a pipeline way. Because the body flit have no routing information and flits of a packet transmitted in contiguous channel of the network that might not be interleaved with the flits of other packets [4]. In wormhole switching if the header flit of a packet is blocked, rest of the flits of a packet stop advancing and block the transmission of any other packets requiring the network resources they occupy.

In wormhole switching an entire packet need not be buffered at the input queue of a router to move to the next node. Which decrease the network latency as compared to store-and -forward routing. Using of FIFO flit buffer it can avoids memory bandwidth in the router [4].

2.7.3 Virtual Cut Through

Store and forward only forwards a packet to the next router after its complete reception. In contrast, virtual cut through [39] forwards a packet immediately after the necessary information for the routing decision, i.e. the packet header has been received. In general the header will arrive in a router before the payload. As soon as

the routing decision has been made and the output link is free, the already received parts of the packet are forwarded. If the desired channel is not available the packet is received and fully buered in the current router. In full virtual cut through, the packet must then be buered entirely before it can be forwarded even if the output link becomes free earlier. Thus a stalling packet does not block any links, however buer space requirements are as high as in SAF. Since a packet is only forwarded if the downstream can accept a complete packet. If there is no blocking, the packet is pipelined through the router along the path determined by the routing function. In a network with high load, virtual cut through will behave like a store and forward.

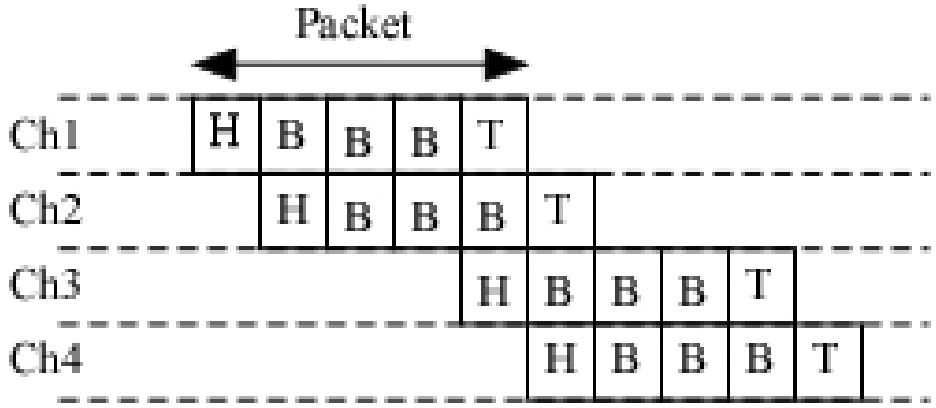

Figure 2.8: Virtual Cut Through Switching

2.8 Wormhole Router

The design of wormhole router consists of the following main components as follows.

- Input and Output FIFO buffer
- Routing and Arbitration unit
- Crossbar Switch

In a wormhole router each input port contain on a module namely input controller which is further divided into three sub modules known as link controller, FIFO and

output controller [17]. The flow of packets across the physical channel between two routers is maintained by the input controller. The function of link controller is to receving the flits of incoming packet from downstream router and stores them in FIFO buffer unit of upstream router. The routing function implements the wormhole routing algorithm, in which a message broken into flits for transmission and flow control. The header flit (containing routing information) governs the route and the remaining data flits follow in a pipelined fashion. If the header blocked, remaining data flits would be blocked in that situation. The computing of the appropriate output port is realized on destination address that stored in the header of a packet. As the header flit arrives at the input of buffer, routing computation executes the wormhole routing algorithm and according to wormhole routing algorithm a packet is split into a fix sequence of multiple flits (flow control digit). The arbitration unit which contains of mutiple arbiters receives requests from the routing function of each input port and assigns available output ports to the requestors. In general the header flit of a packet contains on all the routing information which determine the route for the header flit of a packet and the remaining flits of a packet follow them in a pipelined fashion. There are differnt types of arbiter algorithm have been proposed in which a round robin algorithm is commonly employed in wormhole router. The role of crossbar switch in a router is to connected the five inputs port to five outputs port of router without any intermediate stages and the credit signals from the upstream router is sent to downstream router through the credit switcher module[16]. At very high work load traffic, wormhole router may be create deadlocks issue and highly unpredictable latencies which degrade the efficiency of router [18].

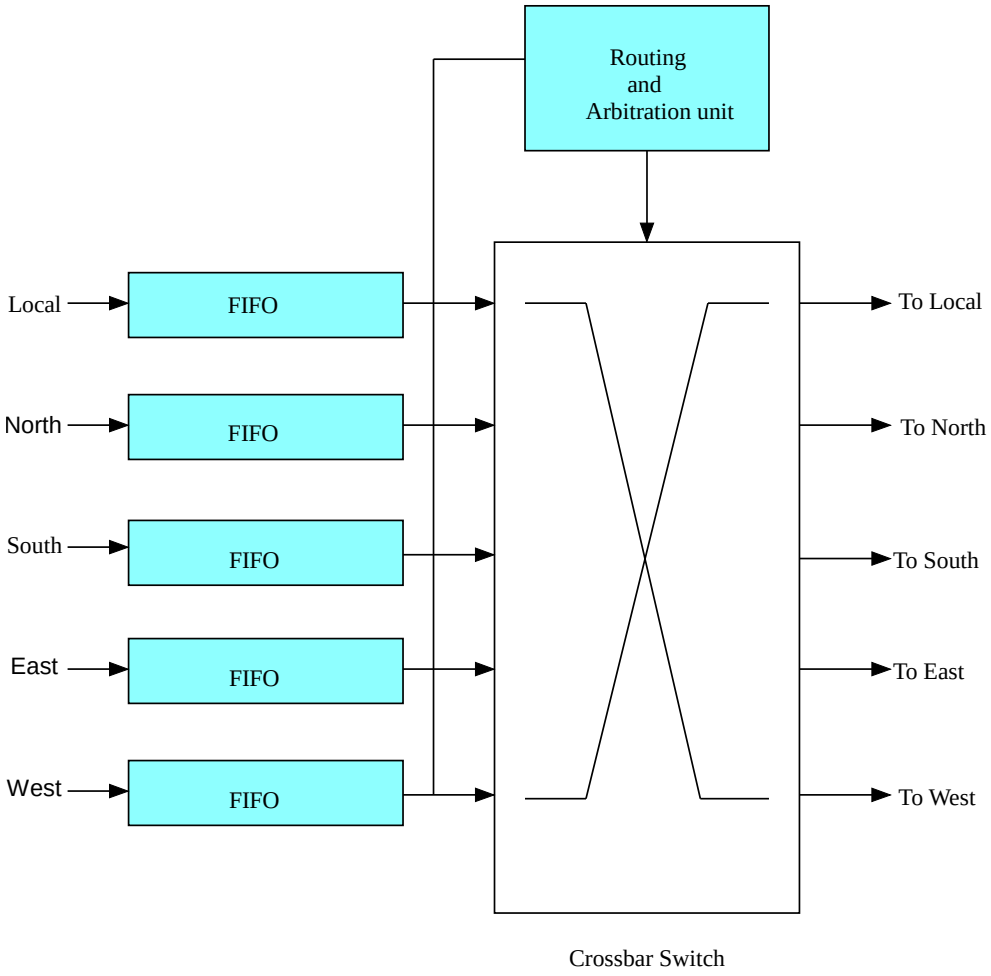

Figure 2.9: Wormhole Router

Chapter 3

Router Model

Designing of efficient channels is largely a circuit design problem [1, 2, 3] and performance of a router, cost and efficiency mostly depend on its microarchitecture. Usually a router has a set input and output ports, namely North, South, East, West and Local which communicate with the IP core through network interface and neighbouring routers. The incoming packets are recevies by these ports and forward them to the appropriate port of the downstream router. each port contain on a small size of a buffer which store the packet temporarily [5]. In this thesis we are using Input-queued Router model under virtual channel flow control in order to investigate the effect of virtual channel on the performance of NoC using mesh topology.

3.1 Input-queued Virtual Channel Router

The router model whcih we used in the Booksim simulator is input queued credit based virtual channel router. In this section we explained the block diagram of a basic input- queued credit based virtual channel router. In which each virtual channel consist on multiple buffers as shown in Fig 3.1. It also conatain an extra component called VC allocator, SW allocator, RC (Route computation) and crossbar switch. VC allocator assigns output virtual channel to an incoming packet if one is available. Credit-based virtual channel router typical operation can be described as follows below. The input unit of virtual channel router contain on input buffers and registers which store the control state of the network because control state defined the state of a virtual channel as idle, active or waiting. When the flit of the incoming packet arrives at the input unit of virtual channel router it stored in the buffer stage until

free output virtual channel is available. At arriving of header flit first route computation unit determined the output port to which can be transmitted after that the switch allocator selects a route for the transmission of flit from input buffer unit to output buffer unit. Similarly, like input buffer unit the output unit also contain on buffers and control stage which forward the incoming packet flits from upstream router to a downstream router using the physical channel. In the virtual channel router packet transmission from upstream to downstream router, route computation and VC allocation performed once per packet whereas switch allocation is performed on flit by flit of the incoming packet. In Fig 3.1 the credit-based virtual channel router block diagram is shown. In this type of router model credit based flow control strategy have been adopted in which a counter is used at input unit and output unit of a router as well which maintained to keep track of the free available number of buffers.on reception of a flit, the counter is decremented and incremented on a flit transmitting. In credit-based virtual channel router, the upstream router takes the credit information from the downstream router for the packet flow. In this such type of router, the packet could be transmitted in one case if credit is available at the neighboring router [15].

3.1.1 Route Computation

This unit selects an appropriate output port and a set of candidate output VCs for each incomming packet at node according to the implemented routing scheme in the network. This unit only needs to be performed for the first it of every packet which is also termed head it of each packet any subsequent its from the same packet simply inherit the selected output port.

3.1.2 VC Allocation

In Virtual Channel (VC) flow control when the head flit of a packet is arrived at a router, it must be required one of the VCs which is associated with the physical channel that connects to its destination output port. In order to achieve this, the head flit forward a request to the VC allocator once it reaches at the front of its input VC. The VC allocator creates a matching phenomena among any such requests from the input VCs on the one hand and those output VCs which are not currently in use by another packet on the other hand [39]. The VC allocator allocates available output

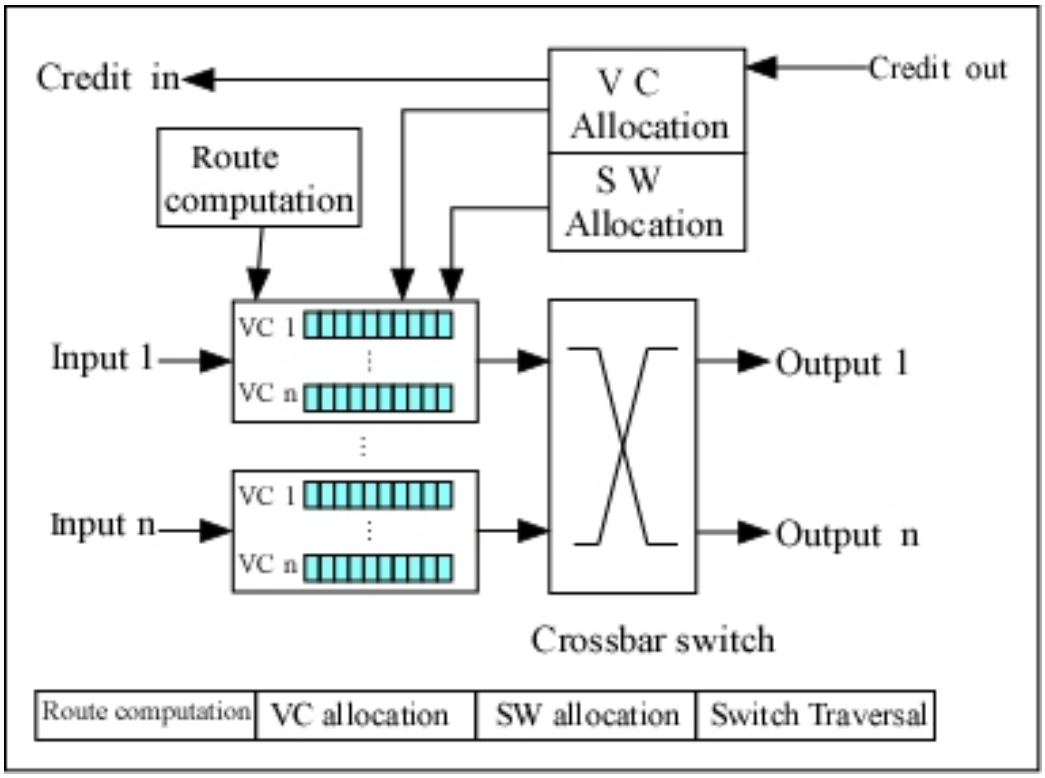

Figure 3.1: Input-queued virtual channel router model

VCs to incoming packets at the routers input ports. Similar to route computation unit, VC allocation unit only needs to be performed for head its as the assigned output VC is inherited by subsequent body and tail its from the same packet. There are different types of algorithm used for virtual channel allocation unit like round-robin algorithm, iSLIP algorithm [32]. In our router model we have considerd iSLIP algoritm.

3.1.3 Switch Allocation

When the head flit of a particular packet reaches the router, it allocates one of the virtual channels associated with the physical channel that connects to its destination output before it can proceed. First, the head flit sends a request to the VC allocator once it reaches the front of its input VC after that, the flits will be forwarded to the destination output port. For each flit to be transferred, a crossbar connection between the corresponding input and output ports must be established for one cycle. The switch allocator is responsible for scheduling such crossbar connections between input virtual channels and output virtual channels. The main function of switch

allocator is to establishing a crossbar schedule with the help of allocating time slots to incoming packets at the router's input buffer [31]. There are two main aims of the design of switch allocator are to ensure a fair distribution of the shared resources across the different packet or flits requesters and second to maximise the number of satisfied. Allocators are usually contains on a groups of arbiters. A single arbiter may decide for a single resource by chosing a single winner from a group of requests for that resource. Much research showed that design of good quality, low-cost on-chip arbiters with round-robin or matrix arbiters being commonly used [32]. We adopted first iteration separable allocator (ISLIP) as a switch allocation algorithm because of good cost and performance balance for allocating across larger numbers of inputs and outputs as shown in Fig 3.2.

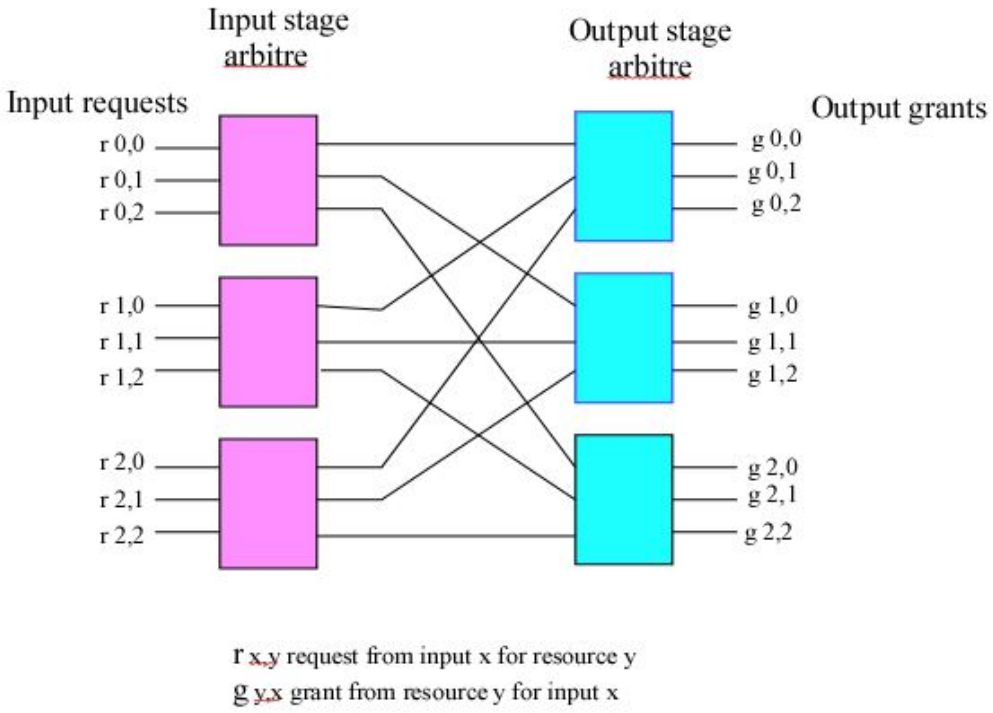

Figure 3.2: Principle of operation of a separable allocator

3.1.4 Switch Traversal

In this stage those flits which won the contention in the switch allocation stage, will go to the next stage by traversing the crossbar switch and arriving at the concerned

output ports.

3.1.5 Link Traversal

After the switch traversal stage, flits traverse the link and go to the neighbouring router according to their destination node.

3.2 Traffic

To simulate the performance of our input-queued virtual channel router model we used a comprehensive cycle accurate interconnection network simulator known as Booksim2 which provide several synthetic traffic pattern for network evalution scenarios. Brief descriptions of each traffic pattern are mentioned in below given table. In network simulator Several synthetic traffic patterns are used to cover a wide range of network utilization scenarios. Descriptions for each traffic pattern are listed in Table 3.1.

Table 3.1: Traffic pattern Description

Traffic pattern	Description
Uniform	Each source sends an equal amount of load to each each destination
Bit complement	Bits at destination point ID are the complement of bits in source point ID
Right Neighbour	source sends to its right neighbour for right edge nodes they
Bitrev	Bit reverse. Bits are reversed at destination point ID Bits compared to source point ID
Transpose Destination	Are the transpose of the sources in a regular mesh topology
Asymmetric	Sources in upper half send to lower half nodes in the same column at same distance from the middle vice versa

3.3 Basic Concept of Arbitration

When more than one input port want to access a common physical channel resource. In that case, an arbiter is required to determine how the physical channel can be shared amongst many requestors. Figure 3.3 shows an example of arbitration in FIFO [35].

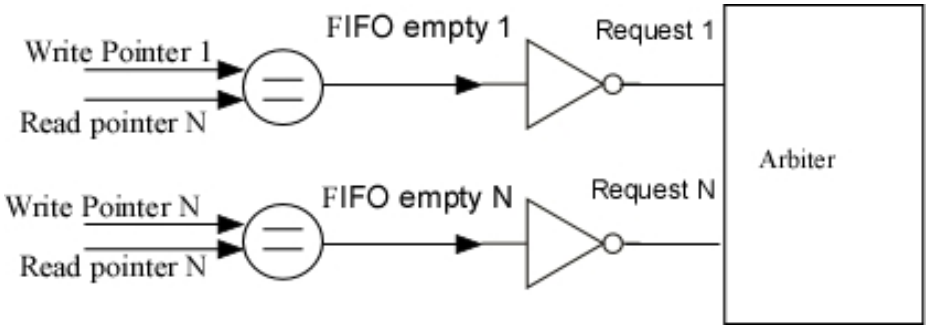

Figure 3.3: FIFO Arbiter

3.3.1 Fixed Priority Arbiter

One general arbitration scheme is a fixed priority arbiter. Each input port has its own fixed priority level, and an arbiter grants an active request signal with the highest priority depending on this priority level. For instance, if request 1 has the highest priority among N requests, and request 1 is active. So it will be granted regardless other request signals. If request 1 is not active, the request signal with the next highest priority will be granted. In other words, the current request (lower priority) only will be served if the previous request (higher priority) has not appeared or been served already. Therefore,fixed priority arbiter could be used where there are a few requesters. The implementation of fixed priority arbiter can be done by the following case statement [35].

3.3.2 Round-Robin Arbiter

In NoC when more than one input request to a share resource at the same time then there could be a contention phenamena exist which degrade the performance of NoC

router. An arbiter might resolve this conflict by allocating priorities to resources based on different scheduling algorithm like matrix arbiter, tree arbiter, fixed priority arbiter or round robin arbiter. It also update the status of all input ports and know the status of port that which ports are free and which ports have established communication. Round Robin Arbiter work on the principle that a request which was just served would have a lowest priority on the next round of arbitration [23]. In a round-robin arbiter every requester might get a turn in an order because a request that was just served could have the lowest priority on the next round of arbitration [24]. In round robin type arbiter uses a pointer register which update the status of all inport requests which request is the next one. A round-robin arbiter consider the last winning request with lowest priority for the next round of arbitration. When there are no requests form other inputs then the priority would not be changed [24]. As comapred to other arbiter types a round-robin arbiter is a strong fairness arbiter.

3.3.3 Tree Arbiter

A tree arbiter conatain on a herarical organization of small number of arbiter as shown in Figure 3.4. A (pk) input tree arbiter uses p independent k-input arbiters of arbitrary scheme in order to determine a winning agent for each individual arbiter in parallel, a single p-input arbiter pick out a winner among all arbiter that have at least one request and enables only that arbiters outputs [21]. In a tree arbiter scheme each arbiter transmit eagerly request sends to the upper tree before it detefines which input will grant [20]. The example of tree arbiter as shown in given below Fig 3.4.

3.3.4 Matrix Arbiter

Matrix arbiter uses as another alternative implementation scheme for delivering strong fairnes between all input requests. In this type of arbiter explicit tracking pairwise precedence are used which store the updating value in a matrix form register that is why it is known as matrix arbiter. This type of arbiter function depend on least recently served priority policy which maintaining a triangle array of state bit for all row and column (row i and column k) indicates that request i takes priority over request k. Only the upper triangular portion of the matrix need be maintained. Figure 3.5 shown that how a single ouput grant is generated by a 4-input arbiter in matrix arbitration. This arbiter make sure that a grant is generated only if an input request with a higher priority is not established. The matrix register is updated after

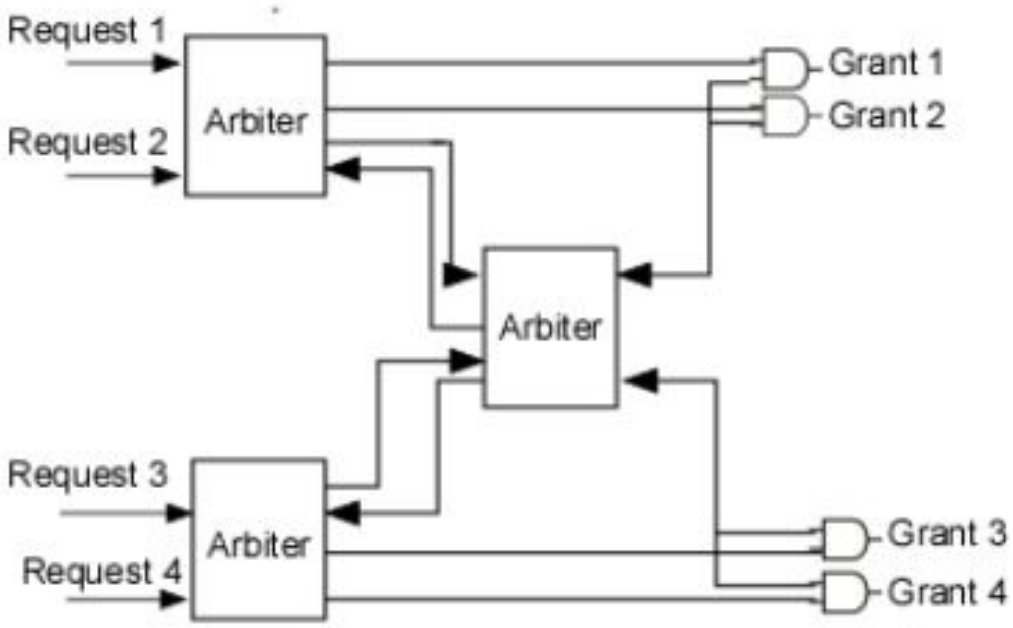

Figure 3.4: Tree Arbiter

each clock cycle to reflect the new request priorities [22]. This type of arbiter is an apprprioate arbiter with a relatively small number of inputs because it is fast inexpensive to implement and generates strong fairness.

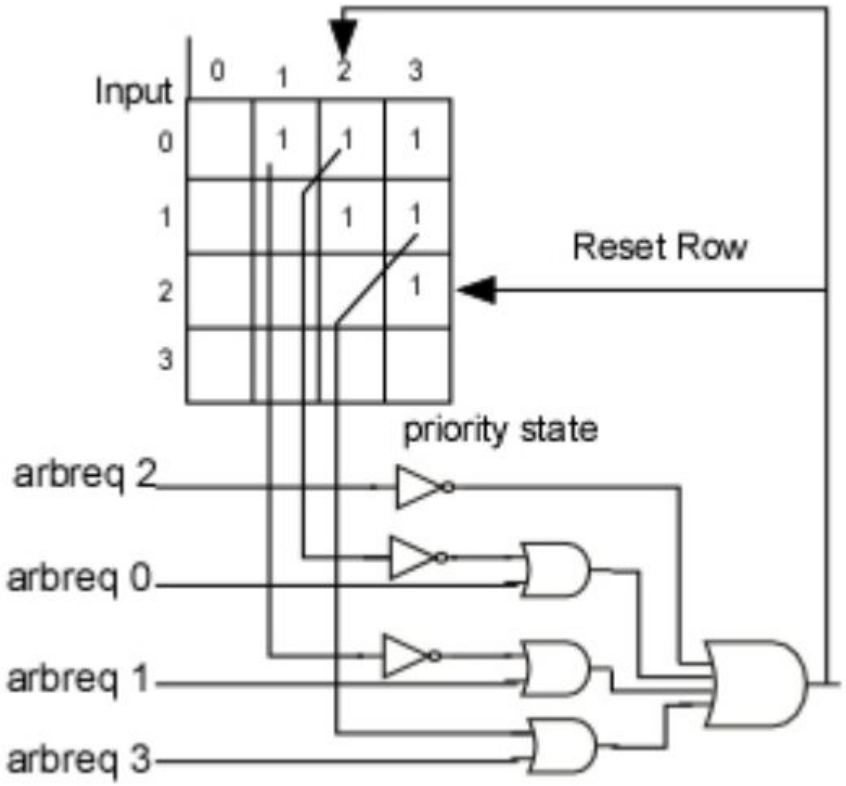

Figure 3.5: Matrix Arbiter

Chapter 4

Simulation and Results

Computer simulation tools are important for the study of a communication networks performance. An appropriate simulation model can help in the process of developing and improving the network performance as well. There are several various types of NoC simulation tools and methodologies which support to research on NoC like Noxim [27] and OMNET++ based HNOCS simulators [28]. Which have been develpoed to explore different aspects of NoC design like.

- Configuration of NoC nodes
- Configuration of the NoC like topology, routing algorithms,virtual channels e.t.c
- Data communication requirements like traffic pattern,channel bandwidth and other.
- Benchmarking and analysis of simulation statistics.

Noxim simulator have developed using SystemC. It has command line interface to configuring differnt components of NoC. Noxim tool only support mesh topology with wormhole router over synthetic traffic patterns. It does not support area, power and hot spot analysis simulation [30]

4.1 Booksim2 Simulation Platform

In this section we would study about open source Booksim2 NoC simulator. It is a comprehensive cycle accurate interconnection network simulator which developed in C++ [29]. Booksim2 support different types of network topologies like mesh,

torus, cmesh, fat, tree and others where it can be used to measure key metrics of NoC design including average, minimum and maximum packet and flit latency and maximum and individula link utilization as well [29]. It support differnt sorts of routing algorithm to direct packets for the supported topologies. For NoC evaluation Booksim2 simulator support two different types of router model like virtual channel router and event driven router model. Booksim2 supports a set of different pre-made traffic patterns like synthetic pattern and customize traffic pattern. Booksim2 is a fast, comprehensive and easy-to-use NoC simulator. We have used it as a base for the simulation tool for evaluation of virtual channel router model.

4.1.1 Simulation Model

The basic setup which we used for the simulation environment is mentioned in given Table 4.1 where we are using the input-queued virtual channel router model which has 5 input and output ports, each input port contain on 8 VC, and used iSLIP algorithm for switch and virtuall channel allocator. The performance of Noc in term of throughput and latency values is extracted from the synthesized result of network model. we considered the warm-up period is 3 cycles and throughout each simulation was carried out for a sample period of 1000 cycles. In order to investigate the effect of virtual channel flow control we varied the Number of Virtual Channels across all our simulations result as VC=1,VC=2,VC=4 and VC=8.

Table 4.1: Key Parameters used in the simulation model

Parameters	Value
Network Size	8X8 nodes
Topology	Mesh
Packet length	20 flits / VC
buffer size	8
Credit delay	2 cycles
Routing delay	0 cycle
VC allocation delay	1 cycle
Switch allocation delay	1 cycle
Switch traversal delay	1 cycle
VC allocation algorithm	iSLIP[32]
Switch allocation algorithm	iSLIP
Arbitration type	Round Robin

4.1.2 Performance Metrics of NoC

There are three basic metrics on which the performance of a network-on-chip can be evaluated are bandwidth,throughput, and latency [40]. Throughput and avaerage latency of a packet that we are considering in our simulation model. Throughput can be dene as the trac accepted by the network where average number of packets delivered per cycle is known as accepted trac.

•**Bandwidth** the term bandwidth denes the channel capacity or the maximum throughput of a logical or physical communication path in a digital communication system. Its unit is measured in bit per second (bps).

•**Throughput** It can be define as the maximum traffic accepted by the network, where traffic accepted is the amount of information delivered per unit time

•**Latency** latency is time elapsed from the instant that the first flit of the packet is injected into the network at the source node until the last flit of the packet is received at the destination node.

4.1.3 Dependence of Latency on Injection Rate and Number of Virtual Channels

Basically there are two main metrics of NOC, throughput and avaerage latency of a packet that we are considering in our simulation model. The basic setup which we used for the simulator is given in Table 4.1 where we fixed the warm-up period for 3 cycles and the simulation was carried out with a sample period of 1000 cycles. We varied the Number of Virtual Channels across all the simulations as respectively 1, 2, 4 and 8. For VC=1 the injection rate or load at which the value of packet latency is reached to 2762.36 is 0.02. For VC=2 the injection rate at which the value of average packet latency is reached to 1391.87 (cycles) is 0.02 which means the load that could be handled by the network with VC=2 increases by 47 percent with respect to VC=1. Similarly for VC=8 at injection rate of 0.02 the average packet latecny is improved by 92 percent respectively from the maximum value of latency for conventional network VC=1.

Table 4.2: Simulation Result For Dependence of Latency on Injection Rate and Number of Virtual Channels

injection rate	virtual channel vc=1	vc=2	vc=4	vc=8
0.005	63	53	55	51
0.01	623.34	233.65	123.32	58.12
0.015	1911.32	445.21	234.12	94.53
0.02	2762.36	1391.87	333.67	220.43

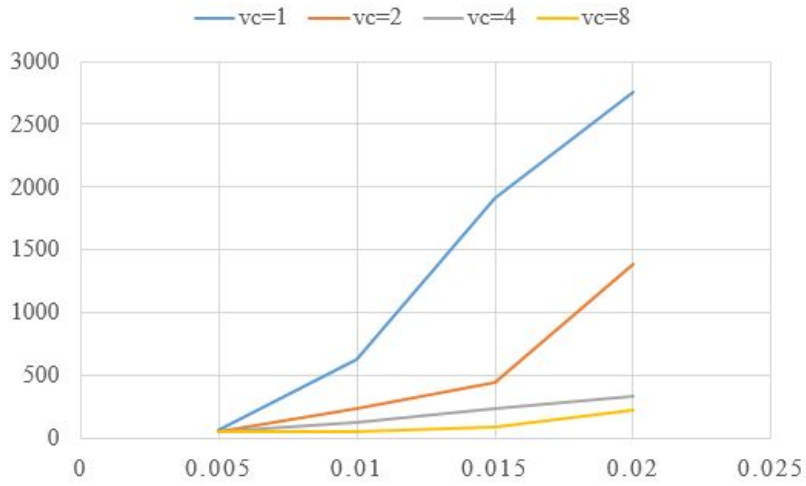

Figure 4.1: Average packet latecy vs injection rate

4.1.4 Dependence of Latency on Packet Size and Number of Virtual Channels

Packet length has an important role in the performance of on-chip network. In this thesis we simulated the effect of various packet length on the performance of on-chip network under synthetic traffic pattern. We varied the regular networks packet size from 20 upto 100 (flits) and keep the traffic pattern is fixed as uniform. This enables us to evaluate the effect of a virtual channel with variation of packet size on the performance of NoC. Latency graph is produced as a function of packet length as shown in Fig 4.2 and we plotted average packet latency as number of cycles. For VC=1 the packet size at which latency reaches a value of 1575 is 20 flits. As we increased the size of packet upto 100 flits the packet latency reaches to high value of latency which leads to degrade the performance of Noc. After adding four virtual channel the value of latency is reached to 3631 at the packet size of 100 flits which improved by 27 percent with respect to nominal latency for VC=1. we can also notices from graph that as the packet size varied to respectively virtual channel the latency reached to a saturation point and remain constant after that point. behind the saturation point there is no further improvement in latency by using adding more virtual channel.

Table 4.3: Simulation Results For Dependence of Latency on Packet Size and Number of Virtual Channels

Packet size(flits)	virtual channel vc=1	vc=2	vc=4	vc=8
20	1575	188	93	91
40	2916	2260	1694	1575
60	4306	3351	2882	2770
80	4783	3493	3398	3712
100	5023	3821	4599	3631

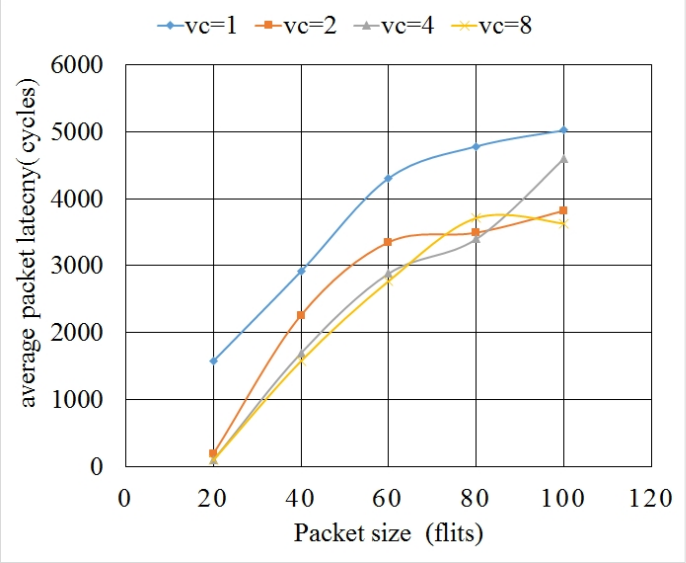

Figure 4.2: Packet Latency vs Packet Size

4.1.5 Throughput vs Virtual Channels

Throughput is the measurement of traffic accepted by network where accepted traffic can be define as the number of packet delivered per cycle [38]. First we simulate the effect of single virtual channel on the throughput of noc as shown in Fig 4.3 under uniform traffic pattern the throughput for VC=1 is 0.0047 (packets/cycle). Fig 4.2 and 4.3 shows when we increases the number of virtual channels the saturation point goes towards higher injection rates. For VC=1 the injection rate or load at which throughput reaches a value 0.0078 is 0.02. For VC=2 the injection load at

which throughput reaches 0.0149 is 0.02 which increased by 57 percent with respect to VC=1.similary at injection rate of 0.02 the throughput improved significantly by 155 percent for VC=8 which means the load that can be handled by the network with VC=8 increases by 64 percent with respect to VC=1. We can observes that after saturation point there is no furhter improvement existed in network throughput when nearly all buffer become saturated in the network.

Table 4.4: Simulation Result For Injection Load vs Throughput

injection rate	virtual channel vc=1	vc=2	vc=4	vc=8
0.005	0.0047	0.0051	0.0051	0.0053
0.01	0.0088	0.0097	0.097	0.0113
0.015	0.0088	0.0144	0.0147	0.0151
0.02	0.0078	0.0149	0.0194	0.0199

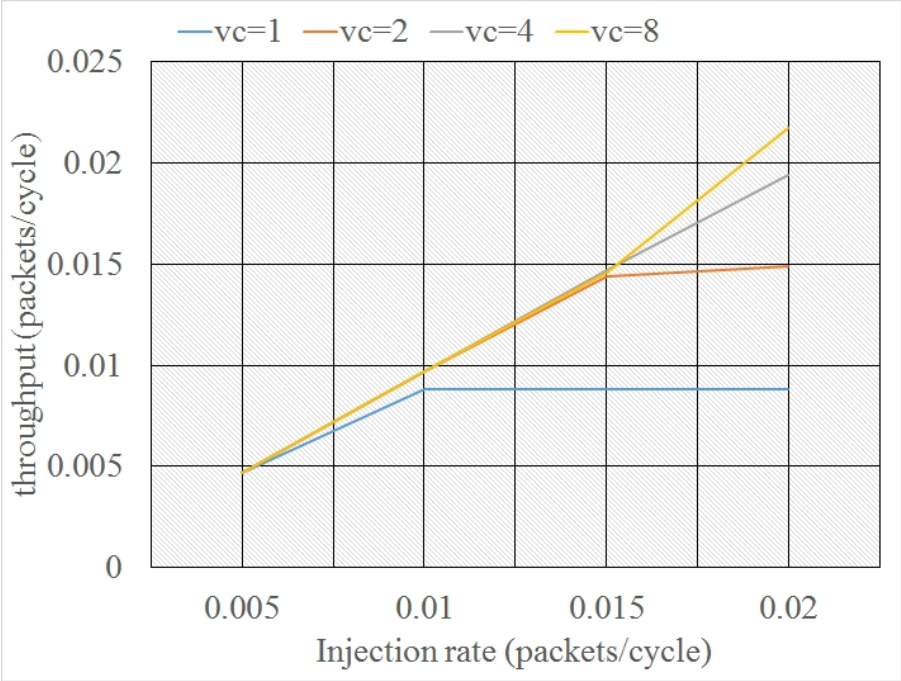

Figure 4.3: Throughput vs Injection Rate

Chapter 5

Conclusions

We have observed from this simulation model that the increase of packet injection rate, the network tends to saturation. Therefore latency of packets in reaching to destination will severely increase with respect to FIFO router model. Adding virtual channels in each direction in routers can increase network throughput. Performance of k-ary-n-cube network using virtual channel could be increase upto a certain number of virtual channels beyond saturation state of network. The simulation result show that there is not much improvement in throughput and latency after saturation point. Using virtual channel flow control the saturation throughput have been increased and reduced the blocking probability of a packet in 2D mesh network with respect to FIFO router model. The simulation graph show that adding a smmal number of virtual channel can increase throughput significantly. Apart this our simulation result suggest that four to eight number of virtual channel is appropriate for most networks.

5.1 Future Work

In the future when the number of cores in a 3D network topology increases simulators will not be effective because of the time which they take for each simulation. So we can evaluate the performance of Noc using alternate platform like FPGA prototyping or analytical model as well. Using VLSI circuits the cost of adding small amount of control state like VC allocator which is required to implement multiple lane per channel could be well worth the cost.

Bibliography

[1] James Chen, Self-Calibrating "On-Chip Interconnects," Ph.D dissertation, Dept. Electrical , Stanford University, Mar. 2012.

[2] Ron Ho, Ken Mai, and Mark Horowitz, "Efficient On-Chip Global Interconnects," In Digest of Technical Papers of the 2003 Symposium on VLSI Circuits, pp.271-274, Jun. 2003.

[3] Ron Ho, Tarik Ono, Robert David Hopkins, Alex Chow, Justin Schauer,Frankie Y Liu, and Robert Drost. "High-Speed and Low-Energy Capacitively-Driven On-Chip Wires," IEEE Journal of Solid-State Circuits, Vol. 43, no.1 pp 52–60, Jan 2008.

[4] Ayse Yasemin Seydim. "Wormhole Routing in Parallel Computers," School of Engineering and Applied Sciences Southern Methodist University May. 1998.

[5] Ruchika Chandravansh, Vivek Tiwari "Network on Chip Router Architecture Performance Analysis by using VHDL," International Journal of Computer Applications (0975 – 8887) Vol. 140, No.13, April. 2016.

[6] C. L. Liu, James W. Layland, "Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment," vol. 20, pp.46–61, Jan.1973.

[7] Andreas Johann Lankes, "A Selective Packet Discard Technique for Efficient Deadlock Recovery in Networks-on-Chip," Ph.D. Dissertation, Dept. Electrical Engineering, September. 2014.

[8] S. Kumar, A. Jantsch, J.-P. Soininen, M Forsell, M. Millberg, J. Oberg, K. Tien syrja, and A Hemani, "A network on chip architecture and desig methodology," In VLSI, 2002. Proceedings. IEEE Computer Society Annual Symposium on, pp 105–112, April. 2002.

[9] Yogita A Sadawarte, Mahendra A.Gaikwad and Rajendra M.Patrikar. "Implementation of Virtual Cut-Through Algorithm for Network on Chip Architecture, " IJCA Proceedings on International Symposium on Devices MEMS, Intelligent Systems and Communication , 2011.

[10] Y.A.Sadawarte, M.A.Gaikwad and Rajendra M.Patrikar, "Review of Switching Techniques for Network-on-Chip Architectures, International Journal on Computer Engineering and Information Technology, Vol. 17, no.22, pp. 52–57, 2011.

[11] Y.A.Sadawarte, M.A.Gaikwad and Rajendra M.Patrikar, "Comparative study of switching techniques for Network on chip Architectures, International Conference on Communication, Computing and Security. NY, 2011.

[12] Parviz, Kermani, Leonard and Kleinrock, "Virtual CutThough A New Computer Communication Switching Technique, " Holland Publishing Company, Computer Networks vol. 3, pp. 267–286, 1979.

[13] Daniel U, Becker, William J, Dally, "Allocator Implementations for Network-on-Chip Routers". In Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis (pp. 1–12). IEEE. November. 2009.

[14] Shaoteng Liu, Axel Jantsch, Zhonghai Lu, "A Fair and Maximal Allocator for Single-Cycle On-Chip Homogeneous Resource Allocation," IEEE Transactions on Very Large Scale Integration (VLSI) Systems) PP. 2230 – 2234, Vol. 22, no 10, Oct. 2014.

[15] William James Dally, Brian Patrick Towles " Principles and Practices of Interconnection Networks, " Morgan Kaufmann Publishers Inc., USA, PP 164–165, Jan 01. 2004.

[16] Subhasakthe SA and Manoj KK and Mithunraj S, "A Reliable Flexi-Router Architecture for NoC Applications," International Journal of Emerging Technology and Advanced Engineering, Vol. 4, no. 1, Jan. 2014.

[17] L.Rooban, S.Dhananjeyan, "Design of Router Architecture Based on Wormhole Switching Mode for NoC, " International Journal of Scientific Engineering Research Volume 3, no.3, ISSN 2229 5518, March, 2012.

[18] Omprakash Ghorse1, Nitin Meena, Shweta Singh "Review On Different Types Of Router Architecture And Flow Control," International Journal of Engineering Trends and Technology (IJETT) Vol. 4, no. 10, Oct. 2013.

[19] William Dally and Brian Towles, "Principles and Practices of Interconnection Networks," Morgan Kaufmann Publishers Inc., San Francisco, USA, December 18. 2003.

[20] R. Mullins, A. West, S. Moore, "Low-latency virtual-channel routers for on-chip networks, " Proceedings of the 31st Annual International Symposium on Computer Architecture, vol.32, no.2, pp. 188–197, Jun. 2004.

[21] Daniel U. Becker, " Efficient Microarchitecture for Network-on-chip Router," Ph.D. dissertation submitted to the department of electrical engineering of stanford university, Aug. 2012.

[22] R. Mullins, A. West, and S. Moore, "Low-latency virtual-channel routers for on-chip networks, " Proceedings of the 31st Annual International Symposium on Computer Architecture, vol.32, no.2, pp. 188–197, Jun. 2004.

[23] Suyog K. Dahule, Reetesh V. Golhar, Mangesh D. Ramteke "The Behavior of Round Robin Arbiter in NOC Architecture, " International Journal of Engineering and Innovative Technology (IJEIT) Vol. 3, no. 5, Nov. 2013.

[24] Zhizhou Fu,Xiang Ling, "The design and implementation of arbiters for Network-on-chips, " International Conference on Industrial and Information Systems, vol. 1, no. 2, pp 292–295, 2010.

[25] Z. Lu and A. Jantsch, "Flit ejection in on-chip wormhole-switched networks with virtual channels," International Conference on Norchip, pp. 273–276, Nov. 2004.

[26] L. M. Ni and P. K. McKinley, "A survey of wormhole routing techniques in direct networks," Computer, vol. 26, no. 2, pp. 62–76, 1993.

[27] Swaminathan K, Thakyal, Deewakar, Nambiar, Sandeep Gopi, Lakshminarayanan G and Ko Seok-Bum, "Enhanced noxim simulator for performance evaluation of network on chip topologies, " Engineering and Computational Sciences (RAECS), 2014 Recent Advances, pp. 1–5, March 6–8. 2014.

[28] Y. Ben-Itzhak, E. Zahavi, I. Cidon, and A. Kolodny, " Hnocs: Modular open-source simulator for heterogeneous nocs, " in Embedded Computer Systems (SAMOS), International Conference on, pp. 51–57, July. 2012.

[29] N. Jiang, D. Becker, G. Michelogiannakis, J. Balfour, B. Towles, D. Shaw, J. Kim, and W. Dally, "A detailed and exible cycle-accurate network-on-chip simulator," in Performance Analysis of Systems and Software (ISPASS), IEEE International Symposium on, pp. 86–96, April 2013.

[30] Sriram Prakash Adiga "NoC characterization framework for design space exploration, " Computer Engineering Department of Electrical Engineering Delft University of Technology, vol.52, pp.6–893662, 2014.

[31] Arnab Banerjee " Communication flows in power-efficient Networks-on-Chips, " Thesis, University of Cambridge, August. 2010.

[32] W. Dally and B. Towles. "Principles and Practices of Interconnection Networks, " Morgan Kaufmann Publishers Inc, 2003.

[33] Aminu, Shehu Mahdi, " Effective Dynamic Re-mapping Algorithm for low power Networkon-chip NoC, " September. 2010.

[34] Wikipedia the free encyclopaedia, " Multi-core processor, " Internet: http://en.wikipedia.org/ wiki/Multi-core.2016.

[35] M. Weber, " Arbiters: Design ideas and coding styles", http://www.siliconlogic.com/sle asic fpga engineering papers.asp; accessed, June 8, 2008.

[36] M. Coppola, M. D. Grammatikakis, R. Locatelli, G. Maruccia, and L. Pieralisi, " Design of cost-efficient interconnect processing units: Spidergon STNoC, " CRC press, 2008.

[37] Sungho park, " A Veriog-HDL implementation of virtual channels in a network-on-chip router, " Thesis, Univ. Texas, August. 2008.

[38] W. Dally and B. Towles, "Principles and Practices of Interconnection Networks, " San Francisco, CA, Morgan Kaufmann Publishers Inc., 2003.

[39] Daniel U. Becker "Efficient Microarchitecture For Network-on-chip Router, " Ph.D. dissertation, Dept. Electrical, Stanford University, Aug. 2012.

[40] Bjerregaard, Tobias and Mahadevan, Shankar " A survey of research and practices of network-on-chip, " ACM Computing Surveys (CSUR) vol. 38, no.1, pp.1, 2006.

[41] Minghua Tang and Xiaola et al,"Injection Level Flow Control for Network-onChip (NoC)," Journal of information science and engineering, vol. 27, no.2, pp. 527–544, 2011.

[42] K. Paliwal, M. Gaur, V. Laxmi, and V. Janyani, " Performance analysis of guaranteed throughput and best eort trafc in network-on-chip under dierent trafc scenario, " in Future Networks, International Conference on, pp. 74–78, March. 2009.

[43] Rabab Ezz-Eldin, Magdy, Ali El-Moursy, Hesham F, A. Hamed, " Analysis and Design of Networks-on-Chip Under High Process Variation, " pp. 508–509, 2015.

[44] Rika, de. Morais Amory, Alexandre, Lubaszewski and Marcelo Soares, " Reliability, Availability and Serviceability of Networks-on-Chip, " pp. 11–24, 2012.