

Simulating NoC Mesh and Torus Topologies

by

Muhammad Ahsan Khan

B. E. SIR SYED UNIVERSITY OF ENGINEERING AND TECHNOLOGY, KARACHI,
PAKISTAN, 2011

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

MASTER OF ENGINEERING

in the Department of Electrical and Computer Engineering

©Muhammad Ahsan Khan, 2017

University of Victoria

All rights reserved. This thesis may not be reproduced in whole or in part, by
photocopy or other means, without the permission of the author.

Supervisory Committee

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

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

Abstract

An interconnection network is a programmable system that transports the data between the terminals. The interconnection is important because of the limiting factor in the performance of many systems. Network on chip (NoC) plays a vital role in the memory latency or memory bandwidth, which are the two key performances in computer systems. Apart from them the topologies are also one of the most important performance factors. In this project the two most significant topologies, mesh topology and torus topology are studied. A study is conducted on the above two mentioned topologies by injecting various flit rates with different combinations of virtual channels. The main objective of this project is to explain how virtual channels are effective on throughput and latency on different topologies. The comparative evaluation of topologies will help to explore more features in detail which will be helping in future developing in NoC.

Contents

Abstract	iii
Table of Contents	v
List of Figures	vi
List of Tables	vii
Acknowledgments	viii
Dedication	ix
Abbreviations	x
Definitions	xi
1 Introduction	1
2 Introduction to Booksim	3
2.1 Need of Simulators	3
2.2 NoC Simulators	4
2.2.1 Origin of Booksim	4
2.2.2 Booksim2.0 Features	5
3 Overview Of NoC	6
3.1 Evolution of NoC	6
3.2 NoC Features	6
3.3 NoC Architecture	7
3.3.1 Switching Technique	7
3.3.2 Flow Control	10
3.3.3 Buffered Flow Control	11
3.3.4 Bufferless Flow Control	12
3.3.5 Virtual Channels	12
3.3.6 Topology	13
3.3.7 Routing Algorithm	15
3.3.8 Routing Microarchitecture	16

4	Simulation Scenario and Results for the selection parameters	18
4.1	4x4 Mesh Topology Simulation Results	18
4.2	4x4 Torus Topology Simulation Results	20
4.3	8x8 Mesh Topology Simulation Results	21
4.4	8x8 Torus Topology Simulation Results	23
5	Conclusion	25
	Bibliography	26

List of Figures

Figure 3.1 Switching Techniques Classification	8
Figure 3.2 Flow Control Techniques Classification	11
Figure 3.3 Ring Topology Layout	14
Figure 3.4 [4] Mesh Topology Layout	14
Figure 3.5 [4] Torus Topology Layout	15
Figure 3.6 [4] Fat Tree Topology Layout	15
Figure 3.7 Router Architecture Layout	16
Figure 4.1 4x4 Mesh Topology Throughput VS Injection rate (flits/cycle) .	19
Figure 4.2 4x4 Mesh Topology Average Packet Latency VS Injection rate (flits/cycle)	19
Figure 4.3 4x4 Torus Topology Throughput VS Injection rate (flits/cycle)	20
Figure 4.4 4x4 Torus Topology Average Packet Latency VS Injection rate (flits/cycle)	21
Figure 4.5 8x8 Mesh Topology Throughput VS Injection rate (flits/cycle) .	22
Figure 4.6 8x8 Mesh Topology Average Packet Latency VS Injection rate (flits/cycle)	22
Figure 4.7 8x8 Torus Topology Throughput VS Injection rate (flits/cycle)	23
Figure 4.8 8x8 Torus Topology Average Packet Latency VS Injection rate (flits/cycle)	24

List of Tables

Table 2.1 NoC Simulator Characteristics 4

Acknowledgments

In the name of Allah, the Most Gracious and the Most Merciful All praises belong to Allah the merciful for his guidance and blessings to enable me complete this project. I would like to thank:

My parents, for their love, support, prayers and for their constant motivation.

My Supervisor, Dr. Fayez Gebali, for all the mentoring and support which enabled me to achieve my academic and research objectives. It would not have been possible to complete my project without his invaluable guidance.

UVIC ECE Dept and Graduate office, Ashleigh Burns , Amy Rowe and Scott Baker for assisting me during the course of my degree.

My best friend, Usama Khan, he helped me alot throughout my graduation.

Dedication

To my father, **Hasan Khan** and my mother, **Najma Hasan** and my wife, **Asna Ahsan** without their care and support it would not have been possible. In difficult times, it proved as key motivating factor and enabled me to maintain focus.

To my Supervisor, **Dr. Fayez Gebali**, he is always helpful and one of the most kindest and knowledgeable person I have ever met. I truly appreciate his dedication and precious time he spent with me for his guidance.

Abbreviations

Network On Chip (NoC)

Physical Channel (PC)

Quality of Service (QoS)

Resource Network Interface (RNI)

Real Time (RT)

Register Transfer Level (RTL)

System On Chip (SoC)

Virtual Channel (VC)

Virtual Component Interface (VCI)

Input-Queued (IQ)

Giga Scale Research Center (GSRC)

Definitions

Flit: Flit is an acronym for Flow control digit. The smaller pieces of a large network packets is termed as Flit.

Topology: Topology determines the physical layout and connection between nodes and channels in the network.

Routing Algorithms: Routing determines a path for a message to take through in the network to reach its destination.

Flow control: Determines how shared resources, such as buffers and channel bandwidth are utilized when contention occurs.

Traffic Pattern: It is the spatial distribution of messages in the network.

Injection Rate: It is defined as average number of packet injected per cycle.

Throughput: It is defined as data rate in bits per second at input port. It measures how fast the message can pass through NoC.

Latency: Time requisite for a packet to navigate the network, time taken by the head-packet to arrive at the input port to the time taken by tail-packet to exit the output port.

Hop Counts: The distance between nodes considered as the average number of channels and nodes a message must navigate from source node to destination node.

Path Diversity: A property of network which has multiple paths between two nodes, this gives robustness to the network.

Chapter 1

Introduction

In order to meet the current requirements of such fast growing technological world where most of the applications are growing as computation-intensive and also the need of low-power, high-performance systems has resulted in an enormous increase in the number of computing resources on a single-chip [9] [6]. Thus to build a System-on-chip comprising of many computing resources like CPU, DSP, specific IPs etc. the interconnection between each other is a very challenging and a critical issue.

A shared bus interconnection is mostly used in System-on-Chip applications. This shared bus interconnection requires arbitration logic. This logic helps to serialize several bus access requests. This is usually adopted because of its low cost and simple control characteristics. Since only one master can use the bus at a time while the remaining requests are serialized by the arbitrator. Therefore, to avoid the scalability limitation some other interconnection method should be considered to cope up with the environment with large number of bus requests and to meet the bandwidth requirement [7].

Thus to overcome the bottlenecks which impact the overall system performance a scalable and modular design approach was proposed as Network-on-chip (NoC). It allows the messages to flow from the source to destination module through several links. The routing decisions in NoC were carried out at switches [4]. The NoC proved to be an effective alternative to the traditional bus based architecture for inter core communication. This is mainly due to they provide

- Scalable bandwidth at low power and area overheads.
- Efficiency in terms of use of wiring and multiplexing.
- Easy verification process.

Therefore to understand the performance characteristics, bottlenecks and the impact on overall system for proper modeling and evaluation a number of simulators were used some of which are: LISNOC, NS-2, Noxim, Orion 3.0, Nirgam, Booksim 2.0 etc [10]. In this project Booksim 2.0 is used as a network simulator. Booksim is detailed and flexible cycle-accurate NoC simulator. It features a modular design and offers a large set of configurable network parameters in terms of topology, routing algorithm, flow control, and router microarchitecture. It also includes buffer management and allocation schemes. Booksim emphasizes on detailed implementations of network components that accurately model the behavior of actual hardware and the accuracy of the simulator matches with the RTL implementations of NoC routers [10].

Chapter 2

Introduction to Booksim

This chapter includes a brief introduction of the simulator used in this project to observe the response of NoC under certain parameters.

2.1 Need of Simulators

Simulation can be defined as the designing of an actual or theoretical physical systems and executing it on a digital computer to analyze the output on the given inputs. This helps the researchers in exploring and analyzing the design model as well as to evaluate its performance and efficiency of the system. Therefore, to assist the researchers simulators were designed. Simulators are defined as software used to model devices to predict output and performance on a given input.

Simulation is classified as the following two types:

i) Cycle-accurate simulation

As the name suggests cycle-accurate simulation, simulates a micro architecture on a cycle by cycle basis. This type of simulation is used where time precision is very important [3]. It helps in checking the events that took place at a current time as it is incremented in fixed steps. The cycle-accurate simulation is necessary when the actual routers RTL description needs to be evaluated and verified.

ii) Event-driven simulation

In this type of simulation the flow of control within the system is driven by events rather than the sequence. The events in such systems are not guaranteed to occur at regular interval of time. Event-driven simulation uses a list of events occurring at various times, and handles them in order of increasing time [3].

2.2 NoC Simulators

The design decisions of NoC architectures are usually made on the basis of simulation before resorting to implementation since it is cheap and flexible. It helps in exploring the various factors like architectural design space, assessing of quality of design on the basis of cost, power consumption, reliability and performance [2]. The characteristics of different types of NoC simulators are described in the below Table 2.1:

Table 2.1: NoC Simulator Characteristics

Simulator	Framework	Availability	Topologies	Open Source
SICOSYS	C++	+	Limited	+
Noxim	SystemC	+	Mesh	+
Booksim	C++	+	Many	+
NNSE	SystemC	+	Mesh/Torus	+
Nirgam	SystemC	+	ALL	+
gpNoCsim	Java	+	ALL	+
Parm NoC	OMNet++	-	ALL	-
HNOc	OMNet++	+	ALL	+

The simulator used for this project is Booksim.

2.2.1 Origin of Booksim

The original simulator (BookSim1) was released as part of an interconnection network textbook by Dally and Towles. It was used to generate the performance graphs in the textbook. BookSim1 was a generic network simulator that did not specifically target the on-chip environment. As a result, it has been widely used for research in

many network contexts, including networks found in large-scale supercomputers and many-core processors [11].

2.2.2 Booksim2.0 Features

BookSim network simulator is a detailed, cycle-accurate simulator for Network-on-Chips (NoC). It can also be used to model interconnection network for a variety of other systems. The current version of the BookSim simulator, BookSim2 improves various aspects of BookSim1 while incorporating new features for simulating NoC. The simulator maintains its cycle-accurate nature [4] [1]. A greater emphasis is placed on the detailed modeling of network components based on realistic hardware implementations. Input-Queued (IQ) router microarchitecture model has been improved to facilitate more accurate simulation. New features and optimizations have been added to the basic structure of IQ Routers to reflect the current trends in the NoC research. Booksim2 also includes increased flexibility for traffic generation and also the integration with other traffic resources [10]. Since channel latency has major impact on the network performance like buffer utilization, credit round-trip latency and the propagation delay for congestion information in adaptive routing algorithms. Therefore, accurate modeling of channels latencies is also introduced which has significantly affected the results of Booksim2 as compared to Booksim1 [8].

Chapter 3

Overview Of NoC

This chapter presents the evolution, features and a brief description of the basic architecture of a Network on Chip (NoC).

3.1 Evolution of NoC

Presence of challenges results in major technological innovations. NoC was also the result of such technological challenges faced. NoC was proposed in around year 2001 in the SoC community. In March 2000, packet-switched networks were proposed in SPIN as a global and scalable SoC interconnection [6] [7]. The term Network-on-Chip appeared initially in November 2000 where NoC was proposed as a platform to cope with the productivity gap. In June 2001, Dally and Towles proposed NoC as a structured way of communication to connect IP modules. The GigaScale Research Center (GSRC) suggested NoC to address interconnection woes. In October 2001, researchers from the Philips Research presented a router architecture supporting both best-effort and guaranteed-throughput traffic for Network-on-Silicon. In January 2002, Luca and De Micheli formulated NoC as a new SoC paradigm.

3.2 NoC Features

As NoC is more of a revolutionary approach to address the SoC design crisis. It holds the following features:

- Assistance in simplifying of the hardware required for routing and switching functions.

- For different areas of a network the multi-topology and multi-option support is possible.
- Enhancement of scalability, interoperability and feature development is observed when combined with network on a chip.
- As compared to other designs the power efficiency of complex SoC is improved with NoC.
- Better handling of synchronization issues.
- Better handling of wire routing congestion present in most SoC.
- Provision of higher operating frequencies.
- Easier implementation of time closure.
- Verification of problems is much easier.

3.3 NoC Architecture

NoC is a nano scale packet switched network. It is universal structure, which connects all functional units on the chip. It is an on-chip network having cores connected by switches [4]. The switches are then connected among themselves through communication channels. The basic architecture of NoC consists of the following:

3.3.1 Switching Technique

Switching is an important parameter of NoC architectures. Switching strategy determines the flow of data through routers in the network. Switching techniques are classified on the characteristics of the network.

The basic two types of switching are demonstrated in the below block diagram Figure 3.1. Apart from the basic two switching strategies commonly adopted, ad-hoc switching techniques can be also developed by combining different switching techniques [1].

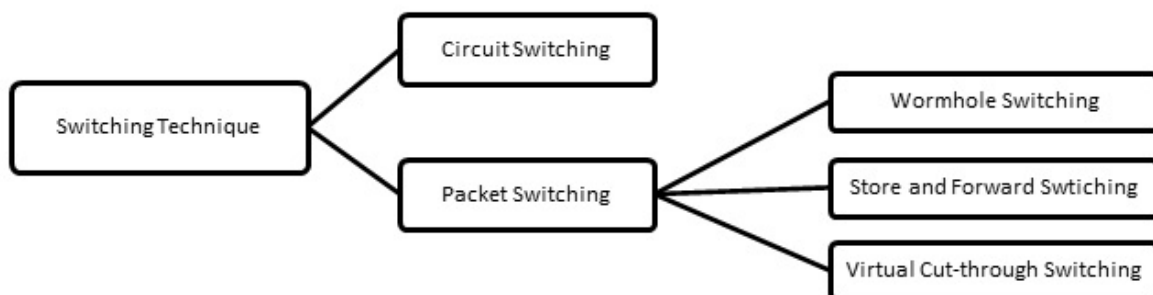


Figure 3.1: Switching Techniques Classification

Circuit Switching

In circuit switching before the transmission of data, a dedicated end-to-end path between the source and the destination is reserved. The reserved path can be a real or virtual circuit. The path reserved with the associated source is released as soon as the data transmission is achieved. There is an explicit connection established which makes this switching technique as connection oriented.

In Circuit switching the data transfer latency is reduced as soon as the path is reserved for transmission which is one of its main advantages. There is very low probability of packet loss since a link is dedicated from source to destination [3] [2]. However this technique does not scale well with NoC size. Also the links in circuit switching are occupied for considerable amount of time.

Packet Switching

The most common switching technique used in NoC is the packet switching. In packet switching the message is delivered without reserving the entire path. The message is delivered from source to destination into a sequence of packets. A packet comprises of a header (it contains the routing and sequencing information), payload (it is the actual data to be transmitted) and a tail (it contains the error checking code). Since packets are transferred to the destination through different routes a variable delay is introduced due to contention in router along packet path [12].

The packet switching technique in NoC is either connection-oriented or connection-less. The main difference between the two is that the resource is preserved in

connection-oriented while the connection-less communication does not preserve the source. Apart from source preservation a certain degree of commitment for message delivery bounds is provided in connection oriented communication. However, in connection-less communication it is difficult to provide bounds since the packets are routed individually and the message delivery is subjected to dynamic contention scenarios. However, there can be a better utilization of network resources. Packet switching technique is further classified as following:

1. Wormhole Switching
2. Store and Forward Switching
3. Virtual Cut-Through Switching

1) Wormhole Switching

In wormhole switching technique the packets are split into several flits. Therefore this results in a drastic reduction in buffer size to the size of flit instead of packet size. In wormhole switching a delay is observed in the decision of path by the header flit only. The remaining flits of the same packet follow the same path as that of the header flit. The main disadvantage observed in wormhole switching is higher latency which makes it unsuitable for real time data transfer. Also, the entire packet is blocked if the header flit is blocked [3]. This switching technique is more vulnerable to deadlock due to dependencies between links.

2) Store and Forward Switching

In Store and forward switching the packet is not divided into flits. A packet is transmitted only when the receiving buffer has enough space available to hold the entire packet. This results in the reduction on the overhead cost, as circuits like flit builder, flit decoder, flit stripper and flit sequencer are not required. However, a large amount of buffer space is required at each node to hold the packet. Therefore, this technique may not be suitable for embedded applications [11].

3) Virtual Cut-Through Switching

In VCT switching, a packet is divided into flits, which may be further divided into flits. Therefore, it has the same buffer requirement as Store and Forward switching technique. However, in VCT a network node does not wait to receive an entire packet. Instead after receiving the packet it forwards it downstream depending on the availability of buffer space in the next switch. Therefore, it is necessary that

the downstream node must be equipped with sufficient buffers to hold the entire packet [6]. If blocking is observed the entire packet is shunt into the allocated buffers.

3.3.2 Flow Control

The allocation of network buffers and links are governed by Flow control. It determines when to which messages buffers and links are assigned, the granularity at which they are allocated. Flow control also determines how these resources are shared among the many messages using the network. A good flow control protocol lowers the latency experienced by messages at low loads [5]. It achieves this by not imposing high overhead in resource allocation, and pushes network throughput through enabling effective sharing of buffers and links across messages.

Flow control is instrumental in determining network energy and power by the rate at which packets access buffers (or skip buffer access altogether) and traverse links [9]. Flow control also critically affects network quality of service since it determines the arrival and departure time of packets at each hop. The implementation complexity of a flow control protocol includes the complexity of the router microarchitecture. It also includes the wiring overhead imposed in communicating resource information between routers [7]. The flow control of a network may be coupled with its switching strategy. The store-and-forward and virtual cut-through switching technique use the packet-buffer flow control, and wormhole switching technique uses the flit-buffer flow control. The flow control is mainly classified as illustrated in the following Figure 3.2.

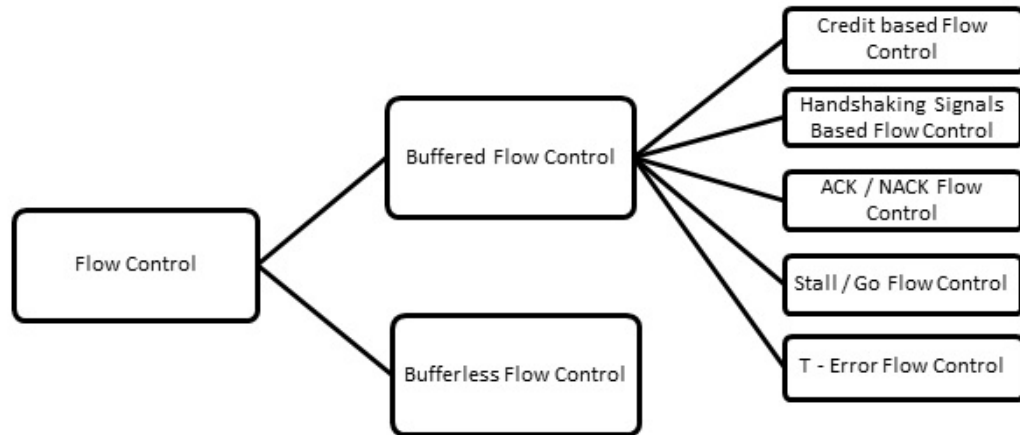


Figure 3.2: Flow Control Techniques Classification

3.3.3 Buffered Flow Control

The buffered flow control technique is mainly used for packet switched networks. In buffered flow control the blocked packets are stored till they wait to acquire network resources [1]. The resource allocation may vary for different buffered flow control techniques. The buffered flow control is further classified into the following:

1. Credit based Flow Control
2. Handshaking Signal Based Flow Control
3. ACK / NACK Flow Control
4. Stall / Go Flow Control

1) Credit based Flow Control

In Credit based Flow Control a count of data transfer is kept by the upstream node. After a data packet is transmitted, a credit (available free slots) is sent back [2].

2) Handshaking Signal Based Flow Control

In Handshaking Signal Based Flow Control, whenever any flit is transmitted by a sender a Valid signal is sent. After consuming the data flit, the receiver acknowledges by asserting a Valid signal [3].

3) ACK / NACK Flow Control

In ACK / NACK flow control the buffer stores the copy of data until an ACK signal is received. The buffer then deleted the copy of the flit on receiving the ACK Signal. However if NACK signal is received the flit is scheduled for retransmission [2].

4) Stall / Go Flow Control

In Stall / Go Flow Control, the flow is controlled with the help of two wires between each pair of sender (producer) and receiver (consumer). A Go signal is activated when there is an empty buffer space while a Stall signal is activated on the unavailability of buffer space. However, this flow control scheme is employed by none of the present implemented NOC [3].

3.3.4 Bufferless Flow Control

The simplest form of flow control is the bufferless flow control. The Bufferless flow control depends on arbitration. After arbitration the winning packet advances over the link while the remaining packets are either dropped or misrouted due to the absence of buffers [6]. The allocation of resources is the link bandwidth due to the absence of buffering in switches. In Bufferless Flow Control more latency and less throughput is present as compared to the Buffered Flow Control.

3.3.5 Virtual Channels

Virtual Channel is another important design aspect of NoC. For the purposes of deadlock avoidance virtual channels were introduced. In a virtual channel a single channel is split into two or more channels (varies from two to eight VC) , thus virtually providing paths for the packets to be routed. A buffer enables a virtual channel to hold one or more flits of a packet and the associated state information. The bandwidth of a single physical channel is shared by several virtual channels [8].

In an interconnection network the most costly resource is physical channel (wire) bandwidth while the second most costly resource is buffer memory. With the addition of virtual channel flow control to a network makes more effective use of both of these resources by decoupling their allocation. The only expense then left is a small amount of additional control logic. In addition to increase in throughput, virtual channels provide an additional degree of freedom in allocating resources to packets in the network. This flexibility allows the use of scheduling strategies, like routing the oldest packet first, which reduces the variance of network latency [5].

3.3.6 Topology

The network graph of the physical structure is referred as topology. It defines the physical connection between the network nodes (switches or routers) and also the connectivity (routing possibilities) between the nodes. Therefore, topology has a profound impact on the network performance as well as the switch structure [12].

Topology also helps in determining the number of hops (or routers) a message must traverse. It also determines the interconnect lengths between hops, which has a significant influence on network latency. The network energy consumption is directly affected by the topology's effect on hop count as traversing routers and links incurs energy. As for its effect on throughput, since a topology dictates the total number of alternate paths between nodes, it affects how well the network can spread out traffic and thus the effective bandwidth a network can support [2]. The topology also greatly influence the network reliability as it dictates the number of alternative paths for routing around faults. The implementation complexity cost of a topology depends on two factors:

- The number of links at each node (node degree).
- The ease of laying out a topology on a chip (wire lengths and the number of metal layers required).

The most common topologies used in a network are

- Ring Topology
- Mesh Topology
- Torus Topology
- Fat Tree Topology

Ring Topology

In Figure 3.3 the node is directly connected with two other nodes forming a circular shape. The packet is sent to the next node around the circle until it reaches its last destination.

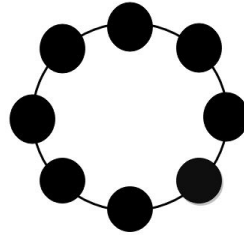


Figure 3.3: Ring Topology Layout

Mesh Topology

The mesh topology consists of connection of each node as a 2D lattice. Each node in a mesh topology is connected to the adjacent node. The total number of nodes in a mesh topology can be defined by where k^n is the network radix while n is the network dimension. Figure 3.4 Shows mesh topology of $k = 3$ -ary and 2nd order $n = 2$. Load imbalance is observed in a mesh topology as the load to the central nodes is more than the edge nodes. Also, there is no edge symmetry present in mesh topology [3].

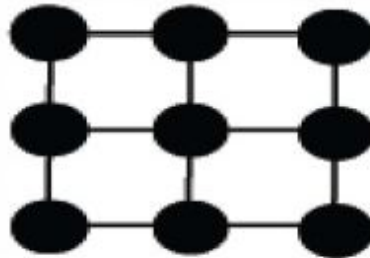


Figure 3.4: [4] Mesh Topology Layout

Torus Topology

Torus topology is the improved version of mesh topology. Unlike mesh topology, torus topology is edge symmetric with lower hop count and greater path diversity. This is achieved by connecting the head of each column to the tail of each column and connecting the leftmost node of each row to the rightmost node of each row. Figure 3.5 illustrates the torus topology [9].

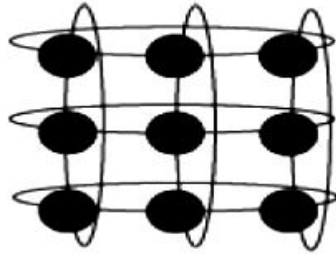


Figure 3.5: [4] Torus Topology Layout

Fat-Tree Topology

The nodes in a fat-tree topology are connected in a tree like structure. In this topology the number of links between the different levels remains constant. The distinctive feature of a fat-tree is the number of links going down to the siblings will be equal to the number of links going up to its parents. Figure 3.6 illustrates example of fat tree topology [4]. Thus, links get fatter towards the top of the tree.

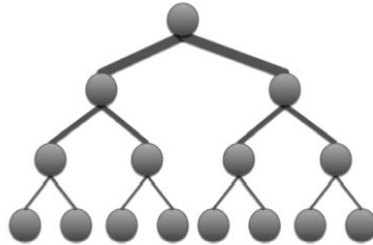


Figure 3.6: [4] Fat Tree Topology Layout

3.3.7 Routing Algorithm

Routing in a network determines a path for a message to take through the network to reach its destination. The goal of routing algorithm is to distribute traffic evenly among the paths supplied by the network topology. It helps in avoiding hotspots and minimizing contention, thus improving network latency and throughput. In addition to this, the routing algorithm is the critical component in fault tolerance. On the identification of faults, the faulty nodes and links must be skirt by the routing algorithm without affecting network performance significantly. All of these performance

goals must be achieved while adhering to tight constraints on implementation complexity [11]. While energy overhead of routing circuitry is typically low, the specific route chosen affects hop count directly which affects energy consumption considerably.

3.3.8 Routing Microarchitecture

Router architecture consists of input buffer, routing logic, virtual channel allocators and crossbar or switch. The function of router architecture is to route the packet from source node to destination node with minimum latency. Each of these affects the performance of NoC [7]. Along with these parameters, there are some synthetic parameters used in simulation like, traffic throughput and injection rate etc. which also affects the performance of NoC.

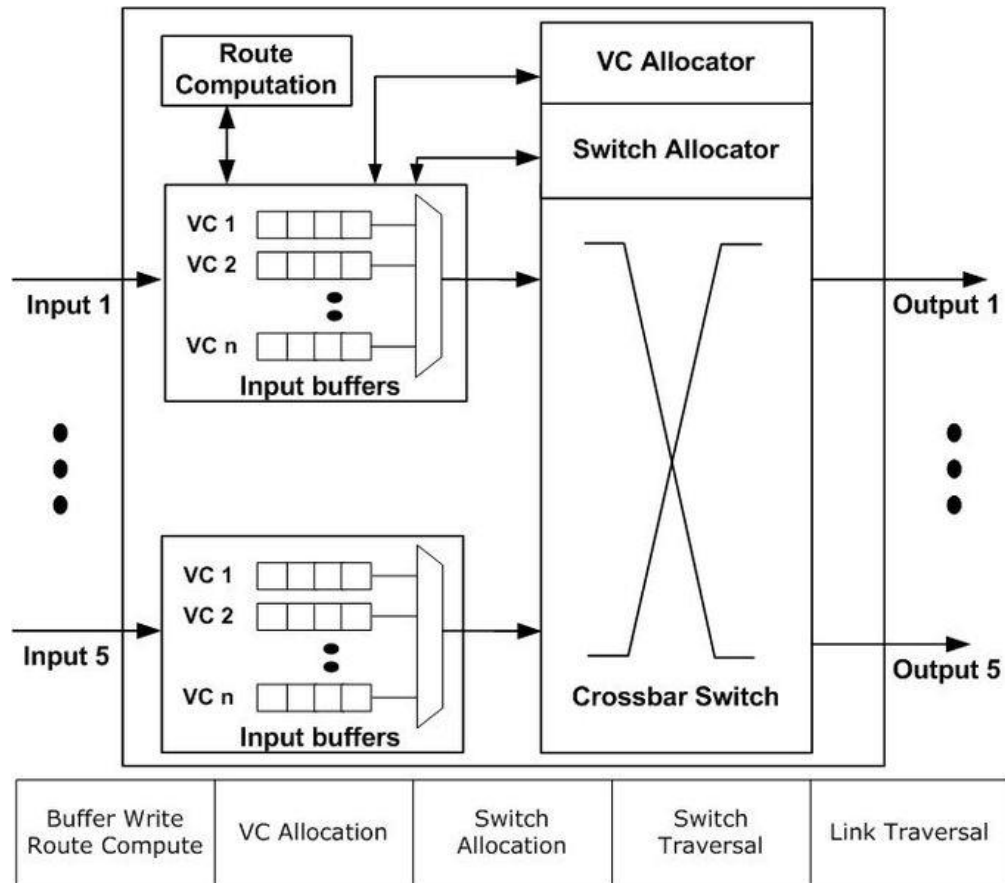


Figure 3.7: Router Architecture Layout

Figure 3.7 can be divided in two groups based on functionality: the control plan and data path. In this virtual channel router the control plan perform virtual-channel

allocation, switch allocation and route computation. This control block is also responsible for coordinating the movement of packets through the resources of the datapath. The data path of the router handles a switch, a set of output buffers, a movement of packet payload and consists of a set of input buffers.

Chapter 4

Simulation Scenario and Results for the selection parameters

Simulation results for various topologies: A detailed comparison of throughput, average packet latency and injection rate has been conducted for two different topologies (Mesh Topology and Torus Topology).

4.1 4x4 Mesh Topology Simulation Results

The graph in Figure 4.1 shows detailed comparison between the throughput and injection rate of 4x4 mesh topology using multiple virtual channels. After achieving the saturation point all the virtual channels behaved steadily. However, the VC8 shows the 60 percent rise as compared to the others and then it gets steady. The lowest peak is experienced by VC1 with the increase of injection rate.

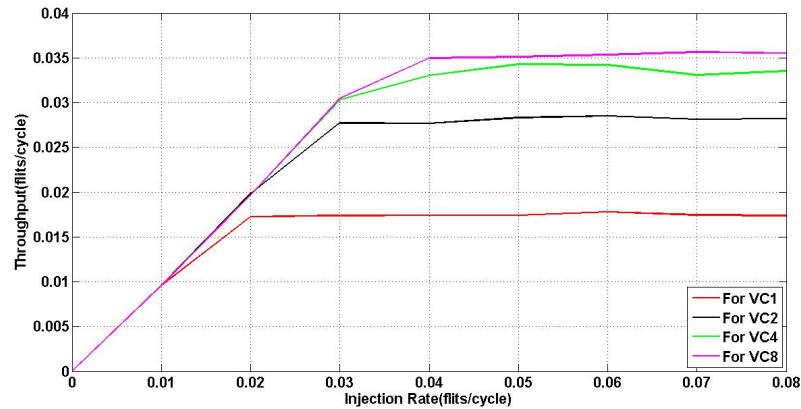


Figure 4.1: 4x4 Mesh Topology Throughput VS Injection rate (flits/cycle)

In Figure 4.2 we have obtained the detailed comparison between average packet latency and injection rate using 4x4 mesh topology. However, the increase in average packet latency can be seen gradually. Similarly VC16 achieved the lowest average packet latency with an increase injection rate. The other channels VC4 and VC8 show consistent average packet latency similar to the VC2.

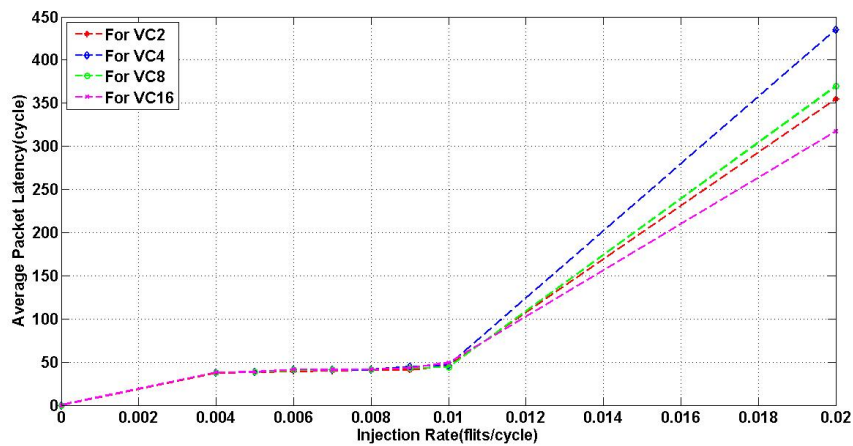


Figure 4.2: 4x4 Mesh Topology Average Packet Latency VS Injection rate (flits/cycle)

4.2 4x4 Torus Topology Simulation Results

In this section a detailed comparison of throughput vs injection rate and average packet latency vs injection rate has been observed for multiple Virtual channels.

In the following Figure 4.3 we have obtained the detailed comparison of throughput vs injection rate by using 4x4 torus topology. It can clearly be observed that the VC16 has highest throughput as compared to the other VC's. The threshold value of VC16 is more than 0.8 flits/cycle.. Similarly, VC2 has shown the lowest throughput value among all the VC's.

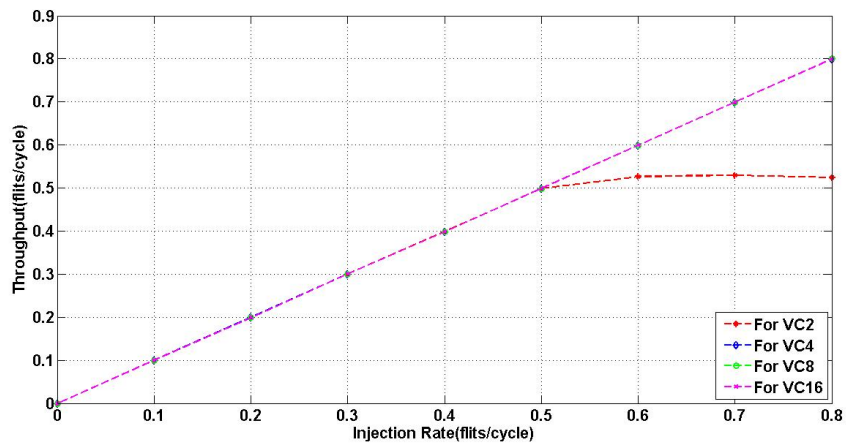


Figure 4.3: 4x4 Torus Topology Throughput VS Injection rate (flits/cycle)

In Figure 4.4 a comprehensive comparison of average packet latency vs injection rate has been obtained which clearly shows that VC4 achieved the highest peak of average packet latency with the increase injection rate. However, the increase in average packet latency can be seen gradually after 0.02 flits/cycle. Similarly, VC16 achieved the lowest average packet latency with an increase injection rate. The other channels VC2 and VC8 show consistent average packet latency similar to the VC16.

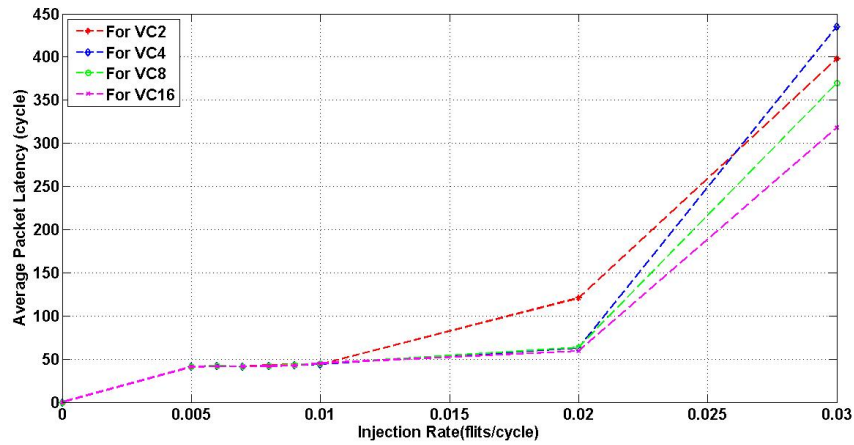


Figure 4.4: 4x4 Torus Topology Average Packet Latency VS Injection rate (flits/cycle)

4.3 8x8 Mesh Topology Simulation Results

Figure 4.5 shows a detailed comparison between the throughput and injection rate for multiple virtual channels by using 8x8 mesh topology which shows how the throughput significantly increases with the increase of injection rate. Initially, there is a peak in the throughput with the lowest injection rate and it gets stable at higher values. The same implication has been carried out for different virtual channels where VC8 has experienced the highest throughput among all the channels with the increase injection rate. However VC1 has shown the lowest throughput with the increase injection rate.

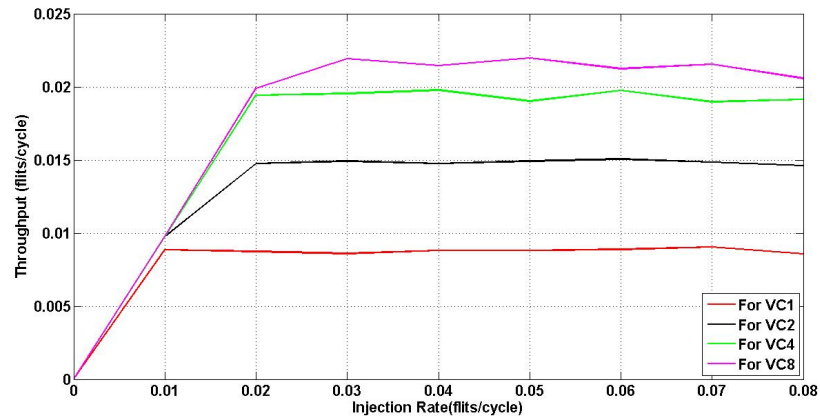


Figure 4.5: 8x8 Mesh Topology Throughput VS Injection rate (flits/cycle)

The graph in Figure 4.6 depicts a change in average packet latency with respect to injection rate. It can clearly be seen that at higher rate of injection rate the average packet latency did not show a significant increase but as it reaches to the higher injection rate the average packet latency increases significantly. For different multiple channels it can clearly be seen the highest average packet latency is achieved for VC 4. And the lower average packet latency is shown by channel VC2.

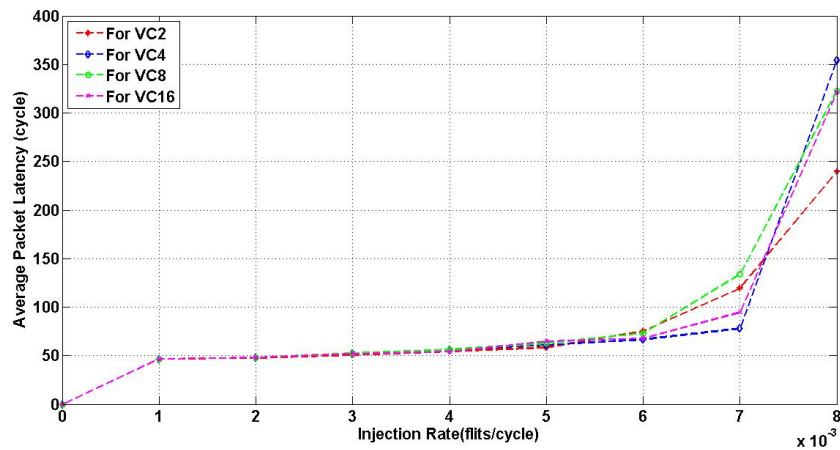


Figure 4.6: 8x8 Mesh Topology Average Packet Latency VS Injection rate (flits/cycle)

4.4 8x8 Torus Topology Simulation Results

In Figure 4.7 we have obtained a detailed comparison of throughput vs injection rate by using the 8x8 torus topology. It can clearly be observed that the VC16 has shown the highest throughput rate with the increase of injection rate till the threshold value of 0.7 flits/cycle and started to decrease after this threshold value. Similarly, VC2 has shown the lowest throughput value among all the virtual channels. VC2 achieved a throughput peak at 0.3 flits/cycle and became stable after a slight decrease in throughput with the increase in injection rate.

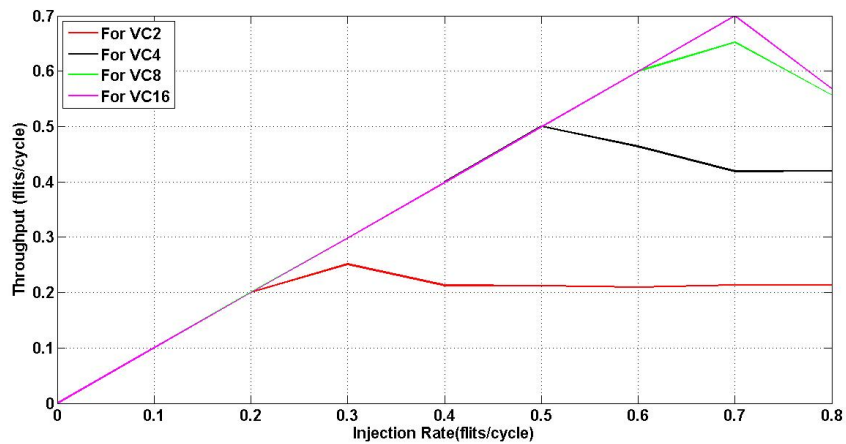


Figure 4.7: 8x8 Torus Topology Throughput VS Injection rate (flits/cycle)

In Figure 4.8 a detailed comparison of average packet latency vs injection rate has been obtained which clearly shows that VC8 achieved the highest peak of average packet latency with the increase injection rate. However, the increase in average packet latency can be seen gradually. Similarly, VC2 achieved the lowest average packet latency with an increase injection rate. The other channels VC4 and VC16 show consistent average packet latency similar to the VC8.

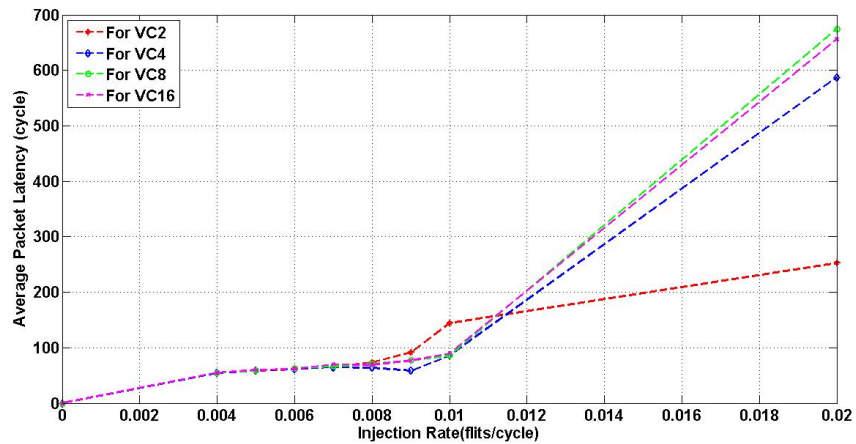


Figure 4.8: 8x8 Torus Topology Average Packet Latency VS Injection rate (flits/cycle)

Chapter 5

Conclusion

From all the simulation results it can be concluded that every topology in network on chip has different behaviour for the performance parameters (throughput, average packet latency and injection rate) considering multiple virtual channels. As shown in Figure 4.3 it can be observed as the injection rate increases the throughput of the 8x8 torus topology increases linearly but at one point it get saturated where we get the highest saturation throughput point as compared to others. From Figure 4.6 it can be seen that 4x4 mesh topology has highest latency rate as the injection rate increase the average latency of the network show consistent line upto the saturation point. However, the only negative aspect of mesh topology is high latency of messages when communicating with faraway nodes. The future scope for this work can be extended by adopting more parameters in conjunction with the existing parameters along with different topologies.

Bibliography

- [1] Sriram Prakash Adiga. Noc characterization framework for design space exploration. *Coordinates*, 52:6–893662, 2014.
- [2] Ankur Agarwal and Ravi Shankar. A layered architecture for noc design methodology. In *IASTED PDCS*, pages 659–666, 2005.
- [3] Khalid M Al-Tawil, Mostafa Abd-El-Barr, and Farooq Ashraf. A survey and comparison of wormhole routing techniques in a mesh networks. *IEEE network*, 11(2):38–45, 1997.
- [4] Abdul Quaiyum Ansari, Mohammad Rashid Ansari, and Mohammad Ayoub Khan. Performance evaluation of various parameters of network-on-chip (noc) for different topologies. In *India Conference (INDICON), 2015 Annual IEEE*, pages 1–4. IEEE, 2015.
- [5] William J Dally. Virtual-channel flow control. *IEEE Transactions on Parallel and Distributed systems*, 3(2):194–205, 1992.
- [6] William James Dally and Brian Patrick Towles. *Principles and practices of interconnection networks*. Elsevier, 2004.
- [7] Jose Duato, Sudhakar Yalamanchili, and Lionel M Ni. *Interconnection networks: an engineering approach*. Morgan Kaufmann, 2003.
- [8] Manoj Singh Gaur, Vijay Laxmi, Mark Zwolinski, Manoj Kumar, Niyati Gupta, et al. Network-on-chip: Current issues and challenges. In *VLSI Design and Test (VDATE), 2015 19th International Symposium on*, pages 1–3. IEEE, 2015.
- [9] Nan Jiang, James Balfour, Daniel U Becker, Brian Towles, William J Dally, George Michelogiannakis, and John Kim. A detailed and flexible cycle-accurate

- network-on-chip simulator. In *Performance Analysis of Systems and Software (ISPASS), 2013 IEEE International Symposium on*, pages 86–96. IEEE, 2013.
- [10] Nan Jiang, George Micheliogiannakis, Daniel Becker, Brian Towles, and William J Dally. Booksim 2.0 user’s guide. *Stanford University*, 2010.
- [11] Zhonghai Lu. *Design and analysis of on-chip communication for network-on-chip platforms*. PhD thesis, Royal Institute of Technology, 2007.
- [12] Fernando Moraes, Ney Calazans, Aline Mello, Leandro Möller, and Luciano Ost. Hermes: an infrastructure for low area overhead packet-switching networks on chip. *INTEGRATION, the VLSI journal*, 38(1):69–93, 2004.