DESIGN AND EVALUATION OF AN ADAPTIVE NETWORK ON CHIP FOR MULTICORE ARCHITECTURES

A Dissertation

Submitted to the African University of Science and Technology

Abuja-Nigeria

in partial fulfilment of the requirement for the award of:

MASTER DEGREE IN COMPUTER SCIENCE

By

GURUMDIMMA, NENTAWE YUSUF

Reg. No: 40036

Supervisor:

Asst. Prof. Abderazek Ben Abdallah

December 2009

Abstract

Network – On – Chip (NoC) communication architecture have emerged as a solution to problem of lack of scalability, clock delay, lack of support for concurrent communication and power consumption exhibited by the *shared bus* communication approach to *System - On - Chip* (SoC) implementations. However, a NoC communication requirement such as bandwidth is affected by architecture parameters as topology, routing, buffer size etc.

In this project, we implement an adaptive approach of NoC to solve the problems of the static approach method such as routing delay, lack of flexibility and inability to predict dynamic behaviour of the applications. The adaptive approach supports several applications by changing parameters at run – time.

Dedication

To God and Saviour Jesus Christ who gave meaning to my life.

To my parents, for your fervent prayers for me always.

Acknowledgments

I would like to sincerely acknowledge my supervisor, Asst. Prof. Abderazek Ben Abdallah, whose meticulous supervision helped me in this work, Dr. E. Okorafor, for your sincere support, Dr. C. Boubou, for your fatherly advice and concern, Prof. C. Chidume, for your advice, the President and Management of AUST, for giving me the opportunity to study here.

I cannot forget you, Mercy Amadi (Emerald), for your support in ways you may not understand. I thank you all.

Table of Contents

Chapter 1 : Introduction

1.1 Introduction to System On Chip

Complex applications, using System On Chips (SoCs) can be implemented by integrating more cores since the number of cores increases rapidly. That is, the rapid development of cores technology allows complex circuits to be integrated into a single chip. This also means that the system's complexity also increases; hence designers tend to keep up with the increased complexity by using larger reusable blocks in their system design.

However, with these different processing elements used together to achieve powerful systems, connecting these cores together posses a great challenge. And as the number of these computational units/processing units increases and are integrated into one silicon chip, communication between them becomes a problem. A communication system that will support these cores must be designed.

Bus – based communication, where bus access request of nodes or cores are serialized through central arbiters, is a simple solution to the communication problem. However, this simple approach presents numerous challenges like *scalability* problem, bus capacitance increases dramatically with increase bus length and more additional cores, *performance* penalties, *inefficient power* or energy as the number of cores increases.

1.2 Emergence of Network On Chip (NOC)

NoC is a concept of communication in System on Chips (SoCs) [6,]. This concept claims to eliminate the problems of the Bus – based communication highlighted above. Unlike Bus – based communication where communication is done through buses and dedicated point-to-point links, Network On Chip, NoC, a more general scheme is adapted, employing a grid of routing nodes spread out across the chip, connected by communication links. The NoC design paradigm is centred communication rather than computation [4]. Each node (or tile) in the on-chip network is composed of a Processing Element (PE) and a communication unit which is so called

a Network Interface (NI) as shown in **Fig 1** below. The communication between the pairs of nodes is organized by connecting a network of routers and switching packages among them. The traditional bus-based on-chip communication architectures, the NoC solution provides higher communication scalability, flexibility, predictability, and power efficiency.

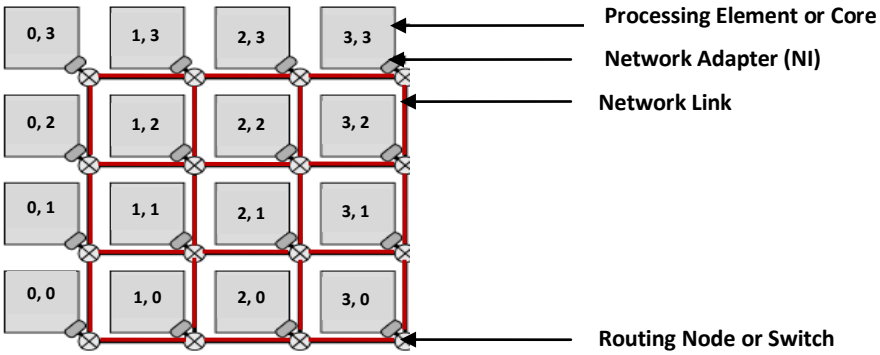


Fig 1: Regular structure of NoC

1.3 Related Work

Despite the fact that the concept of NoC is a relatively new field, it has in recent times, receive a lot of attention from research community. This is because of it has great potential to solving the On – Chip communication problems.

In [6], [19] different NoC topologies are proposed with regular mesh topology being the simplest to implement with most routing schemes.

There have been many routing algorithms which are based on wormhole routing proposed for meshes in the literature [7], [8], [10], [15], [16], all aimed at improving the performance of the routing strategies Network on Chip. These routing algorithms can be generally classified into three categories, depending on the degree of adaptiveness provided by the algorithms. A *non-adaptive* routing algorithm is deterministic and routes a packet from the source to the destination along a unique, predetermined path. A *minimal fully adaptive* routing algorithm routes all packets through any shortest paths to the destinations. A *partially adaptive* routing algorithm allows multiple choices for routing packets via shortest paths; in this case, it does not allow all packets to use any shortest paths.

1.4 Problems of NOC

NoC is an area receiving enough attention from researchers [5,6,7,9,10,12]. However, much work needs to be done in other to achieve the desired implementation of the NoC in practical. The following few problems are briefly discussed.

1.4.1 Topology

The underlying topology greatly affects the network's ability to efficiently disseminate information across the network. The routing strategies of the network depend deeply on the topology of the network [12]. Also, the network topology has great impact on the network latency, fault tolerance, throughput, area, power consumption and mapping the cores to the network nodes.

The grid -like structures are considered the simplest compared to the complexity of custom topologies. Hence mesh topology is considered more than others [12]. Flexibility on the other hand is provided more by customised topology and are most suitable for application – specific NoCs [13]. Regular topologies waste area when the sizes/shapes of the cores vary [13]. Generally, designing the network to meet the requirements of highly communicating cores will result in under-utilization of other components and for average case may result in performance bottleneck [14].

1.4.2 Buffer Size

In NoC, the buffer size of an input channel of the router or switch affects the overall area of the NoC. This implies that there is need for overall reduction in the buffer use to minimize NoC implementation overhead. On the order hand, increasing the buffer size can reduce the network latency, most especially, when the traffic pattern cannot be predicted and changes.

1.4.3 Channel Width

The width of the network channels affects the bandwidth of the NoC's network channel. Increased width reduces latency; however, area increases [14]. NoC's main problem here is how to obtain minimum network latency and maximum network throughput subject to area and power consumption constraints.

1.4.4 Routing

One of the major problems of Network on Chip (NoC) is the routing of packets among the nodes. Deciding which output port to send the next packet as it traverses from source to destination is the key problem here. It has receive much attention by researchers, however with little practical implementation. Routing greatly affects the network performance and power consumption. Complicated routing strategies result in larger design resulting area and performance trade-offs [15 - 16].

Two routing strategies are proposed, d*eterministic* and *adaptive.* Performance requirement and implementation complexity are major consideration when selecting a routing strategy. *Deterministic* routing, though, requires less resources, and guarantee packet arrival in order, it cannot respond to dynamic network conditions like congestion. The *adaptive* provides better throughput and lower latency since alternate path can be taken in case of traffic congestion [16]. Every routing scheme (Deterministic or Adaptive) has a primary goal of ensuring that every packet injected into the network will arrive its destination eventually. In NoC routing, packets may be involved in

- **Livelock:** Packets makes an infinite link traversal without reaching its destination as the time tends to infinity.
- **Deadlock:** This means that at a particular time, packets cannot advance or move further regardless of any policy applied throughout the network. There exists a circular dependency between different packets, in such a way that each packet is holding on to its own resources, on the other hand, it is trying to reserve resources which are being held by other packets

1.5 Thesis Contribution

This work focuses on the routing scheme implementation and the communication infrastructure of the OASIS Network on Chip. The contribution can be seen in two folds as follows: a). Routing algorithm Implementation in OASIS NoC. b). OASIS Run time Monitoring System.

a. Routing Algorithm Implementation in OASIS NoC.

Existing OASIS NoC uses deterministic routing strategy to route packets from one node to another. That is, the X – Y coordinate static offline movement with no consideration to the network traffic. We implement adaptive routing scheme using the Odd/Even turn model. The scheme uses the input from the monitoring system about the traffic of the neighbouring nodes to adaptively route a packets.

b. OASIS Run – Time Monitoring System

We propose a run – time monitoring system that monitors the traffic of the whole network. Each router's buffer size is monitored to provide the routers with

- Traffic information of its neighbours
- In case of high or low traffic in a particular switch, the monitoring system dynamically changes the buffer parameters to accommodate the traffic condition and reduce the overhead cause by increase or reduced buffer size.

1.6 Report Organization

This work will give an introduction to Network on Chip Design in Chapter 2. In this chapter, we will look at the system design problems, the interconnection that exist in the NoC, looking briefly at the bus – based and NoC based interconnection. Chapter 3 will introduce OASIS NoC interconnection where the OASIS architecture is discussed. Looking at its Switch or router architecture, the routing that existed with it flow control. Chapter 4 will discuss the OASIS NoC with run – time monitoring system, the algorithm and the architecture of the system. In Chapter 5, we will look at some hardware evaluation results like the power, speed etc of the OASIS. The chapter will also contain software evaluation of the system with some test bench to find the latency of the system. Chapter 6 will conclude the work with the necessary recommendations.

Chapter 2 : Network On Chip

2.1 Introduction

Deep sub-micron processing has enable application specific implementation of embedded architectures [3]. These architectures integrate software programmable processors and other dedicated hardware components together in a single chip, and the technology came due to the emergence of wireless communication, multimedia Computing, distributed computing, distributed networking/broadband [3]. Designers of these systems are currently confronted with the enormously difficult task of designing these complex heterogeneous multi-core architectures despite the new ways with a high productivity gap predicted [3]. It is however, predicted by the International Technology Roadmap for Semiconductors (ITRS) predicted that ICs will have billions of transistors [3], [6]. Designers are now faced with great challenges of designing powerful systems to keep pace.

Network on chip (NoC) is a new approach to designing the communication subsystem of system on chip. This emerging paradigm of communication within VLSI systems is implemented on a single silicon chip [1]. NoC Modules like processor cores, memories, IP blocks exchange data using a network subsystem for the information traffic. It is constructed from multiple point – to – point data lines interconnected by (routers) switches .Hence messages can be moved from any source to destination over several links by routing decisions at the switches.

The emergence of Network - On – Chip is driven by semiconductor technology advances. Designers can integrate many IP blocks together with large amounts of embedded memory because of the availability of huge transistors on a single chip [5].

Network – On – Chip paradigm provides the property that will close the productivity gap. In NOC each core is connected to a switch/router by a network interface, and communicates with each other by sending packets through a network path of switches and inter-switch links (topology) [7].

2.2 On – Chip System Interconnection Overview

The NoC interconnection paradigm is also characterized by its topology, protocol, and flow control, just as the normal interconnection network in conventional parallel computers. The layered approaches to interconnection is used for on – chip systems which can described protocol functions operating on data units at different abstraction levels.

The NOC architecture and interconnection provides the communication infrastructure for the resources. This is done either by connecting independent stand alone blocks by connecting the blocks as node elements in the network or by scalability and reconfigurability reconstructing the network to maintain workloads [9].

Decisions should be made on the communication protocol, switching style, network topology, clock synchronization method, signalling scheme, etc. when designing such systems. Fig 2b, below shows typical components of a NoC system.

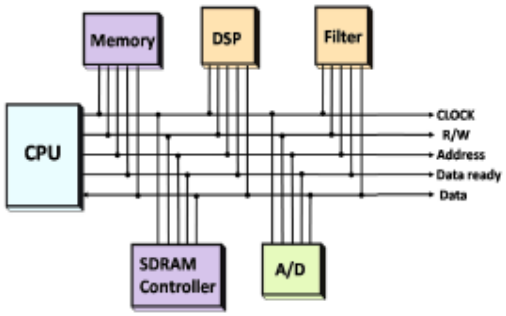


Fig2a. Bus- base communication

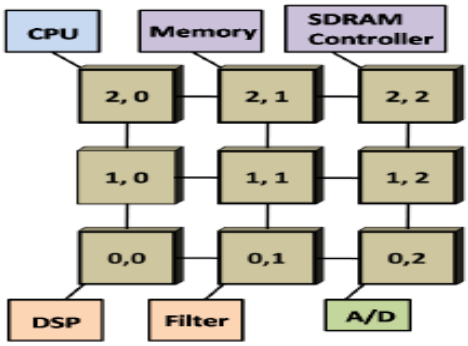


Fig 2b: NoC – Base Architecture.

2.2.1 Bus – Based System

Bus – based communication **(Fig 2a.)** posses the following challenges:

- They are unstructured and have parasitic capacitance and crosstalk to adjacent wires: This makes them difficult to predict early in the design process, also differ from one run of router to next, hence causes delay and high power

dissipation[3].

- Long wires require repeaters at periodic intervals to keep their delay linear, and posses additional constraint.
- The performance of the bus – base is inherently not scalable since there can be only one transaction/ communication at a time over the shared bus. Therefore, performance degrades whenever a slow device accesses the network.
- In bus – base, timing in deep sum-micron is extremely difficult.

Hence, it is clear that from interconnection links level, wires are becoming increasingly the bottleneck, making transistors play second role, As such, performance figures, power consumption and area utilization is dominated by wires[3].

The bus – base advantages:

- The latency here is wire – speed once control has been granted by arbiter.
- All buses are almost directly compatible with most IPs.
- Bus – base concept is simple and clear, not complicated as Network base.

2.2.2 The NOC – Based

Advantages

- **Re usability**:

  According to [5], 1 billion transistors will be on a single chip soon. But the compiler technology has not grown to meet up with the growing IC technology, thereby creating a gap[.3], and reuse is primarily use to bridge the gap. NOC provides a good platform for the reuse since it has the potential in its communication network, due to the fact that its switches or routers, communication protocols, interconnection can be designed and optimised to be reused in a number of resources. Also, the same router may be re instantiated, for all network sizes [6].

- **Predictability:**

  Most regular NoC topologies provide structural and predictability advantage, which provides well controlled and optimised electrical parameters [5]. This brings a significant reduction in the power dissipation and propagation delay.

- **Scalability:**

  As mentioned earlier, soon billions of transistors will be integrated in a single chip, this means improved technology and performance of processing elements. NOC provides the scalability needed to contend with the increasing technology [3]. Most especially the Network Interface with gives room for addition of many IP cores on the NOC.

  Only point-to-point one-way wires are use for all network sizes, thus local performance is not degraded when scaling [6].

- One of the advantages of the network base is that routing decisions are distributed, if the network protocol is made non-central.

Disadvantages

- Latency can be caused by internal network contention.
- It is not as easy as bus re-education is needed for system designers for new concepts.
- Not all buses are compatible with IPs, hence software synchronization is needed.

2.2.3 NOC Designs Issues

2.2.3.1 Topology

The topology of Network -on -Chip NoC, specifies the physical organization of the interconnection network. It simply defines how the nodes, switches and links are connected. Routing algorithms are to some extent, dependent on the topology of the network.

Network – on – Chip, (NoC) Topologies can be classified as *Regular* and *Irregular*.

Regular: Communication between the nodes in the network defines different regular NoC topologies. These topologies include Mesh, Torus, Butterfly, etc as shown in the diagrams below. According to [6], regular form of NoC topology scale predictably with regard to power and area.

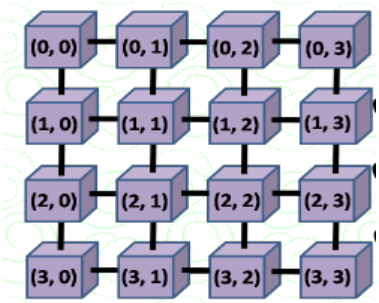


Fig 3a: Mesh topology

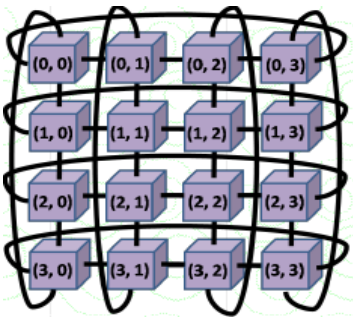


Fig 3b: Torus topology

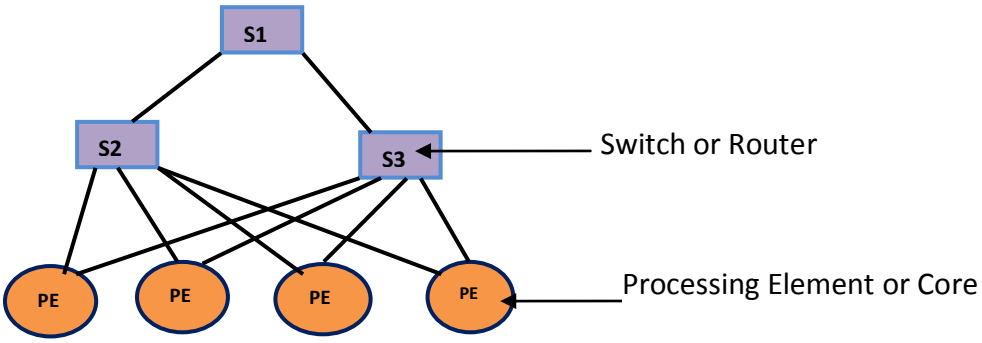


Fig 3c: Fat -tree network.

- **Mesh**. From **Fig 3a** above, it consists of *m* columns and *n* rows, with routers situated in the intersections of two wires and the computational resources are near routers. The routers and resources can be easily defined as x-y coordinates in mesh.
- **Torus**. Torus network is an improved mesh in which the heads of the columns are connected to the tails of the columns and the left sides of the rows are connected to the right sides of the rows as shown above in **Fig 3b**. Torus network is believed to

have better path diversity than mesh network, and it also has more minimal routes.

- **Tree**. **Fig 3c** above, tree topology has nodes as routers and leaves are computational resources. The routers above a leaf are called as leaf's ancestors and the leafs below the ancestor are its children, just as a tree structure. A fat tree topology provides many alternative routes between nodes since each node has replicated ancestors.
- **Butterfly.** It is uni- or bidirectional and butterfly-shaped network typically uses a deterministic routing [8]. As shown in **Fig 3d** below.

Irregular: Irregular forms of topologies are derived by altering the connectivity of a regular structure. For example, in mesh, certain links are removed or by mixing different topologies such as in torus where a ring coexists with a mesh.

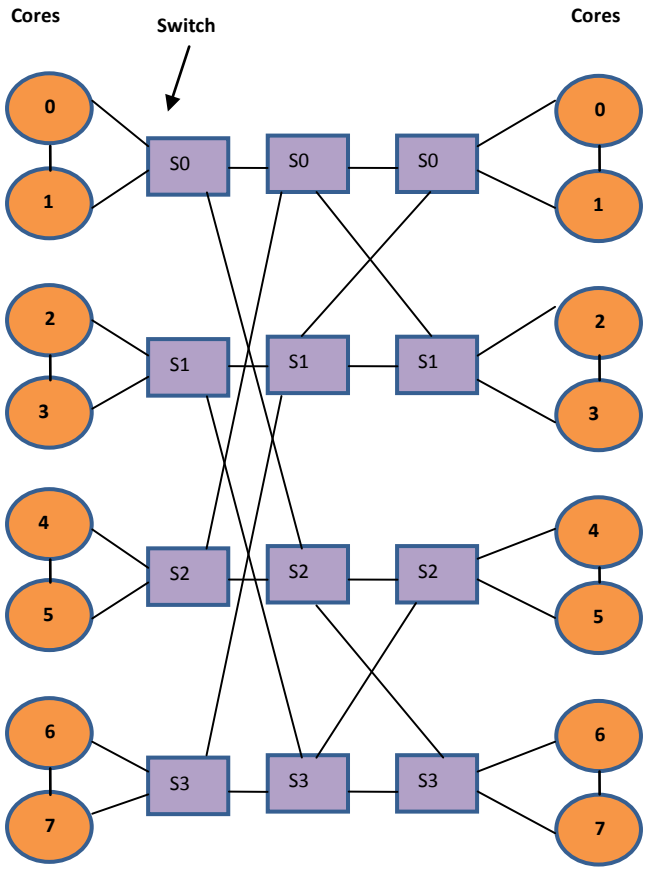


Fig 3d: Irregular butterfly connectivity

2.2.3.2 Protocol

The protocol determines the movement of packets within the NoC's nodes. There are quite a number of the protocol spaces used in NoC.

- **Circuit switching:** A physical channel is established between the sending node and the receiving node during the circuit establishment phase that is prior to the transmission of data [3], [5]. At the transmission phase, packets are not stored in buffer since buffer is not required here; hence, the transmission of data is not interrupted by transmission of other packets. Once the transmission is completed, the channel of transmission established between the source and destination nodes is broken, maybe by sending a special packet [5].

Advantages of circuit switching scheme is that, the network bandwidth is statically reserved for the data transmission duration. Also, area and power consumption is greatly reduced since the scheme does not need packet buffer.

One of the overhead of the scheme is that it's not flexible since setting up of an end – to – end path during transmission causes delay. Hence the scheme has high performance with little flexibility [3].

- **Packet Switching:** In this scheme, data is divided into packets of fixed length that contains enough information about its source and destination, containing control part, header and the data part or payload. A packet is sent by the source, it is buffered at the receiving node and the receiving node inspects the header information to route or switch the packet to the right destination whenever the channel is free and the destination buffer is free enough through the right output port. One of the drawbacks of this scheme is the need to sort all the packets in a switch which increase the need of buffer.

- **Wormhole Packet Switching:** In this scheme, a packet is further divided

into flow control units (flits) as the basic units [10]. These flits are then routed through the network one after another, in a pipelined fashion. The input and output buffers are expected to store few flits [10]. Hence, the buffer requirement for this scheme can be small as compared to packet switching earlier seen. The header flit reserve the routing channel of each switch which is followed by other flits (body flits) and the tail flits then releases the channel reservation. This scheme requires less buffer since it does not require the complete packet to be stored on the switch buffer while it waits for the header flit to be routed. That is to say a packet can occupy different switches at the same time. This scheme increases the possibility of deadlock since message can occupy channels and buffers. In addition, channel utilization is somehow decreased if a flit from a given packet is blocked in a buffer [3]. The deadlock problem is tackled in wormhole switching by using virtual channel solution [11]. The physical channels each, is associated with several FIFO buffer **Fig 4**, providing the flexibility for other flits to use other virtual channel buffers.

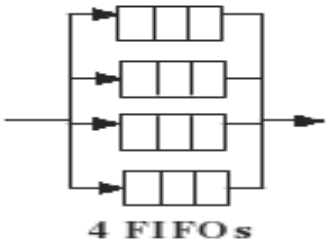


Fig 4: Buffer with 4 virtual channels.

2.2.3.3 Flow Control

Flow control deals with allocation of channels bandwidth and buffers to packet as it traverses the network to its destination. It also involves resolving packets collision. Flow control strategy determines when a packet is dropped, buffered or misrouted. A good flow control policy should avoid channel congestion while reducing the network latency [10].

2.2.3.4 Routing

Routing in NoC, determines the path that each packet follows between a source–destination node pair. Routing can be classified as ***deterministic*** or ***adaptive*** [10]. Routing is deterministic if only one path is provided per node pair, that is, the path followed by a particular packet absolutely depends on the packet's source and destination addresses. On the other hand, adaptive routing provides several paths for packets to follow the path taken by a particular packet depends not only on the source and destination addresses, but also on the network traffic, such as buffer occupancy level of the node, congestion of the network etc. Adaptive routing better balances network traffic by dynamically taking alternate routes in case of congestion, thus allowing the network to obtain a higher throughput. Detailed explanation of the routing is done chapter three.

Chapter 3 : OASIS Interconnection Network

3.1 Introduction

It was seen in Chapter 1 and 2, that Network-on-Chip NoC is becoming an attractive option for solving bus based problems, as it is a scalable architectural platform with huge potential to handle growing complexity and can provide easy reconfigurability [1 -2]. The basic idea is that processing elements are connected via a packet switched communication network on a single chip just as the way computers are connected to Internet. Therefore, in an On Chip based on NoC, the interconnection is made up of several network clients (e.g. processors, memories, and custom logic) which are connected to a network that routes packets between them.

Generally, On chip interconnection networks use layered approaches, which are well suited to describe protocol functions that operate on data units at different levels of abstraction (in the form of streams, packets, bits or analog waveforms) and that are subject to various time granularity constraints [17]. Each layer may include one or more closely related protocol functions, such as data fragmentation, encoding and synchronization. Similar to interconnection networks for conventional parallel computers, the NoC interconnection paradigm is also characterized by its topology, protocol, and flow control [3].

OASIS NoC is an architectural design of an On – Chip network. OASIS Network on Chip system was described in Verilog HDL with CAD synthesis.

3.2 OASIS NoC Architecture

The OASIS is a 4 X 4 Mesh network that adopts wormhole switching. The OASIS Mesh topology is a simple topology where nodes are connected to each other directly (Fig 5). Because of the simplicity, it is easy to route a path. Each switch has XY coordinate called address.

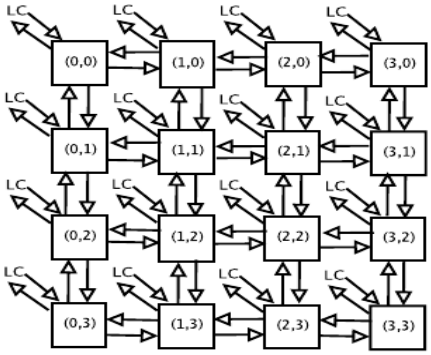


Fig 5: OASIS Mesh interconnection showing switches and ports connections

3.2.1 Switching

OASIS NoC architecture, the on-chip cores interconnection is arranged as a mesh of switches. Each switch has five inputs and five outputs (see Figure 5). One input/output pair is for communication with the resource. The remaining four pairs are connected to the surrounding switches. The size of one flit is 76 bits (Fig 6), and it has information of destination address, next port direction information and payload. Flits are transmitted one by one as shown in Fig.4

OASIS employs wormhole routing, one flit size is 76 bit which contains information of its destination address, next port direction information and payload flit are transmitted one by one, and are buffered in input port FIFO. Each input port has FIFO which is 76 bits and depth is 4. The FIFO's Depth concerns area of switch but performance may increase.

The switch architecture has great impact on the costs and on the performance of the whole network. Each switch has a unique address in the network. To simplify routing on the network this address is expressed in XY coordinates, where X represents the horizontal position and Y the vertical position. OASIS architecture uses wormhole packet switching, and messages are sent by means of packets (several flits). Therefore, the switching has low latency, saves memory buffers and, with appropriate routing algorithm, communication deadlock can be avoided. A flit (flow control unit) is the smallest unit over which the flow control is performed [3].

3.2.1.1 The Flit Structure:

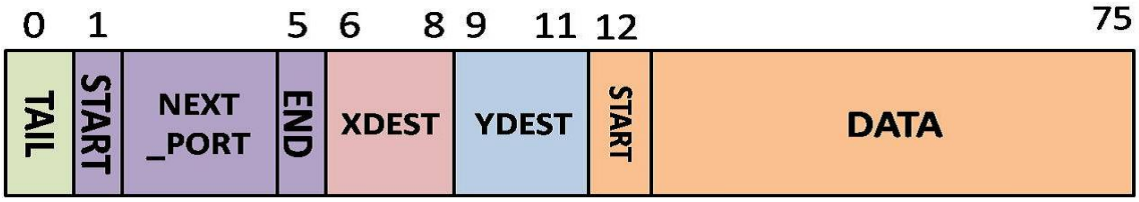


Fig 6: OASIS Flit structure.

Fig 6 above shows the structure of the flit. Each flit contains enough routing information. The flit contains: **DATA** (64 bits) is payload, this contains the information of the flit that needs to reach the destination, the start bit reserve the channel for the packet until the TAIL flit is passed which releases the channel. The **TAIL** (1 bit) actually means last flit of a packet**, NEXTPORT** (5 bits) is direction of output; it contains the next port to be taken by the flit (That is, NORTH, SOUTH, EAST, WEST, LOCAL). **XDEST, YDEST** (3 bits) is destination of x and y-addresses respectively. This two pair (X, Y) gives the address of the next node the flit is routed to.

Too many small flits increase overhead of reassembling reduce performance of the whole system. Hence correct packet size is important for optimum use of network.

3.2.1.2 Switch Architecture:

The OASIS switch hardware consists of ***crossbar*** circuit, 5 buffered ***input ports*** (see Fig.7 & 8a) where all the control flow and routing functions are performed. The **crossbar** (see Fig 9) allows 5 different data from the 5 input ports to be routed to the next output port according to its destination and Next port. The crossbar module synthesized and analysed in Quartus II, the RTL schematic diagram is as shown in **Fig 12**. The **arbiter** or **Switch allocator** (Fig 10) use a simple Round-Robin scheduling scheme, with high priority. The arbiter's main task is to grant data paths by evaluating how many available positions the FIFO has and the data size of all requesting FIFOs. That is, at arbiter, the Next port information transmitted from input port to switch

allocator is decoded to decide packet of the output direction. The switch data path architecture is shown in Fig 6 below.

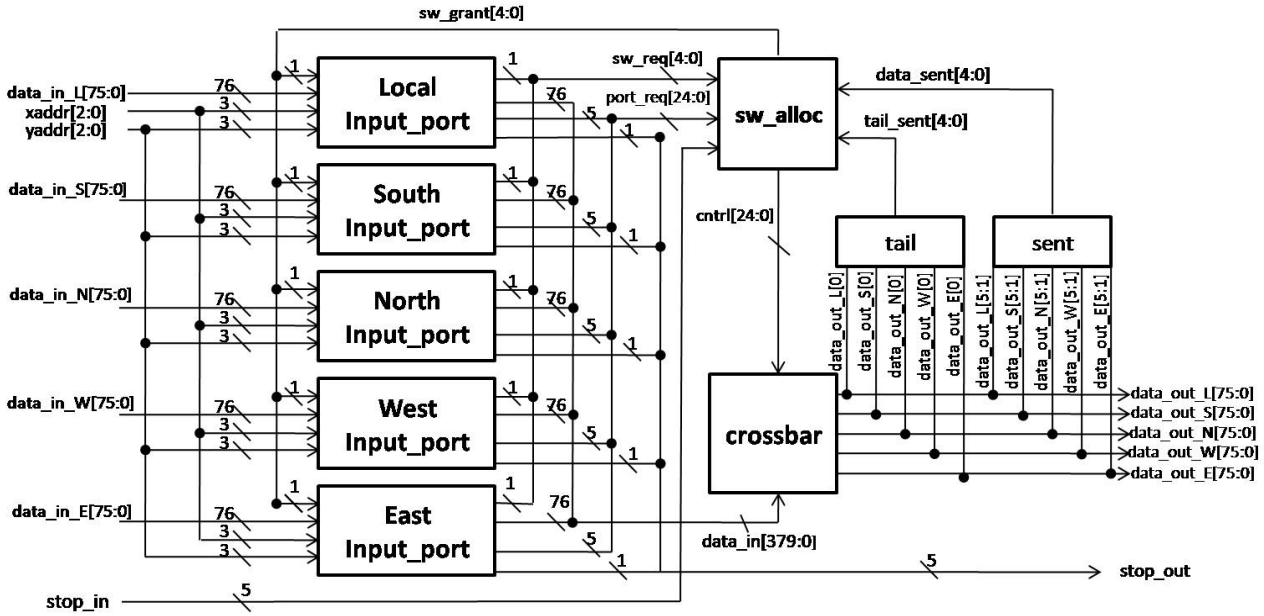


Fig 7: One Switch Data Path: ***L, S, N, W and E indicate direction of port, Local, South, North, West and East.***

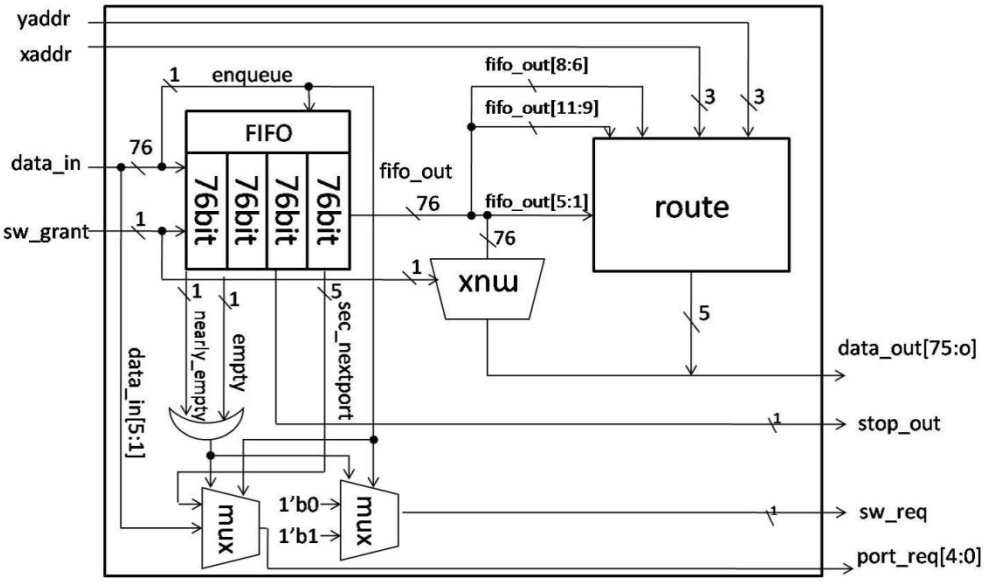


Fig 8a: Input Port Circuit.

Fig 7b: Buffer

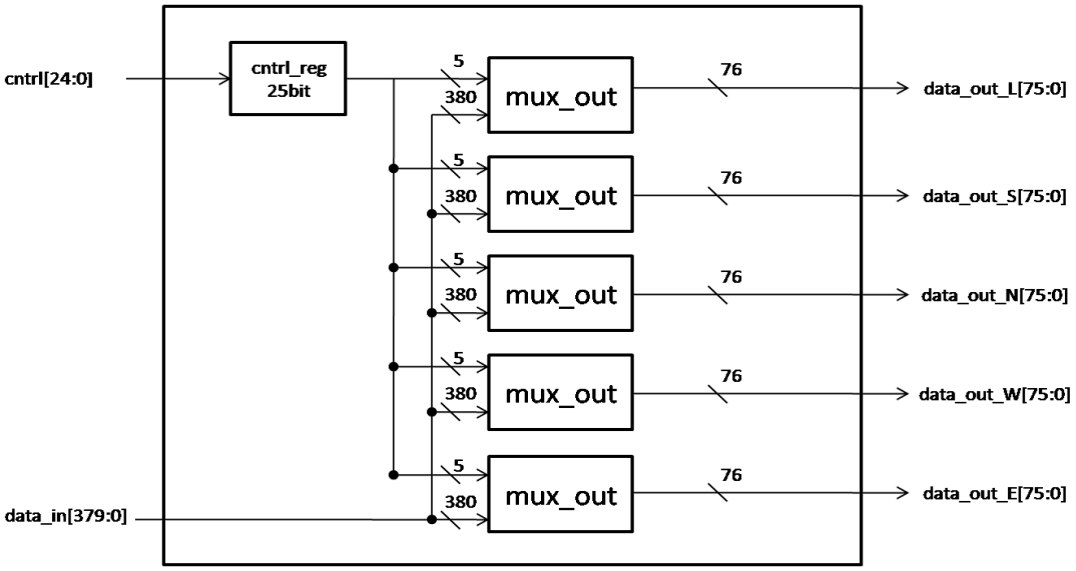


Fig 9: Crossbar Circuit.

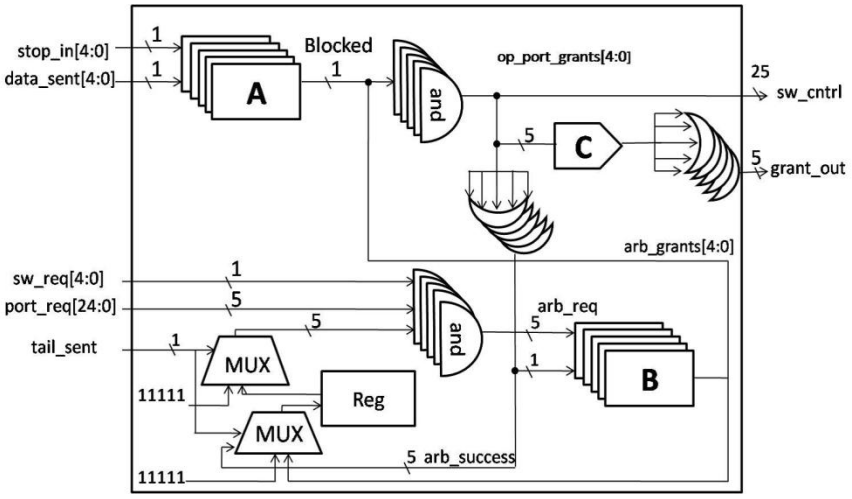


Fig 10: Switch Allocator/Arbiter Circuit

3.2.1.3 Switch Buffer:

The buffer of the switch is a module used for storing several flits which of course cannot store all flits hence buffer overflow occurs. From **Fig.8b,** the buffer (FIFO) is found in each of the five input ports and each can hold at most four (4) flits. The possibility of dropped flits increases since the buffer size is small when there is high traffic. But large buffers consume extra power and area.

3.2.1.4 Switch Arbitration

One of the switching problems of NoC is transmission congestion. When More than one port tries to send packet through the same port at the same time, congestion occur. OASIS NoC uses a simple efficient scheduling scheme called **Round – Robin Arbitration mechanism** to avoid transmission congestion. Supposing North, West and South ports tries to send values at the same time to East port, the arbiter uses the round – robin to schedule the transmission as follows:

North port send first while South and West waits, then followed by South, while North and West waits, then West. With this mechanism, transmission congestion is avoided. Fig 10 above shows **arbiter circuit** diagram.

3.2.2 Routing

OASIS NoC uses deterministic routing where the path that a particular packet follow is completely determined by the source and the destination addresses of the packet.

The next port direction in next address is decided in input port. To compare next address and destination address, next port in next address is decided. If Y of destination address is larger than Y of next address, next port is SOUTH in a contrasting situation, next port is NORTH. If X of destination address is larger than X of next address, next port is EAST in a contrasting situation, next port is WEST. When next address and destination address is equal, next port is SELF. The above algorithm is simplified ad below:

Ydes, Xdes: Y coordinates destination and X coordinate addresses respectively,

Ynextadd, Xnextadd: next nodes y and x – addresses respectively,

nextadd, desadd: next address pair (x,y) and destination address pair (x,y) respectively.

```
IF Ydes  > Ynextadd THEN
                NextPort = NORTH
ELSE
IF Ydes < Ynextadd   THEN
                NextPort = SOUTH
ELSE
IF Xdes  > Xnextadd  THEN
                NextPort = EAST
ELSE
IF Xdes  < Xnextadd  THEN
                NextPort = WEST
ELSE
IF nextadd   = desadd   THEN
                NextPort = LOCAL
ENDIFs
```

This information **(*NextPort*)** is sent into switch allocator. Direction of flit in input port whose output port and when it should be output is decided in switch allocator. If some flits in another input port request same output port, transmitting may be block. To solve this problem, function of arbiter is used. **Arbiter** determines high priority flit in input port, OASIS arbiter schedules in round-robin scheme, and this scheme serves each input port in fair. This module sort which data send into next port, SOUTH, NORTH, WEST, EAST and LOCAL by using control signal from switch allocator. **Fig. 10** indicates switch allocator/arbiter.

Flowchart of the algorithm

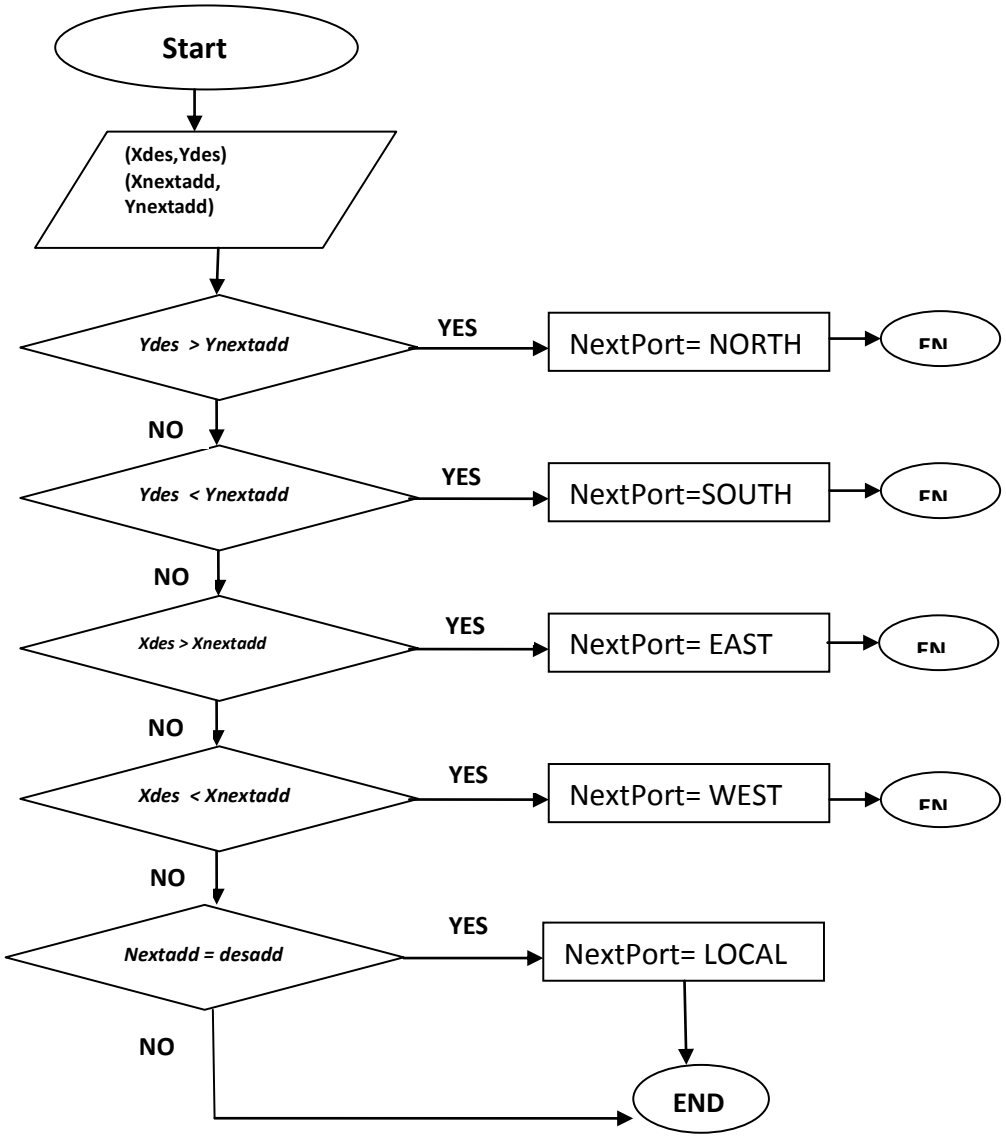


Fig 11: The algorithm Flowchart (X – Y deterministic routing)

The algorithm is implemented in a module called the ***route*** module and described in Verilog HDL. After synthesis and analysis of the algorithm in Quartus II, the schematic design netlist is as shown in **Fig 12** below.

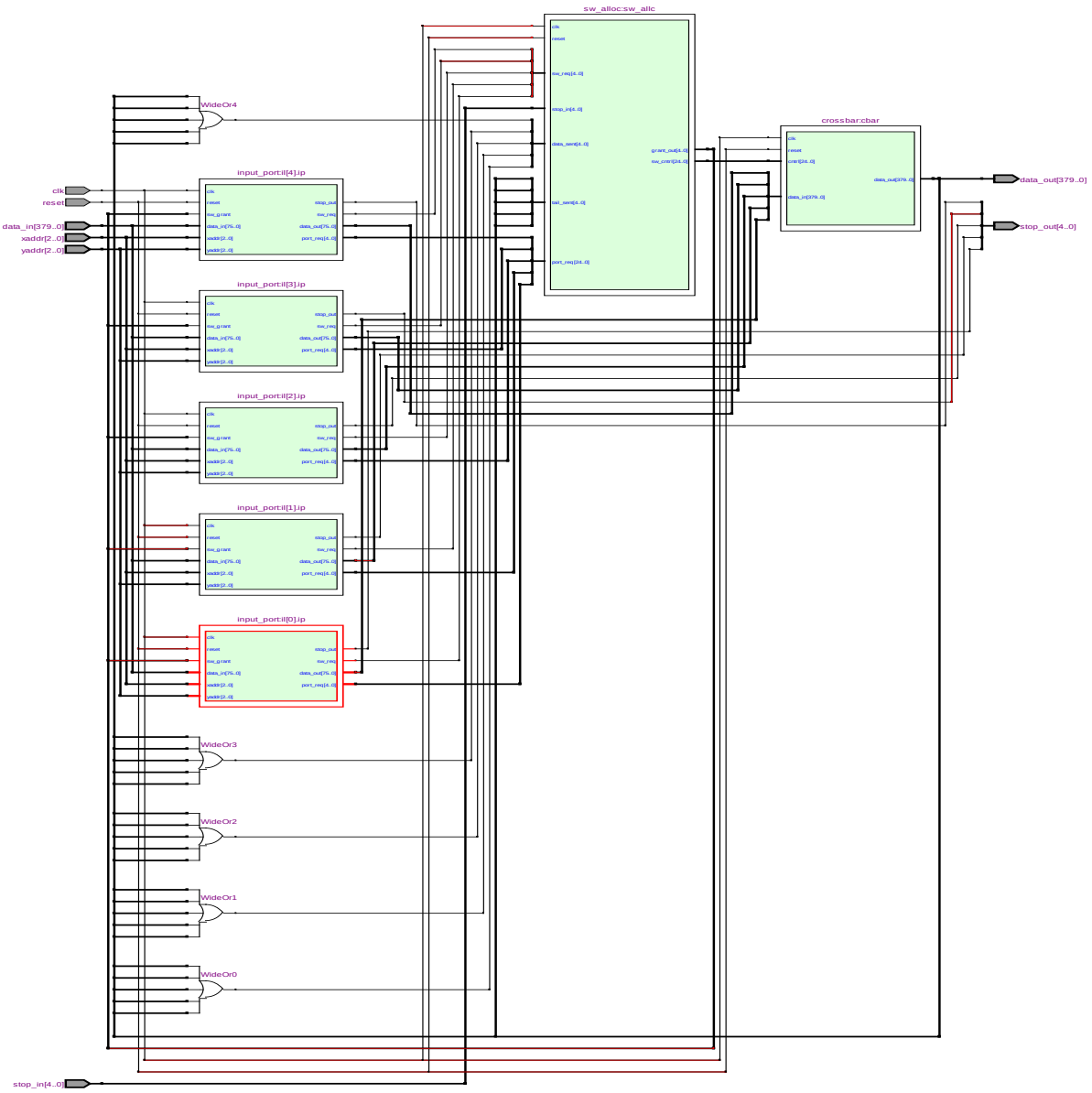


Fig 12: RTL Netlist Schematic diagram of router

The RLT diagrams in Fig 12 and Fig 13, displays the schematic view of the design after Analysis and Elaboration. The RTL views most closely represent the original source design, and showing how the Quartus II interpreted the design files. The RTLNetlist schematic diagram above shows router with its input ports modules, switch allocator, and crossbar.

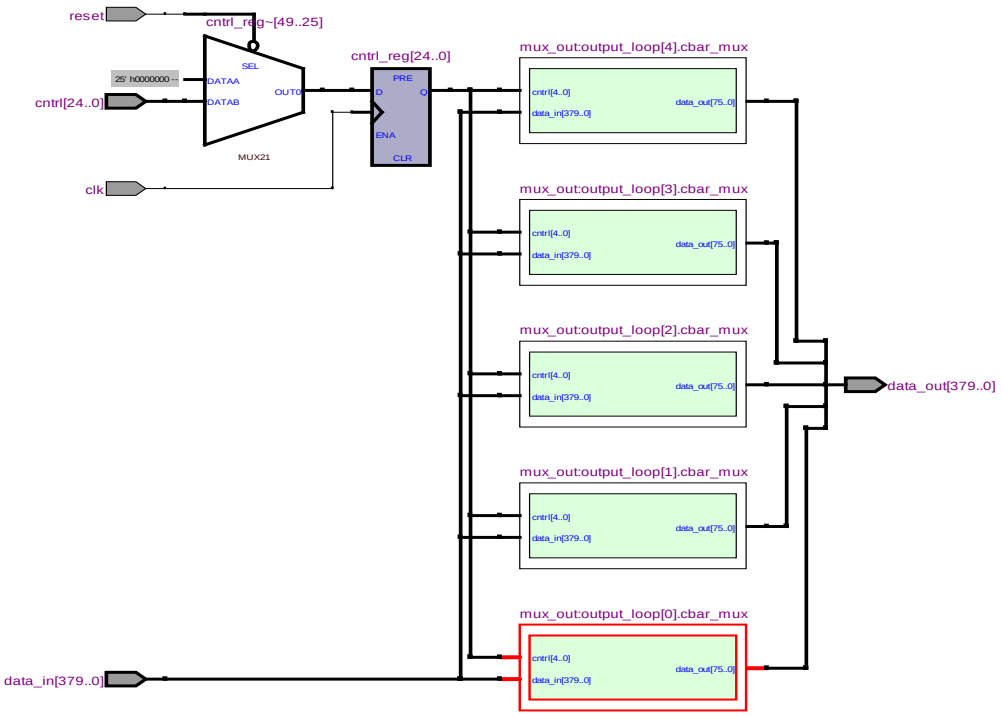


Fig 13: RTL Schematic diagram of Crossbar

3.2.3 Flow Control

Flow control determines how packets traverse the network path determined by the routing algorithm. It allocates network resources in order to ensure proper transmitting and receiving of a packet. Thus, allocation of network resources can be optimized to enhance an interconnection network's performance. OASIS NoC uses a flow control scheme called **Stall – Go.** The number of flits which are stored into FIFO is limited. Input flits will be overflow if FIFO is full; hence queue overflow technique is needed. We use a scheme Stall-Go mechanism. The OASIS Stall-Go mechanism is needed to

prevent FIFO from being full. Therefore, arbiter must determine the output flits, where next switch is nearly full or not nearly full. The output signal is controlled by the switch allocator. Fig. 14 shows the Stall-Go mechanism.

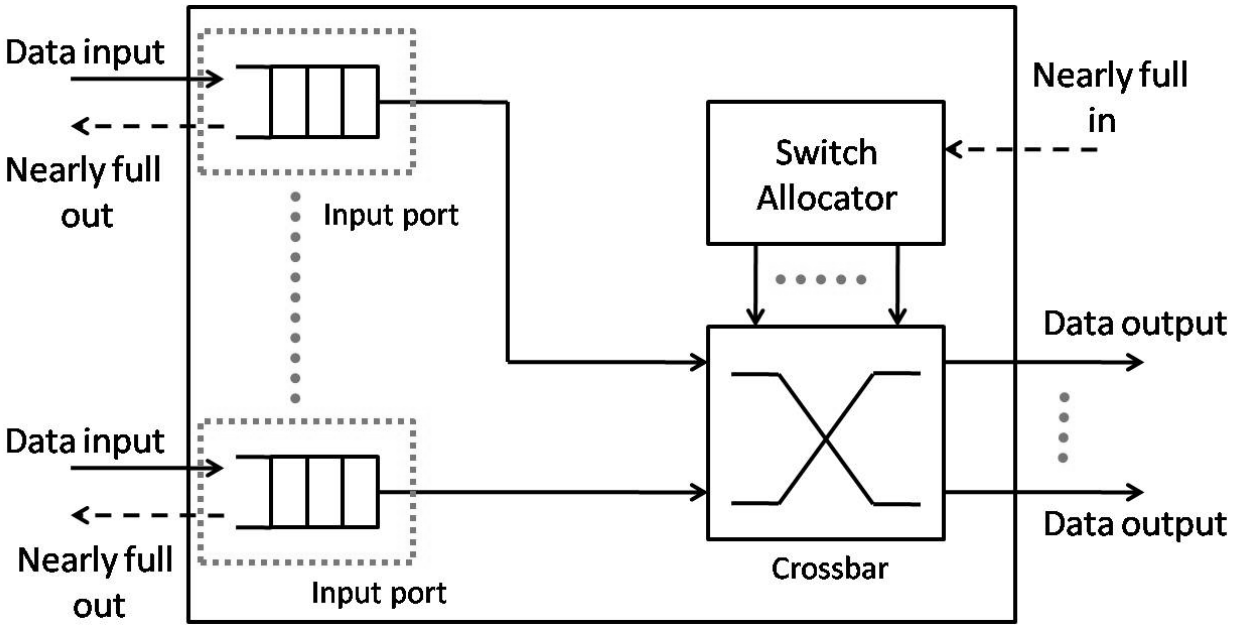


Fig 10: Stall – Go mechanism

Chapter 4 : OASIS With Run Time Monitoring System

4.1 Introduction

We saw earlier that OASIS route packets from one node to the other deterministically using the X – Y coordinate routing. This routing scheme is simple and cheap to implement. However, this simplicity and cheapness comes with a price. The routing scheme is unable to balance the load and to make use of alternative, potentially preferable paths or routes from their source to destination. This wastes available communication resources and causes the performance of a static routed network to degrade in face of unfavourable traffic pattern. In this adaptive Network on Chip (ANOC) with run – time monitoring system, an adaptive routing scheme is implemented. This scheme will dynamically route packets to destination through available routes which are determined at run – time base on the traffic condition of the network. A packet can be routed to an alternatively less congested node as it traverses from source to destination.

The packets' flits are buffered in an input ports' buffer (FIFO) before routing it to the destination. Buffer then plays an important role in keeping the flits from dropping. However the lager the size of the buffer the greater the area, though performance is increased. In this implementation of OASIS with run – time monitoring system (ANOC), the monitoring system try to analyse the network traffic and the buffer occupancy of each switch. It then dynamically change (increase or decrease) the buffer parameter. It increases the buffer parameter of a switch if the switch has high traffic and decreases the buffer parameters otherwise.

4.2 Algorithm

4.2.1 Routing

The Odd/Even turn model algorithm is a suitable algorithm for dynamically gets available routes that leads to a node. In this model, a set of possible route are provided.

In wormhole routing, deadlock is caused by packets that wait on each other in a circle. Considering a Mesh topology OASIS, Odd – Even routing algorithm restricts the location or nodes at which certain turns can be taken so that a circular wait can never occur. Hence, it does not eliminate any turns as messages are routed.

The basic idea of the odd – even turn model routing algorithm is to restrict some of the turn locations where some turns occur so that an East – North and North – West turns are taken at nodes in the same column and also neither are East – South and South – West turns.

These turns are governed by the following rules [20]:

Rule 1: *Any packet is not allowed to take an **East – North** turn at any node located in an even column, and it is not allowed to take a **North – West** turn at any node located in an odd column.*

Rule 2: *Any packet is not allowed to take an **East – South** turn at any node located in an even column and it is not allowed to take **South – West** turn at any node located in an odd column.*

a. The algorithm

Inputs: (Xs Ys) – Node Source address in the form of x – y coordinates.

(Xc Yc) – Current Node Address

(Xd Yd) – Destination Node Address

Variables

F1 = Xd – Xc : This tries to know where the packet is in X – direction

F2 = Yd – Yc : Tries to know the position of the packet in Y – direction

If (F1 == 0 && F2 == 0)

If this is true, then the packet has reached its destination node. Hence the next Port to route the packet,

nextPort = LOCAL.

IF (F1 == 0): *Means the packet has reached destination X – coordinate.*

If (F2 >0): *That is, the packet is at destination X – coordinate, but has not yet reach the Y – coordinate*. Available_route:= **NORTH**,

ELSE

Available_route:= **SOUTH**

ELSE

IF (F1 > 0) : *Means the packets destination is towards the **East** in X – coordinate and has not reach.*

{

IF (F2 == 0) : *Means packet has reach destination Y – coordinate .*

Available_route := **EAST**

ELSE

IF (Xc is odd OR Xc==Xs) if the packet is on odd axis of the mesh topology OR current X – address is equal to source X – address, then

IF (F2 >0): *Packet has not reached the Y – coordinate,*

Available_route:= = **NORTH**

ELSE : *destination Y- addr is smaller than current, hence*

Available_route:= **SOUTH**

IF (Xd is odd OR F1 ≠ 1) : *if the destination X- address is on odd column of the mesh topology OR the packet is not on the X-coordinate before the destination node's X – coordinate, then the possible*

Available route = **EAST**.

ELSE

} Do nothing

ELSE : West bound packets

Available route = **WEST;**

IF (Xc is EVEN) : *The current X – address is Even*

{

IF (F2 >0) : *packet has not reach the Y- coordinate of the destination address, destination addr > current addr*

Available route = NORTH

ELSE: *Destination Y address is less than current Y - address*

Available route = **SOUTH**

}

// Select Route with smallest traffic

For each Available_route

{

get_traffic(Avail_route)

}

NextPort = get_smallest_traffic(Avail_routes)

end for

End

b. The Flowchart

The flowchart below for the algorithm gets each available route and finds the traffic of the route. Compare them and select the one with least traffic and assign it to nextPort.

Variables:

Traffic_In_N = Traffic from the north node

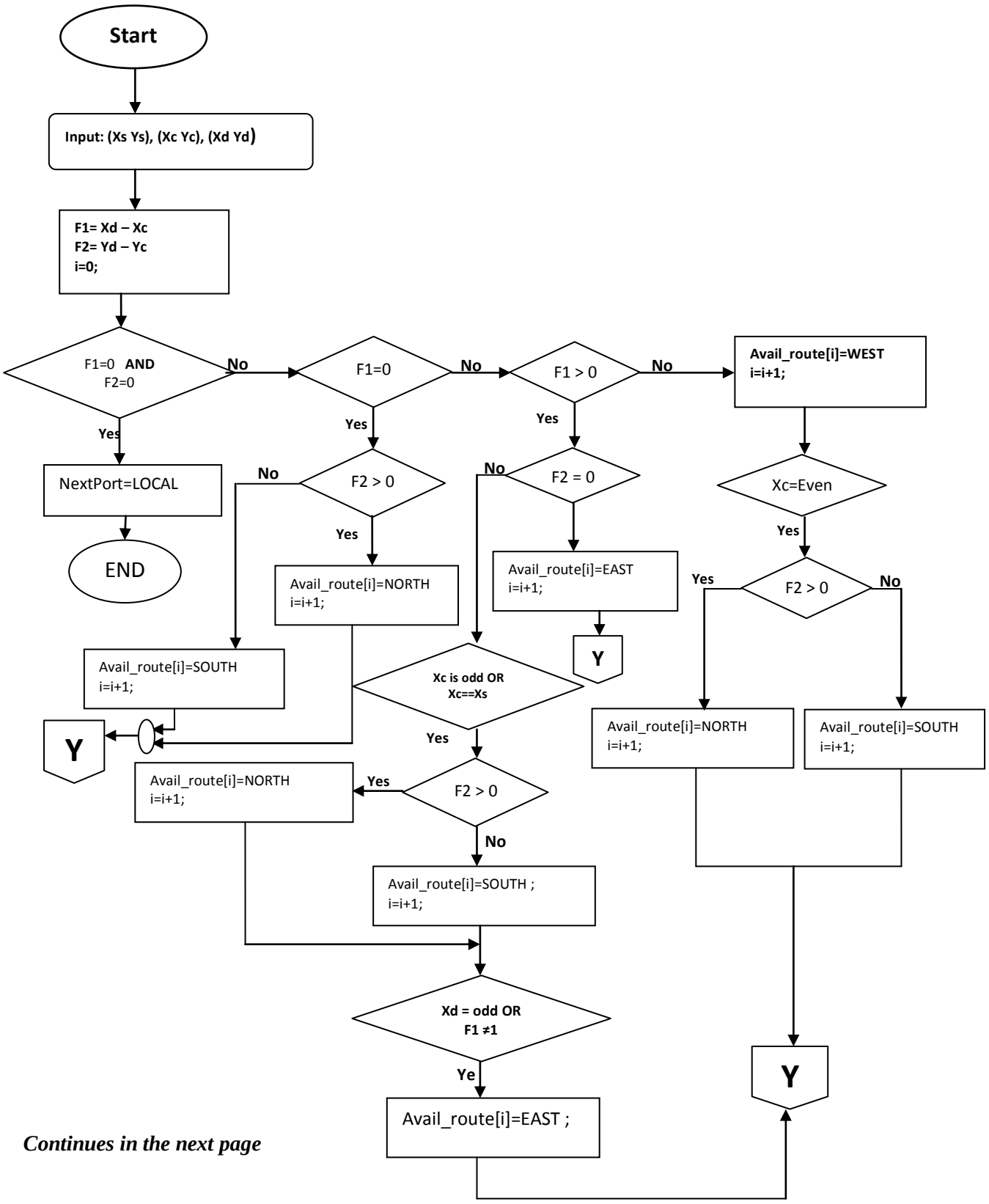
Traffic_In_E = Traffic from the east node

Traffic_In_S = Traffic from the south node

Traffic_In_W = Traffic from the west node

The variable **traffic**, is an array that keeps the traffic of available routes, *Avail_route*, then compare their traffic, that is, the traffic of the available routes and assign the route with smallest traffic to the *nextPort,* as the selected Next port for packet to be routed.

Below is the flowchart diagram.



Continues in the next page

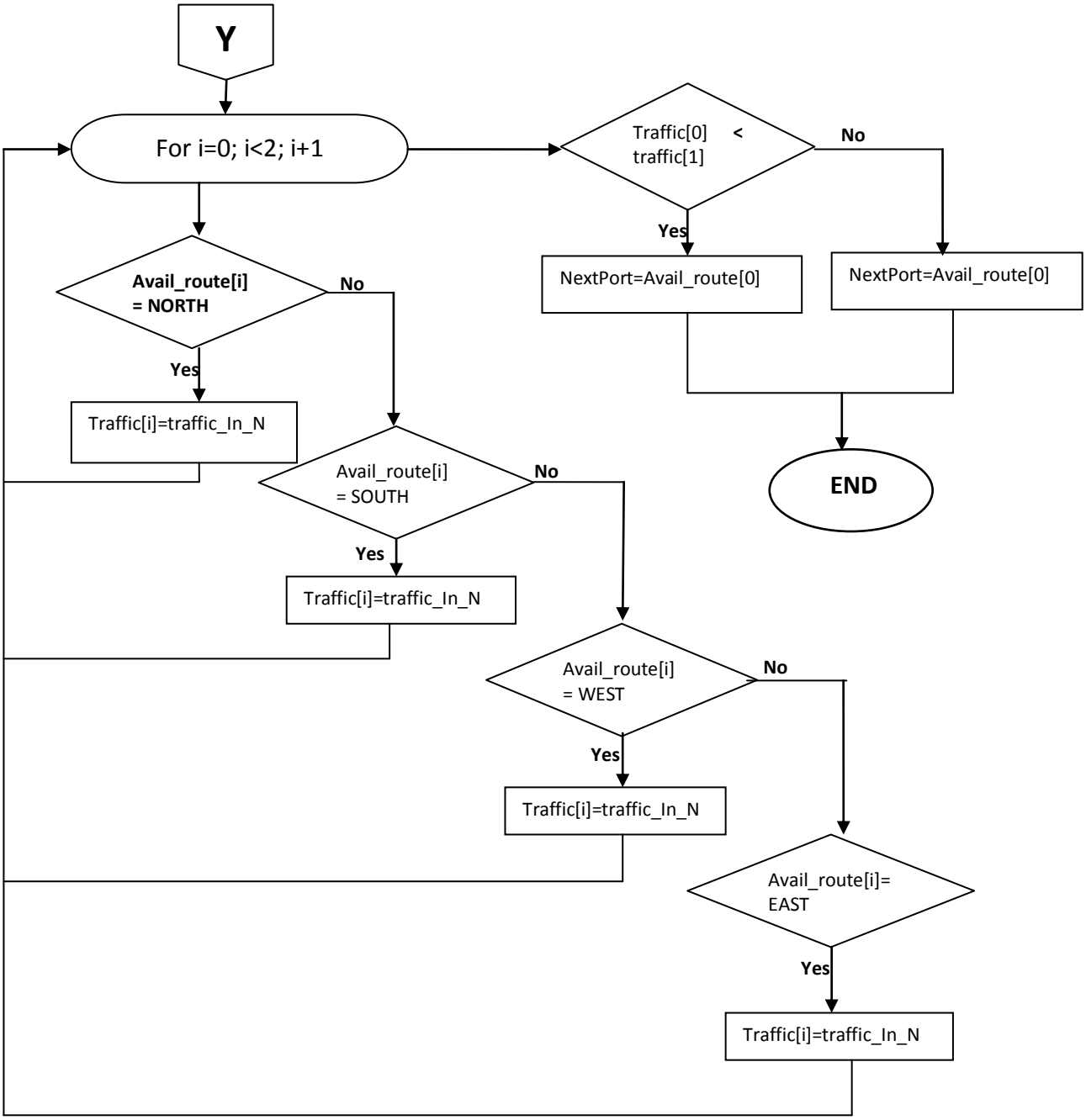


Fig 14: Algorithm Flowchart

4.2.2 Switching

This employs wormhole switching just as the deterministic routing explained in Chapter three. Hence the switching is as explained in chapter three, only that it route packets adaptively based on the above explained algorithm.

4.3 Architecture

The OASIS with the run – time monitoring system, so called Adaptive Network on Chip (ANOC) is a 4 X 4 Mesh network that adopts wormhole switching. The topology is a simple topology where nodes are connected to each other directly and each switch or router has a connection with the monitoring module called **Controller** as shown in Fig 15 below.

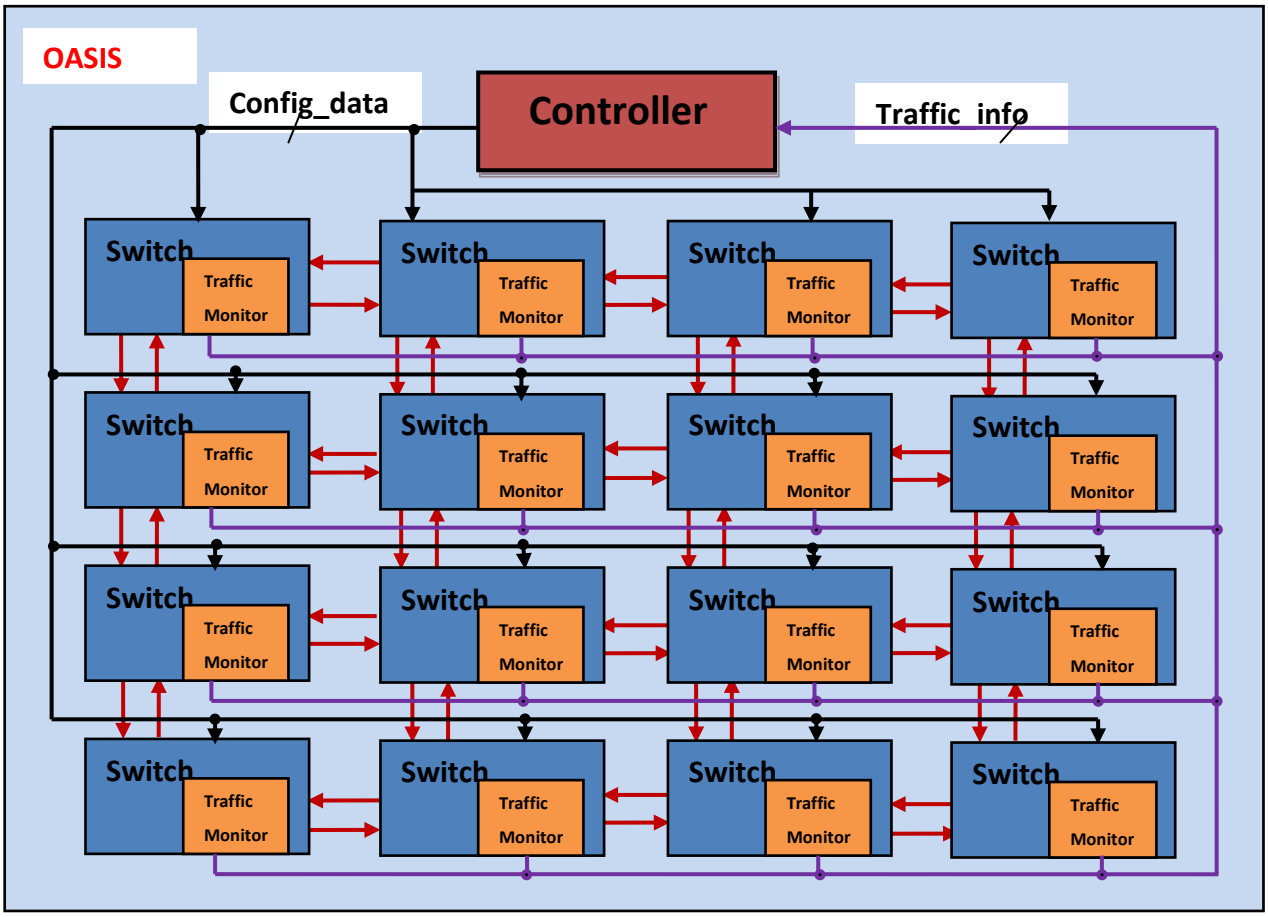


Fig 15: 4 X 4 Mesh topology ANOC Communication Network Architecture

4.3.1 Algorithm Implementation in Hardware

4.3.1.1 Switch

Hardware Description Language, Verilog HDL is used to describe the algorithm.

The addresses of the nodes: (Xc Yc), that is the current nodes' X and Y address and (Xd Yd), the destination nodes' X and Y address is used to determined the direction to move next based on the traffic condition.

Each Router has the *switch_tarffic_monitor*. The **Switch_Traffic_Monitor** module sends ***traffic_info***, to the **Controller** where it analysis the traffic condition of each switch and send back config_data. This config_data contains information about the traffic of neighbouring nodes and the buffer parameters as shown in **Fig 16**. The Quartus II synthesis Netlist RTL diagram is shown in **Fig 20** below.

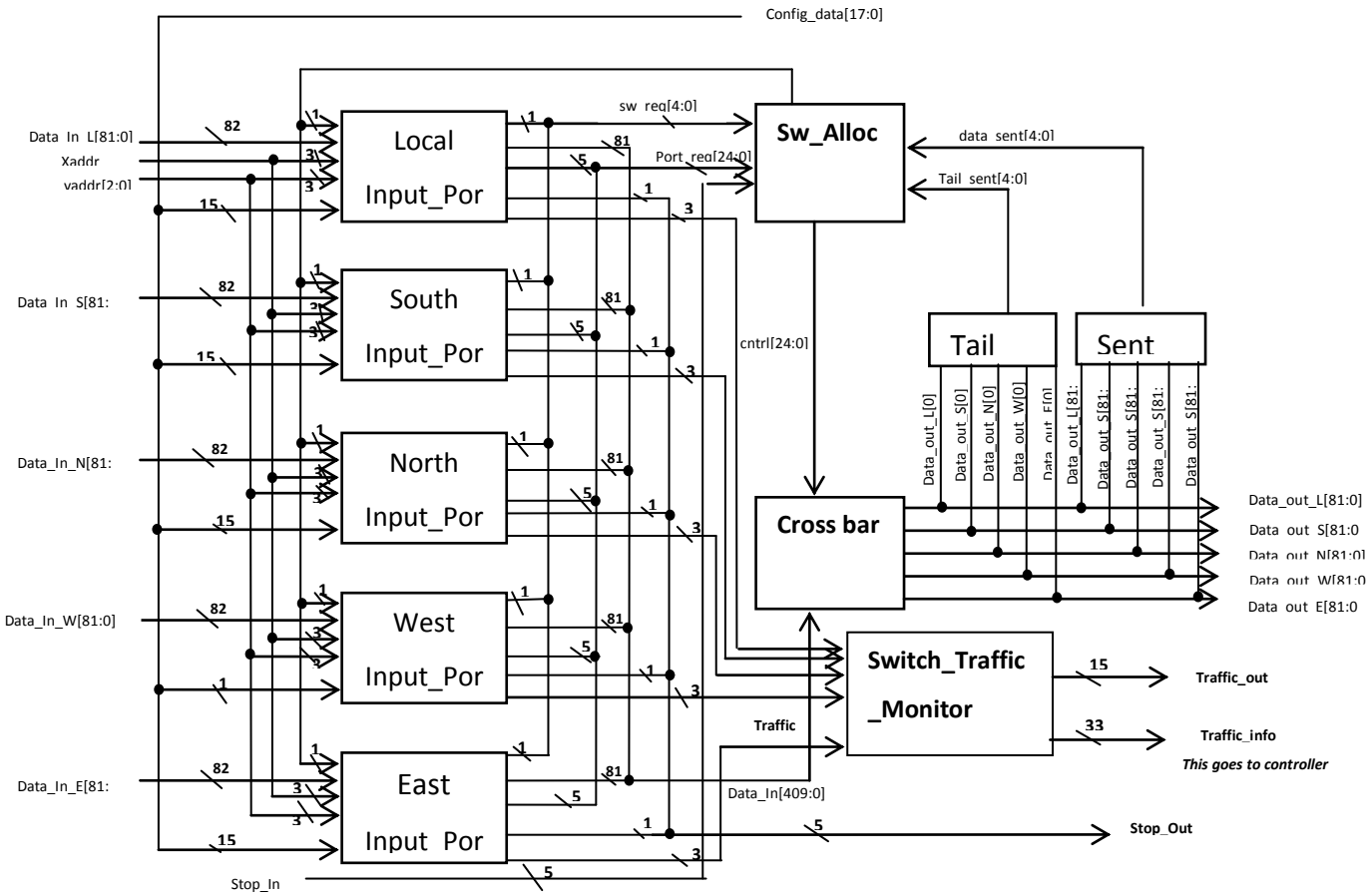


Fig 16: One Switch Datapath

4.3.1.2 Input Ports

The ***Input Ports*** of the switch or router each has a module called **Traffic Monitor (Fig 17) ,** this traffic monitor monitors the state of the buffer of each input port and output the number of flits that the buffer currently holds, *traffic*. This gives the traffic situation of the input port

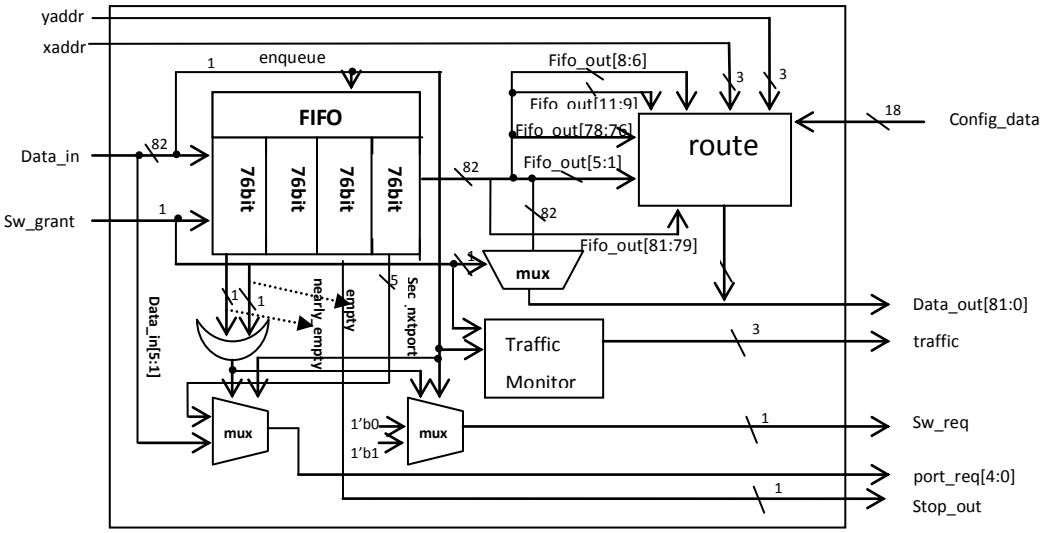


Fig 17: Input Port Circuit

4.3.1.3 Traffic Monitor

The traffic monitor unit is implemented in each of the input ports of a router. It takes as input, **sw_grant** and **enqueue.** These are the two signals that indicates either a packet‘s flit comes in to be queued in the **FIFO** (enqueue) or a switch grant is granted to an already queued flit in the FIFO (sw_grant).

Therefore, these two signals go into the *Traffic_monitor* and whenever there is an *enqueue*, the traffic, which keeps track of the number of flits in the FIFO is increased by

one. Otherwise if there is a *dequeue* or switch grant, then the *traffic* is reduced by one, as show by below short *algorithm.*

```
always @(posedge clk) begin
        IF (enqueue) begin
            assign traffic <= traffic + 3'b001;
        end
        IF (sw_grant == 1'b1 ) begin
        assign traffic <= traffic + 3'b001;
```

The output is *traffic* with 3 bits since all the necessary FIFO status can be captured by three *bits* (2^3). This will be used to determine how busy a route is base on the number of flits occupying the FIFO.

4.3.1.4 Flit Format Structure:

The format of flit is modified from the previous one to accommodate the source address of flit**, XSRC** and **YSRC.** This is because in the algorithm, the source address is needed to determine some turns. The structure of the flit is as shown in **Fig 18** below.

Fig 18: Flit Structure

4.3.1.5 Controller

This module is connected to all the Switches and gets the traffic of each of the Switch **Fig 19.** How it is interconnected in the OASIS is shown in **Fig 15**

Fig 19: Controller

How Controller functions

- ✓ The controller receives input from the switch, the traffic information of the switch (traffic_info).
- ✓ It analyses the traffics of the switches and provide the traffic conditions of the neighboring (N_tr, S_tr, W_tr, E_tr) switches. (Note: N_tr: North traffic, S_tr: South traffic ……)
- ✓ The traffic information of the neighboring nodes is passed to the switch, which is used by the **route** module of the input port to find the best routing path to follow.
- ✓ If the traffic conditions of the neighbouring nodes are not favourable, that is there is high traffic, then, it increase the buffer parameters of that switch otherwise if the there seems to be no traffic, then, the minimal normal size of the buffer is maintained. And send that parameter to the switch which adjusts its buffer size in order to accommodate the traffic.

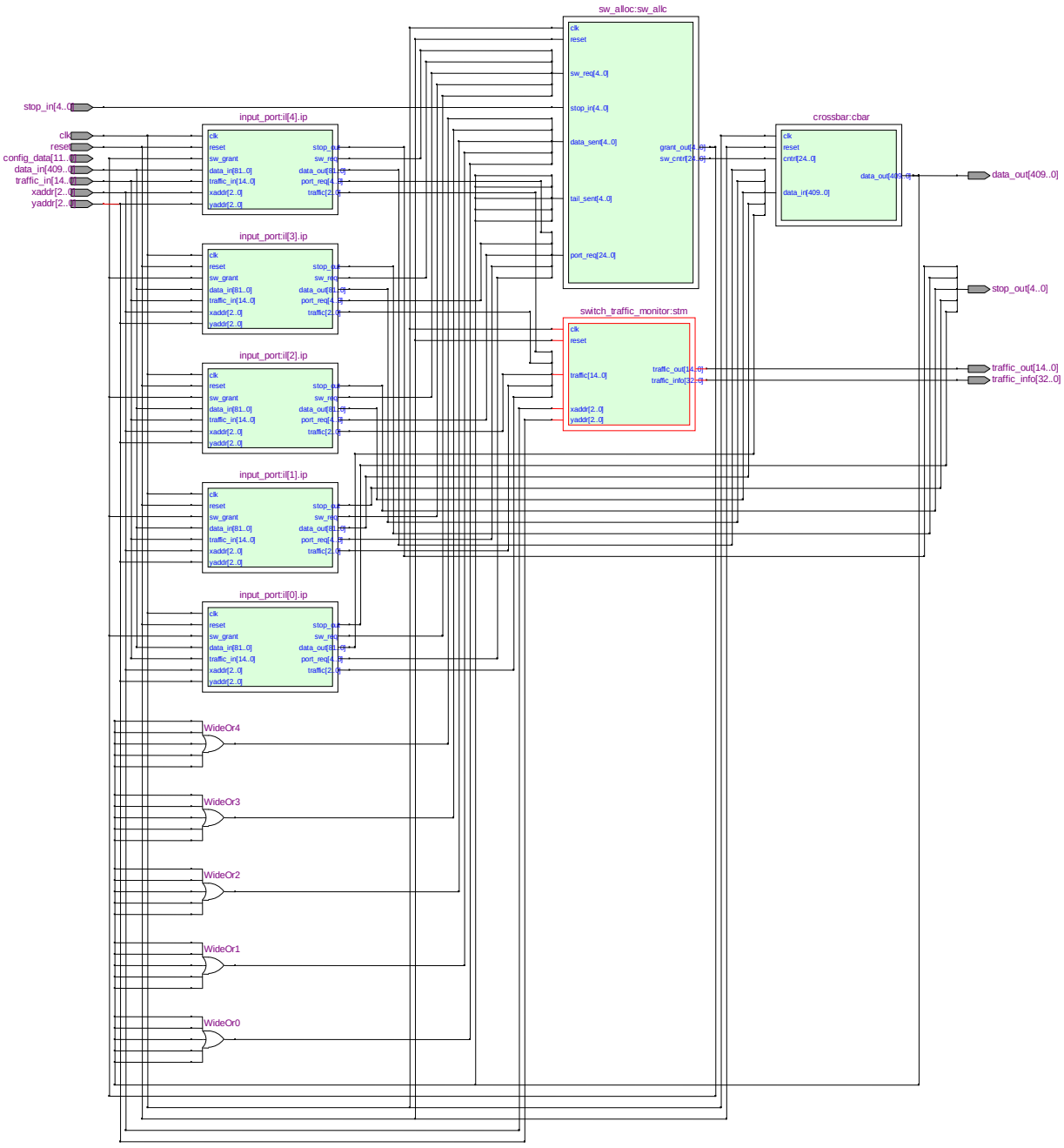


Fig 20: RTL diagram of SWITCH after synthesis with Quartuss II

Chapter 5 : Hardware and Software Evaluation Results

5.1 Hardware Complexity

The architecture of OASIS NoC with runtime monitoring system was designed in Verilog HDL and Altera CAD tools were used for synthesis.

5.1.1 Logic

The design logic elements of different device all of 4 X 4 mesh topology is as shown as Area (ALUTs) in table 1 below. It can be seen that the area/ logic of the adaptive network on chip (ANOC) OASIS, that is OASIS with run -time monitoring system is greater than the OASIS, this is simply because there are additional modules to the ANOC system than the OASIS, hence as expected, has a larger area or number of logic elements.

Device	Total Thermal Power Dissipation (mW)	Core Static Thermal Power dissipation (mW)	Core Dynamic thermal Power dissipation (mW)	I/O Thermal Power Dissipation (mW)	Speed (MHz)	Area (ALUTs)
Stratix III (OASIS)	397.68	370.00	0.00	27.68	478.70	37450
Stratix III (ANOC)	398.69	370.00	0.00	28.65	468.84	38090
Stratix II (OASIS)	325.20	302.99	0.00	22.21	469.90	12080
Stratix II (ANOC)	323.53	302.97	0.00	20.56	465.57	12480

Table 1: Power Dissipation of the Devices

5.1.2 Power

Quartus II synthesis and analysis provided the power dissipation of each of the devices, Stratix II and Stratix III. As shown in the table below.

From the table, different device family exhibits different characteristics. This is because parameters like supply voltage, electrical design, device architecture etc affects device power consumption. Stratix III devices consume power more than Stratix II device as shown in **table 1**.

The power consumed by Stratix III (OASIS) is slightly less than the power consumption of the adaptive Network on Chip (ANOC), that is, OASIS with run – time monitoring system. Though the area of ANOC is higher, the power consumed is not really higher as seen in **table 1** above.

5.1.3 Speed

After synthesis in Quartus II, table 1 above shows the speed of the devices. Observe that the speed of the OASIS is higher than the ANOC or the OASIS with monitoring. This can be because of the additional modules involved, though the disparity is not much.

Device	Votage Supply (V)	Total Current Drawn (mA)	Minimum power Supply Current (mA)
Stratix III (OASIS)	VCC (2.5V)	60.6	60.6
	VCCIO (2.5V)	5.16	5.16
	VCCPD (2.5V)	6.03	350
Stratix III (ANOC)	VCC (2.5V)	60.6	60.6
	VCCIO (2.5V)	5.17	5.17
	VCCPD (2.5V)	6.04	350
Stratix II (OASIS)	VCCINT (1.2V)	252.58	252.58
	VCCIO (3.3V)	4.42	4.42
	VCCPD (3.3V)	2.22	2.22
Stratix II (ANOC)	VCCINT (1.2V)	252.49	252.49
	VCCIO (3.3V)	4	4
	VCCPD (3.3V)	2.22	2.22

Table 2: Current drawn from Voltage supply

In the table above, *Total current Drawn* is sufficient for user operation of the device, *Minimum power Supply Current* is sufficient for device power -up and user operation of the device. From the table, it can be seen that little current is sufficient for the device use.

5.2 Functional Simulation

ModelSim simulator was used to perform simulation of both OASIS NoC and adaptive OASIS with run – time monitoring system (ANOC). A random number generator is used to generate flits and injected into the system at different rates and monitored.

5.2.1 Algorithm Verification

Figure 19 below shows the routing module. The destination of a flit, (xdest, ydest) is (0,0), and the source is (1,1). The possible route to its destination will be WEST which is represented by (10000), as can be seen in the **Fig 19** below.

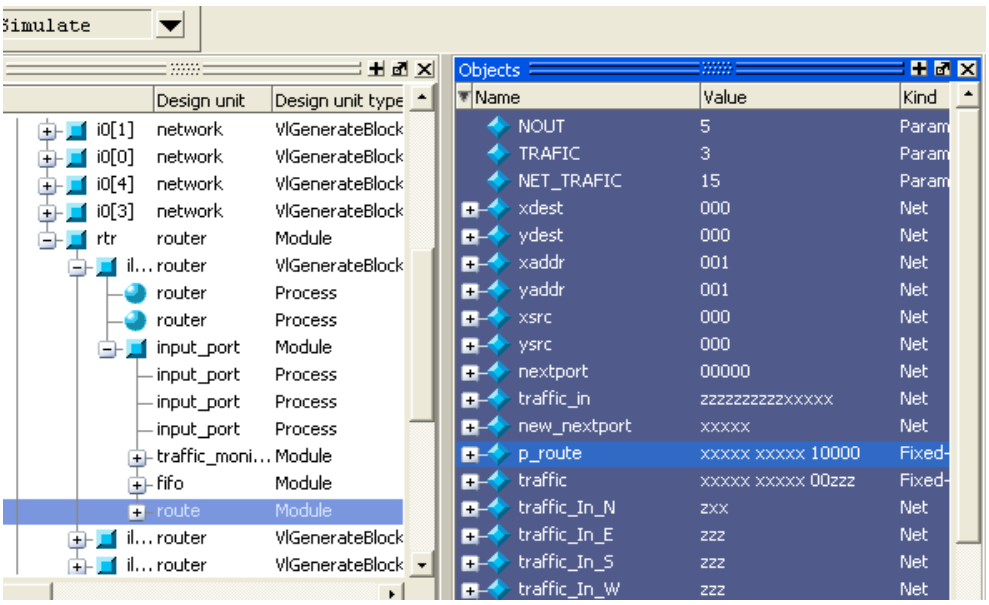


Fig 19 : Simulation algorithm verification

5.2.2 Packet Delay

In the table below, as the rate of flit generation increases, the time taken for the packet to reach its destination equally increases more in OASIS, which is the deterministic approach. In ANOC, the flit has smaller clock cycles (22) to reach its destination even under a higher flit generation rate (0.4). This is because as the network traffic increases, ANOC adaptive finds alternative less traffic route to its destination. The graph on **figure 20** below depicts the whole scenario.

Device	**Rate of Flit generation**	**Time (clock cycles)**
OASIS	0.23	8
	0.3	15
	0.4	38
ANOC	0.23	5
	0.3	10
	0.4	22

Table 2 : Flit generation rate and the time taken for it to reach its destination.

From the graph below, Fig 20, it can be seen that as the flit generation rate increases, the time taken for flit in OASIS (deterministic approach) greatly increases while ANOC (Adaptive approach) slightly increases. This is because as the congestion increases, ANOC adaptive adjust it parameters.

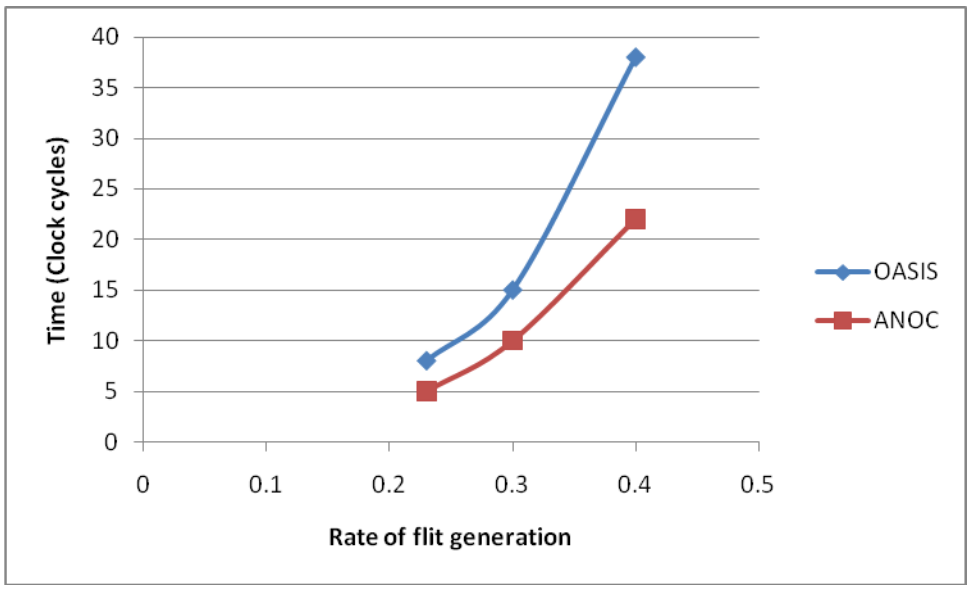


Fig 20: Graph showing delay of flit for both OASIS and ANOC.

Chapter 6 : CONCLUSION

We have implemented an adaptive approach to Network on Chip NoC, and made preliminary evaluation results of the complexity of the adaptive Network on Chip architecture called ANOC. The system was designed using Verilog HDL and was correctly synthesized with Altera CAD tools and ModelSim simulator.

The adaptive approach is not only a flexible and scalable approach for design of multiple cores in a single chip, but has also proven to be power efficient. The approach can dynamically meet the challenge of network traffic at run – time and adjust the parameters for efficient routing. Hence the approach has a less routing delay compared to the static approach.

The approach ANOC, though have shown some reasonable improvement on the routing of packets when the traffic is high, but in a situation of node failure, the approach cannot take care of the situation and packets can be lost. Also, in situation where by the packet movement or destination's address is far from the source, it very possible that this method may not be effective, because there will be more routing routes to follow meaning longer distance between source and destination, and the system cannot adjust the distance.

As a future research, the execution time of the execution speed can be properly measured. In case of high traffic injection and there exist a consistent movement of packets between certain nodes, that is, there are situation whereby node A always sends packets to node D and the distance between them is not close. This means that there will always be a delay in the routing. In this case, *dynamic re-mapping* of nodes can be implemented, whereby nodes A and D can be connected directly. In this way routing delay is greatly reduced. Also, this verification can be done using JPEG applications for proper evaluation of the approach.

REFERENCES

[1] S. Miura, A. B. Abdallah, K. Kuroda. "NoC - Design and Preliminary Evaluation of a Parameterizable NoC for MCSoC Generation and Design Space Eploration". *The 19th Intelligent System Symposium (FAN 2009), pp.314-317, Sep. 2009.*

[2] K. Mori, A. B. Abdallah, K. Kuroda. "Design and Evaluation of a Complexity Effective Network-on-Chip Architecture on FPGA". *The 19th Intelligent System Symposium (FAN 2009),pp.318-321, Sep. 2009.*

[3] B. A. Abderazek, M. Sowa. "Basic Network-on-Chip Interconnection for Future Gigascale MCSoCs Applications: Communication and Computation Orthogonalization". P*roceedings of the tjassst2006 symposium on science, society and technology, sousse, dec. 4-6, 2006.*

[4] W. Liu. "Efficient Application Mapping and Scheduling for Networks-on-Chip".*Technical Report: Department of Computer Science and Engineering Hong Kong University of Science and Technology April 10, 2008* .

[5] J. HU. "Design Methodologies for Application Specific Networks-On-Chip". *A PhD thesis submitted to the department of Electrical and Computer Engineering, Carnegie Mellon University USA, 2005.*

[6] T. Bjerregaard, S. Mahadevan. "A Survey of Research and Practices of Network-on-Chip" *ACM Computing Surveys, Vol. 38, March 2006, Article 1.*

[7] E. Behrouzian-Nezhad, A. Khademzadeh. “BIOS: A New Efficient Routing Algorithm for Network on Chip” *Contemporary Engineering Sciences, Vol. 2, 2009, no. 1, 37 – 46*

[8] V. Rantala T. Lehtonen J. Plosila“Network on Chip Routing Algorithms” *Turku Center for Computer Science, University Turku, Technical Report No 779, 2006.*

[9] S. K. Axel et al. “A Network on Chip Architecture and Design Methodology”. *Proceedings of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI, 2002)*

[10] L. M. Ni and P. K. McKinley, ”A Survey of Wormhole Routing Techniques in Direct Networks,” *IEEE Comp.Mag., vor 26, no. 2, Feb.1993, pp. 62-76.*

[11] W. J. Dally, ”Virtual-channel flow control”, *Parallel and Distributed Systems, IEEE Transactions on Volume 3, Issue 2, March 1992, pp.194-205.*

[12] J. Hu, R. Marculescu. “Energy- and performance-aware mapping for regular NoC architectures”. *IEEE Trans. on CAD of Integrated Circuits and Systems, 24(4), April 2005.*

[13] A. Pinto, et. al. “Efficient synthesis of networks on chip”. *In Proc. ICCD,Oct. 2003.*

[14] U. Ogras, J. Hu, R. Marculescu “Key Research Problems in NoC Design: A Holistic Perspective” *Department of Electrical and Computer Engineering Carnegie Mellon University Pittsburgh, PA 15213-3890, USA*

[15] C. J. Glass, L. M. Ni. “The turn model for adaptive routing”. *In ProcISCA, May 1992.*

[16] J. Hu, R. Marculescu. “DyAD-Smart routing for Networks-on-Chip”. *In Proc. DAC, June 2004.*

[17] M. Sleth Rasmussen, “Network-on-Chip in Digital Hearing Aids”, *Informatics and Mathematical Modelling, pp.38-41, Technical University of Denmark, DTU, 2006*

[18] R. Pui-Hung Pau. “A Configurable Router for Embedded Network-on-Chip Support in Field-Programmable Gate Arrays” *An MSc thesis submitted to the Department of Electrical and Computer Engineering, Queen’s University Kingston, Ontario, Canada September 2008.*

[19] B. Ahmad, T. Erdogan, S. Khawam “Architecture of a Dynamically Reconfigurable NoC for Adaptive Reconfigurable MPSoC”, *Proceedings of the First NASA/ESA Conference on Adaptive Hardware and Systems (AHS'06) 2006.*