A NEW ROUTER ARCHITECTURE FOR DIFFERENT NETWORK-ON-CHIP TOPOLOGIES

[1]Jaya R. Surywanshi, [2]Dr. Dinesh V. Padole
[1,2]Department of Electronics Engineering, G. H. Raisoni College of Engineering, Nagpur

Abstract— Network -On-Chip (NoC) is becoming the backbone of System on chip (SoC) architecture and router is the heart of an NOC architecture. This paper explores FPGA implementation of First Virtual Output Queue based Routers. The implementation is obtained through VHDL coding of router blocks in behavioral fashion which are later connected in structural style to get the complete router. In Virtual Output Queue based router the buffer implemented for virtual channel is in the form of Virtual Output Queue memory that provides low latency to the packets traversing the router. Major components of proposed routers are Input Port, allocators and crossbar switch. After implementation of router different NOC topologies namely ring, mesh and binary tree are carried out with respect to two parameters, Area and Delay and is presented with the help of "Xilinx ISE-13.1" software tool .

Keywords— Router, Topologies, FIFO.

I. INTRODUCTION

With the development of integration technology, System on-Chip (SoC), composed of various heterogeneous and homogeneous cores on a single chip and it has entered billion-transistor era. As the microprocessor industry is moving from single-core to multi-core and eventually to many-core architectures, containing hundreds to thousands of identical cores arranged as chip multiprocessors, which required efficient communications among processors. Both SoC and microprocessor call for a scalable, flexible, high-performance and design-friendly interconnection [2]. How to provide efficient communication poses a challenge to researchers.

Before the advent of network-on-chip, interconnection architectures are usually based on dedicated wires or shared buses. Dedicated wires provide point-to-point connection between every pair of nodes, effective for small systems of a few cores. But as the number of cores increases, the number of communication wires in the point-to-point architecture grows quadratically, making it unable to scale. Compared to dedicated wires, a shared bus which is a set of wires shared by multiple cores is more scalable and reusable. However, due to the inherent disadvantage of buses, only one communication transaction is possible at a time and blocking communication for all other cores. The limitation of shared bus architectures include long data delay, high energy consumption, high-bandwidth, low-latency, low-power consumption and scalability. NoC
poor data delivery increasing complexity in decoding/arbitration, low bandwidth [1]. It would be daunting inefficient if hundreds of nodes are connected by shared buses. Thus, the usage of shared buses is limited to a few dozens of IP cores. To deal with the problems in shared buses, a hierarchical architecture, which segments bus into

shorter ones, is introduced. Hierarchical bus architectures may relax some of constraints faced by dedicated wires and shared buses, since different buses may account for different bandwidth needs, different protocols and it also increase communication parallelism. Nonetheless, scalability remains a problem for hierarchical bus architectures. In order to meet the good communication requirements, accelerate time-to-market and cut down the communication energy consumption of large scale SoCs, there is a great need to find a new design alternative to the conventional point-to-point and bus based computation architectures[2]. NoC has been proposed for highly structured and scalable solution to address communication problems in SoC. On-chip interconnection network has several advantages over dedicated wiring and buses, i.e., high-bandwidth, low-power consumption, low-latency, and scalability. NoC architectures can guarantee communication pipelining with a pre-specified clock rate regardless of the network size, which is infeasible for bus-based architectures. For SoCs, cores can be DSPs, embedded memory blocks, or CPUs, or video processors, etc. Since its inception, NoC has drawn great attention from researchers all over the world. But to fully explore the benefits of NoC, numerous challenges and open problems are to be addressed. Open problems can be classified into four main categories, including Application Modeling and Optimization, NoC communication Architecture analysis and Optimization, NoC Architecture Evaluation for Communication, and NoC Design Validation and Synthesis. Due to the limit of space, we are not going to expand our scope to such a broad range.

Outline for this paper is as follows: After the introduction, we review some classic topologies in section 2. Then we discuss the concept of router and topologies in section 3.In section 4 the RTL and implementation results of router and different topologies are presented. At last section i.e. section 5 we make comments on results.

II. REVIEW OF SOME CLASSIC TOPOLOGIES

In this section, we review the most popular topologies, including ring, star, mesh, tree, fat tree, butterfly, and torus, etc. Later we summarize the strengths and limitations for some of them. In ring architecture[3-6], all nodes are connected in a ring fashion. Every node has two neighbors regardless the size of the ring. Its small degree is preferable, but its diameter increases linearly with the number of nodes.

Its strengths include:

(1) Cable faults are easily located, which makes troubleshooting easier.

(2) Moderately easy to install compared with other architectures.

Its limitations include:

(1) A single break in the cable can disrupt the entire network.

(2) Network Expansion can cause network disruption.

In a star architecture[7,8], assume there are N nodes, $N-1$ nodes connect to a center node.

Only the center node has a degree of N -- 1. Other nodes have degree of 1. Diameter of a star architecture is 2, regardless its size. Its strengths include: (1) A small diameter means a small average hop distance, which is a favorable characteristic. (2) Simplicity to operate. Each node is isolated free of impact from failed nodes.

Its limitations include:

(1) Failure of the central node fails the entire network. (2)Central node is the bottleneck.

In a mesh, nodes are connected as a grid. Expansion is easy for meshes. Little effort is needed when adding more nodes to the existing architecture. Nodes have different degrees according to their locations within the mesh. Nodes at corner position have degree of 2. Edge nodes have degree of 3. Inner nodes have degree of 4. Its strengths include:

(1) Multiple paths between a pair of nodes, tolerant to link failure.

(2) Easy to expand.

Its limitations include:

(1) Diameter can be very large.

(2) Irregularity, less bandwidth for nodes at corners and edges.

In a binary tree, the top node is at the root position and the bottom nodes are at the leaves side. Every node except the root node has two offspring nodes. A node's children are those nodes that appear immediately beneath the node itself. A parent

node's is the node immediately above it. A tree's root structure is the single node that contains no parent. Its advantages include:
(1) Supported by many network vendors and even hardware vendors.
(2)All the nodes have access to the larger and their immediate networks, best for branched out networks.
Its limitations include:
(1)Bottleneck on the root node.
(2) When the tree is big, it is difficult to configure and can get complicated after a certain point.
In a fat binary tree, only leaves are intellectual properties
(IP). Interior nodes are switches. When moving towards the root node, there are more links between a parent node and a child node. The number of inter-node links increases by order of 2. Basic butterfly networks have two main disadvantages. Firstly, it lacks of path diversity. There is only one path from a source node to a destination node. Secondly, long wires are inevitable. Long wires must transverse half of the diameter of the network. A torus architecture is obtained by adding direct connections to two end nodes in the same row or column in a mesh architecture[9]. Compared with mesh, its diameter is reduced. A regular torus has long wrap-around links. By folding a torus, long wires can be avoided at the cost of doubling the wire length.

III. ROUTER AND TOPOLOGIES

A micro-router is built according to the OSI-model where different layers perform specified services in it. This gives a ground structure to design upon and provides ability to change the service on one layer without affecting the other. The micro-router should be able to communicate between micro-routers both on network layer and data- link layer. The reason for this is that it should be possible to make a fast simulation in SDL only using the network layer, or to make a more accurate simulation with both layers.
The micro-router itself is divided into several functional blocks showed in Figure 1,

- I/O-buffers presents, for communication with other micro-routers
- I/O-buffer for RNI communication.
- For switching the packets into the right direction

Route-Control unit is presents.

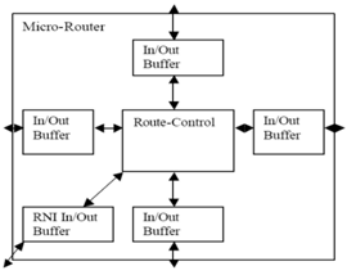

Figure 1: Router Micro-Architecture

Mesh Topologies

A mesh [4] topology is simple topology arranged like lattice has m lines and n rows. A mesh topology consists of resources and switches. A number of resources is same as a number of switches. Interactive communication channels tie to each routers each resources or to a router other routers. Router excluding edge routers has 5 input ports and 5 output ports. A mesh topology can make routing easily[10]. As a results, an architecture of router is simple, router become low area and low power consumption. Layout of mesh is very regular layout . So mesh topology is excellent in the scalability. So if designer want to make a NoC for application need to scalability, mesh topology is as good as one.

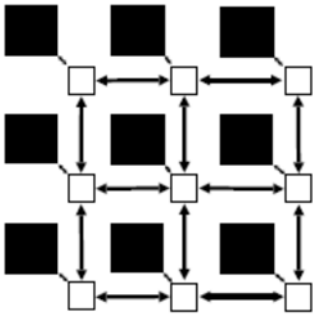

Figure 2: Mesh topology

BFT Topologies

This topology makes tree structure, resources are presents at leaf and switches are internal node. Each node is represented by height, level of tree, and position, a node number from right edge. The simplicity of this topology is that switches number is smaller than other topologies. So the area of whole network can be predict easily.

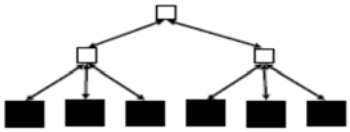

Figure 3: BFT topology

RING Topologies

Basic Ring topology has multiple resources, many routers and different interactive connections. Octagon mechanism is used to configure Ring topology. In basic ring, a number of pop is suppressed to two. And repetition of route of each packet connection is few. So high throughput is obtained. But if resource numbers more than eight, scalability of this topology is very low [11, 12] because physical layout becomes very complex.

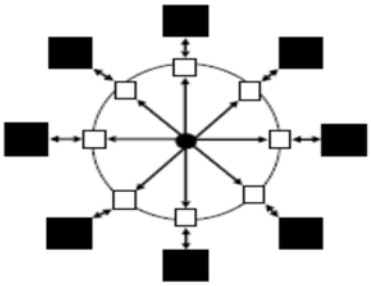

Figure 4: RING topology

IV. IMPLEMENTATION OF ROUTER ARCHITECTURE

The implementation starts with designing of VHDL codes for the blocks of router that include Virtual Output Queue (VOQ), scheduler and Demux . The Virtual Output Queue (VOQ) block consists of control unit and memory block to store the data bytes. All of these blocks are then connected using structural style of modeling to get the complete router blocks. The structural view of the router block is as below:

The task a router is to take the data from the incoming Input Port (IP), check which direction it needs to be forwarded to, and forwarded it on to the next router. The Conventional Switch router is very simple and easy to implement. It provides a simple data-path for packets, being composed only of a crossbar with registered outputs and need not to split the data packet. . This is advantage of the CS router but it requires large buffer size. So some modification should be there in conventional CS.

Virtual Output Queue is new methodology for designing the router. Virtual Output Queue (VOQ) architecture is preferred in our design to achieve less average latency of packets with decrement in power consumption and area compared with that of the traditional Virtual Channel (VC) switch architecture. In this architecture four VOQ are assigned to an input port and each one connected to different output port. The FLITs in different VOQ which compete for a same output port will be transferred to the next hop in the round robin mode. VOQ(i,j) storages the FLIT arrived at input port i which is prepared to be transmitted to output channel j. Each output channel is allocated by the switch allocation unit (SA) when a VOQ with FLIT arrived submits requirements. A FLIT will be delivered to the requiring output port if allowed by the SA. In the VOQ switch architecture, look-ahead routing is used to calculate the output port of the next switch node for a packet in the routing modules to reduce the switch latency. The incoming data is stored in one of the available Virtual Channel (VC). The free VC has been calculated by the routing algorithm in the previous router. Before the flit is stored, it is checked, if the new flit is a header flit. In this case, the next Output Port (OP) and VC number is calculated by the routing algorithm and stored along with the flit. In the next step the neighboring routers report their states of the VCs. This ensures that the IP only considers flits which have a chance to be forwarded. Suppose, if IP 2 of the next router reports that all its VCs are full, the flits which want to be routed towards IP 2 do not need to be considered. This creates a bit array of VCs that contain data and have a chance to be forwarded. An arbiter residing inside the IP, chooses one of the requesting VCs and the IP reports the desired OP to the router. There is possibility that multiple IP might request for the same route, hence another arbitration step (IP/OP arbitration step) is required to resolve this conflict. The IP that won the IP/OP arbitration, transmits its flit and deletes it from the chosen VC. In case of

multi flit environment the OP is bound to the IP for the entire duration of the transmission of the packet to prevent interleaving of flits from other IPs to the same OP. The flit traverses through the crossbar and is received at the OP in which its relative address is updated in case it is a header flit. Since the distance changes when the flit traverses the Network on Chip (NoC), it needs to be updated at every node the flit passes through. If all elements of the address tuple equal 0, the flit reached its destination and is ejected from the network. After the address has been updated and the flit is passed on towards the next router.

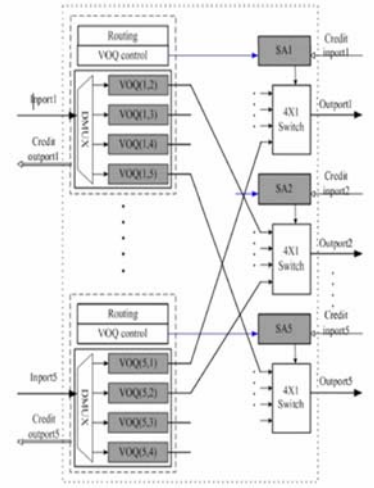

Figure 5: Router Architecture

V. RESULTS AND CONCLUSION

We present here Virtual Output Queue router architecture and different topologies . We are going to implement the RTL for this Router suing " Xilinx ISE-13.1" software tool . The router is then used to construct three network topologies namely Mesh, BFT and Ring and a functional simulation is then carried out to verify the functionality of these topologies. The future score shall be implementation of presented router with different techniques that then present the analysis of delay and area for all such type of routers.

REFERENCES

[1] C. Grecu, P. Pande, A. Ivanov, and R. Saleh, "Timing analysis of network on chip architectures for mp-soc platforms," *Microelectronics Journal*, vol. 36, no. 9, pp. 833–845, 2005.

[2] J. D. Owens, W. J. Dally, R. Ho, D. N. Jayasimha, S. W. Keckler, and L.-S. Peh, "Research challenges for on-chip interconnection networks," *Micro, IEEE*, vol. 27, no. 5, pp. 96 – 108, September, 2007.

[3]http://www.xbitlabs.com/news/mobile/display/20080603141353 Nvidia Unleashes Tegra System on Chip for Handheld Devices.html.

[4] W. J.Dally and B. Towles, "Route packets, not wires: On-chip interconnection networks," in *Proceedings of the 38th Annual Design Automation Conference*, Las Vegas, Nevada, United States, June, 2001, pp. 684–689.

[5] A. El-amawy and S. Latifi, "Properties and performance of folded hypercubes," *IEEE Transaction on Parallel Distributed System*, vol. 2, no. 1, pp. 31–42, 1991.

[6] K. Efe, "The crossed cube architecture for parallel computation," *IEEE Transaction on Parallel Distributed System*, vol. 3, no. 5, pp. 513–524, 2004.

[7] Y. Li, S. Peng, and W. Chu, "Efficient collective communications in dual-cube," *The Journal of Supercomputing*, vol. 28, no. 1, pp. 71–90, 2004.

[8] S. G. Ziavras, "Rh: A versatile family of reduced hypercube interconnection networks," *IEEE Transaction on Parallel Distributed System*, vol. 5, no. 11, pp. 1210–1220, 1994.

[9] K. Ghose and K. R. Desai, "Hierarchical cubic networks," *IEEE Transaction on Parallel Distributed System*, vol. 6, no. 4, pp. 427–435, 1995.

[10] F. P.Preparata and J. Vuillemin, "The cube-connected cycles: A versatile network for parallel computation," *Communication of ACM*, vol. 24, pp. 300–309, 1981.

[11] Y. Li, S. Peng, and W. Chu, "Metacube - a versatile family of interconnection networks for extremely large-scale supercomputers," *The Journal of Supercomputing*, vol. 53, no. 2, pp. 329 – 351, 2010.

[12] W. J. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. Morgan Kaufmann, 2003.