International Journal of Engineering Innovation & Research
Volume 6, Issue 5, (September) ICEMS-2017, ISSN (Online) : 2277 – 5668

International Conference on Advances in Engineering Management & Sciences - ICEMS -2017

NOC: An Efficient Data Transmission with Shared Buffer Routers Using RoShaQ Architecture

C.V.SUBHASKARA REDDY
M.Tech (Ph.D), Associate Professor
Santhiram Engineering College, Nandyal

S.ARIF BASHA
Assistant professor
Santhiram Engineering College, Nandyal

C.MOHAN KUMAR GOUD
M.Tech
Santhiram Engineering College, Nandyal

***Abstract* – IP Routers/ NOC with a new and powerful architecture are highly needed for today's internet/links and plays a vital role in the present modern telecommunication networks, using digital bit-packet switching over multiplexed links. Fortunately, both networking and silicon technologies has merged into a landscape for implementing high-speed routers with shared buffer queues at respective input and output ports to active high efficient transmission with low latency. NOC is one such paradigm that makes an efficient interconnection structure which alleviates traditional on chip inter connections such as p2p links, network interface cards, buses and bridges Router consists of buffers which are dedicated to their input or output ports for temporarily storing packets during the times of congestion. Unfortunately, significant portion of the router area and power is consumed by the buffers alone. While running some tested traffic patterns, however, not all input ports of routers have incoming packets needed to be transferred simultaneously. Therefore, a large number of buffer queues in the network are empty whereas the other queues are mostly busy.**

This observation has led to the design of router architecture with shared queues (RoShaQ) which maximizes the buffer utilization by sharing the multiple buffer queues among input ports. Buffers become more efficient by sharing queues, hence the router is able to achieve higher throughput when the network load becomes heavy. On the other side, at light traffic load, this router is designed to achieve low latency by allowing packets to effectively bypass these shared queues.

***Keywords* – NoC, Router, RoShaQ, Buffer Utilization.**

I. INTRODUCTION

Luca Benini and Giovanni at. el [2] have proposed a new concept on inter control switching architecture: the network on chip (NOC) which includes distribution of communication channels on multiple routers for data transfer. A packet switched NOC consists of routers (switches), network interfaces in between the routers with processing core units (PC) and the interconnection network or physical link that connect switches and NI's. The processing core may be general propose processor/a DSP/an embedded systems etc. Figure 1 below illustrates an example of a 2D mesh NoC.

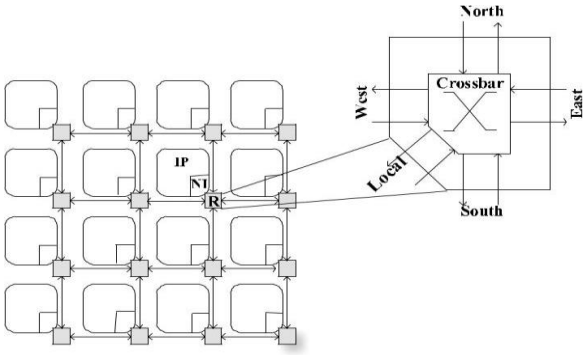

Fig. 1. Network on Chip with mesh topology

High throughput and low- latency are two major design parameters to NOC: that shows tradeoff with pore consumption and area occupancy. This tradeoff leads to new design challenges that includes:

1) Topology, suitable for NOC's such as mesh, ring, torus, tree and butterfly.2) Design N.I to a access the on chip network and routers to provide the physical interconnection mechanism to transport data between processing cores.3) The selection of communication protocols including routing, switching, buffer management and flow control etc, finally scalability and switching speed of the NOC.

In this article we implement a new router architecture with shared queues (Roshaq) at the respective input and output ports of router to achieve efficient data transmission with low-latency.

Generally a router consists of an input port with buffer for temporarily storing packets in cases that output physical channels are busy. These buffers can be single queue as in warmhole (WH) router or multiple queues in parallel as in virtual channel (VC) router. These buffers, in fact consume significant part of area and power.

II. ROUTER COMPONENTS

Router is one constituent element of global internet where its primary role is to transfer packets from inbound network interface to outbound interface so that they can reaches to next hop on the journey to their final destination. These packets processed by processing module and possibly, stored in the buffering module. Bei Yin [3] had developed warmhole router to support multicast environment which is a special case of cut-through switching. Instead of storing a packet completely in a buffer at a node, and then forwarding it to next node, warmhole router operates by advancing the packet head or

header flit directly to the next routing chip. As the header flit governs the route and examines it and selects the next node on the route and starts forwarding flits down that node.

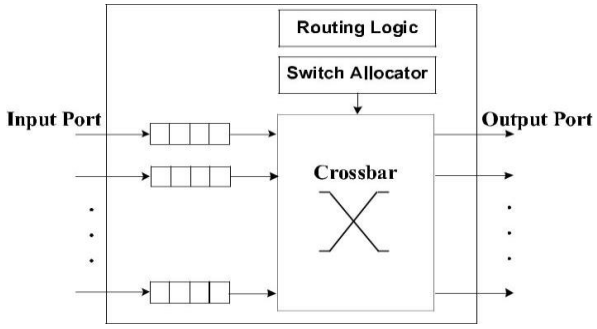

Fig. 2. Structure of Wormhole switch

The drawback in a WH router is, if a packet at the head of a queue is blocked, all packets behind it also stall. This head of line blocking problem can be solved by a VC router as shown in figure3 has multiple queues in parallel for an input buffer. Here each queue is called a VC, which allows packets from different queues to bypass each other to advance to the crossbar stage instead of being blocked by a packet at the head of queue.

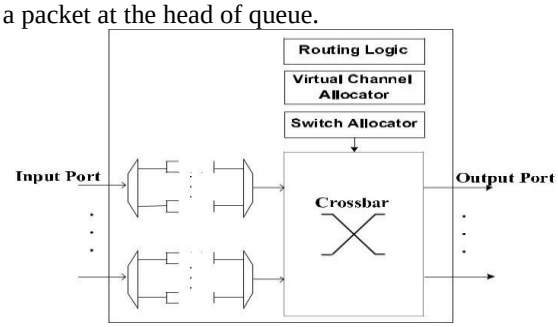

Fig. 3. Structure of Virtual Channel switch

Before reaching an output port, a packet on one VC has to choose a VC of its next router's input port. This operation is performed by virtual channel allocator (VCA) in parallel with look ahead routing computation (LRC) ; hence the VC Router achieves higher throughput [4] than a WH router while having the same number of buffer entries per input port. It also has higher zero-load latency due to deeper pipeline.

III. ROSHAQ: Router Architecture With Shared Queues

In this design, a packet from an input queue simultaneously arbitrates for both shared queues and an output port; if it wins the output port, it would be forwarded to the downstream router at the next cycle. Otherwise, that means having congestion at the corresponding output port; it can be buffered to the shared queues. Intuitively, at low load, the network would have low latency because packets seem to frequently bypass shared queues. While at heavy load, shared queues are used to temporarily store packets hence reducing their stall times at input ports that would improve the network throughput [5].

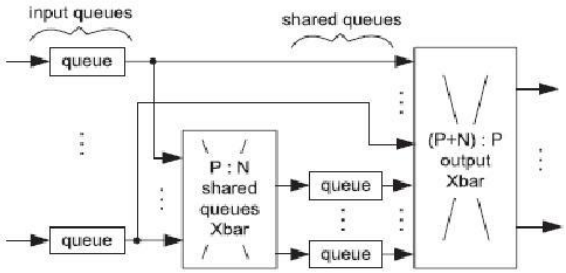

Fig. 4. Allows input packets to bypass shared queues. P: the number of router ports. V: the number of VC queues per input port in a VC router N: the number of shared queues

3.1 ROSHAQ Architecture

RoShaQ [6], router architecture with shared queues based on the idea of Fig. 4 is shown in Fig. 5. When an input port receives a packet, it calculates its output port for the next router (look ahead routing), at the same time it arbitrates for both its decided output port and shared queues. If it receives a grant from the output port allocators (OPAs), it will advance to its output port in the next cycle. Otherwise, if it receives a grant to a shared queue [7][8], it will be written to that shared queue at the next cycle. In case that it receives both grants, it will prioritize to advance to the output port.

Shared-queues allocator (SQA) receives requests from all input queues and grants the permission to their packets for accessing non full shared queues. Packets from input queues are allowed to write to a share queue only if: 1) the shared queue is empty or 2) the shared queue is containing packets having the same output port as the requesting packet. This shared queue writing policy guarantees deadlock-free for the network.

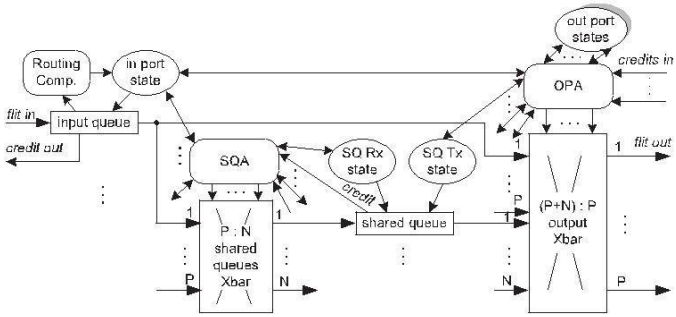

Fig. 5. Generalized architecture of RoShaQ router. SQA: Shared Queue Allocator OPA: Output Port Allocator SQRx State: Shared Queue receiving/writing state SQTx State: Shared Queue Transmitting/reading

The OPA receives requests from both input queues and shared queues. Both SQA and OPA grant these requests in round-robin manner to guarantee fairness and also to avoid starvation and live lock. Input queue, output port, and

shared-queue states maintain the status (idle, wait, or busy) of all queues and output ports, and incorporate with SQA and OPA to control the overall operation of the router. Only input queues of RoShaQ have routing computation logic because packets in the shared queues were written from input queues hence they already have their output port information. RoShaQ has the same I/O interface as a typical l router that means they have the same number of I/O channels with flit-level flow control and credit-based backpressure management

3.2 ROSHAQ'S Properties

1) *A network of RoShaQ routers is deadlock-free.*
At light load, packets normally bypass shared queues, so RoShaQ acts as a WH router hence the network is deadlock-free . At heavy load, if a packet cannot win the output port, it is allowed to write only to a shared queue which is empty or contains packets having the same output port. Clearly, in this case RoShaQ acts as an output-buffered router which is also shown to be deadlock-free.

2) *A network of RoShaQ routers is live lock-free.*
Because both OPA and SQA use round-robin arbiters, each packet always has a chance to advance to the next router closer to its destination; hence the network is also free from live lock.

3) *RoShaQ supports any adaptive routing algorithm.*
The output port for each packet is only computed at its input queue, not at shared queues. Therefore, any adaptive routing algorithm which works for WH routers also works for RoShaQ.

4) RoShaQ can be used for any network topology.
If we hide all design details inside RoShaQ, we would see RoShaQ only has one buffer queue at each input port similar to a WH router. Therefore, we can change the number of RoShaQ's I/O ports to make it compatible with any network topology known in the literature along with an appropriate routing algorithm.

IV. EXPERIMENTAL RESULTS

Fig. 6. Simulation waveforms of warmhole router

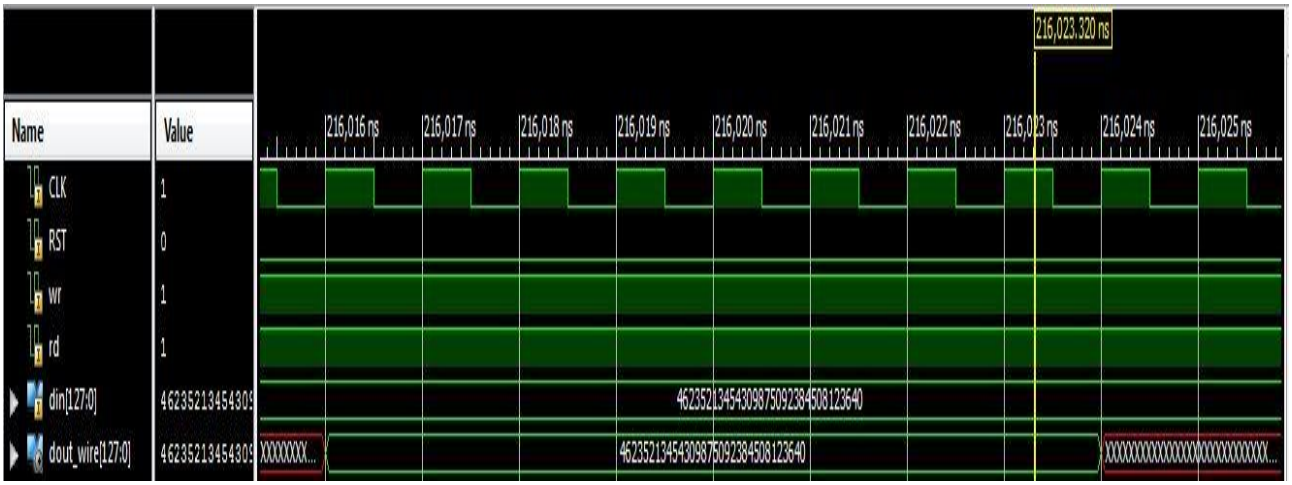

Fig. 7. Simulation waveforms of Proposed Top Module of ROSHAQ Architecture

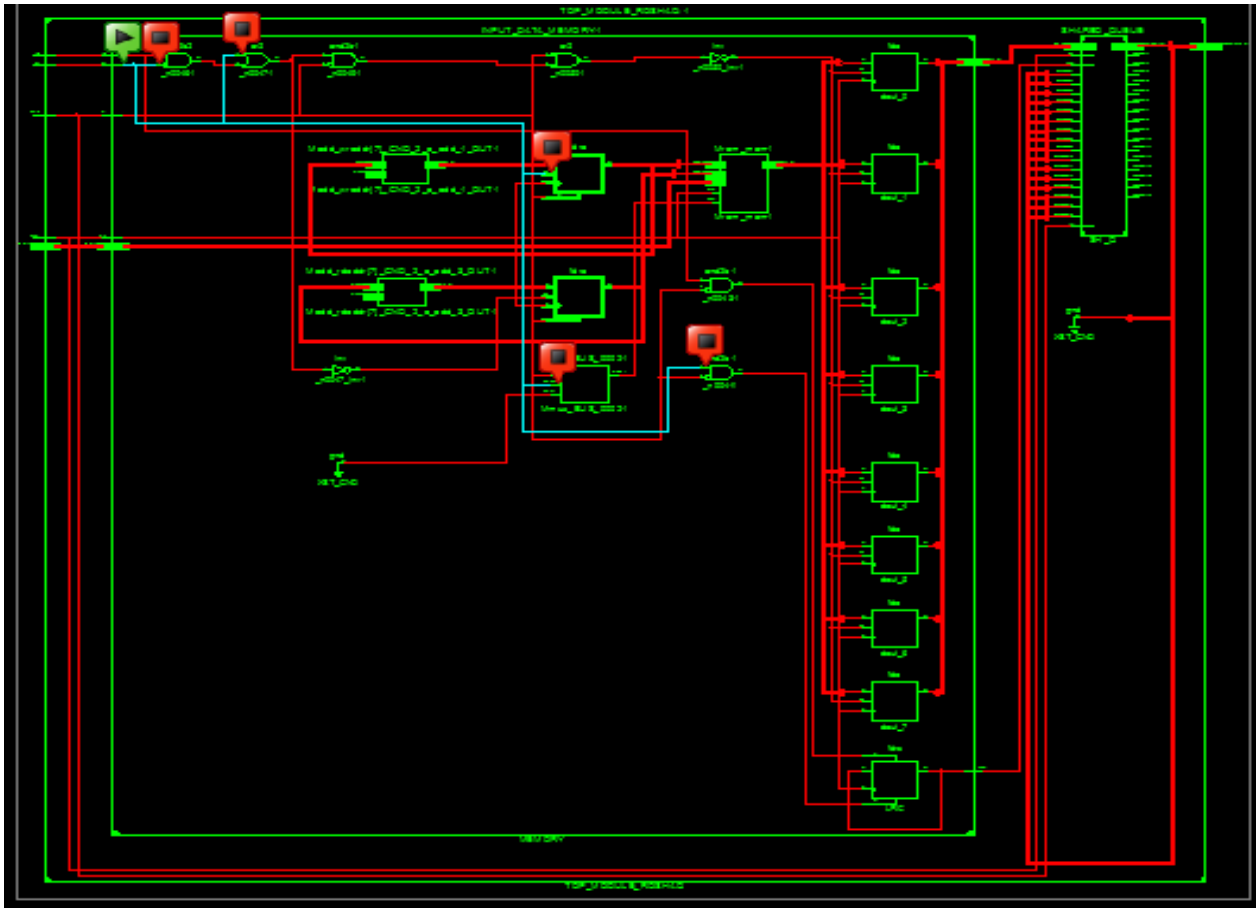

Fig. 8. RTL schematic for Proposed ROSHAQ Top Module

V. CONCLUSION

NoC brings notable improvements over conventional bus and cross bar interconnections. NoC improves the scalability of SoC's and power efficiency of complex SoC's. The NoC area has a significant influence in the design of next generation SoC or multicore architectures. Router is a significant component of the NoC. It is a novel router architecture that allows sharing of multiple buffer queues for improving network performance. In this paper we implemented RoShaQ architecture functionality using Xilinx. Verilog HDL code. RTL schematic view of proposed router was achieved by synthesized by verilog code using Xilinx.

REFERENCES

[1] William J.Dally and Nrian Towles. "Route Packets,not wires:On-chip interconnection Networks",DAC2001.

[2] L.Benini and G.D.Micheli, "Networks on chips: A new SoC paradigm", Computer Vol.35, N0.1,2002.

[3] Bei Yin,"Design and implementation of Wormhole Router Supporting Multicast for NoC", Master Thesis, Sweden 2005.

[4] K. Latif, T. Seceleanu, and H. Tenhunen, "Power and area efficient design of network-on-chip router through utilization of idle buffers," in Proc. 17th IEEE Int. Conf. Workshops ECBS, Mar. 2010,pp. 131–138.

[5] R.S.Ramanujam, V.Soteriou, B.Lin and L.S.Peh, "Extending the effective throughput of NoC's with Distributed Shared Buffer Routers", IEEE Transactions on Computer-Aided Design Integrated Circuits Systems., Vol 30, no. 4, Apr 2011.

[6] A.T.Tran and B.M. Baas,"RoShaQ: High performance on-chip router with shared queues", in Proc.ICCD, Oct 2011.PP 232-238

[7] Anh T. Tran and Bevan M. Bass, "Achieving High Performance On-Chip Networks With Shared Buffer Routers", IEEE Transactions on VLSI Systems, Vol 22, No.6,June 2014

[8] B.S.L.K.Anusha, K.S.N.Raju,"Design of High Performance Shared Buffer NoC Router", IJECS, Vol.4,issue 9,2015.

[9] S.Sivasankari, M.Selvaganesan, "Routers in on chip network using RoShaQ for effective transmission" IJTES,Vol.7, N0.2,2015.

[10] Manel Langar, Raidh Bourguiba, Jaouhar Mouine,"Network on chip,router architectures and performance challenges", IJAER,Vol.11,No.6,2016.