IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

FPGA IMPLEMENTATION OF HIGH-PERFORMANCE NETWORK ON CHIP ROUTER DESIGN USING SHARED BUFFER ARCHITECTURE

[1]Bakialakshmi D,[2]Varsha J

[1]Assistant Professor,[2] Assistant Professor,
[1]Electronics and Communication Engineering,
[1]Unnamalai Institute of Technology, Kovilpatti,Tamil Nadu,India

Abstract : On-chip routers typically have buffers dedicated to their input or output ports for temporarily storing packets in case contention occurs on output physical channels. Buffers, unfortunately, consume significant portions of router area and power budgets. While running a traffic trace, however, not all input ports of routers have incoming packets needed to be transferred simultaneously. Therefore, many buffer queues in the network are empty and other queues are mostly busy. This observation motivates us to design router architecture with shared queues (RoShaQ), router architecture that maximizes buffer utilization by allowing the sharing multiple buffer queues among input ports. A parallel cross bar architecture is proposed in this project to reduce power consumption. Also, a new adaptive WXY routing algorithm for 8-port router Architecture is proposed to increase throughput of the network on chip router. The proposed system is simulated using Modelsim and synthesized using Xilinx Project Navigator.

IndexTerms - **FPGA, RoShaQ**

INTRODUCTION

Systems on chip toward multicore style for taking advantage of technology scaling and additionally for rushing up system performance through accumulated correspondence within the proven fact that power wall limits the rise of the clock frequency**.** Networks on chip square measure shown to be possible and simple to scale for supporting an oversized range of A multicore system during which processors communicate along through a Router Architecture is shown in Fig. 1. Every router has 5 ports that hook up with four neighboring routers and their native processor. A network interface (NI) locates between a processor and its router for remodeling processor messages into packets to be transferred on the network and contrariwise. In a typical router, every input port has an associate input buffer for quickly storing the packets just in case that output channel is busy. This buffer may be one queue as in a warm hole (WH) router or multiple queues in parallel as in virtual channel (VC) routers. These buffers, in fact, consume important parts of space and power that may be over an hour of the total router. Buffer less routers take away buffers from the router thus save abundant space; but, their performance becomes poor just in case packet injection rates square measure high. As a result of having no buffers, previous router styles projected to drop and convey packets or to deflect them once network rivalry happens that may consume even higher energy per packet than a router with buffers. Another approach is by sharing buffer queues that permits utilizing idle buffers or emulating associate degree output buffer router to get higher outturn. Our work differs from those router styles by permitting input packets at input ports to bypass shared queues therefore, it achieves lower zero-load latency. Additionally, the projected router design has easy management electronic equipment creating it dissipate less packet energy than VC routers and achieving higher outturn by property queues share workloads once the network load becomes significant. process components instead of point-to-point interconnect wires or shared buses.

LITERATURE SURVEY

Peh et al. and Mullins et al. projected speculative techniques for VC routers permitting a packet to at the same time intercede for each VCA and militia giving the next priority for non speculative packets to win SA; so, reducing zero load latency during which

the chance of failing speculation is little. This low latency, however, comes with the high complexness of militia electronic equipment and additionally wastes additional power anytime the speculation fails. Recently, Passas et al. designed a 128 × 128 crossbar permitting connecting 128 tiles whereas occupying only 6 June 1944 of their total space. This reality encourages North American country to make RoShaQ that has two crossbars whereas sharing cost expensive buffer queues. **Nicopoulos et al.** projected ViChar, a router design that permits packets to share flit slots within buffer queue so can-do higher output. Our paper manages buffers at coarser grain that\'s at queue-level instead of at flit-level, thence permits reusing existing generic queue style that makes buffer and router style abundant less complicated and easy. Ramanujam et al. recently planned a router design with shared-queues named DSB that emulates associate output-buffered router. the bulk of progressive on-chip router styles utilize input queuing buffers; we tend to, however, will notice within the literature many output queuing router architectures. If applicable to the complete network image, buffers at associate output router port ought to act a similar as input buffers of its downstream router.

3.1 Cartesian Network Initialization

In Cartesian routing, each arterial issues Arterial This Way (ATW) management packets throughout its data format method. An ATW tells the receiving collector router if an blood vessel is accessible through the incoming port. An ATW conjointly specifies what quite affiliation is accessible via the incoming port: north, south, north and south or neither. Upon receiving an ATW, every collector router updates its Arterial Direction Indicator (ADI) and forwards the ATW to the alternative port. ATWs also are accustomed establish Virtual Arterials, created in things wherever it is physically not possible for an arterial to span two collectors. The ADI points within the direction of the arterial router (i.e., east or west) and indicates whether or not the arterial router contains a affiliation to the north, the south, or both. Figure one illustrates a philosopher network.

3.2 Cartesian Routing

Packets will arrive on either a west or east port of a collector router. Packets meant for a special latitude are forwarded out to the other port from that they are received. The ADI determines the packet's initial direction on the collector router once a packet arrives on very cheap port of a collector router. Decide a packet's initial direction, the router initial compares the packet's destination address with its own address. The packet is going to be forwarded within the direction of the destination if the destination latitude is the same because the collectors. The packet is forwarded within the direction of the ADI if the destination is on a completely different latitude..

3.3 Wide Areas Cartesian Networks

A Cartesian network provides an easy topological structure that relieves collector routers from the necessity to keep up routing tables. However, it would be unreal to implement one worldwide Cartesian network. Such a widespread Cartesian network, for instance, needs each packet destined for a router with a similar latitude symbol because the supply router's latitude symbol goes to all the collector routers. It is additionally necessary for such a network to possess one collector for each doable latitude. These limitations recommend that implementing one worldwide Cartesian network would be impractical. An alternative to a worldwide Cartesian network is to make a collection of smaller Cartesian networks and implement a mechanism for interchanging packets between them. One approach to interchanging packets between Cartesian networks is to forward packets towards their destinations.

When a packet reaches the boundary of a network it "falls off" the sting and is delivered to a special router to be forwarded towards the destination address. the method of routing a packet from one network to a different victimization this approach becomes problematic once networks area unit interleaved or overlapped. Two networks are said to be overlapped if there is at least one collector router on one of the networks where its longitude identifier lies between the longitude identifiers of two collectors from the other network and all three of them share the same latitude identifier.

An alternative methodology for delivering a packet to its destination is to seek out the destination network address then to route the packet to the destination network by exploitation mathematician routing algorithms. This implies that every network should be specifiable exploitation the packet's destination address. If we tend to assume that every network features a rectangular form, recognizing the destination network could be a matter of examination the packet's destination address with the network's boundaries. However, there are a range of reasons to assume that it would be chimerical to expect networks to have rectangular borders: geographical barriers and political jurisdictions, as an example. Since Cartesian routing uses latitude and longitude pairs to spot the source and therefore the destination addresses of packets, this data is not spare to see to that network a collector or arterial belongs within the case of interleaved and overlapped networks. This, in turn, suggests that a further set of data is needed to spot to that network a collector or blood vessel is connected. To achieve this, the authors propose a hierarchical data structure for Cartesian networks. Within the next section the likelihood of multiple-layer Cartesian networks as an answer for interchanging packets between arbitrary shaped interleaved and overlapped Cartesian networks are explained. Within the remainder of this paper, the terms "wide area Cartesian networks" and "multiple-layer Cartesian networks" are used interchangeably.

RESEARCH METHODOLOGY

The methodology section outline the plan and method that how the study is conducted. This includes Universe of the study, sample of the study,Data and Sources of Data, study's variables and analytical framework. The detailsare as follows;

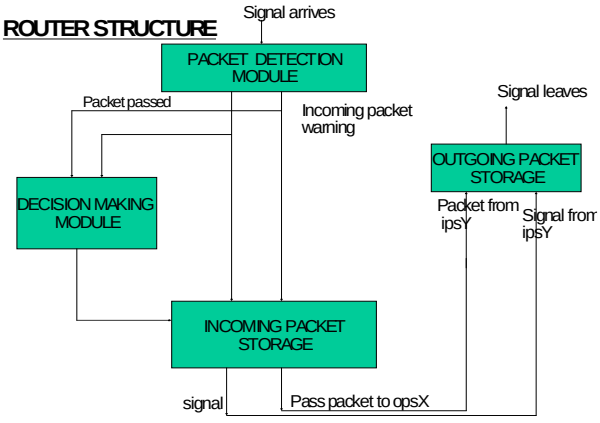


Fig. 1 Router Architecture

3.4 Shared Buffer Architecture

RoShaQ, a router design with shared queues supported the concept of the above figure. Once the associate input port receives a packet, it calculates its output port for the next router (look ahead routing), at constant time it arbitrates for each its set output port and shared queues. If it receives a grant from the output port allocators (OPAs), it will advance to its output port within the next cycle. Otherwise, if it receives a grant to a shared queue, it can be written to that shared queue at the next cycle. Just in case that it receives each grant, it will grade to advance to the output port. Shared-queues authority (SQA) receives requests from all input queues and grants permission to their packets for accessing non full shared queues.

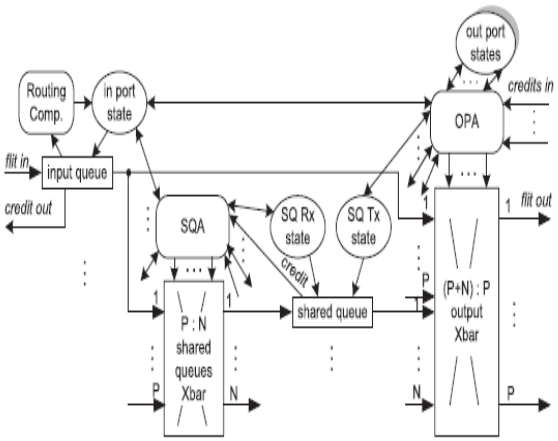


Fig. 2 Shared Buffer Architecture

Packets from input queues area unit allowed to write down to a share queue solely if: 1) the shared queue is empty or 2) the shared queue is containing packets having constant output port because the requesting packet shown in Fig 2. The OPA receives requests from each input queue and shared queues. Both SQA and OPA grant these requests in a round-robin manner to ensure fairness and conjointly 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 regulate the operation of the router. Solely input queues of RoShaQ have routing computation logic because of packets within the shared queues were written from input queues therefore, they have already got their output port data.

3.5 Weighted Routing Algorithm

To provide information measure guarantees in AdNoC, the underlying communication infrastructure wants to offer AN reconciling route allocation theme motivated from the reconciling routing schemes for massive scale networks. During a physically static operation, the routing call may be distributed, or a source-based settled routing theme is also utilized. During a distributed settled routing theme, the routing call is decided domestically at every router victimization predefined rule, e.g., XY-routing algorithmic program within the QNoC design. The source-based settled routing theme (e.g., Xpipe keeps the complete route in the header of dealings packets and wants the world read of the whole chip before execution or even at style time. that\'s why both schemes do not seem to be suitable for the AdNoC architecture where the subset of tasks and their mapping may change during runtime. For a requesting dealing, the route is checked in each doable direction. The weighted XY-routing (wXY-routing) algorithmic rule conferred in Fig. four assigns every output port a weight supported on the market information measure and dx and x coordinate (columns) distance or dysprosium, the y coordinate (rows) distance between this and therefore the destination node. This ideally provides the packet with a most range of smart routing selections on its route because it permits the packet to be routed toward its destination in each the x and y directions. The load is additionally proportional to the on the market information measure. If the output port is chosen with the best associated on the market information measure, the used information measure is distributed as equally as possible among the output ports. Thus, the other output ports are more likely to be able to accommodate future

transactions. By permitting each value to contribute to the load, the load becomes a exchange between these two issues. The weight of each port is given as:

$$w_N = \begin{cases} b_N \times |y_d - y| + b_{\max}, & y_d - y < 0 \\ 0, & b_N < b_p \\ b_N, & \text{else} \end{cases}$$

$$w_E = \begin{cases} b_E \times (x_d - x) + b_{\max}, & x_d - x > 0 \\ 0, & b_E < b_p \\ b_E, & \text{else} \end{cases}$$

$$w_S = \begin{cases} b_S \times (y_d - y) + b_{\max}, & y_d - y > 0 \\ 0, & b_S < b_p \\ b_S, & \text{else} \end{cases}$$

$$w_W = \begin{cases} b_W \times |x_d - x| + b_{\max}, & x_d - x < 0 \\ 0, & b_W < b_p \\ b_W, & \text{else} \end{cases}$$

They are calculated to be proportional to the space from supply to destination and to the on the market information measure if the output direction is facing the destination, and proportional to the on the market information measure if it is not. If there is not enough information measure on the market, the weights square measure zero. The route chosen is then to the direction with the very best weight.

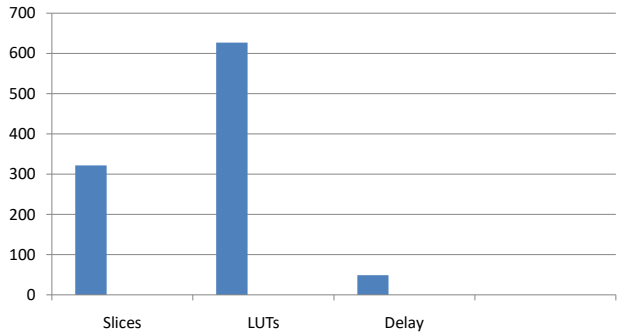
IV. RESULTS AND DISCUSSION

4.1 Results of Hardware Utilization

Table 4.1: **Hardware Utilization**

Parameter	Estimated Results
Gate Count	4464
Slices	322
LUTs	627
Delay	23.21ns

Graphical Illustration Results:



I. ACKNOWLEDGMENT

A router architecture that allows sharing multiple buffer queues for improving network throughput. Input packets also can bypass the shared queues to achieve low latency in the case that the network load was low. It had also higher throughput than a full-crossbar VC router with 3% lower power and less area. While targeting the same average packet latency of 100 cycles where all the routers start saturating, RoShaQ had 9% and 7% lower energy dissipated per packet than typical VC and full-crossbar VC routers, respectively. The comparison between virtual channel and shared buffer queue can be observed for various parameters such as power and latency.

REFERENCES

[[1] J. L. Hennessy and D. A. Patterson, *Computer Architecture: A Quantitative Approach*. San Francisco, CA, USA: Morgan Kaufmann, 2007.

[2] S. Borkar, "Thousand core chips: A technology perspective," in *Proc. 44th DAC*, Jun. 2007, pp. 746–749.
[3] C. H. V. Berkel, "Multi-core for mobile phones," in *Proc. DATE*, 2009, pp. 1260–1265.
[4] W. J. Dally and B. Towles, "Route packets, not wires: On-chip interconnection networks," in *Proc. DAC*, 2001, pp. 684–689.
[5] W. J. Dally, "Virtual-channel flow control," *IEEE Trans. Parallel Distrib. Syst.*, vol. 3, no. 2, pp. 194–205, Mar. 1992.
[6] Y. Hoskote, S. Vangal, A. Singh, N. Borkar, and S. Borkar, "A 5-GHz mesh interconnect for a teraflops processor," *IEEE Micro*, vol. 27, no. 5, pp. 51–61, Sep. 2007.
[7] T. Moscibroda and O. Mutlu, "A case for bufferless routing in on-chip networks," in *Proc. ISCA*, Jun. 2009, pp. 196–207.
[8] M. Hayenga, N. E. Jerger, and M. Lipasti, "SCARAB: A single cycle adaptive routing and bufferless network," in *Proc. 42nd Ann. IEEE/ACM Int. Symp. Microarchitect.*, Dec. 2009, pp. 244–254.
[9] G. Michelogiannakis, D. Sanchez, W. J. Dally, and C. Kozyrakis, "Evaluating bufferless flow control for on-chip networks," in *Proc. 4th NOCS*, 2010, pp. 9–16.
[10] 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.