Five Port Router Architecture

E. Nagaraju[1] A. Umma Maheswari[2]
[1]M. Tech(VLSI) [2]Associate Professor
[1, 2]Department of Electronics and Communication Engineering
[1, 2]Siddharatha Institute of Engineering and Technology

Abstract—In this paper we attempt to give a networking solution by applying VLSI architecture techniques to router design for networking systems to provide intelligent control over the network. Networking routers to have limited input/output configurations, which we attempt to overcome by adopting bridging loops to reduce the latency and security concerns. Other techniques we explore include the use of multiple protocols. We attempt to overcome the security and latency issues with protocol switching technique embedded in the router engine itself. The approach is based on hardware coding to reduce the impact of latency issues as the hardware itself is designed according to the need. We attempt to provide a multipurpose networking router by means of Verilog code, thus we can maintain the same switching speed with more security we embed the packet storage buffer on chip and generate the code as self-independent VLSI based router. Our main focus is the implementation of hardware IP .router. The approach enables the router to process multiple incoming IP packets with different versions of protocols simultaneously, e.g. for IPv4 and IPv6. The approach will results in increased switching speed of routing per packet for both current trend protocols, which we believe would result inconsiderable enhancement in networking systems.

Key words: router, packets, FPGA, RTL, IP

I. INTRODUCTION

Our approach here is to design a variable hard router code by using Verilog and the same to be implemented for the SOC (system on chip) level router. In this paper we are making a VLSI design for the implementation at the synthesizable level the same can be further enhanced to SOC level, but our main aim is limited to t6he netlist generation level which would give the result prediction and workable module vision. Our focus being in this is to make this router as much variable as we can which will give the Robust router in which we can make the same router to not only go for N number of connection but also to detect all variety of packets and route the same. To do so we have to add the code with specific case's for every type of packets we want to add to our router to route, with this pa[per of hardware code our approach is to get the basic packets routing with multiple protocols starting with the IPv4andIPv6.

II. LITERATURE SURVEY

In this we are comparing the existing generic router architecture and our new robust router architecture. This will give the difference in the designing and would reflect our paper enhancements that we are upgrading in our robust router paper.

A. Generic Router Architecture

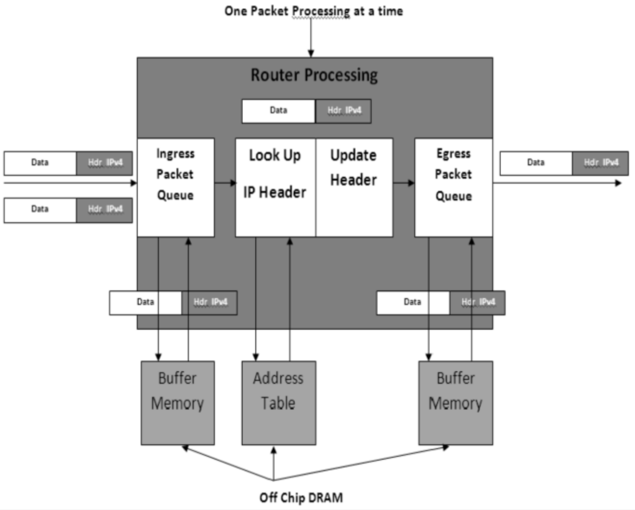


Fig. 1: Generic architecture

In the architecture we can look that the generic architecture is processing a single packet of a specific protocol at a given time and the output queue buffer also one for one egress channel ring by which there is the overloading of the queue and will result in the congestion. The congestion flow is as shown below.

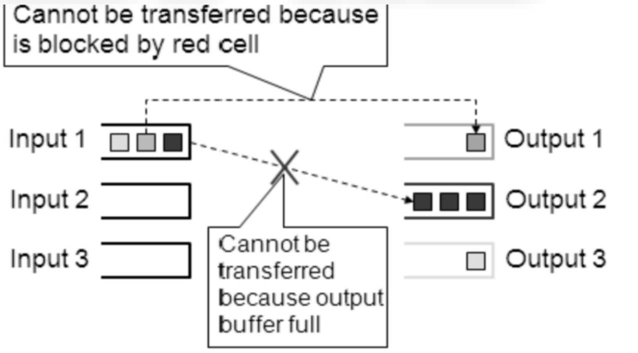

Cannot be transferred because is blocked by red cell
Input 1
Input 2
Input 3
Output 1
Output 2
Output 3
Cannot be transferred because output buffer full

Fig. 2: Congestion Flow

III. A NEW FIVE PORT ROUTER ARCHITECTURE

The architecture of five port router is totally based on the Verilog code which would enable our design in the implementation of parallel packet processing for N number of channels. The intern enables the multi packet processing at the same time. With the Verilog code being the base of design we have an option for the addition of protocol case and respective look-up table makes us go for the multi-protocol processing at the same time. With the Verilog code being the base of design we have an option for addition of protocol case and respective look-up table makes us go for the multi-protocol processing at the same time. By which we are unable to provide the multi-packet multi-protocol routing at the same time with same speed.

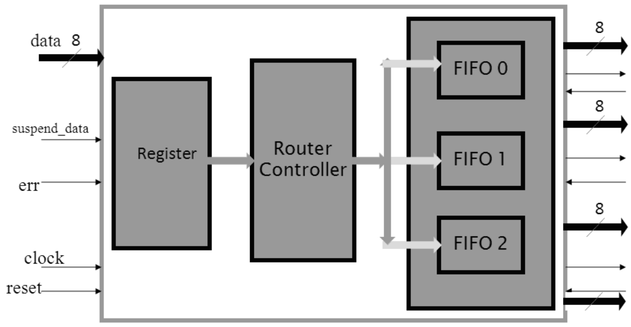


Fig. 3: Five Port Router

While designing the robust router a special concern is kept in the mind of the switching speed issue to give the maximum speed with parallelism being added. The engress output buffer queuing problem was also solved by providing a separate queue for every ingress channel in the egress channel with N vertical queue by which we can avoid the congestion to a remarkable level which is as shown below.

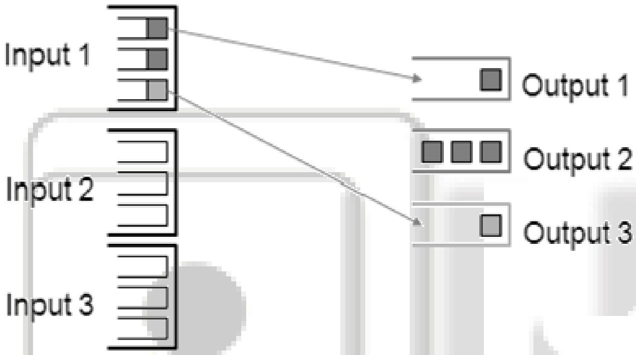


Fig. 4: Congestion Flow with Vertical Queue

The size issue is another special feature of our robust router which makes our robust router a unique system. As discussed earlier in the paper we are trying to make the robust router on to the chip level design so we further advance it to the level of Ethernet based router which will make the router to be implemented on the standalone systems, which will be a revolutionary enhancement in size matter from room full of router to just the PCI slot operating Router and will make network work more faster. It looks something like this below.

Generally, the router can be interfaced with n numbers of I/O devices. Block diagram given below

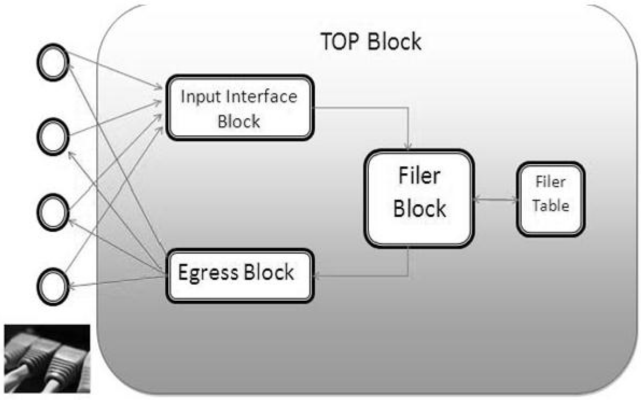


Fig. 5: Router can be Inter faced with N Numbers

The paper design has the following modules.

A. Input Interface Block

This block is mainly responsible receiving the incoming IP packets over multiple input channels. This block asserts the necessary response signals in order to communicate with the IP packet driver modules. After receiving the IP packets, this block forward the same to the ingress block for the further process. This block forwards the same to store block as well as parser block.

B. Packet Store Block

This is responsible for storing the error free received packets. This module receives the packet contents from ingress block and dispatches the same based on the request from Egress block.

C. Parser Block

This block is mainly responsible for parsing the complete packet into multiple set of data according to its field. Parsed contents will be inputted to the filer block. The above three blocks are merged all together as IIB in code to single file.

D. Filer Block

This block is responsible for selecting the egress ring. The block receives the parsed data from the parser block. The parsed data will be forwarded to the filer table. In response to this, the filer table provides the output ring number. Then, the received output from the filer table will be forwarded to the egress block.

E. Filer Table

It is a user configurable table. This table contains a set of data in its each slot, against which the data send by the filer block will be compared. If the filer block inputted data matches with the data of any slot of filer table, then that slot's data will be used as egress ring through which received packet will be forwarded.

F. Egress Block

This block receives the data from filer block as egress ring number through which the received packet shall be forwarded. Upon receiving the egress ring number, this block initiates the communication with packet store block to fetch the packet to be forwarded to the output interface block with the output channel details, over which the packet has to be transmitted.

IV. SYSTEM FLOW DIAGRAM

The system flow diagram is as shown below which makes us to understand the flow of the signals through the system from each block by block and transaction carried between the blocks to accomplish the task of the robust router. The flow diagram described here is a brief one, which helps us to understand the flow of every block. Every block have the state machines cycle included in them to enhance the system logical transaction to the level of parallelism. The flow diagram is as shown in fig. 6.

First the packet is received from the ingress channel ring to the input interface block the packet is parsed to data packet and header packet, the data packet is storied in the parser queue and the header is sent to the filer block. The filer block then checks weather the packet is

IPv4orIPv6 and accordingly send the request to the filer table to router the packet to required channel with it dest-IP address and send the egress ID to the filer block. The block sends and enables the particular egress ring in egree blocks and gives the command to the particular egress ring in egress block. Then in egress block the block stored data packet in the parser queue is added back with header and us sent out with the specified egress ring channel. In this way the every packet is processed and routed in robust router.

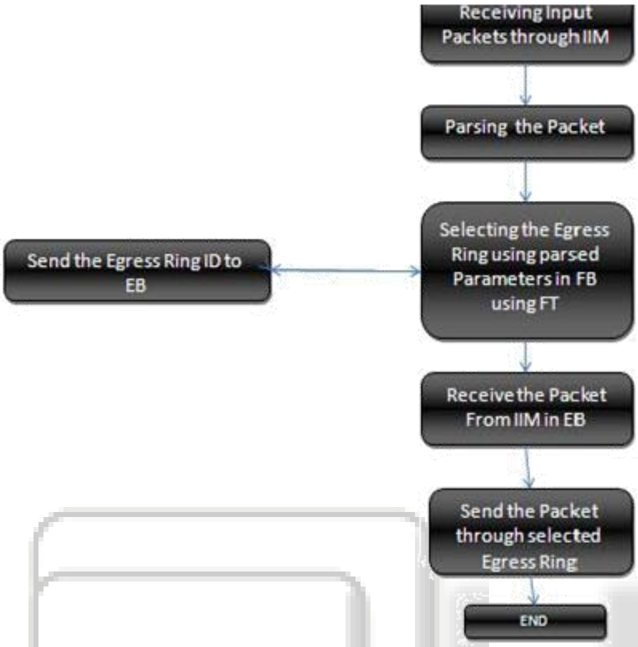

Receiving Input Packets through IIM
Parsing the Packet
Selecting the Egress Ring using parsed Parameters in FB using FT
Send the Egress Ring ID to EB
Receive the Packet From IIM in EB
Send the Packet through selected Egress Ring
END

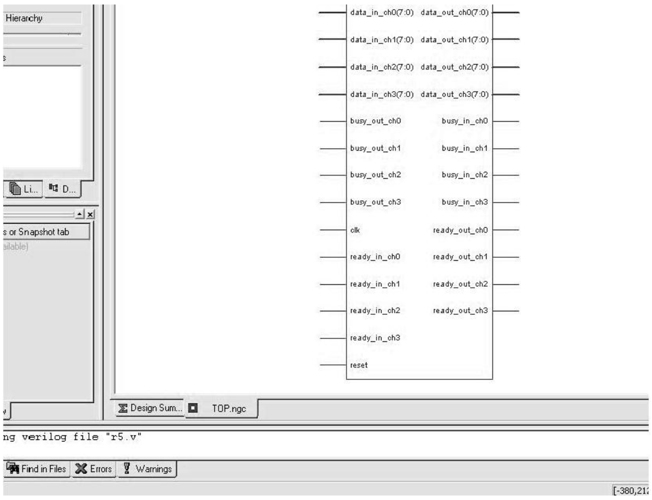
Fig. 6: System Flow Diagram

V. SIMULATION AND DISCUSSION

A. *Net list of the Robust Router*

The net list is a RTL level of the robust router system, which is syntasizable and can be extracted on the XILINX tool. By which we can get preface look of the system and a transition from the frontend of the VLSI designing to backend of the VLSI designing. Which means the same can run on FPGA kit and test its robustness end errors of the system can be debugged before it is taken to SOC level and to Fab-Labs.

The snap below is the pin configuration of the proposed robust router



ng verilog file "r5.v"

Fig. 7: Net list

The net list level can further go after I/O padding get the exact pin configuration which can be derived accordingly but will be similar one.

The snap below is the top level system view of the system at the RTL.

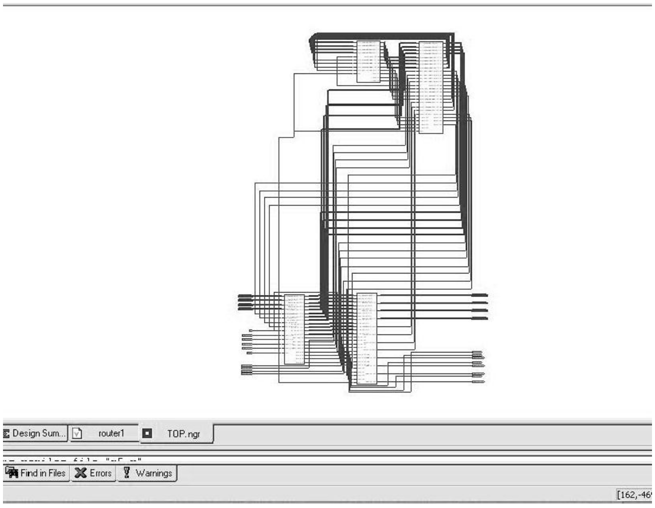


Fig. 8: view of the system at the RTL

B. *SOC Designing*

The further moments of the VLSI designing will require the sharp knowledge of the VLSI backend designing and can be fabricated at the 45 nano technology using the cadence encounter tool which will enable us to take the system to SOC level the steps are as shown below.

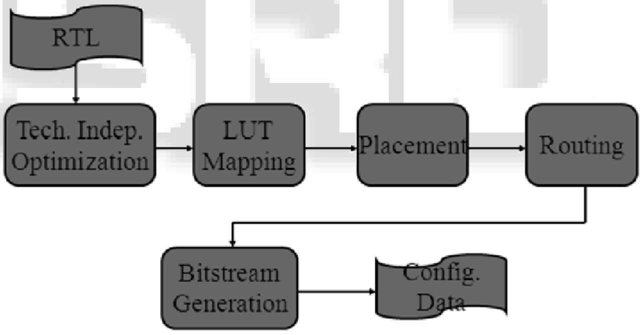


Fig. 9: system to SOC level the steps

The RTL level of design which we get from the Net list of the system will have gate delay and wire delays included in them. These are all calculated and made in to an optimization level. Then the design is fixed into LUT'S and the mapped between the LUT'S are prissily done keeping mind the power utilization and the delay calculated earlier. Then the routing is done between the CLB'S. Further the bit-stream is generated to list output to get the exact design. Then the system design is masked and made to the GDSSI level further to be sent on the Fab-Labs for fabrication.

VI. CONCLUSION

We summarize the advantages and applications below.

Advantages

- General purpose router
- Router hardware code is variable
- High switching speed with any no. of i/o pin connection
- More secured

- Robust router can handle all type of packets

Application

- Can be used as public internetworking router
- Can be used as corporate router
- Software company private router
- Router for networking research
- In other words one point networking solution

System Test Analysis

- Number of gates – 59855
- Total runtime- 9.018nsw
- Switching speed- 662.5Tbps
- System frequency- 0.11GHZ
- Can be fabricated- by 45nano technology with cadence Encounter tool (cost 12000 dollars)
- The same Verilog code design can be taken to the implementation of MPLS (Multi-protocol Label Switching).
- The some code design can be taken to the SOC level and can be implemented as the Ethernet standalone system router.
- The same code can be made variable with TCP and UDP protocols.

REFERENCES

[1] Channamallikarjuna Mattihallii 2. Suprith Ron 3. Naveen Klaa " Four port router architecture: ELECTRICAL AND COMPUTER ENGINEERING Debre Berhan University, Ethiopia 2012 third international conference on intelligent system on modeling and simulation

[2] Y. Katsube, K. Nagami, and H. Esaki, "Toshiba's Router Architecture Extensions for ATM: Overview," IETF RFC2098, April 1997.

[3] Y. Rekhter, B. Davie, D. Katz, E. Rosen, and G. Swallow, "Cisco Systems' Tag Switching Architecture Overview, "IETF RFC 2105, Feb. 1997. Amir. Palnitkar. 2nd Edition

[4] J. Moy, OSPF: Anatomy of an Internet Routing Protocol, 1998

[5] Tobias Bjerregaard and Shankar Mahadevan. A survey of research and practices of network-on-chip. ACM Comput.Surv, 38(1):1, 2006.

[6] Charles .H. Roth, Jr. Digital System Design Using VHDL

[7] Jenkins, Jesse H. Designing with FPGAs and CPLDs

[8] Abromovici, M, Breuer Digital System Testing and Testable Design.

[9] James Balfour and William J. Dally. Design tradeoffs fortiled cmp on-chip networks. In ICS '06: Proceedings of the20th annual international conference on Supercomputing, pages 187–198, 2006.