

IMPLEMENTATION OF HARDWARE IP ROUTER BASED ON VLSI

¹ KARLAPUDI SRAVANI, ² P T BALAKRISHNA

¹(M.Tech) VLSI, Dept. of ECE

²Associate Professor, Dept. of ECE

Priyadarshini Institute of Technology & Science for Women

Abstract:- A Network-on-chip is a new paradigm in complex system-on-chip designs that provide efficient on chip communication networks. It allows scalable communication and allows decoupling of communication and computation. The data is routed through the networks in terms of packets. We attempt to overcome latency and time reduction issue and can provide multipurpose networking router by means of Verilog and it was synthesized in Xilinx 13.2 version, simulated Modelsim 10.0 version. In this paper our attempt is to provide a multipurpose networking router by means of Verilog code, by this we can maintain the same switching speed with more secured way of approach we have even the packet storage buffer on chip being generated by code in our design in the so we call this as the self-independent router called as the VLSI Based router. The three architectures were analyzed for their performance in terms of delay, throughput and latency and we concluded that CDMA router performs better than the other two.

Keywords: Fictional Coverage, assertions, Randomization, Network-On-Chip,, Register blocks.

1. INTRODUCTION

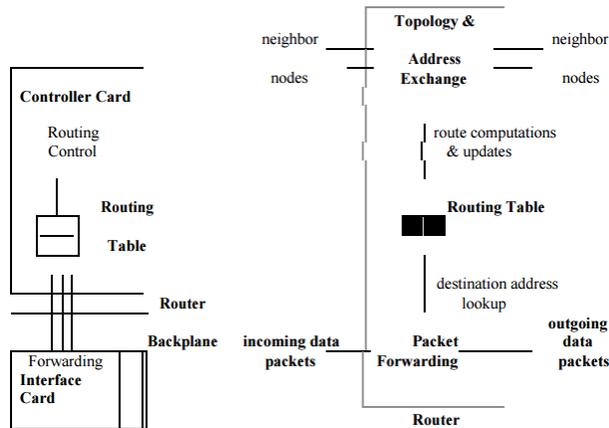
The popularity of the Internet has caused the traffic on the Internet to grow drastically every year for the last several years. It has also spurred the emergence of many Internet Service Providers (ISPs). Our approach here is to design a variable hardware router code by using Verilog and the same to 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 the 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 robustness for the design to be called even as a Robust Router in which we can make the same router to not only go for N number of connections but also to detect all variety of packets and route the same. To do so we have to add the code with specific cases for every type of packets we want to add to our router to route, with this paper of hardware code

our approach is to get the basic packets routing with multiple protocols starting with the IPv4 and IPv6.

Basic IP Router Functionalities

Generally, routers consist of the following basic components: several network interfaces to the attached networks, processing module(s), buffering module(s), and an internal interconnection unit (or switch fabric). Typically, packets are received at an inbound network interface, processed by the processing module and, possibly, stored in the buffering module. Then, they are forwarded through the internal interconnection unit to the outbound interface that transmits them on the next hop on the journey to their final destination. The aggregate packet rate of all attached network interfaces needs to be processed, buffered and relayed. Therefore, the processing and memory modules may be replicated either fully or partially on the network interfaces to allow for concurrent

operations. A generic architecture of an IP router is given the basic architecture of a typical router:



The router special services. The two key functionalities are route processing (i.e., path computation, routing table maintenance, and reachability propagation) and packet forwarding. We discuss the three functionalities in more detail below.

A New Robust Router Architecture

The architecture of robust 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. This 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. By which we are unable to provide the multi-packet multi-protocol routing at the same time with same speed. 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' number of I/O devices. Block diagram given below

NETWORK ON CHIP (NoC)

NoC is a technology that is intended to solve the short coming of buses. It is an approach to design the communication subsystem between intellectual property cores in a SoC design. The communication strategy in system on chip uses dedicated buses between communicating resources. This will not give any flexibility for the needs of the communication, in each case, have to be thought of every time a design is made. Another possibility is the use of common buses, which have the problem that it does not scale very well, as the number of resources grows. NoC is intended to solve the shortcomings of these, by implementing a communication network of switches/micro routers and resources [7,8].

The NoC design paradigm has been proposed as the future of ASIC design [9]. NoC design space is considerably larger when compared to a bus based solution, as different routing and arbitration strategies can be implemented as well as different organizations of the communication infrastructure. The NoC paradigm is highly suited to provide SoC platforms scalable and adaptable over several technology generations. NoC platforms may allow the design productivity to grow as fast as technology capabilities and may eventually close the design productivity gap [12]. In addition, NoCs have an inherent redundancy that helps tolerate faults and deal with communication bottlenecks [13].

2. A NEW ROBUST ROUTER ARCHITECTURE

The architecture of robust 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. This 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. By which we are unable to provide the multi-packet multi-protocol routing at the same time with same speed. 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 egress 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.

2.1 OPERATION

The Five Port Router Design is done by using of the three blocks. The blocks are 8-Bit Register, Router Controller and output block. The router controller is design by using FSM design and the output block consists of four FIFO's combined together. The FIFO's store data packets and when you want to send data that time the data will read from the FIFO's. In this router design has four outputs i.e. 8-Bit size and one 8-bit data port. The ROUTER can operate with a single master device and with one or more slave devices. If a single slave device is used, the RE (read enable) pin may be fixed to logic low if the slave permits it. Some slaves require the falling edge (HIGH→LOW transition) of the slave select to initiate an action such as the mobile operators, which starts conversion on said transition.

2.2 THREE PORT ROUTER ARCHITECTURE

The Four Router Design is done by using of the three blocks. The blocks are 8-Bit Register, Router controller and output block.

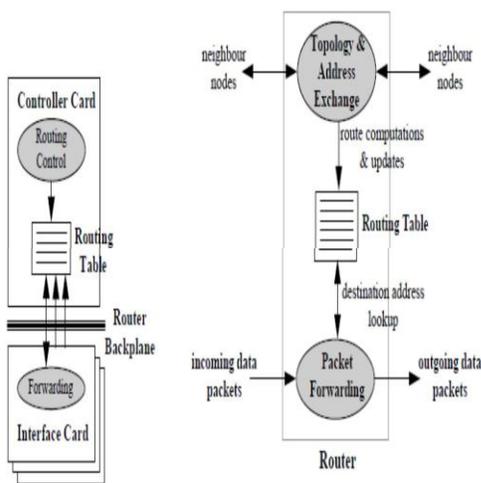


Fig. 1 Three port router architecture

The router controller is design by using FSM design and the output block consists of three FIFO'S combined together the FIFO'S are store packet of data and when u want to data that time the data read from the FIFO's.

In this router design has three outputs that is 8-Bit size and one 8-bit data port it using to drive the data into router we are using the global clock and reset signals, and the err signal and suspended data signals are output's of the router .The CPU in the router typically performs such functions as path computations, routing table maintenance, and reach ability propagation. It runs which ever routing protocols are needed in the router. The interface cards consist of adapters that perform inbound and outbound packet forwarding (and may even cache routing table entries or have extensive packet processing capabilities). The router backplane is responsible for transferring packets between the cards. The basic functionalities in an IP 4 router[6] can be categorized as: route processing, packet forwarding, and router special services.

3. IP PACKET VALIDATION

The router must check that the received packet is properly formed for the protocol before it proceeds with protocol processing. This involves checking the version number, checking the header length field (also needed to determine whether any options are present in the packet), and calculating the header checksum.

4.1 Destination IP Address Parsing and Table Lookup:

The router performs a table lookup to determine the output port onto which to direct the packet and the next hop to which to send the packet along this route. This is based on the destination IP address in the received packet and the subnet mask(s) of the associated table entries. The result of this lookup could imply: A local delivery (that is, the destination address is one of the router's local addresses and the packet is locally delivered). A unicast delivery to a single output port, either to a next-hop router or to the ultimate destination station (in the case of a direct connection to the destination network[5]).

4.2 Packet Lifetime Control

The router adjusts the time-to-live (TTL)[9] field in the packet used to prevent packets from circulating endlessly throughout the internetwork. A packet being delivered to a local address within the router is acceptable if it has any positive value of TTL.

4.3 Checksum Calculation

The IP header checksum must be recalculated due to the change in the TTL[10] field. Fortunately, the checksum algorithm employed (a 16-bit one's complement addition of the header fields) is both commutative and associative, thereby allowing simple, differential recompilation.

5. 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 machine cycle included in them to enhance the system logical transaction to the level of parallelism.

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 stored in the parser queue and the header is sent to the filer block. The filer block then checks weather the packet is IPv4 or IPv6 and accordingly send the request to the filer table to router the packet to required destination. The filer table cross verifies the egress ring channel with it Dest-IP address and send the egress ring ID to the filer block.

The filer block send and enables the particular egress ring ingress blocks[11] and gives the command to the particular egress ring in egress block. Then in egress block the stored data packet in the parser queue is added back with header and is sent out with the specified egress ring channel. In this way the every packet is processed and routed in robust router.

6. VERIFICATION METHODOLOGIES

Verification is not a test bench, nor is it a series of test benches. Verification is a process used to demonstrate that the intent of a design is preserved in its implementation. In this chapter, I introduce the basic concepts of verification, from its importance and cost, to making sure you are verifying that you are implementing what you want.

6.1 TEST BENCH ARCHITECTURE

A collection of all test specifications for a given area. The Test Plan contains a high-level overview of what is tested and what is tested by others for the given feature area. The test plan is implemented using OVM test component. Generally the method to write a test case is to define a base test in which all the environment is instantiated and all the common configurations used for all the test cases are written as

6.2 Control Test Case

Control test case is mainly targeted for the Master section[14] of the Router. Its purpose is to check whether the device is responding to the standard request or not and to check the control endpoints of the device. It generates random standard requests to the Router devices and look for the response for the device. It also checks the operation of the endpoint 0 controller in the device

6.3. Data Receive Test Case

This test case is used to verify the functionality of the master receiver of the Router device. This test case first configures the device for use by using standard requests, issues the command to receive the data from the external i2c device and finally retrieves the data from the master receiver FIFO by using data read commands. It uses standard sequences, write register sequence, data read sequence from the sequence library.

6.4. Slave Data Transmit Test Case

This test case is to verify the slave transmitter of the Router section, but any way this test case should make sure some data in slave transmit FIFO. So the test case will first calls standard commands to set the device up , then it has to write some data on the slave transmit FIFO and finally slave is ready to transmit the data.

6.5. Slave Data Receive Test Case

This test case is used to verify the functionality of the slave receiver of the Router device. This test case first configures the device for use by using standard requests, slave receives the data from the external i2c device and stores it in the slave receiver FIFO. Then agent reads the FIFO by using data reads commands. It uses standard sequences, write register sequence, data read sequence from the sequence library.

7. RESULTS & COVERAGE REPORTS

7.1. Code Coverage

In this paper the code given in the output for IPv4 and IPv6 packets is put in the router at the same time. The Robust Router will route both packets at the same time at the same speed. The Robustness is simulated with Model-Sim Tool with different Test Cases and the same code's NetList is extracted with Xilinx Tool for the synthesizable code. The same can be taken to the SOC (System on chip) level with Cadences Encounter Tool. 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 (System on chip) level and can be implemented as the Ethernet Standalone System Router. The same code can be made variable with TCP and UDP Protocols.

8. SIMULATION AND DISCUSSION

8.1 Net List of the Robust Router

The Net List is 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 and 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. The RTL level of design which we get from the Net List of the system will have gate delay, propagation delay and wire delays included in them. These are all calculated and made into an optimization level. Then the design is fixed into LUT'S and the mapped between the LUT'S further the placement of

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 bitstream is generated to test the system and verification done across the Net List output to get the exact design. Then the system design is masked and made to the GDSSI Level further to be sent on to the Fab-Labs for fabrication.

8.2 Design under Test

The design under test [DUT] is made to test the system robustness under different cases. The DUT architecture Includes the Test case which will define the test. The input driver block will generate the test input signals for the system testing. The input and output transacted will make the system get the input and output according to the system core requirement.

9. CONCLUSION

In this paper the code given in the output for IPv4 and IPv6 packets is put in the router at the same time. The Robust Router will route both packets at the same time at the same speed. The Robustness is simulated with ModelSim Tool with different Test Cases and the same code's NetList is extracted with Xilinx Tool for the synthesizable code. The same can be taken to the SOC (System on chip) level with Cadences Encounter Tool. 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 (System on chip) 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 Mattihalli, Suprith Ron,Naveen Kolla " VLSI Based Robust Router Architecture" Third International Conference on Intelligent Systems Modelling and Simulation,2012.
- [2]Feng Liang, Luwen Zhang, Shaochong Lei, Guohe Zhang, Kaile Gao, and Bin Liang" Test Patterns of Multiple SIC Vectors: Theory and Application in BIST Schemes" iee transactions on very large scale integration (vlsi) systems, vol. 21, no. 4, april 2013.

- [3] James Aweya "IP Router Architectures: An Overview".
- [4] M. SOWMYA C. SHIREESHA G. SWETHA PRAKASH J. PATIL "VLSI Based Robust Router Architecture" .
- [5] M. Waldvogel, G. Varghese, J. Turner, and B. Plattner, "Scalable High Speed IP Routing Lookup," Proc. ACM SIGCOMM'97, Cannes, France, Sept. 1997.
- [6] V. Srinivasan and G. Varghese, "Faster IP Lookups using Controlled Prefix Expansion," Proc. ACM SIGMETRICS, May 1998.
- [7] S. Nilsson and G. Karlsson, "Fast Address Look-Up for Internet Routers," Proc. Of IEEE Broadband Communications'98, April 1998.
- [8] E. Filippi, V. Innocenti, and V. Vercellone, "Address Lookup Solutions for Gigabit Switch/Router," Proc. Globecom'98, Sydney, Australia, Nov. 1998.
- [9] M. Thottethodi, A. R. Lebeck, and S. S. Mukherjee, "BLAM: a high performance routing algorithm for virtual cut-through networks," in Proceedings of the International Parallel and Distributed Processing Symposium
- [10] L. S. Peh and W. J. Dally, "A delay model and speculative architecture for pipelined routers," in Proceedings of the 7th International Symposium on High Performance Computer Architecture (HPCA).
- [11] A. V. Yakovlev, A. M. Koelmans, and L. Lavagno, "High-level modeling and design of asynchronous interface logic," IEEE Design & Test of Computers.
- [12] H. Jingcao and R. Marculescu, "Energy- and performance aware mapping for regular NoC architectures," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 24, pp. 551-562, 2005.
- [13] W. Hangsheng, L. S. Peh, and S. Malik, "A technology-aware and energy oriented topology exploration for on-chip Networks," in Proceedings of the Design, Automation and Test in Europe Conference and Exhibition .
- [14] J. Duato, "A new theory of deadlock-free adaptive routing in wormhole networks," IEEE Transactions on Parallel and Distributed Systems, vol. 4.
- [15] S. Arjun, W. J. Dally, A. K. Gupta, and B. Towles, "GOAL: a load-balanced adaptive routing algorithm for torus networks," in Proceedings of the International Symposium on Computer Architecture.