RTL Design & Logic Synthesis of 1×3 Router in 130nm Technology

Aboli Moharil Head of the Department (SLRTCE)[1], Singh Utkarsh Dinesh (Student, SLRTCE)[2],Raorane Krunal Vijay, SLRTCE)[3], Yadav Pradeep Jungbahadur (Student,SLRTCE)[4], Yadav Durgesh Amarjeet (Student, SLRTC)[5]

[1]Department of Electronics and Telecommunications, SLRTCE, Thane-401107, India[2]Department of Electronics and Telecommunications, SLRTCE, Thane-401107, India[3]Department of Electronics and Telecommunications, SLRTCE, Thane-401107, India[4]Department of Electronics and Telecommunications, SLRTCE, Thane-401107, India[5]Department of Electronics and Telecommunications, SLRTCE,Thane-401107, India

ABSTRACT

Routing the process of delivering a packet of data from a source to a destination, allowing messages to flow from one computer to another until they reach their target. A router is a computer networking device that facilitates the transfer of data packets from one network to another. It has two or more data lines from separate networks attached to it (as opposed to a network switch, which connects data lines from one single network). The top-level design of the router device, as well as how sub-modules like Register, FIFO, and FSM work, and Synchronizer are synthesised, simulated, and finally coupled to the top module are all examined in this research. There are 3 output ports to send the packet. The packet is divided into three sections. The three pieces are the header, data, and frame check sequence. The length of the packet to be sent varies between 1 and 63 bytes, with an 8-bit packet width. Packets are routed to the proper ports by the switch based on their destination addresses. Each output has an own 8-bit port address. If the packet's destination address matches the port address, the switch sends it to the output port.; the data length is 8 bits. In this suggested study, the Xilinx ISE IDE Tool is employed for synthesis and simulation. The FSM in the proposed architecture has a less number of states, due to reduction of states the amount of time to produce the response became less obviously the frequency is improved.

1.INTRODUCTION

Computer networking is an important area of study. It is essential for many industries and applications.

Computer networking is necessary for communication, data transfer, and remote access, and is critical for industries such as finance, healthcare, and education.

Verilog RTL is a hardware description language that allows designers to describe the behavior of digital circuits and verify their functionality before they are physically implemented.

Goal is to design a low-cost, efficient, and reliable router that can handle high volumes of network traffic, which is important for many industries and applications.

My Project could help to improve the reliability and efficiency of network communication, which is essential for many industries and applications, and could also have implications for future developments in computer networking and digital circuit design.

2.OBJECTIVE

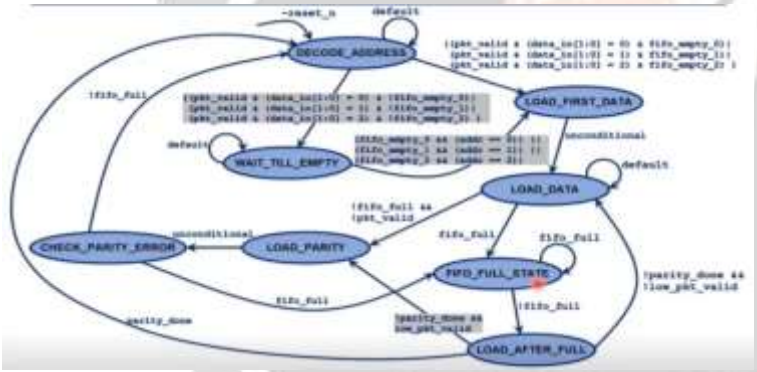
The objective of the project was to design a Verilog RTL router capable of routing data packets from a single source network to three different client networks. To design a Verilog RTL router capable of routing data packets from a single source network to three different client networks. To develop a register module that can hold data packets momentarily before passing them on to three different FIFO memories. To implement a Finite State Machine (FSM) that can manipulate the internal signals to perform the necessary tasks for routing the data packets. To design a synchronizer that can ensure proper timing and synchronization of signals within the router. To simulate the Verilog RTL router using Xilinx ISE and ISIM to ensure its functionality and performance. To perform code coverage analysis on the design using ISim. To verify the router's functionality through testing in a UVM-based environment.

3.LITERATURE SURVEY

• Performance analysis is usually done by simulation at the Transaction Level - TL or at the Register Transfer Level - RTL. RTL provides smaller simulation time, while RTL provides more accuracy in results, which is very necessary during the design phase of a NoC. However, the performance characterization of a large NoCs by means of RTL simulation is time costly and requires several hours of computing. The performance evaluation of a NoC can be also speeded up by doing it directly on hardware as FPGA instead of using simulation models.

• In his 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 today 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 as we embed the packet storage buffer on chip and generate the code as a 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 in considerable enhancement in networking systems.

4.FLOWCHART OF FSM



STATE DECODE_ADDRESS

• This is the initial reset state.
• Signal detect_add is asserted in this state which is used to detect an incoming packet. It is also used to latch the first byte as a header byte. STATE-LOAD_FIRST_DATA
• Signal lfd_state is asserted in this state which is used to load the first data byte to the FIFO. Signal busy is also asserted in this state so that header byte that is already latched doesn‟t update to a new value for the current packet.
• This state is changed to LAOD_DATA state unconditionally in the next clock cycle. STATE-LOAD_DATA
• In this state the signal ld_state is asserted which is used to load the payload data to the FIFO.
• Signal busy is de asserted in this state, so that ROUTER can receive the new data from input source every clock cycle,
• Signal write_enb_reg is asserted in this state in order to write the Packet information (Header+Payload+Parity) to the selected FIFO.
• This state transits to LAOD_PARITY state when pkt_valid goes low and to FIFO_FULL_STATE when FIFO is full. STATE-LOAD_PARITY
• In this state the last byte is latched which is the parity byte.
• It goes unconditionally to the state CHECK_PARITY_ERROR.
Signal busy is asserted so that ROUTER doesn‟t accepts any new data • write_enb_reg is made high for latching the parity byte to FIFO. STATEFIFO_FULL_STATE
• Busy signal is made high and write_enb_reg signal is made low.

• Signal full_state is asserted which detects the FIFO full state. STATELOAD_AFTER_FULL
• In this state lfd_state signal is asserted which is used to latch the data after FIFO_FULL_STATE.
• Signal busy & write_enb_reg is asserted.
• It checks for parity_done signal and if it is high, shows that LOAD_PARITY state has been detected and it goes to the state DECODE_ADDRESS.
• If low_packet_valid is high it goes to LOAD_PARITY state otherwise it goes back to the LOAD_DATA state. STATE-WAIT_TILL_EMPTY
• Busy signal is made high and write_enb_reg signal is made low. STATECHECK_PARITY_ERROR
• In this state rst_int_reg signal is generated, which is used to reset low_packet_valid signal.
• This state changes to DECODE_ADDRESS when FIFO is not full and to FIFO_FULL_STATE when FIFO is full.
• Busy is asserted in this state.

The Result of the project is as follows:-

Output snapshot:

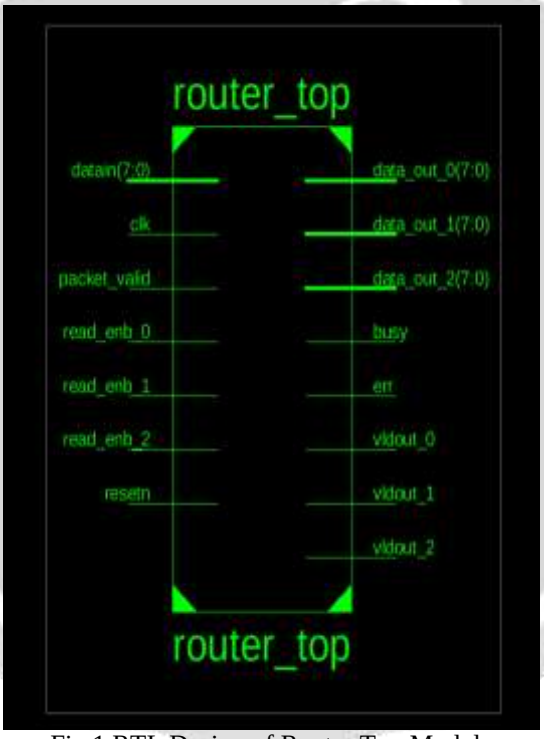


Fig.1 RTL Design of Router Top Module

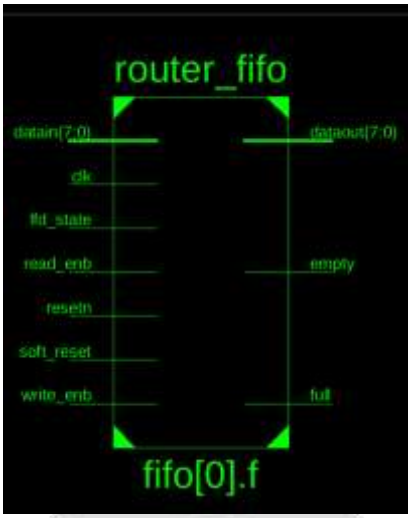


Fig.2 RTL Design of FIFO Module.

Fig.3 RTL Design of Synchronizer Module

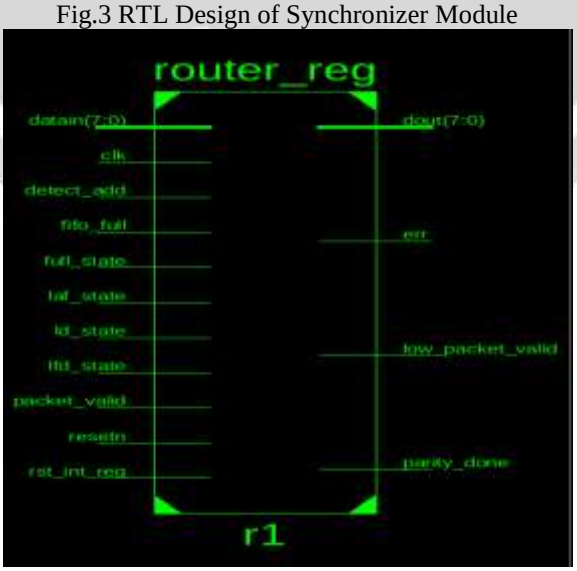


Fig.4 RTL Design of Register Module.

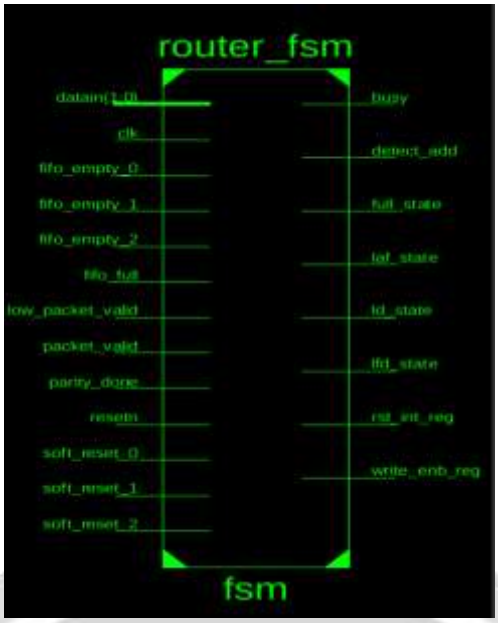


Fig.5 RTL Design of Finite State Machine Module

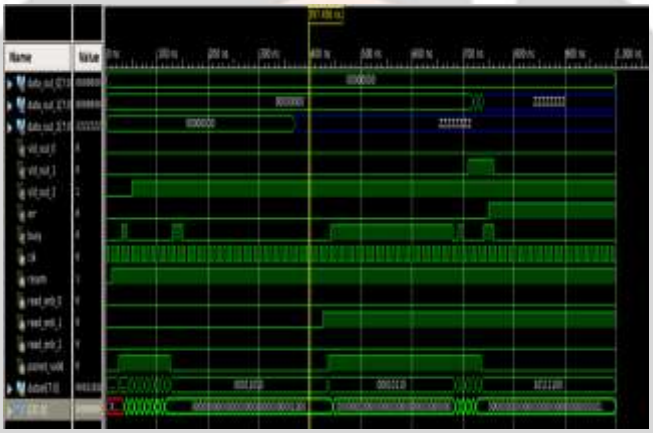


Fig.6 TestBench Output of Router

5.SOFTWARE AND SYSTEM

Xilinx 14.7

Xilinx ISE (Integrated Synthesis Environment) 14.7 is a software suite used for designing and testing Field Programmable Gate Array (FPGA) and Complex Programmable Logic Device (CPLD) circuits. It is developed by Xilinx, Inc., a leading provider of programmable logic devices and related software tools.

Xilinx ISE 14.7 is the final release of the ISE design suite and includes a wide range of features and tools for FPGA and CPLD design. Some of the key features of ISE 14.7 include:

1. Design entry: ISE 14.7 provides a range of design entry methods, including schematics, text editors, and graphical design entry tools.
2. Synthesis: The software includes a high-quality synthesis engine that can optimize and transform a design description into an optimized netlist.
3. Simulation: ISE 14.7 includes a powerful simulator that can simulate and verify designs before they are implemented in hardware.
4. Implementation: The software suite includes advanced placement and routing algorithms that can optimize designs for specific FPGA devices and target technologies.
5. Timing analysis: ISE 14.7 includes tools to perform static timing analysis (STA) and to generate timing constraints for a design.
6. Debugging: The software provides an extensive range of debugging tools, including the ability to view waveforms, signal values, and other design information.

Overall, Xilinx ISE 14.7 is a comprehensive software suite for FPGA and CPLD design and testing. It provides a range of tools and features that can be used throughout the entire design process, from initial design entry to final implementation and verification. The software has been widely used by FPGA and CPLD designers for many years

and is a popular choice for many in the field.

DESCRIPTION

This design consists of 6 main blocks. Which are fsm_router, router_reg, ff_sync, and 3 fifo. The fsm_router block provides the control signals to the fifo, and router_reg module.

1) Router_reg: The router_reg module contains the status, data and parity registers for the router_1x3.These registers are latched to new status or input data through the control signals provided by the fsm_router.This module contains status, data and parity registers required by router.All the registers in this module are latched on rising edge of the clock.Data registers latches the data from data input based on state and status control signals, and this latched data is sent thefifoor storage. Apart from it, data is also latched into the parity registers for parity calculation and it is compared with the parity byte of the packet. An error signal is generated if packet parity is not equal to the calculated parity.

2) FF_sync: Theff_sync module provides synchronization between fsm_router module and 3 fifos, so that single input port can faithfully communicate with 3 output ports.

This module provides synchronization between fsm and fifo modules. It provides faithful communication between single input port and three output ports. It will detect the address of channel and will latch it till packet_valid is asserted, address and write_enb_sel will be used for latching the incoming data into the fifo of that particular channel.Afifo_full output signal is generated, when the present fifo is full, and fifo_empty output signal is generated by the present fifo when it is empty. If data = 00 then fifo_empty = empty_0 and fifo_full = full_0 If data = 01 then fifo_empty = empty_1 and fifo_full= full_1 If data = 10 then fifo_empty = empty_2 and fifo_full = full_2 Else fifo_empty = 0 and fifo_full = 1. The output vld_out signal is generated when empty of present fifo goes low, that means present fifo is ready to read. vld_out_0 = ~empty_0 vld_out_1 = ~empty_1vld_out_2 = ~empty_2The write_enb_reg signal which comes from the fsm is used to generatewrite_enb signal for thepresentfifo which is selected by present address.

3) FSM:The 'fsm_router' module is the controller circuit for the router.This module generates all the control signals when new packet is sent to router. These control signals are used by other modules to send data at output, writing data into the fifo.

4) FIFO Block:There are 3 fifo for each output port, which stores the data coming from input port based on the control signals provided by fsm_routermodule.fresetn is low then full =0, empty = 1 and data_out = 0. Write operation: The data from input data_in is sampled at rising edge of the clock when input write_enb is high and fifo is not full. Read Operation: The data is read from output data_out at rising edge of the clock, when read_enb is high and fifo is not empty. read and Write operation can be done simultaneously. Full – it indicates that all the locations inside fifo have been written.

6.Conclusion

The project successfully achieved its objectives by synthesizing and simulating the router device using Xilinx ISE IDE Tool. The implementation of the Finite State Machine with a reduced number of states resulted in improved response time and frequency. The router device was capable of routing data packets to three different output ports based on their destination addresses. However, some limitations and challenges were encountered during the project, such as the need for careful testing and verification to ensure the correct functioning of the router device. Further research and improvements can be made to enhance the functionality and efficiency of the router device, such as incorporating security features

7.ACKNOWLEDGEMENT

A general expression of thanks towards the project sponsor or funding agency Thanks to the faculty or advisor who provided guidance and support throughout the project Thanks to the colleagues, friends, or family who provided encouragement and assistance Acknowledgement of any institutions or individuals who provided technical or logistical support Mention of any publications or sources that were helpful in the project A brief statement of personal growth or learning during the project. This paper and the research behind it would not have been possible without the exceptional support of our HOD, Prof. Aboli Moharil. Her enthusiasm, knowledge and exacting attention to detail have been an inspiration and kept my work on track.

8.REFERENCES

[1] M. Pirretti, G. M. Link, R. R. Brooks, N. Vijaykrishnan, M. Kandemir, and M. J. Irwin - IEEE Computer society Annual Symposium on VLSI paper on Fault tolerant algorithms for network-on-chip interconnect - year **[2004]**

[2] Cota, É., de Morais Amory, A., & Lubaszew$ki, M. S. - NoC Basics paper on In Reliability, Availability and Serviceability of Networks-on-Chip - year **[2012]**

[3] Swapna, S., Swain, A. K., & Mahapatra, K. K. - Asia Pacific Conference on Postgraduate Research IEEE paper on Design and analysis of f5 port router for network on chip. In Microelectronics and Electronics - year **[2012]**

[4] Choudhari, E. M., & Dakhole, P. K. - IEEE paper on Design and verification of five port router for network on chip. In Communications and Signal Processing (ICCSP) - year **[2014]**

[5] Tatas, K., Siozios, K., Soudris, D., & Jantsch, A. - NY Springer paper on Designing 2D and 3D network-on-chip architectures – year [

[6] Gopal N. - International Journal of Engineering Research and Development paper on Router 1x3 RTL Design and Verification – year **[2015]**

[7] Oveis-Gharan, M., & Khan, G. N. - In VLSI (ISVLSI), 2015 IEEE paper on Index-based round-robin arbiter for NoC routers – year **[2015]**

[8] Mrudula, Y., Harshavardhan, V., Manu, G., & Godavarthi, B. - International Journal of Science and Applied Information Technology (IJSAIT) paper on Design and Verification of Four Port Router for Network – year **[2016]**

[9] Wang, L., Ma, S., Li, C., Chen, W., & Wang, Z. - IEEE paper on A high performance reliable NoC router. Integration, the VLSI – year **[2016]**

[10] Salminen, E., Kulmala, A., & Hamalainen, T. D. - White paper, OCP-IP paper on Survey of network-on-chip proposals – year **[2008]**