Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 06, June-2019

Design and Implementation of Four Port Router for Network on Chip

Dr. Uma B V
Professor, Department of ECE,
R V College of Engineering Bengaluru-560059, India

Raghav Pakala
Student, MTech Department of ECE,
R V College of Engineering Bengaluru-560059, India,

Abstract – **The integration of several cores has been an emerging trend paving way towards integration of a number of systems on a single chip. The Network on chip provides an efficient method for integration and also establishes a communication framework for interaction between the various cores present on the chip. Routers and Switches form a very important part of the Network on chip which facilitate the interaction between the IP cores. In this study we introduce an Network on Chip Router which makes use of optimized FIFO Based Buffers. The Buffers determine the performance of a Network on Chip as they consume significant amount of area and power. The Router uses X-Y Routing protocol and Round Robin Arbitration. The design is modelled using Verilog HDL in Xilinx Vivado Design Suite. The simulations show that there is a reduction in area as compared to the existing router architecture.**

Keywords – Network on Chip, Router, scheduler, Arbitrer, First in First Out.

I. INTRODUCTION

Network on a chip provides a path for flow of messages from a particular source to destination through various links. The decisions are made at the intermediate switches for routing. The Network on Chip can be considered as homogeneous network and is also highly scalable for Multiprocessor System on Chips. Network on Chip can be helpful to,

- Simplify the hardware needed for routing.
- To improve the scalability of the design.
- To Reduce Power Consumption in Complex Designs.
- To operate at much higher frequencies.
- To eliminate the issues relating to synchronization.

The basic architecture of a Network on Chip is made up of routers, Processing Elements and Network Interfaces as the fundamental components. A network interface bridges the core of the router to the various router ports. A Network Interface is mainly made up of a Reorder Unit, a Depacketizer Unit and Packetizer Unit. A Packetizer Unit creates the packet using burst data which is present in input buffer and then the packet is transferred. A Processing Element can be processor or memories. Fig 1 shows the block diagram of the complete network on chip.

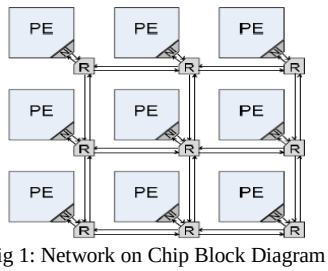

Fig 1: Network on Chip Block Diagram

A router forms the backbone of a Network on Chip system and is built as per the OSI model. A Router is responsible for efficient transfer of packets from the source node to destination node. The router is required to have a high throughput and controls the flow of data in the network. The router is made up of an input port, a switching matrix which links the input to the output and an output port. The routers determine the destination and the path to be traversed on receiving an incoming packet through the input port. The router architecture chosen for implementation identifies the latency that will be incurred in the network and the critical path delay for the implementation. It is essential to design a router efficiently in order to ensure the performance of the network.

The aim of this study is to develop an efficient network on chip router to interconnect the various processing elements and to establish a communication network.

II. ROUTING ALGORITHM

Routing mainly refers to forwarding of packets from source to destination in the network. An efficient routing mechanism is essential as it affects the power consumption, timely delivery of packets and the amount of congestion present in the network.

A. XY Routing

The XY routing model requires the packets to initially traverse along the X direction then along the Y direction in order to arrive at the destination.

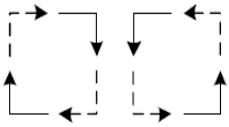

Fig 2: Valid XY Routing Turns

Fig 2 show the set of turns that are valid for XY Algorithm for traversing to the next hop router towards the destination. The routers that adopt the XY routing have four outputs namely east, west, south and north which connect to the next neighboring routers and one local port which is used for providing an external input to the router. The position of a router in a Network on Chip is determined by a (x, y) coordinate in a two-dimensional network with mesh topology. In XY Routing, the address of the current router is compared with that of the destination router obtained from the header of the packet. The Packets with the address of the current router equal to that of the destination router are sent to the router core. The source Router is represented by the coordinates (Ax, Ay) and the destination router is represented by the coordinates (Bx, By). The coordinates (Cx, Cy) represent the current router. The XY Routing algorithm can be given as

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 06, June-2019

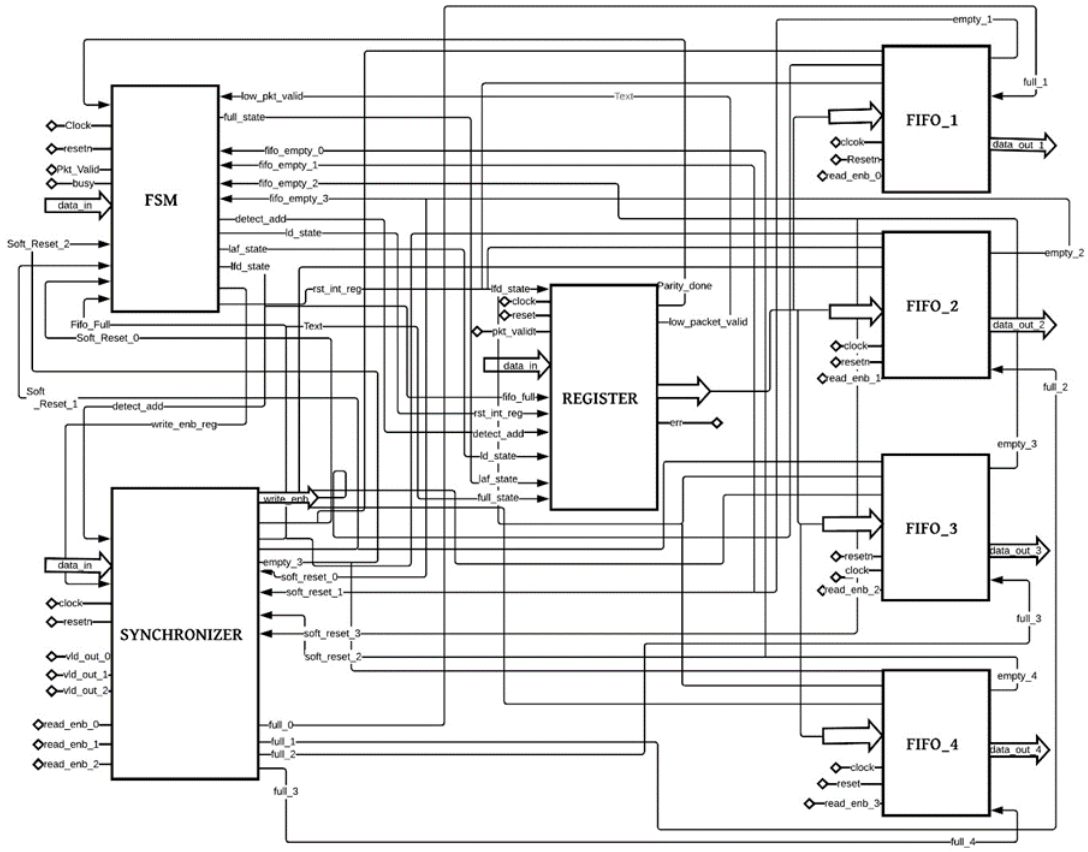

Figure 3: NoC Router Block Diagram

/*XY Routing Algorithm*/
begin
if (B_x>C_x)
return East;
else
if (B_x<C_x)
return West;
else
if (B_x=C_x) {
if (B_y<C_y)
return South;
else
if (B_y>C_y)
return North;
else
if (B_y=C_y) //current router is the destination router
return C;
}
end

Here, C_x<B_x implies that the packets have to be sent to the next hop router through the east port, C_x>B_x implies towards West Port and C_x=B_x implies the packet aligned with the destination router along the horizontal direction and has to traverse in the vertical direction. Now, the address of the router along the vertical direction, B_y is compared with the current router C_y. Packets will be routed to South when C_y<B_y and to North when C_y>B_y. The XY algorithm helps to decide the router port through which the packet traverses in order to reach next hop router towards the destination.

III. RELATED WORK

A FIFO based router is mainly made up of Schedulers, Registers and Demultiplexers. The register stores the data in the buffer and a demux unit which sends the incoming data to the appropriate output. The First in First out module performs the read and write operations when the FIFO is full and empty respectively. The empty state of the FIFO module is determined by the empty flag and the full state is determined by the full flag. The flags are made high in order to identify the state of the FIFO module.

The scheduler unit commonly uses the Round Robin algorithm which allows the packets to repeat in a periodic manner and considers each of the data in the queue equally. The least priority is given to the current instance in the next iteration for scheduling.

The Router implements store and forward mechanism in which the entire packet is stored in the buffer of each router. Each router in the network is required to have a buffer size sufficient to hold the incoming packet. The packet is forwarded to the next hop router only when it has the space to hold the entire packet.

Fig 3. represents the designed router block diagram. Router components refers to the building blocks of the router. Let's investigate each component individually.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 06, June-2019

A. FIFO

A First in First out memory allows for storing of data. The data is arranged in the form of a queue with a control logic

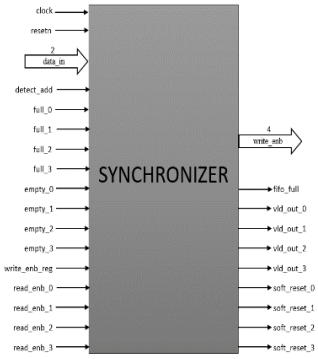

Figure 4: Synchronizer Module

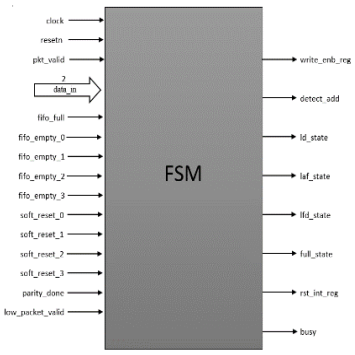

Figure 5: FSM Module

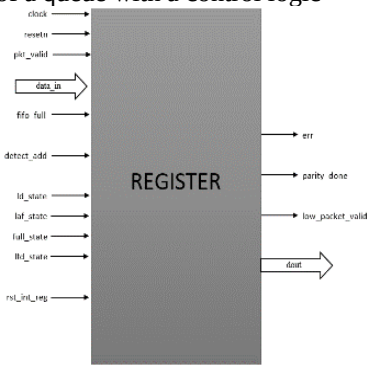

Figure 6: Register Module

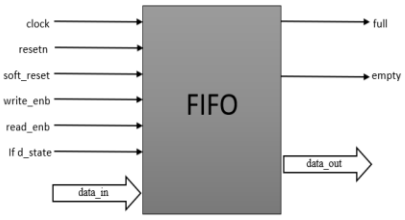

Figure 7: FIFO Module

that manages read and write operations. The width of the FIFO is 32 bit and the depth of the memory array is 16 bytes. The data from the output ports is stored in the FIFO. There are two signals, full and empty which indicate the full and empty states. A clock signal synchronizes the activities of the FIFO module. Fig 7. represents the FIFO module. When the reset is low, the signals are set to their default values. The write operation is carried out when the write enable is made high and the fifo is not full and the read operation is carried out when the read enable is made high and the fifo is not empty. A signal lfd_state detects the incoming packet and loads the first data from the packet. A soft reset signal is also provided which helps to reset the FIFO module when the timeout condition of the module is encountered.

B. FSM

The various control signals required for the functioning of the router are generated by the FSM module. Fig 5. represents the FSM Module. The various states of the FSM are described below.

The first state of the FSM is the Decode_Address. When the detect_add signal goes high, the router detects the arrival of the incoming packet and latches the first byte of the packet i.e. header byte.

The second state in the FSM is Load_First_Data which loads the first data byte from the incoming packet. When the lfd_state signal goes high and the busy signal is made high, first data byte is loaded into the FIFO. In this state the FIFO is empty and the data_in signal determines the next hop router to which the data is to be loaded.

The third state is the Load_Data state in which when the ld_state signal goes high; the entire payload data is being copied to the next hop router FIFO. When the busy signal is low and the write_enb_reg signal is made high, the Packet information is written to FIFO. After the payload data is copied, the next state for the router is LOAD_PARITY if the pkt_valid signal is low and FIFO_FULL_STATE if the full signal of the router is high.

The fourth state is the Load_Parity state in which the Parity byte is being latched into the next hop router FIFO and is also stored in the Register module of the router for the parity computation. When the busy signal is made high and the write_enb_reg is made high, the parity byte is written to FIFO.

The fifth state is the Fifo_Full_State which is attained when the FIFO is full. When the Busy signal is made high, the write_enb_reg signal is made low and the full_state signal is made high the FIFO full state is detected. The router continues to stay in this state as long as the FIFO is full. The next state is the Check_Parity_Error state in which the parity is calculated by performing XOR operation on the payload data present in the packet. This calculated parity value is stored in the register module. The calculated parity is compared with the value in the packet to determine if there is any error in the received packet. When the rst_int_reg is made high in this state, the low_packet_valid signal is made low and the busy signal is asserted.

C. Register

Fig 6. represents the Register Module. The router has four internal registers. The header of a packet is stored and used until the entire packet is transferred to the next hop router. The second register holds the parity value calculated from the received payload data. The parity is calculated as the XOR operation of the current value of parity_reg and the header_byte and the result are stored back in parity_reg. The other two registers hold the parity byte from the packet and the FIFO full state byte.

D. Synchronizer

The module provides synchronization between FSM and router

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 06, June-2019

FIFO modules. It provides faithful communication between the single input port and three output ports. Fig 4. shows the synchronizer block. The detect_add detects the incoming packet. The data_in is used to determine the output port. The full, empty signals form an input which is used to generate the fifo_full signal. The write_enb_reg signal generates signal for write operation and vld_out signal is made high when the incoming packet is valid and does not have any errors.

IV. RESULTS AND DISCUSSION

The results obtained using the developed Network on Chip router is described below. Fig 8 gives the FIFO module Simulation. The synthesized FIFO module is made up of 716 Cells and 2902 Nets.

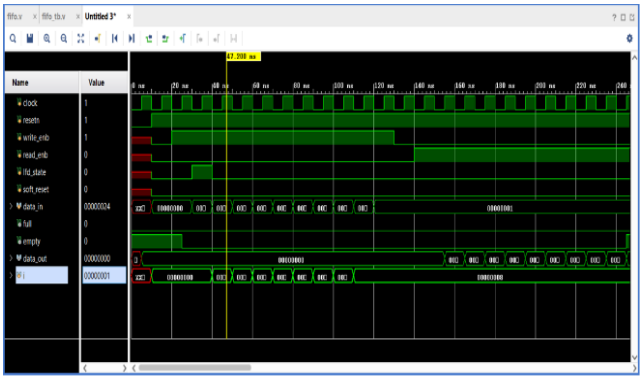

Figure 8: FIFO Module Verilog Simulation

Fig 9 gives the FSM module Simulation. The synthesized Synchronizer module is made up of 63 Cells and 120 Nets.

Figure 9: FSM Module Verilog Simulation

Fig 10 gives the Register module Simulation. The synthesized module is made up of 204 Cells and 404 Nets.

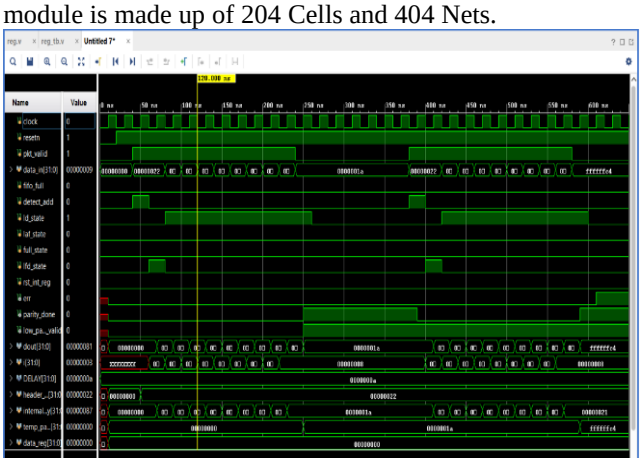

Figure 10: Register Module Verilog Simulation

Fig 11 gives the Simulation of the Router Top Module. The router module consists of FSM, REGISTER, SYNCHRONIZER, FIFO. During the design of each sub-module one by one individually using RTL coding in Verilog The reset signal is active low while all the other signals at the input are active high. All the signals at the input are synchronized to falling edge of the clock. This is done as the Design Under Test router is sensitive to the rising edge of the clock. Therefore, in the testbench, driving input signals on the falling edge ensures setup and hold time.

The packet_valid signal is asserted on the same clock edge when the header byte is driven onto the input data bus. The header byte contains the address of the output channel to which the packet should be routed to (data_out_0, data_out_1, data_out_2, data_out_3). The header byte followed by the payload are driven on the input data bus for every new falling edge of the clock. After the last payload byte has been driven, on the next falling edge of the clock, the packet_valid signal must be de-asserted, and the packet parity should be driven. This signals packet completion. The testbench shouldn`t drive any byte when busy signal is detected instead it should hold the last driven values. The `busy` signal when asserted drops any incoming byte of the data. The “err” signal is asserted when a packet parity mismatch is detected.

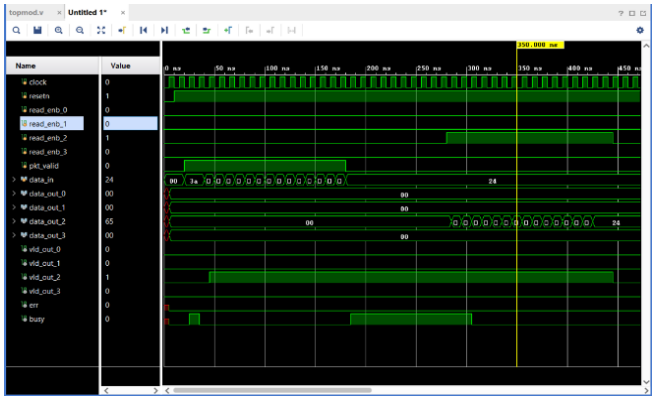

Figure 11: Router Verilog Simulation

Table 1: Resource Utilization of Implemented Router Module

RESOURCE	USED	AVAILABLE	PERCENTAGE UTILIZATION
Slices	984	2600	37.84
LUTs	3328	10400	32
Bonded IOBs	78	106	73.5
GCLK	1	24	4

Table 1 provides the Resources that are utilized for the implementation of the Router Top Module. The design is synthesized on Xilinx Artix 7 FPGA.

V. CONCLUSION AND FUTURE SCOPE

A FIFO Buffer based Network on Chip router has been developed in order to improve the performance of the On-Chip communication. The FIFO based router has 15% lesser area than the standard router implementation and a considerable amount of area is because of the large FIFO buffers present which are optimized for an efficient implementation.

The router can be enhanced further using Virtual Channels and the various routing algorithms which help to solve the problem of deadlock and improve the performance.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 06, June-2019

The router can be used to develop a mesh-based Network on Chip which can provide effective communication between the IP modules of the chip.

REFERENCES

[1] N. V. Anjali and K. Somasundaram, "Design and evaluation of virtual channel router for mesh-of-grid based NoC," 2014 International Conference on Electronics and Communication Systems (ICECS), Coimbatore, 2014, pp. 1-5.

[2] H. El-Sayed, M. Ragab, M. S. Sayed and V. Goulart, "Hardware implementation and evaluation of the Flexible router architecture for NoCs," *2013 IEEE 20th International Conference on Electronics, Circuits, and Systems (ICECS)*, Abu Dhabi, 2013, pp. 621-624.

[3] Feiyang Liu, Huaxi Gu and Yintang Yang, "Performance study of virtual-channel router for Network-on-Chip," *2010 International Conference on Computer Design and Applications*, Qinhuangdao, 2010, pp. V5-255-V5-259.

[4] S. M. Hassan and S. Yalamanchili, "Centralized buffer router: A low latency, low power router for high radix NOCs," *2013 Seventh IEEE/ACM International Symposium on Networks-on-Chip (NoCS)*, Tempe, AZ, 2013, pp. 1-8.

[5] Kiran and K. Solanki, "Design of efficient NOC router for chip multiprocessor," *2016 International Conference on Inventive Computation Technologies (ICICT)*, Coimbatore, 2016, pp. 1-4.

[6] M. Kumar, K. Kumar, S. K. Gupta and Y. Kumar, "FPGA based design of area efficient router architecture for Network on Chip (NoC)," *2016 International Conference on Computing, Communication and Automation (ICCCA)*, Noida, 2016, pp. 1600-1605.

[7] R. Louis, M. Vinodhini and N. S. Murty, "Reliable router architecture with elastic buffer for NoC architecture," *2015 International Conference on VLSI Systems, Architecture, Technology and Applications (VLSI-SATA)*, Bangalore, 2015, pp. 1-4.

[8] S. Malipatil and Rekha S, "Design and analysis of 10 port routers for network on chip (NoC)," *2015 International Conference on Pervasive Computing (ICPC)*, Pune, 2015, pp. 1-3. E.

[9] N. Nasirian and M. Bayoumi, "Low-latency power-efficient adaptive router design for network-on-chip," *2015 28th IEEE International System-on-Chip Conference (SOCC)*, Beijing, 2015, pp. 287-291.

[10] Qing Sun, Lijun Zhang and Anding Du, "Design and implementation of the wormhole virtual channel NoC router," 2015 4th International Conference on Computer Science and Network Technology (ICCSNT), Harbin, 2015, pp. 854-858.

[11] S. Shenbagavalli and S. Karthikeyan, "An efficient low power NoC router architecture design," *2015 Online International Conference on Green Engineering and Technologies (IC-GET)*, Coimbatore, 2015, pp. 1-8.