

DESIGN OF A VLSI ROUTER FOR THE FASTER DATA TRANSMISSION USING BUFFER

Mr. V.V.G.S Rajendra Prasad¹

¹²³⁴Ugschoolar, Associate professor¹

P PENCHALA SATWIK², G. SUNIL³, C. SIVA⁴, M. KOTABABU³

Department of Electronics and Communication Engineering

R K college of Engineering

Vijayawada, India

venuguntav@gmail.com, sivagowdchennu@gmail.com, sathwikppp@gmail.com,

sunilgoug734@gmail.com, m.chintu9550@gmail.com

Abstract: The design and implementation of efficient routing architectures is a critical aspect of modern communication systems. This paper proposes a modified VLSI-based router architecture that is optimized for high-speed data transfer and low power consumption. The proposed architecture utilizes advanced routing algorithms and state-of-the-art VLSI design techniques to achieve a high level of performance and scalability. The performance of the design is evaluated through simulations. The simulation was carried out in a software called Xilinx and it is written using VHDL language. Design contains blocks called Arbiter, Cross bar and FIFO. The results show that the proposed architecture is able to achieve high throughput while maintaining a high level of scalability. This work is a significant step towards the development of high-performance communication systems.

KEYWORDS: Crossbar, Arbiter, FIFO, Round robin algorithm, NOC

I.INTRODUCTION

System on chip (SOC) is a complex interconnection of various functional elements. It creates communication bottleneck in the gigabit communication due to its bus based architecture. Thus there was need of system that explicit modularity and parallelism, network on chip possess many such attractive properties and solve the problem of communication bottleneck. It basically works on the idea of interconnection of cores using on chip network. The communication on network on chip is carried out by means of router, so for implementing better NOC, the router should be efficiently design. This router supports four parallel connections at the same time. It uses store and forward type of flow control and FsmController deterministic routing which improves the performance of router. The switching mechanism used here is packet switching which is generally used on network on chip. In packet switching the data transfers in the form of packets between cooperating routers and independent routing decision is taken. The store and forward flow mechanism is best because it does not reserve channels and thus does not lead to idle physical channels. The arbiter is of rotating priority scheme so that every channel once get chance to transfer its data. In this router both input and output buffering is used so that congestion can be avoided at both sides.

A router is a device that forwards data packets across computer networks. Routers perform the data "traffic direction" functions on the Internet. A router is a microprocessor- controlled device that is connected to two or more data lines from different networks. When a data packet comes in on one of the lines .The router reads the address information in the packet to determine its ultimate destination. Then, using information in its routing table, it directs the packet to the next network on its journey. The router is a " Four Port Network Router" has a one input port from which the packet enters. It has three output ports where the packet is driven out. Packet contains 3 parts. They are Header, data and frame check sequence. Packet width is 8 bits and the length of the packet can be between 1 bytes to 63 bytes. Packet header contains three fields DA and length. Destination address (DA) of the packet is of 8 bits. The switch drives the packet to respective ports based on this destination address of the packets. Each output port has 8-bit unique port address. If the destination address of the packet

matches the port address, then switch drives the packet to the output port, Length of the data is of 8 bits and from 0 to 63. Length is measured in terms of bytes. Data should be in terms of bytes and can take anything. Frame check sequence contains the security check of the packet. It is calculated over the header and data.

A data packet is typically passed from router to router through the networks of the Internet until it gets to its destination computer. Routers also perform other tasks such as translating the data transmission protocol of the packet to the appropriate protocol of the next network. The idea was explored in more detail, with the intention to produce a prototype system, as part of two contemporaneous programs. One was the initial DARPA-initiated program, which created the TCP/IP architecture in use today.

Sometime after early 1974 the first Xerox routers became operational. The first true IP router was developed by Virginia Strazisar at BBN, as part of that DARPA-initiated effort, during 1975-1976. By the end of 1976, three PDP-11-based routers were in service in the experimental prototype Internet.

The first multiprotocol routers were independently created by staff researchers at MIT and Stanford in 1981; the Stanford router was done by William Yeager, and the MIT one by Noel Chiappa; both were also based on PDP-11s.

II. LITERATURE SURVEY

A router is a device that forwards data packet between computer networks, creating an overlay internetwork. A router is connected to two or more data lines from different networks. When a data packet comes in one of the lines, the router reads the address information in the packet to determine its ultimate destination. Then, using information in its routing table or routing policy, it directs the packet to the next network on its journey. Routers perform the "traffic directing" functions on the Internet. A data packet is typically forwarded from one router to another through the networks that constitute the internetwork until it reaches its destination node.

Routers may also be used to connect two or more logical groups of computer devices known as subnets, each with a different sub-network address. The subnets addresses recorded in the router do not necessarily map directly to the physical interface connections. Forwarding an IP datagram generally requires the router to choose the address and relevant interface of the next-hop router or (for the final hop) the destination host.

In Transmission Control Protocol/Internet Protocol (TCP/IP) networking, routers are used to interconnect the hardware and software used on different physical network segments called subnets. Routers are also used to forward IP packets between each of the subnets. Determine the physical layout of your network, including the number of routers and subnets you need, before proceeding with the instructions in this guide.

Routers may provide connectivity within enterprises, between enterprises and the Internet, and between internet service providers (ISPs) networks. The largest routers (such as the Cisco CRS-1 or Juniper T1600) interconnect the various ISPs, or may be used in large enterprise networks. Smaller routers usually provide connectivity for typical home and office networks. Other networking solutions may be provided by a backbone Wireless Distribution System (WDS), which avoids the costs of introducing networking cables into buildings. All sizes of routers may be found inside enterprises. The most powerful routers are usually found in ISPs, academic and research facilities. Large businesses may also need more powerful routers to cope with ever increasing demands of internet data traffic.

2.1 Overview of Network-on-Chip:

The growing computation-intensive applications and the needs of low-power, high-performance systems, the number of computing resources in single-chip has enormously increased, because current VLSI technology can support such an extensive integration of transistors. By adding many computing resources such as CPU, DSP, specific IPs, etc to build a system in System-on-Chip, its

interconnection between each other becomes another challenging issue. In most System-on-Chip applications, a shared bus interconnection which needs arbitration logic to serialize several bus access requests, is adopted to communicate with each integrated processing unit because of its low-cost and simple control characteristics. However, such shared bus interconnection has some limitation in its scalability because only one master at a time can utilize the bus which means all the bus accesses should be serialized by the arbitrator. Therefore, in such an environment where the number of bus requesters is large and their required bandwidth for interconnection is more than the current bus, some other interconnection methods should be considered.

Such scalable bandwidth requirement can be satisfied by using on-chip packet-switched micro-network of interconnects, generally known as Network-on-Chip (NOC) architecture. The basic idea came from traditional large-scale multi-processors and distributed computing networks. The scalable and modular nature of NOCs and their support for efficient on-chip communication lead to NOC-based system implementations. Even though the current network technologies are well developed and their supporting features are excellent, their complicated configurations and implementation complexity make it hard to be adopted as an on-chip interconnection methodology. In order to meet typical SOC or multi-core processing environment, basic module of network interconnection like switching logic, routing algorithm and its packet definition should be lightweighted to result in easily implemental solutions.

2.2 Background:

The router used here is it avoid congestion and communication bottleneck. Although there are number of router implementation has already been done. Some of the related works are included here. Marescaux presented the implementation of router for NOC based system which has 2D torus network topology. Packet size was 8 bits and 2 control bits. The main drawback here was it was a 2D torus formed using 1D router which creates a serious bottleneck in traffic. Zerferino presented a soft core router for NOC ,the problem with this router implementation was it uses 4 flit buffer having 8 bit implementation which is quite high.

Its input and output channel has four distinct blocks and uses a large decoding logic. Moraes also presented its work but the drawback with it was that its packet has two headers which are quite expensive. The buffer here is present only with input channel. The absence of output buffer creates a serious problem in the implementation of router as it increases the problem of congestion.

Our paper removes most of the problems cited above and improves the performance of router. The most familiar type of routers are home and small office routers that simply pass data, such as web pages and email, between the home computers and the owners' cable or DSL modem, which connects to the internet (ISP).However more sophisticated routers, which connect large business or ISP networks up to the powerful core routers that forward data at high speed along the optical fiber lines of the Internet backbone.

III EXISTING SYSTEM

Router is a packet based protocol. Router drives the incoming packet which comes from the input port to output ports based on the address contained in the packet. The router has a one input port from which the packet enters. It has three output ports where the packet is driven out. The router has an active low synchronous input resetn which resets the router.

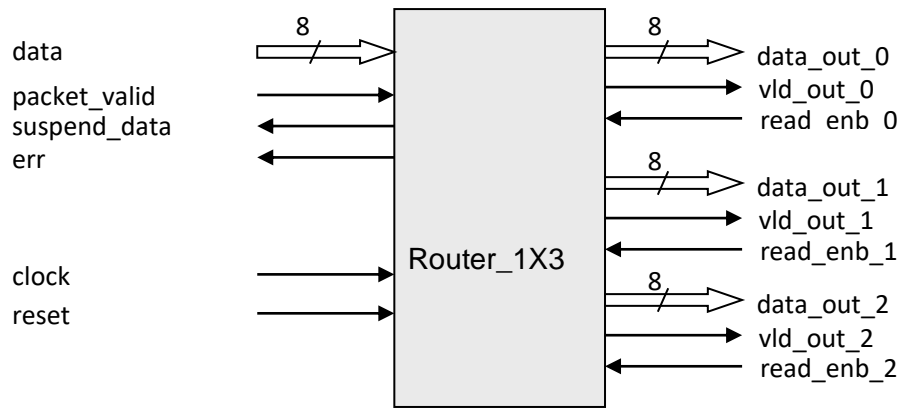


Figure 3.1- Block Diagram of Four Port Router

Data packet moves in to the input channel of one port of router by which it is forwarded to the output channel of other port. Each input channel and output channel has its own decoding logic which increases the performance of the router. Buffers are present at all ports to store the data temporarily.

The buffering method used here is store and forward. Control logic is present to make arbitration decisions. Thus communication is established between input and output ports. According to the destination path of data packet, control bit lines of FSM are set. The movement of data from source to destination is called switching mechanism. The packet switching mechanism is used here, in which the flit size is 8 bits. Thus the packet size varies from 0 bits to 8 bits. A detailed explanation of Design is as follow.

3.1 Packet Format:

Packet contains 3 parts. They are Header, payload and parity. Packet width is 8 bits and the length of the packet can be between 1 bytes to 63 bytes.

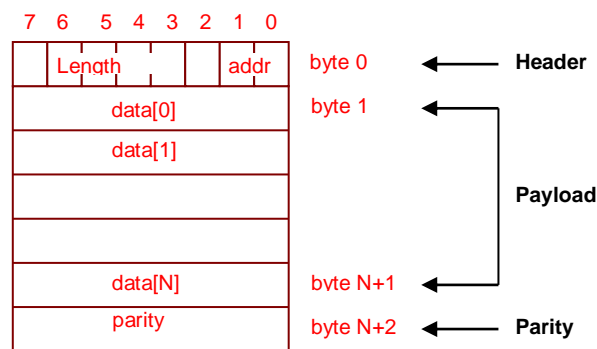


Figure 3.2- Data Packet Format

3.2 Packet Header:

Packet header contains two fields DA and length.

DA: Destination address of the packet is of 2 bits. The router drives the packet to respective ports based on this destination address of the packets.

[1] Each output port has 2-bit unique port address. If the destination address of the packet matches the port address, then router drives the packet to the output port. The address "3" is in valid.

[2] Length: Length of the data is of 6 bits and from 1 to 63. It specifies the number of data bytes.

[3] A packet can have a minimum data size of 1 byte and a maximum size of 63 bytes.

If Length = 1, it means data length is 1 bytes

If Length = 2, it means data length is 2 bytes

If Length = 63, it means data length is 63 bytes

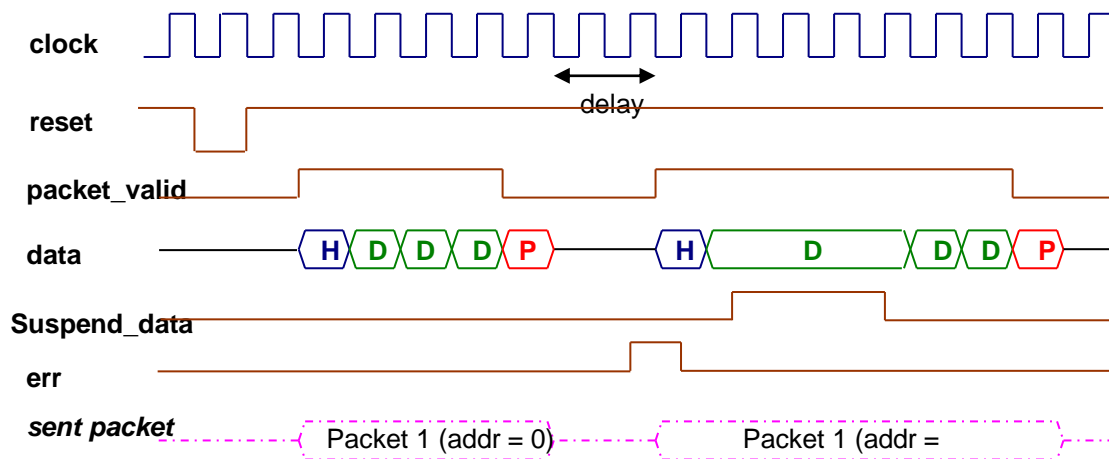
3.2 Packet – Payload:

Data: Data should be in terms of bytes and can take anything.

Parity: This field contains the security check of the packet. It should be a byte of even, Bitwise parity, calculated over the header and data bytes of the packet.

3.4 Router Input Protocol:

The characteristics of the DUV input protocol are as follows:



H = Header, D = Data, P = Parity

Figure 3.3- Router Input Protocol

[1]All input signals are active high and are synchronized to the falling edge of the clock. This is because the DUV router is sensitive to the rising edge of clock. Therefore, driving input signals on the falling edge ensures adequate setup and hold time, but the signals can also be driven on the rising edge of the clock.

[2]The packet_valid signal has to be asserted on the same clock as when the first byte of a packet (the header byte), is driven onto the data bus.

[3]Since the header byte contains the address, this tells the router to which output channel the packet should be routed (data_out_0, data_out_1, or data_out_2).

[4]Each subsequent byte of data should be driven on the data bus with each new rising/falling clock.

[5]After the last payload byte has been driven, on the next rising/falling clock, the packet_valid signal must be deasserted, and the packet parity byte should be driven. This signals packet completion.

[6]The input data bus value cannot change while the suspend_data signal is active (indicating a FIFO overflow). The packet driver should not send any more bytes and should hold the value on the data bus. The width of suspend_data signal assertion should not exceed 100 cycles.

[7]The err signal asserts when a packet with bad parity is detected in the router, within 1 to 10 cycles of packet completion

VI. PROPOSED METHOD

The Four 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 threefifo'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 FSM controller gives the err and suspended_data_in signals .this functions are discussed clearly in below FSM description.

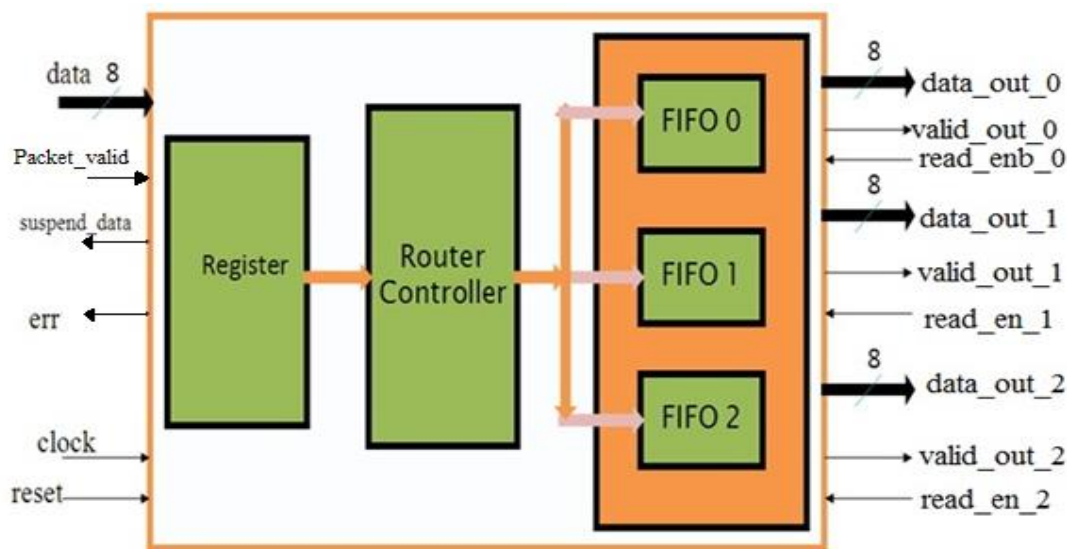


Figure-

4.1 Four Port Router Architecture

The router_reg module contains the status, data and parity registers for the Network router_1x3.

These registers are latched to new status or input data through the control signals provided by the fsm_router.

There are 3 FIFO for each output port, which stores the data coming from input port based on the control signals provided by fsm_router module.

The fsm_router block provides the control signals to the fifo, and router_reg module. The Router blocks Diagram shown below fig...

Router blocks are

- Register
- Router controller(FSM)
- FIFO Output Block

4.1 Register Block:

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 to the fifo for 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.

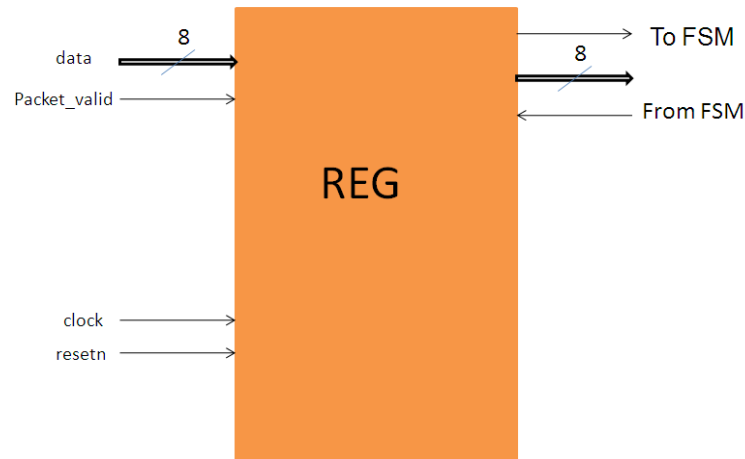


Figure4.2- Register Block

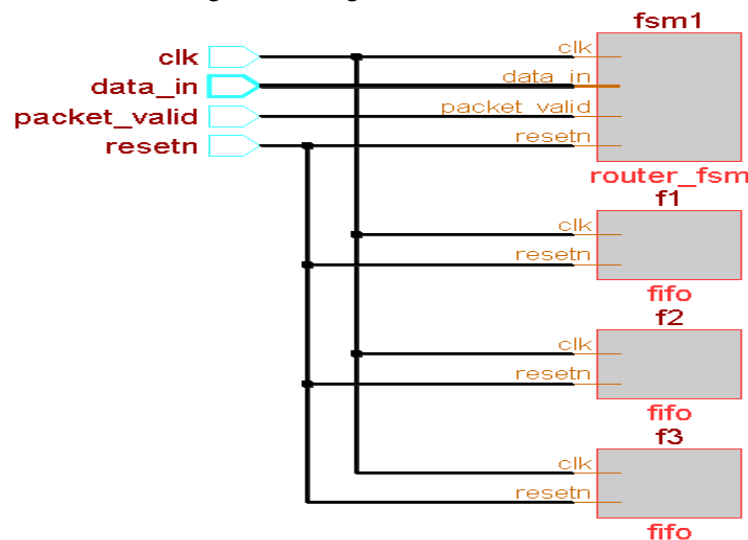


Figure 4.3-Register block synchronization

In the above figure register block is synchronize with the fsm to latch input data to it. Here, clk, resetn signals are synchronous with the entire module.

Eg: We are giving packet data as input to it and making read single (re1, re2, re3) as high w.r.t input first data byte of the packet. The receiving data is driven to the Router Controller for reaching its destination port. Which has 11 input pins (data_in [7:0], packet_valid, clk, reset).

Eg: data_in = 8'b10101010, clk, reset, packet_valid are HIGH

4.2 Router Controller (FSM):

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.

4.3 CDMA Receiver:

After de spreading the received signal with the corresponding code, it is compared with the same PN code, which is converted into parallel, using an 8 bit comparator. The comparator uses 0.33GHz clock frequency. If the actual transmitted data was a high then the de spread output will be same as that of the PN sequence. So the comparison function is performed in such a way that, it compares the de spread output with PN sequence. If it is same, then it can be concluded that the data send is a high and if it is not, then the data will be a low. So the comparator output corresponds to the actual

transmitted data of a particular user. Thus it is able to reconstruct the original data from the spreaded output.

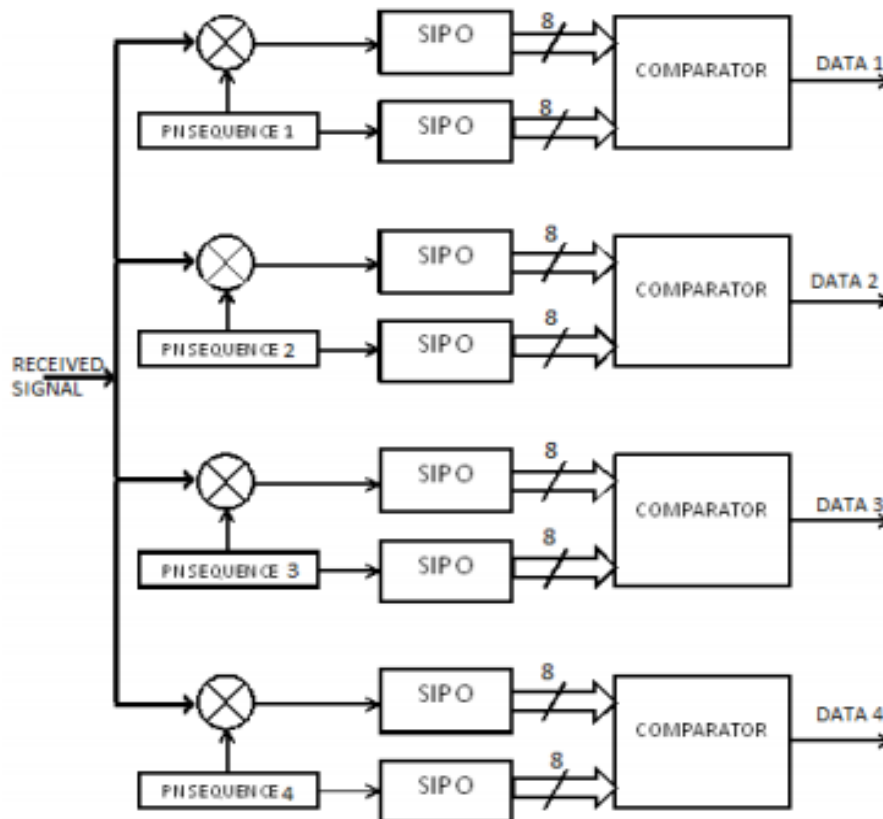


Fig: CDMA Receiver

4.4 PN sequence generator

Linear feedback shift registers are used for generating PN sequences. Components of D ip ops are used for this since structural modeling is used. To generate the sequence, rst it is necessary to initialize the ip ops to a particular value. Since 15 bit long PN sequence is being used, four ip ops are required and these four ip ops are required to be initialized. For that purpose, init signals are used. After the initialization, the xor feedback logic will provide a method to generate a PN sequence. Orthogonal sequences are required in this system. Time shifted versions of a PN sequence will be nearly orthogonal. So to shift the sequences, shift registers are used in which the sequence is given as input to the registers. The outputs from intermediate ip ops are taken which will be time shifted. So at the output of PN generator four PN sequences are obtained.

V CONCLUSION

An advanced FIFO structure based NoC is simulated and synthesized in Xilinx 14.7 ISE and implemented Vertex-6 FPGA device to analyze the performance in terms of occupied area, latency, power consumption and throughput. Single router is designed initially and then designed mesh based NoC to realize the memory utilization of FPGA. Fig.4 indicates that Register Transfer Level (RTL) schematic of single NoC router which is composed with input and output ports, arbiter, crossbar and channel control modules. The figure also describes the utilizations in terms of memory units each component individually. Each module of NoC designed using Verilog Hardware Description Language (HDL) separately and integrated as one module.

REFERENCES

- [1] K. Asanovic et al., "The landscape of parallel computing research: A view from berkeley," Dept.EECS, Univ. California, Berkeley, CA, USA, Tech.Rep. UCB/EECS-2006-183, 2006.
- [2] P. Bogdan, "Mathematical modeling and control of multifractal workloads for data-center-on-a-chip optimization," in Proc. 9th Int. Symp. Netw.-Chip, New York, NY, USA, 2015, pp. 21:1–21:8.
- [3] Z. Qian, P. Bogdan, G. Wei, C.-Y. Tsui, and R. Marculescu, "A traffic aware adaptive routing algorithm on a highly reconfigurable network-on-chip architecture," in Proc. 8th IEEE/ACM/IFIP Int. Conf. Hardw./Softw. Codesign, Syst. Synth., New York, NY, USA, Oct. 2012, pp. 161–170.
- [4] Y. Xue and P. Bogdan, "User cooperation network coding approach for NoC performance improvement," in Proc. 9th Int. Symp. Netw.-Chip, New York, NY, USA, Sep. 2015, pp. 17:1–17:8.
- [5] T. Majumder, X. Li, P. Bogdan, and P. Pande, "NoC enabled multicore architectures for stochastic analysis of biomolecular reactions," in Proc. Design, Autom. Test Eur. Conf. Exhibit. (DATE), San Jose, CA, USA, Mar. 2015, pp. 1102–1107.
- [6] S. J. Hollis, C. Jackson, P. Bogdan, and R. Marculescu, "Exploiting emergence in on-chip interconnects," IEEE Trans. Comput., vol. 63, no. 3, pp. 570–582, Mar. 2014.
- [7] S. Kumar et al., "A network on chip architecture and design methodology," in Proc. IEEE Comput. Soc. Annu. Symp. (VLSI), Apr. 2002, pp. 105–112.
- [8] T. Bjerregaard and S. Mahadevan, "A survey of research and practices of network-on-chip," ACM Comput. Surv., vol. 38, no. 1, 2006, Art. no. 1.
- [9] Y. Xue, Z. Qian, G. Wei, P. Bogdan, C. Y. Tsui, and R. Marculescu, "An efficient network-on-chip (NoC) based multicore platform for hierarchical parallel genetic algorithms," in Proc. 8th IEEE/ACM Int. Symp. Netw.-Chip (NoCS), Sep. 2014, pp. 17–24.
- [10] D. Kim, K. Lee, S.-J. Lee, and H.-J. Yoo, "A reconfigurable crossbar switch with adaptive bandwidth control for networks-on-chip," in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), May 2005, pp. 2369–2372.
- [11] R. H. Bell, C. Y. Kang, L. John, and E. E. Swartzlander, "CDMA as a multiprocessor interconnect strategy," in Proc. Conf. Rec. 35th Asilomar Conf. Signals, Syst. Comput., vol. 2, Nov. 2001, pp. 1246–1250.
- [12] B. C. C. Lai, P. Schaumont, and I. Verbauwhede, "CTbus: A heterogeneous CDMA/TDMA bus for future SOC," in Proc. Conf. Rec. 35th Asilomar Conf. Signals, Syst. Comput., vol. 2, Nov. 2004, pp. 1868–1872.
- [13] S. A. Hosseini, O. Javidbakht, P. Pad, and F. Marvasti, "A review on synchronous CDMA systems: Optimum overloaded codes, channel capacity, and power control," EURASIP J. Wireless Commun. Netw., vol. 1, pp. 1–22, Dec. 2011.