Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 2, Issue. 12, December 2013, pg.308 – 317

RESEARCH ARTICLE

Design of Synchronous NoC Router for System-on-Chip Communication and Implement in FPGA using VHDL

Arivu P [1], Shanmugasundaram N [2], Kamalanathan C [3], Dr.S.Valarmathi [4]
[1,2]Master of Engineering Student - BIT,Sathyamangalam.
[3]Associate Professor-BIT,Sathyamangalam.
[4] Professor-BIT,Sathyamangalam.

[1] arivup.ae12@bitsathy.ac.in, [2] shanmugasundarann.ae12@bitsathy.ac.in,
[3] kamalanadhan@gmail.com, [4] atrmathy@rediffmail.com

Abstract: System on chip (SoC) is widely used in VLSI technology. In SoC more number of devices can be placed on a single chip. In the communication between the devices Network-on-Chip (NoC) concepts can be used .NoC is a highly integrated heterogeneous SoC architecture, it provides the platform to be reliable, cost and energy efficient. For effective communication, on-chip routers provided routing functionality with low complexity and relatively high performance. The low latency and high speed is achieved by allowing routing function for each input port and distributed arbiters which gives high level of parallelism. In this paper we discuss about the design and implementation of on-chip router architecture using VHDL.

Keywords: **Cross-matrix; Router; Buffer; XY-Routing; Network-on-Chip; System-on-Chip**

I. INTRODUCTION

The System on Chip architectures suffer from communication bandwidth bottleneck, due to its bus based structure. To overcome this Network–on-Chip is used. Noc adopts the design data-routing consisting of communication links between the nodes and shared the information between the routing nodes, and provide global communication with in the same chip. By using the NoC concept in Multiprocessor SoC can eliminate the classical multiple buses or spaghetti-like chip structures and replace the well-structured, easily scalable and modular interconnected architecture.

Typical elements of Network-on-Chip are processing elements, network interface card and router. The router is a main part of Network-on-Chip. The network interface (NIC) is used for communication between router and the processing elements (PE). The router are used connected the various elements in the chip and mainly used for communication packets routing, safe transfer from the source to the destination in the sub-system and also prevented the congestion. The efficient design of router with routing algorithm provided the competitive Network-on-Chip architecture.

In this paper we presented 4 inputs to 4 outputs synchronous network-on-chip router for a packet switched SoC communication. Each input and output port of the router connects to a FIFO buffer to store the packets temporary. The central processing element of a router presents a route engine its function depends on the input arbiter and the look-up table. The router provides XY deterministic type of routing algorithm for input channel. Fig.1. shows general router architecture.

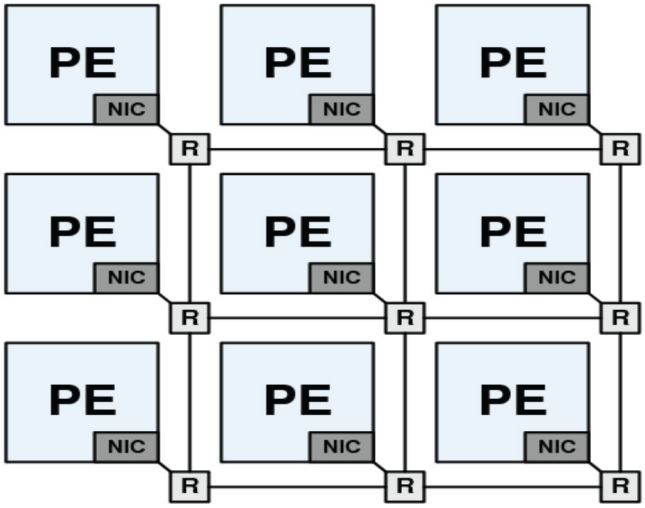

Figure.1. General NoC Router Architecture.

The entire processing element (PE) is connected to a network interface (NIC) which further connects PE to a local router. A packet is forward from the source PE to the destination PE, the packet is transmitted hop by hop across the network based on the routing path defined by the XY routing algorithm of each router. For each router, a packet received first and stored at an input buffer of that channel. Input channels arbitration is done through function of input router arbiter. Input router defines priority order of the input channels. In the receiver side of the router there are output FIFO buffers. Successful transfer of a data packet means that each input packet is received at the appropriate destination FIFO buffer. The selection

of routing algorithm is very important in router. The routing algorithm is divided into two groups they are: deterministic and un-deterministic. In both cases, chosen routing algorithm must ensure freedom of packets deadlocks, live-locks and starvation. Secondly, an adequate routing algorithm with efficient input channel arbitration makes the two third of the total system reliability. Implemented XY routing algorithm increases router performance due to its simplicity and low overhead [2].

II. PROPOSED ROUTER ARCHITECTURE

In this section described about the architecture and the detailed design of network-on-chip router solution and its parts. It is 4 inputs and 4 outputs synchronous router with an in/out FIFO buffers. The main parts of the router are input arbiter, routing-engine with its look-up table and 4x4 crossbar matrix with deterministic XY routing algorithm. We used XILINX software tool to describe the VHDL design of the router. The proposed router architecture is presented in Fig.2.

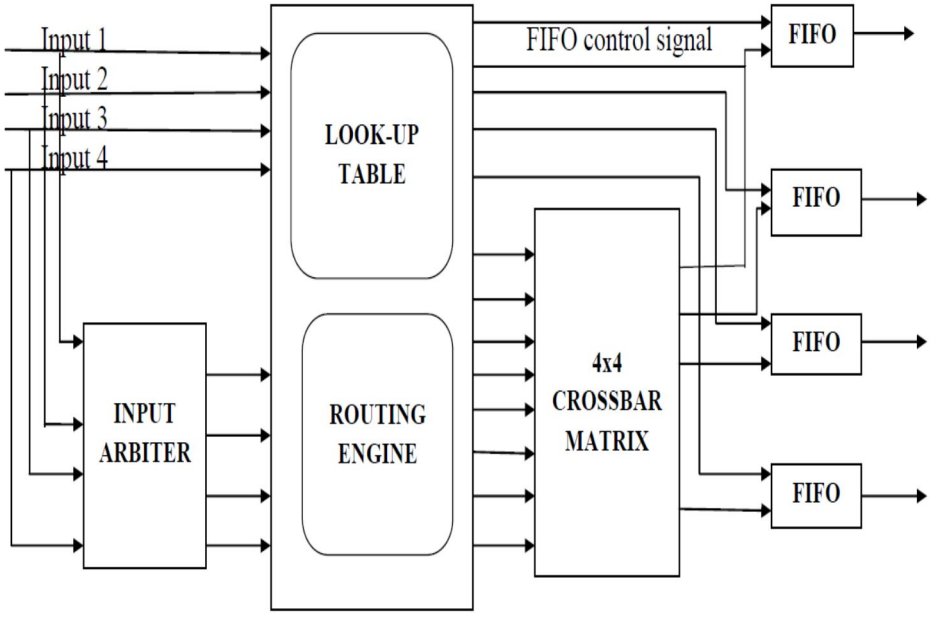

Figure.2. Proposed Router architecture.

A. ARBITER

Input arbiter controls the arbitration of the input ports or channels and resolves connection problem. It keeps the updated status of all the ports and knows which port is free and which port is communicating with each other. Arbiter also solves the problem of multiple requests coming at single output port. We are using round robin arbitration algorithm [4] in our simulation. The round robin arbiter scheduled the packets with same priority and destined to the same output. The round robin operates on the principle that an input channel requests which was served has the lowest priority in the next round arbitration.

In a proposed designed input arbiter as a finite state machine of Moore type. Each input channel request (R) signal is in arbitration process of assigning grant (G) signal. In this way achieved proper input channel arbitration. In fig.3. is presented VHDL description segment of an input arbiter finite state machine. The grant (G) signal controls the route engine and the crossbar matrix.

```
process(state_reg, R)
begin
G <= "0000";
case state_reg is
when waitr =>
if R(3) = '1' then
state_next <= grant3;
elsif R(2) = '1' then
state_next <= grant2;
elsif R(1) = '1' then
state_next <= grant1;
elsif R(0) = '1' then
state_next <= grant0;
else
state_next <= waitr;
end if;
```

Fig.3. VHDL description segment of an input arbiter.

B.ROUTING ENGINE WITH LOOKUP TABLE

In proposed work the function of routing engine is received packets routing to the destination output ports. It inspects the input data stream, an address from which a data packet is coming, and determines a destination address and output port. Routing algorithm, this defines as the path taken by a packet between the source and the destination. Here we use the XY routing algorithm. This way we implemented X (source address) – y (destination address) algorithm of routing. A XY routing algorithm mainly used in NOC because for its simplicity [6]. Is the look-up table; containing the parameter (rotation, translation, etc.) values for the movement of each joint. Here look-up table provides a set of source-destination address data pairs. Routing engine "takes a look" into the look-up table and matches source packet address to the destination one. For the purpose of a practical design simulation, we decided for ATM

protocol packets. ATM provides functionality that is similar to both circuit switching and packet switching networks. ATM uses the asynchronous time-division multiplexing that encodes data into small and fixed size packets. In Fig.4 is presented ATM cell which consists of a 5-byte header and a 48-byte payload [5].

Routing engine as inputs has grant (G) signals from input arbiter while its outputs are write enable signals (*WEs*) for FIFO control, crossbar-matrix multiplexers *Sel* signals and *Validout (port number)* signals for output data ready notification. RTL description segment of the routing engine is presented in Fig. 5.

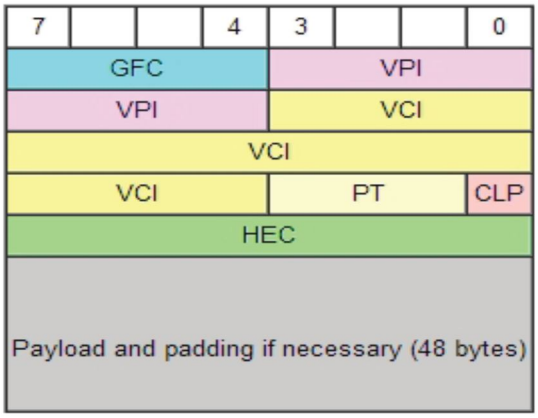

Fig.4. ATM protocol packet

case input_db_stream_s1(39 downto 16) is

when x"070809" =>

output_port_no_s <= "010";

if (full2 = '0' and empty2 = '1') then

WE2 <= '1';

Validout2 <= '0';

output_db_stream_1 <=

input_db_stream_s1(47 downto 40)

&c1&input_db_stream_s1(15 downto 0);

sel2 <= "00";

else

RE2 <= '1';

Validout2 <= '1';

end if;

Fig.5. RTL description segment of the routing engine.

C.CROSSBAR MATRIX

In proposed work the design of crossbar used 4-in-1 multiplexer array structure which enables each input data channel to be connected to each output data channel.Crossbar-matrix is controlled by two bit selected (sel) signals that are coming as process out from the routing engine. Crossbar – based systems can be significantly less expensive than bus or ring systems with equivalent performance because the crossbar allows multiple data transfers to take place simultaneously. Input packets from input channel0 to input channel4 to each crossbar multiplexer. Thus every crossbar multiplexer is having four packets as input. Output of every crossbar depends on bits occurs on select lines of that particular crossbar multiplexer. The crossbar output lines are inputs to output FIFO buffers. In Fig.6. is presented block diagram of crossbar matrix.

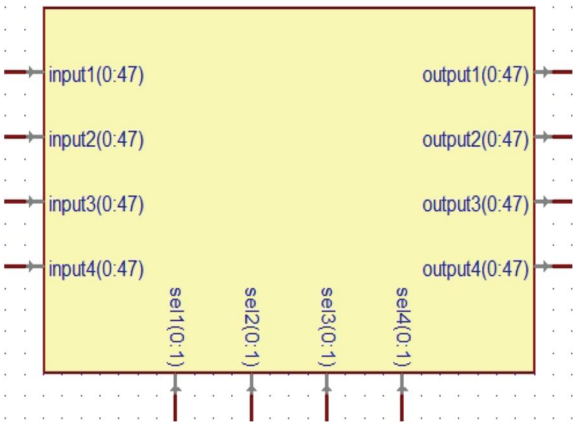

Fig.6. Block diagram of 4x4 Crossbar matrix

D.FIFO BUFFERS

In proposed router buffering is required to provide temporary storage of packets that are in input and output. The FIFO buffers can be variable size and depth, depending on implementation specification. Here implement the circular FIFO buffers of 48B size and 64B depth. The FIFO notify the following flags empty and full and also have the read and write pointers. Those flags are used to implement FIFO buffer overflow mechanism. The status of FIFO decides if the communication can start or not. The use empty and full flags are if the FIFO is empty (empty=1), data packets can be stored in it and start the communication. If FIFO is full (full =1) data packets can be read from buffer and forwarded to the destination port. The empty flag also indicates the termination of communication.

Increase the system reliability and controllability. We introduced two additional flags in the FIFO buffer implementation, they are *almost full (afull)* and *almost empty (aempty)*. The flag *afull* is set during the transmission when a FIFO reaches 85% of its total depth. The field *aempty* is set the FIFO can reaches 15% of its total depth. Also use the *validout (port number) flag* is also implemented to ensure valid FIFO output data value. In Fig. 7. is presented block diagram of a FIFO.

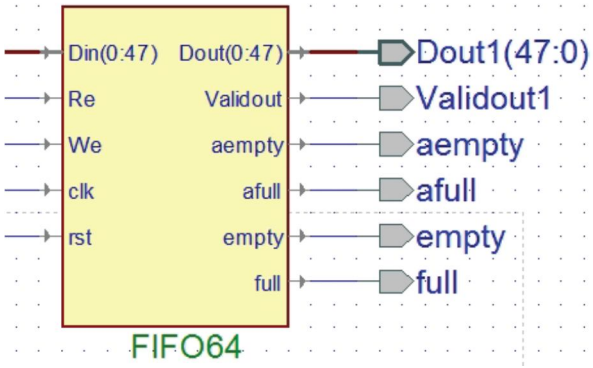

Fig 7. Block diagram of FIFO

III. ROUTER TEST BENCH DEVELOPMENT

The network-on-chip router entities were designed by using VHDL description, we developed a test bench for each entity. The test bench for each entity was designed and verifies simulation for each individual entity. After testing all the sub-modules, we integrated all entities into the single entity to design router top level. To be able to test the implemented router design and also developed a top level test bench. The test bench initializes communication and starts the data transfer. At the test bench stimulation we use the hexadecimal values from input_data.txt file. Then the data packets reach to the destination ports, they are being written into output_data.txt file. In this way, comparing the content of the input and output data files, we checked the correctness and regularity of data packets transferred over the network-on-chip router

IV. SIMULATION AND RESULTS

As it can be seen on the ISE simulation diagram, request signal (R) has a vale $F_{(16)}$ or $1111_{(2)}$. That means all the four input ports request a grant signal to start the data packets transfer. According to the round robin arbitration algorithm the arbiter assigns a grant (G) signal to the previous cycle last served input. The grant (G) signal has a value $8_{(16)}$ or $1000_{(2)}$. That means the input channel 4 has a grant to start the data packet transfer. At the time input port 4 is currently valid data packet "060708090A3A".

Now, the routing engine match this input value according to the look-up table content and determines which output port should be assigned to this packet. Then routing engine replaces 24 bits of input data into matched 24 bits from the look-up table. Then packets are forward to the destination port. The select signal for crossbar matrix and FIFO control signals are set it controls the output port. The data packets are flow through crossbar matrix and finally the packets are been stored into appropriate FIFO buffer. This process is repeated until input channels FIFO buffers are not empty. In table.1. is have the content of the input_data.txt file and output_data.txt file when the data transfer is finished. In fig .8. shows the waveform of the VHDL representation.

Input_data.txt	**Output_data.txt**
060708090A3A	060E963390A3A
8182838485B5	818FFFF485B5
3C3D3E3F4070	3C3210AF4070
7273747576A7	72756C9576A7
15161718A49	151010481A49

Table 1 input and output_data.txt file.

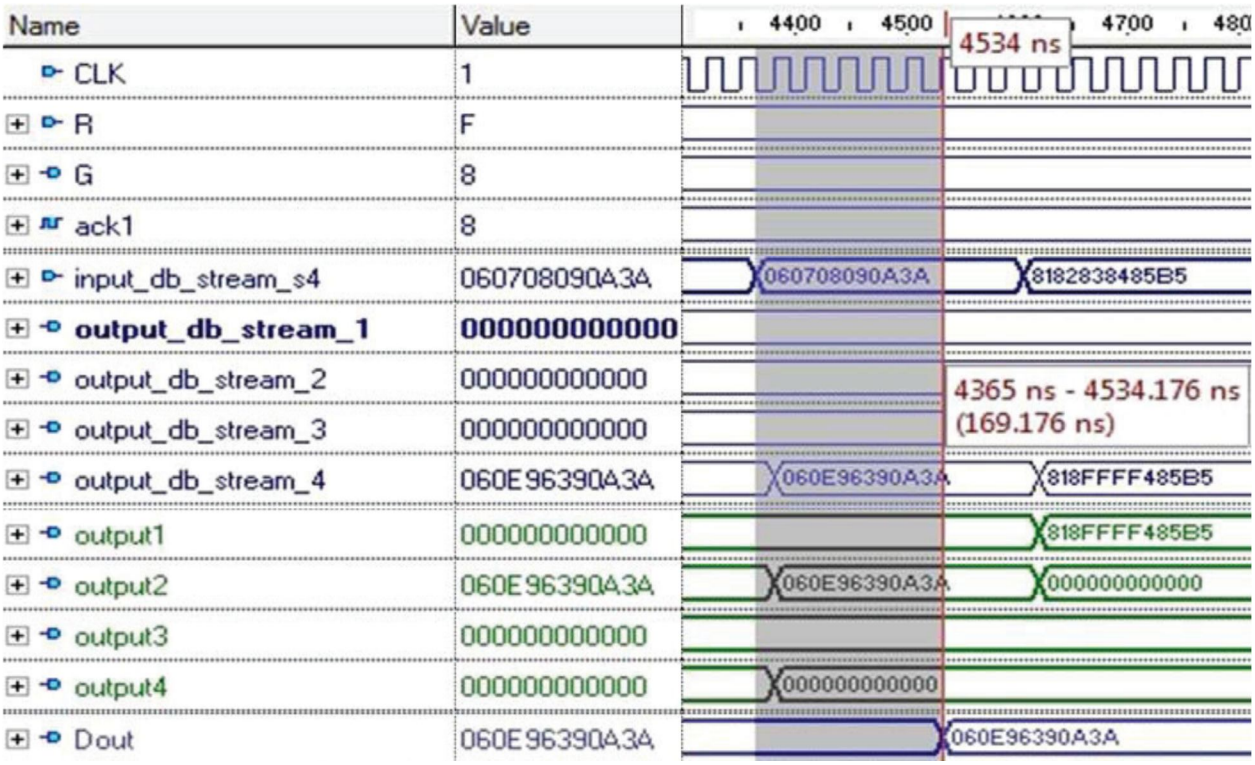

Fig.8. Waveform representation.

V. CONCLUSION

In this paper, we have proposed a possible synchronous network-on-chip router architecture with VHDL design. This router architecture is flexible, scalable and offers reliable data packets transfer. Next step and future work to perform power analyze area and timing characteristics of the router. Another possible further work design asynchronous router and perform comparative analysis with synchronous router architecture.

VI. ACKNOWLEDGEMENT

I convey sincere Thanks to management of Bannari Amman Institute of Technology, Sathaymagalam. I thank our project guide MR.C.Kamalanadhan for his support and also thanks to my parents and my friends for their blessings for making this work as successful.

.

REFERENCES

[1] G. De Micheli, C. Seiculescu, S. Murali, L. Benini, F. Angiolini, A.Pulini, “Networks-on-chip, From Research to Products”, DAC’10,June 13-18, 2010, Anaheim, California, USA

[2] Swati Malviya (M. Tech – Digital Communication), Anurag Jaiswal (Capgemini Consulting India Pvt. Ltd., Former HOD-IT, FSM,MITS University), “Five Port Router for Network-on-chip”, to be published.

[3] ALDEC, The Design Verification Company, Active-HDL Software Tool for FPGA Design Creation and Simulation,

http://www.aldec.com/en/products/fpga_simulation/active-hdl

[4] Abbas Noon, Seifedine Kadry, Ali Kalakech, "A New Round Robin Based Scheduling Algorithm for Operating Systems: Dynamic Quantum Using the Mean Average", IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 3, No. 1, May 2010

[5] Asynchronous Transfer Mode Protocol, Structure of an ATM Cell,

http://en.wikipedia.org/wiki/AsynchronousTransfer Mode

[6]Shubhangi d chawade, mahendra a gaikwad, rajendra m patrikar, "Review of xy routing algorithm for Network-on-chip architecture" international journal of internet computing issn no: 2231 – 6965, vol- 1, iss- 4 2012.

[7] Marko R. Ili , Vladimir Z. Petrovi and Goran S. Jovanovi "Simulation of synchronous Network-on-chip router for System-on-chip communication" 20th Telecommunications forum TELFOR 2012

