Network on Chip Router Architecture Performance Analysis by using VHDL

Ruchika Chandravanshi
Dept Electronics & Communication,
SIRTS, Bhopal (M.P.) India

Vivek Tiwari
Dept Electronics & Communication,
SIRTS, Bhopal (M.P.) India

ABSTRACT

Networks-on-Chip (NoCs) are increasingly used in many-core architectures. Todays technology for ASICs supports Networks-on-Chip designs which can have 100 million gates on a single chip. In order to implement a competitive NoC architecture in FP-GAs, the area occupied by the network should be kept to a minimum. This helps in utilizing maximum area by the logic while maintaining the performance of the router network. Reducing area also reduces the power consumption. NOC's designs promise to offer considerable advantages over the traditional bus-based designs, in solving the numerous technological, economic and productivity problems associated with billion-transistor system-on-chip development. Here, a very flexible network is designed to propose a highly scalable and can be easily changed to accommodate various needs. This proposed work focused on implementation, analysis and verification of a five port routers. The building blocks of the router are input/output, multiplexer/demultiplexer, crossbar switch, buffer register and arbiters. The arbiters uses the round robin algorithm. The proposed NOC architecture is simulated in Xilinx ISE10.1 software. The source code is written in VHDL. This design is suitable for building networks with irregular topologies and with low latency and high throughput.

Keywords

NOC, VHDL, Router Architecture

1. INTRODUCTION

In future SOC designs, one of the major problems arises from nonscalable global wire delays. The global wires carry signals across a chip, but these wires typically do not scale in length with technology scaling . In [1], NoC is implemented using Torus topology. They proposed router architecture, a router algorithm and also provided solution to the problem introduced by the long wires in torus topology by pipe lining both the long wires and short wires and by lengthening the input buffers attached to the long wires. Since gate delays can be scale down with technology. Global wire delays typically increase exponentially or, at best, linearly by inserting repeaters[2]. The delay may exceed the limit of one clock cycle or often, multiple clock cycles, even after repeater insertion. In ultra-deep submicron processes, 80 percent or more of the delay of critical paths will be due to interconnects [3] [4]. In fact, many large designs today use FIFO (first-in, first-out) buffers to synchronously propagate data over large distances to overcome this problem. In [5], FPGA implementation of network on chip framework is given. In this they conclude that as compare to common bus architecture NoC architecture is effective in terms of speed performance and throughput. The most frequently used on-chip interconnect architecture is the shared medium arbitrated bus, where all communication devices share the same transmission medium. The advantages of the shared-bus architecture are simple topology, extensibility and low area cost. This new approach employs explicit parallelism; exhibits modularity, therby need for global synchronization can be disappear. In a network-centric approach, the communication between IPs can take place in the form of packets. We suggest that a network-on-chip (NoC) resemble the interconnect architecture of high-performance parallel computing systems. The common characteristic of these kinds of architectures, is that the functional IP blocks communicate with each other with the help of intelligent switches.These switches can be considered as infrastructure IPs (I2Ps) [6] providing a robust data transfer medium for the functional IP modules. The proposed router is designed to be used in a reconfigurable computing platform.To minimize the area of an FPGA based device, the size of the router should be of small. Although there are a number of ASIC based router implementations, we restrict our discussion of the related work to FPGA-based implementations only. . Marescaux et al. [7] present the first working implementation of a NoC based system on FPGAs. Their system has scalable 2D-Torus, XY blocking, hop-based and deterministic routing. The packet size has 16 data bits and 3 control bits. It has two virtual channels each operating at 40MHz. Their switch takes 446 slices on a Xilinx Virtex XCV 800. The main drawback is that it is a 2D torus formed using 1D-routers. This creates a serious bottleneck in the traffic as it has to go through a linear structure of 1D-routers. They also extended their work to a highly scalable network on chip for reconfigurable systems.

In [8] analysis of five port router is given. The diagonal Mesh topology is designed to offer a good tradeoff between hardware cost and theoretical quality of service in [9]. Design can contain large number of nodes without changing the maximum diameter. They also present a new router architecture called Flexible, Extensible Router NoC (FeRoNoc). The NoC router architecture with five port presented in [10], which utilizes dual crossbar arrangement. Due to which latency arises, this reduces using predominant routing algorithm.

In [11] the authors present an energy- efficient NoC for heterogeneously tiled reconfigurable SoC. It has a 5×5 wormhole router with virtual channels along each direction. There is a central arbiter which establishes necessary connections between various channels. The bottleneck occurs at the central arbiter, which follows a round-robin approach of service. Theoretically, there are five possible parallel connections out of the total of twenty five combinations in a five port router. If the arbitration scheme is central, it leads to performance loss because of queuing of the requests from all ports at one central place.. This large area usage results in two drawbacks: First, it is not energy efficient and second, it will put severe constraints on the total available area for the user logic.

2. ROUTER ARCHITECTURE

A router has a set of ports, namely, Local (L), North (N), East (E), South(S) and West (W), to communicate with the local logic element and the neighboring routers. These ports receives the incoming packets and forwards them to the appropriate port. To store the packets temporarily, buffers are present at various ports. Control logic will be present to take routing decisions and arbitration decisions. Here , we design a light weight, energy efficient, parallel router. The motivation is to reduce the area which also reduces the power consumed. In this work, we choose one of the popular methods of buffering called store and forward. The motivation behind choosing such a scheme is to have the simplest possible decoding logic, thereby, reducing both area and power. Establishment of connections are made automatically without any complex decoding logic. The router switches with a set of inter-communicating ports, define the physical layer of the NoC system. For communication there are two types of ports, namely, the Input and output port. Communication is done by use of two handshake signals (Req/Ack) between the co-operating ports (output and input). This forms the data link layer. Dynamic establishment of connections and routing of packets constitutes the network layer. Here, the cross-point matrix is a very important component, the controls of which are maintained by the output channels. Inside the router, the Gnt/Ack signals are used to access the FIFO, without the need of any explicit signals.

3. IMPLEMENTATION OF THE ROUTER

The block-level diagram of the proposed router is given in the figure(a).

The router has three main blocks, namely the Input Channel, Cross point Matrix and Output Channels.

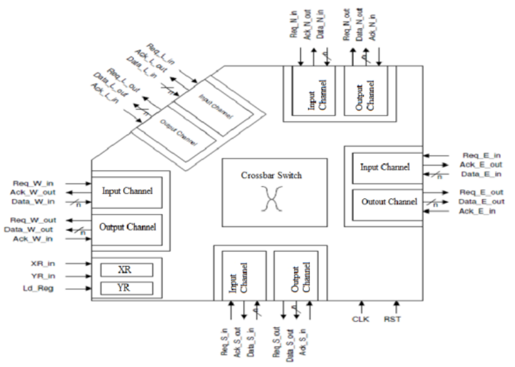


Fig: Router Architecture

3.1 Input channel

There is one input channel at each port, each running its own control logic. Each Input Channel has a FIFO of depth 16 and data width of 8 bits and a Control Logic which has been implemented as a FSM. The input channel accepts request from other neighboring router. On receiving the request, if it is free, it will acknowledge the request. The first flit is the header and following flits constitute the data. It will accept data as long as the request signal is held high. The previous router's output channel ensures that the request line is held high until it empties the packet of data, being accepted by the input channel. The input channel sets the acknowledge line high, as long as there is a transfer taking place (indicated by request line).When the transfer is complete, the request and acknowledge lines go low in sequence. The packet of data received from the previous router is stored locally in the FIFO thereby implementing a store-and-forward dataflow. Next the control logic reads the header of the packet and using XY Routing (described later) decides which output channel is to be requested for sending the packet out of the router and sends the request to that output channel. It is to be noted that each of the input channel is running an independent FSM and hence can initiate five possible parallel connections at the same time. Once the input channel gets a grant from the requested output channel, the control bits of the cross point matrix are set appropriately by the granting output channel. An important optimization is done here by performing a simple logical OR (instead of the costly Multiplexer-based implementation) of the Gnt and Demux select lines, thereby gaining in area and performance. The fact that at a time only one output channel will be requested by an input channel is exploited here. The inter channel data transfer is also governed by the empty status of the FIFO, thereby, removing complex decoding logic. Empty condition automatically triggers the next transfer.

3.2 Crosspoint Matrix

The Crosspoint Matrix is a set of multiplexers and demultiplexers having an interconnection allowing all possible connection between the 5 input and 5 output channels. The output channel while granting the request of an input channel configures the multiplexers and demultiplexers of the cooperating input and output channels thereby establishing the connection between them for the transfer of the packet.

3.3 Output Channel

There is one output channel at each port which has an 8-bit FIFO of depth 16 and an control logic (FSM) making arbitration decisions. The output channel gets requests from the different input channels and andvising a simple decoding logic. When transfer is complete the crosspoint matrix controls are reset. FSM then initiates the process to send the data into neighboring router using handshake mechanism. Empty status of its FIFO triggers the next inter-channel transfer.

3.4 XY Routing

XY routing algorithm comes under deterministic routing algorithm. This algorithm can be implemented for both for regular and irregular network topology. It is called dimension order routing (DOR). It follows the concept of minimal turning routing. In this routing each node or router of NoC is identified by the (x, y) co-ordinates of that node for a 2D mesh. According to this algorithm the data packets will traverse in X-direction towards the destination column. After finding the destination column the data packets will traverse to the destination node. Data movement in this algorithm is described in an example in Figure 4.2. This algorithm simply states that "First the data will move in X-direction and then in Y-direction". That is why the name of the algorithm is XY Routing algorithm. According to this algorithm the packets can't move first in Y-direction then in X-direction. So it has some routing or turning restrictions. Due to which it becomes deadlock free. According to this algorithm, (x ,y) coordinate of Current router is compared to the (x,y) coordinate of Destination router [6]. If they are equal, this will indicate data packets have reached its destination and they will be routed to the resource through the local port. If they are not equal, the x-coordinate of destination is initially compared to the x-coordinate of current router. When x-coordinate of current router is less than x-coordinate of destination router, the data packets will be transmitted to the East direction, when x-coordinate of current router is greater than x-coordinate of destination router, then packets will be routed to West direction. Then check if the data packet is already on the destination column or not. If this last condition is true, the vertical address of destination is compared to the vertical address of current router. Data packets will be traversed to South direction when vertical address of current router is less than vertical address of destination router, to North when vertical address of current router is greater than vertical address of destination router.

3.5 Round-Robin Arbiter (RRA)

Round Robin Arbiter is implemented as FSM at each output channel. RRA arbitrates and decides which input channel is to be given access to that output channel when many channels are requesting the same output. Generally, the output channel must follow a priority based arbitration. If a fixed priority scheme is followed, the same input channel may get access repeatedly. Hence in our arbiter, the priorities of the input ports are changed dynamically taking the last input port serviced into account. The priorities are implemented in a clockwise fashion i.e., if the last input port serviced was North, then during next service, the priorities will be in the order of East, South, West, Local and North. It should be noted that no clock cycles are wasted in our scheme as the grant is issued only if there is a request from corresponding input channel. Since each input channel has its own XY Routing FSM and each output channel has its own RRA FSM, there is no latency in establishing the connections. This allows different requests to be granted simultaneously at the same time, when five requests come for different output channels. This provides a significant improvement in the performance of our router. It is to be noted that the router coordinates are stored in two registers inside each of the router, which can be accessed from primary inputs. This facilitates easy reconfiguration of the router coordinates in case of system change, compared to the hard-coded coordinates, where one has to re-synthesize with new coordinates. We extend our work to build a 1×2 and a 3×3 mesh-type router network.

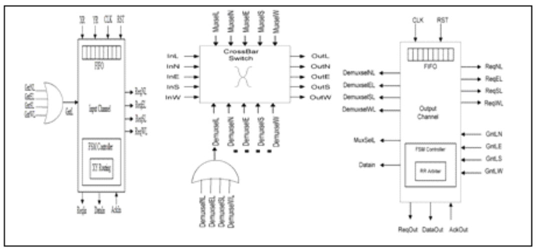


Fig: Input channel, Crossbar matrix and Output channel of proposed router

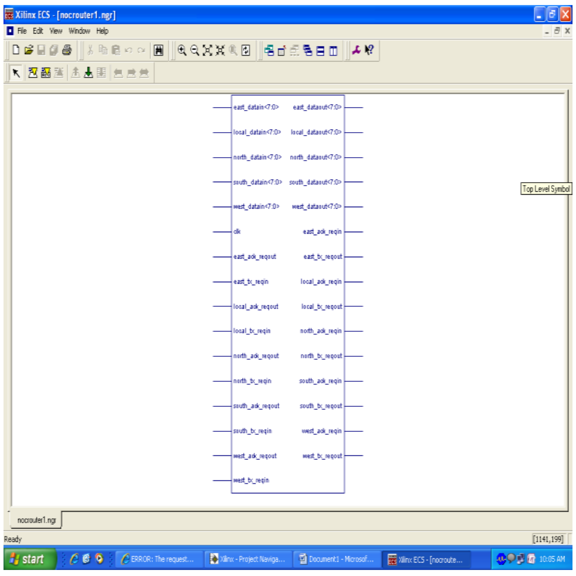
4. ROUTING ALGORITHM

The routing algorithm is another important design choice: The taxonomy of the routing algorithms is very complex but from the adaptivity point of view they can be divided into adaptive and deterministic algorithms. Our router uses a deterministic algorithm with a limited amount of adaptivity. The routing algorithm can be implemented as a finite state machine or as a look-up table. We have chosen a look-up table approach in the current implementation. The table has an entry for each IP in the network, and its content can be dynamically changed by the OS, at run time. Having one entry for each IP in every router gives us the possibility to customise the routing for each IP at the link level. The drawback is of course that the required memory space grows proportionally with the number of IPs. This is to be compared with a finite state machine implementation that has a fixed size independent of the number of IPs but has no flexibility. This higher-level form of adaptivity, could be useful in networks with a known traffic pattern. The IPs can be characterized in terms of the required data rate, and knowing the senders and the receivers for each IP, the routing tables can be modified to balance the network traffic. Each time a task that produces a change in the communication between the IPs is started, the OS can compute a new optimum distribution of the communication channels and the tables changed accordingly.

This feature can be exploited in reconfigurable systems as well. When two IPs exchange many packets it is preferable to place them next to each other, to exploit traffic locality. In reconfigurable systems adjacent positions can be freed by swapping IPs. However, IP swapping can take a significant amount of time, not to mention the temporary loss in performance: the tasks will have to be interrupted, their state saved, then they are swapped and their state restored. The flexibility in reprogramming the routing tables can be used to avoid frequent IP relocation. Providing that there are enough free communication resources, packets can be made to travel over low traffic connections incurring the minimum extra delay but taking full advantage of the maximum available bandwidth and saving the time required by IP swapping and maintaining undisturbed the currently executing tasks. The update of the routing tables would be performed by the OS, and would therefore require considerably more time than just a few clock cycles. Therefore, this type of adaptivity does not offer all the power of a hardware implemented adaptive algorithm. However, for systems with well known communication pattern, this type of flexibility will be adequate for many situations, presenting the advantage of much lower hardware requirements than would be needed for true adaptive routing algorithms.

5. FPGA IMPLEMENTATION OF NOC ROUTER

Once the basic algorithm of Router is tested in software, a hardware model for Router model is developed at Register Transfer Level (RTL) manually inVHDL.


Xilinx ECS - [nocrouter1.ngr]

File Edit View Window Help

Ready [1141,199]

Figure 4. RTL View of NoC Router

6. RESULT

The input output waveform can be verified using simulator or I/O of FPGA device. The simulation of NoC Router is shown in figure 5.

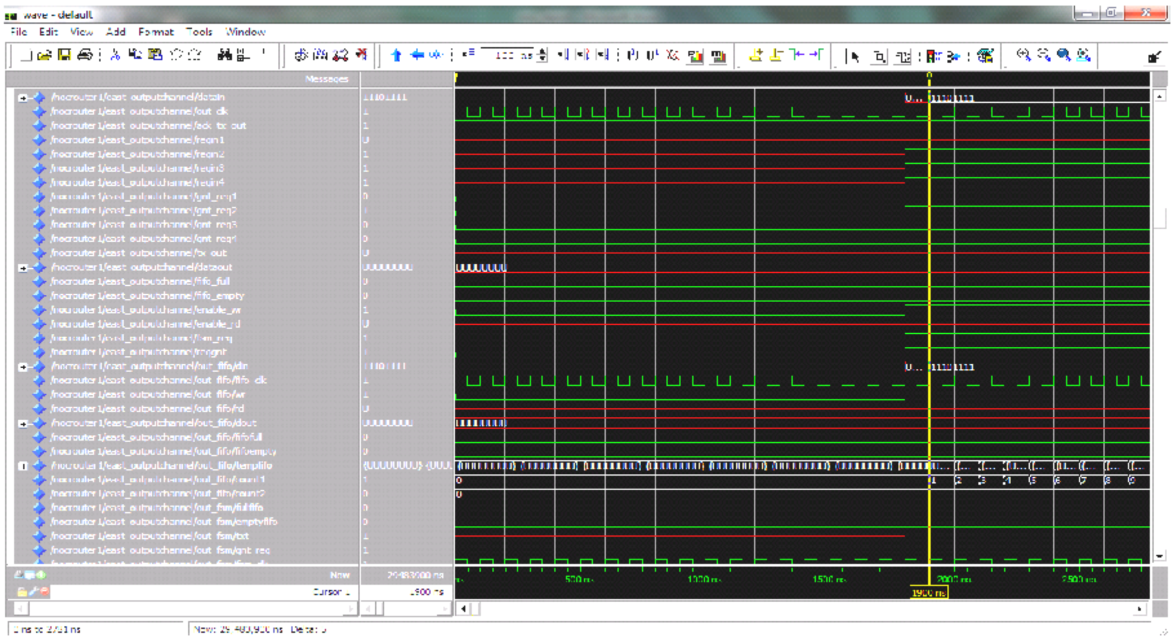


Figure 5. Simulation result of NoC Router

7. CONCLUSION

A light-weight parallel router architecture for implementing Networks-on-Chip is implemented on FPGAs. We implement optimizations in the FSMs and decoding logic, thereby, saving significant area. we intend to build an advanced router prototype supporting HLP having less area overhead.

8. REFERENCES

[1] Angelo Kuti Lusala, Philippe Manet, Bertrand Rousseau, Jean-Didier Legat "NOC IMPLEMENT A nON IN FPGA USING TORUS TOPOLOGY" International Conference on Field Programmable Logic and Applications, PP 778-781, IEEE-2007.

[2] U. Y. Ogras and R. Marculescu, "„It's a small world after all‟: NoC performance optimization via long-range link insertion," IEEE Trans. on VLSI,14(7), 2006.

[3] A. Hemani et. al. Network on Chip: An architecture for billion transistor era. In IEEENor Chip Conference, November 2000.

[4] Fernando Moraes et. al. A Low Area Overhead Packet-switched Network On Chip: Architecture and Prototyping. In IFIP VLSI-SOC 2003, pages 318–323, 2003.

[5] Shubha B C, Srinkanta P, "FPGA Implementation of Network on Chip Framework using HDL", Students' Technology Symposium(TechSym), IEEE PP. 151-155, April 2010.

[6] M. Lai, Z. Wang, L. Gao, H. Lu, K. Dai, "A Dynamically-Allocated Virtual Channel Architecture with Congestion Awareness for On-Chip Routers," in Proceedings of the 46thDesign Automation Conference (DAC), pp. 630-633, 2008.

[7] Theodore Marescaux et.a!. Interconnection Networks Enable Fine-Grain Dynamic Multi-Tasking on FPGAs. In FPL' 2002, pages 795-805, September 2002.

[8] Swapna S, Ayas Kanta Swain, Kamala Kanta Mahapatra,"Design And Analysis of Five Port Router For Network On Chip", Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics(PrimeAsia) 2012.

[9] Majdi Elhajji, Brahim Attia, Abdelkrim Zitouni, Rached Tourki,Jean-luc Dekeyser, "FERONOC:Flexible and Extensible Router Implementation For Diagonal Mesh Topology" Author manuscript, published in Conference on Design and Architecture for Signal and Image Processing(2011).

[10] M.S.Suraj, D.Muralidharan, K. Seshu Kumar, "A HDL Based Reduced Area NoC Router Architecture"978-1-4673-5301-4/\3/$31.00 © 2013 IEEE.

[11] Rachid Dafali, Jean-Philippe Diguet, "Self-Adaptive Network-on-Chip Interface", IEEE Trans. on VOL. X, NO. X,2005.

[12] E. M. Choudhari, Dr. P. K. Dakhole, ' Design And Verification Of Five Port Router For Network On Chip', IEEE International Conference on Communication and Signal Processing, April 3-5, 2014.