

# A Flexible Design of Network on Chip Router based on Handshaking Communication Mechanism

Seyyed Amir Asghari<sup>1</sup>, Hossein Pedram<sup>1</sup> and Mohammad Khademi<sup>2</sup>

<sup>1</sup>Amirkabir University of Technology

<sup>2</sup>Shahid Beheshti University

seyyed\_asghari@aut.ac.ir, pedram@aut.ac.ir, m.khademi@mail.sbu.ac.ir

## Abstract

One of the important issues in power and performance trade off analysis in Network on Chip designs is communication. Communication portion in the power consumption of System on Chip can be up to 50% of the whole power of consumption of the chip. This deems to be more important for Network on Chips which center around an intercommunication networks. Many Networks on Chip routers have been designed; however most of them have not been implemented until now. In this paper, design and implementation of a synchronous Network on Chip router based on asynchronous communication mechanism are presented. We designed a router with scalability feature which is synthesized in both FPGA and ASIC infrastructures. In addition, the proposed router uses low resource utilization percentage of FPGA and ASIC.

## 1. Introduction

Power and performance are two essential features which in corresponded with each other, produce main concerns in design and implementation. Nowadays, very large integrated digital systems [1-4] (Systems on Chip) may contain different components such as processor, input-output units and different types of memories. Likewise, each of these components may include different specifications such as variable bandwidth, buses and different communicative protocols. Generally, bus is utilized for interconnecting the processing elements of System on Chip (SoC). However by increasing the number of processing elements, the bus itself is transmuted into a bottleneck. To obviate this difficulty, the idea of Network on Chip (NoC) has been introduced [5].

This network can be modeled as a graph wherein nodes, processing elements and edges are the connective links of the processing elements. In this article, design and implementation of an NoC router are presented. In the second section of this article, the utilized routing algorithm is briefly analyzed. In implementation, XY routing algorithm is utilized [6-7]. In the third section, the wormhole switching which is used in implementation is reviewed [8, 9]. In the fourth of this article, the utilized traffic pattern is briefly explained. In the fifth section which considers being the main body of this article, handshaking communication mechanism is introduced and analyzed. In this section, the structure of information packets, router function and different states of the router are analyzed. Furthermore, the practical results of implementation and synthesis of this routing are presented in the final section of this article. In this routing, handshaking communication protocol is utilized to interconnect different processing elements.

## 2. The Utilized Routing Algorithm

The utilized topology for implementation is an  $n \times n$  regular two dimensional mesh. A sample of this topology is shown in Figure 1 shows a NoC.

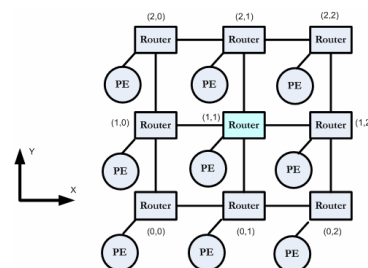


Figure 1. A regular 3x3 mesh topology

The features which are shown in rectangles represent NoC routers and those which are shown in circles represent the processing elements of this network. By the use of communication links and routers, these processing elements which are connected to each other transfer information. Routers are named based on their position in coordinate system. Router ports are also named based on their geographical direction.

However, as it is shown in Figure 1, the number of the ports connected to each other is different due to its position in topology. For example, the router which is placed in the northeast of topology in  $2 \times 2$  coordinates  $([2,2])$ , possesses 3 ports and the router in the center of the topology in  $1 \times 1$  coordinate  $([1,1])$ , has 5 ports.

For n-dimensional mesh topologies in NoCs, dimension order routing produces deadlock-free routing algorithms. These algorithms are very popular, like XY routing (for 2-D mesh). The routing algorithm which is used in this design is a version of XY algorithm. This algorithm is deterministic algorithm in which packet takes routing in one dimension and it continues till this packet attains the desired coordinate in that dimension. Then routing is fulfilled in the same way. This method warrants no deadlock to occur [8, 9]. In this algorithm, according to the coordinates of each router and destination address, routing takes place first in X direction and then in Y direction. This algorithm warrants dead-lock free routing. However, it is sensitive to link destruction and may not be able to adopt a substituting router. It is due to the fact that these types of algorithms adopt routing only based on the source-destination address of packets. Therefore, two packets with the same source and destination address necessarily cross the same route and do not consider the momentary traffic in the route.

### 3. The Utilized Switching

The need to buffer complete packet within a router can make it difficult to construct low area, compact and fast routers. In implementation, wormhole switching is used which is utilized in almost all of NoCs [8].

In wormhole switching message packets are also pipelined through the network. A message packet is broken up into flits that the flit is the unit of message flow control. Therefore, input and output buffers at a router are typically large enough to store a few flits [9].

As we said, in this switching, message packets are divided into equal smaller sections named as flit. Flits are concurrently transferred in the network. Therefore if 16-bit flits are ready to be transferred, 32 signals between two routers are considered to transfer the flits.

16 signal for sending and 16 signal for receiving. In this way, flits are transferred in parallel. Other switching techniques are not commonplace in NoCs usages. For instance, circuit switching technique due to its low performance contradicts with power and performance parameters, similarly packet switching as a result of its big buffers requirement shows the same contradiction.

## 4. The Utilized Traffic Pattern

The traffic model is one of the important parameters in evaluating the latency time of interconnection networks. These models are produced according to the application programs which are run on the machine. In different applications, different models are used. Traffic models are defined according to three parameters [9]: a) The entrance time to networks b) Message Length and c) Address distribution type.

### 4.1 The uniform traffic model

The uniform traffic model is the simplest traffic model which is used in most of evaluations. In this model, each node sends message to the other nodes in network with equal probability. For example in a  $6 \times 6$  mesh topology, each node sends message to the other nodes with the probability of %2.85. All source or destination nodes are selected with equal probability. The selection of source and destination node for each message will be independent from other messages [9].

## 5. Asynchronous Communication Mechanism

For making interaction between routers, handshaking communication protocol is utilized in case the data is put on the line; the existence of the data is informed to the next router. Next router takes the data from the line and transmits its confirmation to the sender router. So in addition to the flits sending and receiving channels, TX, ACK-TX, RX and ACK-RX signals are required. TX base is the output and whenever the data is ready in the output port, this base equals to one and waits for ACK-TX to be equal to one. Likewise each input port after finding the RX input base to be one, reads the data on this port and equals the ACK-RX output base to one. The link of two ports from two neighbor routers is shown in Figure2.

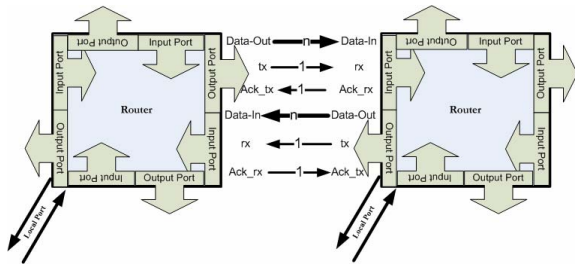


Figure 2. Communication signals in handshake protocol between two NoC routers

### 5.1 The structure of information packets

In each communication standards, the communication payload contains a series of control fields. These fields can be put in the main frame as the redundant fields in order to increase the controllability, fault tolerance, security and some other issues like these. In our intercommunication protocol, flits are used to structuralize. A flit structure is considered in the way that the first bit shows the flit to be the header-trailer or the data. When the first bit equals one, this flit is a header or trailer. In this case, the 2<sup>nd</sup> bit determines which one is the header and which one is the trailer. This representation is shown in Table I.

TABLE I. THE DEFINED PROTOCOL THAT CHARACTERIZE THE FLIT TYPE

First bit	Information type	Second bit	Information type
0	Data	*	Data
1	Header/Trailer	0	Trailer
		1	Header

Header flit contain source and destination address and if it is needed, the length of packet based on the number of flits. The structure of packed based on flits is shown in Figure 3. The first flit is the header flit containing the destination address. The next part is the data flit and final part is the trailer flit.



Figure 3. Transfer packet structure

### 5.2 Routing function

Each router by receiving the header flit from input, accomplish routing and updates routing tables according to its source and destination addresses based on XY algorithm.

Henceforth, all of the flits take routing based on the tables till receiving the final flit (trailer). Routing tables conclude two tables: routing table and output table. The first table represents the out port for each input and the second represents the state of each out port (busy or free). In Figure4, you can see an NoC central router in mesh topology. The central router has 5 in/out port. The local port is utilized to connect the correspondent circle to the processing element (IP block) and other ports are for connecting to other routers.

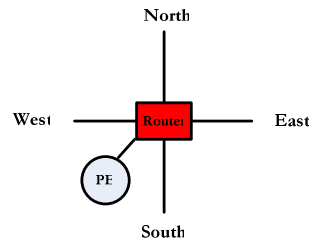


Figure 4. Central NoC router in mesh topology with its ports

The main point here is that the correspondent circle with this routing should have the same interface to be able to use this routing. The internal sight of the features of a routing with their details is not shown and only the main features are considered (in Figure 5).

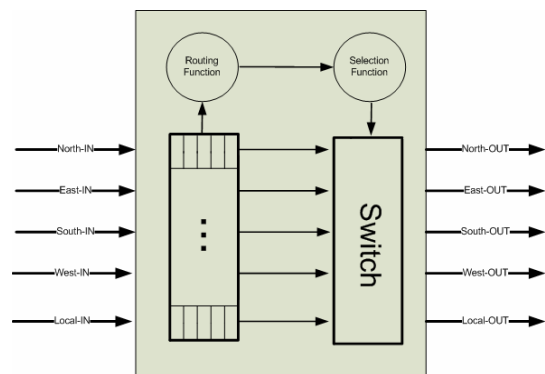


Figure 5. A NoC router with its main components

Routing function feature takes the charge of routing based on routing algorithm and selection function feature under takes the responsibility of choosing out

port in competition circumstances based on the defined priority mechanism. In our designing, mechanisms is implemented by the software in the manner that it gives priority to input port and whatever an input port has a higher priority. It selects its desired output port faster. However, we should consider that competition circumstance only take place when in one moment, there is a request from two input port for one output port.

Our fulfilled designing is implemented by the use of VHDL hardware describing language. In order to router implementation, one entity is designed for whole routing. In code segment of Figure 6, size and type of input/output port are shown.

```
Entity router is
Port(
Clock: in  std_logic;
Reset: in  std_logic;
Data_in:   in  arrayPortsRegisters;
Rx:        in  PortsRegisters;
Ack_rx:    out PortsRegisters;
Data_out:  out arrayPortsRegisters;
Tx:        out PortsRegisters;
Ack_tx:    in  PortsRegisters);
End router;
```

Figure 6. Size and type of input-outputs

Types of `arrayPortsRegisters` and `PortsRegisters` signals are defined in one packet. In order to implement, we defined a machine of definite state for input which you can see in Figure 7.

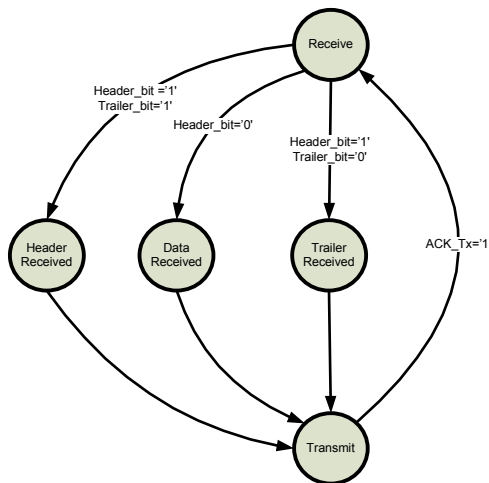


Figure 7. Finite State Machine for flit and router status analyze

### 5.2.1 Received state

In this state, the routing await for its Rx base to be one. In case this happens, firstly the data in Data-in is need and then the correctness of this data is examined. In case of being correct, ACK-RX equal one. Then the next state is defined according to the header/trailer bit.

### 5.2.2 Header received state

In this state, the appropriate output port is defined based on the source and destination addresses and output table. Then routing table and output table are updated. Finally we alter routing state to transmit state.

### 5.2.3 Trailer received state

In this state, after the destination port is determined by the routing table, this table of output table is updated. In order to do this, the home correspondent with the input is equaled to NO PORT and also the output port state in output table is equaled to Free.

### 5.2.4 Data received state

In this state, after finding the output port by routing table, the received flit is put in the output port.

### 5.2.5 Transmit state

In this state, after placing the flit in the output port and equaling the desired output port TX base to one, we await for receiving ACK-TX and after it's receiving we equal TX to zero and turn back to the received state.

## 6. Experimental results

A testbench is considered to test the routing which alternating enters some packets from input port to routing and saves the output port packets. This simulation is fulfilled by the use of the Modelsim software and uniform traffic model as utilized for packet injection to network [9]. In the best state, receive time and Header-Received, Trailer-Received and Data-Received states are one clock cycle and Transmit time is two clock cycles which is a good scale. After simulation, the beginning routing synthesis operation are fulfilled by the use of Leonardo spectrum software and its synthesis results are carried out on ASIC and FPGA that you can see in the following tables.

Total reports of router synthesis on FPGA (SPARTAN3E) have been showed in Table II. For instance, the sum of the all needed gates is 544. The operating frequency and data arrival time of this router in FPGA have been estimated 153.0 MHz and 5.81 ns.

TABLE II. TOTAL ACCUMULATED AREA IN SPARTAN3E

Number of ports	102
Number of nets	881
Number of instances	830
Number of references to this view	0
Number of BUFGP	1
Number of Dffs or Latches	176
Number of Function Generators	548
Number of IBUFG	51
Number of MUXF5	4
Number of OBUF	50
Number of gates	544
Number of accumulated instances	830

Table III shows the synthesis results (resource utilization percentage) of the NoC router in FPGA. In this table, we see the available and utilized resource of SPARTAN 3E platform.

TABLE III. 8 BIT NOC ROUTER RESOURCE UTILIZATION PERCENTAGE OF SPARTAN 3E

Resource	Used	Avail	Utilization
IOs	101	194	52.06%
Global Buffers	1	24	4.17%
Function Generators	548	21712	2.52%
CLB Slices	274	8672	3.16%
Dffs or Latches	176	22100	0.80%
Block RAMs	0	28	0.00%
Block Multipliers	0	28	0.00%
Block Multiplier Dffs	0	2016	0.00%

Table IV shows the total statistical information of 16-bit router synthesis on FPGA.

TABLE IV. TOTAL ACCUMULATED AREA IN SPARTAN3E 16BIT

Number of ports	182
Number of nets	1356
Number of instances	1264
Number of references to this view	0
Number of BUFGP	1
Number of Dffs or Latches	256
Number of Function Generators	824
Number of IBUF	91
Number of MUXF5	2
Number of OBUF	90
Number of gates	816
Number of accumulated instances	1264

Table V shows the percentage of resource utilization of 16-bit router from FPGA (SPARTAN3E model). The

operating frequency and data arrival time of this router in FPGA have been estimated 132.2 MHz and 6.84 ns.

TABLE V. PERCENTAGE OF RESOURCE UTILIZATION OF SPARTAN 3E16 BIT

Resource	Used	Avail	Utilization
IOs	181	194	93.30%
Global Buffers	1	24	4.17%
Function Generators	824	21712	3.80%
CLB Slices	412	8672	4.75%
Dffs or Latches	256	22100	1.16%
Block RAMs	0	28	0.00%
Block Multipliers	0	28	0.00%
Block Multiplier Dffs	0	2016	0.00%

Figure 8 shows total accumulated area comparison between 8 bit and 16 bit flit sizes (bandwidth) in SPARTAN 3E.

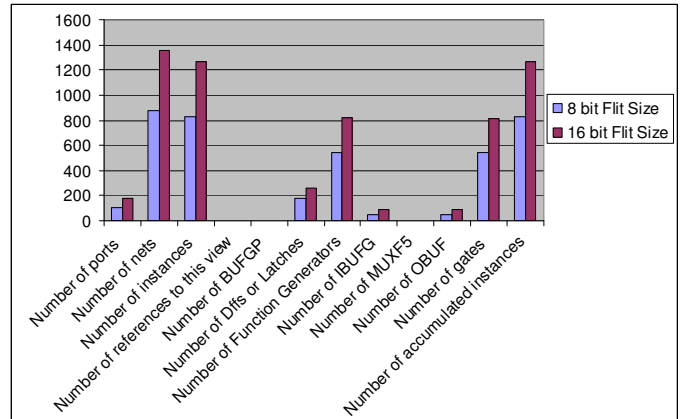


Figure 8. Comparison of Total Accumulated Area in SPARTAN 3E between a 8 bit flit size router and 16 bit

Table VI shows the total statistical information of our 8 bit router synthesis on ASIC (TSMC 13U). For instance, the sum of the all needed gates is 10936. The operating frequency and data arrival time of this router in FPGA have been estimated 303.2 MHz and 3.01 ns.

TABLE VI. TOTAL ACCUMULATED AREA IN 8BIT TSMC13U (ASIC)

Number of ports	102
Number of nets	847
Number of instances	716
Number of references to this view	0
Number of gates	10936
Number of accumulated instances	716

Table VII shows the total statistical information of our 16 bit router synthesis on ASIC. For instance, the sum of the all needed gates is 14885. The operating frequency and data arrival time of this router in FPGA have been estimated 299.2 MHz and 3.03 ns.

TABLE VII. TOTAL ACCUMULATED AREA IN 16 BIT TSMC13U (ASIC)

Number of ports	182
Number of nets	1082
Number of instances	879
Number of references to this view	0
Number of gates	14885
Number of accumulated instances	879

Figure 9 shows total accumulated area comparison between 8 bit and 16 bit flit sizes (bandwidth) in TSMC 13u.

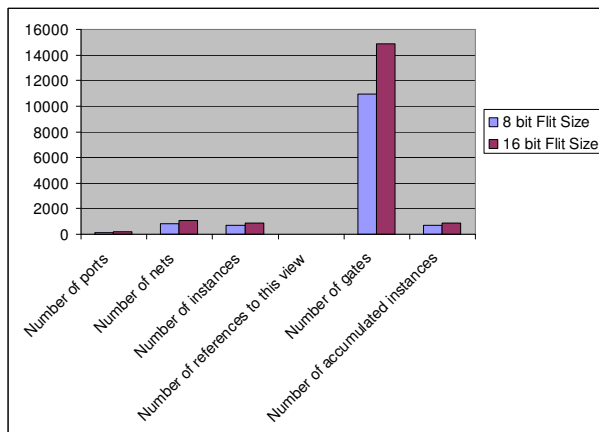


Figure 9. Comparison of Total Accumulated Area in TSMC13u between a 8 bit flit size router and 16 bit

The presented figures and tables, show the bandwidth effect in router elements area. So the bandwidth increase factor has not linear impact in area.

## 6. Conclusion

In the presented article, designing and implementing an NoC router were analyzed. In today's very integrated digital systems, power and performance correspond closely with each other. In engineering designing, always need to analyze the loss for each option for each optimum solution. One of the features, which directly influence on power, is the communication issue in NoCs. In this article not only we used an asynchronous communication mechanism based on handshaking to transfer information (which implies low power consumption and developing capability) but also by

using statistical data, we showed that this designed router occupies very little space.

Scalable design of this router leads to easy and efficient addition of new capabilities like 8-bit and 16-bit bandwidth. The resource utilization of this router is more efficient than similar implementation on FPGA and ASIC platforms.

## 7. References

- [1] L. Benini and D. Bertozzi, Network-on-chip architectures and design methods, IEE Computers and Digital Techniques, Volume 152, Issue 2, p. 261-272, March 2005
- [2] X. Chen and L. Shiuian Peh, Leakage Power Modeling and Optimization in Interconnect Networks, ISLPED'03, August 25-27, 2003, Seoul, Korea.
- [3] P. Pratim Pende, C. Grecu, M. Jones, A. Ivanov and R. Saleh, Performance Evaluation and Design Trade-Offs for Network-on-Chip Interconnect Architectures, IEEE TRANSACTIONS ON COMPUTERS, VOL. 54, NO. 8, pp:1025:1040, AUGUST 2005
- [4] N. Easley and L. Shiuian Peh, High-Level Power Analysis for On-Chip Networks, CASES'04 September 22-25, 2004, Washington, DC, USA
- [5] G.M. Chiu, "The odd-even turn model for adaptive routing," IEEE Transactions on Parallel and Distributed Systems, vol. 11, pp. 729-38, 2000.
- [6] R. Holtsmark, M. Palasi and S. Kumar, Deadlock Free Routing Algorithms for Mesh Topology NoC Systems with Regions, Digital System Design: Architectures, Methods and Tools, 2006. DSD 2006. 9th EUROMICRO Conference on Volume, Issue, 0-0 0 Page(s):696 - 703
- [7] Zh. Xiaohu and C. Yang, W. Liwei, A Novel Routing Algorithm for Network-on-Chip, Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on Volume, Issue, 21-25 Sept. 2007 Page(s):1877 - 1879
- [8] J. Duato, "A new theory of deadlock-free adaptive routing in wormhole networks," IEEE Trans. On Parallel and Distributed Systems, vol. 4 No. 12, pp. 1320-1331, December 1993
- [9] W. Hsh, Performance issues in wire-limited hierarchical networks, PhD Thesis, University of Illinois-Urbana Champaign, 1992.