

## Designing and Implementation of a Network on Chip Router Based on Handshaking Communication Mechanism

Seyyed Amir Asghari, Hossein Pedram, Mohammad Khademi, and Pooria Yaghini

Amirkabir University of Technology, Computer Engineering and Information Technology, Tehran, Iran

**Abstract:** Power and performance play a significant role since the size of technology to build modern digital systems are reduced. Therefore, in designing these systems, all of the designing features shall somehow acquire their confirmation from the standpoint of these two parameters. One of the important features is communication. Communication portion in the power consumption of System on Chip can be up to 50% of the whole power consumption of the chip. This deems to be more important for Network on Chips which center around an intercommunication networks. In this article, designing and implementation a NoC router based on handshaking communication protocol are presented.

**Key words:** Power . Performance . Intercommunication network . Handshaking communication protocol . System on chip . Network on chip

### INTRODUCTION

Power and performance are two essential features which are corresponded with each other, produce main concerns in designing and implementation. Nowadays, very large integrated digital systems [1-4] (System 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, designing and implementing 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 section which consider to be 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,

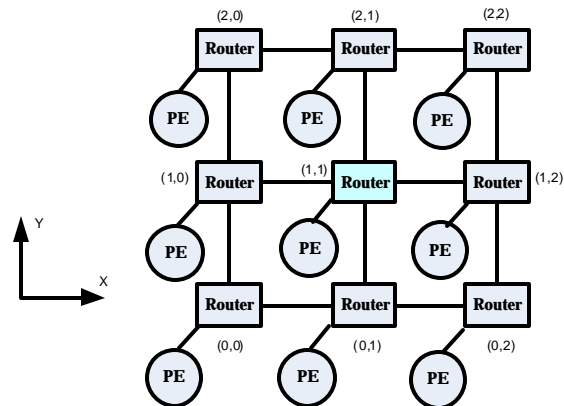


Fig. 1: A regular 3×3 mesh topology

the practical results of implementation and synthesis of this routing are presented in the final section of this article. In this routing, handshaking communicative protocol is utilized to interconnect different processing elements.

In our designing, a dead-lock free routing algorithm (XY routing) and wormhole switching are also used.

The utilized topology for implementation is a 3×3 regular two dimensional mesh. This topology is shown in Fig. 1 shows a NoC.

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 are connected to each other and 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 Fig. 1, a number of ports that are connected to each other are different due to their positions in topology. For example, the router which is placed in the northeast of topology in  $2 \times 2$  coordinate  $([2, 2])$ , possesses 3 ports and the router in the center of topology in  $1 \times 1$  coordinate  $([1, 1])$ , has 5 ports.

### THE UTILIZED ROUTING ALGORITHM

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 which packet takes routing in one dimension and it continues till this packet attains the desired coordinate in that dimension. After that, routing is continued to do the same procedure in the other dimension. This method warrants no deadlock to occur [8, 9].

According to the coordinate of each router and destination address, routing takes place first in X direction and then in Y direction. As mentioned above, the algorithm warrants dead-lock free routing. However, it is possible that the algorithm is unable to adopt a substituting router due to sensitivity of link destruction. These types of algorithms adopt routing only based on the source-destination address of packets. Therefore, two packets with the same source and

destination addresses necessarily cross the same route and do not consider the momentary traffic in the route.

### 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, each packet is 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 signals for sending and 16 signals 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.

### HANDSHAKING COMMUNICATION MECHANISM

For making interaction between routers, handshaking communication protocol is utilized in case the data is put on the line; the existence of 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

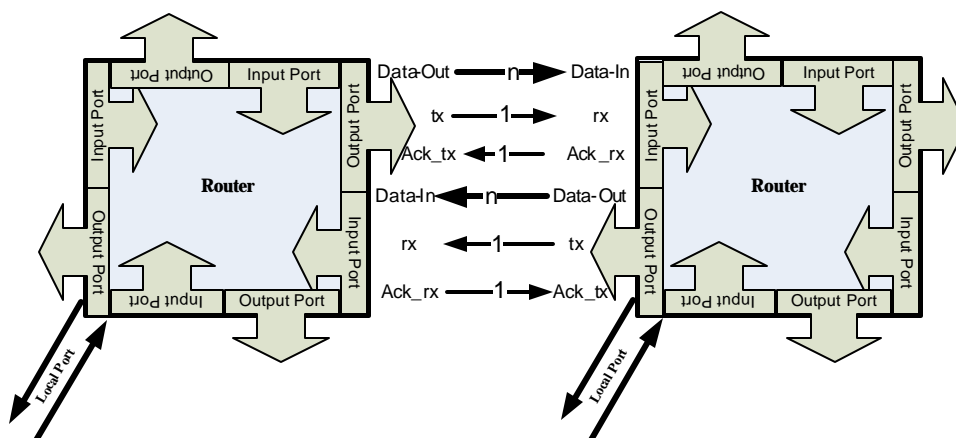


Fig. 2: Communication signals in handshake protocol between two NoC routers

Table 1: The defined protocol that characterize the flit type

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



Fig. 3: Transfer flit structure

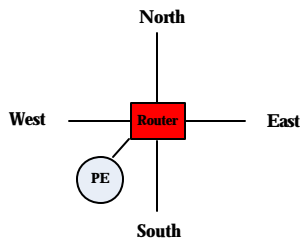


Fig. 4: Central NoC router in mesh topology with its ports

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 between two ports from two neighbor routers is shown in Fig. 2.

**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 1.

Header flit contains source and destination address and if it is needed, it will contain the length of packet. The structure of packet based on flits is shown in Fig. 3. The first flit is the header flit that contains the destination address. After that the data flit is the next and finally is the trailer flit.

**Routing function:** Each router by receiving the header flit from input, accomplish routing and updates routing tables according to its source and destination address and it 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 in utilized to connect the correspondent circle to the processing element (IP block) and other ports are for connecting to other routers.

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 (Fig. 5).

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, this port 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 design 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 Fig. 6, size and type of input/output port are shown.

Types of array Nport and regNport signals are defined in one packet. In order to implement, we defined a machine of definite state for input which you can see in Fig. 7.

**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.

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

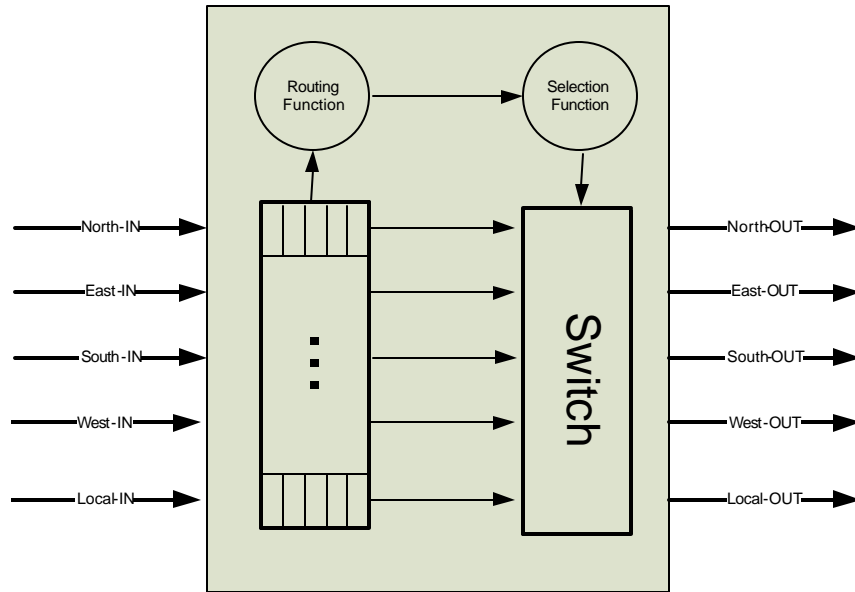


Fig. 5: A NoC router with its main components

```

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

Fig. 6: Size and type of input-outputs

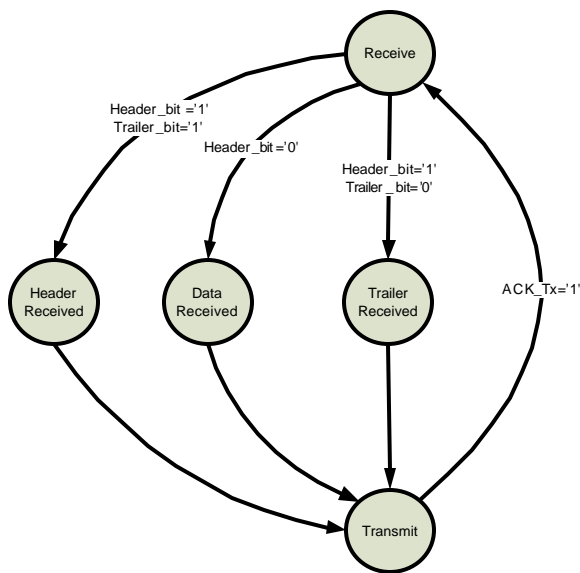


Fig. 7: Finite State Machine for flit and router status analyze

**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.

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

**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.

## RESULTS

A testbench is written 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 modelsim software and uniform traffic pattern is utilized for packet injection to network [10].

In the best state, receive period and Header-Received, Trailer-Received and Data-Received states are one clock cycle and Transmit period is two clock cycles which is a good scale.

After simulation, the beginning routing synthesis operations are fulfilled by the use of Leonardo spectrum software and its synthesis results are carried out on

Table 2: Logic circuit statistical information of synthesis on ASIC

Cell name	Library name	Input -output	Area (number of gates)
AN2T0	scl05u	5×2	9
AN5T0	scl05u	9×4	38
AO1I0	scl05u	6×20	124
AO1I1	scl05u	6×2	13
AO1I2	scl05u	7×1	7
AO2I0	scl05u	8×61	464
AO2I2	scl05u	8×1	8
AO2L0	scl05u	8×2	15
AO2L1	scl05u	8×1	8
AO3I0	scl05u	8×3	23
AO3I2	scl05u	8×1	8
AOA4I0	scl05u	8×11	84
FD1H0	scl05u	9×140	1260
FD1H1	scl05u	10×1	10
FD1I0	scl05u	11×23	244
FD1I1	scl05u	11×11	121
FD1I2	scl05u	13×1	13
IV1N0	scl05u	3×68	211
IV1N1	scl05u	3×6	19
IV1NP	scl05u	4×2	8
MX2L0	scl05u	6×8	50
ND2N0	scl05u	5×171	770
ND2N1	scl05u	5×2	9
ND3N0	scl05u	6×75	465
ND4N0	scl05u	8×38	289
NR2R0	scl05u	5×80	360
NR2R1	scl05u	5×14	66
NR2R2	scl05u	5×5	26
NR3R0	scl05u	6×1	6
OAI1A0	scl05u	6×29	180
OAI1A2	scl05u	3×7	20
OAI2N0	scl05u	8×14	106
OAI2N2	scl05u	8×3	25
OAI3N0	scl05u	8×59	448
OAI3N1	scl05u	8×6	48
OAI3N2	scl05u	8×1	8
OAI3R0	scl05u	8×2	15
OAI3R2	Scl05u	8×1	8
OAI5N0	Scl05u	11×11	11

Table 3: Total statistical information of synthesis on ASIC

Component	1 XPHU
Port	102
Net	1037
Instance	884
Gate	5701
Accumulated Instances	884
Clock	159.9 MHz
Data Arrival Time	5.56 ns

Table 4: Logic circuit statistical information of synthesis on FPGA-XILINX (VirtexII-Pro 2VP70ff1704 model)

Cell name	Library name	Input-output	Area (number of gates)
BUFGP	xcv2p	×1	-
FDCE	xcv2p	1×30	30 Dffs or Latches
FDE	xcv2p	1×141	141 Dffs or Latches
FDPE	xcv2p	1×5	5 Dffs or Latches
GND	xcv2p	×1	-
IBUF	xcv2p	×51	-
LUT1	xcv2p	1×17	17 Function Generators
LUT2	xcv2p	1×80	80 Function Generators
LUT3	xcv2p	1×109	109 Function Generators
LUT4	xcv2p	1×410	410 Function Generators
MUXF5	xcv2p	1×98	98 MUXF5
OBUF	xcv2p	×50	-
VCC	xcv2p	×1	-

Table 5: Total statistical information of synthesis on FPGA-Xilinx

Component	1 XPHU
Port	102
Net	1046
Instance	994
Number of Dffs or Latches	176
Number of Function Generators	616
Number of MUXF5	98
Number of gates	616
Number of accumulated instances	994

ASIC and FPGA that you can see them in the following tables.

Table 2 shows the synthesis results of the NoC router which is written in VHDL hardware language on ASIC. In this table we can see the library file which is used in synthesis procedure (scl05u). Also input-output ports (needed pin) and gates which are needed for our router in this ASIC design have been showed.

Table 3 shows the total statistical information of our router synthesis on ASIC. For example, In Table 2, we observed the number of gates in detail and total reports have been showed in Table 3. For instance, the sum of the all needed gates is 5701. The operating frequency and data arrival time of this router in ASIC have been estimated 159.9 MHz and 5.56 ns.

Table 4 shows the synthesis results of the NoC router in FPGA (VirtexII-pro). In this table, we see the library file which is used in synthesis procedure (xcv2p). Also input-output ports (needed pin) and gates which are needed for our router in this FPGA design have been showed.

Table 6: Percentage of resource utilization of FPGA-Xilinx

Resource	Used	Available	Utilization percentage
I/O	102	996	10.24
Function Generator	616	66176	0.93
CLB Slice	308	33088	0.93
Dff ?? Latch	176	69164	0.25
Block RAM	0	328	0.00
Block Multiplier	0	328	0.00

In Table 5 we can see the total statistical information of our router synthesis on this FPGA model. In this table total report have been showed, for example the sum of the all needed gates is 616 (this number is the sum of input-output has been brought in Table 4). The operating frequency and data arrival time of this router in ASIC have been estimated 159.9 MHz and 5.56 ns.

We can estimate the designed router area utilization of FPGA model target. As you see in Table 6, the designed router has a very small size on this chip.

### CONCLUSION

In engineering design, always analyze the loss for each option for each optimum solution is needed. Nowadays, in very integrated digital systems, power and performance correspond closely to each other. One of the features, which directly influence on power, is the communication issue in NoCs.

In this paper, design and implementation of an NoC router are analyzed. We have used an asynchronous communication mechanism based on handshaking to transfer information which implies low power consumption and scalability features. Also by using of the statistical data, we showed that this designed router occupies a very little space. This router has the synthesizability feature, because it synthesized in both ASIC and FPGA and it is showed that our design, utilizes a very small areas and a few resources.

### REFERENCES

1. Benini, L. and D. Bertozzi, 2005. Network-on-chip architectures and design methods. *IEE Computers and Digital Techniques*, 152 (2): 261-272.
2. Chen, X. and L. Shiuan Peh, 2003. Leakage Power Modeling and Optimization in Interconnection Networks. *ISLPED'03*, August 25-27, 2003, Seoul, Korea.
3. Pratim Pende, P., C. Grecu, M. Jones, A. Ivanov and R. Saleh, 2005. Performance Evaluation and Design Trade-offs for Network-on-Chip Interconnect Architectures. *IEEE Transactions on Computers*, 54 (8): 1025-1040.
4. Eislely, N. and L. Shiuan Peh, 2004. High-Level Power Analysis for On-Chip Networks. *CASES'04* September 22-25, Washington, DC, USA.
5. Chiu, G.M., 2000. The odd-even turn model for adaptive routing. *IEEE Transactions on Parallel and Distributed Systems*, 11: 729-738.
6. Holsmark, R., 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*, pp: 696-703.
7. Xiaohu, Zh., C. Yang and W. Liwei, 2007. A Novel Routing Algorithm for Network-on-Chip, *Wireless Communications, Networking and Mobile Computing*, 2007. *WiCom 2007. International Conference*, pp: 1877-1879.
8. Duato, J., A new theory of deadlock-free adaptive routing in wormhole networks. *IEEE Trans. On Parallel and Distributed Systems*, 4(12): 1320-1331.
9. Duato, J., S. Yalamanchili and L. Ni, 2003. *Interconnection Networks, An Engineering Approach*, pp: 55-57 and 146-147 by Elsevier Science.
10. Hsh, W., 1992. Performance issues in wire-limited hierarchical networks. Ph.D Thesis. University of Illinois-Urbana Champaign.