Performance Improved Router Architecture for Bidirectional NoC Using Flit Level Speed up Scheme

[1]Reshma P Vengaloor, [2]Karthika Manilal
[1] PG Student [VLSI and ES] [2] Assistant professor,
Department of ECE, TKM Institute of Technology, Kollam
[1]reshma.vengaloor@gmail.com [2] karthikamanilal@gmail.com

***Abstract*— In this paper, a Bidirectional Network on Chip (BiNoC) using flit level speed up scheme (FSNoC) is proposed for enhancing the performance of on chip communication, by optimizing the available bandwidth. To achieve this goal, a flit level speed up scheme using self reconfigurable bidirectional channel is developed. For interrouter bandwidth utilization, a distributed channel configuration scheme is developed, which dynamically changes the link direction. Intrarouter bandwidth utilization is possible by allowing multiple flits from same packet to use the idle channel bandwidth. In this way the effective channel bandwidth between two routers can change adaptively depending on the network traffic. An input buffer architecture, which supports reading /writing two flits from/to same virtual channel at the same time and a switch allocator for supporting the flit level parallel arbitration, is also designed. A virtual cut through routing is used for router design, which helps to reduce the packet delay and the memory requirement. So the data to be transferred is divided into flits of equal size. Routing decisions for flits are made by using XY routing algorithm. FSNoC provide better bandwidth utilization, low latency and reduction in area occupancy. FSNoC is designed using VHDL language and synthesized in Xilinx ISE Design Suit 13.2 and it is simulated in Model Sim SE 6.3f.**

***Index Terms* — On chip communication, Inter connection networks, Network on Chip (NoC), Bidirectional channel, Flit level speed up.**

I. INTRODUCTION

With the vigorous advancement in the semiconductor processing technologies, the chip integration has reached a stage, where the bus based architecture became unable to handle the communication among computational units. Consequently, Multi Processor System on Chip (MPSoC) and Chip Multi Processor (CMP) were adopted as new platforms. It is very important to handle the communication among on chip resources efficiently. These requirements lead to the invention of more flexible and scalable on chip communication scheme known as Network on Chip (NoC).

Recently Networks on Chip (NoC) plays vital role in development of VLSI. Increased levels of integration cause the systems with different types of applications, which is of having its own I/O characteristics. But it is less desirable to use the buses for establishing the communication among processors. Consequently, network technology is developed. It is possible to use all the links in NoC simultaneously for transmission of data. So it helps provide a high level of parallelism. NoC platform is scalable and it has a potential to keep up with the pace of technology advances.

Fig. 1 shows the basic NoC architecture, which basically includes three building blocks: Processing Elements (PE), routers and Network Interface (NI). The latter two will comprise the communication architecture. NI will packetize the data. PE is connected to NI which connects the PE to local router. Each PE is attached to NI which connects the PE to a local router. Instead of buses or point- to - point communication NoC utilizes routers for sending and receiving packets between processing elements. There are five ports in a generic router architecture: from / to four cardinal directions (EAST, WEST, SOUTH and NORTH) and from / to local Processing Element (PE). The main building blocks of a generic router includes input buffer, routing computation logic, virtual channel allocator, switch allocator and cross bar. For improving the performance, routers process the packet with four pipeline stages (Fig. 2): routing computation (RC), VC allocation (VC), switch allocation (SA), and switch traversal (ST). By looking the destination address of a packet RC stage will direct it to the proper output port. Then VA will allocate the available virtual channel (VC) of the downstream router. The SA will arbitrate the input and output ports of the crossbar and successfully granted flits traverse the crossbar during the ST stage.

In an NoC, the neighbouring routers are connected through a pair of unidirectional channel. Each of them are hard wired to handle the outgoing or incoming traffic. But the use of unidirectional channel causes some disadvantages. The handling of traffic only in a single direction causes problems such as performance degradation and ineffective resource utilization. The inefficient link band width utilization increases the system latency and limits the throughput. The problems such as channels overflow and data loss may occur in conventional NoCs due to heavy traffic in single direction.

In this paper, the above mentioned disadvantages are reduced by introducing bidirectional NoC (BiNoC) architecture. In a BiNoC, each communication channel allows itself to be dynamically reconfigured for transmitting flits in either direction. Interrouter bandwidth and intrarouter bandwidths are the two types of bandwidths available in router architecture.. For utilizing all these bandwidth, a flit level speed up scheme is introduced. It allows the development of a new input buffer organization and switch allocator for flit level parallel arbitration, which utilizes intrarouter bandwidth. For achieving interrouter bandwidth utilization, a Channel Direction Control (CDC) scheme is introduced. The simulation results shows the better latency and throughput with moderate power and area overhead

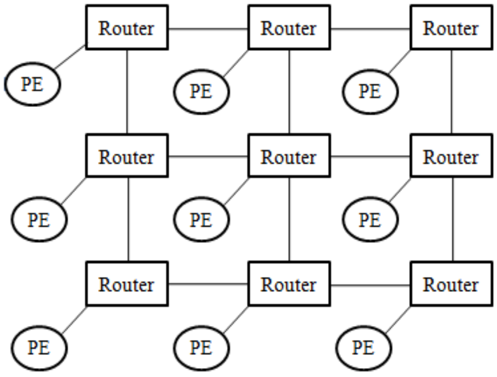


Fig. 1. Basic NoC architecture

II. LITERATURE REVIEW

For the performance improvement in NoC, many approaches have been introduced. A low latency router architecture for NoC is proposed in [1]. A fine grained buffer utilization scheme known as ViChaR is proposed in [2] to dynamically adjust the size of the buffer according to the network traffic. In [3], a dynamic channel width configuration algorithm is proposed for configuring the link direction between two neighbouring routers for a given application. An Adaptive Physical Channel Regulator (APCR) [4] is proposed in which, the size of each flit will be less than that of a physical channel unit called phit. APCR introduces three regulation schemes: monopolizing, fair sharing and channel stealing. These schemes help to properly regulate the channel utilization at run time and hence avoid the wastage of channel bandwidth. In [5] heterogeneous router architecture is proposed for reducing the delay by combining two flits and send it through the wider link.

All the above approaches were developed for performance enhancement in unidirectional NoC. Apart from these a new progress can be made by utilizing one unidirectional link is more frequently used than the other in opposite direction [6],[8],[9]. This inefficient link direction leads latency and also it limits the throughput. Hence an architecture is proposed in [8], where the link directions are reconfigured at architectural level. In [8] the channel direction is decided at run time depending on the traffic of two routers using either an external bandwidth allocator or that in [8] utilizes a Channel Direction Control (CDC) scheme. For both cases, the channel direction will be reversed if there are two or more packets requesting to traverse along one direction and no packet requesting along opposite direction.

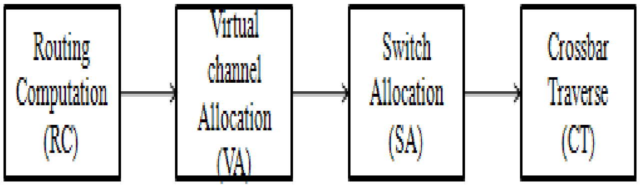


Fig. 2. Four stage pipelined architecture

In [6], the direction of each of the bidirectional channel is determined by a distributed channel direction control protocol known as CDC. CDC is implemented using Finite State Machine (FSM).HP FSM will serve for the master link and LP FSM will serve for the slave link. Three states of the FSM are wait, ready and idle state. CDC controls the channel direction depending on the state of the corresponding FSM. But the use of FSM causes some disadvantages such as delay due to the wait state and area problems.

III. PROPOSED SYSTEM

Further performance enhancement requirement lead to the invention of bidirectional NoC using flit level speed up scheme for NoC, known as FSNoC. Performance enhancement is achieved by the optimization of available bandwidth. For supporting intrarouter bandwidth utilization a new router data path is developed, which includes an input buffer organization and switch allocator. Input buffer will help to support reading and writing two flits from same virtual channel at the same time. Switch allocator will support the flit level parallel arbitration. For the interrouter data path utilization a CDC protocol is developed [7], which avoids the use of FSMs [5]. Instead it uses a Request Extract module (RE) and an Output Width Control module (OWC). This distributed channel configuration scheme adaptively changes the available bandwidth between two routers.

Fig. 3 shows basic bidirectional NoC architecture. In this, both the input and output are reconfigurable. Each of the two bidirectional channels between a pair of routers will determine its own transmission direction based on a distributed channel control protocol. When both the channels have same direction, then it means that two packets should sent concurrently, which effectively double the bandwidth. Within BiNoC, each port can be either an input port or an output port and is free of conflicts. In BiNoC architecture, the number of channel bandwidth available in each output direction is doubled from p to 2p.

A. *Switching technique*

In this project packet switching is used for determining how the data packets move within the router. In packet switching, longer messages to be transferred is segmented into smaller data packets, and then forwarding these packets individually from the sender to the receiver possibly with different routes and delays for each packet. Basically, packet switching schemes are of three types: Store And Forward (SAF), Virtual-Cut-Through (VCT) and Worm Hole (WH) switching.

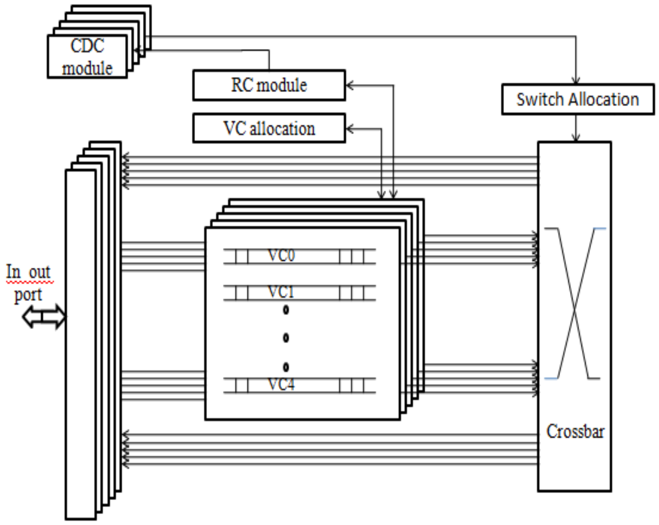


Fig. 3. Basic Bidirectional NoC router architecture

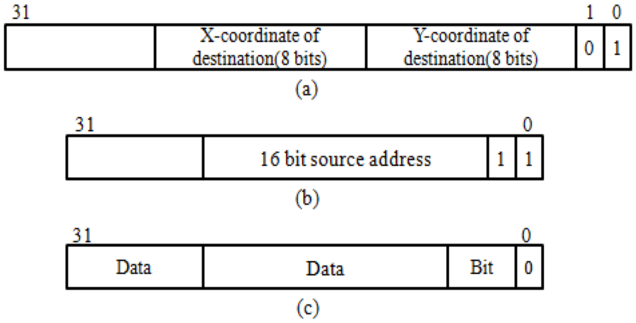

31
1
0
X-coordinate of destination(8 bits)
Y-coordinate of destination(8 bits)
0
1
(a)
31
0
16 bit source address
1
1
(b)
31
0
Data
Data
Bit
0
(c)

Fig. 4. (a). Header Flit (b). Tail Flit (c). Body Flit

B. *Data structure*

The data packet is divided into flits of equal size. Each flit contains 32 bits. A data packet includes header flit, tail flit and body flit. Header flit contains the information about destination address and length of the data. The actual data that is to be transmitted is contained by body flit. Tail flit includes the information about source address. The structure of header flit, tail flit and body flit is shown in Fig. 4.

C. *Network Flow Control*

Network flow control is the routing mode with which the data is transmitted within the router. The flow control method used in this project is virtual cut through. Virtual cut-through is an improved version of store-and-forward mode. As soon as the next router gives a permission, a router can begin to send packet to the next router. The router will store the data until the beginning of data transmission. It is possible to start forwarding the data before the whole packet is received and stored to router. This mode of flow control needs as much buffer memory as store-and-forward mode, but latencies are lower.

D. Routing Algorithm

Routing algorithm is an essential thing to determine the destination to which the flit has to go. XY routing algorithm is considered in this project because of it's ability to avoid dead lock situations. XY routing algorithm routes the data first in horizontal (X Direction) direction and then in vertical direction (Y Direction). XY-coordinates of the router indicates its address.

IV. FSNoC ROUTER ARCHITECTURE WITH CDC MODULE

In FSNoC with bidirectional channels, the direction of both links connecting to same port of the router can be reconfigured as either sending or receiving. Innorder to keep the order of the flits in the same packet, one link can be defined as master and another one as the slave. The master link of the router is also the slave link of its neighbor and vice versa. If there is traffic in both the directions between two.

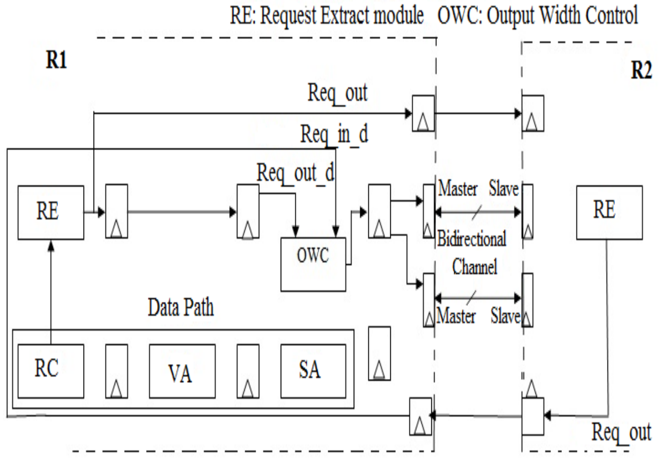


Fig. 5. FSNoC router architecture with CDC module

Routers, the master and slave link work as the sending and receiving links respectively. When transferring two flits from same VC to output direction, the CDC module will always send the first flit through the master and the second flit through the slave link

The channel control scheme for FSNoC is shown in Fig. 5. Instead of two separate FSMs two modules namely, the request extractor (RE) and the Output Width Controller (OWC), are work together with the conventional four pipeline stages in the data path of a router. The RE module monitors the input channel status and generates the pressure signals to the corresponding OWC modules in the current and the neighboring router named as req_out_d and req_out respectively.

OWC module will determine the number of flits that can send at a time by checking the availability of master and slave. To achieve this goal, OWC module will check the pressure signal from RE module of neighboring module in the corresponding direction. If both the master and slave link are available then OWC module will determine the flit width as 2 for transmitting two flits. If only master is available then the OWC module will determine the flit width as 1.

V. DESIGN OF FSNoC ROUTER ARCHITECTURE

Fig. 6 shows the detailed structure of router module for FSNoC. A common universal clock is provided to all the blocks.

A. Port Selector

The flit, which is transmitted from a particular port is selected using a port selector module. The signals te, tw, ts, tn and tl will enable the east, west, south, north and local port respectively. Whenever any of these signals become high, then the corresponding port become active and port selector will forward the data from its active port to output links. One of the output will carry the data from master link and another will carry the data from slave link. The outputs of the port selector are given to the buffer organization. At the starting of packet transmission, the header flit will be transmitted through master link. Header is used for routing computation and hence the output of the port selector, which carries the master data will also be connected to the input of RC module

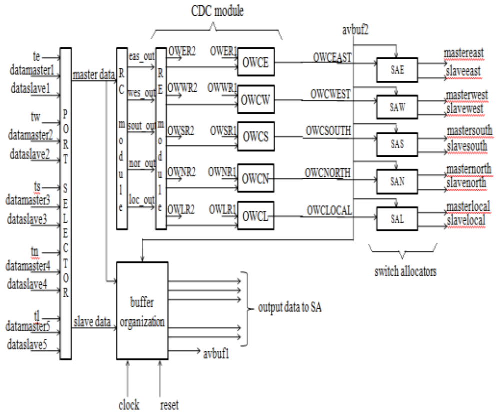


Fig. 6. Internal architecture of FSNoC.

B. Routing Computation (RC) module

Routing computation module will determine the destination of a packet. To achieve this goal, header flit will give as the input to the routing computation module. Routing computation module will check the last two bits of each flit. If it is 01 then it indicates that the flit is a header one, which is used for for routing computation. RC module uses XY routing algorithm for determining the destination port. The local port is set to (1,1). ie, the x coordinate and y coordinate of the local port is 1. RC module will check the bits 10 down to 3 and 18 down to 11 together of header flit for calculating the direction of destination port.

RC module has 5 output ports: eas_out, wes_out, sou_out, nor_out,loc_out. After the routing computation one of these outputs become high. For example, if eas_out is high then it indicates that the flit has to move in the east direction.

C. Request Extract(RE) Module

Request extract module will check the status of the routing computation module. So all the outputs of RC module are connected to the input of the RE module. RE module will activate the OWC module, which is residing in the selected direction. ie, if eas_out is high, then the RE module will generate a pressure signal towards the OWC module in the east direction. Also RE module will generate a pressure signal towards the OWC module of the neighbouring router module. It indicates that, there is a data has to be transmitted from current router to neighbouring router.

D. Output Width Control (OWC) Module

Output Width Control (OWC) module is used to determine the flit width to be transmitted. It has two inputs. One of the input is the pressure signal from RE module of the current router R1 in the active direction and the other is from the RE module of the neighbouring router. This second input will determine whether there is a data transmission from neighbouring router in any of the direction to the current router R1. According to this signal, the OWC module will generate the flit width. OWER1, OWER1, OWSR1, OWNR1, and OWLR1 indicate the pressure signal from neighbouring router's RE to OWC of the current router in its corresponding direction. Any of these signals is 1 is indicating that, there will be a data transmission from R2 to R1. Thus the slave link is not available for the data transmission and hence the flit width will be determined as 1. If any of the signal is 0 then it indicates that, there is no data transmission from R2 to R1. So both the master and slave will be available for data transmission and hence the flit width will be determined as 2.

E. Switch Allocator

There are five inputs provided corresponding to each direction. Switch allocator will determine the flit width and a signal which indicates the buffer availability in neighbouring router as its input. SA (Switch Allocator) helps to allocate the master and slave links to the flits.

VI. ROUTER DATA PATH DESIGN

A router data path is designed to support the efficient flit-level parallel transmission in FSNoC. It includes the design of input buffer organization and switch allocator.

A. Input Buffer Organization

There may two incoming flits belong to same VC sub buffer. Therefore, it is required to read/ write two flits from/ to the same VC at the same time. So a new buffer organization to satisfy read/ write two flits simultaneously is explained below. Fig. 7 shows the structure of input buffer organization.

A demux is provided to select a particular path for flits. Input to the demuxes is coming from port selectors. 1st demux will process the data through slave and the 2nd demux will process the data through master. Three select lines are provided to each demux and they will be selected automatically according to the status of the empty signals. The nth output of the 1st demux and the nth out put of the 2nd demux are connected to the nth flit assembler.

Flit assembler helps to arrange the flits in correct order. The detail of flit assembler is presented in Fig. 8. It is composed of two muxes and a back register. It is designed for in- order assembling of the flits into the sub-buffer of the VC. ie, when two flits are sent over the bidirectional links, the first is always put on the master link of the sender. To assemble the flits correctly, the slave link need to connect to the sub-buffer s0 and master link need to connect to the s1. Innorder to achieve this goal the back register is permanently set to 0. The output of the flit assembler is associated with two passing modules. Passing module will forward the data from flit assembler's output. Whenever the passing module forwards the data, then it will generate a high signal also.

Virtual channel sub buffers are provided for storing blocked flits. There are two sub buffers are associated with the flit assembler. Sub buffers are designed as FIFO buffer. The high signal provided by the passing module will be acting as the write enable of the FIFO buffer. It enables the writing of two flits into the sub buffer. Bufavl2 is the signal, which indicates the availability buffer space in R2. It will also act as the read signal of the FIFO. Bufavlr2= 1 means that, enough space is available in second router. Then it enables the reading of data out of sub buffers. When the data is read out from the buffer, then it will generate a high signal also.

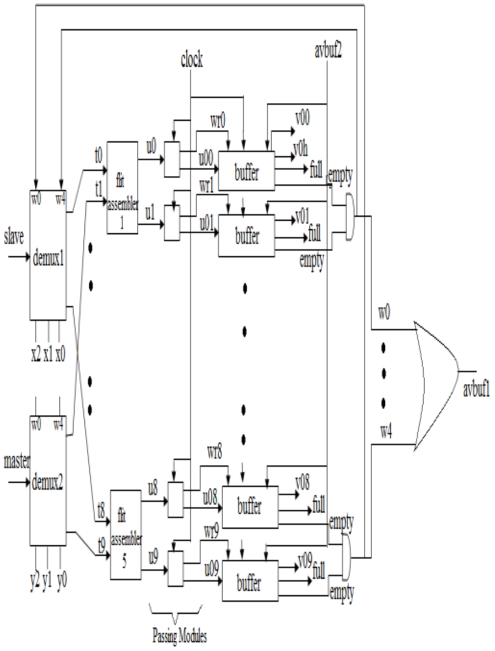


Fig. 7 . Input buffer organization

A. Switch Allocator

Switch allocator helps to allocate the flits to master and slave. There are 5 SA (switch allocator) are provided in each direction. Flit from each sub buffer and high signal associating with each filt is connected to the input of the SA. Output from the OWC module and the signal bufavlr2 are also be the inputs of the SA. According to the status of these signals, SA will allocate the master and slave. The signal bufavlr2 should be high to enable the switch allocator. Because it indicates the availability of buffer space in neighbouring router and so it is possible to forward the data.Thus the switch allocator becomes on and then it will check whether output from output module. If the count generated by OWC module is 2, then SA will identify the availability of both master and slave. So SA will forward two flits to master and slave. i.e. 1st flit to master link and the second flit to slave respectively. If the count generated by OWC module is 1, then the SA will understand that only master is available. Then it will allow to forward only one flit (first flit) through master link. Structure of switch allocator is shown in Fig. 9.

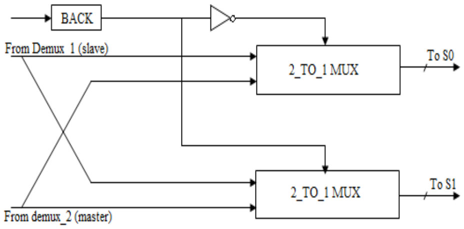


Fig. 8. Flit Assembler

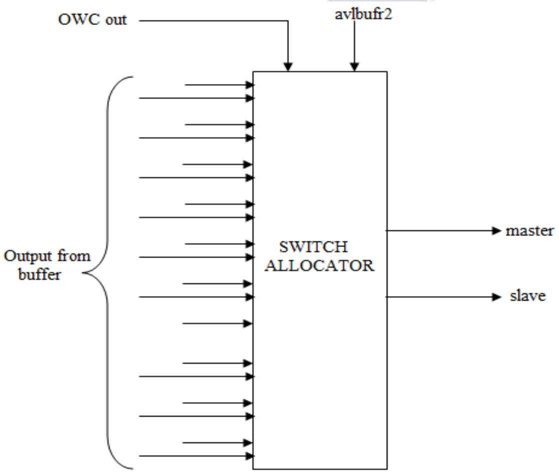
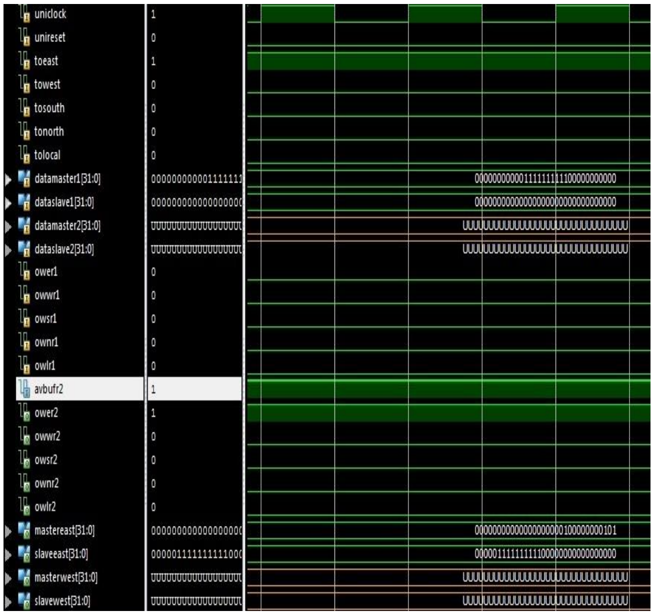


Fig. 9. Switch allocator

VII.EXPERIMENTAL RESULTS

Fig. 10 shows the output of BiNoC router using flit level speed up scheme. In this project we are considering 4 stage pipeline architecture. The router architecture has 5 virtual channel sub buffers. Flit level speed up scheme used in BiNoC allows the transmission of two flits at a time. Hence the latency is reduced. It is about 4-5 cycles. Storage of two flits is provided by sub buffers. Each sub buffer is capable of handling two flits at a time. Whenever the increased traffic occurs in one direction, the channel in opposite direction will be reversed. This also helps to reduce the latency and improve the available band width utilization. This is achieved by the help of link direction control scheme known as CDC protocol. Thus we will get better performance. VHDL language is used to develop the FSNoC router architecture and it is synthesized in Xilinx ISE Design Suit 13.2 and simulated in Model Sim SE 6.3f.



```
uniclock          1
unireset          0
toeast            1
towest            0
tosouth           0
tonorth           0
tolocal           0
datamaster1[31:0] 000000000001111111   00000000000111111111100000000000
dataslave1[31:0]  000000000000000000   00000000000000000000000000000000
datamaster2[31:0] UUUUUUUUUUUUUUUUUU   UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU
dataslave2[31:0]  UUUUUUUUUUUUUUUUUU   UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU
ower1             0
owwr1             0
owsr1             0
ownr1             0
owlr1             0
avbufr2           1
ower2             1
owwr2             0
owsr2             0
ownr2             0
owlr2             0
mastereast[31:0]  000000000000000000   00000000000000000000100000000101
slaveeast[31:0]   000001111111111100   00000111111111110000000000000000
masterwest[31:0]  UUUUUUUUUUUUUUUUUU   UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU
slavewest[31:0]   UUUUUUUUUUUUUUUUUU   UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU
```

Fig. 10. Output of BiNoC router using flit level speed up scheme

VIII. CONCLUSION

In this paper, a flit level speed up scheme is proposed for enhancing the performance of the NoC using self reconfigurable bidirectional link. To support sending two flits at a time, a link direction control scheme known as CDC protocol is proposed, which will determine the link direction dynamically at run time. To allow two flits from same packets to participate in the data transmission, a new input buffer organization and a switch allocator is also customized. It avoids the disadvantages of the unidirectional NoC such as inefficient bandwidth utilization, time delay etc. Also, the FSNoC reduces the area utilization, by using Request Extract (RE) and Output Width Control (OWC) for CDC, as comparing with the BiNoC with CDC using FSMs. From the simulation result it is shown that, FSNoC improves the latency and throughput performance.

REFERENCES

[1] R. Mullins, A. West, and S. Moore, “Low-latency virtual-channel routers for on-chip networks,” in *Proc. 31st Annu. Int. Symp. Comput. Archit. (ISCA)*, 2004, p. 188.

[2] C. A. Nicopoulos, D. Park, J. Kim, N. Vijaykrishnan, M. S. Yousif, and C. R. Das, “ViChaR: A dynamic virtual channel regulator for networkon- chip routers,” in *Proc. 39th Annu. IEEE/ ACM Int. Symp. Microarchit.*, Dec. 2006, pp. 333–346.

[3] [3] J. Meng, C. Chen, A. K. Coskun, and A. Joshi, “Run-time energy management of manycore systems through reconfigurable interconnects,” in *Proc. 21st Ed. Great Lakes Symp. Great Lakes Symp. VLSI (GLSVLSI)*, 2011, pp. 43–48.

[4] [4] L. Wang, P. Kumar, K. H. Yum, and E. J. Kim, “APCR: An adaptive physical channel regulator for on-chip interconnects,” in *Proc. 21st Int. Conf. Parallel Archit. Compilation Techn. (PACT)*, 2012, pp. 87–96.

[5] [5] A. K. Mishra, N. Vijaykrishnan, and C. R. Das, “A case for heterogeneous on-chip interconnects for CMPs,” in *Proc. 38th Annu. Int. Symp. Comput. Archit. (ISCA)*, 2011, pp. 389–400.

[6] [6] Y.-C. Lan, H.-A. Lin, S.-H. Lo, Y. H. Hu, and S.-J. Chen, “A bidirectional NoC (BiNoC) architecture with dynamic self reconfigurable channel,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 30, no. 3, pp. 427–440, Mar. 2011.

[7] [7] Zhiliang Qian, Syed Mohsin Abbas, Chi Ying Tsui, “FSNoC: A Flit-Level Speedup Scheme for Network-on-Chip Using Self-Reconfigurable Bidirectional Channels”, *IEEE Transaction on VLSI Systems*, Aug. 2014.

[8] [8] M. A. Al Faruque, T. Ebi, and J. Henkel, “Configurable links for runtime adaptive on-chip communication,” in *Proc. Design, Autom. Test Eur. Conf. Exhibit. (DATE)*, Apr. 2009, pp. 256–261.

[9] [9] M. H. Cho, M. Lis, K. S. Shim, M. Kinsy, T. Wen, and S. Devadas, “Oblivious routing in on-chip bandwidth-adaptive networks,” in *Proc. 18th Int. Conf. Parallel Archit. Compilation Techn. (PACT)*, Sep. 2009, pp. 181–190.