Implementation of NoC Router/Switch

Vineeta Singh, Student, *MSEE*,
Department of Electrical and Microelectronic Engineering,
Rochester Institute of Technology, New York
vxs9946@rit.edu

***Abstract*— Multiprocessor SoC platforms are coming up as an important trend for SoC design. The constraints of wire design and power are forcing SoCs to adopt new methodologies incorporating parallelism and modularity. Scalable interconnects like NoC possessing different features are being studied by different researchers. Interconnect can be characterized by latency energy dissipation, throughput, required area. NoC basically helps in supporting the integration of SoC which consists of on-chip packet and switched network. The comparisons of performance of different NoC architectures have been done in detail. The interconnect experiences some problem for NoC architecture as buses and point links which are used for the communication between IPs, cannot be enough from NoC performance point of view. For this paper, the author has designed a 5 input (core, north, south, east, and west) NoC architecture with 32-bit port width. Cores can access the network by using proper interfaces and forwarding the packets to destination with the help of multihop routing path. The architecture uses wormhole routing and a crossbar switch is used. Router efficiency is important as it is the main component of NoC. Communication speed can be improved using simple routing design, the way the decoding logic is implemented. A parallel router supporting five requests has been implemented.**

The design has been implemented using the ASIC flow, and simulation results are performed in ModelSim and ISim simulator using Xilinx.

***Index Terms*—Router, NoC, SoC, interconnect**

I. INTRODUCTION

Interconnect topology is one of the important components of the multiprocessor SoC platforms. The SoCs consist of the integration of different IPs which is operating at different frequencies implementing different functions. The design faces some problems because of delays of non-scalable global wires which carry signals across the chip. [1] Global wire delays usually increases exponentially with technology. More than 80 percent of the delay of critical paths is due to interconnects. Hence most of the designs use FIFO buffers to help in the synchronization of data propagating over large distances. The most popular and frequently used arbitrated bus is the one where all the communicating devices are connected using shared medium arbitrated bus and it comes with certain advantages like the simplicity of topology used, extensibility, reduction in cost for area. The parasitic capacitances and resistances can be high for a long bus line, and also every IP block adds its own, thus increasing the propagation delay by quite a lot. Thus it can be inferred that for the bus-based systems, the interconnect architecture can lead to serious problems as the bus bandwidth is shared by all of the attached devices. Thus NoC systems are more preferable as compared to buses, as they provide better functionality with respect to performance, level of parallelism, flexibility and scalability. NoCs help in routing the messages between the IPs and the processor elements using interconnects and routers. The routing algorithm determines the path used for the messages to propagate from the source to destination. The wires used as links help routers to connect different inputs to different outputs in the system. The router architecture implemented over here is for the 32-bit port width and are designed using the crossbar switch which has been implemented using multiplexers, but can also be implemented using tri-state buffers. The designed router is a parallel router supporting five requests at a time. Buffers are added at the input and at the output sides in order to avoid congestion at both sides. The sizes of the buffers used depends on what type of switching technique or routing algorithm has been used and they have a significant effect on the aver overhead and the power consumption. The router complexity depends on the parameters like the topology, the routing algorithm, and the switching technique used. Topology helps in defining the way nodes and channels can be connected. Different types of topologies exist such as Torus, Spin, Octagon, Cliché, Bft etc. The arbiter used decides which request has to be given priority and in what order. The arbiter can be implemented in different ways, however the one used in this design is of fixed priority scheme.

The organization of the paper is as follows: In Section II,

Manuscript received December 16, 2014. This work was done as graduate research paper for the course Design and Test Multi-Core Chips.

The author is a student in the department of electrical and microelectronic engineering, Kate Gleason College of Engineering, Rochester Institute of Technology, NY 14623, USA phone: 585-351-8703; e-mail: vxs9946@rit.edu

different types of switching methodologies have been explained. In Section III, the overall router design is discussed. In Section IV, performance metrics are discussed, Section V presents the router architecture, and Section VI provides an overview of the virtual channels. Section VII contains the experimental results including simulation and implementation results of the router. Finally, the conclusions are made in Section VIII.

II. SWITCHING METHODOLOGIES

Switching techniques help decide the way input and output blocks are connected inside the router switch. They also help in determining at what time what message components should be transferred along the paths. Different types of switching methods are circuit switching, packet switching, and wormhole switching.

A) Circuit switching-

For implementing circuit switching, the physical path between the source and destination is reserved and held till all the data has been transferred. The positive side of this method is that the bandwidth is reserved for the entire duration. But on the negative side, the valuable resources are also rendered useless for the entire duration of time when the data is being transmitted.

B) Packet switching-

For implementing packet switching, the data is divided into small packets and it transmits the packet whenever the source has a packet which needs to be sent. The need to store all the packets in a switch makes the buffer requirement high. The requirement for SoC is that the switches should not consume a major portion of the silicon area as compared to the IP blocks.

C) Wormhole switching-

For implementing wormhole switching, the packets are divided into small flits (fixed length flow control units). The buffers at the input and the output side need to store only few flits. Thus when compared to packet switching, the space required by the buffers are really small. Hence the switches can be designed in a small and compact way. The first flit is called as the header flit and it contains the routing information. Decoding the header flit helps in establishing the path between switches and the flits coming after that follow that path in a pipelined manner. So all the input flits can be forwarded to the same output channel as the preceding header flit, and there is no need to reorder the flits at the output. When a condition where a certain flit is facing a channel which is busy occurs, then the subsequent flits need to wait in their current locations as well. The only drawback of this switching method is that the messages which are transmitted cannot be multiplexed or interleaved. The channel cannot be used by another message till the entire previous message has reached the destination and used the channel fully. So the channel utilization decreases in situations where one packet is blocked in a buffer. That problem can be solved by using virtual channels in the input and output ports. If one flit is blocked, then alternate flit packets can use the buffers and thus the entire physical channel [1].

III. ROUTER DESIGN

The router consists of a number of input ports and output ports which are connected to share channels, a switching matrix which decides the way inputs and outputs can be connected. A logic block which helps in implementing the flow control is also part of the design. Flow control help in the optimization of the NoC resources (Channels, bandwidth) etc. NoCs usually have a distributed control and it lets each router makes the decisions locally. [5]

The routing algorithm based on the routing information extracted from the packet header helps in selecting the output port where the data from the input port will arrive.

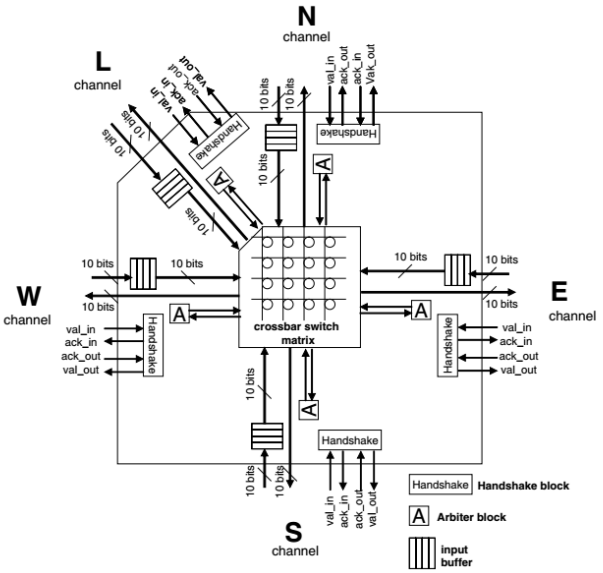

Figure.1 Typical Router Architecture [5]

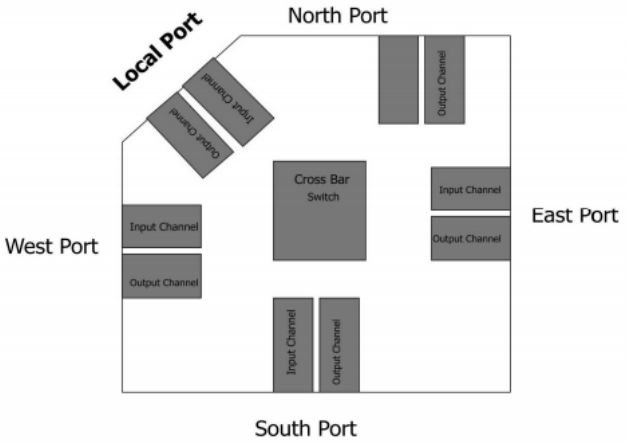

Figure.2 Router design implemented

The architecture implemented consists of five input ports which are named as core, north, south, east and west, and one crossbar switch. The data packets move from the input to the output channel. Each input and output channel have separate decoding logic which helps in improving the performance of the router. Buffers help in the temporary storage of the data. The buffering method used is store and forward. Control logic helps in determining the arbitration decisions. For establishing

the communication between the input and the output logic, a communication link has been established. The arbiter decides the control signals of the switch and sets it up based on the information deciphered as to where the data needs to go.

The input channel consists of four FIFOs, one port arbiter, and decoding logic. The decoding logic block sends the data to the appropriate four FIFOs used in the input channel based on the information extracted. The port arbiter is controlled by the main arbiter used in the router, and when it is enabled, it selects the appropriate output based on the logic applied to it. The depth of the FIFO is kept four for keeping the design simple. The read and write operations performed on the FIFOs are controlled by the port arbiter (FSM). If the FIFO is empty, it will not send the data to the port arbiter, if it is full, then the data is not written in the FIFOs anymore. The output channel has similar control and decoding logic as the input channel port. The arbiter helps in solving the problems of multiple requests coming at a time. Based on the fixed priority scheme, it issues grants to the different input ports accordingly.

IV. PERFORMANCE METRICS

It is desirable to get a high throughput, low energy efficiency, low area overhead, low latency etc.

A) throughput-

The performance can be measured in terms of bandwidth in bits/sec. the rate at which he message is transferred across the network i.e throughput is an important metric. For the message passing systems, the throughput can be defined as

TP = ((Total messages completed)*(length of message))/ ((number of IP blocks)*(total time))

Total messages completed means the total number of messages arrived successfully at the destination. The length of message is measured in flits, and the number of IP blocks is the total number of functional blocks involved in the communication. Total time refers to the time that elapses between the first messages sent and last message received.

B) Latency

Transport latency can be defined as the total time elapsed from the moment the first message header occurs at the source and the last tail flit received at the destination node. The path consists of switches and interconnects. Depending on different types of routing algorithms used, and on the source and destination pair, different messages have different latency. Even the overhead in the source and destination contributes to the latency. It can be defined as

L = sender overhead + transport latency + receiver overhead.

C) Energy

As the flits travel on the interconnection network, the switch wires, logic gates etc toggle and those results into energy dissipation. The dynamic energy dissipation which is caused by the communication process is of concern. Also the energy dissipation by the flits for each switch hop and interconnect needs to be determined. The energy for hop per flit can be defined as

$E_{hop} = E_{switch} + E_{interconnect}$.

Both the energies related to switch and interconnect depends on the total capacitances as well as the signal activity of the switch.

D) Area requirements

The silicon area requirements of the different interconnect schemes need to be evaluated for determining their feasibility. That will include the storage buffers at the input and the output side, the interconnect switch wires etc.

V. SWITCH ARCHITECTURE

Crossbar switch architecture is used as part of the design. It can be implemented in two ways, one by using tri-state buffers, the other by using multiplexers. For the design implemented in the paper, the multiplexers are used for the switching. Each input and output port is connected to the multiplexers and the select lines. The output address is generated by the arbiter. Between the two designs, it can be said that the design using the multiplexers is better that the one using crossbar switches.

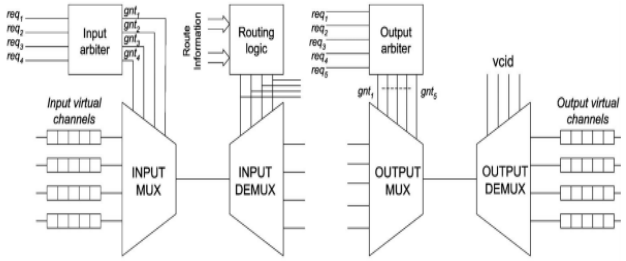

Figure.3 Typical switch architecture

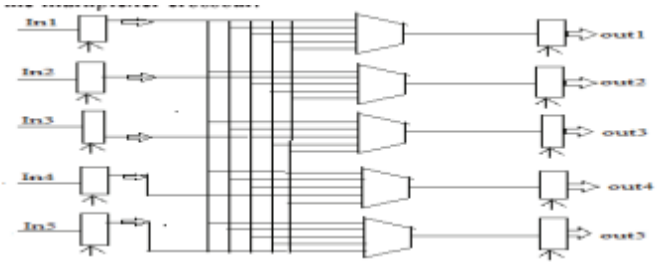

Figure.2 Crossbar switch using multiplexers

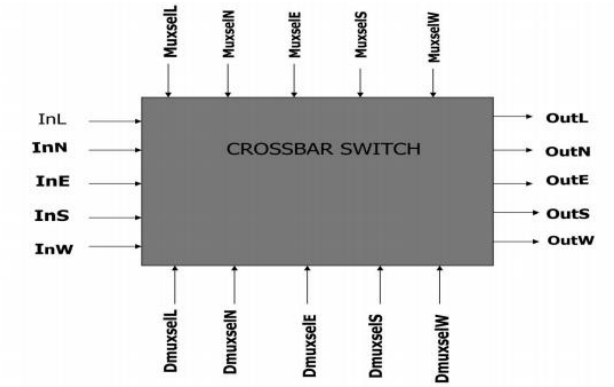

Figure.3 Simplified Block diagram of Crossbar Switch

VI. VIRTUAL CHANNELS

Virtual channels help in flow control by using the method of multiplexing one single physical channel over other separate channels with incorporating independent buffer queues. It helps in avoiding deadlocks, in optimizing the wire usage. When the resources are occupied fully and is waiting for one another to be released in order to proceed ahead with the communication, deadlock occurs. The situation where the communication is not yet completed, but the resources keep changing their status, livelock occurs. [5]

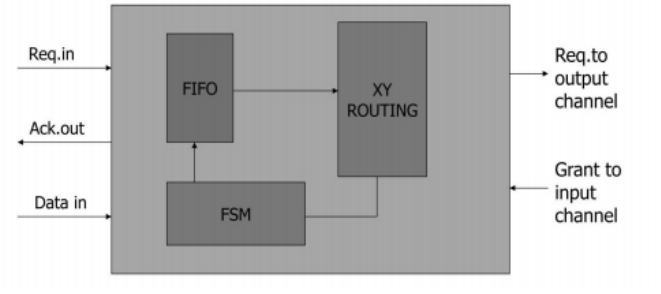

Figure.4 Input Channel

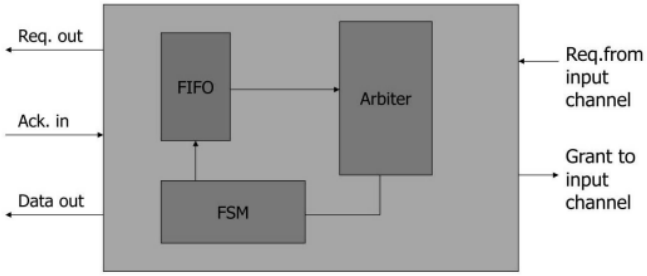

Figure.5 Output Channel

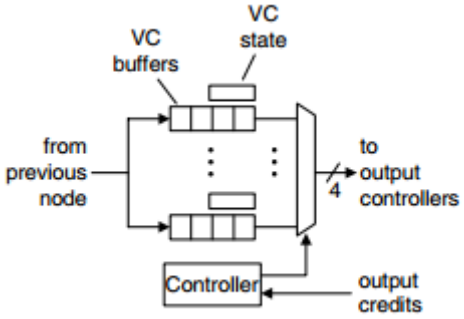

Figure.6 Virtual channels

VII. EXPERIMENTAL RESULTS

The design was captured in VHDL RTL and was synthesized using the Synopsys Educational Design Kit (PDK) for a generic 90nm process technology. This design kit is portable to design rules of real processes such as TSMC 90nm or IBM 90nm. Using this PDK, the area, power etc are calculated for the router design.

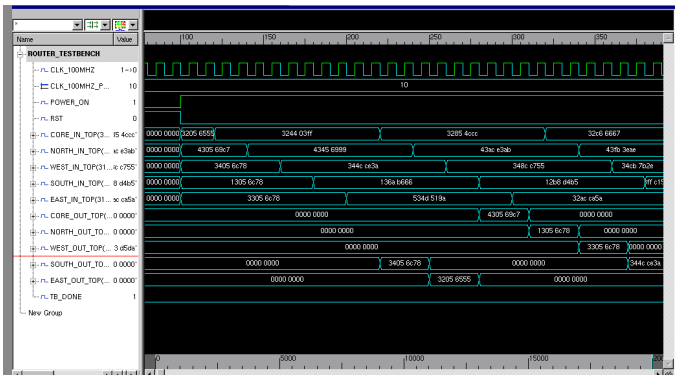

Figure.7 Router Simulation Results (A)

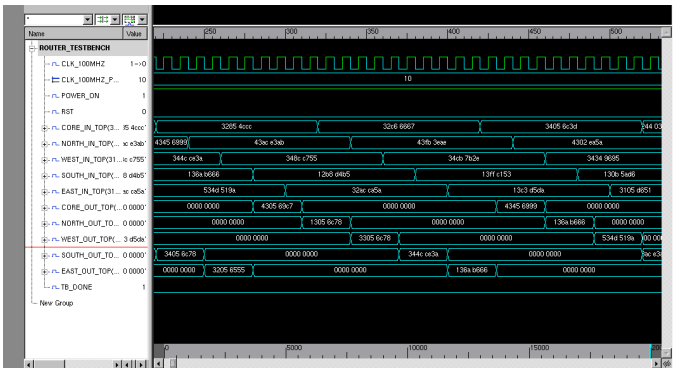

Figure.8 Router Simulation Results (B)

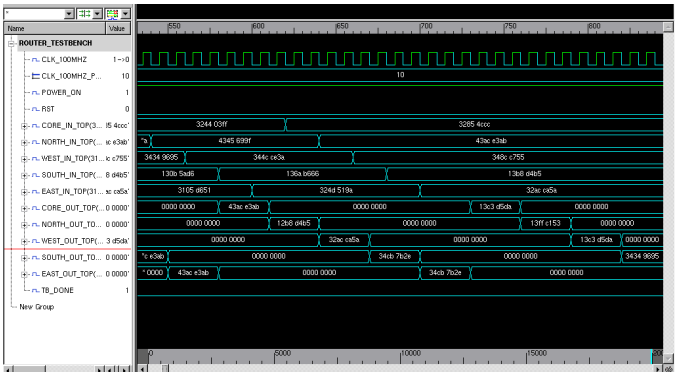

Figure.9 Router Simulation Results (C)

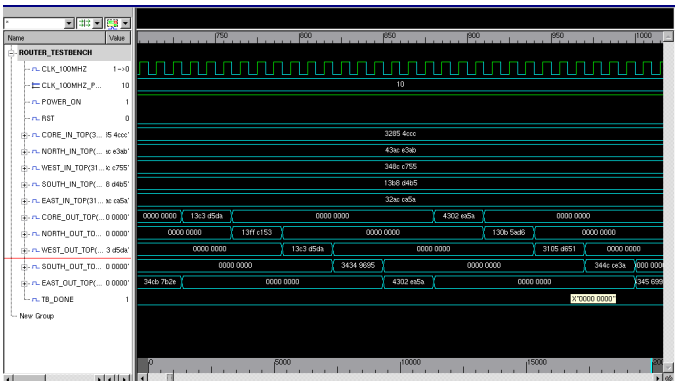

Figure.10 Router Simulation Results (D)

The synthesis has been performed using the Synopsys tool Design Vision for the TSMC 90nm technology library and the path was set. The reports have been generated for are and power. The whole design was run at 100MHz.

Area Distribution-

```
Number of ports:              323
Number of nets:               669
Number of cells:                8
Number of references:           8

Combinational area:       104717.721595
Noncombinational area:    116023.909981
Net Interconnect area:    18514.749149

Total cell area:          220741.631576
Total area:               239256.380725
```

Power Results-

```
   Cell Internal Power  = 501.7351 uW   (89%)
   Net Switching Power  =  58.9527 uW   (11%)
                          ---------
Total Dynamic Power     = 560.6877 uW  (100%)

Cell Leakage Power      = 736.6701 uW
```

VIII. Conclusion

NoCs architectures are emerging as interconnect architectures for different multiprocessor SoC platforms. Different architectures in terms of energy dissipation, powe, area overhead etc are observed.

In this paper, Router supporting five connections at the same time has been designed and tested successfully. The input and output port width used is 32-bit. Crossbar switch using the multiplexers was used. The Arbiter implemented used the fixed priority scheme.

Acknowledgment

The author would like to thank Dr. Amlan Ganguly for suggesting this as one of the topics for authoring the research paper and encouraging us with the work.

References

[1] P. P. Pande, C. Grecu, M. Jones, A. Ivanov, R. Saleh, "Performance evaluation and design trade-offs for network-on-chip interconnect architectures," Computers, IEEE Transactions on, Aug. 2005, vol.54, no.8, pp.1025-1040.

[2] W. J. Dally, B. Towles, "Route packets, not wires: on-chip interconnection networks," Design Automation Conference, 2001, pp. 684-689.

[3] S. Sahu, H. M. Kittur, "Area and power efficient network on chip router architecture," Information & Communication Technologies, AApril 2013, pp. 855-859.

[4] S. Malviya, A. Jaiswal, "Five Port Router for Network on Chip".

[5] Chapter 2 on NoC basics from 'Reliability, Availability, and Serviceability of Networks-on-Chip.

Vineeta Singh received the B.E. degree in Electronics and Telecommunication from Pune University, India in 2012. She is currently enrolled as a graduate student in department of Electrical and Microelectronic engineering at Rochester Institute of Technology, New York.

She is working as a research assistant and as a teaching assistant for Professor Kermani in the same institution.

Her Research interests include Reconfigurable Computing, Image Processing, Design and test Multi-Core Chips, Digital Design, Cryptography, Micro-fabrication, VLSI process modeling.