DESIGN AND ANALYSIS OF FIVE PORT ROUTER FOR NETWORK ON CHIP

Swapna S
Department of Electronics
and Communication Engineering,
National Institute of Technology,
Rourkela, Odisha
Email: s.swapna.pai@gmail.com

Ayas Kanta Swain
Department of Electronics
and Communication Engineering,
National Institute of Technology,
Rourkela, Odisha
Email: swain.ayas@gmail.com

Kamala Kanta Mahapatra
Department of Electronics
and Communication Engineering,
National Institute of Technology,
Rourkela, Odisha
Email: kkm@nitrkl.ac.in

***Abstract*—With the technological advancements a large number of devices can be integrated into a single chip. So the communication between these devices becomes vital. The network on chip (NoC) is a technology used for such communication. A router is the fundamental component of a NoC. This paper focuses on the implementation and the verification of a five port router. The building blocks of the router are buffering registers, demultiplexer, First In First Out registers, and schedulers. The scheduler uses the round robin algorithm. The proposed architecture of five port router is simulated in Xilinx ISE 10.1 software. The source code is written in VHDL.**

***Index Terms*—network on chip, router, round robin algorithm**

I. INTRODUCTION

Nanometer technologies allow integration of millions of transistors on a single chip. As the integration goes on increasing it exacerbates the design productivity gap and timing closure problems. A system on chip is an integrated circuit that integrates all components of a computer or other electronics system into a single chip. The major challenge the designers of these systems must overcome is to provide functionally correct and reliable operations of the interacting components. On-chip physical interconnections will present a limiting factor for performance and energy consumption[1].

Initially the chip designing was aimed at producing chips that contained a single stand alone design. As the number of transistors that is fabricated onto a single chip increased, the concept of system on chip was made possible. In a system on chip(SoC), multiple stand alone designs, referred to as cores or Intellectual Property cores, are stitched together on a chip to provide a functional system. NoC is an approach to design the communication subsystem between intellectual property cores in a system on chip. The communication strategy in system on chip uses dedicated buses between communicating resources. This will not give any flexibility since the needs of the communication, in each case, have to be thought of every time a design is made. Another possibility is the use of common buses, which have the problem that it does not scale very well, as the number of resources grow. NoC is intended to solve the shortcomings of these, by implementing a communication network of switches/microrouters and resources [2,3].

The NoC design paradigm has been proposed as the future of ASIC design[4]. The major driving force behind the transition to NoC based solutions is the inadequacy of current day VLSI inter-chip communication design methodology for the deep sub-micron chip manufacturing technology[5]. The negative effect of technology scaling on global interconnects, increased dependence on fault-tolerant mechanisms as feature size reduces, increasing use of parallel architectures are the reasons why NoC is becoming popular.

The NoC based system on chips impose various design issues on the fabrication of such integrated chips. Firstly, the suitable topology for the target NoCs such that the performance requirements and design constraints are satisfied. Secondly, the design of network interfaces to access the on chip network and routers to provide the physical interconnection mechanisms to transport data between processing cores. Thirdly, the selection of communication protocols which are suitable for on chip interconnection networks. Finally, as technology scales and switching speed increases, future network on chips will become more sensitive and prone to errors and faults. Fault tolerance is becoming critical for on-chip communications [6].

Todays SoCs need a network on chip IP interconnect fabric to reduce wire routing congestion, to ease timing closure, for higher operating frequencies and to change IP easily. Network on chips are a critical technology that will enable the success of future system on chips for embedded applications. This technology of network on chips are expected to dominate computing platforms in the near future.

The remainder of this paper begins by discussing a background on NoC under section II. Section III contains a description on network topology. In section IV we describe about the NoC router. Section V contains our proposed five port router architecture. Section VI contains the simulation and results. We conclude this paper in section VII and we end this paper in section VIII by highlighting the future work for the development of our NoC.

II. NETWORK ON CHIP - A BACKGROUND

The scaling of chip technologies has enabled large scale SoCs, where all components of an electronic design are inte-

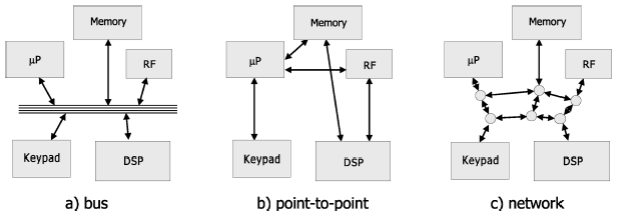

Fig. 1. Communication structures in SoC a)traditional bus based communication, b)dedicated point to point links c)network on a chip [7]

grated on a single chip. In SoCs the communication systems used are conventional bus systems and point-to-point links. Since these two alternatives are not suitable for highly complex systems, new approaches for intrachip communication are adopted to achieve high performance in SoCs. NoC is one such approach where communication between processors, memories, IPs, and IOs is achieved by exchanging data packets using a network. The NoC approach borrows concepts and techniques from the well-established domain of computer networking.

Fig. 1 shows some basic communication structures in SoC designing. The solutions for SoC communication structures have generally been characterized by custom designed ad hoc mixes of buses and point to point links. The bus builds on well understood concepts and is easy to model. In a highly interconnected multicore system it can quickly become a communication bottle neck as more units are added to it.

There are various advantages of networks over buses. In bus communication systems every unit attached adds parasitic capacitance, therefore electrical performance degrades as system grows. Bus timing is difficult in a deep submicron process. Bus arbitration can become a bottleneck. The arbitration delay grows with the number of masters. The bus arbiter is instance-specific. Bus testability is problematic and slow. The bandwidth for communication is limited and shared by all units attached. In network based communication only point-to-point one-way wires are used, for all network sizes, thus local performance is not degraded when scaling. Network wires can be pipelined because links are point-to-point. Routing decisions are distributed, if the network protocol is made non-central. The same router may be reinstantiated, for all network sizes. Locally placed dedicated built in self test mechanism is fast and offers good test coverage. The aggregated bandwidth scales with the network size[7].

The NoC paradigm is highly suited to provide SoC platforms scalable and adaptable over several technology generations. NoC platforms may allow the design productivity to grow as fast as technology capabilities and may eventually close the design productivity gap[8].

III. Network Topology

Fig. 2 shows a sample NoC as a 4x4 mesh topology which provides global chip level communication. As shown in the figure the NoC contains some fundamental components like network adapters, routing nodes and links. Network adapters

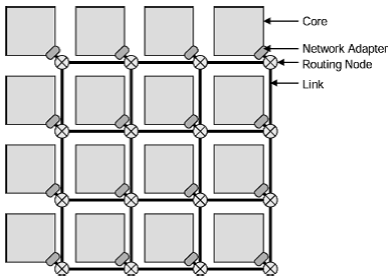

Fig. 2. A 4x4 mesh network

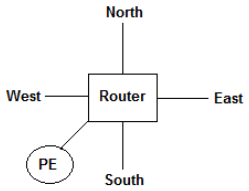

Fig. 3. A central NoC Router

implement the interface by which cores (IP blocks) connect to the NoC. Routing nodes route the data according to chosen protocols. Links connect the nodes, providing the raw bandwidth.

The network topology defines the interconnection of nodes in a network (processing elements) using links (channels). It refers to the shape of the network and how they are connected to each other and communicate with each other. There are various topologies available like mesh, torus, tree, butterfly, polygon, and star topology[9]. But mesh topology is favored by many research groups because of its layout efficiency, good electrical properties and simplicity in addressing on-chip resources[10].

A mesh-shaped network consists of m columns and n rows. The routers are situated in the intersections of the two wires and the computational resources are near routers. Addresses of routers and resources can be easily defined as x-y coordinates in mesh. The regular mesh network is also called as Manhattan Street network.

IV. NOC Router

The router is the most important component in a network on chip[11]. It is the communication backbone of a NoC system. So it should be designed with maximum efficiency. Routers are used on a network for directing the traffic from the source to the destination. It coordinates the data flow which is very crucial in communication networks. Routers are intelligent devices that receive incoming data packets, inspect their destination and figure out the best path for the data to move from source to destination. A router is built according to the OSI model of network on chip. Each layer performs its own specific functions.

In Fig. 3, an NoC central router in mesh topology is shown. The central router has 5 in/out port. The local port in utilized

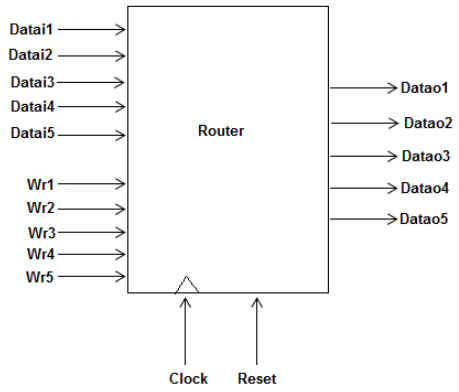

Fig. 4. The router ports

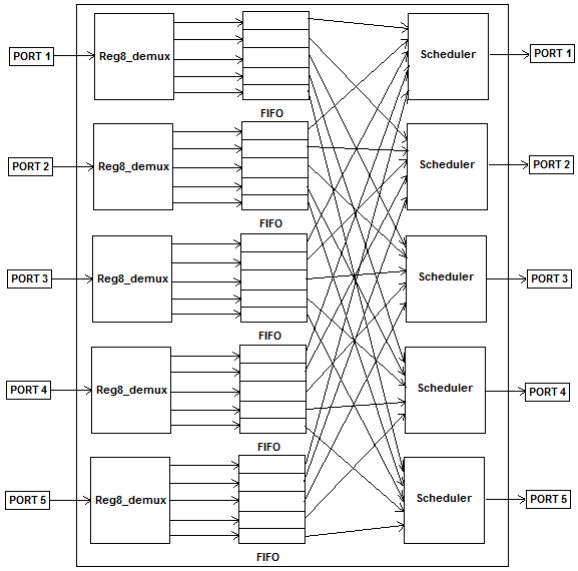

Fig. 5. The router architecture

to connect the correspondent circle to the processing element (PE) and other ports are for connecting to other routers[12].

V. Router Architecture

Fig. 4 shows the entity of the designed router. It consists of five data input ports (datai1, datai2, datai3, datai4, datai5), five data output ports (datao1, datao2, datao3, datao4, datao5), five packet available indicators (Wr1, Wr2, Wr3, Wr4, Wr5), a clock(Clock) and a reset(Reset) signal.

Fig. 5 shows the internal architecture of the designed router. The building blocks of router consist of mainly three parts

1. Registers and demultiplexers
2. First In First Out Registers
3. Schedulers.

A. Register and demultiplexer

Fig. 6 shows the combined register and demux design. The register is used for storing the input data in the form of a buffer. The demultiplexer will direct the input to the appropriate output port.

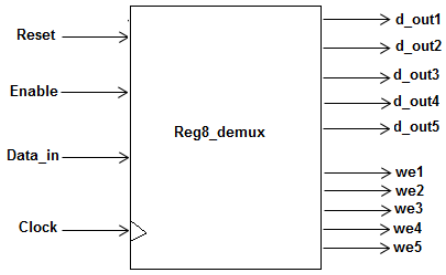

Fig. 6. Combined register and demux design

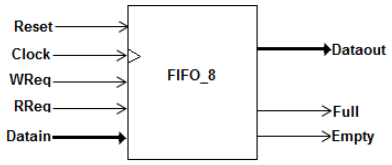

Fig. 7. FIFO Unit

1) 8-bit register: The register has a positive edge clock, an active high clock enable and an active high asynchronous reset. The output of the register is the input of the demultiplexer. The data input to the register is transferred to the output port at the positive edge of the clock if and only if the enable is 1 and the reset is 0. If the reset is 1, then the output port of the register is set to zeros. If the enable is 0, then the output port keeps its current value.

2) 1-to-8 8-bit demultiplexer: The demultiplexer directs the input to the proper output port according to the select signal. The select signal is taken as the first three bits of the input data. The demultiplexer also has an enable. If this enable is set to 1, then the input data is transferred to the appropriate output port and the corresponding write enable is set to 1, while the other output ports and write enables are set to zeros. If the enable signal is 0, then the output ports and the write enables are all set to zeros.

B. FIFO Unit

The FIFO unit, as shown in Fig. 7 consist of two parts RAM memory and FIFO Control. The FIFO receives read and write requests from RReq and WReq signals respectively. When the FIFO is full, write operations are disabled and when it is empty, read operations are disabled. The FIFO empty flag is set to high when the FIFO is empty and full flag is set to high when the FIFO is full.

1) RAM: The write and read operations in the memory are synchronized with the memory clock input (Clk). If (Clk) is rising and (Wr_Enable) is 1, then the input data word (D_in) is written to the memory location with address Addr. If (Clk) is rising and (Rd_Enable) is 1, then the output data word (D_out) is read from the memory location with the address (Addr). The input bus (Datain) of the FIFO is the input bus of the RAM (D_in), while the output bus (Dataout) of the FIFO is the output bus of the RAM (D_out). The

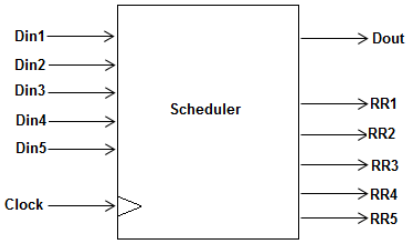

Fig. 8. Scheduler

width of all the data ports is eight bits. When the memory is reset asynchronously by the reset signal (Rst), all its locations become zeros and its output (D_out) is set to zeros as well.

2) FIFO Controller: The FIFO controller receives read and write requests from the (R_Req) and (W_Req) signals respectively. It checks the validity of the read or write operations and generates valid signals on ($Read\_En$) or ($Write\_En$) ports. Then it outputs the corresponding read or write address on (Add_Output). The ($Read\_En$) and ($Write\_En$) ports are connected to the (Rd_Enable) and (Wr_Enable) ports of the RAM, respectively. The (Add_Output) port is connected to the (Addr) port of the RAM.

C. Scheduler

The scheduler used here (Fig. 8) is a round robin scheduler which uses the round robin algorithm. It assumes that all data are equally valid for selection. The algorithm lets every active data flow that has data packets in the queue to take turns in transferring packets on a shared channel in a periodically repeated order. The port through which the data comes at the present instance should have the lowest priority at the next round of scheduling. The outputs RR1, RR2, RR3, RR4 and RR5 are used to indicate the next port to be read and they are further connected to the read request ports in the corresponding FIFO.

VI. Simulation Results

Simulation refers to the verification of a design, its function and performance. It is the process of applying stimuli to a model over time and producing corresponding responses from a model. The simulation is performed in XILINX ISE 10.1 [13] software. A test bench is also written to test the routing pattern from various data packets.

Fig. 9 shows the simulation result of five port router. The reset (rst) signal must be kept low during normal operations. The write enable (wr) signals, when high, will enable the corresponding demultiplexers. The input signals (datai) are the data packets given to the router. The routing pattern is observed from the output signals (datao). The output signals follow the round robin scheduling algorithm. The data at the input ports is directed to the output port depending on the first three bits of the input data which act as the select lines of the demultiplexer. The input data F1 from port 1 is directed towards port 1 of the output since the first three bits of the data is 1. At that instance the round robin scheduling signal of r2 will be high which indicates the port to be read in the next instance. Like wise the other data packets are also routed.

From the simulation it is also observed that, depending on the amount of input data packets the router will have a delay before the output data becomes error free.

Table I shows the synthesis results of the NoC router in FPGA.From Table II the total statistical information of our router synthesis on the FPGA model is obtained. The resource utilization of the router in the FPGA is calculated from Table III. The operating frequency of this router in ASIC is 144.196 MHz. The minimum input arrival time before clock and maximum output time required time after clock is estimated as 7.599 ns and 4.283 ns respectively.

TABLE I
LOGIC CIRCUIT STATISTICAL INFORMATION OF SYNTHESIS ON FPGA-XILINX

Cell name	Library name	Number of gates
BUF	xcv2p	3
GND	xcv2p	1
INV	xcv2p	55
LUT2	xcv2p	317
LUT3	xcv2p	722
$LUT3\_L$	xcv2p	75
LUT4	xcv2p	657
$LUT4\_D$	xcv2p	50
$LUT4\_L$	xcv2p	25
MUXF5	xcv2p	352
MUXF6	xcv2p	160
VCC	xcv2p	1
FD	xcv2p	10
FDC	xcv2p	100
FDCE	xcv2p	1700
FDPE	xcv2p	25
FDR	xcv2p	25
FDS	xcv2p	32
FDSE	xcv2p	5
BUFGP	xcv2p	1
IBUF	xcv2p	46
OBUF	xcv2p	40

TABLE II
TOTAL STATISTICAL INFORMATION OF SYNTHESIS ON FPGA-XILINX

Component	Estimated value
Port	87
Instance	4402
Number of Dffs or Latches	1897
Number of Function Generators	1846
Number of MUX	512
Number of gates	1904
Clock	144.196 MHz

TABLE III
PERCENTAGE OF RESOURCE UTILIZATION OF FPGA-XILINX(DEVICE SELECTED - 3S500EFG320-4)

Resource	Used	Available	Utilization percentage
Slices	1464	4656	31
Slice Flip Flops	1897	9312	20
4 input LUTs	1901	9312	20
Bonded IOB	87	232	37
GCLK	1	24	4

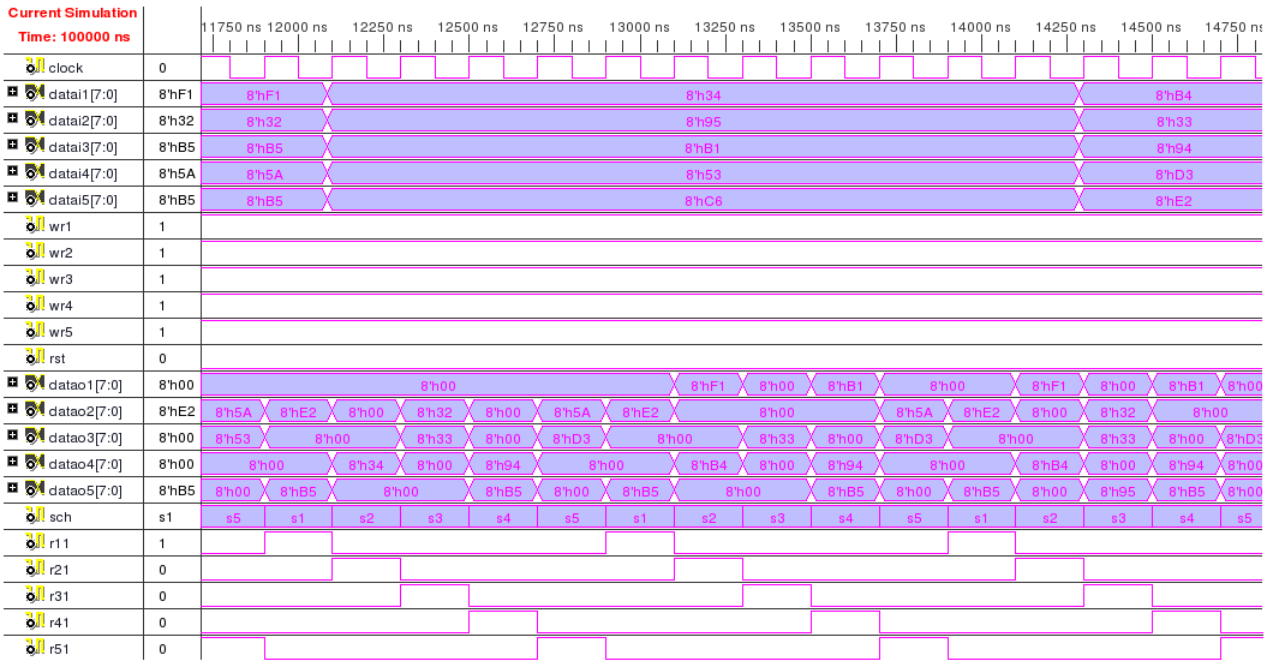

Fig. 9. Simulation result of five port router

VII. Conclusions and Future Work

A five port router using simple decoding logic is proposed in this paper. The synthesis and simulation of the proposed router is verified through VHDL codes using XILINX ISE 10.1 software. The simulation facilitates clear understanding of routing pattern and the functionality of a five port router for a network on chip. In future, we intend to work on improving the scheduling algorithm used in the router for better performance.

Acknowledgment

The authors would like to thank the Ministry of Communication and Information Technology (MCIT), Government of India for providing us with the CAD tools, boards and other facilities required in our research work.

References

[1] Luca Benini and Giovanni De Micheli, 'Networks on chips: a new SoC paradigm," *Computer*, vol. 35, no. 1, pp. 70-78, 2002

[2] Rickard Holsmark and Magnus Hgberg, 'Modelling and Prototyping of a Network on Chip," Master of Science Thesis, 2002 Electronics

[3] Muhammad Ali, Michael Welzl, Martin Zwicknagl, 'Networks on Chips: Scalable Interconnects for Future Systems on Chips," *4th European Conference on Circuits and Systems for Communications*, pp. 240-245, 2008

[4] M. Pirretti, G. M. Link, R. R. Brooks, N. Vijaykrishnan, M. Kandemir, M. J. Irwin, 'Fault tolerant algorithms for network-on-chip interconnect," *Proceedings. IEEE Computer society Annual Symposium on VLSI*, pp. 46-51, Feb 2004

[5] Israel Cidon and Idit Keidar, 'Zooming in on network-on-chip architectures," *16th international conference on Structural Information and Communication Complexity*, 2009

[6] Xinan Zhou, 'Performance evaluation of network-on-chip interconnect architectures," Master of Science Thesis, 2009 Electrical Engineering

[7] Tobias Bjerregaard and Shankar Mahadevan, 'A Survey of Research and Practices of Network-on-Chip," *ACM Computing Surveys*, vol. 38, no. 1, pp. 1-51, 2006

[8] Axel Jantsch and Hannu Tenhunen(Eds.), *Networks on chip*, Kluwer Academic Publishers, 2003

[9] Ville Rantala, Teijo Lehtonen, Juha Plosila, 'Network on Chip Routing Algorithms," *Journal of Systems Architecture*, TUCS Technical Report No 779, pp. 5-7, August 2006

[10] Rickard Holsmark, Maurizio Palesi, Shashi Kumar, 'Deadlock free routing algorithms for irregular mesh topology NoC systems with rectangular regions," *Journal of Systems Architecture*, vol. 54, no. 3-4, pp. 427-440, 2008

[11] A.S. Kale and M.A.Gaikwad, 'Design and Analysis of On-Chip Router for Network On Chip," *International Journal of Computer Trends and Technology*, vol. 9, no. 6, pp. 182-186, 2011

[12] Seyyed Amir Asghari, Hossein Pedram, Mohammad Khademi, and Pooria Yaghini, "Designing and Implementation of a Network on Chip Router Based on Handshaking Communication Mechanism," *World Applied Sciences Journal*, vol.6, no. 1, pp.88-93, 2009

[13] Xilinx ISE 10.1 User Manual, www.xilinx.com