ISSN 2319-2720
Volume 2, No.4, October - December 2013
International Journal of Computing, Communications and Networking
Available Online at http://warse.org/pdfs/2013/ijccn09242013.pdf

Low-Latency and Low- Area Overhead Based Performance Modeling of Network on Chip Architecture for FPGA Based Computing System

Sudhir N. Shelke[1], Pramod B. Patil[2]
[1]Head of Electronics & Telecommunication Engineering. Department, J D College Of Engineering & Management, Nagpur, India
[2]Principal, J D College of Engineering & Management, Nagpur, India.
[1]sudhirshelke1976@gmail.com
[2]ppamt07@yahoo.com

ABSTRACT

This paper contributes in the enhancement of NoC architecture with respect to area & Latency. A switch/Router size 4*4 is targeted. The input/output module with efficient buffer to store the data packet while waiting for the scheduling is optimized at architecture level with respect to area & latency and observed that the design is operating at 305.573 MHz, with latency 2 per clock cycle and result are compare with other publications.

Key Words: FPGA, NoC, PNoC, SNoC, NoCem, SoCBUS.

1. INTRODUCTION

Ever-increasing requirements on electronic systems are one of the key factors for evolution of the integrated circuit technology. Multiprocessing is the solution to meet the requirements of upcoming applications. Multiprocessing over heterogeneous functional units require efficient on chip communication [1, 2]. Network-on-Chip (NoC) is a general purpose on-chip communication concept that offers high throughput, which is the basic requirement to deal with complexity of modern systems, as shown in Figure 1. All links in NoC can be simultaneously used for data transmission, which provides a high level of parallelism and makes it attractive to replace the typical communication architectures like shared buses or point-to-point dedicated wires.

Apart from throughput, NoC platform is scalable and has the potential to keep up with the pace of technology advances [3]. But all these enhancements come at the expense of area and power. In the RAW multiprocessor system, interconnection network consumes 36% of the total chip power [4].

A typical NoC system consists of processing elements (PEs), network interfaces (NIs), routers and channels. The router further contains scheduler, switch and buffers. Buffers consume the 64% of the total node (router + link) leakage power for all process technologies, which makes it the largest power consumer in any NoC system [5]. Moreover, buffers are dominant for dynamic energy consumption [6].

2. NoC Architecture

Network-on-Chip has been proposed on various topologies [7-10]. A simple NoC architecture consists of three components: the routing nodes, the links, and network interfaces (or network adapters in some literature), as shown in Figure 2.

Routers direct data over several links (hops). Topology defines their logical lay-out (connections) whereas floor plan defines the physical layout. The function of a network interface (adapter) is to decouple computation (the resources) from communication (the network). Routing decides the path taken from source to the destination whereas switching and flow control policies define the timing of transfers. Task scheduling refers to the order in which the application tasks are executed and task mapping defines which processing element (PE) executes certain task. IP mapping, on the other hand, defines how PEs and other resources are connected to the NoC [11].

The major goal of communication-centric design and NoC paradigm is to achieve greater design productivity and performance by handling the increasing parallelism, manufacturing complexity, wiring problems, and reliability. The three critical challenges for NoC are: power, area, latency, and CAD compatibility [12].The key research areas in network-on-chip design [13, 14].are as:

- Communication infrastructure: topology and link optimization, buffer sizing, floorplanning, clock domains, power.
- Communication paradigm: routing, switching, flow control, quality-of-service, network interfaces
- Application mapping: task mapping/scheduling and IP component mapping.

All of these challenges result in area, power, and performance tradeoffs [13].Area and power can be estimated from hardware requirements. Performance is generally estimated using analytical model.

This paper proposes Low latency; Low-Area Overhead & High throughput NoC for FPGA based computing system.

3. PROBLEM STATEMENT

The implementation of network-on-chip presents certain challenges. Two of the most critical design metrics for networks-on-chip are area requirements and power consumption. Due to the fact that die area per wafer of silicon is limited, the NoC implementation should be carried out using an approach that minimizes area requirement. Also due to likelihood of most SoCs being implemented in battery powered devices, power consumption of the NoC should also be as low as possible. Usually, reduction in area results in a saving in power requirements due to the fact a smaller area is achieved using fewer components on-chip. Fewer components on-chip will consume less power compared to architecture requiring more components on-chip [15].

Many authors tried to fulfill the requirement of NoC design using ASIC implementation but we foresee on implementation of FPGA base NoC. The main goal of the NoC architecture described here is high throughput, low latency, and low overhead. Also made possible to easily interface a standard bus protocol called wishbone to it.

4. DESIGN AND IMPLEMENTATION OF PRESENTED NOC ARCHITECTURE

Given an existing implementation of a routing node for on-chip networks, it is the goal of this work to present a modified implementation of the routing node to minimize the area requirements and as a result lower the power requirement.

The routing node consists of four basic components: the input ports, the output ports, the crossbar switch, and the scheduler. The components arranged in decreasing order of size are the input blocks, the scheduler, the output blocks, and the crossbar switch as shown in Figure 1.

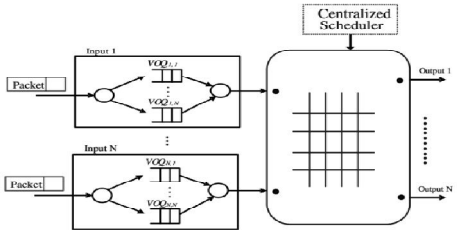

Figure 1: Router Components

The primary function of the input block is to store incoming packets before they can be routed to their respective output ports.

The function of the scheduler/Arbiter is to arbitrate between conflicting requests for access to the crossbar switch shared medium. The scheduler architecture is based on a symmetric implementation of round-robin like algorithm requiring one set of grant arbiters and one set of accept arbiters to perform arbitration.

Area and power are two important parameters which need to be optimized for better NoC performance. The NoC consists of three basic components which are the routing node, the routing links, and network interfaces. Optimization of the routing nodes will lead to improvement in the area and the power requirements of the NoC, as it is the most redundant component which lies in association with every processing element.

The main goals of the NoC architecture described in this paper is high throughput, low latency (especially for small messages), and low area overhead. Another goal is to make it possible to easily interface a standard bus protocol such as Wishbone to it. A third goal is that there should be a certain amount of flexibility in regards to the choice of topology.

The authors' experience from SoCBUS [16] also indicates that the large latency involved in transmitting small messages can be a huge problem in a real system. Since it is critical to be able to handle small messages in a system where a standard bus is connected to a NoC, the architecture presented in this paper is based upon packet switching. Wormhole routing is used to avoid the need for large packet buffers and to reduce the latency.

Table 1: The Signaling Used On The Noc

Signal	I/O	Width	Description
Strobe	I	1	Valid data is present
Data	I	36	Used as data signals
Last	I	1	Last data in a transaction
Dest	I	5	Address of destination node
Route	I	3-4	Destination port on the switch (one hot coded)
Ready	O	1	Signals that the remote node is ready to receive data

4.1 Input Part

An incoming packet is first buffered in an input FIFO. As long as an output port is available, the input FIFO will be emptied as fast as it can be filled. However, if no output port would be available, the input FIFO will quickly fill up. To avoid overruns, the input module will signal the sender that no further data should be sent as soon as only a few entries are left. This is required because the pipeline latency will

cause additional entries to be written before the sender can react.

The most complex part of this switch is the input part, which is shown in Figure.2. The FIFO is based on SRL16 a primitive that allows a very compact 16 entry FIFO is constructed. As the SRL16 has relatively slow outputs, a register is placed immediately after the SRL16. This means that the input part has a latency of two cycles in case the FIFO is empty and the output port is available. The block named "check empty" makes sure that no spurious ROUTE_* signals are sent to the arbiter if the FIFO is empty. By doing this, the arbiter is simplified as compared to having both the ROUTE_* signals and separate signals for WEST_EMPTY, NORTH_EMPTY, etc. In particular, it is easier to identify the case where only one input port needs to send a packet to the output port and send the packet immediately without any arbitration delay. Finally, the READY signal is adjusted for pipeline latency so that the FIFO will not overflow if the sender does not stop sending as soon as READY goes low.

The other critical path of the input signal is the read enable signal of the input FIFO. In order to keep the latency down, the read enable signal is generated by looking at the destination port of all other input ports. If no other input port is trying to communicate with the selected output port and the output port is ready to send, the packet will be sent immediately.

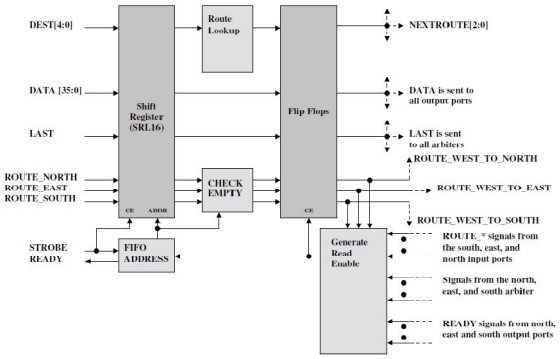

Figure 2: A detailed view of an input port of the packet switched NoC switch.

4.2 Output Part

Once the first part of a packet is available in the input FIFO, the arbiter of the selected output port will be notified. If the port is already busy or if several input ports are trying to send at once, the arbiter uses round robin arbitration to choose the next packet to be sent once the current sender is finished. The arbitration is therefore distributed between the input port where the read enable signal has to be generated without waiting a clock cycle on the arbiter. If the output port is available and no other input port is trying to send to this port, the arbitrator will allocate the output port for the duration of the incoming packet.

Figure.3. shows a detailed view of an output port of a 4-port switch. Each output port can only select from one of three input ports since there should be no need to route a packet back to where it came from in most topologies. If more than one input port needs to send a packet to the same output port, an arbiter in the output port uses round robin to select the port that may send. In the four port NoC switch, the output port is essentially a 3-to-1 mux controlled by the arbiter (or a 4-to-1 mux in the case of a five port switch). The DAT_* signals are formed by combining the destination address with the NEXTROUTE_* signals and the payload signal. It should also be noted that the part inside the dotted rectangle can be implemented inside one LUT for each wire in the DAT_* signals which will reduce the delay for this part somewhat. The latency of the switch when the input FIFO is empty and the output port is available is 3 clock cycles. If more than one input port has data for a certain output port, the latency is increased to 4 clock cycles due to an arbitration delay of one clock cycle for the packet that wins the arbitration. There are two main critical paths in this switch. One path is caused by the read enable signal that is sent to the input FIFO. The other is from the FIFO to the route look-up due to the slow output of the SRL16 elements.

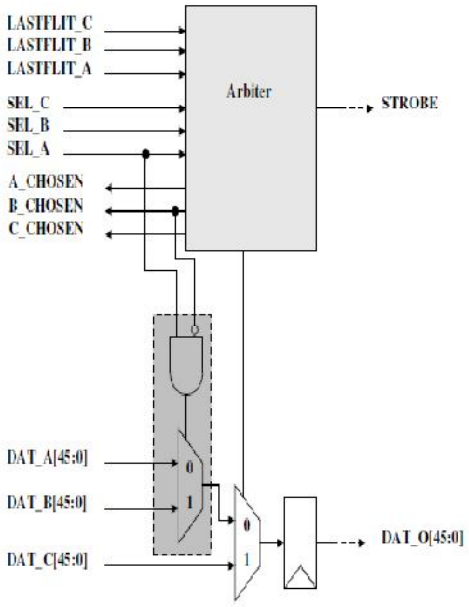

Figure 3: A detailed view of an output port of a 4-port switch

5. DESIGN OF ROUND ROBIN ARBITER

Table 2: The List of Inputs And Output Of Arbiter Is As Shown below

Signals	I/O	Descriptions
Asel, Bsel, Csel, Dsel	I	From output of fifo of other directional sub-node of 'noc' than the current sub-node.
oktosend	I	From the registered input pin of the node corresponding to the current direction.
Alast, Blast, Clast, Dlast	I	Used to define priorities of nodes for different direct as desired by the user.
Override	O	To indicate transmission of data through any of the directional node.
Aoverride Boverride Coverride Doverride	O	To transmit data through the respective direction of the sub-node.
Clk, rst	O	Indicates clock pulse and reset signal respectively.

5.1 Designing Rule of Arbiter

- Temporary intermediate signals are Adone, Bdone, Cdone, Ddone, only A, Only B, Only C and only D.
- As name suggests, "only A" is high i.e at logic 1, when only "A sel" i/p is high, "only B" is high when only "Bsel" i/p is high and so on.

  Only= 1
  Asel= 1
  Only= 1
  Bsel= 1
- Now, the priorities of the inputs are defined using the following logic and only one input signal is granted permission to transmit its data; or in other words, only one of the "overrides" output signal is activated or set high.
- When "A sel" is at logic '1' and "override" is '0' means if A needs to be transmitted and use of the signal is being presently transmitted and "only A" is '1' or "Adone" is '0' then "Aoverride" is set to '1'.
- Similarly, "Boverride" is set '1' when "Bsel" is '1' and "override" is '0' and "Bdone" is '0' or "only B" is '1'. Additionally to give highest priority to "Asel" the two related signals ("Asel"= '0' or "Adone" = '1') are also checked before asserting the signal "Boverride".
- Similarly for asserting "Coverride" and setting its priority level after "Asel" and "Bsel" following conditions are checked: Csel= '1', "Cdone"= '0' or "only"= '1', Asel= '0' or Adone = '1' and Bsel= '0' or Bdone= '1' and override should be at logic '0'.
- And finally for "Dovrride" conditions similar as above are checked for A, B and C Hence D has the least priority amongst all the four i/ps.
- Now, for terminating the transmission following condition need to checked when "Aoverride" is '1' i.e. A is being transmitted or only A is present and "oktosend" is asserted and "Alast"= '1' then, Adone is asserted and "Aoverride" is set to logic '0' i.e. it is terminated.
- For B to be terminated, "only B" should be '1' or "Boverride"= '1' and "Blast" and "oktosend" should be asserted Hence, "Bdone" is asserted and "Boverride"= '0'.
- Similarly, C and D are checked for the corresponding terminating condition at every clock pulse.
- Also, to check availability for transmission of one of the signal rest of the three signals should be low and the respective "DONE" signals should be asserted.
- For example, for "Adone" to be set to logic '0', "Bdone", "Cdone" and "Ddone" should be at logic '1' and "Bsel", "Csel" and "Dsel" should all be '0'.
- Something checking is also done to ensure that only one signal is being granted permission to transmit data at a particular instance of time.

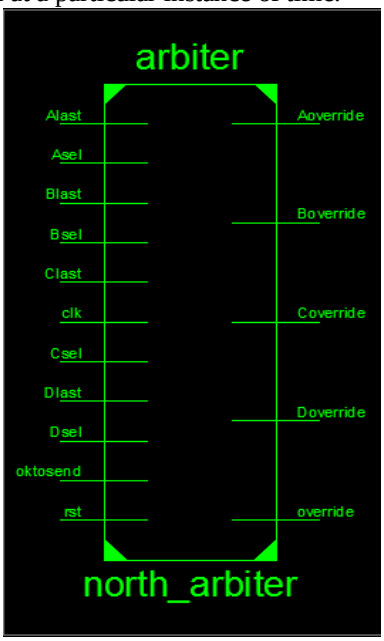

Figure 4: RTL View of Arbiter.

6. DESIGN OF SERIAL FIFO (SRLFIFO)

Table 3: The List of Input and Output Pins of SRLFIFO is shown below

Signals	I/O	Descriptions
clk	I	clock
rst	I	reset.
control	I	control input to srl block 1(width: 12bits)
d	I	data input to srl block 2 (width : 36 bits)
we	I	write enable.
rd	I	read enable.
sendok	O	Transmit data
avail	O	Fifo available or not full
q	O	Data to be transmitted; output from the srl block 2 (data) (width: 36 bits)
control q	O	'control' arl block 1 output (width: 12 bits)
A sel	O	Select A
B sel	O	Select B
C sel	O	Select C
D sel	O	Select D
to A	O	Transmit data to A
to B	O	Transmit data to B
to C	O	Transmit data to C

6.1 Designing Rule of SRLFIFO

- For the output signal 'avail' the following condition is checked:
  Avail{= '0'; 'addrptr' = "1111".
  = '1' ; 'addrptr' = "1111"}.
- For the signal 'sendok' at every clock's positive edge 'addrptr' is checked and if it is "1111" or less then "1001" (i.e.g) then "sendok" is asserted to '1'.
- Also, when 'we' (i.e. write enable) signal is '1' and if any or both of the signals 'rd' and 'avail' are at logic '0' them the 'new_addrptr' signal; which is internal to the fifo module; is incremented by '1'. i.e.
  'new_addrptr'= 'addrptr' + 1.
- When 'we' signal is '0' and both 'rd' and 'avail' is at logic '1' then 'new_addrptr' is decremented by '1'.i.e.
  'new_addrptr'= 'addrptr' -1.
- The signals 'Asel', 'Bsel', Csel' and 'Dsel' are asserted to logic '1', when both signal 'control q' (4^{th} bit) and 'avail' are '1'. Here, 5th bit of 'control q' signal corresponds to 'Asel', 6th bit of 'control q' signal corresponds to 'Bsel', and so on.
- Finally the addrptr is checked to opt for the desired destination sub-node for data and accordingly signals 'to A', 'to B' and 'to C' are asserted.
- Error check is also done for this module, if 'avail' = '1' and all the signals 'Asel', 'Bsel', 'Csel' and 'Dsel' are at logic '0' and none of them is asserted to '1', then execution is terminated indicating "no select signal active".

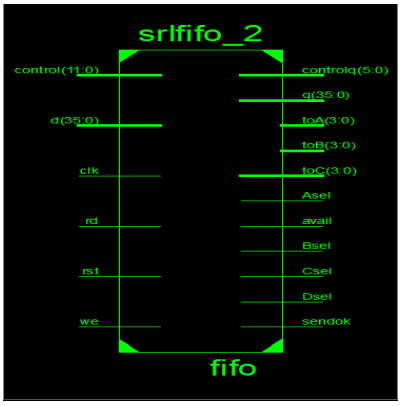

Figure 5: RTL view of SRLFIFO

7. Testing Scheme

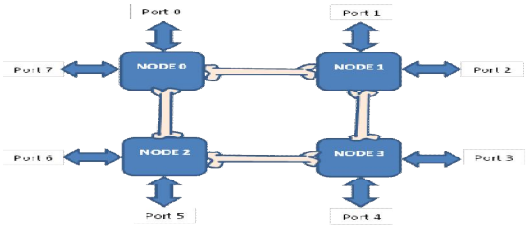

Figure 6: Four nodes NoC system

As in the above given figure.6.The proposed system consists of four nodes having four ports each. Two ports of each node are used for internal inter-node communication i.e. east and south port of node 0 is used for communication with node 1 and node 2 through west port and north port respectively. Consider an example where Port 4 wants to send a packet to NODE 0 and X-Y routing is used. NODE 4 would therefore say < WEST, 0> where WEST would be the direction the packet should take right now and 0 is the ultimate destination. NODE 3 would then send the packet to the WEST and at the same time perform the next route lookup and send < NORTH, 0> to node 2.

In this way a 2x2 array is implemented on the FPGA using the wish-bone technology. Therefore, each port (port 0, port 1, port 2, port 3, port 4, port 5, port 6 or port 7) can communicate with the rest of the ports using the wish-bone bus through the interconnecting nodes.

8. Hardware Implementation

As 36-bit version of the proposed system could not be implemented on Virtex-5 FPGA, a 8-bit version of the same

is implemented and tested using the previously discussed testing scheme in VII.

The system is tested for transmission of data from port 4 of node 3 to port 0 of node 4. The DIP switches are used as input to the system and LED for representing the output of system. The GPIO_DIP_SW1 switch is used as an active high reset for the system. The GPIO_DIP_SW8 switch is used as the input representing input transmission data to the source port. A red LED designated as 'Error 1' is used to represent the transmission of data at the destination port. An internal clock pulse of 33 MHz is used which is designated as CLK_33MHZ_FPGA. Whenever the GPIO_DIP_SW8 is turned on or made '1' then the data is transmitted from source port to destination port making the red LED glow. And if the GPIO_DIP_SW8 is kept low then no transmission takes place keeping the LED OFF. Also, by default the output LED is a sourcing one i.e. it is at logic '1'; so when the system is reset then output LED is set ON.

9. RESULTS

The resource utilization of our design is shown below and compared with four other publications. When compared to the packet switched architecture in [20] as their highest grade FPGA is Virtex-4 and Highest operating frequency mentioned is 272 MHz but we worked on Vertex -5 and our architecture is operated at the frequency of 305.573 with higher grade FPGA technology whereas our switch only uses 20% of the slices and 25% of LUTs and 55% of flip-Flops and Latency reduced to 2. When compared to [22], the system is capable of operating at a significantly higher frequency while being only slightly larger. The authors also do not mention how deadlocks are avoided or handled in their design. The latency of their NoC is also unknown. Our NoC is operating almost double clock frequency than NoCem [23] with less resource usage.

10. CONCLUSION

In this paper we have presented a low latency, low area overhead & high throughput an open source Network-on-Chip architecture optimized for the Virtex-5 FPGA. The network can operate at over 305.573 MHz and the area for a NoC switch is significantly smaller than for previous results at double operating frequency. We have also used a bridge which allows Wishbone compatible components to be connected to this NoC.

Table 4 : The Performance Of Presented Noc Compared To Other Fpga Based Nocs.

	Data Width in bits	Virtex-II 6000-4	Virtex-II Pro 30-7	Virtex-4 LX80-12	**Virtex-5 xc5v1x110t-3**	Latency (cycles)	Slices	LUTs	Flip Flops
Present work 4 port	36	166 MHz	257 MHz	232.561MHz	**301.12 MHz**	**2**	**421**	**772**	**452**
Present work 5 port	36	151 MHz	244 MHz	260 MHz	**305.573 MHz**	**2**	**519**	**759**	**572**
[20] 5 port	36	151 MHz	244 MHz	260 MHz	-	3	659	826	615
[17] 4 port	36	166 MHz	257 MHz	272 MHz	-	3	431	780	452
[18] 4 ports	32	166 MHz	-	-	-	6	1464	-	-
[20] 4 ports NoC	32	-	138 MHz	-	-	-	364	-	-
[21] NoCem	32	-	150 MHz	-	-	-	-	1455	-

Table 5: Device Utilization Summary

Device utilization summary			
Selected Device : 5vlx110tff1136-3			
Slice Logic Utilization:			
	Used	Out of	%
Number of Slice Registers:	14730	69120	21
Number of Slice LUTs:	17057	69120	24
Number used as Logic	14616	69120	21
Number used as Memory	2441	17920	13
Number used as RAM	8	-	-
Number used as SRL	2433	-	-
Slice Logic Distribution			
Number of LUT Flip Flop pairs used:	22265		
Number with an unused Flip Flop	7535	22265	33
Number with an unused LUT	5208	22265	23
Number of fully used LUT-FF pairs	9522	22265	42
Number of unique control sets	821	-	-
IO Utilization			
Number of IOs	12	-	-
Number of bonded IOBs	12	640	1
Specific Feature Utilization:			
Number of Block RAM/FIFO	49	148	33
Number using Block RAM only	49	-	-
Number of BUFG/BUFGCTRLs	2	32	6
Timing Summary:			
Speed Grade: -3			
Minimum period: 3.273ns (Maximum Frequency: 305.573MHz)			
Minimum input arrival time before clock: 2.468ns			
Maximum output required time after clock: 2.775ns ≈ 2 clock cycles.			

REFERENCES

1. Khalid latif, Tiburiu Seceleanu , hannu Tenhunen **Power and Area Efficient Design of Network-on-Chip Router through Utilization of Ideal Buffer** *Proc.17th IEEE International Conference and workshop on Engineering of Computer based System.* , 2010.
2. Luca Benini , Giovanni De Micheli, **Networks on Chips, Morgan`** *Kaufmann Publishers*, 2006.
3. W. Hangsheng, L. S. Peh, and S. Malik. **Power- driven design of router micro architectures in on-chip networks**. *Proc. of the 36th Annual IEEE/ACM International Symposium on Micro architecture (MI-CRO)*, pp.105-116, 2003.
4. Xuning Chen , Li-ShiuanPeh, **Leakage power modeling and optimization of interconnection networks** *Proc. of International Symposium on Low Power Electronics and Design*, pp. 9095, (2003).
5. N. Banerjee, P. Vellanki and K.S. Chatha. **A Power and Performance Model for Network-on-Chip Architectures** *DATE*, pp.1250-1255 ,2004,Vol.2,
6. ARTISAN **A comparison of network-on-chip and busses**, *white paper*, 2005.
7. W. J. Dally , B. Towles, **Principles and practices of interconnection networks** , *Morgan Kaufmann Publishers*, 2004.
8. T. Bartic, **Topology adaptive network-on-chip design and implementation**, *IEEE Proc. Comput. Digit Tech.*, vol. 152, no. 4, pp. 467– 472, Jul. 2005.
9. E. Beigne, **An asynchronous NOC architecture is providing low latency service and its multi-level designframework**, *in ASYNC*, Mar., pp. 54–63. 2005
10. L. Benini , G. de Micheli, **Networks on chips: A new SoC paradigm**, *IEEE Computer*, vol. 35, no. 1, pp. 70–78, Jan. 2002.
11. J. Owens et al., **Research challenges for on-chip interconnection networks**, *IEEE Micro*, vol. 27, no. 5, pp. 96–108, Sep-Oct.2007.
12. T. Bjerregaard , S. Mahadevan, **A survey of research and practices of network-on-chip,** *ACM Computing Surveys*, vol. 38, no. 1, p. article No. 1, Jun. 2006.
13. T. Bjerregaard , J Sparose, **A router architecture for connection oriented service guarantees in the MANGO clock less network-on-chip**, *in DATE*, vol. 2, p p. 1226-1231. Mar.2005.
14. C. Wang et al, **Area and power efficient innovative NoC Architecture**, *Proc. of 18thEuroMicro International Conference, PDP 2010*, Pisa, Italy, 2010.
15. Sudhir N. Shelke, Pramod B. Patil, **Dynamic and Leakage Power Optimization Using Power Design Methodology-Gate level Power Optimization for Implementation of 2-D Mesh Network-On-Chip-Router** *IJEEER*, Vol.3,Issue.1, March-2013,PP-147-160.TJPRC.
16. W. Dally , B. Towles, **Principles and Practices of Interconnection Networks**, *Morgan Kaufmann*, 2004.
17. A. Ehliar , D. Liu, **An FPGA based open source network-on-chip architecture**, *17th FPL*, pp.800-.803, *IEEE Amsterdam*, Holland, 2007.
18. N. Kapre, N. Mehta, M. deLorimier, R. Rubin, H. Barnor, M. J.Wilson, M. Wrighton, and A. DeHon, **Packet switched vs. time multiplexed FPGA overlay networks** , *Symposium on Field-programmable Custom Computing Machines, IEEE* 2006.
19. C. Hilton , B. Nelson, **PNoC: a flexible circuit-switched NoC for fpga-based systems**, vol. 153, *Computers and Digital Techniques, IEEE Proceedings*-2006.
20. G. Schelle, D. Grunwald, **Onchip interconnect exploration for multicore processors utilizing**

FPGA's, 2nd *Workshop on Architecture Research using FPGA Platforms*, 2006.
21. T. Mak, P. Sedcole, P. Y. Cheung,Luk, **On-FPGA communication architectures and design factors**, *16th International Conference on Field Programmable Logic and Applications*, 2006.
22. D. Wiklund and D. Liu, **SoCBUS: switched network on chip for hard real time embedded systems,** *Parallel and Distributed Processing Symposium. Proceedings. International*, 2003.
23. "**Wishbone system-on-chip (soc) interconnection architecture for portable IP cores**, http://www.opencores.org/, 2002.

ABOUT AUTHORS

Sudhir Shelke was born in Nagpur, India in Jun 1976. He has received Degree of Engineering in 2000 in Electronics Engineering from SRTM University, Nanded, India and He has obtained Master of Engineering in 2007 in Digital Electronics from SGB Amravati University, India. He is currently working as a Head of Department of Electronics Engineering / Electronics & Telecommunication Engineering, J D College of Engineering & Management, Nagpur; He has More than 14 years of Teaching & Research experience; His principal research area includes VLSI Chip Design, Embedded System Design.etc. His undergoing Research is on Low Power VLSI Designs Techniques & Physical Challenge for his doctorate Program. He has published twenty papers in International journals, International Conferences & National Conferences so far. He is reviewer of Micro Electronics Journal of Elsevier. His one of the paper is awarded as the best paper of the session of International Conference ICSCI-2008 Hyderabad, India. He had been worked on PCI Express 3rd Generation I/O Interconnect.

Dr. Pramod B. Patil was born in Amravati, India in May 1968. He has received Degree of Engineering in 1989 in Electronics Engineering from SGB Amravati University, India and he has obtained Master and Doctorate in Engineering in 1997 & 2007 respectively. Currently working as a Principal in J.D. College of Engineering & Management Nagpur; He has More than 22 years of Teaching & Research experience. His principal research area includes VLSI Chip Design, Digital Image Processing, & Signal Processing etc. He is reviewer of IEEE transaction of Signal, Speech & Audio processing & associate editor of two international Journals, He has more than 37 research papers in International Journals & International conferences on his credit so far and many research scholars are working under his esteemed guidance.