Design And Implementation Of An Efficient Noc Router

1st Smita P. Dharmale
Department of Electronics & Telecommunication
Sipna College of Engineering & Technology, Amravati,
SGBAU Amravati University - 444602 (India)

2nd Dr. P. R. Gumble
Professor Department of Electronics & Telecommunication
Sipna College of Engineering & Technology, Amravati,
SGBAU Amravati University - 444602 (India)

***Abstract*—Networks-On-Chip (NoCs) have been proposed as a promising solution for power, performance demands and scalability of next generation Systems-On-Chip (SOCs) to overcome the several challenges of current SoC with conventional architecture. In this article, NoC, its architecture and features are presented. Further the article is extended with research challenges. Major areas of scope for research are addressed briefly with the view that microelectronic field researchers get benefitted. Performance analysing parameters and simulation tools for NoC are also included. Future SoC design needs lot of innovations and creativity to explore its complete features. Research on NoC is mandatory at this critical juncture. This paper provides complete design and implementation of network on chip router architecture. Router is the basic element of NoC which have multiple connections connected to other router and to a local processing element in NoC. This router architecture used for building a NoC with low latency, high speed and high maximum peak performance. Low latency and high speed is achieved by allowing router function for each input port and distributed arbiter which gives high level of parallelism This paper presents an On-chip routers design using First In First out (FIFO) buffer. The concept of virtual channel (VC) is eliminated from the previous design by using an efficient flow control scheme that uses the storage already present in pipelined channels in place of explicit input virtual channel buffer. Through this design power is saving up to 66.66% and delay is reduced by 99.80% as compared to generic and power is saving up to 60.48% and delay is reduced by 90.88% as compared to virtual channel router.**

***Index Terms*—Network on Chip, System on Chip, Power Dissipation, Research Challenges, NOC Architecture, Network Interface**

I. Introduction

Advancements of deep submicron technology highlighted the criticality of the on-chip interconnects. As diminishing features sizes have led increases in global wiring delays. NoC architectures are look up as possible solution to the wiring challenges. Both NoC performance and energy budget depends mostly on the router buffer resources. Technology scales continuously increasing number of components and complexity for soc design. Commonly used shared bus on chip interconnect is not a good choice for large systems due to global wiring delays, noise, power dissipation and complexity of arbitration. With the advancements in IC technology, the gate delay decreases which lead to relatively increases the wire delay. The wire delay decides the overall performance of the system. Many researchers are trying to solve long global wire problem through the buffer insertion. Many SoCs use a system bus to connect several functional units. SoC system bus can support only limited number of functional units and there may be a scalability problem in heterogeneous multiprocessor system on chip (MPSoC). In order to solve these long global wire delay and scalability issues, many researchers suggested Packet based communication Network which is known as Network-on-Chip (NoC). NOC is used to connect many functional units (IP) with a universal com- munication network [1,2,3] global synchronization becomes harder to achieve due to costly clock skew[4,5]. Under such circumstances, on chip global wires begin to behave like transmission lines, which requires higher power consumption and chip resources in order to meet timing constraints. To overcome the above problem, here use the communication centric approach to integrate functional elements in complex SoCs. This new design allows the decoupling of the processing element from the network. Therefore global synchronization is not required. Today many router designs use first-in-first- out (FIFO) buffers to synchronously propagate data over large distances to overcome this problem. Clocked and synchronous FIFOs are essentially the same.

II. Proposed Methodology

Fig. 1 shows the block diagram of proposed router which has three main blocks, First in First out (FIFO) buffer, crossbar Switch and arbiter. The Router is packet switched and it provides five input/output ports namely local, North, East, South and West, to communicate with the local logic element and neighboring routers. In proposed architecture, data transfers by segmenting longer messages into smaller data packets, and forwarding these packets individually from sender to the receiver possibly with different routes and delays for each packet. Packet switching offers the potential for

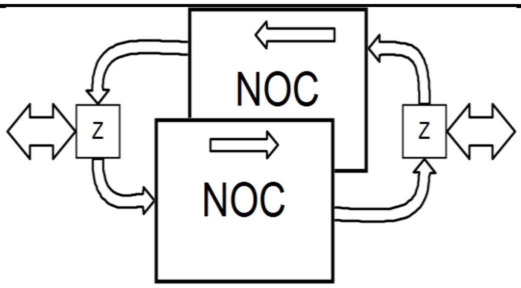

Fig. 1. Basic Block Diagram Of NOC

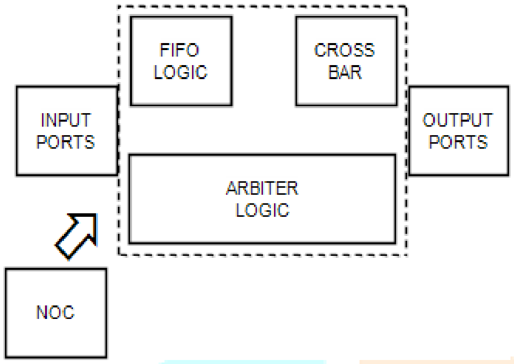

Fig. 2. Inside NOC

scalability.Packets are composed of fields as shown in Fig.4.2, each field carrying specific information. Here packet size is of 40 bits. The first part is the header that contains three bit source address. Second part is destination address of three bit and remaining bits indicate payload portion in which user specify its contents. Packetization of data allow to use wide interconnects for on-chip networks, thus increases the performance.

A. FIRST-IN, FIRST-OUT BUFFER

In proposed router buffering is required to provide temporary storage of packets that are in transit. There is one input channel at each port, each running its own finite state machine (FSM) control logic. Each input channel has a first in first out (FIFO) buffer of depth 4 and data width of 40 bits and a control Logic which has been implemented as a FSM. Complete transmission of packet occurs if and only if FIFO buffer of that input channel is not full and width of the buffer storage and the on-chip interconnect equal to the packet size. Thereby avoiding the need to transfer a packet in segments. In this manner, the requirement for full packet reception is easily met with reduced complexity. The input channel sets the acknowledge line high, as long as there is a transfer taking place. The packet of data received from the previous router is stored locally in the FIFO buffer thereby implementing a store-and-forward data flow.

B. CROSSBAR ARCHITECTURE

Multiplexer based crossbar switch shown in Fig.3.4. Single large crossbar switch used in this proposed router which

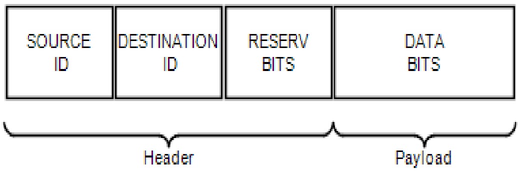

Fig. 3. Packet Format

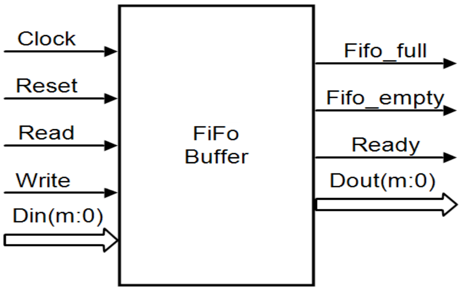

Fig. 4. Fifo Block

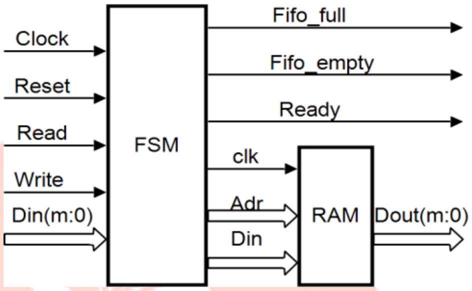

Fig. 5. Inside Fifo Block

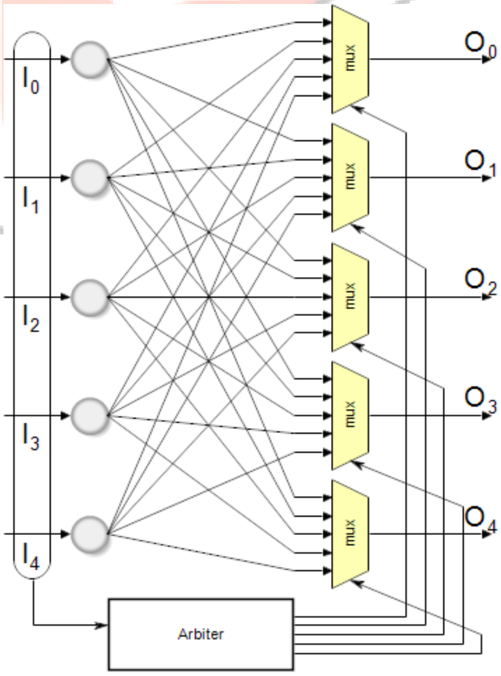

Fig. 6. Crossbar Architecture

Algorithm 1: pseudo code in algorithm form

```
Input  :  Clock        = clk signal
          Reset        = tia to main reset
          Read         = arbiter read signal
          Write        = write signal
          Din(m:0)     = data input bus
Output:   fifofull     = ram is full signal
          fifoempty    = ram is empty signal
          ready        = ready for read write
          clk          = ram clk
          adr          = ram adr
          din          = ram data
Initialize:
fifoemty = 1 ; fifofull = 0 ; Ready = 0 ; caddr = 0;
start:
if Write==1 then
    Ready = 0
    goto: wrsteps
else if Read==1 then
    Ready = 0
    goto: rdrsteps
end if
  wrsteps:
if vfifofull==1 then
    goto: start
else
    din = Din(m:0)
    adr = cadr
    clk = 0 1 0
    cadr = cadr + 1
    goto: statusfifo
end if
  rdrsteps:
if vfifoempty==1 then
    goto: start
else
    adr = cadr
    outdata = rdata
    cadr = cadr - 1
    goto: statusfifo
end if
   statusfifo:
if caddr==0 then
    fifoemty = 1
    fifofull = 0
else if caddr > 0 AND caddr < 4 then
    fifoemty = 0
    fifofull = 0
else if caddr == 4 then
    fifoemty = 0
    fifofull = 1
end if
ready=0
goto: start
```

decreases router latency. In the proposed router architecture, crossbar switch connects the 5 inputs port to 5 outputs port in a matrix manner and it contains 5 multiplexer. Five, 5:1 multiplexers are used, one at each input to the crossbar. All five inputs to the crossbar is fed to each multiplexer. As five input packets of 40 bits each from five input ports of router, therefore five, 5:1 multiplexers used inside the crossbar. All five inputs are connected to all the multiplexers. Which input is forwarded to the output is decided by the select lines generated by the arbiter. Select line is of three bit. Out of five select lines for five different multiplexer which one is selected is depend on the logic of arbiter. Outputs of multiplexers are the output ports of router. Three select lines are generated by arbiter control logic by reading header information of receiving packets. From that information arbiter computes the number of multiplexer i.e. select output channel. Once multiplexer is selected for outputting data out of router, then three select lines as a input to that multiplexer decides which input will outputted among five inputs coming to that selected multiplexers.

C. ARBITER

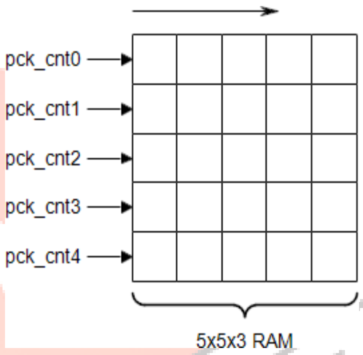

Fig. 7. arbiter ram Architecture

Arbiter controls the arbitration of the ports and resolves contention problem. It keeps the updated status of all the ports and knows which ports are free and which ports are communicating with each other. Packets with the same priority and destined for the same output port are scheduled with a round-robin arbiter.Suppose in a given period of time, there was many input ports request the same output, the arbiter is in-charge of processing the priorities among many different request inputs. The arbiter will release the output port which is connected to the crossbar once the last packet has finished transmission. So that other waiting packets could use the output by the arbitration of arbiter. In proposed work, round robin arbitration algorithm use to assign priorities when many input ports request the same output. A round-robin arbiter operates on the principle that a request which was just served should have the lowest priority on the next round of arbitration.

D. XY ROUTING

In XY routing, a packet is forwarded horizontally till the target column is reached and is then forwarded vertically to the

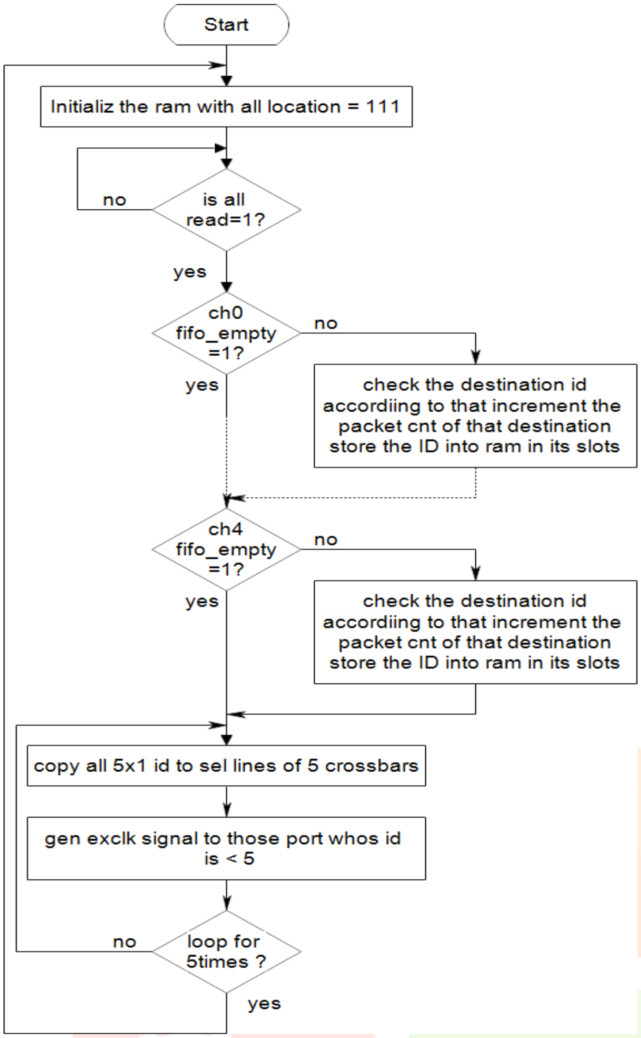
Start
Initializ the ram with all location = 111
no
is all read=1?
yes
ch0 fifo_empty =1?
no
yes
check the destination id accordiing to that increment the packet cnt of that destination store the ID into ram in its slots
ch4 fifo_empty =1?
no
yes
check the destination id accordiing to that increment the packet cnt of that destination store the ID into ram in its slots
copy all 5x1 id to sel lines of 5 crossbars
gen exclk signal to those port whos id is < 5
no
loop for 5times ?
yes

Fig. 8. Arbiter Flowchart

destination router. This means that there is no request for the East or West output ports by the North or South input ports. This fact is exploited and the FSMs of the mentioned output channels are simplified, as they need not service the mentioned input ports.. This translates to significant area saving and reduction in number of clock cycles in servicing requests. In proposed work this routing scheme is used be since it is simple and easy to adopt. This helps in the implementation of a light weight router, having area overheads at the minimum with acceptable level of performance.

E. Implementation

Design of NoC router with single crossbar architecture has been implemented in VHDL and is synthesized for FPGA technology. The router designed consists of four stages; routing, arbitration control logic, crossbar traversal, and four places FIFO buffer for each input channel. The functionality of each stage is replicated at each port to support concurrent connections for overall router. Fig. 5.1 illustrates the three components that implement the functionality of each stage; crossbar switch, arbitration unit, and four places FIFO buffer. Five read signals for five different input ports is as shown in fig.5.2 .Read signals are generated by considering current status of FIFO buffer signals of that particular port only. Fig. 5.3 shows the design of crossbar using five, 5:1 multiplexers.

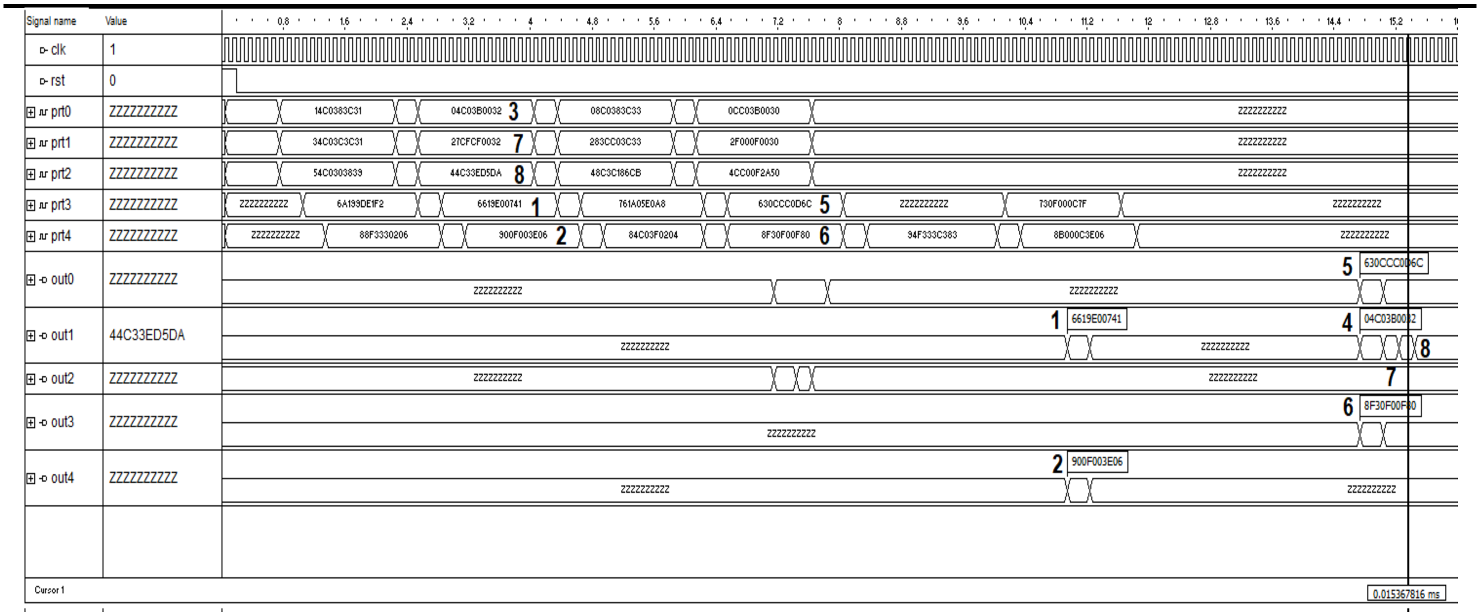

Fig. 9. Noc WaveForm

III. Conclusion and Disscusion

The overall router design provides general Network on Chip support with reduced complexity, thereby achieving area efficiency in conjunction with field programmable gate array device. Light-weight parallel router architecture for implementing Networks-on-Chip is implemented on FPGAs. In this implement optimizations in the FSMs and decoding logic, thereby, saving significant area. A majority of the efficiency for router comes from adaptation of single crossbar design. Such a design allows the switch module to be more area-efficient than other equivalent module. The routing algorithms adapted for crossbar design provide low-complexity implementation of their logic. Complexity is further reduced by using the store-and-forward switching technique. A handshaking signal is used to provide low-overhead link-level flow control for reli- able communication between router's ports. FSM description control logic has been used to test the router with different scenarios to ensure correctness. The operational results have shown the functionality of the router to reflect the fact that the router design is suitable not only as an embedded module in future FPGAs, but also as a conventional programmable design in existing FPGAs.

References

[1] "A survey of research and practices of Network-on-chip,ACM Computing Surveys (CSUR)", T. Bjerregaard,S. Mahadevan, ,Volume 38 ,Issue1 (2006)

[2] "Low-power network-on-chip for highperformance SoC design" IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Volume 14, Issue 2, Feb. 2006, pp.148-160. by K. Lee, S. J. Lee, H. J. Yoo

[3]S. J. Lee, K. Lee; H. J. Yoo, "Packet -Switched On-chip Interconnection Network for System-on-Chip Applications", IEEE Trans. on Circuit and Systems , Vol. 52, N.06, June 2005.

[4]P.P. Pande et al.,"Performance Evaluation and Design Trade-offs for Network-on-Chip Interconnect Architectures", IEEE Trans. Computers, vol. 54, no. 8, Aug. 2005, pp. 1025-1040.

[5]D. Bertozzi, A. Jalabert, et. al.," NoC Synthesis Flow for Customized Domain Specific Multiprocessor Systems-on-chip". IEEE Transaction on Parallel and Distributed Systems, Vol. 16, NO.2 Feb, 2005.

[6]E. Rijpkema, K. G.W. Goossens, and A. Radulescu, "Trade-offs in the design of a router with both guaranteed and best-effort services for networks-on-chip," in Proceedings of the Design Automation and Test in Europe Conference, pp. 350–355,March 2003.

[7]H. S. Wang, L. S. Peh, and S. Malik, "A power model for routers: modeling alpha 21364 and InfiniBand routers," IEEE Micro, vol. 23, no. 1, pp. 26–35, 2003.

[8]R. Mullins, A. West, and S. Moore, "Low-latency virtualchannel routers for on-chip networks," in Proceedings of the 31st Annual International Symposium on Computer Architecture, pp. 188–197, June 2004.

[9]K. Kim, S. J. Lee, K. Lee, and H. J. Yoo, "An arbitration lookahead scheme for reducing end-to-end latency in networks on chip," in Proceedings of the IEEE International Symposium on Circuits and Systems, (ISCAS '05), pp. 2357–2360, May 2005.

[10]P. Guerrier and A. Greiner, "A generic architecture for onchip packet-switched interconnections," in Proceedings of the Design Automation and Test in Europe Conference, pp. 250–256, March 2000.

[11]R. Hegde and N. R. Shanbhag, "Toward achieving energy efficiency in presence of deep submicron noise," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 8, no. 4, pp. 379–391, 2000.

[12]L. Benini and G. De Micheli, "Networks on chips: a new SoCparadigm," IEEE Transactions on Computers, vol. 35, no. 1, pp.70–78, 2002.

[13]L. Benini and G. D. Micheli. Networks on chips: A new SoC paradigm. IEEE Computer, 35(1):70–78, January 2002.

[14]L. Benini and G. D. Micheli, editors. Networks on Chips: Technology and Tools. Morgan Kaufmann, 2006.

[15]A. Benveniste and G. Berry. The synchronous approach to reactive and realtime systems. Proceedings of the IEEE, 79(9):1270–1282, 1991.

[16]A. Benveniste, P. Caspi, S. A. Edwards, N. Halbwachs, P. L. Guernic, and R. D. Simone. The synchronous languages 12 years later. Proceedings of The IEEE, 91(1):64–83, January 2003.

[17]T. Bjerregaard and S. Mahadevan. A survey of research and practices of network-on-chip. ACM Computing Survey, 38(1):1–54, 2006.

[18]T. Bjerregaard and J. Sparso. A router architecture for connection-oriented service guarantees in the MANGO clockless network-on-chip. In Proceedings of the Design, Automation and Test in Europe Conference, pages 1226– 1231, 2005.

[19]E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny. QNoC: QoS architecture and design process for network on chip. The Journal of Systems Architecture, December 2003.

[20]J. T. Brassil and R. L. Cruz. Bounds on maximum delay in networks with deflection routing. IEEE Transactions on Parallel and Distributed Systems, 6(7):724–732, July 1995.

[21]L. P. Carloni, K. L. McMillan, and A. L. Sangiovanni-Vincentelli. Theory of latency-insensitive design. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 20(9):1059–1076, September 2001.

[22]S. Chalasani and R. V. Boppana. Fault-tolerant wormhole routing algorithms for mesh networks. IEEE Transactions on Computers, 44(7):848–864, 1995.

[23]T. Claasen. An industry perspective on current and future state-of- the-art in system-on-chip (SoC) technology. Proceedings of the IEEE, 94(6):1121– 1137, June 2006.

[24]M. Coppola, S. Curaba, M. Grammatikakis, and G. Maruccia. IPSIM: SystemC 3.0 enhancements for communication refinement. In Proceedings of Design Automation and Test in Europe, 2003.