p-ISSN : 2335-1357 e-ISSN : 2437-1122

Mediterranean Journal of Modeling and Simulation

Med. J. Model. Simul. 11 (2019) 036-048

Efficient Crossbar Switch Design For NOC

Vivek Tiwari [a] *, Kavita Khare [a]

[a] Department of Electronics and Communication Engineering MANIT, Bhopal, India

ARTICLE INFO

Article history :
Received May 2019
Accepted August 2019

Keywords :
Network-on-Chip ;
NoC router ;
Crossbar switch ;
Xilinx ;
RTL simulation ;
MUX.

ABSTRACT

Network-on-Chip is an emerging paradigm for integrating a very high number of Intellectual Property blocks on a single Integrated Chip. The crossbar switch is one of the important parts of NoC. In this paper, a new crossbar switch is proposed which reduces the complexity of conventional switch by simplifying its data transport mechanism and using the dedicated input-output channel between source and destination. The proposed design reduces area by 40% and delays by 7.14% as compared to 2-D crossbar switch and area by 40% and delay by 9.53% as compared to the conventional crossbar switch. The functional verification and synthesis of proposed and conventional Crossbar switch design is done by using Xilinx ISE 9.2i

1. Introduction

Device sizes are continuously reducing as a result of advancement in process technology. Due to this multiple processing elements (PEs) can be integrated into a single a chip, referred as system-on-chip (SoC). SoC technology offers many advantages over conventional approach such as higher performance and low power consumption. But in SoC due to a large number of PEs, communication between cores becomes very complicated. In some SoC system where communication links are much larger than the number of PEs multiple communication links may require and it can make these communication networks very complex.

In conventional on-chip communication architecture, it consists of bus-based infrastructure and point-to-point communication network, may not be able to provide adequate communication support for SoC. Generally, bus-based infrastructure and point-to-point communication network are used in conventional on-chip communication architecture, as the number of cores is increasing continuously over SoC, so communication infrastructures are not efficient. To provide better communication between cores of SoC Network-on-chip architecture is a better solution nowadays [1].

*Email : vivek642@gmail.com

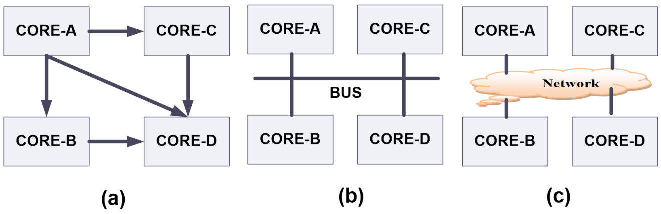


Fig. 1 – Connection between different cores of multiprocessor (a) Points to point connection (b) Bus based connection (c) Network-based connection[2].

Fig.1 (a) shows point to point connection between two modules through direct link assign to them. In this type connection, a large number of the link will require which will increase the size of the chip.

Because of long communication wire, there is an excessive signal delay. Fig.1 (b) shows bus-based communication network which is efficient and flexible compared to point to point communication. In bus-based communication the whole communication network is utilized only by two modules at a time because of this all module become idle at this time. Thus the bus may become a bottleneck of overall system performance where heavy communication is required. Fig.1(c) shows a packet switched networking infrastructure based solution that can be a promising solution for the above-discussed problem. This type of approach for communication between nodes referred to as network-on-chip (NoC)[3]–[6][2].

In NoC, the router is used to transfer data in a packet switched network; a package of data can be further divided into multiple flow information units (flits) that are actually transmitted. NoC greatly improves the scalability of SoCs and achieves higher power efficiency compared to other types of communication structures. As compared to other types of communication structure, NoC provides great improvement in scalability of SoC. it also provides higher efficiency in term of area, delay and area [7].

In NoC based communication system different cores such as processors, memories application specific integrated circuits and intellectual properties exchange their data through the NoC, which consists of router, data links, and network interfaces Data links are used to transmit data over communication media and NI provide interface between PE and a router, where it is Responsible for transforming data into packet and vice versa. Physical path for data transmission is decided by routers [8].

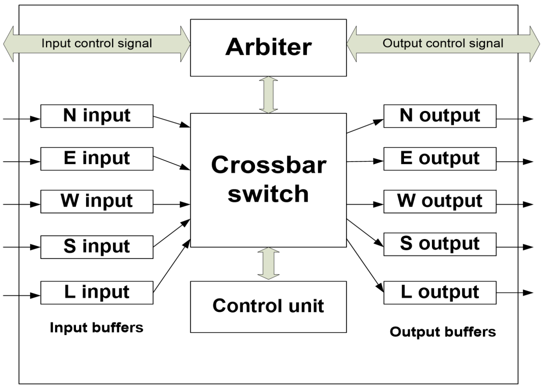


Fig. 2 – 5x5 NoC Router architecture [25]

A 5x5 NoC router architecture shown in Fig.2, which consists of the input port, an output port, an input buffer, arbiter and crossbar switch. In this NoC router, it is shown that a 5x5 crossbar switch is used to connect 5 input channels to 5 output channels [9][10].

In this paper new efficient crossbar switch, architecture is proposed, which also consists five input ports and five output ports. This crossbar switch provides a connection between north ports to south port, east port to west port and simultaneously, which is the main feature of our proposed work. It also provides a connection between all the four ports (i.e. east port, west port, north port, south port) to local port same as a conventional crossbar switch.

This proposed design is simulated in Xilinx 9.3 and its result i.e. RTL view, simulation, area, power delay is compared with 2-D crossbar switch and conventional crossbar switch[3]. In this proposed efficient crossbar switch area and delay significantly reduces[3]. And it is also proposed that when this crossbar switch is implemented within NoC router, it will become Faster and more efficient [11].

2. Related work

In NoC we use a Crossbar switch to connect input port to output port which is a collection of switches arranged in the form of matrix organization [12].It has various input and output lines arranged in the form of row and column. Input-Output lines get arranged in such a manner that they cross themselves and a connecting switch is placed between them, by closing this switch connection could be established. When switches at interconnection are closed connectivity is established between one of the inputs to an output.

The crossbar switch is a non-blocking kind of switch ; it does not affect any other simultaneous input, output connection. In any NoC router, crossbar switch and buffer used at input and output channel are main power and area contributor[13]–[18]. By using adaptive bandwidth control an area and power efficient crossbar switch are verified[19].

Segmentation and Decomposition these two methods are used for reduction of area and power[14], [15], [20]–[24]

Segmentation method is based that the power reduction is achieved by activating only the necessary wire segment that is responsible for connectivity between input and output ports by using a tri-state buffer. Decomposition method based on that the larger crossbar switch is decomposed into smaller crossbar switch due to which smaller crossbar area and less power consumption are achieved. But it bound connectivity between some input-output pair.

FPGA implementation of a reconfigurable crossbar switch is proposed to reduce area and make it flexible.[19].Modification in arbiter and network hardware is done to make crossbar switch lightweight and configurable FLEXBAR structure is proposed and implemented[25]A number of methods and technologies have been discussed to make crossbar switch more efficient and suitable for NoC architecture.

In conventional NoC router, a swizzle switch is proposed as a crossbar switch design [26].The switching stages have been improved with the help of virtual input crossbar switch which gives the opportunity to use more than one input virtual channel for flit transmission [27].

For bidirectional NoC router high-performance reconfigurable crossbar switch has been proposed which is capable to handle data traffic in different environment changes[12]. In modular decoupled crossbar design decomposition & segmentation technique is used, which divide crossbar switch design into three modules. In this crossbar structure design, it utilized extra tri-state buffer and feeder wire logic for communicating in inter-module paths, due to which its area and power consumption increases[13].

In 2-D crossbar switch, a control cell based on Nand-Nand architecture is used to reduce area and delay of crossbar switch [3]. In our work whole crossbar architecture is replaced by four 2x1 MUX and one 4 x1MUX based architecture which provides a significant improvement in area and delay.

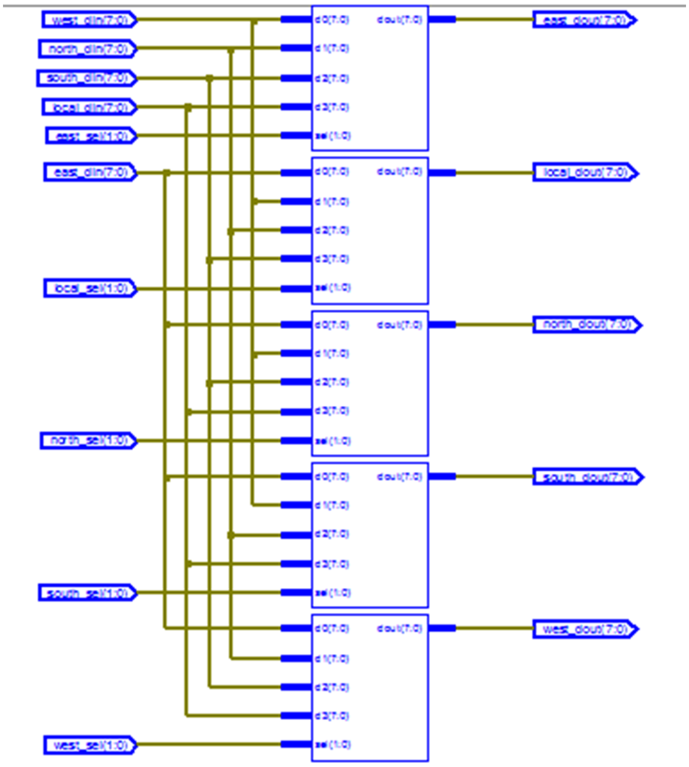


Fig. 3 – RTL view of the conventional crossbar switch

In conventional crossbar switch, five 4x1 MUX is used. Its RTL view is shown in Fig. It requires 40 slices and a total number of 80 four inputs LUT. Fig.7 is a block diagram of a slice of FPGA [10].

3. Proposed crossbar switch design

2-D Mesh-based NoC shown in Fig.4, where five port router is used as a switch to connect intellectual Property (IP) block (also known as processing element) to all direction switches i.e. East, West, North, South With the help of local input-output channel [28].

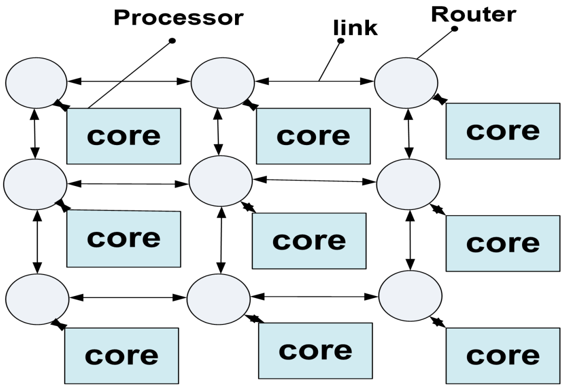


Fig. 4 – 2-D Mesh-based NoC

Conventional crossbar switch that generally used in every NoC router is used is shown in Fig.5 having four directional input channel, output channel and one local input-output channel as shown in fig.4 [29].

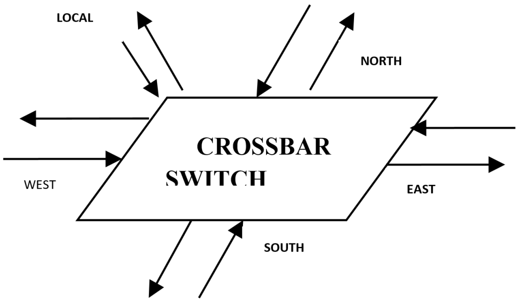


Fig. 5 – Crossbar switch for 2-D Mesh-based NoC and its all direction[3]

A block diagram of conventional crossbar switch as shown in Figure.6, When any input channel among the five input channels wants to transfer their data from source to destination, it requires to compete with four inputs request i.e. north request, south request, east request, west request and local request. Among these requests, at a time only one request of data transfer is possible to accept, which is decided by arbiter circuit. There are different arbitration techniques are available [30].due to these arbitration techniques and crowd of requests crossbar switches requires some more time to transfer data from input channel to output channel, which causes delay in data transformation, meanwhile these data are stored in input buffers and output so more number of buffer required at input and output channel [31].

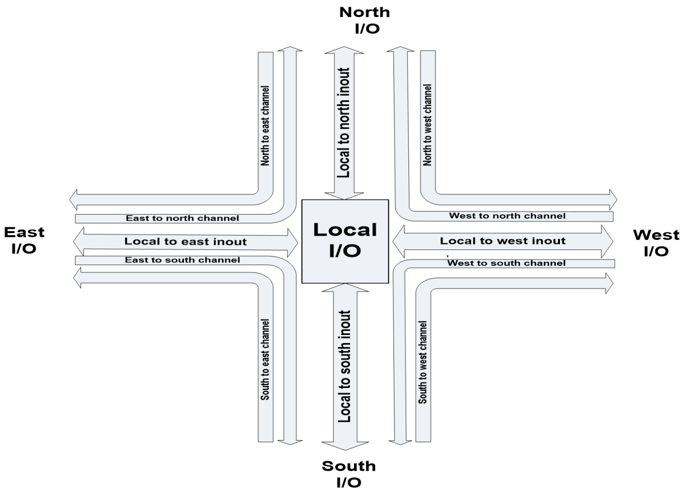


Fig. 6 – Block diagram of the logic of conventional crossbar switch

To solve above discussed problem a new crossbar switch is proposed. Block diagram of that crossbar switch is shown in Figure.7 In which it uses dedicated path to transfer data from north input to south output and east input to west output and simultaneously.

This new crossbar switch is designed in such a way that at every input channel only two requests are arriving and to select one request among these two requests arbiter is chosen in such a way that it always give higher priority to dedicated input-output and least priority to local request in this way there is no need to wait to transfer data from east to west, west to east, north to south and south to north direction. They are able to transfer their data immediately. Only data that come from local direction needed to wait until mainstream data are transmitted completely. So it required very less number of buffers at its input and output channels.

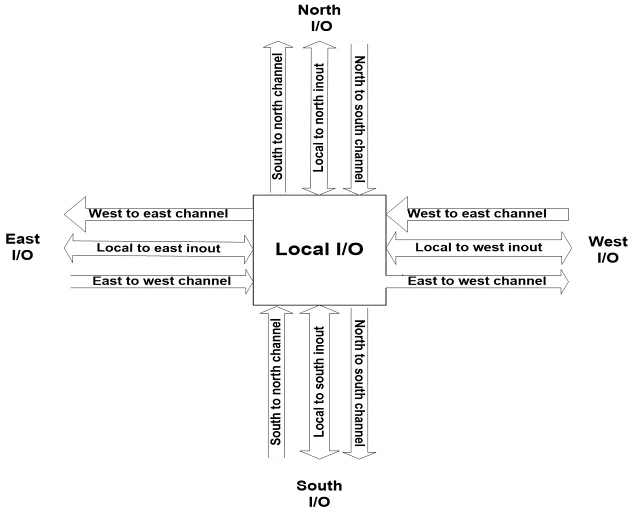


Fig. 7 – Block diagram of proposed crossbar switch logic

In this way delay and the number of buffers used in crossbar switch are reduced which makes this crossbar switch more efficient as compared to conventional crossbar switch[32].

The proposed crossbar switch internal structure is shown in Figure.9. It consist of four input channel in a row i.e. north input, south input, east input, west input and local input, four output channel in column i.e. north output, south output, east output, west output and local output, an internal structure as shown in block diagram it consist four 2x1 MUX where each MUX have two input channel, one output channel and one selection line[10].and it also consists of one 4x1 MUX which has four input channel and four output channel with two selection line[28].

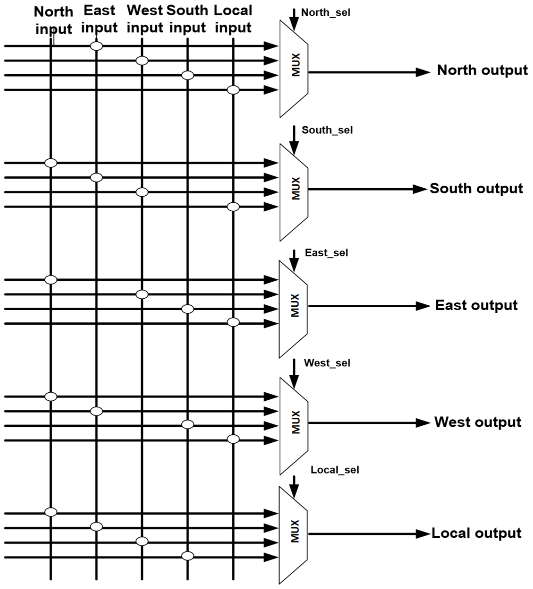


Fig. 8 – Block diagram of a Logical circuit used in the conventional Crossbar switch

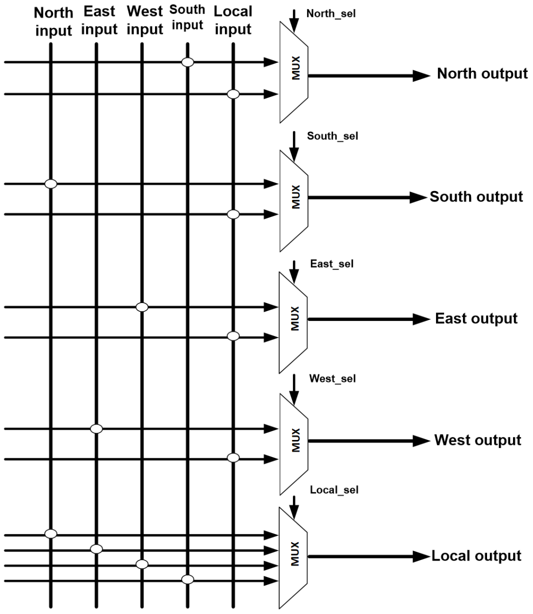


Fig. 9 – Block diagram of a proposed logical circuit for the new Crossbar switch

By implementing idea of four 2x1 MUX it is proposed that data will only transfer in one direction in each of four input direction, such as from north to south, south to north, east to west, west to east and in this type of data transmission method transmission path doesn't block by any other channel. In this way, it provides high-speed data transmission between two channels and also contributes to minimizing the area of the crossbar switch. Apart from providing connectivity between directional input and output channel, each four input channel (i.e. east (E), west (W), north (N), and south (S)) is also connected with local output channel to connect intellectual property (IP) to the communication network. In our proposed model it is used a 4x1 MUX to connect four directional input channel i.e. north input channel, south input channel, east input channel, and west input channel to local output channel such as conventional NoC router architecture. In conventional NoC router, it uses 4X1 mux for all five input channels as shown in Figure.8. [28].

Proposed digital circuit i.e. a crossbar switch is simulated in Xilinx9.2 ISE, where total delay from input to output port is calculated which is divided in two part i.e. delay due to the logic used in circuit and delay due to the path of the circuit mathematically it can be expressed as

$$t_{total} = t_{logic} + t_{route} \tag{1}$$

Where
t_{total} = total delay from input to output in ns
t_{logic} = total delay due to logic used in circuit in ns
t_{route} = total delay due to path in ns

Above proposed model is designed by using VHDL language and this design is simulated in Xilinx 9.2 ISE software, outcomes of results, i.e. RTL view of the proposed efficient crossbar switch is shown in Fig.10, where all input line, outline, and selection line is shown in the Fig.5 schematic diagram of the result is shown in Fig.8 where it is shown that four 2x1 MUX is connected with four input and output channel and a 4x1 MUX is used to connect four input channel to one output channel. Simulation result of the above-proposed model is shown in Fig.12. Power, delay, and area of the proposed model are analyzed by using x-power tools of Xilinx software. And the result of the proposed crossbar switch is compared with conventional and 2-D mesh crossbar switch that is shown in Table.2. The result shows significant improvement in proposed model as compare to 2-D mesh Crossbar switch as well as conventional crossbar switch [3].

4. RESULTS AND DISCUSSIONS

Design of proposed crossbar switch has been synthesized and simulated on Xilinx9.2i Spartan3E FPGA Device. RTL view and the simulation result of the proposed method are shown in Fig.12 and Fig.10 simultaneously. Fig.13 shows the results of switching of input to the output at different control signals in the form of a wave. Fig.11 shows the schematic view of the proposed crossbar switch.

An FPGA consists a slice to implement any Boolean logic and a slice consist of two 4 input LUT (look up table). In conventional crossbar design has 40 slices and 80 no. LUTs. In proposed crossbar switch since four 4x1 MUX are replaced by 2x1mux, it required only 24 slices and 48 LUT.

In Table.2 it is indicated the compression of delay, area (area in terms of a number of slice LUT's) and power of the proposed crossbar switch design with compare to 2-D crossbar switch design[3]. In table.2 FPGA Resource utilization of conventional and proposed crossbar switch is given. A proposed crossbar switch has a delay due to logic circuit i.e. $t_{logic} = 5.515ns$, due to path i.e. $t_{route} = 1.629ns$ and total delay i.e. $t_{(total)} = 7.144ns$, as compared to conventional crossbar switch which has delay due to logic circuit i.e. $t_{logic} = 6.289ns$, delay due to path i.e. $t_{route} = 3.720ns$ and total delay i.e. $t_{(total)} = 10.009ns$. it has 77.2% delay due to logic circuit and 22.8% due to path as compared to conventional crossbar switch which has 62.8% delay due to logic circuit and 37.2% due to path. Total memory usage in proposed crossbar switch is 163400 kilobytes as compare to conventional crossbar switch where memory usage is 163464 kilobytes.

Table 1 – Area, delay and power of proposed crossbar switch design and their compression

Crossbar switch	**Area (slice/lut count)**	**Power**	**Delay**
Conventional Crossbar switch	40/80	351mw	7.864ns
2-D Cross bar switch [9]	40/80	14 mw	7.694ns
Proposed Crossbar switch	24/48	102mw	7.144ns

Table 2 – Compression of FPGA Resource utilization of Conventional and Proposed crossbar switch (Selected device-Automotive Spartan 3E)

Resource Available		Conventional Crossbar Switch		Proposed Crossbar Switch	
		Used	Utilization	Used	Utilization
Slice	960	40	4%	24	2%
LUTs	1920	80	4%	48	2%
I OBs	66	90	136%	86	130%

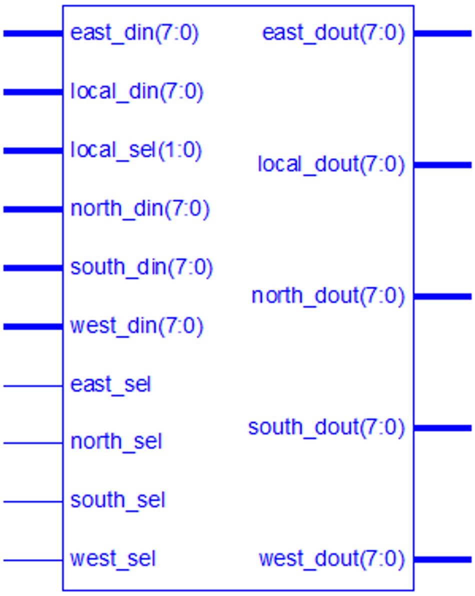


Fig. 10 – RTL view of proposed crossbar switch design

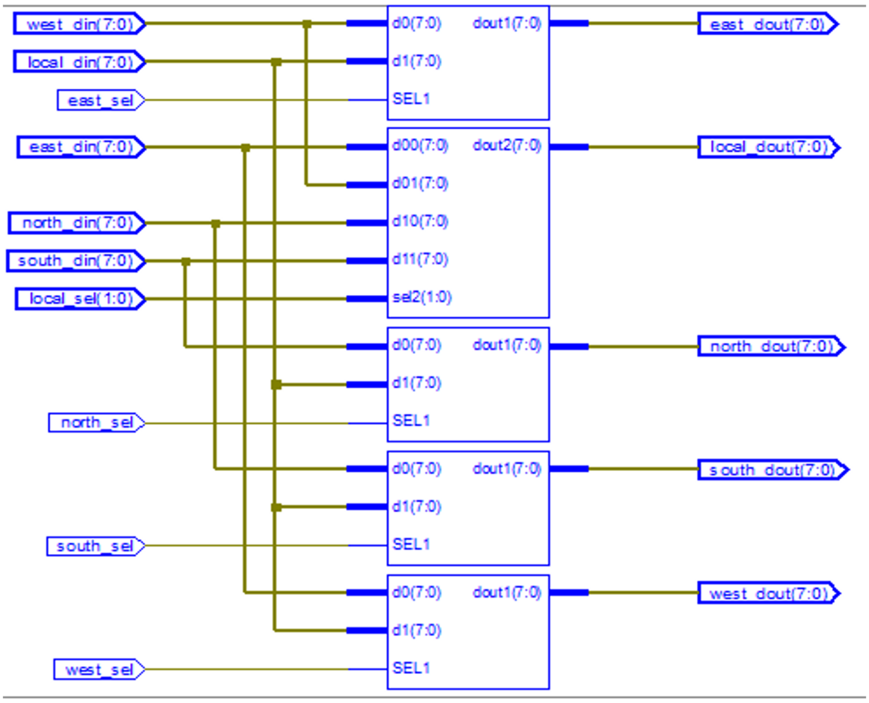
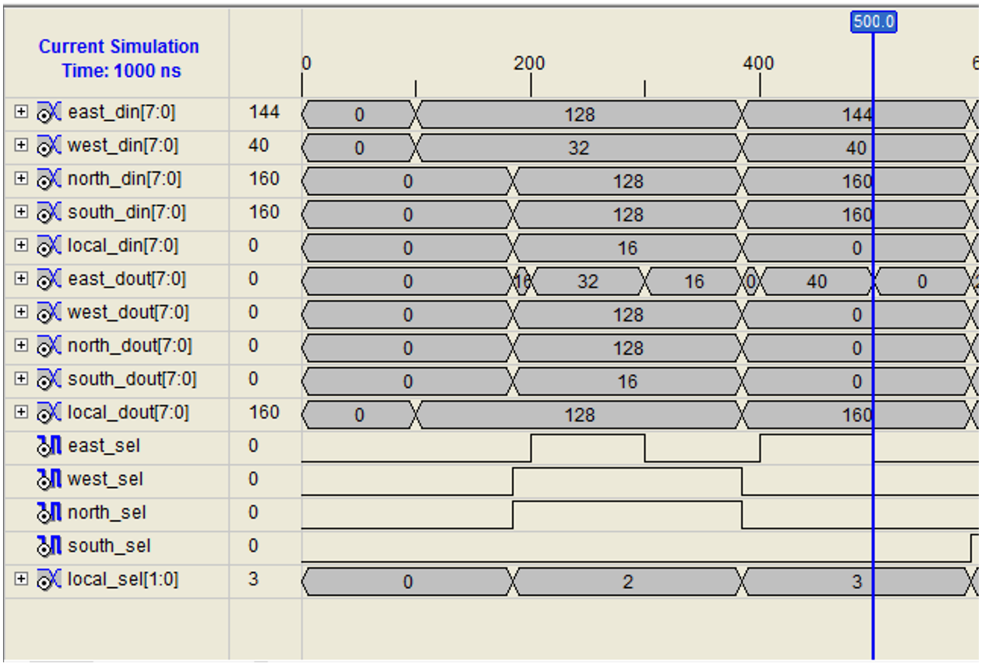


Fig. 11 – Schematic of proposed crossbar switch design


Current Simulation Time: 1000 ns

east_din[7:0]	144
west_din[7:0]	40
north_din[7:0]	160
south_din[7:0]	160
local_din[7:0]	0
east_dout[7:0]	0
west_dout[7:0]	0
north_dout[7:0]	0
south_dout[7:0]	0
local_dout[7:0]	160
east_sel	0
west_sel	0
north_sel	0
south_sel	0
local_sel[1:0]	3

Fig. 12 – Simulation result for the proposed crossbar switch

Conclusion

In this paper, an efficient crossbar switch design is proposed for high-speed NoC. a proposed crossbar switch has four 2x1 MUX and a 4x1 MUX as compared to Five 4x1

MUX in 2D existing crossbar switch. Simulation results show that delay is reduced by 7.14% and area by 40% as compared to 2-D crossbar switch and delay by 9.53 % and area by 40% as compared to the conventional crossbar switch.

5. Conclusion

In this paper, a new intelligent method for fault detection in PV module is introduced. The case of partial shading effect is studied. The main advantage of the developed ANNBM is that doesn't require a complex system for the estimation of the photovoltaic module output power, neither a mathematical model, it can also detect any power decreasing carried out by a large types of failures that can be happened in the PV panel. This new strategy can be easily implemented in a numeric calculator using FPGA, and could also be integrated as a function for PV applications in a numeric instrument that will be our subject in the future works.

REFERENCES

[1] A. Ehliar and D. Liu, "An FPGA Based Open Source Network-on-Chip Architecture," in Field Programmable Logic and Applications, 2007. FPL 2007. International Conference on, 2007.

[2] K. S. Li, B. Core, A. Core, B. Core, and C. Core, "CusNoC : Fast Full-Chip Custom NoC Generation," vol. 21, no. 4, pp. 692–705, 2013.

[3] S. Bansal, "Design of Configurable Power Efficient 3-Dimensional Crossbar Switch For Network-on-Chip (NoC)," pp. 1–5, 2016.

[4] B. S. Feero and P. P. Pande, "Networks-on-chip in a three-dimensional environment : A performance evaluation," IEEE Trans. Comput., 2009.

[5] K. Sewell et al., "Swizzle-switch networks for many-core systems," IEEE J. Emerg. Sel. Top. Circuits Syst., 2012.

[6] R. Marculescu, U. Y. Ogras, L. S. Peh, N. E. Jerger, and Y. Hoskote, "Outstanding research problems in NoC design : System, microarchitecture, and circuit perspectives," IEEE Trans. Comput. Des. Integr. Circuits Syst., 2009.

[7] M. Oveis-Gharan and G. N. Khan, "Efficient Dynamic Virtual Channel Organization and Architecture for NoC Systems," IEEE Trans. Very Large Scale Integr. Syst., 2016.

[8] B. P. Shrivastava and K. Khare, "Synthesis and Simulation of Enhanced Buffer Router vs. Virtual Channel Router in NOC ON Cadence," vol. 7, no. 2, pp. 285–289, 2013.

[9] L. Mingche, G. Lei, S. Wei, and Z. Wang, "Escaping from blocking : A dynamic virtual channel for pipelined routers," in Proceedings - CISIS 2008 : 2nd International Conference on Complex, Intelligent, and Software Intensive Systems, 2008.

[10] P. Poluri, S. Member, and A. Louri, "Shield : A Reliable Network-on-Chip Router Architecture for Chip Multiprocessors," vol. 9219, no. c, 2016.

[11] B. P. Shrivastava and K. Khare, "Area and Power Efficient Router Design for Network on," vol. 3, no. 5, pp. 1–5, 2013.

[12] A. Khodwe, V. K. Rajput, P. C. N. Bhoyar, and P. P. M. Nerkar, "VHDL Implementation Of Reconfigurable Crossbar Switch For Binoc Router," vol. 2, no. 5, pp. 150–156, 2013.

[13] D. Park, A. Vaidya, A. Kumar, and M. Azimi, "MoDe-X : Microarchitecture of a

layout-aware modular decoupled crossbar for on-chip interconnects," IEEE Trans. Comput., 2014.

[14] J. Kim, C. Nicopoulos, D. Park, V. Narayanan, M. S. Yousif, and C. R. Das, "A gracefully degrading and energy-efficient modular router architecture for on-chip networks," in Proceedings - International Symposium on Computer Architecture, 2006.

[15] H. Wang, L. S. Peh, and S. Malik, "Power-driven design of router microarchitectures in on-chip networks," in Proceedings of the Annual International Symposium on Microarchitecture, MICRO, 2003.

[16] C. H. Hoo and A. Kumar, "An area-efficient partially reconfigurable crossbar switch with low reconfiguration delay," in Proceedings - 22nd International Conference on Field Programmable Logic and Applications, FPL 2012, 2012.

[17] L. Mhamdi, K. Goossens, and I. V. Senin, "Buffered crossbar fabrics based on networks on chip," in CNSR 2010 - Proceedings of the 8th Annual Conference on Communication Networks and Services Research, 2010.

[18] K. Goossens, L. Mhamdi, and I. V. Senín, "Internet-router buffered crossbars based on networks on chip," in 12th Euromicro Conference on Digital System Design : Architectures, Methods, and Tools, DSD 2009, 2009.

[19] H. Freitas, M. Carvalho, and A. Amaral, "Reconfigurable crossbar switch architecture for network processors.," Iscas, pp. 4042–4045, 2006.

[20] S. Abovyan, G. Petrosyan, and T. Harutyunyan, "Architecture of queued-free crossbar for on-chip networks," in Proceedings of IEEE East-West Design and Test Symposium, EWDTS'10, 2010.

[21] S. Swapna, A. K. Swain, and K. K. Mahapatra, "Design and analysis of five port router for the network on chip," in the Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics, 2012.

[22] S. Kumar and A. Srivastava, "Design and implementation of crossbar switch in NS2," in 5th International Conference on Computing Communication and Networking Technologies, ICCCNT 2014, 2014.

[23] Y. L. Lee, J. M. Jou, and Y. Y. Chen, "A high-speed and decentralized arbiter design for NoC," in 2009 IEEE/ACS International Conference on Computer Systems and Applications, AICCSA 2009, 2009.

[24] R. Pau and N. Manjikian, "Implementation of a configurable router for embedded network-on-chip support in FPGAs," in 2008 Joint IEEE North-East Workshop on Circuits and Systems and TAISA Conference, NEWCAS-TAISA, 2008.

[25] J. Chang, S. Ravi, and A. Raghunathan, "FLEXBAR : A crossbar switching fabric with improved performance and utilization," Proc. Cust. Integr. Circuits Conf., 2002.

[26] R. Dreslinski et al., "Swizzle Switch : A self-arbitrating high-radix crossbar for NoC systems," 2012 IEEE Hot Chips 24 Symp. HCS 2012, 2016.

[27] S. Rao, S. Jeloka, R. Das, D. Blaauw, R. Dreslinski, and T. Mudge, "VIX : Virtual Input Crossbar for Efficient Switch Allocation," Proc. 51st Annu. Des. Autom. Conf., 2014.

[28] B. P. Shrivastava and K. Khare, "Design of Improved Routers for Network on Chip," vol. 4, no. 9, pp. 2975–2980, 2013.

[29] B. P. Shrivastava and K. Khare, "SMART MULTICROSSBAR ROUTER DESIGN IN NOC," Int. J. VLSI Des. Commun. Syst., vol. 4, no. 2, pp. 75–82, 2013.

[30] R. Kamal and J. M. M. Arostegui, "RTL implementation and analysis of fixed priority, round robin, and matrix arbiters for the NoC's routers," in Proceeding - IEEE International Conference on Computing, Communication and Automation, ICCCA 2016, 2017.

[31] M. Daneshtalab, M. Ebrahimi, P. Liljeberg, J. Plosila, and H. Tenhunen, "Memory-efficient on-chip network with adaptive interfaces," IEEE Trans. Comput. Des. Integr. Circuits Syst., 2012.

[32] D. Bafumba-Lokilo, Y. Savaria, and J. P. David, "Generic crossbar network on chip for FPGA MPSoCs," in 2008 Joint IEEE North-East Workshop on Circuits and Systems and TAISA Conference, NEWCAS-TAISA, 2008.

Vivek Tiwari received B.E (Electronics and Communication) in 2007 From BCE Mandideep, Bhopal and M.Tech in 2011 with specialization in VLSI and Embedded System from Bhopal, Currently he is pursuing part-time Ph.D. in Electronics and Communication in MANIT, Bhopal. Network-on-Chip (NoC), Reliability and Modelling and Simulation. He is currently an assistant professor at SIRTS Bhopal.