

Review on Optimized router pipeline Stages used for Network on Chip

Yudhishtir U. Bhangale¹, Prof. Neetesh Raghuvanshi², Prof. Jayant P. Bhoge³

^{1,2}Department of Electronics & Communication Engineering

³Department of Electrical Engineering

^{1,2}RKDF Institute of Science and Technology (RKDFIST), Bhopal (MP), India.

³NMKC COET, Jalgaon, Maharashtra, India

Abstract- As the feature size is continuously decreasing and integration density is increasing, interconnections have become a dominating factor in determining the overall quality of a chip. Due to the limited scalability of system bus, it cannot meet the requirement of current System-on-Chip (SoC) implementations where only a limited number of functional units can be supported. Long global wires also cause many design problems, such as routing congestion, noise coupling, and difficult timing closure. Network-on-Chip (NoC) architectures have been proposed to be an alternative to solve the above problems by using a packet-based communication network. In this paper, the Circuit-Switched (CS) Router was designed and analysed the various parameters such as power, timing and area. The CS router has taken more number of cycles to transfer the data from source to destination. So the pipelining concept was implemented by adding registers in the CS router architecture. The proposed architecture increases the speed of operation and reduces the critical path of the circuit. The router has been implemented using Verilog HDL. The parameters area, power and timing were calculated in 130 nm CMOS technology using Synopsys tool with nominal operating voltage of 1V and packet size is 39 bits. Finally power, area and time of these two routers have been analysed and compared.

Keywords- Network-on-chip, XY routing algorithm, Field Programmable Gate Array(FPGA), VHDL, system-on-chip (SoC), Latency.

I. INTRODUCTION

Recently Networks on Chip (NoC) is playing vital role in development in VLSI. Increasing levels of integration resulted in systems with different types of applications, each having its own I/O traffic characteristics. Since the early days of VLSI, communication within the chip dominated the die area and dictated clock speed and power consumption. Using buses is becoming less desirable, especially with the ever growing complexity of single-die multiprocessor systems. As a consequence, the main feature of NoC is the use of networking technology to establish data exchange within the

chip. 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 [1]. Apart from throughput, NoC platform is scalable and has the potential to keep up with the pace of technology advances. NoC network can be modelled as a graph where in nodes, processing elements and edges are the connective links of the processing elements. Figure 1 Internal architecture of one 3X3 crossbar, it basically includes processing element (PE), router. Each PE is attached to NI which connects the PE to a local router. When a packet was sent from a source PE to a destination PE, the packet is forwarded hop by hop on the network via the decision made by each router. Like in any other network, router is the most important component for the design of communication back bone of a NoC system. In a packet switched network, the functionality of the router is to forward an incoming packet to the destination resource if it is directly connected to it, or to forward the packet to another router connected to it. It is very important that design of a NoC router should be as simple as possible because implementation cost increases with an increase in the design complexity of a router [2].

The remaining section of the paper is organized as follows: In section II describe the Literature overview of Router and related work. In Section III describe Different Routing Algorithm. Section IV describe Switching Methods used in router, Last V section gives overall architecture of Router, at last conclusion of the paper.

II. LITERATURE SURVEY

The author illustrated the impact of repeater insertion on inter-router links with adaptive control and eliminating some of the buffers in the router. The approach saved appreciable amount of power and area without significant degradation in the throughput and latency, though there is still some scope to increase the buffer utilization inside the router [3]. Reference [4] shows work on router for NoC to increase throughput of the network and they introduced the architecture

which shows a significant improvement in throughput at the expense of area and power due to extra crossbar and complex arbitration scheme. The throughput increased up to 94 by the factor of 1.28 [4]. By utilizing the buffer with bidirectional channels indicated significant improvement in system performance, though in this case, each channel controller will have two additional tasks: dynamically configuring the channel direction and to allocate the channel to one of the routers, sharing the channel. Also, there is a 40% area overhead over the typical NoC router architecture due to double crossbar design and control logic [5]. The author developed an algorithm to optimize size of decoupling buffers in network interfaces. The buffer size is proportional to the maximum difference between the number of words produced and the number of words consumed at any point in time. This approach showed significant improvement in power dissipation and silicon area. The buffer size can be further optimized by considering the idle time of buffer. If some buffer is idle at some time instant, it can share the load of neighbouring input channel and thus increase the utilization of existing resources with small control logic [6]. The author proposed the router architecture with Reliability Aware Virtual Channel (RAVC). In this approach, more memory is allocated to the busy channels and less to the idle channels. This dynamic allocation of storage shows 7.1% and 3.1% latency decrease under uniform and transposes traffic patterns respectively at the expense of complex memory control logic, though this solution is latency efficient but not area and power efficient [7].

III. ROUTING ALGORITHM

Routing methods for network decides the path for data transfer to the destination point. The overview of some methods is provided below.

A. XY Routing Algorithm

Routing algorithm is used to route the packets from source PE to destination PE. XY routing algorithm is used to route the packet in proposed router designing. It is a type of kind of distributed deterministic routing algorithms. In 2-D mesh topology NoC, each router can be identified by its coordinate as shown in fig .1. The XY routing algorithm compares the current router address (C_x, C_y) to the destination router address (D_x, D_y) of the packet, stored in the header flit. Flits must be routed to the core port of the router when the (C_x, C_y) address of the current router is equal to the (D_x, D_y) address. If this is not the case, the D_x address is firstly compared to the C_x (horizontal) address. Flits will be routed to the East port when $C_x < D_x$, to West when $C_x > D_x$ and if $C_x = D_x$ the header flit is already horizontally aligned. If this last

condition is true, the D_y (vertical) address is compared to the C_y address. Flits will be routed to South when $C_y < D_y$, to North when $C_y > D_y$. If the chosen port is busy, the header flit as well as all subsequent flits of this packet will be blocked. The routing request for this packet will remain active until a connection is established in some future execution of the procedure in this router [8].

B. Surrounding XY Routing

Surrounding XY routing (S-XY) has three different routing modes. N-XY (Normal XY) mode works just like the basic XY routing. It routes packets first along x-axis and then along y-axis. Routing stays on NXY mode as long as network is not blocked and routing does not meet inactive routers [11]. SH-XY (Surround horizontal XY) mode is used when the routers left or right neighbour is deactivated. Correspondly the third mode SV-XY (Surround vertical XY) is used when the upper or lower neighbour of the router is inactive. The SH-XY mode routes packets to the correct column on the grounds of coordinates of the destination. The algorithm bypasses packets around the inactive routers along the shortest possible path. The situation is a little bit different in the SV-XY mode because the packets are already in the right column. Packets can be routed to left or right. The routers in the SH-XY and SV-XY modes add a small identifier to the packets that tells to other routers that these packets are routed using SH-XY or SV-XY mode [11]. Thus the other routers do not send the packets backwards. Surrounding XY routing is used in a DyNoC. It is a method that supports communication between modules which are dynamically placed on a device.

C. OE Routing Algorithm

OE routing algorithm is a distributed adaptive routing algorithm which is based on odd-even turn model. It has some restrictions, for avoiding and preventing from deadlock appearance [12]. Odd-even turn model facilitates deadlock-free routing in two-dimensional (2D) meshes with no virtual channels. In a two-dimension mesh with dimensions $X*Y$ each node is identified by its coordinate (x, y) . In this model, a column is called even if its x dimension element is even numerical column. Also, a column is called odd if its x dimension element is an odd number. A turn involves a 90-degree change of travelling direction [12]. A turn is a 90-degree turn in the following description. There are eight types of turns, according to the travelling directions of the associated channels. A turn is called an ES turn if it involves a change of direction from East to South. Similarly, we can define the other seven types of turns, namely EN, WS, WN, SE, SW, NE, and NW turns, where E, W, S, and N indicate East, West, South, and North, respectively. As a whole, there

are two conditions [12]. 1. No packet is permitted to do EN turn in each node which is located on an even column. Also, No packet is permitted to do NW turn in each node that is located on an odd column. 2. No packet is permitted to do ES turn in each node that is in an even column. Also, no packet is permitted to do SW turn in each node which is in an odd column.

IV. SWITCHING METHODES

It is an important method that can determine connections between input port and output port. The crossbar switch is used in most of the router design for providing full connectivity [10]. There are two basic data switching methods.

A. Circuit Switching

In circuit switching routing decision is made when path is set up across the sender and receiver. A dedicated path is established between the sender and receiver which are maintained for the entire duration of transmission [9]. Moreover, links remain occupied even with the absence of data transmission. There is no delay in data flow because of the dedicated path. The major drawback of circuit switching is its limiting scalability.

B. Packet Switching

In packet switching, data is broken up into packets. Individual packets take different routes to reach the destination. Each packet includes a header with source, destination and intermediate node address information. The performance can be increased due to segmentation of data. There are three types of packet switching: wormhole (WH), store and forward (SAF) and virtual cut through (VCT) switching [9]. The need to buffer complete packet within a router can make it difficult to construct low area, compact and fast routers. In wormhole switching message packets are also pipelined through the network [10]. A message packet is broken up into flits that the flit is the unit of message flow control. Therefore, input and output buffers at a router are typically large enough to store a few flits. As we said, in this switching, each packet is divided into equal smaller sections named as flit [1]. Flits are concurrently transferred in the network. In SAF switching router should have sufficient buffer space to store the entire packet. Router in every hop must wait to receive the entire packet before forwarding header flit to the neighbouring router [9]. So, the buffer size should be large enough to store entire packet which suffers it from larger latency compared with other technique.

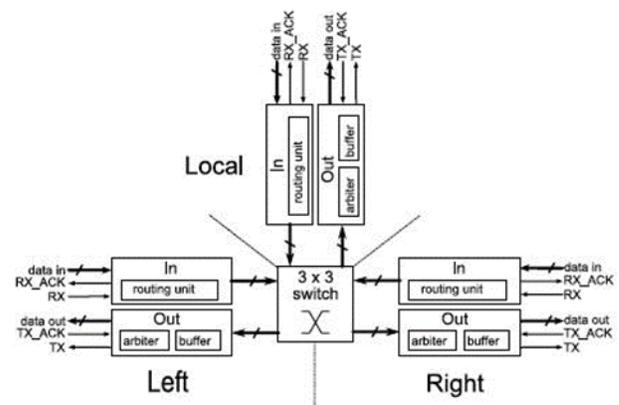


Figure 1. Internal architecture of one 3X3 crossbar

V. ROUTER ARCHITECTURE

The increasing logic and memory capacity of FPGA chips enables the implementation of large systems such as application-specific single-chip multiprocessors. To effectively support the interconnection of many processor elements and other components that are implemented in the fully programmable logic of an FPGA, the configurable router that is described in this paper is intended to be included as a dedicated hardware component that is distributed in sufficient numbers throughout a large FPGA, similar to how embedded memory blocks or multipliers are distributed throughout an FPGA. Given the expected usage for interconnection of distributed processor/memory elements, the router employs a packet format that is suitable to carry address and data together, along with source/destination and other information. Configurable routers that are connected to form the supported network topologies utilize packet switching to convey the address/data information in each packet from source to destination. The configurable router consists of control logic and 5 bidirectional ports: Local, North, East, South, and West. The local bidirectional port establishes connections with an associated node element, and the remaining four bidirectional ports provide support for different network topologies. Switching is performed by two 3X3 crossbars instead of one full 5X5 crossbar in order to reduce area and power consumption [6]. The configurable router is packet-switched and employs handshaking for flow control with buffering at the outputs. Finally, the control logic in each crossbar orchestrates all of the switching activities and channel arbitration based on the selected routing algorithm and the traffic conditions during operation.

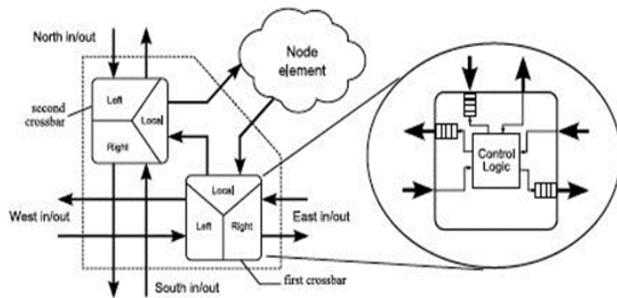


Figure 2. Dual-crossbar arrangement and high-level architecture

- 1) **Dual-crossbar Architecture:** Two smaller crossbars provide the same functionality as one large crossbar, requiring less area and power with some increase in router latency. As shown in Figure 1, each 3x3 crossbar contains three bidirectional connections: local, left, and right. The local connections of each crossbar are associated with the Local port of the overall router, and the left/right connections of each crossbar are associated with the North, East, South, and West ports that are used to create the desired network topology. Outgoing packets from the node element that is attached to the overall router pass through the local connection of the first crossbar. Based on the selected network topology, the packet is switched directly to the East/West router outputs or to the second crossbar for the North/South router outputs. Similarly, incoming packets that arrive through the North/South ports are switched directly to the attached node element, but packets arriving through the East/West ports must first be switched to the second crossbar in order to reach the attached node element. Figure 2 illustrates the five network topologies supported by the dual-crossbar arrangement: uni- and bi-directional ring, uni- and bi-directional octagon, and mesh. The mesh and ring topologies are common NoC topologies. The octagon network configuration [7] is similar to a ring but enhanced with cross connections between opposite nodes. Table I summarizes the left/right connections used for the two crossbars to support these topologies. The rings use only the East/West router ports, hence the first crossbar is for routing and the second crossbar only serves as the local connection to the associated node element. The octagons use the first crossbar for routing, just as for a ring, but also use the second crossbar for cross-link routing in addition to the local connection. Finally, the internal nodes in a mesh use all of the remaining crossbar connections for routing (corner and side nodes need fewer connections).
- 2) **Buffering:** Each port of the proposed configurable router contains a buffered output channel. Output buffering requires significantly less area and control logic than a configuration with input and output buffers [8]. Packets are buffered until the downstream port connected to the output is ready to accept the packets. Switching is performed before the buffering, hence inputs are isolated from output congestion up to the point that a target output has no remaining buffer capacity.
- 3) **Control Logic:** Figure 1 illustrates the internal architecture of each 3x3 crossbar. The control logic consists of a routing unit at each input channel and an arbitration unit at each output channel. The three routing units and three arbitration units work together to forward packets from an input port to a desired output port. With two crossbars, simultaneous transfers may be performed on all ports of the overall router. The routing unit at each input channel uses the routing algorithm for a particular network topology to interpret the destination node ID from a received packet and then arbitrate for the appropriate output channel accordingly. Each routing unit implements direct hardware support for the routing algorithms appropriate for the five topologies. Deterministic routing is used for routing in all topologies. The mesh uses XY routing where a packet is first routed along the horizontal direction of the mesh until the target column is reached, then routed vertically to its destination. XY routing naturally suits the dual-crossbar arrangement; the first crossbar handles horizontal routing and the second router handles vertical routing. For rings, the first crossbar handles routing in one or both directions. The octagons initially route packets along the ring using the first crossbar before using a cross link with the second crossbar, if necessary. The configurable router uses a handshake for flow control. In Figure 3 shows basic noc architecture with mesh topology, a pair of handshaking signals is associated with the data bus for each port. The request signal is asserted by the packet source, and the assertion of the acknowledge signal by the receiver indicates acceptance of the packet.
- 4) **Round Robin Arbitration:** If multiple packets are arrived at the input channel for same destination then these are scheduled with a round robin arbiter. It will check the priority of the channels decided by arbiter(2). Depending on that channel with first priority will be served first, till other channel will have to wait until it has higher priority. Now the channel served recently will have least priority.

Depending upon the control logic arbiter generates select lines for multiplexer based crossbar and read or write signal for FIFO buffers.

VI. CONCLUSION

A different type of XY routing algorithm is used in different network condition. The selection of XY routing algorithm it totally depends upon the application and the traffic of packet in the network. As simplicity in implementation is important in all architecture so an XY routing algorithm widely used. For fewer collisions Intermittent Routing Algorithm is preferred. The XY routing algorithm achieves better balance in load distribution as well as provide deadlock-free and livelock-free facility. We can't compromised with accuracy of received data, we go for the fault-tolerant routing like XYX. If application is focused on network resources utilization the Adaptive XY routing algorithm is best choice.

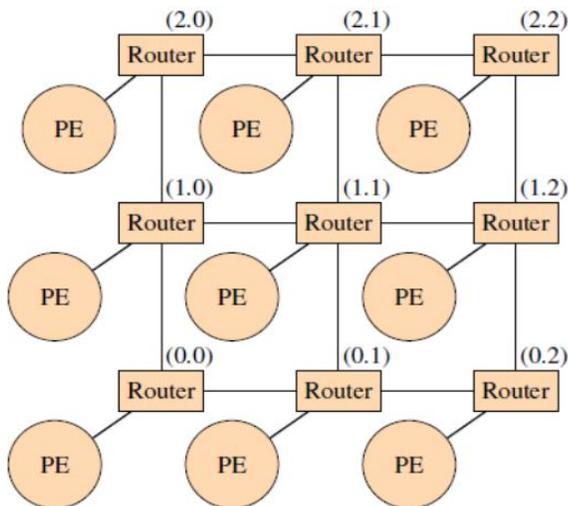


Figure 3. Basic NoC architecture with mesh topology

XY routing algorithm is one of the simplest and most commonly used NoC routing algorithms. It is deterministic, static and deadlock free routing algorithm. We observed that most of the XY routing algorithms are implemented on 2D mesh topology to increase throughput and reduce latency. But most of them is facing problem of traffic congestion in the centre. There are number of topologies available but the torus topology has gained lots of consideration by designer due to their simplicity. So we propose new design method using a XY routing algorithm for 2D torus topology to solve above problem. We will use xilinx simulator for simulation and performance analysis. We expect the result with reduce the average latency per packet and increase average throughput. Finally we can say that choice of XY routing algorithm is

totally depends upon environmental condition of NOC architecture.

REFERENCES

- [1] Bouraoui Chemli and Abdelkrim Zitouni, " Design and Evaluation of Optimized router pipeline stages for Network on Chip, IEEE IPAS16: International Image Processing Applications And Systems Conference 2016, IEEE, pages 1-5, 2016.
- [2] Maha Beheiry, Hassan Mostafa2, Yehea Ismail and Ahmed M. Soliman, "3D-NOCET: A Tool for Implementing 3D-NoCs based on the Direct-Elevator Algorithm, 18th Int'l Symposium on Quality Electronic Design, pages 144-149, 2017.
- [3] Atef Dorai, Virginie Fresse, Abdellatif Mtibaa, El Bay Bourennane, "Backoff Hardware Architecture for Inter FPGA Traffic Management, 2017 International Conference on Advanced Systems and Electric Technologies (ICASET),IEEE, pages 104-019, 2007.
- [4] Adewale Adetomi, Godwin Enemali, and Tughrul Arslan, "Clock Buffers, Nets, and Trees for On Chip Communication: A Novel Network Access Technique in FPGAs, 2017 IEEE International Parallel and Distributed Processing Symposium Workshops, IEEE computer Society, IEEE, Pages 219-222, 2017.
- [5] Jems Rottekowski, Dinna Goheringer, "Data Stream Processing in Net-work on Chip, 2017 IEEE Computer Society Annual Symposium on VLSI, IEEE. Pages 633-638, 2017.
- [6] Nachiket Kapre, "Implementing FPGA overlay NoCs using the Xilinx Ultra Scale memory cascades, IEEE 25th Annual International Sympo-sium on Field-Programmable Custom Computing Machines, IEEE, pages 40-47, 2017.
- [7] Nachiket Kapre, "one bit serial noc for FPGA, IEEE 25th Annual International Symposium on Field-Programmable Custom Computing Machines, IEEE, pages 32-39, 2017.
- [8] Khaled E. Ahmed, Mohamed R. Rizk and Mohammed M. Farag, " Overloaded CDMA Crossbar for Network-On-Chip, IEEE Transactions On Very Large Scale Integration (Vlsi) Systems, IEEE, pages 1-14, 2017.
- [9] Jinglei Huang, Xiaodong Xu and Lan Yao, Song Chen, " Reconfigurable Topology Synthesis for Application-

- Specific NoC on Partially Dynamically Reconfigurable FPGAs, SLIP17 June 17, 2017, Austin, TX, USA, IEEE, 2017.
- [10] Hadi Mardani Kamali and Shahin Hessabi, "AdapNoC: A Fast and Flexible FPGA-based NoC Simulator, IEEE, 2016.
- [11] Huan-Yuan Chen, Shu-Hao Hsu, Wen-Jyi Hwang and Chau-Jern Cheng, "An Efficient FPGA-Based Parallel Phase Unwrapping Hardware Architecture, IEEE Trans. Computational Imaging, pages 1-13, 2016.
- [12] Kavyashree G S, Saroja S. Bhusare and Sunita Shirahatti, "Architectural Based Congestion Management for Network on Chip implemented on FPGA, IEEE, 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), IEEE, pages 258-263, 2016.
- [13] Chethan Kumar H B, Shubham Agarwal and Nachiket Kapre, "Deflection Routing for Multi-Level FPGA Overlay NoCs, IEEE, 2016.
- [14] Hung K. Nguyen and Xuan-Tu Tran, "Design and Implementation of a Hybrid Switching Router for the Reconfigurable Network-on-Chip, 2016 International Conference on Advanced Technologies for Communications (ATC), IEEE, pages 328-333, 2016.
- [15] Yu-Kun Song, Qing-Song Qian and Duo-Li Zhang, "Design and Implementation of Dual-Port Network on Chip Based on Multi-core System, IEEE, 2016.
- [16] Mayank Kumar, Kishore kumar, Sanjiv kumar gupta and Yogendera Kumar, "FPGA Based Design of Area Efficient Router Architecture for Network on Chip (NoC), International Conference on Computing, Communication and Automation (ICCCA2016), IEEE, 1600-1605, 2016.
- [17] Serif Yesil, Suleyman Tosun and Ozcan Ozturk, "FPGA Implementation of a Fault-Tolerant Application-Specific NoC Design, 11th International conference on design and technology for Nano-scale era(DTIS), IEEE, 2016.
- [18] Geethu Jayan and Pavitha P. P., "FPGA Implementation of an Efficient Router Architecture Based on DMC, 2016 International Conference on Emerging Technological Trends [ICETT], IEEE, 2016.
- [19] Sukhbani Kaur viridi, Shushant Shekhar, gaurav Varma Shekhar Mahesh-wari, "Implementation of crossbar switch for NOC on FPGA, International Conference on Computing for Sustainable Global Development (INDIA-Com), IEEE, pages 2087-2091, 2016.
- [20] Siddhartha, Nachiket Kapre, "eBSP: Managing NoC traffic for BSP workloads on the 16-core Adapteva Epiphany-III Processor, 2017 Design, Automation and Test in Europe (DATE), IEEE, pages 73-78, 2017.
- [21] Khaled E. Ahmed, Mohamed R. Rizk and Mohammed M. Farag, "Overloaded CDMA Interconnect for Network-on-Chip (OCNoC), 2016 IEEE.
- [22] Taimour Wehbe and Xiaofang Wang, "Secure and Dependable NoC-Connected Systems on an FPGA Chip, IEEE Transactions On Reliability, IEEE, Pages 1-12, 2016.
- [23] Otavio A. de Lima Jr., Otavio A. de Lima Jr. and Frederic Rousseau, "A survey of NoC evaluation platforms on FPGAs, IEEE, pages 1-6, 2016.
- [24] Helio Fernandes da Cunha Junior, Bruno de Abreu Silva and Van-derlei Bonato, "Parameterizable Ethernet Network-on-Chip Architecture on FPGA, Euromicro Conference on Digital System Design, conference publishing services, IEEE, pages 263-266, 2015.
- [25] Yao Chen, Swathi T. Gurumani, Yun Liang, Guofeng Li, Donghui Guo, Kyle Rupnow and Deming Chen, "FCUDA-NoC: A Scalable and Efficient Network-on-Chip Implementation for the CUDA-to-FPGA Flow, IEEE Transactions On Very Large Scale Integration (Vlsi) Systems, IEEE, pages 1-14, 2015.
- [26] Amit Bhanwala, Mayank Kumar and Mayank Kumar, "FPGA based Design of Low Power Reconfigurable Router for Network on Chip (NoC), International Conference on Computing, Communication and Automation (ICCCA2015), IEEE, pages 1320-1326, 2015.
- [27] Manel Langar, Riad Bourguiba, Jaouhar Mouine, "Design and Implementation of Enhanced on chip Router, 12th International Multi Conference on Systems, Signals & Devices, IEEE, pages 1-4, 2015.
- [28] Zalak Dave, Shivank Dhote, Jonathan Joshi, Abhay Tambe and Sachin Gengaje, "Network on Chip Based Multi-function Image Processing System using FPGA, 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI),

IEEE, Pages 488-492, 2015.