[1]Liyaqat Nazir,
[2]Mohd Asifuddola,
[3]Mumtaz Ahmed,
[4]Mir Mohsina Rahman

Implementation of Dimension Order Routing Algorithm on Reconfigurable Hardware for NoC applications

Journal of Electrical Systems

Abstract: - Inter-processing element communication encounters significant hurdles arising from constraints in power consumption, physical space, and data transfer speed. Storing data packets temporarily during communication uses a lot of the chip's power. Many contemporary network-on-chip (NoC) architectures are heavily resource-intensive due to the extensive use of router buffers. Eliminating these buffers and virtual channels can streamline router design and reduce power consumption. The routing and arbitration strategies implemented within a NoC router are pivotal for optimizing the overall performance of the NoC mesh. The routing algorithm plays a critical role in NoC systems, as it is responsible for balancing traffic load across network channels, even in scenarios of asymmetric traffic distribution. This paper introduces a model for X-Y routing algorithms, specifically tailored for implementation on FPGA platforms utilizing a flexible state diagram. The distinctive contribution of this research lies in its synthesis and optimized realization of the X-Y routing algorithm on FPGAs, providing Network-on-Chip (NoC) designers with a robust framework for developing efficient routers tailored to their FPGA architectures.

Keywords: Network-on-chip, Dimension order, Buffer.

I. INTRODUCTION

Network-on-Chip (NoC) communication architecture has appeared with promising solutions in communication in the field of integrated circuits. The NoC approach has provided an approved data transfer by delivering improved bandwidth and superior performance for communication between different components on a chip. The paradigm has also proffered scalable design, making it specifically effective when implemented on reconfigurable platforms, yielding for flexibility and adaptability in various applications.[1]. Network-on-chip establishes an effectual communication paradigm that transforms system-on-chip (SoC) design, having significant improvements in efficiency and overall performance [2] as illustrated in figure 1. Several processing elements within a System on a Chip (SoC) are interconnected through Network-on-Chip (NoC) routing nodes. These routing nodes are spatially organized in various configurations, for example mesh, linear, toroidal, and both 2D and 3D topologies, as depicted in Figure 2. To ensure optimal performance, routers must ensure the delivery of high bandwidth and low latency. [3]. The performance of a Network on Chip (NoC) is primarily reviewed through its throughput, a crucial metric which is a function of various factors. While the performance parameters of the NoC are normally perceived by its throughput, which is characterized by various parameters such as routing node throughput, the traffic load within the network, and the network topology [4]. The throughput of the routing node is controlled by the data path unit's most significant critical path in the node and the efficiency of control path units [5-8]. The control paths of on-chip communication routing fabric are chiefly constituted of arbitration and allocating units [9]. Allocators perform a crucial role in efficiently assigning Virtual Channels (VC) and expertly matching resources in each cycle. [10-12]. The data paths of the on-chip router incorporate buffering structures, VCs, and switching fabric. When the flits at the input port arrive for transmission, conflict for access to the fabric arises, with flits competing at both the input and output stages. The router unit enforces particular handshake signals to facilitate the transfer of data or flits. Subsequently, a VC allocating unit manages the allocation among the incoming flits, permitting only one flit competing at the input port to be routed to the designated output port.[13]. To mitigate the issue of blocking, the competing flits are temporarily stored in the VCs or buffering space of the router unit, enabling their processing during the subsequent appropriate clock cycles. [14]. Eliminating input buffers can streamline the architecture by removing the necessity for VCs. However, this approach may lead to increased head-of-line blocking, which can

[1]Electronics and Communication Engineering, Islamic University of Science & Technology Awantipora, India

[2]Department of Electronics Engineering, Z.H.C.E.T, Aligarh Muslim University, Aligarh, India

[3]Department of Computer Engineering, Jamia Millia Islamia, New Delhi, India, (Corresponding Author)

Email: mahmed1@jmi.ac.in

[4]National Institute of Technology, Srinagar J&K, India

adversely affect overall performance. It is essential to carefully consider the balance between system simplicity and operational efficiency. The routing scheme implemented on the on-chip networks affects the performance in many ways by harmonizing the load through several network channels even in the presence of the irregular traffic patterns present on the link pathways of the mesh architecture. The more balanced load is responsible of providing a close to ideal throughput of the network and a poor job balancing router results in the lesser ideal throughput. A non-uniform traffic pattern induces a large load imbalances and with the help of such routing algorithm a sub optimal throughput is obtained. A well-structured algorithm efficiently reduces path lengths, thereby decreasing the number of hops and the overall latency associated with message packets. It has been demonstrated that source routing is particularly advantageous for small Network-on-Chip (NoC) configurations, especially in the development of application-specific multi-core systems. [17]. Routing algorithms can be categorized into several types, including centralized, source, distributed, and multiphase routing. In the case of source routing, each packet contains comprehensive information regarding the pathway from the source to the destination. This approach facilitates simplified router design and minimizes delay. However, as the network size expands, the requirement to store path information can lead to considerable overhead and present a significant drawback. Consequently, various sets of routing procedures have been proposed and implemented in diverse ways to accommodate the throughput demands of the system. Additionally, a number of routing algorithms that do not utilize virtual channels have been developed specifically for two-dimensional networks. The dimension-order routing algorithm for 2D meshes routes packets in a manner that avoids deadlock in the channel dependency graph of the 2D network. However, it lacks adaptiveness. Glass et al. proposed a comprehensive technique called as the turn model, which helps in the realization of incompletely adaptive wormhole routing protocols exclusive of the requirement of virtual channels.[18], [19]. Moreover, a large number of logic elements are required in order to designing NoC routing blocks. This work, therefore, focuses on the optimised implementation and evaluation of Dimension order X-Y routing algorithm on reconfigurable platforms. The paper mainly highlights the efficient exploitation the FPGA resources in order to carry out the optimised implementation of dimension order routing logic function on FPGA Target device. The paper thus shows the optimized mapping, implementation and synthesis of X-Y Routing Algorithm on Virtex 5 FPGA platforms.

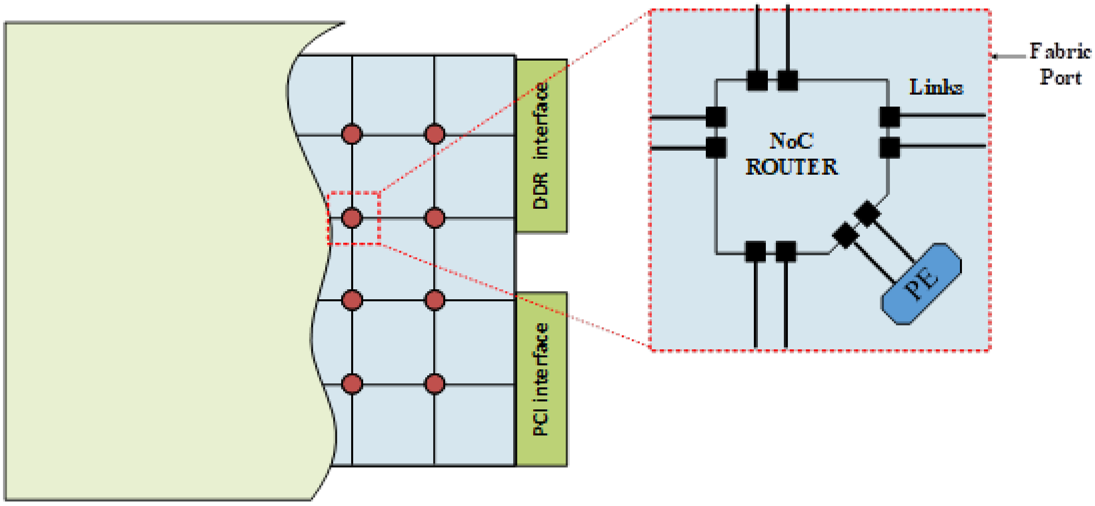

Figure1. Block diagram of NoC based system

The entire research article is organized as follows. Section 1 gives the introduction. Section 2 defines the routing topologies in NoC based systems. Section 3 given a detailed literature review hence highlights the importance of the topic/area. Section 4 mentions the details of algorithmic construction and operational details. Section 5 talks about synthesis and implementation and discusses the results obtained. The conclusion and future scope have been given in Section 6.

II. Routing Algorithm

As stated in the earlier section, the process of routing data flits is critical in a NoC fabric which involves directing data packets through the network from their origin to their intended destination. This process ensures that information travels efficiently and reliably throughout the communication fabric. At its core, a routing algorithm is composed of two essential components: the routing function, which determines the overall path that packets should take, and the selection function, which chooses the most suitable route among multiple options. Together, these components work in harmony to optimize data transmission and enhance network performance [20, 21]. Initially, the routing function determines the set of permissible output channels that can facilitate the forwarding of the packet to its intended destination. Subsequently, the selection function is employed to identify one output channel from the set of admissible output channels generated by the routing function. In routers utilizing a deterministic routing algorithm, the selection block is rendered unnecessary, as the routing function yields only one output port. Conversely, in routers that implement an oblivious routing algorithm, the selection process is based exclusively on the information contained within the header flit. Furthermore, routers that adopt more sophisticated routing algorithms leverage network status information—such as buffer occupancy and link utilization —within their selection function.

III. Literature review

The X-Y routing algorithm, also known as dimension-ordered routing algorithm, is a deterministic routing algorithm considered by its ease. In a 2D mesh NoC, a dimension order routing algorithm causes routing of packets first along the X-dimension until they reach the destination column and then along the Y-dimension so that the destination row is reached. This simplistic approach makes it easy to implement in hardware, resulting in reduced router complexity and lower power consumption. This aspect has also highlighted its suitability for mesh type NoC architectures. The implementation of X-Y routing algorithms for Network-on-Chip (NoC) architectures on reconfigurable platforms has therefore acquired significant attention in recent research, given the increasing complexity and performance demands of modern solutions to the challenges in embedded systems. This literature review highlights key findings from various studies, emphasizing the relevance of X-Y routing in enhancing communication efficiency, adaptability, and fault tolerance in NoC systems. This has been confirmed by the authors in [22] where in the work highlights the scope of X-Y routing in NoC implementations due to their simplicity and efficiency in managing on-chip communication. The authors tried to mention the suitability of dimension order algorithm for grid-based architectures and emphasized its straightforward approach which aids in reducing congestion and optimizing throughput, which is essential in many-core embedded systems that require reliable communication pathways [22]. The integration of X-Y routing algorithms in reconfigurable tools, like Xilinx Versal Adaptive Compute Acceleration Platform (ACAP), enhances communication efficiency between processing units. This was presented by the authors in the work [23] who also concluded that the programmable nature of ACAP allows for dynamic adaptation of routing algorithms to optimize data flow and reduce latency. This adaptability is crucial as it enables the separation of data movement within the computing architecture, facilitating the deployment of advanced routing strategies that can respond to changing communication patterns [23]. Fault tolerance is a significant concern in NoC systems, especially in reconfigurable platforms where routing requirements may change dynamically. The authors in [24] suggested a Fault-Tolerant Deflection Routing (FTDR) scheme, based on reinforcement learning that complements the X-Y routing algorithm by allowing for adaptive reconfiguration of routing tables in response to network faults. Additionally, the integration of the Minimal and Defect-Resilient (MD) routing algorithm enhances X-Y routing by providing fault-tolerant capabilities through adaptive routing along the shortest paths as suggested by the authors in [25, 26, 27]. These algorithms highlight the necessity of incorporating fault tolerance mechanisms into X-Y routing to ensure reliable communication under adverse conditions.

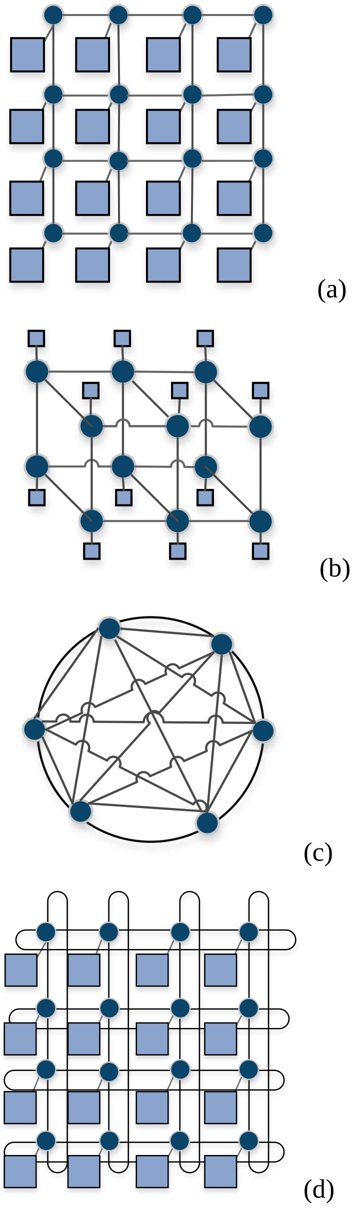

Figure 1. Block level illustration of NoC Topologies (a) 2-D mesh (b) 3-D Mesh (c) Ring type NoC (d) Torrous NoC.

Furthermore, traffic congestion in NoC systems can severely impact performance, necessitating adaptive routing strategies. The recently proposed Traffic- and Throttling-Awareness Routing (TTAR) algorithm by the authors in [28] enhances X-Y routing by managing network traffic dynamically, ensuring efficient packet routing even under varying conditions. Furthermore, coarse and fine-grained look-ahead algorithms provide adaptive capabilities that minimize congestion by allowing routers to make informed decisions based on real-time traffic conditions [29]. Such adaptations are essential for maintaining high performance in reconfigurable NoC systems. The introduction of the bidirectional channel network-on-chip (BiNoC) architecture, which allows for dynamic self-reconfiguration of communication channels, significantly enhances the flexibility of X-Y routing decisions as proposed by the authors in [30]. This has resulted in an improved bandwidth utilization and has reduced latency, making it a suitable framework for implementing X-Y routing algorithms. Additionally, the routing algorithms can be explored for partially connected 3D-NoCs aligned with X-Y routing principles, with a special focus on fault tolerance and adaptability has also been proposed by the authors in [31]. Moreover, the authors in [32] propose an enhanced version of the X-Y routing algorithm by introducing a single-side buffer and a modified scheduling mechanism in the routers. This improvement has reduced the communication hops, optimizing latency and area on FPGAs. The modified X-Y routing has eventually reduced the average packet latency and increased throughput in an 8x8 mesh topology. The concept of dynamic X-Y routing has been further explored by the authors in large-scale NoCs [33]. They developed an enhanced Long Edge First (LEF) routing, which uses X-Y routing as a base but can adapt to traffic patterns dynamically to reduce congestion in large networks. This technique showed up to 36.7% higher throughput compared to traditional X-Y routing under non-uniform traffic conditions. The above-mentioned advancements underscore the importance of developing routing algorithms that can operate efficiently in complex and evolving network topologies. Besides recent research accentuates the need for co-optimization of, communication energy, performance and reliability within NoC architectures. A proposed mapping approach has been reported by the authors, that evaluates communication energy, latency, and reliability can enhance the effectiveness of X-Y routing [34]. The above comprehensive approach is significant for addressing the diverse requirements of modern applications, thus ensuring that the NoC systems remain efficient and robust. Despite the advancements in implementing X-Y routing algorithms for NoC on reconfigurable platforms, with advancements in hardware, there is a continuous scope in terms of area/power-efficient implementation of the algorithm. Hence the efficient implementation of X-Y routing algorithms for NoC on reconfigurable platforms can provide substantial improvements in the performance if the simplicity in the operation is combined with careful mapping, adaptability, and fault tolerance.

IV. Operational details

The dimension order routing algorithm operates within a 2D mesh network by leveraging the coordinates of nodes to direct data packets toward their destinations. Each node is identified by a unique (W(or X), Z(or Y)) coordinate, and when a packet is sent, it contains the destination coordinates (W_d, Z_d). The routing process begins with the packet moving in the W-direction until it reaches the destination's W-coordinate (W_d), making left or right turns as necessary. Once aligned on the W-axis, the packet then shifts to the Z-direction, progressing up or down to reach the final destination's Z-coordinate (Z_d). This straightforward approach ensures that the routing is deterministic and typically results in minimal path lengths. While the dimension order routing algorithm is simple and efficient, it can lead to congestion when multiple packets target the same coordinates, as it does not adapt dynamically to network conditions. Special modifications can be further added to perform efficiently in the dynamic load environment as discussed in the previous section. The routing algorithms can operated in a unicast and multicast mode or environment. However, this paper focuses on an algorithm dedicated to the unicast mode of communication, which involves each source transmitting its message to a single specified end point.

A. Description of the turn model

The dimension order turn model in a 2D mesh network focuses on evaluating the directional changes packets can make as they traverse the network, particularly examining the cycles formed by these turns. In this context, a 90-degree change in the direction of travel is classified as a turn. [35], A total of eight distinct turns can be allowed in a 2D mesh using the principal directions: east (E), west (W), north (N), and south (S). These turns represent the various directional changes possible within a grid-based layout. These turns comprise two distinct abstract cycles. By limiting the lowest number of turns—specifically, one from every cycle—one can devise routing algorithms that are highly adaptive and deadlock-free, without necessitating the utilization of virtual channels (VCs)[36]. When designing an algorithm to prevent deadlocks, it’s crucial to avoid specific cycles that contribute to these issues, resulting in a partially adaptive routing strategy. The unicast routing methods developed under this framework maintain deadlock-freedom by ensuring that the six permissible turns do not create cyclic paths. However, simply prohibiting any two turns isn't sufficient to ensure deadlock-free operation; the selection of these prohibited turns must be strategic. Each selected turn should effectively disrupt potential cycles, thereby ensuring comprehensive avoidance of deadlock scenarios.[37].

B. Logic implementation of the algorithm.

The paper presents a novel idea of efficiently realizing an arbitrary routing algorithm implemented from a state diagram as shown in figure 4 .This state diagram demonstrates the node decision diagram of the proposed dimension order turn procedure, which ensures the smooth routing in a two-dimensional (2D) mesh without necessitating the need of virtual channels. A 2D mesh can be assumed to have $p\times\ q$ nodes, whereas p (respectively., q) is the radix of dimension x (respectively., y) of the mesh fabric. Each node d is assumed to have an address d ;(dp, dq), where $dp\ \in\{0,1,2,...,p-1\}$ and $dq\ \in\{0,1,2,...,q-1\}$. Two nodes d: (dp, dq) and e : (ep, eq) are said to be the neighbors in dimension p (resp., q). if and only if $|dp-ep|$= 1(one hop) and $d_q = e_q$ (resp.,

$|dq - eq|$= 1 (one hop) and $dq = eq$). In a two-dimensional NoC topology, a node W is recognized by a two-element vector ($W_0; W_1$), and similarly, node Z is recognized by a two-element vector ($Z_0; Z_1$). Where as to the coordinates of the dimension 0 and the dimension 1 of W, and Z directions are denoted by W_0 , W_1 and Z_0, Z_1 respectively. To enhance the clarity of the presentation, the sides of the two-dimensional mesh fabric are designated as E(East), W (West), S(South), and N(North), as depicted in Figure 3. Nodes that possess identical coordinates in dimension 0 are classified as forming a column, while nodes sharing the same coordinates in dimension 1 are categorized as constituting a row. The row channels relate to connections along dimension 0; specifically, a row channel facilitates the linking of two adjacent nodes within the same row. In contrast, column channels pertain to connections along dimension 1. The SN (South to North) channel notation denotes a column channel when its directionality is from South to North, and as an NS (North to South) channel when it flows from North to South. When a channel establishes a link between any two arbitrary nodes, designated as Node A and Node B, Node A is referred to as the tail node, whereas Node B is termed the head node of the channel.

A turn operation in routing consists of rotation along a column channel and a row channel, where the tail node of one channel connects to the head node of the other. The node where these channels intersect is termed the turning node. Essentially, this configuration facilitates a 90-degree change in travel direction. In this context, unless stated otherwise, we will interpret a turn as a 90-degree turn. In the realm of 2D mesh topologies employing dimension-order X-Y routing, turns can be categorized based on the directional changes of the associated channels.

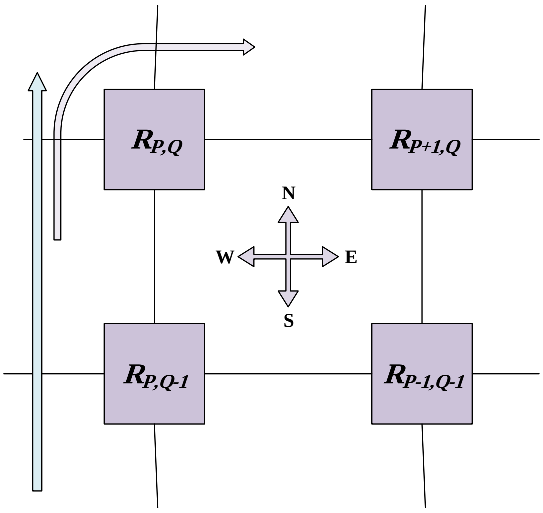

Figure 3. Block level representation of Turn Node

An East-to-South transition is termed an ES turn. The full taxonomy of turns includes EN (East to North), WS (West to South), WN (West to North), SE (South to East), SW (South to West), NE (North to East), and NW (North to West). Here, the notations E, W, S, and N have their usual meaning. A flit is deemed to have executed a turn at node X (or W) when it completes the transition through the turn, with X functioning as the turning node for that specific routing maneuver.

V. Synthesis and implementation

A. Methodology.

The design process commenced with a high-level architecture representation, which was then translated into VHDL. The synthesis was carried out using Xilinx Vivado, focusing on optimizing for area and speed. The synthesis parameters included constraints that ensured the design met the required timing specifications. Following the synthesis, the place-and-route phase was conducted, resulting in the efficient allocation of FPGA resources. This work is focused on the implementation for the 5th generation Virtex FPGA family from Xilinx, specifically utilizing the VLX series, which is well-suited for wide-ranging logic applications. The proposed algorithm is designed to operate within a 3×3 2D Mesh network architecture. Key performance metrics evaluated include timing, utilization of the resources, and power dissipation. Resource utilization is assessed based on the specific on-chip FPGA components employed. Timing considerations encompass the design's clock speed, constrained by factors such as input/output register setup times, clock-to-output delays of flip-flops, propagation delays along critical paths, and skew between departure and catch registers. A thorough timing analysis is conducted, incorporating suitable timing constraints to ensure optimal performance. The average power dissipation in the design is primarily attributed to dynamic power, with static power dissipation being negligible in comparison. To ensure a valid comparison across all implemented topologies, the test bench utilizes a consistent clock period and uniform input statistics. Constraints regarding both the clock period and offset have been carefully established to achieve total timing closure. The processes of synthesis, followed by implementation, and PAR mapping were executed using Xilinx Electronics Design Automation Tools [38]. Power metrics were acquired through the use of the XPower Analyzer, a sophisticated tool designed to accurately assess power consumption in various operational contexts. Concurrently, the simulator database provided essential information regarding the clock period and operating frequency of the final implementation, facilitating a comprehensive evaluation of the system's performance and efficiency. The Post-Place and Route (P&R) data is suitably provided, NCD file was generated for a significant traffic load. Using the Physical Constraints (PCF) file, the static power (leakage power when no activity occurs) and dynamic power (power due to switching activity), VCD (Value Change Dump) were created, accordingly dynamic power was estimated.

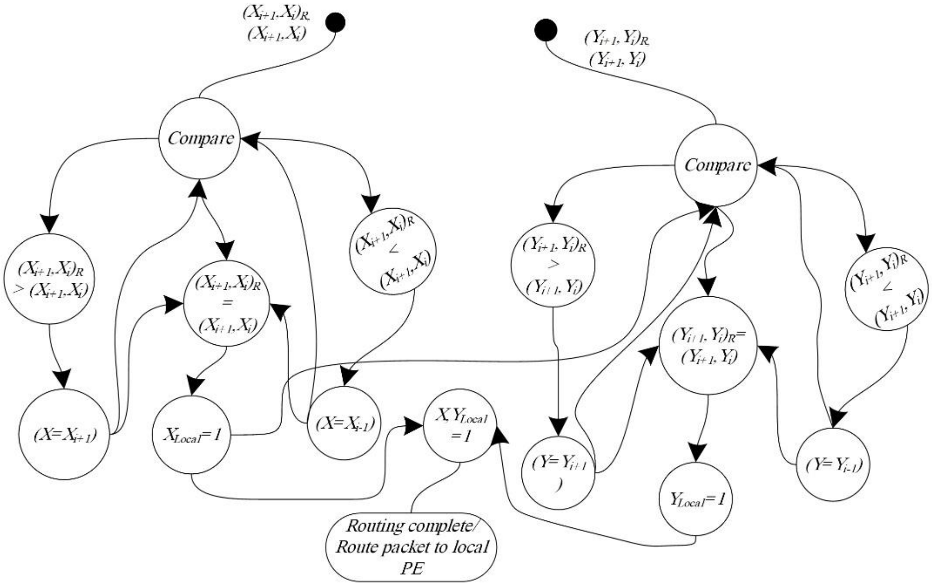
$(X_{i+1}.X_i)_R$
$(X_{i+1}.X_i)$
$(Y_{i+1}.Y_i)_R$
$(Y_{i+1}.Y_i)$
Compare
Compare
$(X_{i+1}.X_i)_R > (X_{i+1}.X_i)$
$(X_{i+1}.X_i)_R = (X_{i+1}.X_i)$
$(X_{i+1}.X_i)_R \angle (X_{i+1}.X_i)$
$(Y_{i+1}.Y_i)_R > (Y_{i+1}.Y_i)$
$(Y_{i+1}.Y_i)_R = (Y_{i+1}.Y_i)$
$(Y_{i+1}.Y_i)_R \angle (Y_{i+1}.Y_i)$
$(X=X_{i+1})$
$X_{Local}=1$
$(X=X_{i-1})$
$X.Y_{Local} = 1$
$(Y=Y_{i+1})$
$Y_{Local}=1$
$(Y=Y_{i-1})$
Routing complete/ Route packet to local PE

Figure 4. State decision representation of a Dimention order X-Y routing algorithm.

B. *Experimental results.*

Resource utilization refers specifically to the utilization metrics of the FPGA resources. Table 1 presents a detailed breakdown of the resource consumption associated with the implementation of the Dimension Order X-Y routing algorithm on the targeted FPGA platform. Table 2 mentions the highest attainable clock rate post-implementation for the implemented design.

From Table 2, we can see that the dimension order routing algorithm can operate at highest frequency of 551.542 MHz. Ultimately the static and the dynamic power dissipation for for the dimension order routing algorithm is take into consideration. The static power dissipation in a reconfigurable hardware is comprised of both device-level and design-level components. While the latter contributes only a small fraction to overall dynamic power dissipation, it remains an important factor to consider. Dynamic power dissipation is influenced by several parameters, namely the square of the supply voltage (V^2), clock frequency (f_{clk}), load capacitance (C_L), switching activity (α), and the number of elements integrated into the design. Mathematically it can be given as:

$$P = N.C_L V^2 f_{clk} \qquad (1)$$

The analysis was performed under the assumption of a fixed supply voltage and at the maximum operating frequency for each architecture. To ensure a meaningful comparison, a varied set of input test vectors was chosen during post-route simulation to mimic worst-case switching scenarios. Power analysis was carried out using the physical constraint file (PCF) and the design node activity file obtained during this simulation, utilizing the

Xpower Analyzer tool. As shown in Table 3, the power dissipation associated with the dimension-order routing algorithm varies depending on the clocking resources' activity level, which is directly influenced by clock frequency as specified in the PCF.

TABLE I. SUMMARY OF RESOURCE UTILIZATION COMPARISON DIMENSION ORDER XY ON VIRTEX 5 FPGA.

On chip Resource	Dimension order XY routing
No of Slice LUT	4
No of Slice Registers	14
No. of occupied Slices	2
No of Occupied Slice	3
IOB Flip Flops	4

TABLE II. TIMING SUMMARY FOR DIMENSION ORDER XY ON VIRTEX 5 HARDWARE

Timing Parameter	**Dimension order XY routing**
Maximum frequency (MHz).	551.542
Min available offset-in (ns).	1.397
Min available offset-out (ns).	3.844
Minimum period (ns)	-

The figure 5. Shows the relative delay obtained in the operation of the algorithm using various traffic scenarios. Verification was achieved through extensive simulation and hardware testing, confirming that the algorithm performed as intended under various conditions. The implementation demonstrated a maximum clock frequency of 551.542 MHz,

TABLE III. SUMMARY OF POWER DISSIPATION FOR DIMENSION ORDER XY ON VIRTEX 5.

FPGA Resource	**Power dissipation (mW)**
	Dimension order XY routing
Clock	2.00
Logic	.00
Signals	0.46
I/Os	0.58
Dynamic	3.04

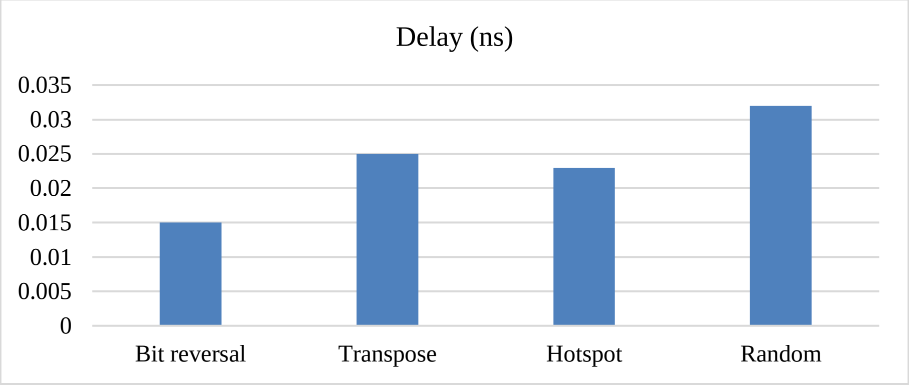

Figure 5. Delay obtained in the implementation using various traffic scanarios.

VI. Conclusion and Future work

This paper introduces a new method for using the FPGA's fabric for optimization of the Dimension Order X-Y routing algorithm, a prerequisite for microarchitecture routers designed in NoC systems. The implementation of the work was aimed at the reconfigralbe Virtex 5 hardware device by Xilinx/AMD. The resultant structures ebbed and flowed in resource utilization within the target FPGA device. It is important to note that the present method for the algorithm's implementation revealed a great deal more efficiency in comparison to work cited in [24]. Ultimately, it is the balance among area, power and throughput parameters, along with certain application requirements, that will render any expandable buffer the most appropriate to enhance the design of an on-chip router. Given the relative importance of area-delay product and power-delay product when targeting an FPGA design, deploying these elastic buffers shall benefit the NoC communication architecture community. In the future, we will explore further configurations such as pipelined buffers, bypass elastic buffers, full-throughput elastic buffers, and fully generic elastic buffers, experimentally evaluating the performance of a credit-based flow control protocol during NoC router communications across neighboring routers.

References

[1] B. Osterloh, H. Michalik, B. Fiethe, K.Kotarowski, “SoCWire: A Network-on-Chip Approach for Reconfigurable System-on-Chip Designs in Space Applications,” in proc of NASA/ESA Conference on Adaptive Hardware and Systems, pp 51-56, june 2008.

[2] Abdelrasul Maher, R. Mohhamed, G. Victor., “Evaluation of The Scalability of Round Robin Arbiters for NoC Routers on FPGA,”7th International symposium on Embedded Multicore/Manycore System-on-chip, pp61-66,2013.

[3] Akram Ben Ahmaed, Abderazek Ben Abdallah, Kenichi, Kuroda, “Architecture and Design of Efficient 3D Network-on-Chip (3D NoC) for custom multicore SoC,” in International confrence on Broadband, Wireless Computing, communication and Application, FIT,Fukuoka, Japan, Nov 2010.

[4] T. Anderson, S. Owicki, J. Saxe, and C. Thacker, “High speed switch scheduling for local area networks,” ACM Trans. Comput. Syst., vol. 11, no. 4, pp. 319–352, Nov. 1993.

[5] M. Karol and M. Hluchyj, “Queueing in high-performance packetswitching,” IEEE J. Select. Areas Commun., vol. 6, pp. 1587–1597, Dec. 1988.

[6] N. McKeown, V. Anantharam, and J. Walrand, “Achieving 100% throughput in an input-queued switch,” in Proc. IEEE INFOCOM ‘96, San Francisco, CA, pp. 296–302.

[7] Yuan-Ying Chang, Huang, Y.S.-C., Poremba, M. Narayanan, V.Yuan, Xie King, C, “Title TS-Router: On maximizing the Quality-of-Allocation in the On-Chip Network,” in IEEE 19th International Symposium on High Performance Computer Architecture (HPCA2013), pp 390-399, Feb 2013.

[8] B. Phanibhushana, K. Ganeshpure, S. Kundu, “Task model for on-chip communication infrastructure design for multicore systems,” in proc of IEEE 29th International Conference on Computer Design (ICCD), pp 360-365, oct 2011.

[9] J. Guo, J. Yao, Laxmi Bhuyan, “An efficient packet scheduling algorithm in network processors,” in proceedings of 24th Annual Joint Conference of the IEEE Computer and Communications Societies ,pp 807- 818, march 2005.

[10] William John Dally, Brain Towels, Principles and Practices of Interconnection Networks, Ist ed. Morgan Kaufmann publications, 2003.

[11] N.Mckeown, “Scheduling algorithms for input buffered cell switches,” Ph.D thesis, University of Calfornia at Berkely,1995.

[12] K. Lee, Se-jee Lee, hoi-jun Yoo, “A Distributed Crossbar Switch Scheduler for On-Chip Networks,” in Custom Integrated Circuits Conference, 2003. Proceedings of the IEEE ,pp 671-674, Sept. 2003.

[13] P. Gupta, N. Mckeown, “ Designing and implementing a Fast Crossbar Scheduler,” in proc of Micro, IEEE, vol 19, no 1, pp 20-28, Feb 1999

[14] Dally, W.J., "Virtual-channel flow control," in Parallel and Distributed Systems, IEEE Transactions on , vol.3, no.2, pp.194-205, Mar 1992

[15] G. Michelogiannakis, J. Balfour, and W.J. Dally, “Elastic Buffer Flow Control for On-Chip Networks,” Proc. IEEE 15th Int’l Symp. High-Performance Computer Architecture (HPCA ’09), pp. 151-162, 2009.

[16] Seitanidis, I.; Psarras, A.; Chrysanthou, K.; Nicopoulos, C.; Dimitrakopoulos, G., "ElastiStore: Flexible Elastic Buffering for Virtual-Channel-Based Networks on Chip," in Very Large Scale Integration (VLSI) Systems, IEEE Transactions on , vol.23, no.12, pp.3015-3028, Dec. 2015

[17] Mubeen, S.; Kumar, S., "Designing Efficient Source Routing for Mesh Topology Network on Chip Platforms," Digital System Design: Architectures, Methods and Tools (DSD), 2010 13th Euromicro Conference on , vol., no., pp.181,188, 1-3 Sept. 2010

[18] C.J. Glass and L.M. Ni, ªThe Turn Model for Adaptive Routing,º Proc. 19th Ann. Int'l Symp. Computer Architecture, pp. 278±287, May 1992.

[19] C.J. Glass and L.M. Ni, ªThe Turn Model for Adaptive Routing,º J. ACM, vol. 41, no. 5, pp. 874±902, Sept. 1994.

[20] A.A. Chien and J.H. Kim, ªPlanar-Adaptive Routing: Low-Cost Adaptive Networks for Multiprocessors,º J. ACM, vol. 42, no. 1, pp. 91±123, Jan. 1995

[21] C.J. Glass, L.M. Ni, The turn model for adaptive routing. J. Assoc. Comput. Mach. **41**(5), 874–902 (1994).

[22] Mak, Terrence, Maurizio Palesi, and Masoud Daneshtalab. "Special issue on emerging on-chip networks and architectures." IET Computers & Digital Techniques 7.6 (2013): 235.

[23] Gaide, B., Gaitonde, D., Ravishankar, C., & Bauer, T. (2019, February). Xilinx adaptive compute acceleration platform: Versaltm architecture. In Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (pp. 84-93).

[24] Feng, F., Jun, Y. B., Liu, X., & Li, L. (2010). An adjustable approach to fuzzy soft set based decision making. Journal of Computational and Applied Mathematics, 234(1), 10-20.

[25] Ebrahimi, M., Daneshtalab, M., Plosila, J., & Mehdipour, F. (2013, January). MD: minimal path-based fault-tolerant routing in on-chip networks. In 2013 18th Asia and South Pacific Design Automation Conference (ASP-DAC) (pp. 35-40). IEEE.

[26] Zhang, X., Ebrahimi, M., Huang, L., & Li, G. (2015, September). Fault-resilient routing unit in nocs. In 2015 28th IEEE International System-on-Chip Conference (SOCC) (pp. 164-169). IEEE

[27] Salamat, R., Khayambashi, M., Ebrahimi, M., & Bagherzadeh, N. (2016). A resilient routing algorithm with formal reliability analysis for partially connected 3D-NoCs. IEEE Transactions on Computers, 65(11), 3265-3279.

[28] Lin, H., Li, M. F., Jia, C. F., Liu, J. N., & An, H. (2019). Degree-of-node task scheduling of fine-grained parallel programs on heterogeneous systems. Journal of Computer Science and Technology, 34, 1096-1108.

[29] Lin, S. Y., Yin, T. C., Wang, H. Y., & Wu, A. Y. (2011, April). Traffic-and thermal-aware routing for throttled three-dimensional network-on-chip systems. In Proceedings of 2011 International Symposium on VLSI Design, Automation and Test (pp. 1-4). IEEE.

[30] Lan, Y. C., Lin, H. A., Lo, S. H., Hu, Y. H., & Chen, S. J. (2011). A bidirectional NoC (BiNoC) architecture with dynamic self-reconfigurable channel. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 30(3), 427-440.

[31] Salamat, R., Khayambashi, M., Ebrahimi, M., & Bagherzadeh, N. (2018). LEAD: An adaptive 3D-NoC routing algorithm with queuing-theory based analytical verification. IEEE Transactions on Computers, 67(8), 1153-1166.

[32] Shahane, P., & Pisharoty, N. (2019). Modified xy routing algorithm for mesh topology based noc router on fpga. IET Circuits, Devices & Systems, 13(01).

[33] Van Chu, T., & Kise, K. (2019). LEF: An effective routing algorithm for two-dimensional meshes. IEICE TRANSACTIONS on Information and Systems, 102(10), 1925-1941.

[34] Ni, W., & Liu, Z. (2015, November). A routing algorithm for Network-on-Chip with self-similar traffic. In 2015 IEEE 11th International Conference on ASIC (ASICON) (pp. 1-4). IEEE.

[35] G.-M. Chiu, “The odd-even turn model for adaptive routing,” IEEE Trans. Parallel Distrib. Syst., vol. 11, no. 7, pp. 729–738, 2000.

[36] C. J. Glass and L. M. Ni, “The turn model for adaptive routing,” Journal of Assoc. Comput. Mach. (JACM), vol. 41, no. 5, pp. 874–902, 1994.

[37] Killian, Cédric, et al. "A new efficient and reliable dynamically reconfigurable network-on-chip." Journal of Electrical and computer Engineering 2012 (2012).

[38] http://www.xilinx.com.

[39] Romli, N., Minhad, K., Reaz, M., & Amin, M. S. (2015). An overview of power dissipation and control techniques in cmos technology. J. Eng. Sci. Technol, 10(3), 364-382.