J. Integr. Sci. Technol. 2024, 12(2), 733 Article

Journal of Integrated
SCIENCE & TECHNOLOGY

Area and delay trade offs in fracturable LUT-based FPGA architectures

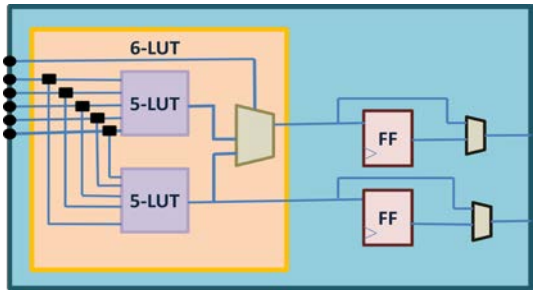
Neeti,* Sunita Dahiya

Department of Electronics and Communication Engineering, DCR University of Science and Technology, Murthal, Haryana. India

Received on: 17-Jul-2023, Accepted and Published on: 02-Oct-2023

ABSTRACT

Now-a-days, most commercial Field- Programmable Gate Arrays (FPGAs) are based on fracturable look-up tables (LUTs). A fracturable LUT based FPGA can operate in two modes: one without shared input and other with shared inputs. This paper investigates area and critical path delay of 6-LUT based fracturable FPGAs for different cluster sizes and cluster inputs. It is found experimentally that compared to non-fracturable 6-LUT based FPGAs, the fracturable 6-LUT based FPGAs with cluster sizes 7 to 10 show significant improvement in the area-delay results, with different chosen values of cluster inputs.



Keywords: Fracturable, non-fracturable, FPGA, logic block, LUT, cluster.

INTRODUCTION

The reconfigurability of Field Programmable Gate Arrays (FPGAs) allows them to be used in myriad application areas such as networking,[1] data security[2] and image compression,[3] among others. However, FPGA-based implementations are comparatively larger and slower in comparison to the Application-Specific Integrated Circuit (ASIC) implementations.[4] The choice of logic block in an FPGA is an important parameter which influences the area and delay of an FPGA. Hence, the logic element chosen should lower this area and performance gap. Prior research works[5–9] on FPGA architectures were based on homogeneous logic blocks, with most of the studies using a k-input look-up table (LUT) as the logic element. The flexibility of LUTs is the prime reason for their widespread use in FPGAs. However, the area of LUTs increases exponentially with the increase in their size. Therefore, LUTs with larger inputs are not generally used. Some of the recent studies have investigated alternative FPGA architectures based on fracturable LUTs.[10–13] A fracturable LUT can operate either as a single larger LUT or be split into two smaller LUTs using input sharing. Fracturable LUTs involve sharing of inputs which enables FPGA designers to get the benefits of comparatively larger input LUTs. This paper investigates fracturable 6-input LUT based clusters with different cluster inputs and cluster sizes. The impact of fracturable LUTs on FPGA area as well as performance has been evaluated using VTR tool flow.

The contribution of the paper is as given here:

- This paper evaluates area as well as critical delay for various fracturable LUT-based FPGA architectures.
- Further, the area as well as speed of these fracturable architectures is compared with the non-fracturable LUT-based FPGAs.

The rest of the paper is categorized as under: Section II outlines the related work in the field of FPGA logic block architectures. Section III describes a fracturable LUT-based FPGA architecture. The methodology used for evaluation of different FPGA architectures has been discussed in Section IV. Section V presents the area as well as delay results. The conclusion and future work are summarized in Section VI.

RELATED WORK

The earlier published studies[5–9] on FPGAs were based on homogeneous logic blocks. J. Rose et. al. [5] used an arbitrary logic block for investigation whereas the research study[6] was based on a k-input LUT. Experimentally, it was found that 3 or 4-input LUTs are most efficient in terms of area. Further, the studies[7–9] investigated the impact of various logic blocks on FPGA speed performance.

Relative to the prior studies[5–9] which investigated the impact of homogeneous logic elements on FPGA area or performance, the

*Corresponding Author: Neeti, DCRUST Murthal, Haryana, India.
Email: neeti9chahar@gmail.com

Cite as: *J. Integr. Sci. Technol., 2024, 12(2), 733.*

 ISSN: 2321-4635
http://pubs.thesciencein.org/jist

research works[14–19] focused on the FPGAs based on group of LUTs forming a logic cluster. These studies evaluated the impact of different cluster parameters on FPGA area or speed or both.

Further, the studies[10–13] evaluated FPGA architectures based on fracturable LUTs. The research works [10,11] introduced a fracturable LUT based logic module. It was found experimentally that the proposed adaptive logic element improved performance by 15% and area by 12% as compared to traditional BLE4 based FPGAs. The study[12] combined edge recovery technique with LUT balancing in order to reduce the number of fracturable LUTs in the mapping. G. Zgheib et. al.[13] evaluated fractuable k-LUT based FPGAs for different cluster sizes. Three choices for shared inputs were considered for investigation. However, the results do not show any improvement as compared to non-fracturable architectures.

In the earlier works on fracturable FPGAs, the impact of varying cluster inputs has not been discussed. Also, area-delay tradeoffs in case of fracturable LUTs were not examined. The present research work investigates fracturable 6-LUT based FPGAs with varying cluster sizes as well as cluster inputs in 40nm technology.

FRACTURABLE LUTS

A fracturable LUT has two different operating modes: normal and fractured. In normal mode, the fracturable LUT acts as k-input LUT whereas in fractured mode, the fracturable LUT is split into two (k-1) LUTs with shared inputs. These two modes of fracturable LUTs are shown in figure 1.

The important parameters in a fracturable LUT-based logic cluster are:

1. LUT size (k): the total number of inputs given to a LUT.
2. Cluster size (N): number of basic logic elements (BLEs) in a cluster.
3. Cluster inputs (I): number of distinct inputs to a cluster.
4. Shared inputs (S): the number of inputs shared in fractured mode.

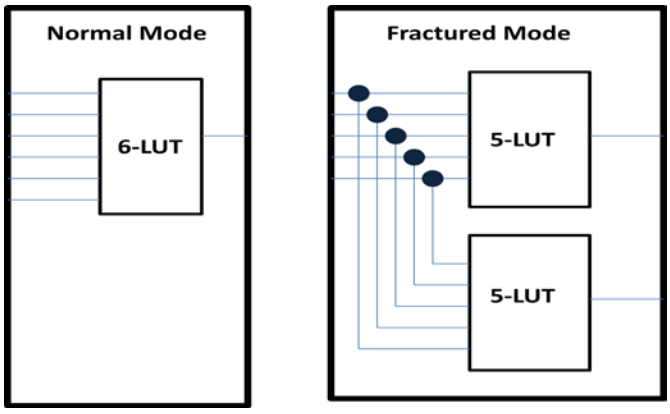


Figure 1: Fracturable LUT: two operating modes.

In the present study, we use fracturable 6-input LUTs, where each 6-LUT can be split into two 5-LUTs with input sharing. In the experiments, all the inputs are shared in fractured mode. FPGA architectures use logic cluster sizes varying from 7 to 10.

METHODOLOGY

This section describes the methodology used to evaluate the area and speed of FPGAs employing fracturable LUT based logic blocks. The present study uses 20 MCNC benchmark circuits that have been widely used in FPGA architecture based research works. The VTR CAD flow is used for the evaluation of the impact of benchmark circuits on the area and performance of the fracturable-LUT based FPGAs.

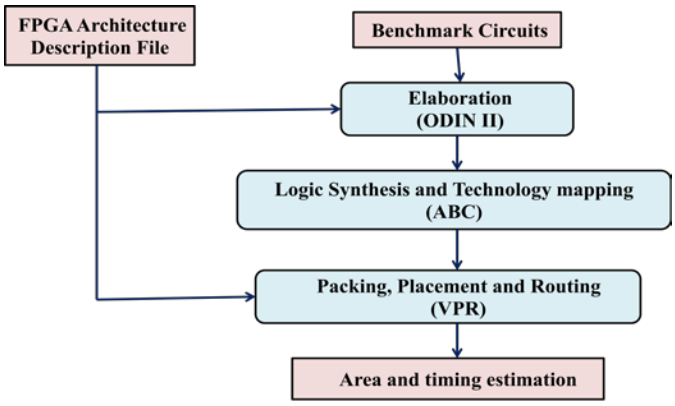


Figure 2: VTR CAD flow.

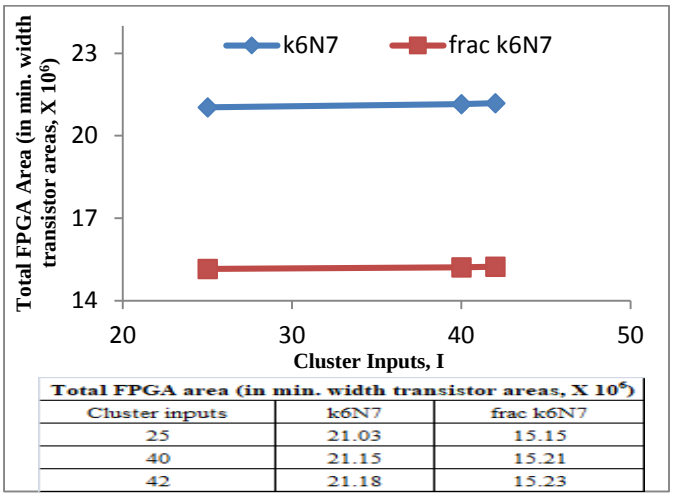
As illustrated in figure 2, there are two inputs to the VTR CAD flow, one is benchmark circuit and the other is FPGA architecture description file.[20] Further, ODIN II[21] performs elaboration and partial synthesis of the Verilog code so as to create a netlist. Then, ABC tool is used to do logic synthesis as well as mapping of the logic into k-LUTs and flip-flops. Finally, VPR tool is used for the clustering, placement and routing of the benchmark circuit.[22]

RESULTS

In the present study, we evaluate the impact of different cluster parameters on the area and performance of fracturable LUT based FPGAs. For all the experiments, LUT size is chosen to be 6 whereas cluster size is varied from 7 to 10. The experiments are repeated for three different values of cluster inputs, I. Fracturable 6-LUTs involve sharing of all the inputs in fractured mode. The area-delay results obtained for fracturable FPGAs are compared with that of non-fracturable FPGAs.

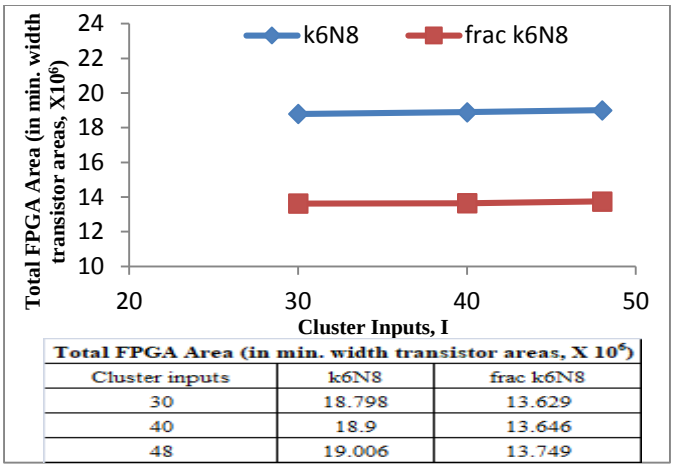
Area results for fracturable 6-LUT vs. non-fracturable 6-LUT based clusters for different cluster inputs

Here, we will evaluate the effect of varying cluster inputs on total FPGA area for different cluster sizes in case of fracturable as well as non-fracturable 6-LUT based FPGA clusters. The variation of geometric mean of total FPGA area for cluster sizes 7, 8, 9 and 10 is illustrated in figure 3, 4, 5 and 6 respectively. As shown in the graph, the fracturable LUT based FPGA architectures exhibit better area results as compared to non-fracturable FPGAs and the graphs show similar trend in area results for all the cluster sizes.



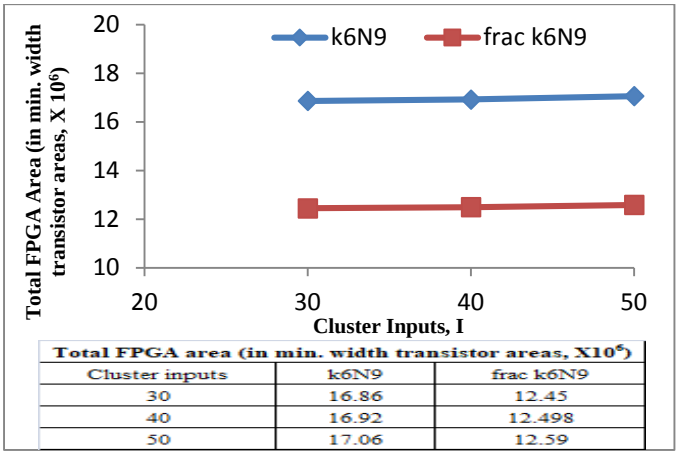
Total FPGA area (in min. width transistor areas, X 10^6)		
Cluster inputs	k6N7	frac k6N7
25	21.03	15.15
40	21.15	15.21
42	21.18	15.23

Figure 3. Total FPGA Area for different cluster inputs in fracturable 6-LUT and non-fracturable 6-LUT based clusters (Cluster size, N=7).



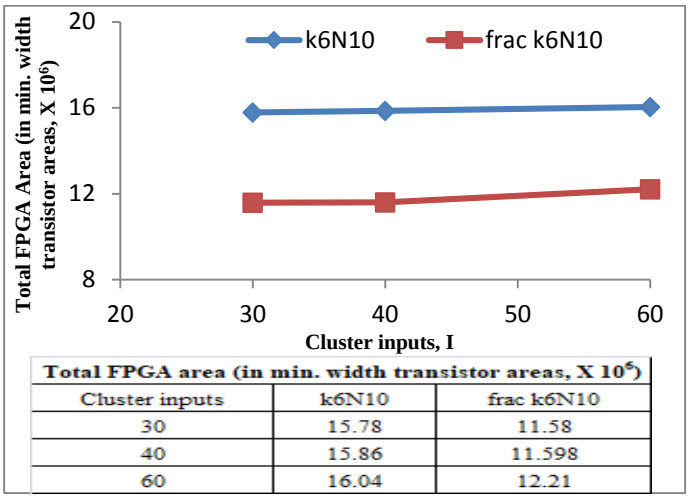
Total FPGA Area (in min. width transistor areas, X 10^6)		
Cluster inputs	k6N8	frac k6N8
30	18.798	13.629
40	18.9	13.646
48	19.006	13.749

Figure 4. Total FPGA Area for different cluster inputs in fracturable 6-LUT and non-fracturable 6-LUT based clusters (Cluster size, N=8).



Total FPGA area (in min. width transistor areas, X10^6)		
Cluster inputs	k6N9	frac k6N9
30	16.86	12.45
40	16.92	12.498
50	17.06	12.59

Figure 5. Total FPGA Area for different cluster inputs in fracturable 6-LUT and non-fracturable 6-LUT based clusters (Cluster size, N=9).



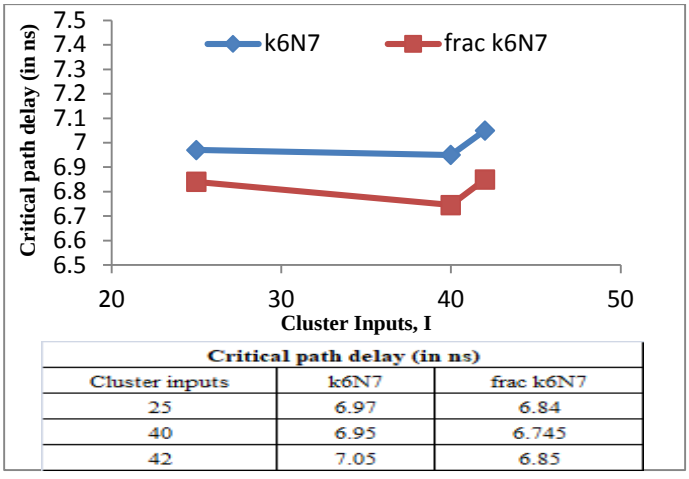
Total FPGA area (in min. width transistor areas, X 10^6)		
Cluster inputs	k6N10	frac k6N10
30	15.78	11.58
40	15.86	11.598
60	16.04	12.21

Figure 6. Total FPGA Area for different cluster inputs in fracturable 6-LUT and non-fracturable 6-LUT based clusters (Cluster size, N=10).

Critical path delay results for fracturable 6-LUT vs. non-fracturable 6-LUT based clusters for different cluster inputs

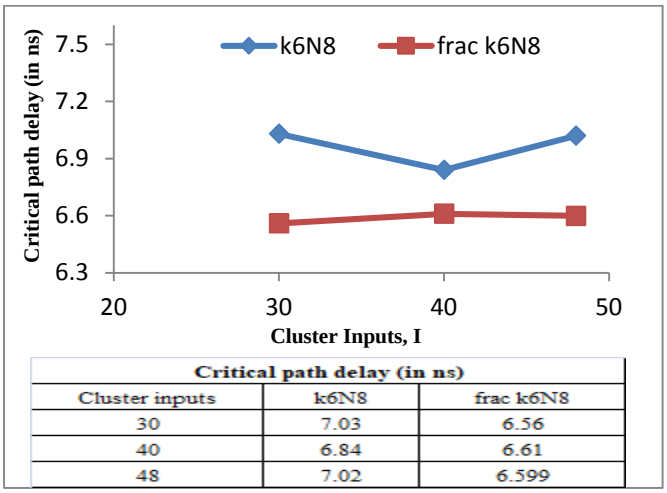
Area and delay are two important metrics which are used to examine various FPGA architectures. Here, we will investigate the impact of varying cluster inputs on the critical delay values of fracturable as well as non-fracturable FPGAs for different cluster sizes. Figure 7, 8, 9 and 10 show how the geometric mean of critical delay changes with the cluster inputs for clusters of size 7, 8, 9 and 10 respectively in case of fracturable and non-fracturable FPGAs.

For all the cluster sizes shown in the graphs, it is clear that the delay results are better for fracturable FPGAs in comparison to the non-fracturable ones. However, the best delay values are obtained for fracturable 6-LUT based FPGA with cluster size 10 using 40 cluster inputs.



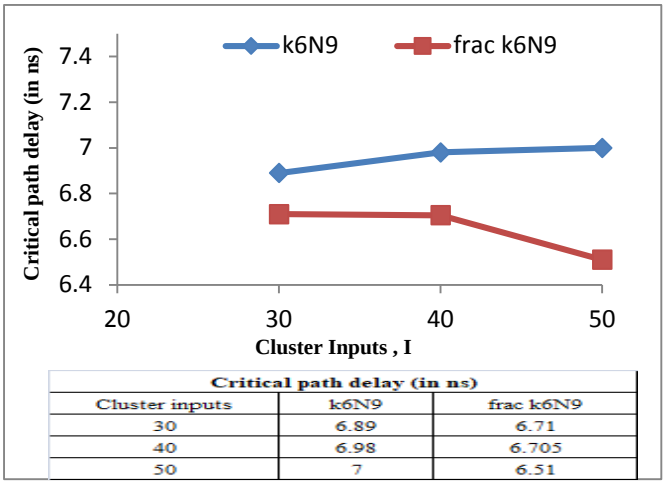
Critical path delay (in ns)		
Cluster inputs	k6N7	frac k6N7
25	6.97	6.84
40	6.95	6.745
42	7.05	6.85

Figure 7. Total Critical path delay for different cluster inputs in fracturable 6-LUT and non-fracturable 6-LUT based clusters (Cluster size, N=7).



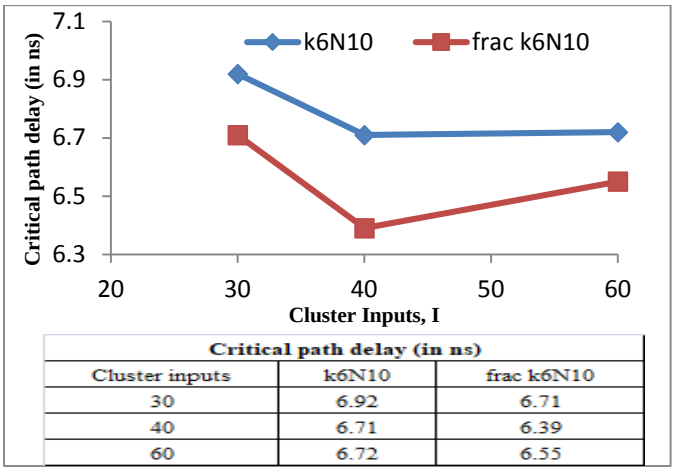
Critical path delay (in ns)		
Cluster inputs	k6N8	frac k6N8
30	7.03	6.56
40	6.84	6.61
48	7.02	6.599

Figure 8. Total Critical path delay for different cluster inputs in fracturable 6-LUT and non-fracturable 6-LUT based clusters (Cluster size, N=8).



Critical path delay (in ns)		
Cluster inputs	k6N9	frac k6N9
30	6.89	6.71
40	6.98	6.705
50	7	6.51

Figure 9. Total Critical path delay for different cluster inputs in fracturable 6-LUT and non-fracturable 6-LUT based clusters (Cluster size, N=9).



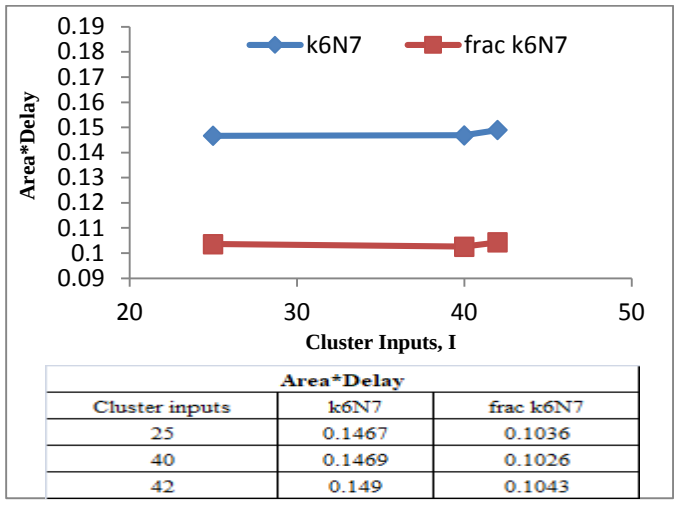
Critical path delay (in ns)		
Cluster inputs	k6N10	frac k6N10
30	6.92	6.71
40	6.71	6.39
60	6.72	6.55

Figure 10. Total Critical path delay for different cluster inputs in fracturable 6-LUT and non-fracturable 6-LUT based clusters (Cluster size, N=10).

Area-delay trade-off in fracturable 6-LUT vs. non-fracturable 6-LUT based clusters for different cluster inputs

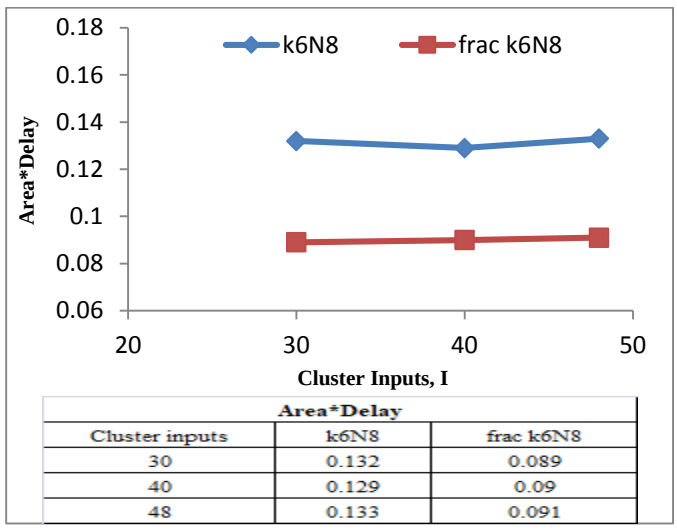
Area-delay product is used as the criteria to compare the quality of different FPGA architectures. The FPGA architecture which yields minimum value for area-delay product is considered best. Figure 11, 12, 13 and 14 show the geometric mean of the area-delay product versus cluster inputs for cluster sizes 7, 8, 9 and 10 respectively.

As can be seen in the graphs, fracturable architectures exhibit lower values of area-delay product as compared to that of non-fracturable ones for the cluster sizes 7 to 10. The results demonstrate that any fracturable 6-LUT based cluster of size 7 to 10 is a reasonably good choice. Specifically, the area-delay product is found to be lowest for fracturable 6-LUT based cluster of size 10 with 40 distinct inputs.



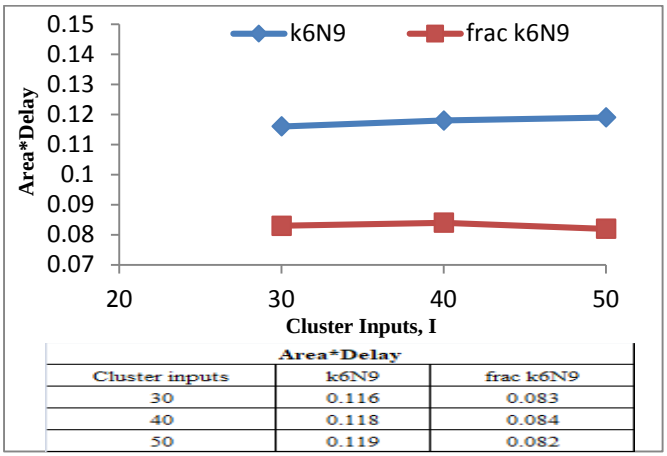
Area*Delay		
Cluster inputs	k6N7	frac k6N7
25	0.1467	0.1036
40	0.1469	0.1026
42	0.149	0.1043

Figure 11. Area-delay product for different cluster inputs in fracturable 6-LUT and non-fracturable 6-LUT based clusters (Cluster size, N=7).



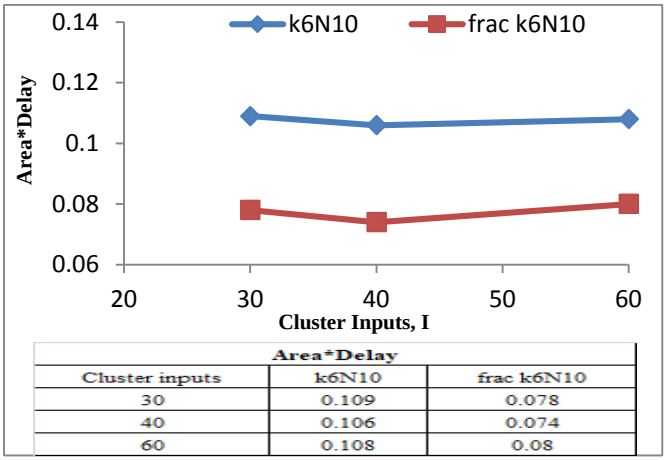
Area*Delay		
Cluster inputs	k6N8	frac k6N8
30	0.132	0.089
40	0.129	0.09
48	0.133	0.091

Figure 12. Area-delay product for different cluster inputs in fracturable 6-LUT and non-fracturable 6-LUT based clusters (Cluster size, N=8).



Area*Delay		
Cluster inputs	k6N9	frac k6N9
30	0.116	0.083
40	0.118	0.084
50	0.119	0.082

Figure 13. Area-delay product for different cluster inputs in fracturable 6-LUT and non-fracturable 6-LUT based clusters (Cluster size, N=9).



Area*Delay		
Cluster inputs	k6N10	frac k6N10
30	0.109	0.078
40	0.106	0.074
60	0.108	0.08

Figure 14. Area-delay product for different cluster inputs in fracturable 6-LUT and non-fracturable 6-LUT based clusters (Cluster size, N= 10).

CONCLUSION

This paper examines area as well as delay of FPGA architectures based on fracturable 6-LUTs. The impact of varying cluster sizes for different cluster input values has been investigated. The experimental results demonstrate that the fracturable 6-LUT based FPGA architectures exhibit better area as well as delay results as compared to the non-fracturable FPGAs for all the cluster sizes from 7 to 10. Specifically, the fracturable 6-LUT based size 10 cluster using 40 cluster inputs is found to have the lowest area-delay product.

In the future, we would like to explore more fracturable LUT based FPGAs with different LUT sizes and considering different values of shared inputs in fractured mode.

CONFLICT OF INTEREST

The authors do not have any academic, financial or otherwise any conflict of interest for publication of this work in this journal.

REFERENCES

1. K. Bandura, A.N. Bender, J.F. Cliche, et al. ICE: A Scalable, Low-Cost FPGA-Based Telescope Signal Processing and Networking System. *J. Astron. Instrum.* **2016**, 05 (04), 1641005.
2. S. Sanap, V. More. Field Programmable Gate Array (FPGA) based implementation of efficient RC6 block cipher. *J. Integr. Sci. Technol.* **2022**, 10 (2), 168–172.
3. N. Patra, S.S. Nayak. Discrete Hartley Transform and its applications - A review. *J. Integr. Sci. Technol.* **2022**, 10 (2), 173–179.
4. I. Kuon, J. Rose. Measuring the gap between FPGAs and ASICs. In *Proceedings of the 2006 ACM/SIGDA 14th international symposium on Field programmable gate arrays*; ACM, Monterey California USA, **2006**; pp 21–30.
5. J. Rose, R.J. Francis, P. Chow, D. Lewis. The effect of logic block complexity on area of programmable gate arrays. In *1989 Proceedings of the IEEE Custom Integrated Circuits Conference*; IEEE, San Diego, CA, USA, **1989**; p 5.3/1-5.3/5.
6. J. Rose, R.J. Francis, D. Lewis, P. Chow. Architecture of field-programmable gate arrays: the effect of logic block functionality on area efficiency. *IEEE J. Solid-State Circuits* **1990**, 25 (5), 1217–1225.
7. S. Singh, J. Rose, D. Lewis, K. Chung, P. Chow. Optimization of field-programmable gate array logic block architecture for speed. In *Proceedings of the IEEE 1991 Custom Integrated Circuits Conference*; IEEE, San Diego, CA, USA, **1991**; p 6.1/1-6.1/6.
8. J.L. Kouloheris, A. El Gamal. FPGA performance versus cell granularity. In *Proceedings of the IEEE 1991 Custom Integrated Circuits Conference*; IEEE, San Diego, CA, USA, **1991**; p 6.2/1-6.2/4.
9. S. Singh, J. Rose, P. Chow, D. Lewis. The effect of logic block architecture on FPGA performance. *IEEE J. Solid-State Circuits* **1992**, 27 (3), 281–287.
10. 1. D. V. Sorokin, D.E. Dragunov, M.Y. Lyapustin, N.A. Semenchenko, K.A. Sharganov. FPGA-Based Methods for High-Speed Processing of Video Images with Large Brightness Scatter. *J. Commun. Technol. Electron.* **2023**, 68 (3), 360–365.
11. M. Hutton, J. Schleicher, D. Lewis, et al. Improving FPGA Performance and Area Using an Adaptive Logic Module. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Becker, J., Platzner, M., Vernalde, S., Eds.; Springer, Berlin, Heidelberg, **2004**; Vol. 3203, pp 135–144.
12. D. Dickin, L. Shannon. Exploring FPGA technology mapping for fracturable LUT minimization. In *2011 International Conference on Field-Programmable Technology*; IEEE, New Delhi, India, **2011**; pp 1–8.
13. G. Zgheib, P. Ienne. Evaluating FPGA clusters under wide ranges of design parameters. In *2017 27th International Conference on Field Programmable Logic and Applications (FPL)*; IEEE, Ghent, Belgium, **2017**; pp 1–8.
14. V. Betz, J. Rose. How much logic should go in an FPGA logic block. *IEEE Des. Test. Comput.* **1998**, 15 (1), 10–15.
15. V. Betz, J. Rose. Cluster-based logic blocks for FPGAs: area-efficiency vs. input sharing and size. In *Proceedings of CICC 97 - Custom Integrated Circuits Conference*; IEEE, Santa Clara, CA, USA, **1997**; pp 551–554.
16. A. Marquardt, V. Betz, J. Rose. Speed and area tradeoffs in cluster-based FPGA architectures. *IEEE Trans. VLSI Syst.* **2000**, 8 (1), 84–93.
17. E. Ahmed, J. Rose. The effect of LUT and cluster size on deep-submicron FPGA performance and density. *IEEE Trans. VLSI Syst.* **2004**, 12 (3), 288–298.
18. I. Kuon, J. Rose. Area and delay trade-offs in the circuit and architecture design of FPGAs. In *Proceedings of the 16th international ACM/SIGDA symposium on Field programmable gate arrays*; ACM, Monterey California USA, **2008**; pp 149–158.
19. Neeti, S. Dahiya. Evaluating the impact of cluster parameters on FPGA performance and density. *J. Integr. Sci. Technol* 11 (3), 520.
20. J. Luu, J.H. Anderson, J.S. Rose. Architecture description and packing for logic blocks with hierarchy, modes and complex interconnect. In *Proceedings of the 19th ACM/SIGDA international symposium on Field programmable gate arrays*; ACM, Monterey CA USA, **2011**; pp 227–236.
21. P. Jamieson, K.B. Kent, F. Gharibian, L. Shannon. Odin II - An Open-Source Verilog HDL Synthesis Tool for CAD Research. In *2010 18th IEEE Annual International Symposium on Field-Programmable Custom Computing Machines*; IEEE, Charlotte, NC, USA, **2010**; pp 149–156.
22. J. Luu, J. Goeders, M. Wainberg, et al. VTR 7.0: Next Generation Architecture and CAD System for FPGAs. *ACM Trans. Reconfigurable Technol. Syst.* **2014**, 7 (2), 1–30.