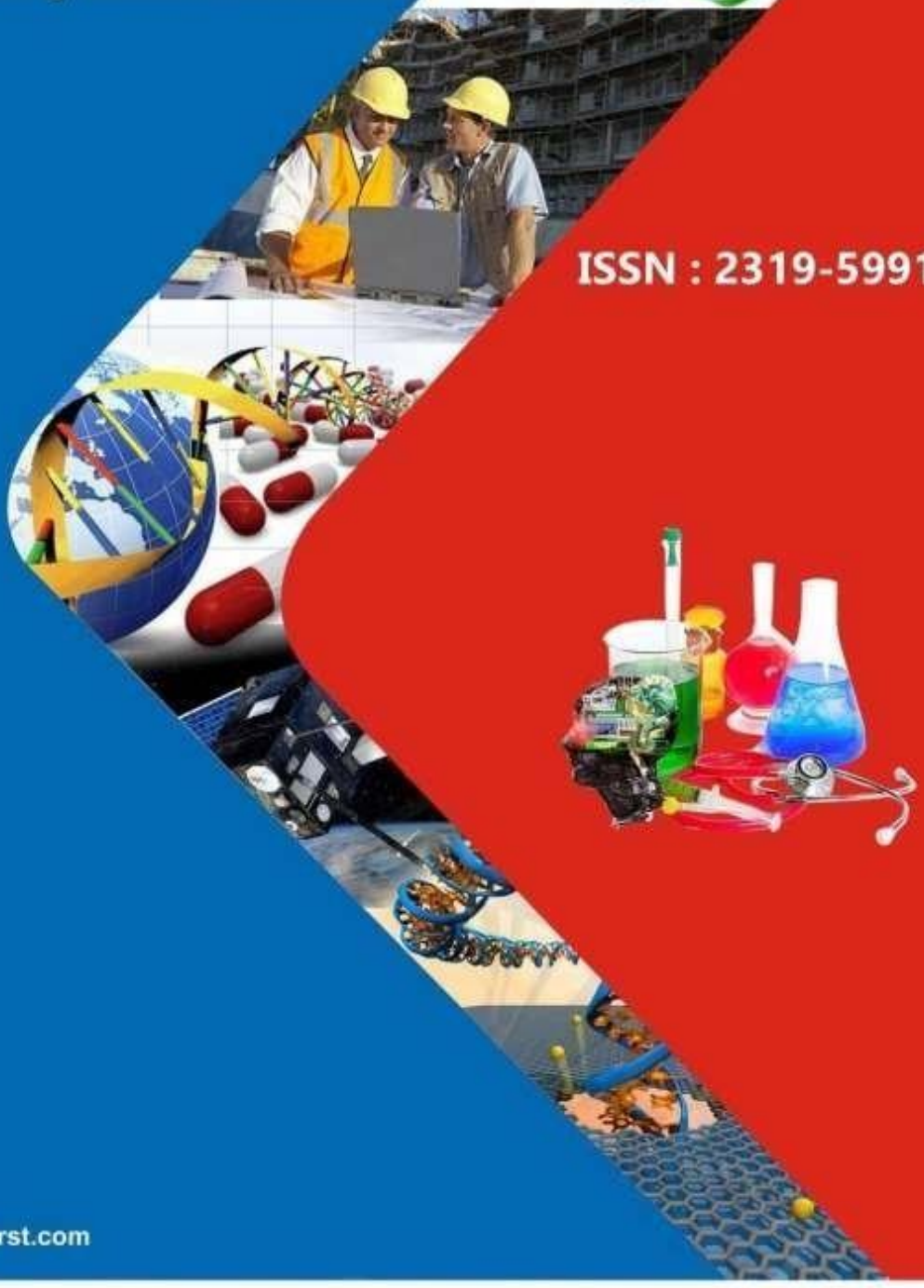
International Journal of Engineering Research and Science & Technology

IJERST

ISSN : 2319-5991



www.ijerst.com

Email: editor@ijerst.com or editor.ijerst@gmail.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 20, Issuse.1, March 2024

DESIGN AND IMPLEMENTATION OF DIRECT AND POST TRUNCATED ADDER TREE

Mr. Y.VIJAY KUMAR [1], T. S.N.LAKSHMI SIRISHA [2], N.A.V.LAKSHMI PARVATHI[3], V.S.SRI SARANYA[4] , M.N.L.SAHITY[5], J.GOPICHAND[6]

[1]Assistant Professor , Dept.of ECE, PRAGATI ENGINEERING COLLEGE

[23456]UG Students,Dept.of ECE, PRAGATI ENGINEERING COLLEGE

ABSTRACT

In this project, a novel scheme is presented to obtain fixed width AT design using truncated input. Conventionally, fixed-width adder-tree (AT) design is obtained from the full-width AT design by employing direct or post-truncation. In direct-truncation, one lower order bit of each adder output of full-width AT is post-truncated, and in case of post-truncation, “p” lower order-bits of final-stage adder output are truncated. The proposed fixed-width AT design for input-vector sizes 8 offers (37%; 23%; 22%) and (51%; 30%; 27%) area saving for word-length sizes (8; 12; 16), respectively, and calculates the output almost with the same accuracy as the post-truncated fixed-width AT which has the highest accuracy among the existing fixed-width AT

INTRODUCTION

Introduction to Ripple Carry Adder and Adder Tree

The need for fast and efficient addition of binary numbers is a critical requirement in many digital systems. In modern high-speed digital systems, the use of multiple binary numbers in parallel is becoming increasingly common. This has led to the development of various techniques for performing the addition of multiple binary numbers in parallel, including the use of adder trees.

An adder tree is a circuit that performs the addition of multiple binary numbers in parallel. It consists of a series of full adders arranged in a tree-like structure, with each level of the tree performing the addition of a subset of the input numbers. This approach allows for fast and efficient addition of multiple binary numbers, with the carry generated by each level of the tree being propagated to the next level until it reaches the top level, where the final sum is obtained. One type of adder tree that is commonly used is the ripple carry based adder tree. This type of adder tree combines the concept of a ripple carry adder with the efficiency of an adder tree to provide a fast and efficient way to add multiple binary numbers.

ISSN 2319-5991 www.ijerst.com
Vol. 20, Issuse.1, March 2024

LITERATURE SURVEY

1. **"Survey of Adder Architectures for Fixed-Width Integer Addition"** by Chien-MoLi et al. (2015): This paper provides a comprehensive survey of the state-of-the-art fixed-width adder architectures for integer addition. It covers both parallel and serial adders and discusses their advantages and disadvantages in terms of performance, power consumption, and area.

2. **"Efficient Design Techniques for Fixed-Width Adder Trees"** by Alok Kumar et al. (2017): This paper presents a survey of the various design techniques used for fixed-width adder trees. It covers carry look-ahead adders, carry select adders, carry skip adders, and other related designs. The authors also discuss the impact of technology scaling on the performance and power consumption of adder trees.

3. **"A Survey of Low-Power Adder Design Techniques**" by Rupali Shukla et al. (2019):This paper provides a survey of low-power adder design techniques, with a focus on fixed- width adder trees. It covers various techniques such as gate sizing, transistor sizing, and clock gating. The authors also discuss the trade-offs between power consumption, area, and performance for each technique.

4. **"A Survey of High-Performance Adder Designs**" by Yong-Bin Kim et al. (2020): This paper presents a survey of high-performance adder designs, including fixed-width adder trees. It covers various techniques such as carry-save adders, pipelined adders, and hybrid adders. The authors also discuss the impact of process variation on the performance and reliability of adder designs

PROPOSED SYSTEM

The proposed system introduces an efficient design for a fixed-width adder tree, aiming to address the limitations of the existing full-width adder tree architecture. The key innovationlies in optimizing the adder tree structure to achieve improved performance and versatility, particularly in applications with specific fixed-width requirements. Unlike the existing system, which processes input operands of full width in each adder stage, the proposed system focuses on designing adder stages tailored to handle fixed-width inputs.

By customizing the adder stages to match the required operand size, the proposed system reduces unnecessary overhead and improves overall efficiency. Additionally, the proposed system incorporates techniques to enhance scalability and adaptability. It allows for easy expansion or contraction of the adder tree structure to accommodate varying fixed-width

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 20, Issuse.1, March 2024

requirements, ensuring optimal performance across a range of applications. Furthermore, the efficient design of the fixed-width adder tree in the proposed system helps minimize power consumption and area utilization, making it suitable for low-power and resource- constrained environments. In summary, the proposed system offers a more optimized andadaptable solution for fixed-width addition operations compared to the existing full-widthadder tree architecture. By leveraging tailored adder stages and scalable design principles, it provides enhanced performance, flexibility, and efficiency in digital arithmetic circuits.

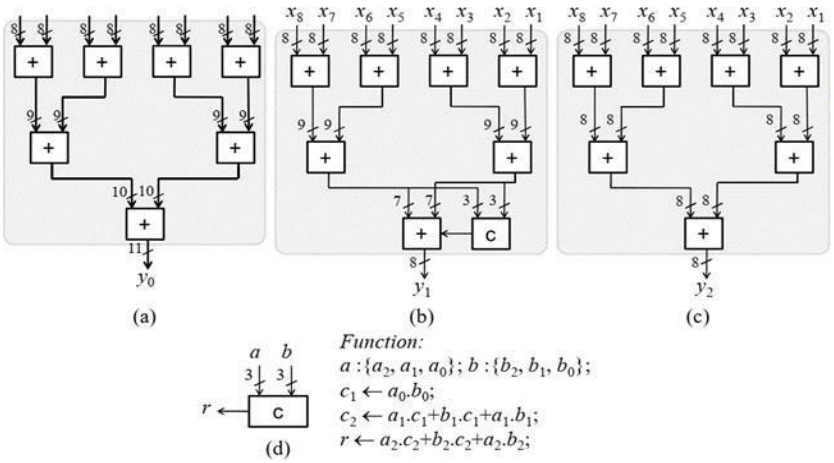


Figure.1 Proposed Truncated Fixed-width Adder-Tree:

SIMULATION, SYNTHESIS RESULTS

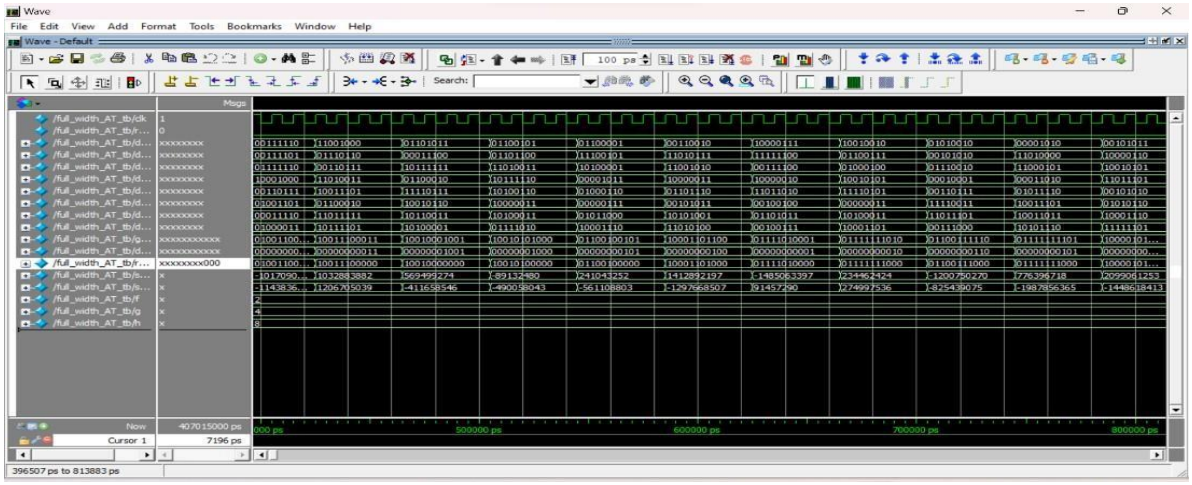


Figure.2 Direct truncated adder tree (8x8)

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 20, Issuse.1, March 2024

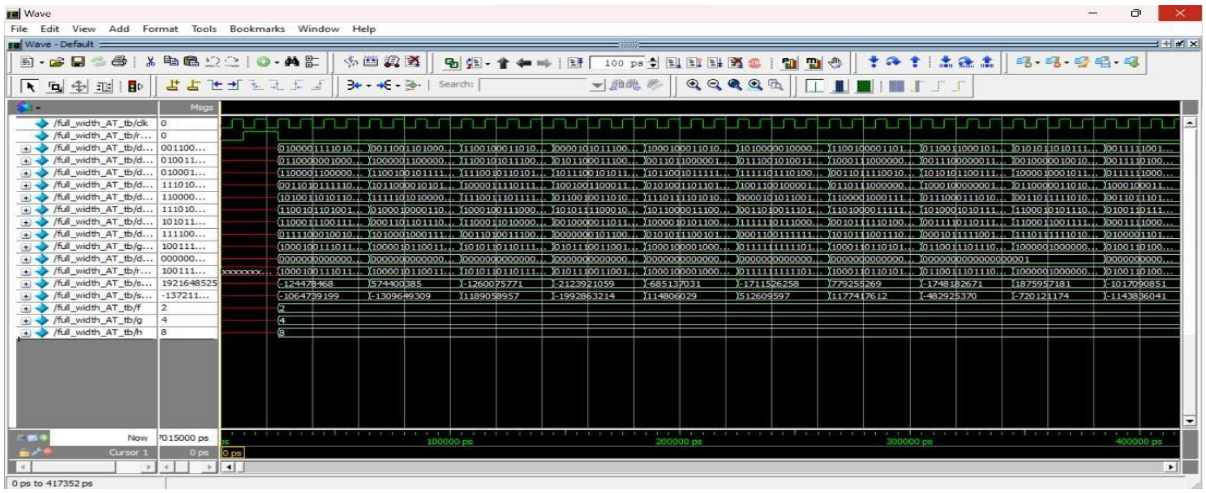


Figure.2 Post truncated adder tree(8x12)

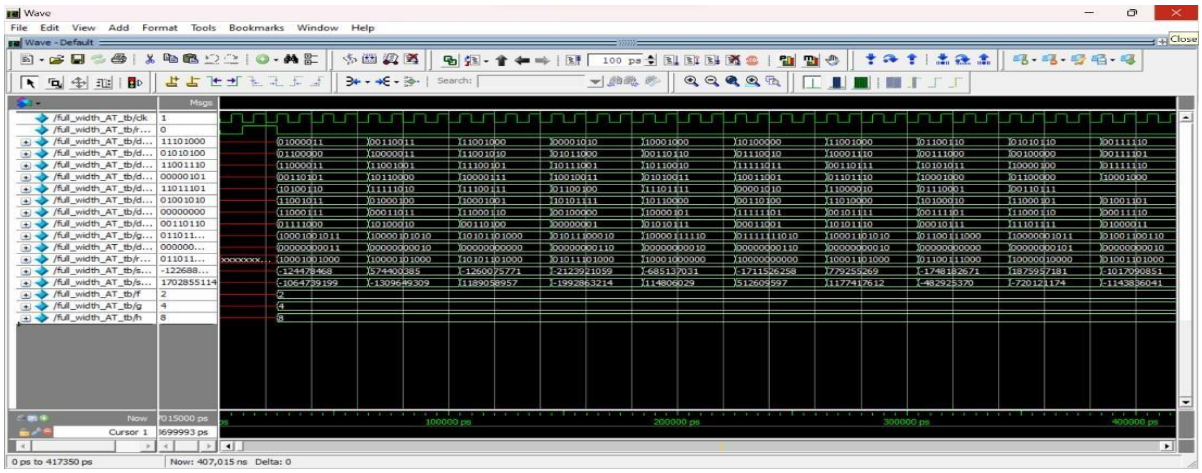


Figure.3 ITFX adder tree(8x16)

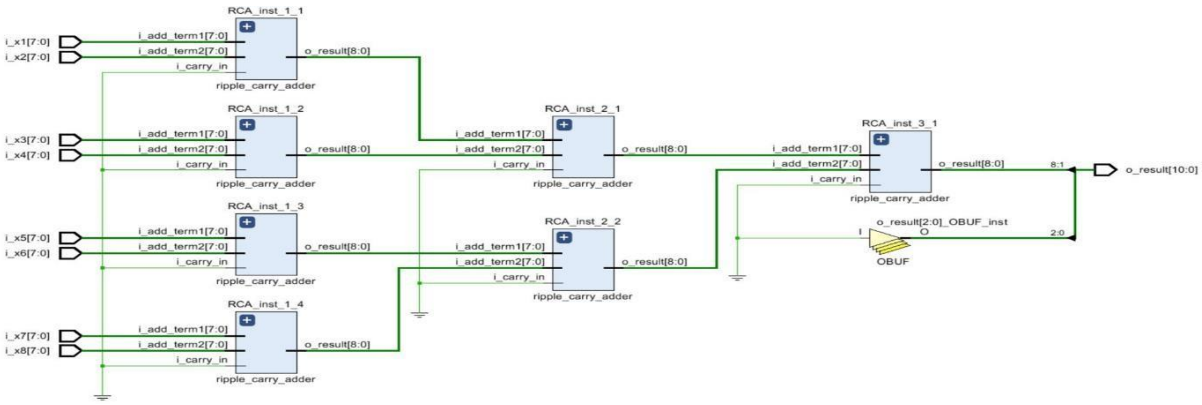


Figure.4 Schematic diagram of direct truncated adder tree

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 20, Issuse.1, March 2024

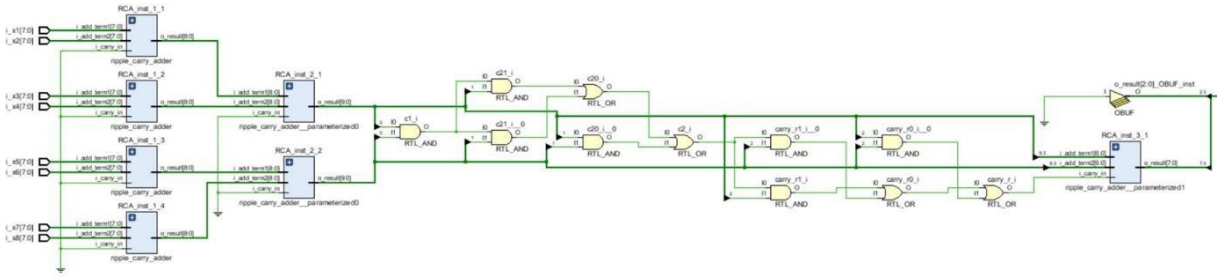
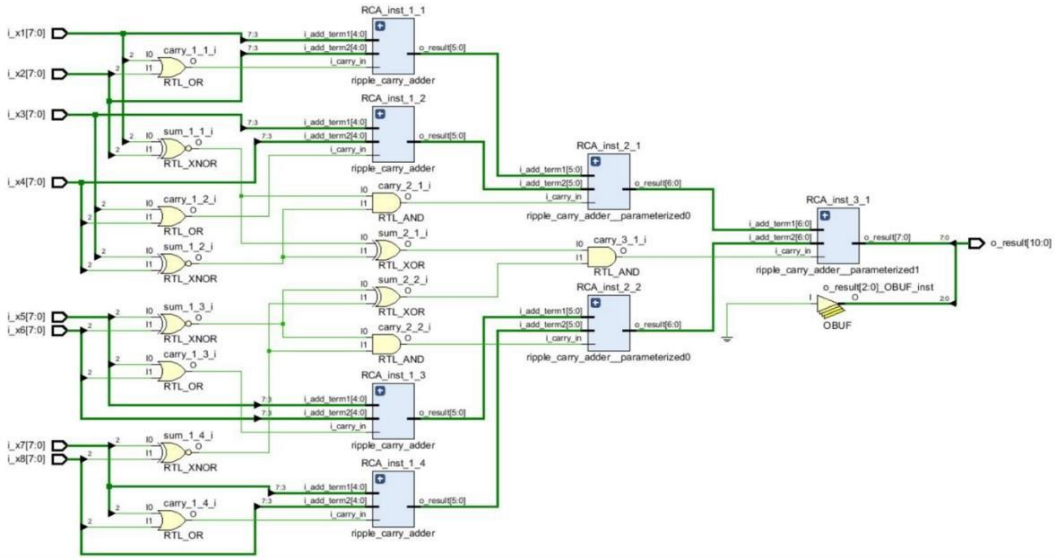


Figure.5 Schematic diagram of PT Area Report ITFX

Site Type	Used	Fixed	Available	Util%
Slice LUTs*	63	0	303600	0.02
LUT as Logic	63	0	303600	0.02
LUT as Memory	0	0	130800	0.00
Slice Registers	0	0	607200	0.00
Register as Flip Flop	0	0	607200	0.00
Register as Latch	0	0	607200	0.00
F7 Muxes	0	0	151800	0.00
F8 Muxes	0	0	75900	0.00

Ref Name	Used	Functional Category
IBUF	48	IO
LUT6	31	LUT
LUT4	17	LUT
LUT5	16	LUT
OBUF	11	IO
LUT2	9	LUT
LUT3	8	LUT

Figure.6 Schematic diagram of ITFX Error Analysis



The results of 8x8_FX_AT_PT	
The total data analysed in diffrence is	10000
The total data analysed in golden is	10000
The summation of diffrence is	35107
The summation of diffrence/golden is	36.017642189989
The AED(Average Error Difference) is	3.5107000000019
The MED(Maximum Error Distance)	7
The AAC is	99.6398235781007

The results of 8x8_ITFX_AT	
The total data analysed in diffrence is	10000
The total data analysed in golden is	10000
The summation of diffrence is	33617
The summation of diffrence/golden is	34.6450430502611
The AED(Average Error Difference) is	3.36170000000017
The MED(Maximum Error Distance)	14
The AAC is	99.6535495694969

ADVANTAGES

Faster processing: An efficient adder-tree design can process large amounts of data much faster than a less efficient design, as it reduces the number of stages required to complete the addition.

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 20, Issuse.1, March 2024

Reduced area: An efficient adder-tree design requires fewer logic elements and interconnects than a less efficient design, resulting in a smaller and more compact circuit that saves on space.
Lower power consumption: An efficient adder-tree design typically requires less switching activity, resulting in lower power consumption.
Improved timing performance: An efficient adder-tree design can achieve better timing performance, reducing the risk of timing violations and improving the overall performanceof the circuit.
Better accuracy: An efficient adder-tree design can reduce the likelihood of errors due to rounding or truncation, resulting in more accurate results.

APPLICATIONS

Fast Fourier Transform (FFT): An FFT is a common algorithm used in signal processing and can benefit from an efficient adder-tree design to reduce processing time.
Digital filters: Digital filters are used in various applications, including audio and image processing, and require efficient adder-tree designs to perform calculations quickly.
Image and video compression: Compression algorithms such as JPEG and MPEG use adder-trees extensively and benefit from efficient designs to reduce processing time and improve compression ratios.
Cryptography: Encryption and decryption algorithms require significant arithmetic operations, which can benefit from an efficient adder-tree design to speed up processing time and reduce power consumption.
High-speed computing: High-performance computing applications such as scientific simulations and modelling require fast and efficient arithmetic operations, making an efficient adder-tree design a desirable choice.

CONCLUSION

In this project, a novel scheme is presented to obtain fixed width AT design using truncated input.A probabilistic approach is presented to compensate the truncation error. Based on the proposed scheme, two separate fixed-width AT designs are derived. Both the proposed designs offer a substantial amount of area and CPD saving over the existing fixed-width AT designs.

ISSN 2319-5991 www.ijerst.com
Vol. 20, Issuse.1, March 2024

Forvectorizes 8 and 16, the proposed ITFX-AT offers ADP saving for word-length sizes (8; 12; 16), respectively, and calculates the output almost with the same accuracy as the post-truncated fixed- width AT which has the highest accuracy among the existing fixed-width AT.

FUTURE SCOPE

Future application of fixed width adder trees can be implementation of the Fast Fourier Transform (FFT), which is a widely used algorithm for computing the Discrete Fourier Transform (DFT) of asequence of complex numbers.

The FFT algorithm involves performing a large number of additions and multiplications on the input sequence, which can be efficiently implemented using fixed width adder trees. The input sequence is typically divided into smaller sub-sequences, which are processed recursively using abutterfly structure.

The butterfly structure involves performing two additions and two multiplications on pairs of inputsamples, which can be implemented using a fixed width adder tree. The output of the butterfly structure is then combined with the outputs of other butterfly structures using another fixed widthadder tree.

REFERENCES

[1] B. K. Mohanty and V. Tiwari, ”Modified probabilistic estimation bias formulation for hardware efficient fixed-width Booth multiplier”, Circuits, Systems and Signal Processing, Springer, vol.33, no.12, pp. 3981–3994, Dec., 2014,

[2] H. R. Mahdiani, A. Ahmadi, S. M. Fakhraie and C. Lucas, ”Bioinspired imprecise computational blocks for efficient VLSI implementation of soft-computing applications”, IEEE Transactions on Circuits and Systems-I, Regular Papers, vol. 57, no. 4, pp. 850–862, Apr.2010.

[3] V. Gupta, D. Mahapatra, A. Raghunathan, and K. Roy, ”Low-power digital signal processing using approximate adders”, IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, vol. 32,

no. 1, pp. 124–137, Jan.2013.

[4] J. Liang, J. Han and F. Lombardi, ”New metrics for the reliability of approximation and probabilistic adders”, IEEE Transactions on Computers, vol. 62, no. 9, pp. 1760–1771,

ISSN 2319-5991 www.ijerst.com
Vol. 20, Issuse.1, March 2024

Sept.2013.

[5] H. Jiang, J. Han, F. Qiao and F. Lambardi, ”Approximate radix-8 Booth multipliers for low- power and high-performance operations”, IEEE Transactions on Computers, vol. 65, no. 8, pp. 2638–2644, Aug.2016.

[6] Y. Pan and P. K. Meher, ”Bit-level optimization of adder-trees for multiplie constant multiplications for effcient FIR filter implementation”, IEEE Transactions on Circuits and Systems-I, Regular Papers, vol. 61,
no. 2, pp. 455–462, Feb.2014.

[7] R. O. Julio, L. B. Soares, E. A. C. Costa and S. Bampi, ”Energy-efficient Gaussian filter for image processing using approximate adder”, In. Proc. International Conference on Electronics, Circuits and Systems, 2015, pp. 450-453, 2015