iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** VI **Month of publication:** June 2026

DOI: https://doi.org/10.22214/ijraset.2026.83812

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VI June 2026- Available at www.ijraset.com

Area-Delay Efficient Hybrid Ling–Kogge–Stone Adder for High-Speed Processing

AP Harini Sree[1], D. Srinivasulu Reddy[2]
[1]*Dept. of ECE, Sri Venkateswara College of Engineering, Tirupati, Andhra Pradesh, India*
[2]*Professor, Dept.of. ECE, Sri Venkateswara College of Engineering Tirupati, Andhra Pradesh, India*

Abstract: ***The adder is a crucial component of the Arithmetic Logic Unit (ALU) in contemporary high-performance VLSI computers. and frequently sets a processor's maximum clock frequency. This project demonstrates the creation and implementation of a 32-bit Hybrid Ling–Kogge–Stone (LKS) adder that combines the high-speed parallel prefix structure of the Kogge–Stone adder with the lower logic cost of Ling's carry formulation. Performance is negatively impacted by increasing fan-in and logic depth in the first carry-propagate stage of conventional Kogge-Stone adders. The suggested architecture effectively removes one logic step from the critical route and streamlines carry generation by utilizing Ling's transformation. Verilog HDL is used to model the Hybrid LKS adder, and the Xilinx Vivado Design Suite is used to synthesis it. Improved signal integrity and high-speed operation are ensured by using a five-level parallel prefix tree to achieve an delay while retaining a maximum fan-out of two at each node. Compared to traditional RippleCarry and ordinary Carry LookAhead adders, post-synthesis results show a notable decrease in Total Combinational Path Delay (TCPD). Additionally, a thorough examination of Area, Power, and Delay (PPA) validates the effectiveness of the suggested design. The Hybrid LKS adder is ideal for high-performance RISC-V execution units and cryptographic accelerator applications because of its exceptional speed performance and scalable architecture.***

Keywords: Ling adder, Kogge–Stone adder, parallel prefix adder, Verilog HDL, Xilinx Vivado, VLSI design, high-speed arithmetic.

I. INTRODUCTION

Parallel prefix adders are crucial for performing fast arithmetic operations, which are becoming more and more popular in VLSI, a class of digital circuits and systems.

They are quite efficient in terms of power usage, area, and speed. These adders, which are frequently used in VLSI circuits like microprocessors and digital signal processors, among others, were developed using design techniques including Brent-Kung, Kogge-Stone. Parallel prefix adders are favored over accurate parallel prefix adders in certain scenarios when precision is not essential and a certain amount of error can be tolerated. These adders are designed to provide output more quickly while consuming less power and space than precise parallel prefix adders.

II. LITERATURE REVIEW

A. Introduction

Parallel prefix adders are important parts of VLSI processing chips because they help add two large binary numbers and perform addition. The main goals of this type of adder are to use less space, work faster, and make fewer mistakes. Improvements have been made to make the process more effective and time-saving in order to address the difficulties in the exact the PPA field. The current PPA adder was made with the aim of reducing area, time, and error rates.

However, the PPA was created to better improve the error rate and reduce delay. This new approach tries to cut down time even more while keeping the mistake rate low, leading to better results. The results of these nearly correct calculations have been checked and show that they can be used in real-world image processing applications.

Computing, sometimes called AxC, is a new design method that uses the built-in ability of different applications to tolerate errors to make computing more efficient across the whole system. To assist balance advancements, AxC adds accuracy, or the quality of the output, as a new, obvious component of the design. Using this method can greatly reduce the area of VLSI circuits and the energy they use for computation.

B. Existing Ppa

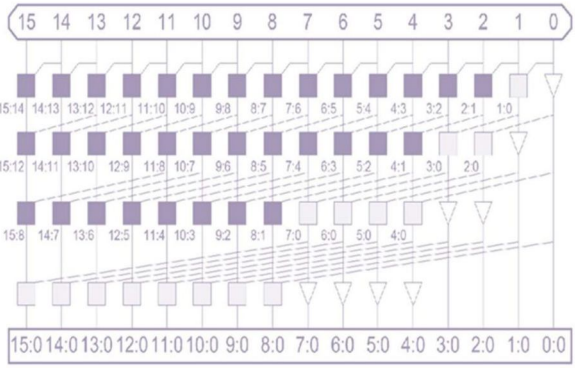


Fig 2.2.1: Exact 16-Bit Kogge- Stone Adder

The Kogge- Stone Adder is a carry look-ahead adder with a parallel prefix shape. It was originally created by Harold S. Stone and Peter M. Kogge, who published it in 1973. Because it produces carry signals quickly and performs satisfactorily overall in VLSI implementations, the KS adder is fast adder layout. Because it significantly reduces the critical direction, the Kogge- Stone adder is frequently employed in 32-bit and 64-bit systems with high overall performance. The 16-bit Kogge -Stone adder construction, with fanout of 2 at each level, is shown in Figure 2.2.1. There are fewer levels with logical operational architecture that are highly complex. KS Adder is widely utilized in high-performance addition devices with reduced latency and area efficiency.

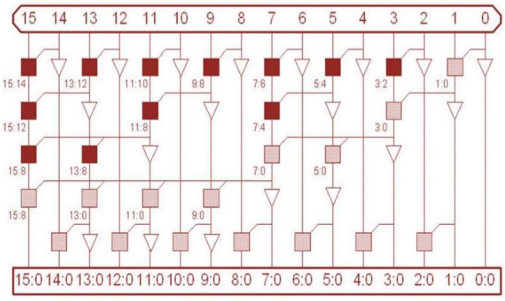


Fig 2.2.2 Exact 16-Bit Ladner Fischer Adder

In PPA, carries are created at the same time, and different tree structures vary in how they provide cells during intermediate steps, especially in the prefix stage. Cells are groups of logic gates that can take up a lot of space. By reducing the number of operators in stages, these cells can be optimized to take up less space, which reduces the overall area. One benefit of using adders or multipliers is that they can produce slightly incorrect results without affecting the overall computation. For our proposed work, we limit the number of cells in the prefix stage of the adder. Instead of using a single black cell for the final stage, we can use the same black cell for the last two stages, and since the result is not affected, this is shown in Fig. 2.2.3. The decision to increase from one stage to two or more stages depends on the number of bits in the architecture. In this design, each black cell at the top is used to handle the last two stages. This change can be made by modifying the VHDL module for simulation.

III. PROPOSED PPA

A. Proposed Kogge Stone Ppa

The Proposed Hybrid Kogge Stone PPA, shown in Figure 3.3, is a 32-bit adder that is divided into two phases. It comprises of one distinct Exact Kogge Stone PPA and one Structure. The 8-bit structure, which takes bits from LSB 0 to 7, of inputs A and B, produces sum output values from S_0 to S_7 using the logical OR operation in the final stage and carry output C_7. This carry output C_7 is used as carry input for the next stage of Exact Kogge Stone 24-bits PPA. The Exact Kogge Stone 24-bits PPA structure, taking bits from 8 to 31 of inputs A and B, generates a sum output with Exact values from S_8 to S_{31} and carry out of Cout. High-performance computing systems commonly employ the suggested Kogge-Stone PPA, notably in 32- and 64-bit architectures, because it considerably shortens the critical path.

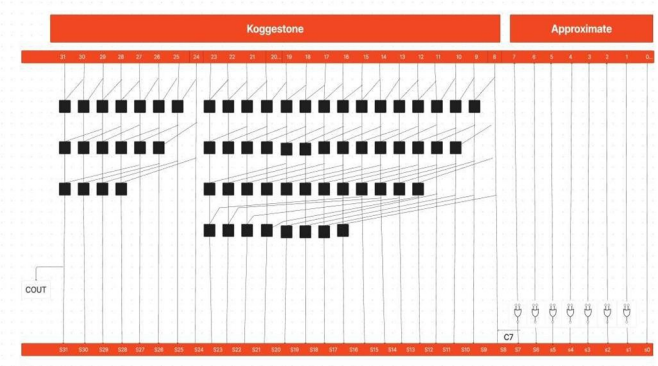


Fig 3.1.1: Architecture of Proposed Kogge Stone PPA

IV. PROPOSED HYBRID PPA

A. Proposed Hybrid PPA1

The Proposed Hybrid PPA1, shown in abve Figure, is a 32-bit adder that is divided into three phases. It comprises of two distinct Exact PPAs and one Structure. The 8-bit structure, which takes bits from LSB 0 to 7, of inputs A and B, produces sum output values from S0 to S7 using the logical OR operation in the final stage and carry output C7. This carry output C7 is used as carry input for the next stage of exact Kogge stone 12-bit adder. The Kogge stone 12-bit adder structure, taking bits from 8 to 19 of inputs A and B, generates a sum output with Exact values from S8 to S19 and carry out of C19. This carry out C19 is taken as input for the next exact Ladner Fischer 12-bit Adder. The Ladner Fischer 12-bit adder structure, taking bits from 20 to 31 of inputs A and B, generates a sum output with exact values from S20 to S31 and carry out of C_{out} at the end.

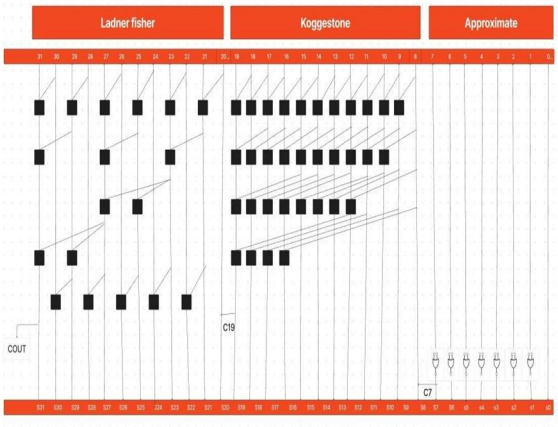


Fig 4.1: Proposed Hybrid PPA1

V. SMULATION RESULTS

Figures 6.1, 6.2, and 6.3 display the simulation results for the 32-bit Exact KS, AxKS, and Proposed AxKS adders, respectively. These figures show that the sum outputs for the Exact KS, AxKS, and Proposed AxKS adders are 42412741H, 3D308639H, and 3D30D6BDH, respectively. The two input numbers utilized are 8AB87B67H and B788ABDAH. The current and suggested AxKS adders have error rates of 0, 1.571725781914016, and 1.571344539474199, respectively. The latency and area in terms of LUTs for the current and suggested AxKS adders for 8-bit, 16-bit, and 32-bit designs are quantitatively analyzed in Table 6.4. Table 6.4 shows that the suggested AxKS 32-bit adder has a lower delay and uses less space than the current one.

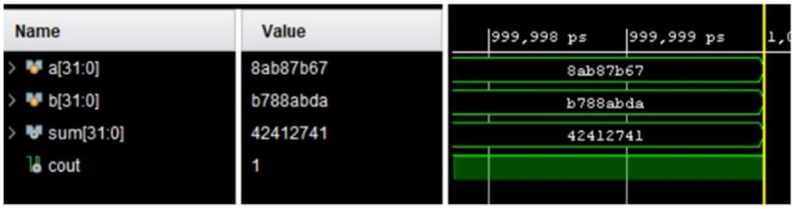


Fig. 5.1.1 Simulation result of 32-bit Exact KS Adder

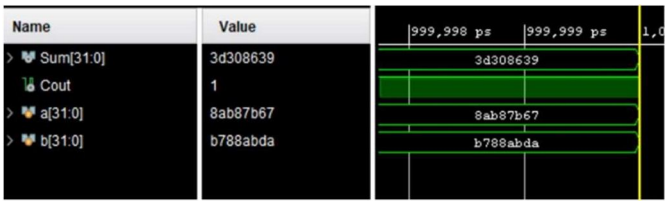


Fig. 5.1.2 Simulation result of 32-bit AxKS Adder

Fig. 5.1.3 Simulation result of 32-bit Proposed AxKS Adder

Table 5.1.4Analysis of Area and Delay of existing and Proposed AxKS Adder

No. of bits	Exact KS Adder LUT [2]	Exact Adder [2] Delay (ns)	AxKS Adder LUT [2]	AxKS Adder Delay (ns) [2]	Proposed AxKS Adder LUT [2]	Proposed AxKS Adder Delay(ns)[2]
8	1389	8.095	1386	8.024	1386	8.022
16	1525	9.482	1453	9.067	1453	9.066
32	1659	17.460	1654	7.086	1654	7.084

A. Xilinx Vivado Simulation and Synthesis Results

Table 5.2.1 Performance Analysis of Proposed PPA

S. No	Types of Proposed PPA (24+8) bits	LUT	Delay (ns)
1	Proposed AxBK PPA	1663	7.532
2	Proposed AxKS PPA	1676	6.975
3	Proposed AxLF PPA	1664	7.538
4	Proposed AxSK PPA	1658	8.236

Table 5.2.1 presents the performance analysis of various types of 32-bit Proposed PPA. Among them, the Kogge Stone adder demonstrates superior speed and reduced delay compared to the others. Additionally, the Brent Kung PPA exhibits a smaller area when compared to other types of Proposed PPA.

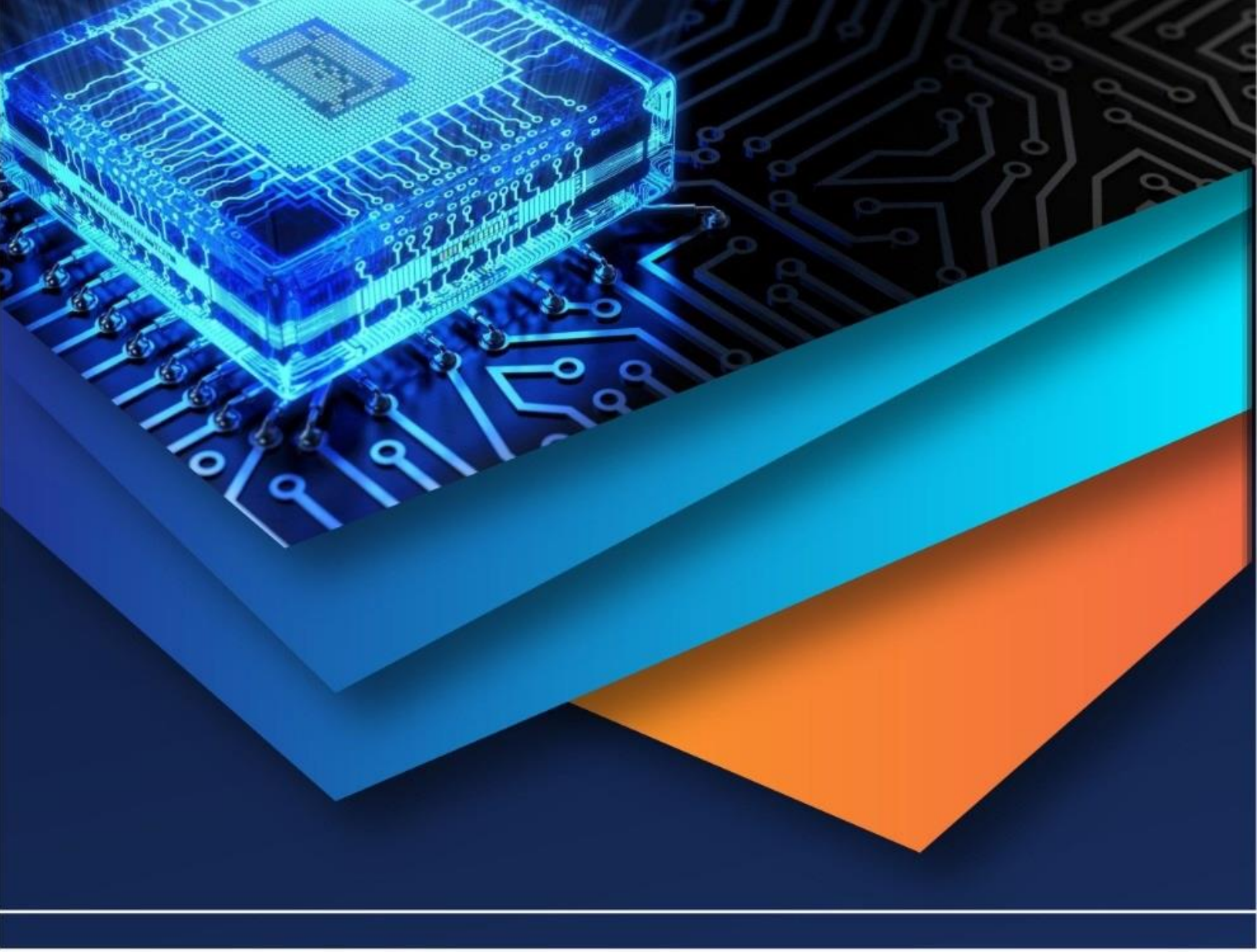
VI. CONCLUSION

This work has looked at, simulated, and assessed several approximation adders. The results of the experiment were subsequently put into practice. In order to achieve high PSNR values and reduced area and delay, new parallel prefix adders were developed and installed for this project.

This addition method is very quick and produces better results when applied to huge numbers in a much shorter amount of time. The suggested Ax parallel Prefix adders show promise for use in image processing, ALU units, and digital signal processing. Applications for picture contrast and enhancement are used in this study.

REFERENCES

[1] R. P. Brent and H. T. Kung, “A regular layout for parallel adders,” Compute. Eng., vol. C-31, IEEE, Trans., vol. C-31, no. 3, pp. 260-264, 1982 [doi:10.1109/TC.1982.1675982].
[2] Dr.M. Sarada Musala, Satish Kanapala, Sudhakar Reddy Dantla, Prudhvi Tummala, “ Brent Kung Adder for Image Processing Applications.”, Proceedings of 2023, Springer.
[3] Dr.M. Sarada Musala, Satish Kanapala, Sudhakar Reddy Dantla, Prudhvi Tummala, “High Speed Area Efficient Kogge Stone Adder.”, IEEE Proceedings of 2023.
[4] Y. Choi, “Parallel prefix adder design” in Proc. 17th IEEE Symposium on Computer Arithmetic, Jun. 27 2005, pp. 90-98.
[5] "A spanning tree carry look-ahead adder," T. Lynch and E. E. Swartzlander, IEEE Trans. Compute., vol. 41, no. 8, pp. 931-939, Aug. 1992 [doi:10.1109/12.156535].
[6] "Navya and Pramod," performed by B. C. Geetha and D. M. Lohitha.
[7] M. M. Azevedo da Rosa et al., "AxPPA: parallel prefix adders," IEEE Trans. Very Large Scale Integrated; Anal. Parallel Prefix Adder Datapath Design, IEEE Xplore Compliant; Part Number: CFP18BAC-ART; ISBN: 978-1-5386-1974-2, p. V. (VLSI) Syst., vol. 31, no. 1, January 2023.
[8] P. da Costa et al., "Improved multipliers for single-precision floating-point hardware design," in Proc. IEEE, 13th Latin American Symp. Circuits Syst. (LASCAS), March 2022, pp. 1-4.
[9] Megha Talsania and Eugene John, "Comparative analysis of parallel prefix adders."
[10] "Parallel prefix adder design," Andrew Beaumont-Smith, IEEE Trans. Compute., 2001.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY