iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 3 **Issue:** VI **Month of publication:** June 2015

DOI:

www.ijraset.com

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

Volume 3 Issue VI, June 2015
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Comparison of Various 32-Bit Parallel Prefix Adders

Mallapu Santhosh Kumar[1], K.Dhanunjaya(Ph.d)[2]
Department of Electronics and communication Engineering
Audisankara College of Engineering & Technology, Gudur, (Autonomous)

Abstract— In Very Large Scale Integration (VLSI) designs, Parallel prefix adders (PPA) have the higher delay performance. This paper investigates four varieties of PPA's (Kogge Stone Adder (KSA), Spanning Tree Adder (STA), Brent Kung Adder (BKA) and Sparse Kogge Stone Adder (SKA) and Ladner-Fisher adder.These adders are implemented in verilog Hardware Description Language (HDL) using Xilinx Integrated software system environment (ISE) 14.3 designs Suite. These designs area unit enforced in Xilinx Virtex five Field Programmable Gate Arrays (FPGA) and all these adder's delay, power and area are investigated and compared finally.
Key words —parallel prefix adders; carry tree adders; FPGA; delay; power.

I. INTRODUCTION

The binary addition is that the basic arithmetic operation in digital circuits and it became essential in most of the digital systems together with Arithmetic and Logic Unit (ALU), microprocessors and Digital Signal processing (DSP). At present, the analysis continues on increasing the adder's delay performance. In several practical applications like mobile and telecommunications, the Speed and power performance improved in FPGAs is healthier than microprocessor and DSP's based mostly solutions. in addition, power is additionally a vital side in growing trend of mobile electronics, that makes large-scale use of DSP functions. attributable to the Programmability, structure of configurable logic blocks (CLB) and programming interconnects in FPGAs, Parallel prefix adders have higher performance. The delays of the adders area unit are [1].
In this paper, above mentioned PPA's and RCA and CSA are implemented and characterised on a Xilinx virtex five FPGA. Finally, delay, power and space for the designed adders are conferred and compared.

II. DRAWBACKS OF RIPPLE CARRY AND CARRY LOOK AHEAD ADDER

In figure1, the first sum bit should wait until input carry is given, the second sum bit should wait until previous carry is propagated and so on.
Finally the output sum should wait until all previous carries are generated. So it results in delay.

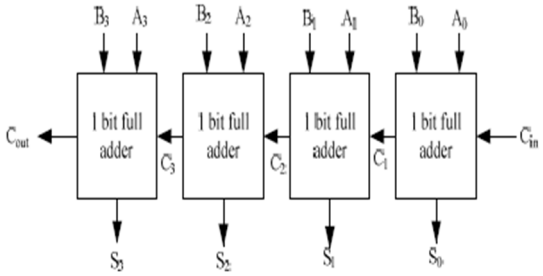


Fig. 1. 4 bit ripple carry adder

In order to reduce the delay in RCA (or) to propagate the carry in advance, we go for carry look ahead adder .Basically this adder works on two operations called propagate and generate The propagate and generate equations are given by

$$P_i = A_i \oplus B_i \quad (1)$$

$$G_i = A_i . B_i \quad (2)$$

For 4-bit CLA,the propagated carry equations are

$$C_1=G_0+P_0C_0 \quad (3)$$

$$C_2=G_1+P_1G_0+P_1P_0C_0 \quad (4)$$

$$C_3=G_0+P_2G_1+P_2P_1G_0+P_2P_1P_0C_0 \quad (5)$$

Equations (3),(4),(5) and (6) are observed that, the carry complexity increases by increasing the adder bit width. So designing higher bit CLA becomes complexity. In this way, for the higher bit of CLA's, the carry complexity increases by increasing the width of the adder. So results in bounded fan-in rather than unbounded fan-in, when designing wide width adders. In order to compute the carries in advance without delay and complexity, there is a concept called Parallel prefix approach.

III. DIFFERENCE BETWEEN PARALLEL-PREFIX ADDERS AND OTHERS

The PPA's pre-computes generate and propagate signals are presented in [2]. Using the fundamental carry operator (fco), these computed signals are combined in [3].The fundamental carry operator is denoted by the symbol *"o "*,

$$(g_{L,}\, p_L)0(g_{R,}\, p_R)=(g_L+p_L.g_R,p_L,p_R) \quad (7)$$

For example, 4 bit CLA carry equation is given by

$$C_4=(g_4,p_4)\ 0\ [(g_3,p_3)\ 0\ [(g_4,p_4)\ 0\ (g_3,p_3)]]$$

For example, 4 bit PPA carry equation is given by

$$C_4=(g_4,p_4)\ 0\ [(g_3,p_3)\ 0\ [(g_4,p_4)\ 0\ (g_3,p_3)]]$$

Equations (8) and (9) are observed that, the carry look ahead adder takes 3 steps to generate the carry, but the bit PPA takes 2 steps to generate the carry

IV. PARALLEL-PREFIX ADDER STRUCTURE

Parallel-prefix structures are found to be common in high performance adders because of the delay is logarithmically proportional to the adder width [2]
PPA's basically consists of 3 stages

The Parallel-Prefix Structure is shown in figure 2.

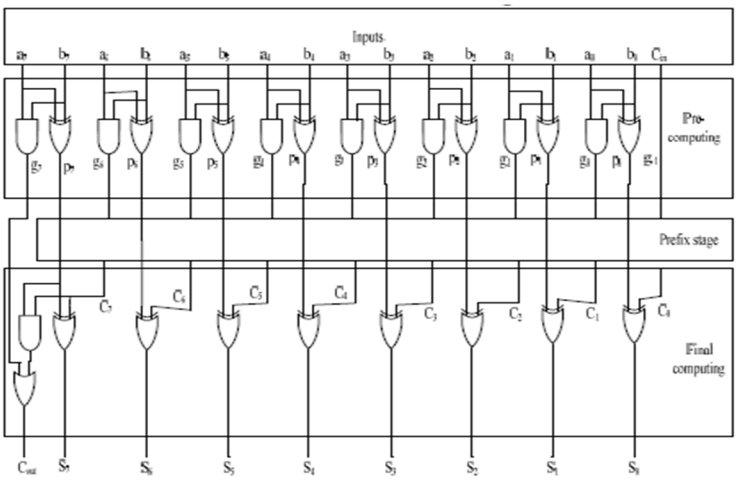


Fig. 2. Parallel-Prefix Structure with carry save notation

A. Pre computation

In pre computation stage, propagates and generates are computed for the given inputs using the given equations (1) and (2).

B. Prefix stage

In the prefix stage, group generate/propagate signals are computed at each bit using the given equations. The black cell(BC) generates the ordered pair in equation (7), the gray cell(GC) generates only left signal, following [2].

$$G_{i:k} = G_{i:j} + P_{i:j}. G_{j-1:k} \quad (10)$$

$$P_{i:k} = P_{i:j}. P_{j-1:k} \quad (11)$$

More practically, the equations (10) and (11) can be expressed using a symbol "o "denoted by Brent and Kung. Its function is exactly the same as that of a black cell i.e.

$$G_{i:k} :P_{i:k}=(G_{i:j} :P_{i:j}) O (G_{j-1:k}: P_{j-1:k}) \quad (12)$$

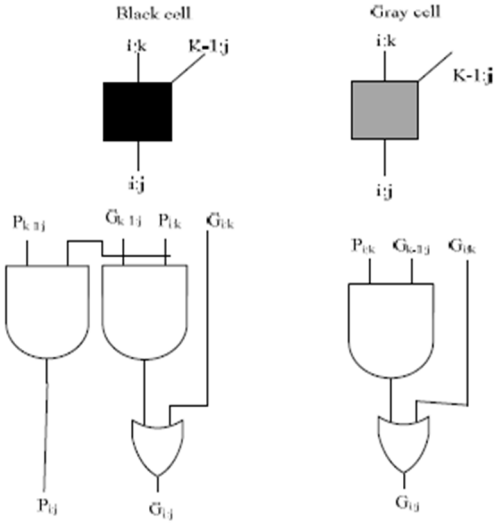


Fig. 3. Black and Gray Cell logic Definitions

The "o" operation will help make the rules of building prefix structures.

C. Final computation

In the final computation, the sum and carryout are the final output.

$$S_i = P_i. G_{i-1:-1} \quad (13)$$

$$C_{out}=G_{n:-1} \quad (14)$$

Where "-1" is that the position of carry-input. The generate/propagate signals is classified in numerous fashion to induce constant correct carries. based on different ways of grouping the generate/propagate signals, completely different prefix architectures is created. Figure three shows the definitions of cells that are utilized in prefix structures, as well as bc and gc. For analysis of assorted parallel prefix structures, see [2], [3] & [4].

The 16 bit SKA uses black cells and grey cells likewise as full adder blocks too. This adder computes the carries exploitation the BC's and GC's and terminates with four bit RCA's. completely it uses sixteen full adders.

The 16 bit SKA is shown in figure four.In this adder, initial the input bits (a, b) square measure reborn as propagate and generate (p, g). Then propagate and generate terms are given to BC's and GC's. The carries are propagated in advance using these cells. Later these are given to full adder blocks.

Another PPA is understood as STA is additionally tested [6]. just like the SKA, this adder conjointly terminates with a RCA. It

Volume 3 Issue VI, June 2015
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

conjointly uses the BC's and GC's and full adder blocks like SKA's however the distinction is that the interconnection between them [7].The 16 bit STA is shown within the below figure

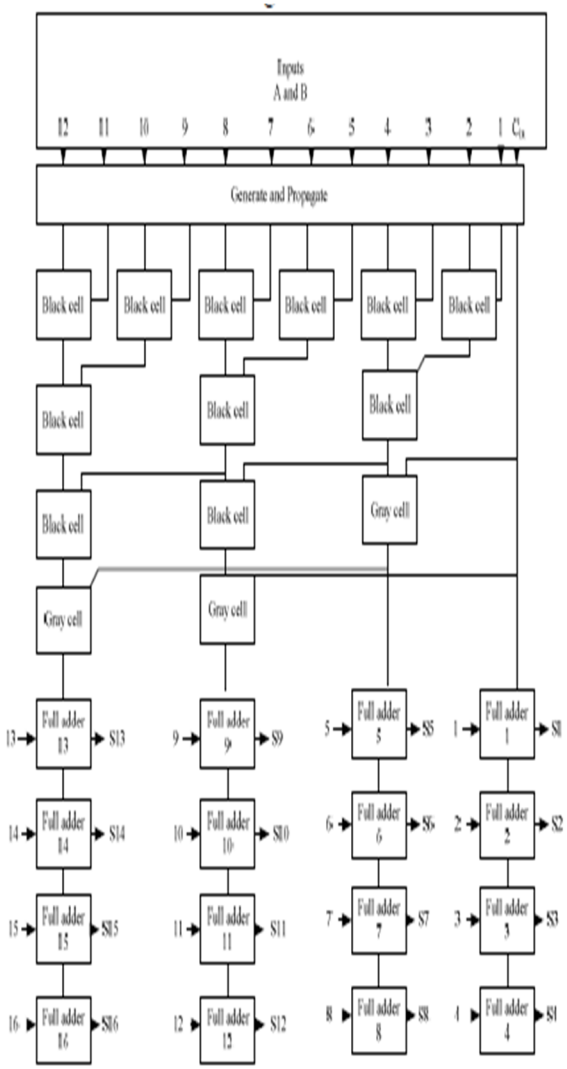


Fig. 4. 16 bit sparse kogge-Stone adder

KSA is another of prefix trees that use the fewest logic levels. A 16-bit KSA is shown in Figure six. The 16 bit kogge stone adder uses BC's and GC's and it won't use full adders. The 16 bit KSA uses thirty six BC's and fifteen GC's. And this adder entirely operates on generate and propagate blocks. that the delay is a smaller amount in comparison to the previous SKA and STA. The 16 bit KSA is shown in figure six.In this KSA, there aren't any full adder blocks like SKA and STA [5] & [6].

Another carry tree referred to as BKA that additionally uses BC's and GC's however but the KSA. therefore it takes less space to implement than KSA. The sixteen bit BKA uses fourteen BC's and eleven GC's however kogge stone uses thirty six BC's and fifteen GC's. therefore BKA has less architecture and occupies less space than KSA. The sixteen bit BKA is shown within the below figure 7.

Volume 3 Issue VI, June 2015
ISSN: 2321-9653

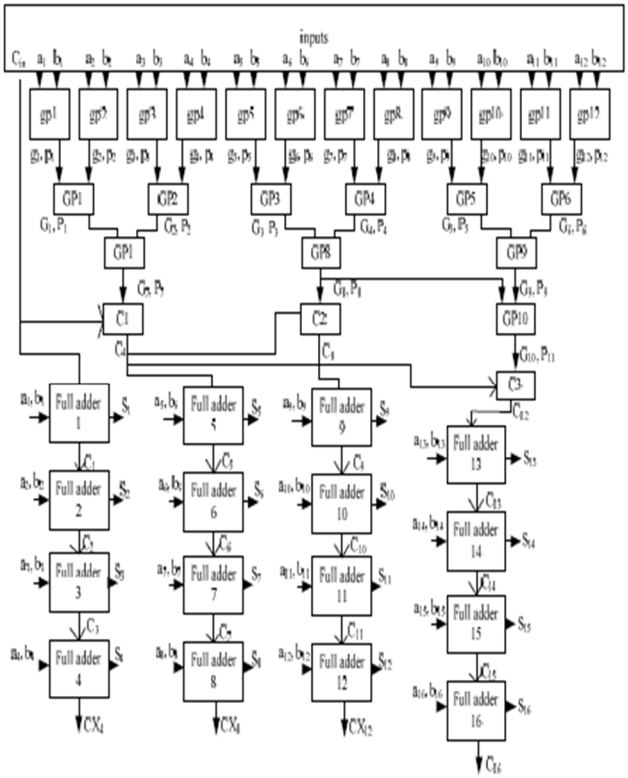


Fig. 5. 16 bit spanning tree adder

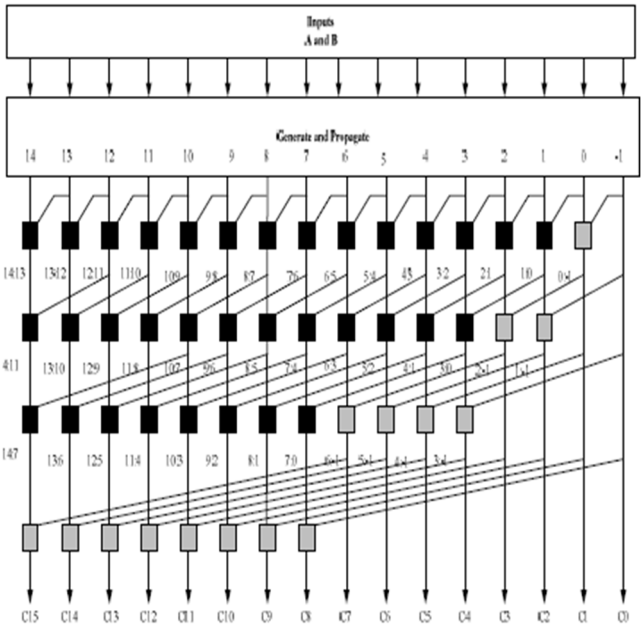


Fig. 6. 16 bit kogge stone adder

BKA occupies less area than the opposite three adders referred to as SKA, KSA, STA. This adder uses restricted variety of propagate and generate cells than the opposite three adders. It takes less space to implement than the KSA and has less wiring congestion. The operation of the 16 bit brent kung adder is given below [3]. This adder uses less BC's and GC's than kogge stone adder and has the higher delay performance

Volume 3 Issue VI, June 2015
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

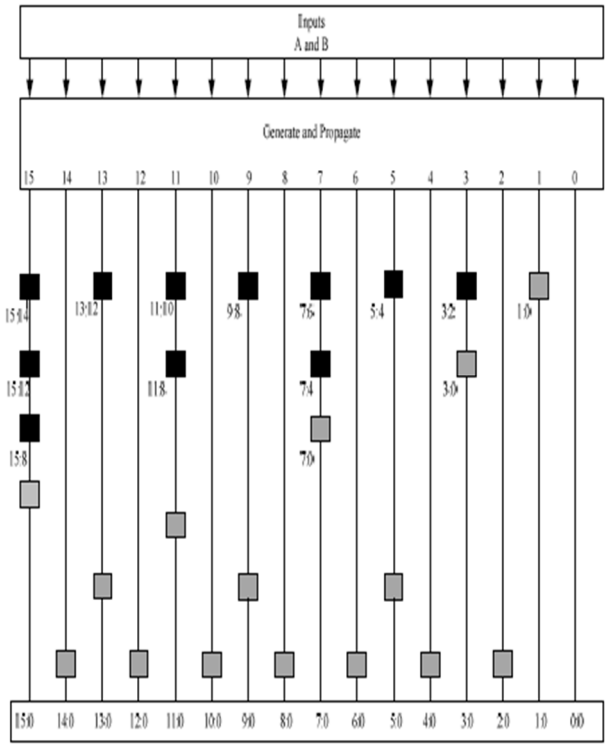


Fig. 7. 16 bit brent kung adder

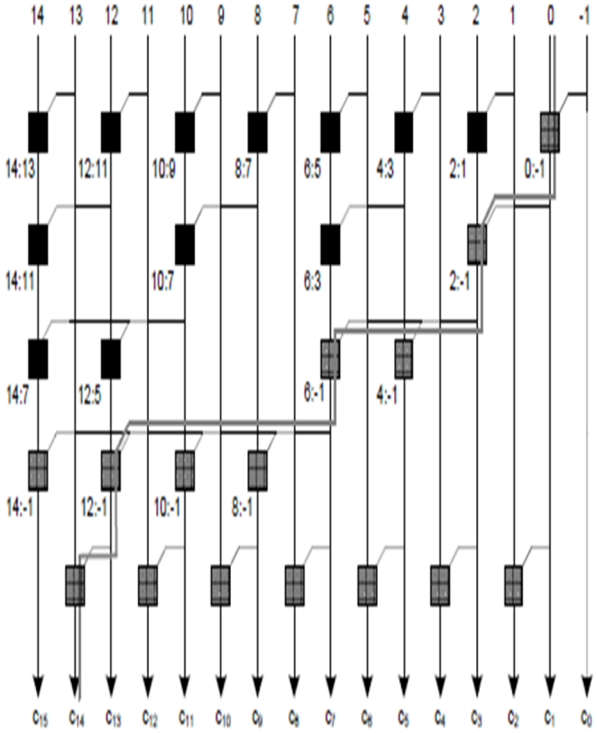


Fig8.16 bit Ladner fisher adder

Volume 3 Issue VI, June 2015
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

V. DISCUSSION OF RESULTS

The delays observed for adder designs from synthesis reports in Xilinx ISE 14.2 synthesis reports are shown in Figure9

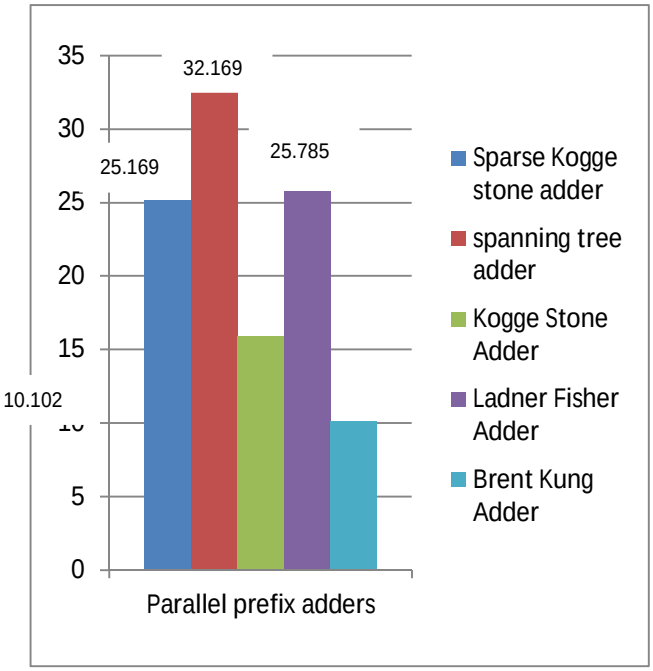


Fig. 9. Comparison in delays of various PPA's

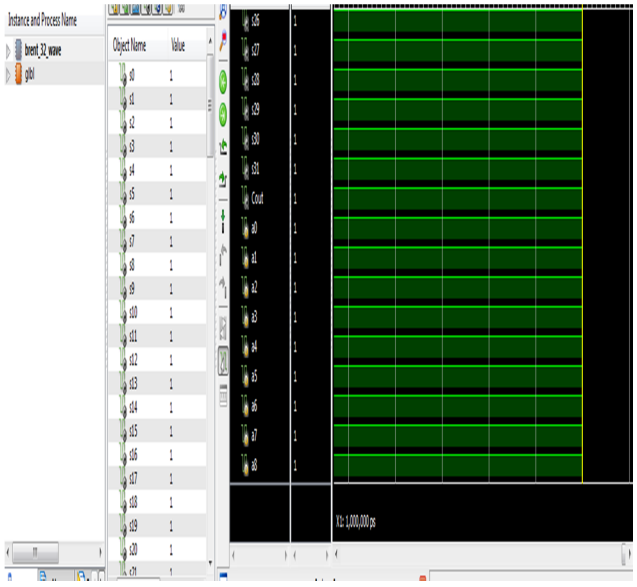


Fig. 10 simulation results of BKA

The delays observed for adder designs from synthesis reports in Xilinx ISE 14.2 synthesis reports and delays were investigated, compared and shown in figure10.

Volume 3 Issue VI, June 2015
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

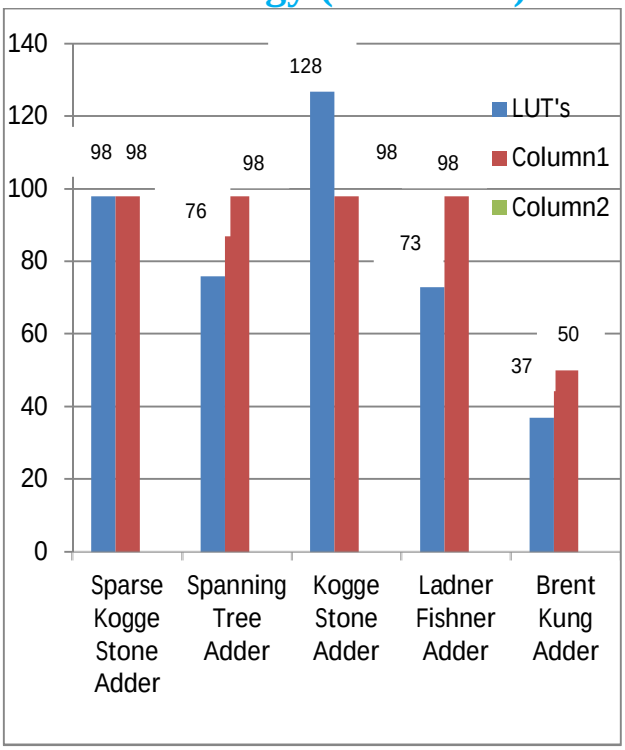


Fig. 10 Utilization of LUT's and IOB's

The area of the adder designs is measured in terms of look up tables (LUT) and input output blocks (IOB) taken for Xilinx virtex 5 FPGA is plotted in the figure 10.As per reference [1], ISE software doesn't give exact delay of the adders because it is not able to analyze the critical path over the adder. From the figure10, the CSA has more delay when compared to other adders. Out of all adders, RCA has less delay. SKA adder and BKA has about the same delay, where as KSA and STA has same delay. According to the synthesis reports, out of four parallel prefix adders, STA has better delay

TABLE I. COMPARISON OF DELAY, POWER AND AREA FOR ADDERS

S.No	ADDER NAME(32-Bit)	Xilinx ISE 14.2 Delay in ns	Utilization (LUT's,IOB's)
1	Sparse kogge Stone Adder	25.169	98,98
2	Spanning Tree Adder	32.186	76,98
3	Kogge Stone Adder	15.889	128,98
4	Ladner Fishner Adder	25.785	73,98
5	Brent Kung Adder	10.102	37,50

The Fourth column gives the area of the adder designs in terms of LUT's and IOB's. . Out of five PPA's, BKA has better delay and has taken less LUT's and IOB's. All mentioned adders have delay of approximately 1ns (nanoseconds).

VI. CONCLUSION

From the study of analysis done on area and Delay, we have concluded that Out of five PPA's, BKA has better delay and has taken less LUT's and IOB's but it has less operational speed. Out of those all PPA's Kogge Stone Adder has high operational speed.

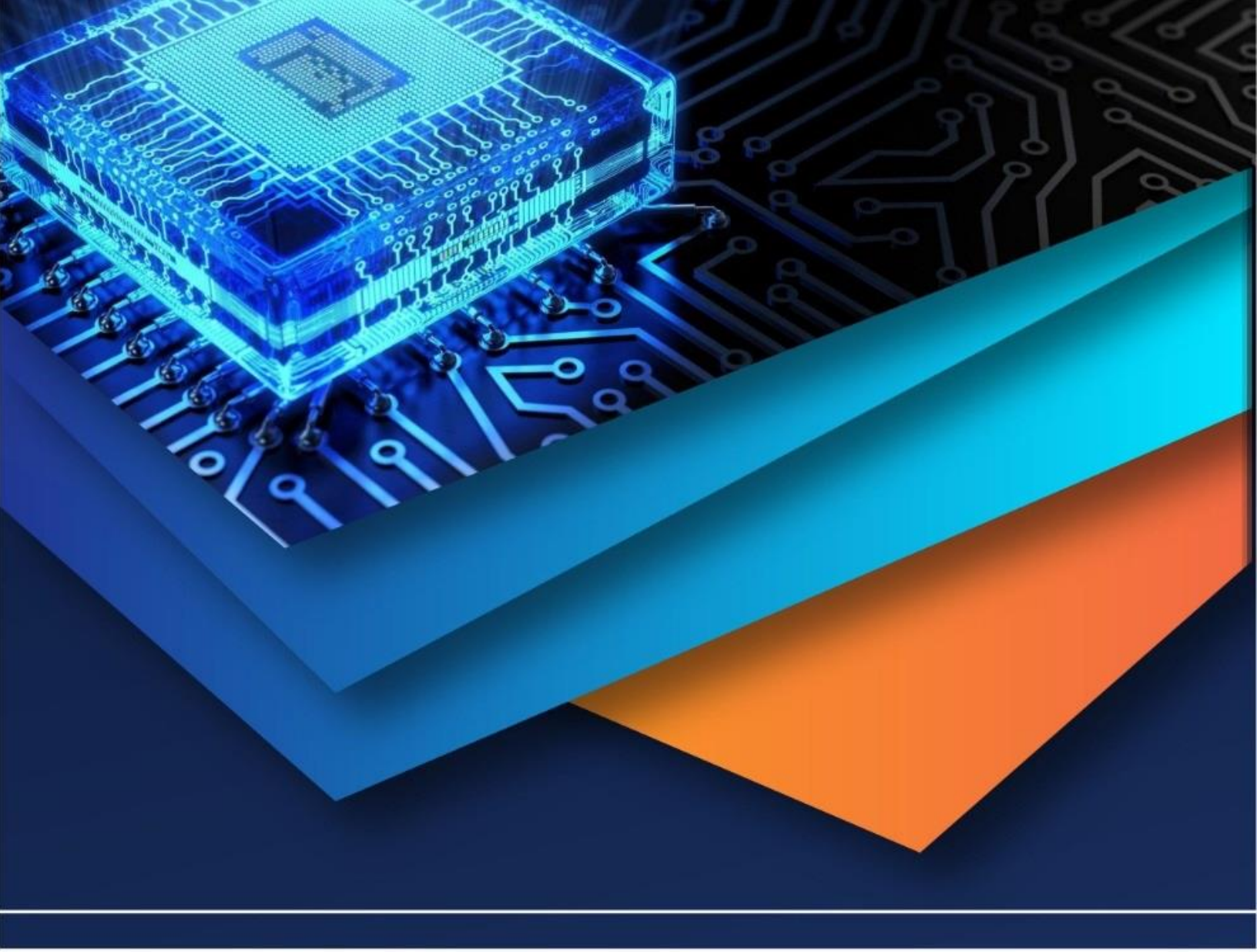
REFERENCES

[1] David H.K.Hoe, Chris Martinez and Sri Jyothsna Vundavalli", Design and Characterization of Parallel Prefix Adders using FPGAs", 2011IEEE 43rd Southeastern Symposium in pp. 168-172, 2011.
[2] N. H. E. Weste and D. Harris, CMOS VLSI Design, 4th edition,Pearson–Addison-Wesley, 2011.
[3] R. P. Brent and H. T. Kung, "A regular layout for parallel adders," IEEE Trans. Comput., vol. C-31, pp. 260-264, 1982.
[4] D. Harris, "A Taxonomy of Parallel Prefix Networks," in Proc. 37th Asilomar Conf. Signals Systems and Computers, pp. 2213–7, 2003.
[5] P. M. Kogge and H. S. Stone, "A Parallel Algorithm for the Efficient Solution of a General Class of Recurrence Equations," IEEE Trans. On Computers, Vol. C-22, No 8, August 1973.

AUTHOR'S PROFILE

Mallapu Santhosh Kumar received his B.Tech degree in Electronics and Communication Engineering from Sri Venkateswara College of Engineering, Tirupathi,Chitoor District, affiliated to JNTU Anantapur. He is currently pursuing M.Tech VLSI in Audisankara college of Engineering and Technology, Gudur(Autonomous), SPSR Nellore (Dist), affiliated to JNTU Anantapur.

K. Dhanunjaya received his B.Tech Degree in Electronics and Communication Engineering from G. Pulla Reddy Engineering College, Kurnool, AP in 1998. He received his M.Tech in ECE from Jawaharlal Nehru Technological University, Anantpur. He has 14 years teaching experience, presently working as Professor and Head of the department of ECE, Audisankara College of Engineering and Technology (Autonomous), Affiliated to JNTU, Anantpur. Andhra Pradesh, India



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY