Design and Implementation of Wallace Tree Multiplier Using Parallel Prefix Adders

A. M.Bhavani[1], B. L.Jahnavi[2], C. K.Vasanthi[3], D. K. Kavitha Rani[4], E. M. Sandhya Rani[5], F. K. Krissi Praneetha[6]

[1]*M.Bhavani, Assistant Professor,* Bapatla Women's Engineering College, *Andhra Pradesh*

[2,3,4,5,6] *L.Jahnavi, UG Student,* Bapatla Women's Engineering College, *Andhra Pradesh*

***Abstract*— Delays have become increasingly crucial in modern VLSI technology. In order to design the circuit, an efficient ALU is required. All logical computations, such as addition and multiplication, are handled by the ALU. Multiplication is used to reduce the number of partial products while increasing the speed of the operation. An adder is the fundamental building block of every digital design. Any adder should be able to satisfy in terms of speed and area. The area (number of LUTs), delay (ns), and number of bonded IOBs of the 16-bit Wallace tree multiplier and 16-bit Parallel prefix adders (Carry look-ahead adder, Kogge stone adder, and Brent Kung adder) are compared in this project. VLSI and simulation were used to design these, and Xilinx was used to synthesis them (ISE) 14.7.**

***Index Terms*— Arithmetic Logic Unit (ALU), Brent Kung adder (BKA), Carry look-ahead adder (CLA), Kogge stone adder (KSA), Parallel prefix adders (PPAs), Wallace tree multiplier**

I. INTRODUCTION

Multipliers and adders are critical components of the ALU, and their speed and delay should be high. In today's technology, a multiplier is used to efficiently build the circuit. A multiplier provides fast speed and minimizes latency. Multipliers are devices with a high surface area that multiply two integers. Multiplication is completed in three steps: partial product production, partial product addition, and final addition. The main goal is to provide fast speed with minimal delay, so an increase in speed results in a large area. There is a huge need for high-speed multiplication with minimal hardware. Wallace tree multiplier, proposed by an Australian computer scientist Chris Wallace, is a tree structure built using a multiplier.

For the creation of partial products, this multiplier incorporates a half adder, a full adder, and the multiplication of multiplier and multiplicand bits. High-speed multipliers are parallel multipliers. Parallel prefix adders (PPAs) are derived from the Carry look-ahead adder, and there are other variants of PPAs, such as the Kogge stone adder (KSA) and the Brent Kung adder (BKA), that speed up binary addition. A parallel-prefix adder performs well since the latency is proportional to the logarithm of the adder width, resulting in rapid and reliable prefix arithmetic computation. Pre-processing stage, Carry generation network, and Post-processing stage are the three steps of a parallel prefix adder. The first stage creates and propagates bits, while the second stage performs prefix operations. At the end of the process, the sum and carry are obtained.

II. WALLACE TREE MULTIPLIER

The Wallace tree multiplier is a high-speed multiplication device. A series of adders is used by the Wallace tree multiplier to generate the final outputs. It is a component of combinational logic circuits that multiplies two binary values and is built with full adders and half adders to efficiently perform the multiplication. The shift-add method is the most common way to accomplish multiplication. The time required to calculate products using the shift-add method grows as the number of bits in the operand increases. In digital design, there are a variety of multipliers to choose from. Because the Wallace tree multiplier performs better in terms of speed, it should take up less slices and LUTs.

Wallace tree multiplier mainly consists of three steps:
1. Generation of partial products
2. Partial product reduction
3. Final addition.

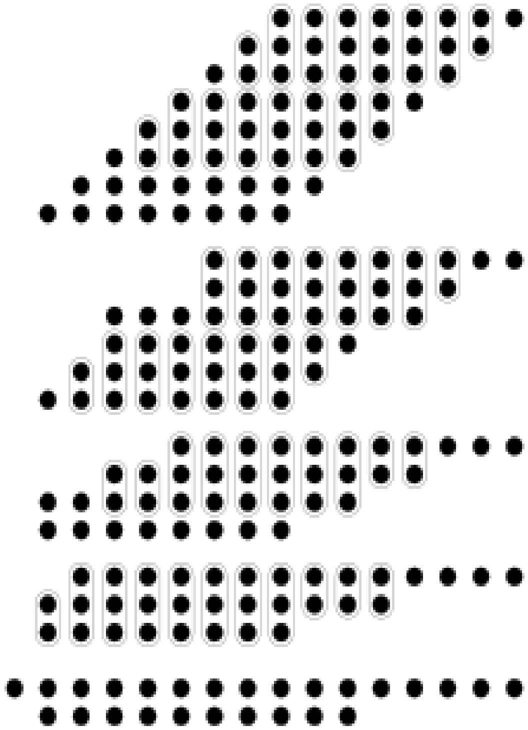


Figure 1: 16- bit Wallace tree multiplier

III. CARRY LOOK-AHEAD ADDER

In digital design, a carry look-ahead adder is a fast adder that performs binary addition using carry produces and carry propagate. It boosts performance by lowering the time it takes to identify carry bits. The carry signal must travel through each stage of this adder. The time it takes to propagate the carry is the limiting element for the ripple-carry adder. The carry look-ahead adder solves this problem by calculating carry signals ahead of time using input signals. The result gained is a decrease in carry propagation time due to the use of a complicated circuit. The two main components of the carry look-ahead adder's operation are: a) Determine whether each digit is propagating a carry bit from its correct place by calculating each digit.

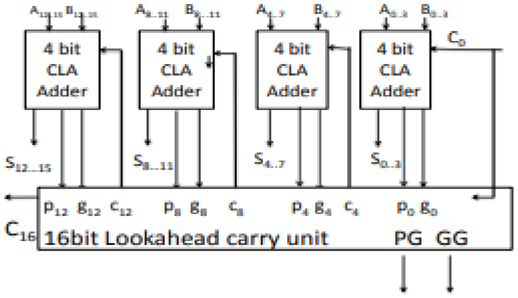


Figure 2: 16-bit Carry look-ahead adder

By layering the 4-bit adders, a 16-bit carry look-ahead adder can be built. We must change the Boolean phrase dealing with the full adder in order to comprehend how the carry look-ahead adder operates. Propagate P and create G in a full-adder:

Pi = Ai ⊕ Bi Carry propagate

Gi = Ai & Bi Carry generate

Notify that both propagate and generate signals are solely dependent on the input bits, and are hence valid after one gate delay. The following are the new expressions for the output sum and carryout: Si = Pi ⊕ Ci-1

Ci+1= Gi + Pi & Ci

The carry output Boolean function of each stage in a 4-bit carry look-ahead adder can be expressed as:

C1 = G0 + P0C0

C2 = G1 + P1C1 = G1 + P1 (G0 + P0C0) = G1 + P1G0 + P1P0C0

C3 = G2 + P2C2 = G2 + P2G1 + P2P1G0 + P2P1P0C0

C4 = G3 + P3C3 = G3 + P3G2 + P3P2G1 + P3P2P1G0 + P3P2P1P0C0.

IV. PARALLEL PREFIX ADDERS

Parallel prefix adders are used for high speed performance because of its delay is proportional to the adder width. There are many types of PPAs which are basically generated from carry look-ahead adder.

Types of Parallel Prefix adders:

- Sklansky Adder
- Kogge Stone Adder
- Brent Kung Adder
- Han Carlson Adder
- Ladner Fischer Adder

In this paper we chosen two parallel prefix adders, those are Kogge stone adder and Brent Kung adder.

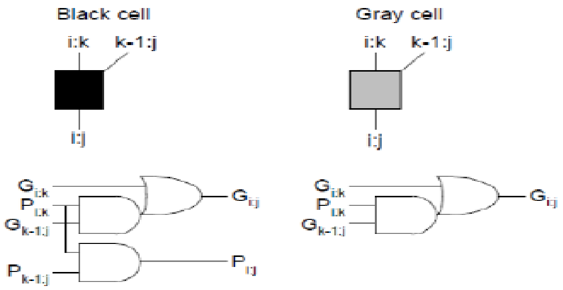
Three stages of parallel prefix adders are:

Pre-Processing Stage: In this stage for the given inputs there exists an operation of generate and propagate:

Pi = Ai ⊕ Bi Carry propagate

Gi = Ai & Bi Carry generate

Prefix Stage: A carry is generated for each bit and final cell present in each bit gives carry further generated by carry propagate and carry generate. Carry of last bit helps to produce sum of future bit.



The black cell consists of carry generate and propagate signals:

CPi: j = Pk-1: j and Pi: k

CGi: j = Gi: k + 1 or (Pi: k and Gk-1: j)

The gray cell consists of only carry generate signals:

CGi: j = Gi: k + 1 or (Pi: k and Gk-1: j)

Post-Processing Stage**:** This is the final step and is common to all adders of this family (carry look ahead). It involves computation of sum bits.

Si = Pi ⊕ Ci-1

Ci+1= Gi + Pi & Ci

V. KOGGE STONE ADDER

The Kogge Stone adder is a parallel prefix form carry look-ahead adder that is usually regarded as the quickest in the market for high-performance arithmetic circuits. It was created in 1973 by Peter M. Kogge and Harold S. Stone. In comparison to other adders, Kogge Stone adders have reduced latency for broad adders. Carry is generated in O (log n) time. The structure's delay is determined by log2n. There are [n (log2n)-n+1] computation nodes in this structure. The area is large, the logic depth is shallow, and the wiring complexity is high. It entails the action of parallel prefix adders in three stages:

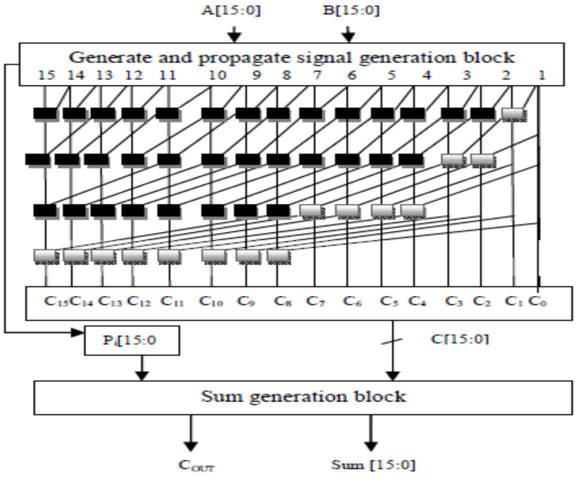

A[15:0]
B[15:0]
Generate and propagate signal generation block
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
$C_{15}C_{14}$ C_{13} C_{12} C_{11} C_{10} C_9 C_8 C_7 C_6 C_5 C_4 C_3 C_2 C_1 C_0
P[15:0
C[15:0]
Sum generation block
C_{OUT}
Sum [15:0]

Figure 3: 16-bit Kogge Stone adder

A 16-bit Kogge stone adder structure which consists of 34 black cells and 15 grey cells

.

VI. BRENT KUNG ADDER

We come up with Brent Kung adder to solve the shortcomings of Kogge stone adder. In 1982, Richard Peirce Brent and Hsaing Te Kung devised the Brent Kung adder, which is a parallel prefix adder. The parallel prefix adder is a high-performance carry tree adder that performs signal pre-computing for propagation and generation. Due to the complexity (log2n) delay through the carry path, the Parallel-prefix tree adders are more favorable in terms of speed as compared to other adders. It takes up the least amount of space and has the most depth. The number of cell of Brent Kung adder can be calculated by (2n-1)-log2n and the delay of the structure is (2n-1)-log2n can be used to calculate the number of cells in a Brent Kung adder, and the structure's latency is (2log2n-2).

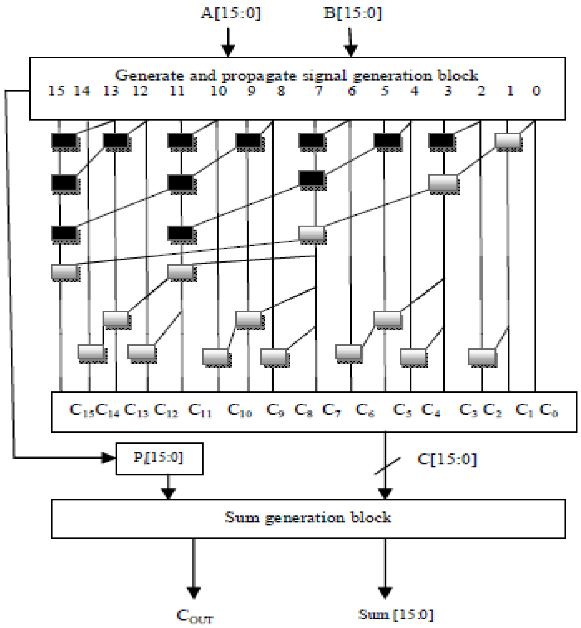

A[15:0]
B[15:0]
Generate and propagate signal generation block
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
$C_{15}C_{14}$ C_{13} C_{12} C_{11} C_{10} C_9 C_8 C_7 C_6 C_5 C_4 C_3 C_2 C_1 C_0
P[15:0]
C[15:0]
Sum generation block
C_{OUT}
Sum [15:0]

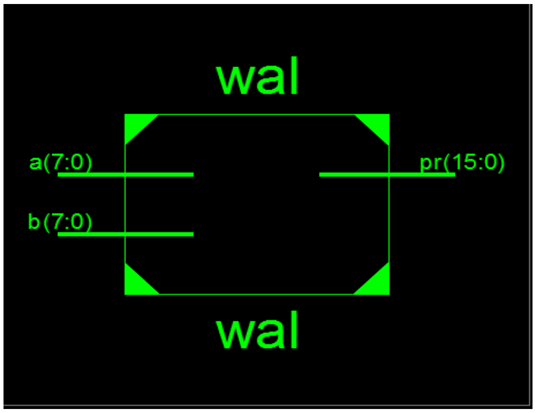
Figure 4: 16-bit Brent Kung adder

As the figure indicates 16-bit Brent Kung adder it consists of 12 black cells and 15 grey cells.

VII. SIMULATION RESULTS:

In the Xilinx ISE Design Suite version 14.7, various parallel prefix adders and Wallace tree multipliers are created and compared. All of the designs were created using the Xilinx Synthesis Tool and then tested using the Xilinx ISE simulator.

A. 16-bit Wallace tree multiplier



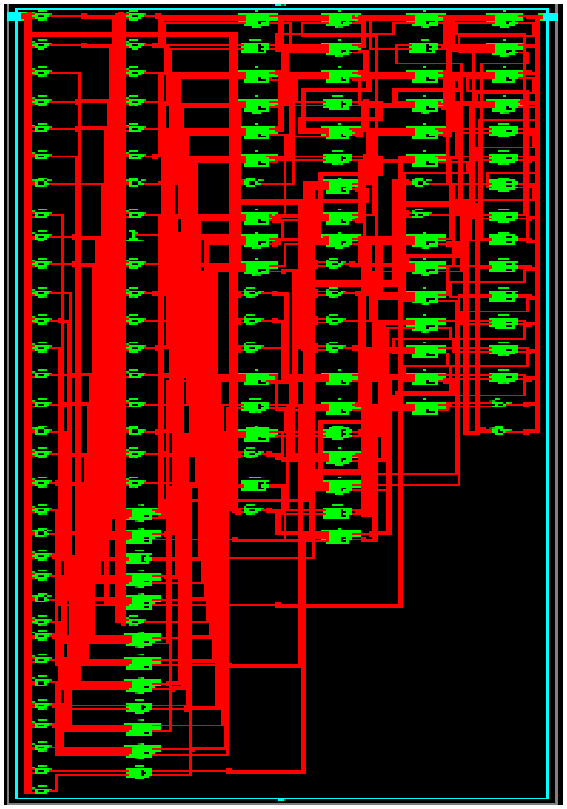


Figure (a) :RTL Schematic of 16-bit Wallace tree multiplier

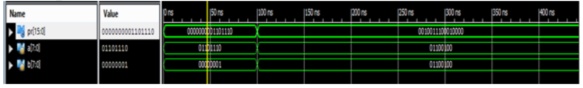


Figure (b): Simulation Wave forms for 16-bit Wallace tree multiplier

16-bit Wallace tree Multiplier is designed by using half adders and full adders, then the RTL schematic is shown in figure(a) and the number of LUT's are used are 120 and number of bonded IOB's are 32. Time delay for this design is 7.316ns and wave forms are shown in figure (b).

B. 16-bit Carry look-ahead adder

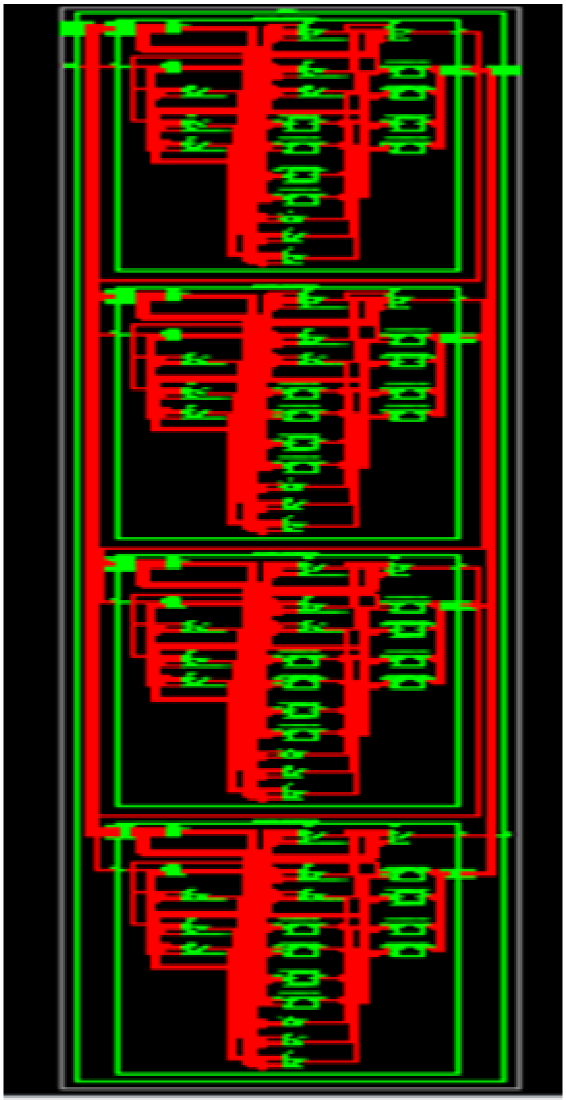


Figure (c): RTL Schematic of16-bit Carry look-ahead adder

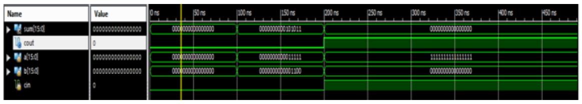


Figure (d): Simulation Wave forms for 16-bit Carry look-ahead adder

16-bit Carry-Look ahead adder is designed, then the RTL schematic is shown in figure (c) and the number of LUT's are used are 32 and number of bonded IOB's are 50. Time delay for this design is 21.378ns and wave forms are shown in figure (d)

C. 16-bit Kogge Stone adder

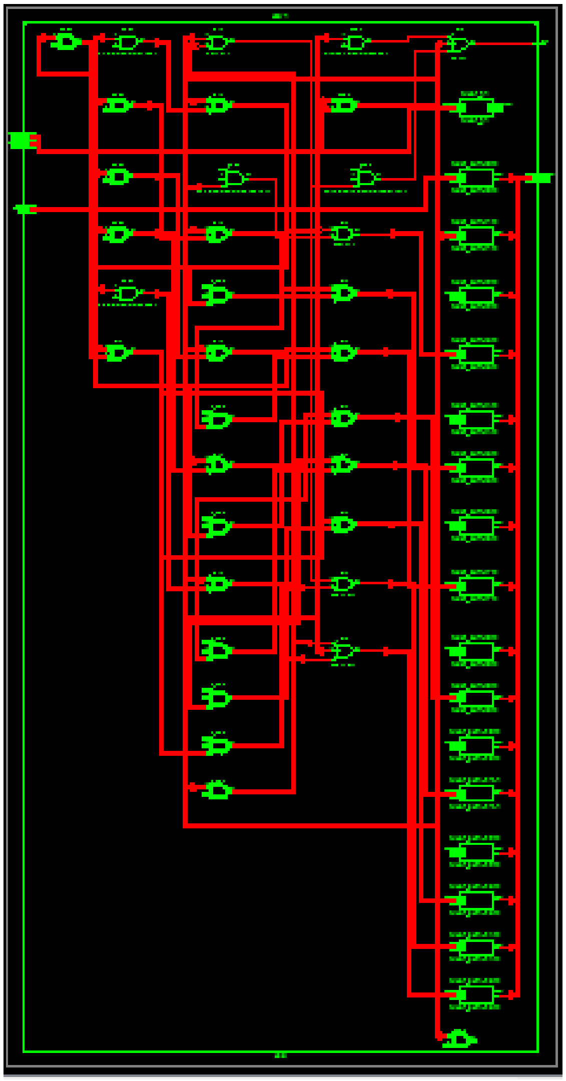


Figure (e): RTL Schematic of 16-bit Kogge Stone adder

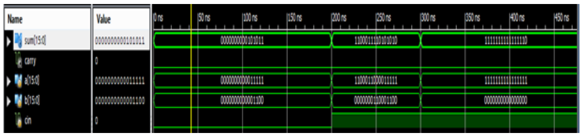


Figure (f): Simulation Wave forms for 16-bit Kogge Stone adder

16-bit Kogge stone adder is designed, then the RTL schematic is shown in figure (e) and the number of LUT's are used are 44 and number of bonded IOB's are 50. Time delay for this design is 1.821ns and wave forms are shown in figure (f).

D. 16-bit Brent Kung adder

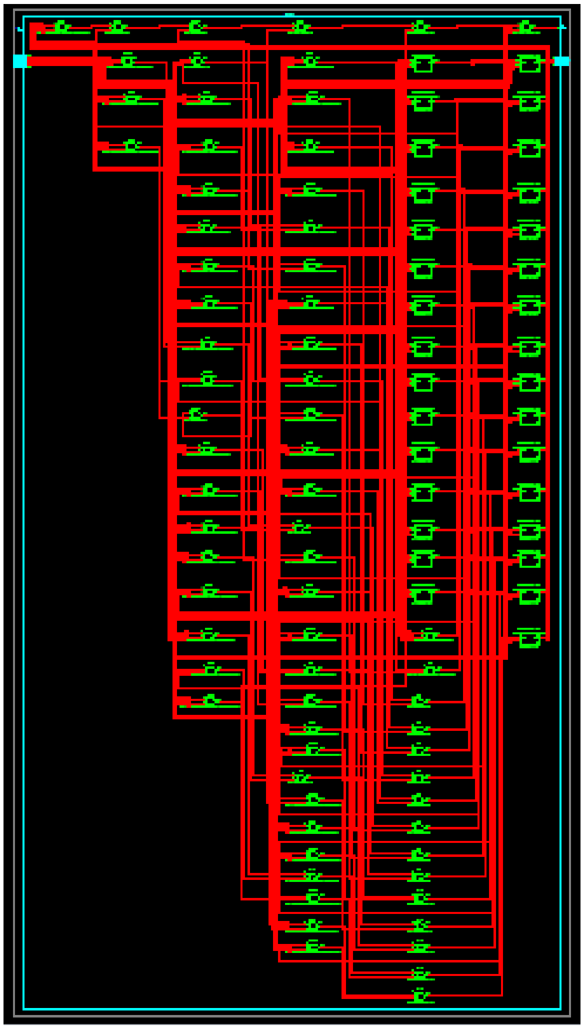


Figure 4(g): RTL Schematic of 16-bit Brent Kung adder

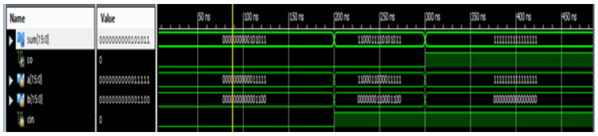


Figure (h): Simulation Wave forms for 16-bit Brent Kung adder

16-bit Brent Kung adder is designed, then the RTL schematic is shown in figure (g) and the number of LUT's are used are 48 and number of bonded IOB's are 49. Time delay for this design is 13.773ns and wave forms are shown in figure (h).

Table 1: Comparison of Wallace tree multiplier and Parallel prefix adders

Designs	No. of Slices	No. of LUTs	No. of IOBs	Delay(ns)
Wallace tree multiplier	0	120	32	7.316
Carry look-ahead	18	32	50	21.378

adder				
Kogge Stone adder	44	0	50	1.821
Brent Kung adder	27	48	49	13.77

ACKNOWLEDGMENT

We sincerely thank to my project guide, who helped us in all aspects of our project to complete in short term. We also thank Bapatla Women's Engineering College for providing necessary facilities towards carrying out this work.

REFERENCES

[1] Himanshu Bansal, K. G. Sharma, Tripti Sharma, "Wallace Tree Multiplier Designs:A Performance Comparison Review, "Innovative Systems Designs and Engineering Vol.5, No.5, 2014.

[2] Adilakshmi Siliveru Design of Kogge stone and Brent kung adders using degenerate pass transistor logic‟, International journal of emerging science and engineering 2013.

[3] G. Shireesha & Dr. G. Kanaka Durga "Design and Implementation of Wallace Tree Multiplier Using Kogge Stone Adder and Brent Kung Adder" International Journal of Emerging Engineering Research and Technology V3. I8. August 2015 112.

[4] S. Rajaram; K. Vanithamani, "Improvement of Wallace Multipliers using Parallel Prefix Adders", 2011 International Conference on Signal Processing, Communication, Computing and Networking Technologies.

[5] MeghaTalsania and Eugene John "Comparative Analysis of Parallel Prefix Adders".

[6] Sudheer Kumar Yezerla, B Rajendra, "Design and Estimation of delay, Power and area for Parallel Prefix Adders," Proceedings of 2014 RAECS UIET Panjab University Chandigarh, 06-08 March, 2014.

[7] Published By: Blue Eyes Intelligence Engineering & Sciences Publication "Design & Implementation of Wallace Tree Multiplier and Kogge Stone Adder".

[8] Y.d. Ykuntam, K. Pavani and K. Saladi, "Design and analysis of High Speed Wallace tree multiplier using Parallel prefix adders for VLSI circuit designs, "2020 11th International Conference on Computing Communication and Networking Technologies (ICCCNT), 2020, pp.1-6.

[9] D. Harris, Taxonomy of Parallel Prefix Networks," in Proc. 37th Asilomar Conf. Signals Systems and Computers, pp. 2213–7, 2003.