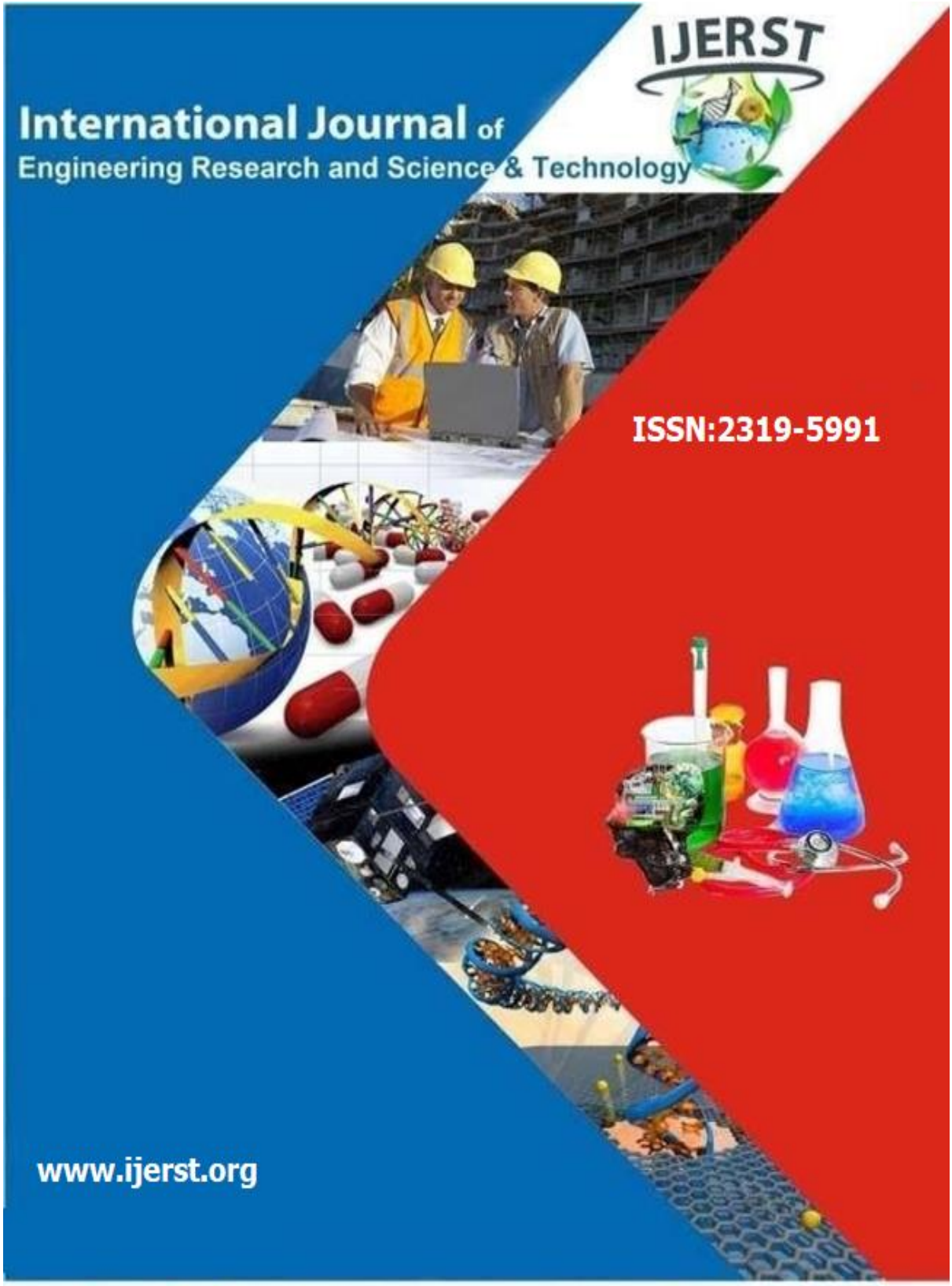
International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025



E-mail: editor@ijerst.org or ijerst.editor@gmail.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

DESIGN AND ANALYSIS OF HIGH SPEED WALLACE TREE MULTIPLIER USING PARALLEL PREFIX ADDRESS FOR VLSI CIRCUIT DESIGN

[1] Shaik Mastanvali,[2] Shaik Rasool, [3] Vallepu Triveni

[12]Assistant Professor,[3]Student

Department of Electronics and Communication Engineering

Prakasam Engineering College (Autonomous)

ABSTRACT

Multiplication is a fundamental operation in digital circuits, and its efficiency directly impacts the performance of high-speed computing systems. The Wallace Tree Multiplier (WTM) is widely used in Very Large Scale Integration (VLSI) designs due to its parallel reduction capability, significantly reducing delay compared to conventional multipliers. However, the accumulation of partial products in WTM still introduces latency and power consumption challenges. To further enhance speed and power efficiency, this study proposes a high-speed Wallace Tree Multiplier integrated with Parallel Prefix Adders (PPA). The PPA optimizes the final addition stage, reducing propagation delay and improving overall performance. The proposed design is implemented and analyzed in terms of delay, area, and power consumption, demonstrating superior efficiency over traditional array multipliers and conventional Wallace multipliers. The results indicate that the proposed approach achieves significant reductions in critical path delay and power dissipation, making it ideal for high-performance VLSI applications.

I. INTRODUCTION

1.1 Overview

Very Large Scale Integration (VLSI) based multipliers are crucial components in modern digital integrated circuits and microprocessors, serving as the fundamental building blocks for arithmetic operations such as multiplication. These multipliers are designed to efficiently and rapidly compute the product of two binary numbers, a task of paramount importance in various computational and signal processing applications.

VLSI multipliers are typically implemented using complex combinations of logic gates and flip-flops, allowing them to process binary numbers of varying lengths with high speed and precision. They come in various architectures, each offering a trade-off between area, speed, and power consumption, tailored to the specific requirements of the target application.

One of the most common VLSI multiplier architectures is the array multiplier, which leverages an array of AND gates and adders to perform partial products multiplication and accumulation. This architecture is highly parallel and can achieve high-speed multiplication but often consumes more area due to the large number of gates.

Another popular design is the Wallace Tree multiplier, which reduces the number of partial products by exploiting the properties of binary numbers. It uses a tree structure of adders and shifters to optimize area and power consumption, although it may sacrifice some speed in the process.

Furthermore, there are more advanced multiplier architectures like the Booth multiplier, which optimizes the number of partial product terms processed, offering a good balance between speed and area. Additionally, in recent years, VLSI multipliers have seen innovations with the introduction of

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

techniques like carry-save and modified Booth encoding, as well as parallel and pipelined designs, to further enhance performance.

1.2 Motivation

The motivation for conducting research in the field of VLSI-based multipliers stems from their critical role in modern digital systems and the ever-increasing demand for improved computational efficiency, power savings, and performance optimization. Several key factors drive researchers to explore and innovate in this domain.

Firstly, the relentless quest for faster and more efficient computational systems is a driving force. Multipliers are essential components in a wide range of applications, including microprocessors, digital signal processors, and embedded systems. As these applications become more complex and demand higher computational throughput, researchers are motivated to develop VLSI multipliers that can meet or exceed these escalating performance requirements.

Secondly, the growing emphasis on energy efficiency in electronic devices has sparked interest in designing low-power VLSI multipliers. The proliferation of mobile and battery-powered devices, as well as the need for environmentally friendly computing solutions, has created a strong incentive to develop multipliers that minimize power consumption without compromising performance. Researchers are exploring various techniques, such as voltage scaling, clock gating, and algorithmic optimizations, to create energy-efficient VLSI multipliers.

1.3 Problem Statement

The problem statement in the context of VLSI-based multipliers research revolves around the pressing need to continually advance the design and optimization of these crucial components to address the evolving demands of modern digital systems. The challenges are multifaceted and encompass several key aspects:

Performance Optimization: As computational requirements continue to grow, there is a persistent demand for VLSI multipliers that can operate at higher speeds and handle larger data sets efficiently. Achieving this while maintaining accuracy and minimizing latency is a central challenge.

Energy Efficiency: With the proliferation of battery-powered devices and the global push for energy-efficient computing solutions, reducing the power consumption of VLSI multipliers is of paramount importance. Balancing performance and power efficiency remains a complex trade-off.

Area Constraints: Silicon real estate is finite, and as electronic devices become smaller and more integrated, optimizing the physical footprint of VLSI multipliers is a critical problem. Researchers need to devise innovative architectural and circuit-level solutions to minimize space requirements.

Technology Evolution: The continuous evolution of VLSI fabrication technologies introduces both opportunities and challenges. Researchers must adapt multipliers to leverage emerging technologies like advanced semiconductor processes and packaging methods, ensuring compatibility and performance improvements.

Reliability and Fault Tolerance: As VLSI multipliers are integral to mission-critical systems, ensuring their reliability and resilience to hardware faults and errors is a critical concern. Developing fault-tolerant designs and testing methodologies is an ongoing challenge.

1.4 Applications

VLSI-based multipliers find a multitude of applications across various domains due to their fundamental role in arithmetic and mathematical computations. These applications include:

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

— **Microprocessors and Digital Signal Processors (DSPs)**: Multipliers are essential in the heart of microprocessors and DSPs, enabling rapid execution of arithmetic operations for general-purpose computing and signal processing tasks. These processors are ubiquitous in computers, smartphones, and embedded systems, where speed and efficiency are critical.

— **Communication Systems**: VLSI multipliers play a pivotal role in wireless communication systems, including cellular networks and wireless standards like 4G and 5G. They are used for channel coding, error correction, and complex modulation schemes, where high-speed multiplication is essential for data processing and transmission.

— **Digital Image and Video Processing**: In applications like image and video compression, multimedia codecs, and computer vision, multipliers are used for tasks such as discrete cosine transform (DCT) and matrix multiplication, enhancing image and video quality and enabling real-time processing.

— **Cryptographic Systems**: Security protocols, encryption, and decryption algorithms rely on complex mathematical operations that heavily involve multiplication. High-performance multipliers are essential for ensuring the security and privacy of sensitive data in applications such as e-commerce, online banking, and secure communication.

— **Scientific Computing**: In scientific simulations, finite element analysis, and numerical modeling, VLSI multipliers are indispensable for performing large-scale matrix operations, solving complex equations, and accelerating simulations in fields such as physics, chemistry, and engineering.

— **Artificial Intelligence and Machine Learning**: Multipliers are integral to neural network training and inference, where they perform countless multiply-accumulate (MAC) operations for weight updates and activations. Accelerated hardware for machine learning, such as GPUs and TPUs, relies on optimized VLSI multipliers.

II. LITERATURE SURVEY

Kumari, A., Kharwar, S., Singh, S., Mohammed, M.K.A., Zaki, S.M. (2023) presented model consists of slice LUT‘s and power of the proposed 2 × 2 and 4 × 4 novel decoder based Wallace tree multiplier using Urdhva Tiryakbhayam sutra were calculated and compared with conventional multiplier. Therefore, utilizing the advantages of Vedic architectures with the proposed idea to solve the problem of balancing power consumption and speed increased in circuits. The simulations carried out and synthesis of the proposed 2 × 2 bit and 4 × 4 bit multiplier has been implemented using artex-7 on Xilinx Vivado. The results of the proposed Wallace tree multiplier with existing Wallace tree multiplier exhibits a significant improvement in term of resource utilization.

Patel, Chiranjit R., et al. paper proposes the design and implementation of an enhanced binary multiplication technique.. The main objective of this paper is to design an improved binary multiplier which is faster and low-powered. The performance of our proposed full adder design is proven to be more effective in comparison with the standard full adder cell both designed in 90nm. The proposed modified 2-Bit and 4-Bit Wallace tree multipliers also beat the existing Wallace tree multiplier based in Urdhva Tiryagbhyam sutra in terms of operating frequency, energy.

Dhanasekar, S., et al. optimized area for a 64-point FFT processor using a Wallace tree multiplier with a modified compressor

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

adder has been proposed. Multi-radix approaches such as radix-23, radix-22, and radix-24 decrease the computational cost of the FFT processor. In the FFT processor, 8-bit Urdhva Tiryakbhyam wallace tree multipliers have been introduced, which reduces the total length of multiplication operations. A modified 4:2 Compressor adder will be used to speed up the wallace tree multipliers, minimizing carry delay. Urdhva Tiryakbhyam wallace tree multipliers uses modified compressor adders in the multi-radix FFT processor to reduce the silicon chip hardware area.

Akhter, Shamim, and Saurabh Chaturvedi paper presents a modified binary multiplier using Vedic mathematics. The paper proposes a modification in the previously published Wallace tree multiplier circuit. The suggested modified Vedic multiplication technique is more efficient in terms of delay and area. The proposed circuit is implemented in VHDL. The Mentor Graphics ModelSim tool is used for HDL simulation, and the Xilinx ISE Design Suite 14.1 is used for circuit synthesis. The simulation is done for 4-bit, 8-bit, and 16-bit multiplication operations. In this paper, the simulation waveforms are shown only for 4-bit multiplication operation based on the modified Vedic multiplication technique.

Kumar, Prashant, and Sangeeta Singh (2019) proposed design demonstrates significant improvements in circuit complexity, area efficiency, and quantum cost while retaining performance in terms of latency and area usage. Coplanar crossovers are properly realized using 180° clock zones. The performance of the proposed design surpasses that of recent literature designs, with a 25% reduction in the circuit complexity, a 28% saving in the total area, a 24.57% decrease in the cell area, and an approximately 1/4 reduction in the quantum cost. To verify the feasibility of the proposed design, its thermal robustness is analyzed and hazard analysis is also performed. The proposed full adder is further employed to realize reversible ripple-carry adders (RCAs) of variable size. The proposed RCAs also show significant improvements in terms of the mentioned performance parameters.

Murugesh, M. Bala, et al. paper presents a modified binary multiplier using Vedic mathematics. The paper proposes a modification in the previously published Wallace tree multiplier circuit. The suggested modified Vedic multiplication technique is more efficient in terms of delay and area. The proposed circuit is implemented in VHDL. The Mentor Graphics ModelSim tool is used for HDL simulation, and the Xilinx ISE Design Suite 14.1 is used for circuit synthesis. The simulation is done for 4-bit, 8-bit, and 16-bit multiplication operations. In this paper, the simulation waveforms are shown only for 4-bit multiplication operation based on the modified Vedic multiplication technique.

Praveen Kumar, Y. G., et al. primitive constraints in any VLSI system design are power, delay and area. Systems based on CMOS logic consume more power and area. Higher power dissipation will have a direct effect on the lifetime and performance of digital systems. Adders and multipliers form the core of almost all the digital systems like Microprocessor, Digital Signal Processors (DSPs), etc., so the adders and multipliers need to be optimized in terms of power, area and delay for an efficient and cost-effective processor design. This paper presents an approach for implementing 8-bit Array multiplier, Wallace Tree multiplier and Wallace tree multiplier using modified Gate Diffusion Input (m- GDI) technology and comparative analysis against area, power and delay.

III. Types of Multipliers

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

3.1 EXISTING METHODS-MULTIPLERS:

3.1.1 MULTIPLERS

Multipliers play an important role in today's digital signal processing and various other applications. With advances in technology, many researchers have tried and are trying to design multipliers which offer either of the following design targets

1. High speed,
2. Low power consumption,
3. Regularity of layout and hence less area or even combination of them in one multiplier thus making them suitable for various high speed,
4. Low power and compact VLSI implementation.

The common multiplication method is "add and shift" algorithm. In parallel multipliers number of partial products to be added is the main parameter that determines the performance of the multiplier. To reduce the number of partial products to be added, with increasing parallelism, the amount of shifts between the partial products and intermediate sums to be added will increase which may result in reduced speed, increase in silicon area due to irregularity of structure and also increased power consumption due to increase in interconnect resulting from complex routing. On the other hand "serial-parallel" multipliers compromise speed to achieve better performance for area and power consumption. The selection of a parallel or serial multiplier actually depends on the nature of application. In this lecture we introduce the multiplication algorithms and architecture and compare them in terms of speed, area, power and combination of these metrics. AND gates are used to generate the Partial Products (PP). If the multiplicand is N-bits and the Multiplier is M-bits then there is N* M partial product.

3.1.2 HISTORY OF MULTIPLIERS

The early computer systems had what are known as Multiply and Accumulate units to perform multiplication between two binary unsigned numbers. The Multiply and Accumulate unit was the simplest implementation of a multiplier. The basic block diagram of such a system is given below.

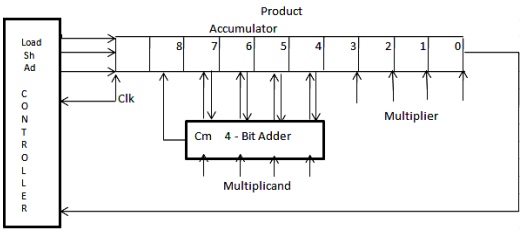


Fig.3.1 Multiplier Block Diagram

3.1.3 IMPLEMENTATION

The MAC unit requires a 4-bit multiplicand register, 4-bit multiplier register, a 4-bit full adder and an 8-bit accumulator to hold the product. In the figure above the product register holds the 8-bit result. In a typical binary multiplication, based on the multiplier bit being processed, either zero or the multiplicand is shifted and then added.

The desire to speed up the rate at which the output is generated resulted in the development of this category of multiplier. In a serial-parallel multiplier discussed above, it takes one clock cycle to process one bit of the data input at any given time. Therefore, when working on an N-bit input it would take at least N clock cycles to generate the final output. In a parallel array multiplier the result is obtained as soon as inputs are presented to the multiplier. This is mainly because of the use of AND array structure to compute the partial product terms. Once the partial product terms are generated the only delay in generating the output is contributed by the adders which sum the partial product terms column wise to generate the result. The figure below represents a parallel array multiplier with N=8 bit inputs. In Figure block A stands for an AND gate. Block AHA stands for AND

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

GATE and HALF ADDER structure and AFA stands for AND GATE and FULL ADDER structure. FA stands for full adder.

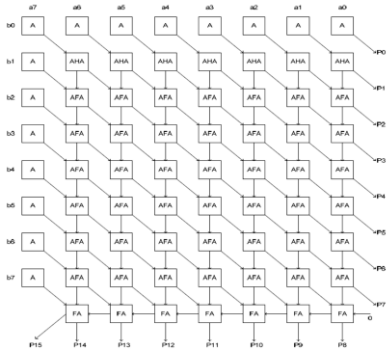


Fig.3.2 Parallel array multiplier for N=8 bits.

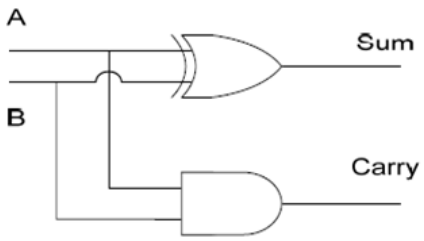


Fig.3.3 Gate level implementation of a HALF ADDER.

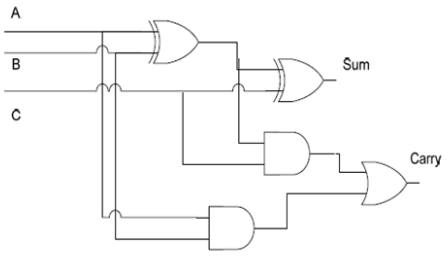


Fig.3.4 Gate level implementation of a FULL ADDER.

3.2 TYPES OF MULTIPLIERS

3.2.1 CARRY SAVE MULTIPLIER

This is responsible for multiplying the unsigned significand and placing the decimal point in the multiplication product. The result of significand multiplication will be called the intermediate product (IP). The unsigned significand multiplication is done on 24 bit. Multiplier performance should be taken into consideration so as not to affect the whole multiplier's performance. A 24x24 bit carry save multiplier architecture is used as it has a moderate speed with a simple architecture. In the carry save multiplier, the carry bits are passed diagonally downwards (i.e. the carry bit is propagated to the next stage). Partial products are made by ANDing the inputs together and passing them to the appropriate adder. This is done in significand multiplication process which is one of the important steps in floating point multiplication.

Carry Save Multiplier Has Three Main Stages:

1. The first stage is an array of half adders.
2. The middle stages are arrays of full adders.
3. The number of middle stages is equal to the significand size minus two.
4. The last stage is an array of ripple carry adders. This stage is called the vector merging stage

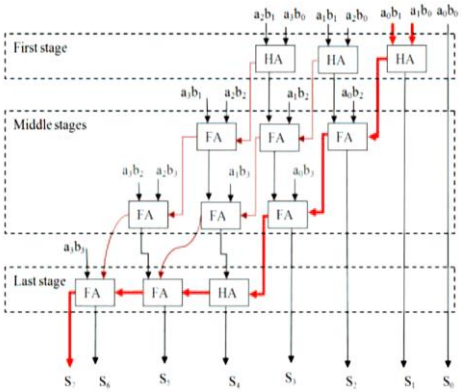


Fig.3.6 Carry Save Multiplier

3.2.2 ARRAY MULTIPLIERS

Array multiplier is an efficient layout of a combinational multiplier. In array multiplier, consider two binary numbers A and B, of m and n bits. There are mn summands that are produced in parallel by a set of mn AND gates. n x n multiplier requires n (n-2) full adders, n half-adders and n2 AND gates. Also, in array multiplier worst case delay would be (2n+1) td. Array Multiplier gives

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

more power consumption as well as optimum number of components required, but delay for this multiplier is larger. It also requires larger number of gates because of which area is also increased; due to this array multiplier is less economical .Thus, it is a fast multiplier but hardware complexity is high.

Array multiplier is well known due to its regular structure. Multiplier circuit is based on add and shift algorithm. Each partial product is generated by the multiplication of the multiplicand with one multiplier bit. The partial product are shifted according to their bit orders and then added. The addition can be performed with normal carry propagate adder. N-1 adders are required where N is the multiplier length.

- Each stage of parallel adders should receive some partial product inputs.
- Since it has regular structure, it is easy to layout and has a small structure.
- Design time of array multiplier is much less than tree multiplier.
- These have worst delay and slow speed for wide multiplier

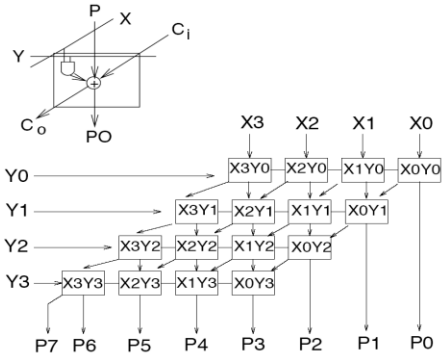


Fig.3.7 Array Multipliers

3.2.3 LINEAR ARRAYS

A faster version of the basic iterative multiplier adds more than one operand per clock cycle by having multiple adders and partial product generators connected in series .This is the equivalent of "unrolling" the simple iterative method. The degree to which the loop is unrolled determines the number of partial products that can be reduced in each clock cycle, but also increases the hardware requirements. Typically, the loop is unrolled only to the point where the system clock matches the clocking rate of this multiplier. Alternately, the loop can be unrolled completely, producing a completely combinatorial multiplier (a full linear array). When contrasted with the simple iterative scheme, it will match the system clock speed better, making the clocking much simpler. There is also less overhead associated with clock skew and register delay per partial product reduced. The operation which we would like to realize, i.e. C = A × B has the following structure for n = 4,

$$
\begin{array}{cccccccc}
 & & & & a_3 & a_2 & a_1 & a_0 \\
\times & & & & b_3 & b_2 & b_1 & b_0 \\
\hline
 & & & & a_3b_0 & a_2b_0 & a_1b_0 & a_0b_0 \\
 & & & a_3b_1 & a_2b_1 & a_1b_1 & a_0b_1 & \\
 & & a_3b_2 & a_2b_2 & a_1b_2 & a_0b_2 & & \\
+ & a_3b_3 & a_2b_3 & a_1b_3 & a_0b_3 & & & \\
\hline
c_7 & c_6 & c_5 & c_4 & c_3 & c_2 & c_1 & c_0
\end{array}
$$

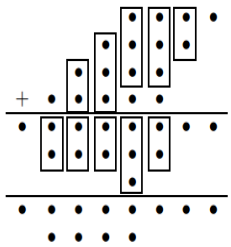


Fig.3.8 Linear Arrays

3.2.4 SIGNED PARALLEL MULTIPLIERS

Using 2's complement representation we can represent and add/subtract signed quantities easily. Consider the partial product matrix

$$
\begin{array}{cccccccc}
 & & & & a_3 & a_2 & a_1 & a_0 \\
\times & & & & b_3 & b_2 & b_1 & b_0 \\
\hline
 & & & & a_3b_0 & a_2b_0 & a_1b_0 & a_0b_0 \\
 & & & a_3b_1 & a_2b_1 & a_1b_1 & a_0b_1 & \\
 & & a_3b_2 & a_2b_2 & a_1b_2 & a_0b_2 & & \\
+ & a_3b_3 & a_2b_3 & a_1b_3 & a_0b_3 & & & \\
\hline
c_7 & c_6 & c_5 & c_4 & c_3 & c_2 & c_1 & c_0
\end{array}
$$

If a and b are signed quantities then the rows of the partial products are signed

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

numbers as well. Remember that if b is signed then we can build (and sum) the partial products as signed 2's complement integers. Indeed the terms a_3b_0, a_3b_1, a_3b_2, a_3b_3 represent the sign bits of the integers we are adding. T o properly add the partial products together we need to align the precisions and sign extend the rows. That is

$$\begin{array}{cccccccccc} & \alpha & \alpha & \alpha & \alpha & x & x & x & x \\ & \beta & \beta & \beta & x & x & x & x & \\ & \gamma & \gamma & x & x & x & x & & \\ + & \theta & x & x & x & x & & & \\ \hline \end{array}$$

Note that α β γ θ represent the sign bits of the rows. Hence, we can summarize the method as sign extend and add as if we are handling unsigned numbers and ignore carry out.

3.3 PROPOSED MULTIPLIER:

In this work, we present a technique that allows partial product arrays of maximum height of n/m (with the goal of not increasing the delay of the partial product generation stage), for r > 4 and unsigned multipliers. Since for the standard unsigned multiplier the maximum height is (n + 1)/m, the proposed method allows a reduction of one row when n is a multiple of m. Our technique is general, but its impact (reduction of one row without increasing the critical path of the partial product generation stage) depends on the specific timing of the different components. Therefore, we can not claim a successful result for all practical values of r and n and different implementation technologies. Thus, we concentrate on an specific instance: a 64-bit radix-4Booth recoded unsigned multiplier implemented with a synthesis tool and a standard-cell library. We use radix-4since it is the most complex case, among the practical values of the radix, for the design of our scheme. The unsigned multiplier is also more complex for the design of our scheme than the signed multiplier. We use 64 bits, since it is a representative large word length. The method proposed can be adapted easily to other instances (signed, combined unsigned/signed, radix-8 recoding, different values of n).

IV. EXISTING METHOD

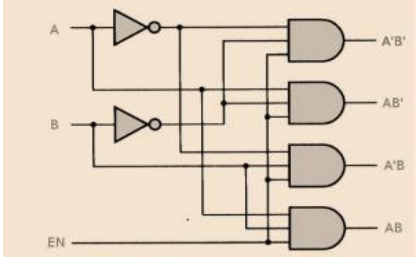
4.1 FA using decoder

Decoder is a combinational circuit that converts binary information from n input lines to 2^n unique output lines. Apart from the Input lines, a decoder may also have an Enable input line A Decoder with Enable input can function as a demultiplexer. A demultiplexer is a circuit that receives information from a single line and directs it to one of 2^n possible output lines.

A 2^n demultiplexer receives as input, n selection lines and one Input line. These selection lines are used to select one output line out of 2^n possible lines. To implement a 2^n demultiplexer, we use a n:2^n decoder with Enable input.

The n selection lines of the demultiplexer are the n input lines that the decoder gets and the one input line of demultiplexer is the Enable input of the Decoder. Making 1:4 demultiplexer using 2:4 Decoder with Enable input. Let A, B be the selection lines and EN be the input line for the demultiplexer.

The decoder shown below functions as a 2:4 demultiplexer when EN is taken as a data input line and A and B are taken as the selection inputs. The single input variable E has a path to all four outputs, but the input information is directed to only one of the output lines, as specified by the binary combination of the two selection lines A and B. This can be verified from the truth table of the circuit.



International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

Fig: 4.1 Decoder Circuit.

Truth table:

INPUTS			OUTPUTS	
A	**B**	**Cin**	**Cout**	**S**
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Table: 4.1 Truth table for Full Adder using Decoder circuit

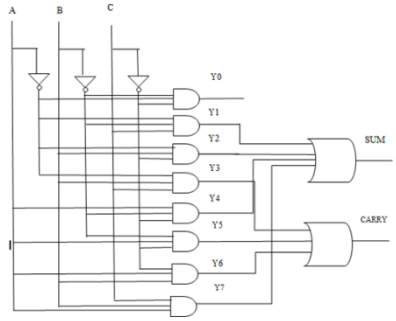


Fig 4.2 Vedic Decoder Multiplier

4.2 Vedic Wallace Multiplier

In general, Wallace tree addition uses full adders to extensively reduce the partial products.

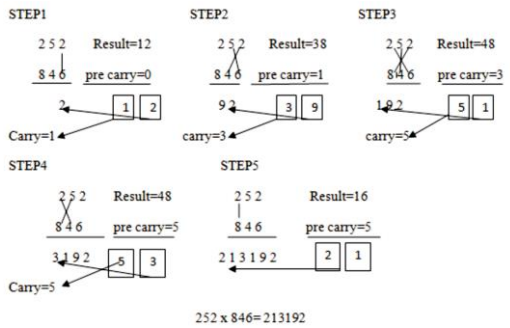


Figure 4.1 Multiplication of Two Decimal Numbers: 252x846

When the critical path is compared between the critical path in 4 bit conventional and Wallace tree multiplier, for a 4-bit multiplier, 4 partial products will be generated, as shown in Figure 2 and are named as p0 to p3. For Wallace tree multiplier, a 3:2 reduction is used, so that the partial products are reduced from 4 to 3. The Delay in critical path is given by the addition of 3 full adder sums, 2 full adder carry, and half adder carry. The critical path for Vedic mathematics as shown in Figure 3, is given by 2FAS is reduced by 3HAS and in terms of XOR gates, Vedic-Wallace uses 3XOR gates instead of 4XOR i.e., less carry propagation delay than the conventional method. Hence, Vedic-Wallace has a variable improvement over design ware depending upon the number of bits in multiplication

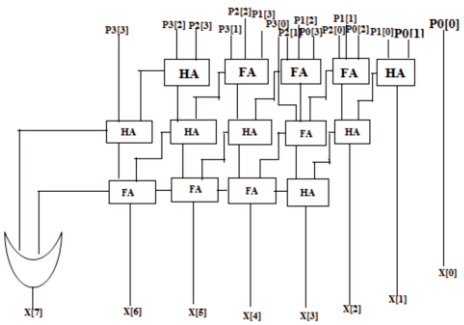


Figure 4.2 4x4 Multiplier using Wallace Tree

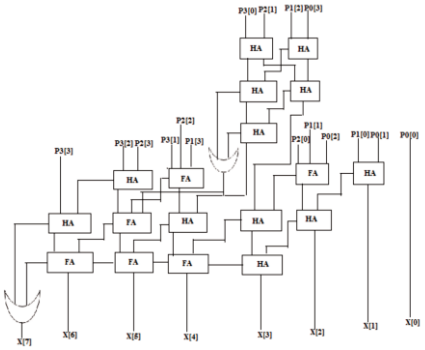


Figure 4.3 .4x4 Multiplier using Vedic Reduction

4.3 Different Multiplier Architectures

The hardware architecture of 2x2, 4x4, 8x8, 16x16, 32x32 bit Vedic Wallace multiplier (VW) modules are displayed in below

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

sections. Here, "Urdhva-Tiryakbhyam" (Vertically and crosswise) sutra is used to propose such an architecture for the multiplication of two binary numbers. The beauty of Vedic Wallace multiplier is that, here partial product generation and additions are done concurrently. Hence, it is a well adapted parallel processing. The features make it more attractive for binary multiplications. This reduces delay and this is the primary motivation behind this work.

V. PROPOSED METHOD

5.1 Introduction

In economics, a multiplier refers to the effect that a change in one economic variable has on another variable. For example, the government spending multiplier refers to the increase in overall economic output that results from an increase in government spending. Similarly, the investment multiplier refers to the increase in overall economic output that results from an increase in private sector investment. The size of the multiplier depends on various factors, such as the type of spending or investment and the state of the economy.

Wallace tree multipliers are a type of multiplier circuit that are based on ancient Indian mathematics. The term "Vedic" refers to the Vedas, which are a collection of ancient Hindu texts. Vedic mathematics is a system of mathematics that is based on 16 sutras, or aphorisms, and 13 sub-sutras, or corollaries. These sutras are used to perform complex mathematical operations quickly and efficiently. Wallace tree multipliers use the principles of Vedic mathematics to perform multiplication operations. They are known for their speed, low power consumption, and simplicity of design. There are several different types of Wallace tree multipliers, such as the Urdhva Tiryakbhyam (UT) multiplier, Nikhilam multiplier, and the Paravartya Yojayet (PVY) multiplier. The Urdhva Tiryakbhyam multiplier is the most used Wallace tree multiplier. It works by first multiplying the two most significant bits of the two input numbers and then adding the result to the product of the two least significant bits. This process is repeated for the other pairs of bits, and the partial products are added together to obtain the result.

PPA adders are a type of adder that use a decoder to generate partial sums, which are then combined to produce the final sum. They are known for their low power consumption and are often used in low-power applications. Combining modified Wallace tree multipliers with PPA adders is an area of research that aims to develop more efficient arithmetic units for digital signal processing and other applications.

The input bits are fed into the decoder, which generates the sum and carry bits based on the input combination. The generated sum and carry bits are then combined to obtain the result. The advantage of a PPA adderis that it is faster and more efficient than other types of adders, such as the ripple carry adder or the carry lookahead adder. It has a lower delay and requires fewer logic gates, which results in lower power consumption and area. However, the disadvantage of a PPA adderis that it requires a large decoder circuit, which can be complex and difficult to design. PPA adders are often used in high-speed digital circuits, such as microprocessors, digital signal processors, and graphics processing units. They are also used in arithmetic logic units (ALUs), which are the fundamental building blocks of digital processors.

5.2 CSLA using PPA adder

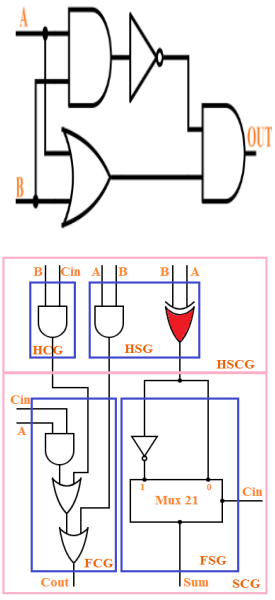
The present generation chips consist of various types of adders such as CSLA, RCA, carry bypass adder, carry look ahead adder and still many more. But this addition requires huge resources such as LUTs and PLBs, which causes to increase the power, delay consumption. To overcome these drawbacks,

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

the proposed PPA ADDER will be effectively used in ALUs. As per this addition process the proposed structure is modified from SQRT based CSLA. And from SQRT based CSLA RCA-BEC based structure will be modified with HSG, HCG, FSG and FCG units. For implementing these building blocks, in this paper new XOR gate is developed with reduced gate count thus quantum cost also reduces. Generally, the XOR gate consists of two NOT gates, two AND gates and one OR gate in its logic realization. The proposed XOR gate is realized as shown in Figure 2(a), while it consists of two AND gates, one OR gate and one NOT gate. By modifying the gate level structure, its logical operation is justified as original XOR function with optimized Quantum Cost.



(a) (b)

Figure 1: (a) XOR gate (b) PPA-FA.

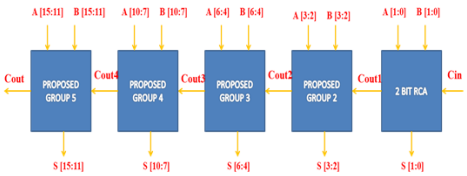


Figure 2. Block diagram of proposed 16-bit HSCG–SCG-CSLA.

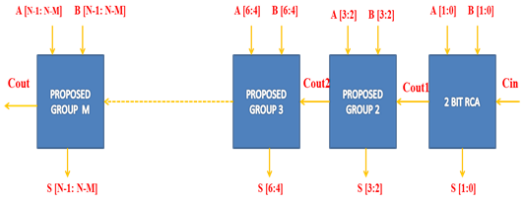


Figure 3: Block diagram of proposed N-bit HSCG–SCG-CSLA.

Figure 3 represents the 16-bit HSCG–SCG-CSLA block diagram, which contains the 5 stages such as $(\sqrt{16}+1)$.The input bit length for each group is incremented order with respect to the group size. Every group has two major input namely A and B, other than carry input. Here, the bit length of each group varies chronologically with respect to its group size. For an instance, group 2 has the 2-bit width for $A$and B inputspositioned from [3:2] each, group 6 has the 6-bit width for A and B inputspositioned from [21:16] each. Each group is connected in ripple carry propagation manner, thus the carry out from the one group is applied as the input to the next group.From the basis of figure 3, the N-bit HSCG–SCG-CSLA is developed with group M stages as shown inFigure 4, so it can be used for 8-bit,16-bit, 32 bit and 64-bit implementations, respectively and the major use of this proposed adder is thatit can be used as reconfigurable purpose. Generally, in signal processing domain and in arithmetic operations, the input variables sizes are changes continuously; in those situations, the reconfigurable adder is required. By changing the parameter of N, the length of the adder changes according to the user requirement to that of corresponding application and "reconfigurability property" will be achieved. The internal structure of the above-mentioned group M is presented in Figure 5.

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

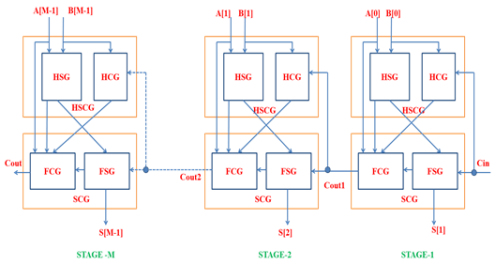


Figure 4: Group structure for proposed PPA ADDER.

VI. RESULTS AND DISCUSSIONS

6.1 Existing Results

Fig 6.1 shows. RTL Schematic for 16-bit Multiplier. A Register Transfer Level (RTL) schematic for a 16-bit multiplier typically illustrates the digital logic circuitry and data flow within the multiplier. It shows how the 16-bit inputs are processed to produce a 32-bit output (the result of multiplying two 16-bit numbers). You might expect to see components like adders, shifters, and multiplexers in this schematic, illustrating how partial products are generated and combined to produce the final product.

Fig 6.2 shows synthesized Design for 16-bit Multiplier. The synthesized design figure represents the result of taking the RTL schematic and implementing it using a hardware description language (e.g., VHDL or Verilog) and then synthesizing it into a specific hardware technology, like an Application-Specific Integrated Circuit (ASIC) or Field-Programmable Gate Array (FPGA). It shows the actual gate-level implementation of the multiplier, detailing how the digital logic gates are interconnected to form the functional circuitry.

Fig 6.3 shows Simulation for 16-bit Multiplier. A simulation figure demonstrates how the 16-bit multiplier operates in a virtual environment. It typically includes a waveform diagram or graphs that show input signals, intermediate signals, and the final output signal over time or cycles. The simulation helps verify the correctness and performance of the multiplier design without physically implementing it. It can provide insights into timing, power consumption, and potential issues that may need to be addressed before the design is fabricated.

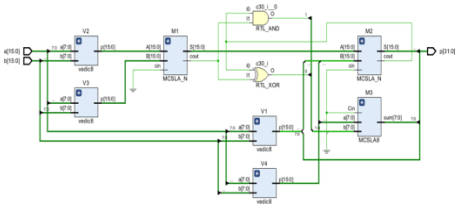


Fig 6.1. RTL schematic for 16-bit multiplier.

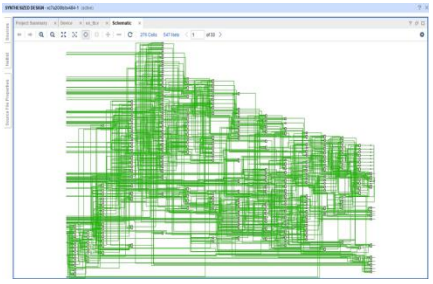


Fig 6.2. Synthesized Design for 16-bit multiplier

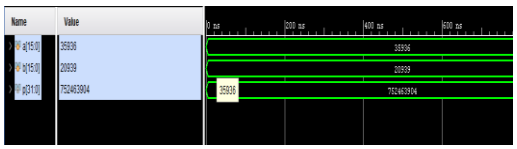


Fig: 6.3. Simulation for 16-bit multiplier.

6.2 Proposed Results

Fig 6.4 shows 4-Bit Proposed Wallace tree multiplier Simulation Result. This figure likely displays the simulation results of a proposed 4-bit Wallace tree multiplier. It include waveform diagrams or graphs showing input signals, intermediate signals, and the final output signal. The simulation results help evaluate the functionality and performance of the 4-bit Wallace tree multiplier design.

Fig: 6.4. 4 Bit Proposed Wallace tree multiplier Simulation Result.

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

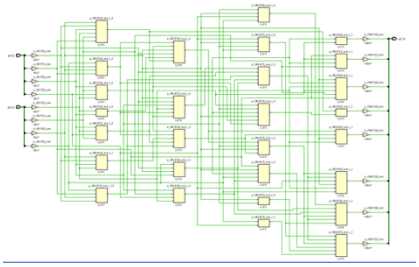


Fig: 6.5. Proposed Wallace tree multiplier Synthesised Design.

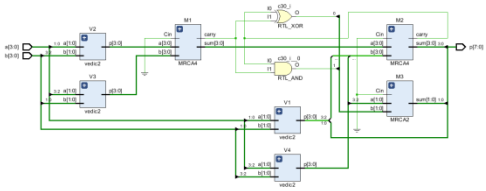


Fig: 6.6. Proposed Wallace tree multiplier RTL Schematic.

Fig 6.5 shows the Proposed Wallace tree multiplier Synthesized Design. This figure likely shows the synthesized design of the proposed Wallace tree multiplier. It illustrates the gate-level implementation of the multiplier, detailing how digital logic gates are interconnected to create the functional circuitry. The synthesized design is a crucial step in transforming the high-level design into hardware.

Fig 6.6 shows Proposed Wallace tree multiplier RTL Schematic. The RTL schematic for the proposed Wallace tree multiplier provides an overview of the digital logic circuitry and data flow within the multiplier. It illustrates how the multiplier's components are interconnected to perform multiplication operations. This figure helps understand the multiplier's architecture at a higher level of abstraction.

Fig 6.7 shows 8-Bit Proposed Wallace tree multiplier Simulation Result. Similar to Fig 7.4, this figure likely displays the simulation results, but for an 8-bit version of the proposed Wallace tree multiplier. It may show how the multiplier behaves when processing 8-bit input data.

Fig 6.8 shows16-Bit Proposed Wallace tree multiplier Simulation Result This figure probably displays the simulation results for a 16-bit version of the proposed Wallace tree multiplier. It provides insights into the multiplier's performance and functionality when dealing with 16-bit input data.

Fig 6.9 shows 32-Bit Proposed Wallace tree multiplier Simulation Result. Likewise, this figure likely shows the simulation results for a 32-bit version of the proposed Wallace tree multiplier, indicating how it operates with larger input data.

Fig 6.10 shows 64-Bit Proposed Wallace tree multiplier Simulation Result. This figure probably shows the simulation results for the most significant version, a 64-bit proposed Wallace tree multiplier. It demonstrates the multiplier's performance and scalability when handling significantly larger input data.

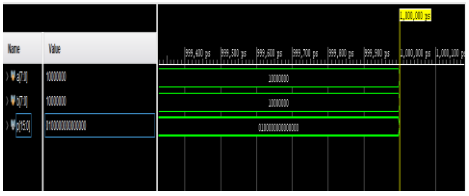


Fig: 6.7. 8-Bit Proposed Wallace tree multiplier Simulation Result

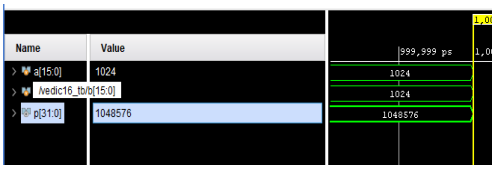


Fig: 6.8. 16-Bit Proposed Wallace tree multiplier Simulation Result.

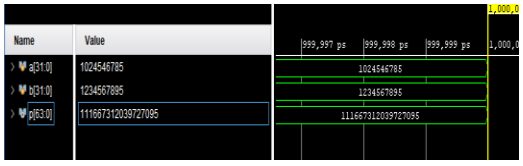
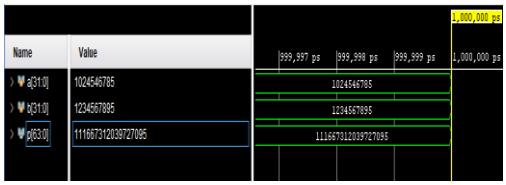


Fig: 6.9. 32-Bit Proposed Wallace tree multiplier Simulation Result.



ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

Fig: 6.10. 64-Bit Proposed Wallace tree multiplier Simulation Result.

VII. CONCLUSION

7.1 CONCLUSION

This work has presented a systematic method for binary multiplier circuits which is based on Vedic mathematics. When it comes to the terms of time delay then the proposed system is more efficient than existing methods. Elongation for a higher bit size can be done with help of proposed technique. Moreover, adders of different architectures can be used in the PPA ADDER design used in the proposed modified Wallace tree multiplier. Among many techniques modified architecture is used to increase and speed up the multiplication.

7.2 FUTURE SCOPE

The future scope of research and development in the field of Wallace tree multipliers and digital arithmetic operations is promising and can lead to several exciting directions:

- High-Performance Computing (HPC): As computational demands continue to increase in fields like scientific simulations, artificial intelligence, and data analytics, there is a need for highly efficient multipliers that can accelerate complex calculations. Future research can focus on developing Wallace tree multipliers optimized for HPC applications, potentially incorporating advanced techniques like approximate computing to balance speed and accuracy.
- Energy-Efficient Processors: With a growing emphasis on energy efficiency and sustainability, Wallace tree multipliers can play a crucial role in designing energy-efficient processors. Future work might involve integrating Wallace tree multiplier designs into low-power processors for mobile devices, IoT applications, and green computing.
- Quantum Computing: Quantum computing represents a revolutionary paradigm shift in computing. Researchers are exploring the potential application of Vedic mathematics principles in quantum algorithms and quantum circuit design. Wallace tree multiplier architectures might find use in quantum computing systems for certain types of calculations.
- Machine Learning Acceleration: Machine learning and deep learning algorithms involve a significant number of matrix operations, which heavily rely on multipliers. Future research can investigate specialized Wallace tree multiplier designs for accelerating neural network training and inference, leading to more efficient AI hardware.
- Custom Hardware Accelerators: Field-Programmable Gate Arrays (FPGAs) and Application-Specific Integrated Circuits (ASICs) are increasingly used for custom hardware acceleration. Optimized Wallace tree multiplier designs can be integrated into these specialized hardware platforms to boost specific application workloads, such as cryptography, image processing, or signal processing.
- Error-Resilient Computing: In scenarios where error tolerance is acceptable, Wallace tree multipliers can be leveraged for designing error-resilient computing systems. Future research may explore how these multipliers can be used in fault-tolerant computing architectures, especially for mission-critical applications.

REFERENCES

[1] Kumari, A., Kharwar, S., Singh, S., Mohammed, M.K.A., Zaki, S.M. (2023). Design and Implementation of Modified Wallace tree multiplier Using Modified PPAAdder. In: Al-Sharafi, M.A., Al-Emran, M., Al-

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

Kabi, M.N., Shaalan, K. (eds) Proceedings of the 2nd International Conference on Emerging Technologies and Intelligent Systems.

[2] Javeed, S., Patil, S.S.: Low power high speed 24-bit floating point Wallace tree multiplier using cadence (2018)

[3] Krishna, A.V., Deepthi, S., Devi, M.N.: Design of 32-bit mac unit using Wallace tree multiplier and XOR logic. In: Proceedings of International Conference on Recent Trends in Machine Learning, IoT, Smart Cities and Applications, pp. 715–723. Springer (2021)

[4] O'Connor, M., Swartzlander, E.E.: Exploiting asymmetry in booth-encoded multipliers for reduced energy multiplication. In: 2015 49th Asilomar Conference on Signals, Systems and Computers, pp. 722–726. IEEE (2015)

[5] Kumar, P., Singh, S.: Optimization of the area efficiency and robustness of a QCA based reversible full adder. J. Comput. Electron. 18(4), 1478–1489 (2019)

[6] Gowthami, P., Satyanarayana, R.: Design of an efficient multiplier using Vedic mathematics and reversible logic. In: 2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), pp. 1–4. IEEE (2016)

[7] Marchesan, G.C., Weirich, N.R., Culau, E.C., Weber, I.I., Moraes, F.G., Carara, E., de Oliveira, L.L.: Exploring RSA performance up to 4096-bit for fast security processing on a flexible instruction set architecture processor. In: 2018 25th IEEE International Conference on Electronics, Circuits and Systems (ICECS), pp. 757–760. IEEE (2018).

[8] Morghade, K., Dakhole, P.: Design of fast Wallace tree multiplier with fault diagnostic capabilities. In: 2016 International Conference on Communication and Signal Processing (ICCSP), pp. 0416–0419. IEEE (2016).

[9] Abhilash, R., Dubey, S., Chinnaiah, M.: ASC design of signed and unsigned multipliers using compressors. In: 2016 International Conference on Microelectronics, Computing and Communications (MicroCom), pp. 1–6. IEEE (2016).

[10] Ram, C.G., Rani, D.S., Balasaikesava, R., Sindhuri, K.B.: VLSI architecture for delay efficient 32-bit multiplier using Vedic mathematic sutras. In: 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), pp. 1873–1877. IEEE (2016).

[11] Pohokar, S., Sisal, R., Gaikwad, K., Patil, M., Borse, R.: Design and implementation of 16 × 16 multiplier using Vedic mathematics. In: 2015 International Conference on Industrial Instrumentation and Control (ICIC), pp. 1174–1177. IEEE (2015).

[12] Ram, C.G., Lakshmanna, Y.R., Rani, D.S., Sindhuri, K.B.: Area efficient modified Wallace tree multiplier. In: 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT), pp. 1–5. IEEE (2016).

[13] Mistri, N.R., Somani, S., Shete, V.: Design and comparison of multiplier using Vedic mathematics. In: 2016 International Conference on Inventive Computation Technologies (ICICT), vol. 2, pp. 1–5. IEEE (2016).

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

[14] Ravali, B., Priyanka, M.M., Ravi, T.: Optimized reversible logic design for Wallace tree multiplier. In: 2015 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), pp. 127–133. IEEE (2015).

[15] Vamsi ASK, Ramesh SR (2019) An efficient design of 16 bit MAC unit using vedic mathematics. In: International conference on communication and signal processing, pp 0319–0322.

[16] Jithin S, Prabhu E (2015) Parallel multiplier—accumulator unit based on vedic mathematics. ARPN J Eng Appl Sci 10(8):3608–3613.

[17] Gadakh SN, Khade A (2016) Design and optimization of 16 × 16 bit multiplier using vedic mathematics. In: International conference on automatic control and dynamic optimization techniques (ICACDOT), pp 460–464.

[18] S.P. Pohokar, R.S. Sisal, K.M. Gaikwad, M.M. Patil and Rushikesh Borse, "Design and Implementation of 16 × 16 Multiplier Using Vedic Mathematics",.

[19] M. C Sudeep, M Sharath Bimba and Mahendra Vucha, "Design and FPGA Implementation of High Speed Wallace tree multiplier", International Journal of Computer Applications, vol. 90, no. 16, March 2014.

[20] Kumari,SKharwar,S.Singh‖ Design and Implementation of 12-bit Wallace tree multiplier using Optimized PPAAdder‖ -2022