

# Optimized Multiplier Architectures for Enhanced Performance and Efficiency in MAC Units

<sup>1</sup>G Priyanka, <sup>2</sup>N Gireesh, <sup>3</sup>S Hemachandra

<sup>1</sup>Research Scholar, Department of ECE, School of Engineering, Mohan Babu University, Tirupati-517102. [22201R030005@mbu.asia](mailto:22201R030005@mbu.asia)

<sup>2</sup>Professor, Department of ECE, School of Engineering, Mohan Babu University, Tirupati-517102.

<sup>3</sup>Dean, IIR & Professor, Department of EEE, School of Engineering, Mohan Babu University, Tirupati-517102.

**Abstract:** This paper investigated the performance of Vedic multipliers in a 32-bit Multiplier-Accumulator Unit (MAC) by comparing Urdhva Tiryakbhyam and Nikhilam Sutras with various adder architectures. The goal was to identify the optimal combination of speed and resource efficiency. Urdhva Tiryakbhyam with CLA emerged as the fastest option, achieving a minimal delay of 0.709 ns. However, this came at the cost of higher resource utilization, measured in Logic Look-Up Tables (LUTs). Conversely, Nikhilam implementations generally required fewer LUTs, making them more resource-efficient, but they exhibited slightly slower performance. CLA consistently delivered the best delay for both Vedic multiplier types among the adder architectures. All the explored configurations are viable for practical implementation on Xilinx ISE 14.7. The key takeaway is that the choice between Urdhva Tiryakbhyam and Nikhilam and the specific adder architecture hinges on the application's priorities.

**Keywords:** Deep Learning Model, MAC, Vedic Multipliers, Wallace Tree.

## 1 INTRODUCTION

Device dimensions are shrinking, posing a clear challenge for VLSI; however, as they ask closer to leakage grows, particularly at sub-threshold atomic dimensions, as various aspects of the performance decrease, their sensitivity to changes in a manufacturing process substantially increases while the gain dramatically decreases. Reliability and yield decrease as a result. Some of the areas of circuit design in VLSI-CAD that are raising much focus are the computationally efficient methods for predicting the outputs based on the inputs. Each VLSI configuration is streamlined to use less space, distribute little power, and attain outstanding throughput to boost dependability and yield. The MAC unit is critical for high-speed computation in digital signal processing systems. This paper reviews various techniques to improve MAC units' efficiency, balancing power consumption, speed, and area constraints in VLSI designs. Traditional designs necessitate separate multiplier and adder units, increasing processor workload and latency. Recent innovations integrate these functions into a single MAC unit, alleviating the Central processing unit (CPU) burden and enhancing overall system performance. This paper's proposed methodology aims to implement MAC units with different multipliers and adders. They estimate the performance of the designed algorithms by comparing them in terms of delay, power, and area.

Numerous digital signal processing applications require the multiplier-and-accumulator unit to perform critical tasks. Various techniques have been brought to create multipliers that provide high speed, low power consumption, and smaller space as the outcome of technological innovations. Thus, they are appropriate for rapid and low-power reduced VLSI executions. These three boundaries, i.e., power, region, and speed, are constantly compromised. Multipliers and adders need to be generated individually in traditional multiplier systems. The demand for the processor will grow as a result. Thus, there may be a little delay in the result. The MAC unit enters the scene as technology improves, which will decrease the burden on the CPU [1][2].

MAC units started emerging in recent years for several kinds of high-performance applications. An essential fundamental element in computers, especially Digital Signal Processors (DSP), is the MAC unit [3][4]. The multiplication and accumulation processes are completed via the MAC unit. An adder, an accumulator, and a multiplier form the basic Multiplier and accumulator unit.

**Adder:** it is used to sum up or add value.

**Accumulator:** given a commutative function to combine values or perform some action.

In most digital and high-performance systems, like filters, microprocessors, and DSP, multipliers are vital to the hardware block. MAC units have been developed recently to support a variety of high-performance applications. The MAC unit is a crucial computer component, especially for DSP. The MAC unit executes the multiplication and accumulation processes [5]. An accumulator, multiplier, and adder make up the fundamental MAC unit. When it comes to the efficient multiplier, which serves as the foundational part of many digital circuits, the processor's performance is dependent on it. The addition process uses a hybrid carry choose adder, and the MAC unit is constructed with a modified Wallace tree multiplier [6]. The Vedic multiplication method, which employs the Urdhavatiryakbhyam sutra and adder employing carry save adder, is necessary to build the MAC unit [7][8].

The multiplier typically takes up more area and is the slowest element of the whole network. Although it incurs a speed cost, the carry select adder provides high performance in the carry propagation adder design [9]. As soon as possible, partial products generated by the proposed Wallace multiplier are reduced, and then a final carry propagation path decides which adder is employed. The modifier carry select adder (MCSA) has a smaller overall footprint than a standard Wallace Tree-multiplier compared to a conventional carry select adder architecture [5]. It proposes developing Vedic multipliers at high-speed using modified forms of Vedic mathematics for better performance. A particular calculation method utilized in Sixteen Sutras is the base of Vedic mathematics [10]. For many applications, a high-performance multiplier design and real-time low power, low cost, and reduced size requirements are needed. The approach adopted by Vedic multipliers focuses on a high-speed, add-and-shift-based 8X8 Vedic multiplier design, which differs from ordinary multipliers. The 8X8 bit multiplier algorithms' Urdhva-Tiryagdhya sutra is developed, and their FPGA implementation is done through XILINX system tools [11].

The adder is the other primary combinational circuit needed to add the binary integers. The ALU frequently uses adders in processors and computers. The design of the Vedic multipliers depended primarily on adders with efficient power and speed use. Here, some of the adders used in the design of the Vedic multiplier are listed.

#### **Carry Skip Adder (CSA):**

The carry skip adder is replaced with a carry-by-pass adder. A CSA separates the required bits to be added to the blocks. Sum and carry bits are generated in these blocks. The CSA latency decreases by avoiding the continuous adder stages in the carry computations.

#### **Carry Look Ahead Adder (CLA):**

CLA is called a "fast adder" as it defines the carry bits in less time. Hence, it is speed efficient. CLAs undergo three phases. Principal configurations provide the Gi and Pi factors. It offers carry signals for the second phase's last step. The ultimate yield can be seen in the previous stage. In addition to taking less time, as the number of variables rises, the circuit of this adder also grows more complex, increasing production costs since it needs more hardware than a standard adder.

#### **Kogge Stone Adder (KSA):**

KSA is one of the high-speed adders, and it is primarily used for higher-order arithmetic circuits. It consists of three computation phases and is a kind of parallel prefix category of carry Look Ahead adder:

1. Preprocessing: Propagate and generate signals of every input pair are generated.
2. Carry generation stage: Every bit's carry is computed for every bit.
3. Post Processing: In which the sum of all bits is calculated.

It is a fast adder primarily utilized in circuits for higher-order computation. In KSA, carriers are quickly registered by generating them in parallel at the cost of additional space. The primary disadvantage of a Kogge stone adder is that it has a low delay and takes up a vast area.

## **2 LITERATURE SURVEY**

Vijay Pratap Sharma et al. compared all the characteristics of 4-bit, 8-bit, and 16-bit data and discussed the implementation of the MAC unit [2]. LUTs at room temperature and the Vivado HLS software are used to construct and analyze the designs. Hossam O. Ahmed et al., instead of relying on the traditional DSP blocks and embedded memory units on the FPGAs architecture silicon fabrics, proposed an 8-bit fixed-point parallel MAC unit architecture aims to create a fully customized MAC unit for the Convolutional Neural Networks (CNN) [6]. It was observed that the 8-bit fixed-point parallel MAC unit architecture is designed using the VHDL language and can perform a computational speed of up to 4.17 Giga Operation per Second (GOPS) using high-density FPGAs.

Nusrat Jabeen M. Anees discussed the FIR digital filter, which is the device that is used most frequently in DSP systems [12]. As a crucial component of both theoretical development and practical application, this can be accomplished by employing the filter coefficient optimization technique, a single MAC unit, and the parallel form (transpose) of two structures because it involves addition and multiplication, which lowers the hardware cost. The suggested work appears to be a positive step toward digital filter area optimization. Satheesh. K, P. P. Nagaraja Rao highlighted the effect of hybrid carry select adder [13]. It was found that the calculation speed increased by 35.17% compared to RCSLA. Verilog-HDL is used to code the entire design, and the Xilinx ISE12.2 tool is used for synthesis.

Jatin Yadav et al. conducted a performance analysis of various adder topologies such as Carry Look Ahead Adder, Kogge Stone Adder, and Carry Skip Adder using the metrics area, latency, and power [4]. The techniques under consideration are applied to CMOS technology at 45, 90, and 180 nanometres (nm). KSA adder was found to use less power; latency was measured at 606 ns, although the Carry Look Adder's area utilization was just 7 microseconds square. Gopal Raut et al. said that the MAC and AF operations of the CORDIC-based architecture are realized through the application of linear and trigonometric relationships, respectively [7]. As revealed by the dynamic power variation, the worst-case mean for this design is 189.73 $\mu$ W, or 63% of the state-of-the-art. Deepthi Amuru et al. said that when it comes to Very Large-Scale Integration (VLSI) design and testing, the unique learning strategies of artificial intelligence (AI) offer a variety of fascinating automated approaches for managing intricate and data-intensive activities [14].

The time and effort required to comprehend and process data within and across different abstraction levels using automated learning algorithms is decreased when artificial intelligence (AI) and machine learning (ML) algorithms are used in VLSI design and production. Consequently, they address Opportunities and Challenges FOR AI/ML in VLSI as their conclusion. K Praveen Kumar Reddy et al. stated that the Vedic multiplier-based MAC (multiplier and accumulator) unit's performance parameters are compared to the current modified Wallace multiplier-based MAC unit. The multiplier uses the Vedic multiplication algorithm, which employs the Urdhava tiryagbhyam sutra [15]. The adder is completed with a carry-save adder. Verilog-HDL is used to code the entire design. The comparison results show that while the Wallace multiplier uses an area of 5770 slices, the Vedic multiplier uses an area of 5547 slices when using a carry-save adder. Additionally, the Vedic multiplier uses a carry-save adder with a delay of 47.156nS, while the Wallace multiplier uses an area of 5770 slices.

Sumant Mukherjee et al. addressed the introduction of a new, energy-efficient MAC unit that will reduce hardware complexity and improve SPAA (Speed, Power, Area, Accuracy) measures [16]. Accuracy is a crucial factor in digital circuits, in addition to speed, size, and power consumption. According to the overall area, delay, and frequency analysis, which are provided, compared, and illustrated, up to 25–35% percent reduction at all levels is attained.

Table 1. Vedic Sutra and their meanings

Vedic Sutra	Meaning	Brief Explanation
Ekadhikena Purvena	One more than the previous one	In instances in which the number is close to the base, such as 10,100,1000, etc., this sutra is helpful. It is used primarily to find squares of integers close to these bases.
Nikhilam Navatashcaramam Dashatah	All of the 9s and the 10s in the end.	This sutra is useful when deducting a number from a considerable round number like 10,100,1000, etc.
Urdhva-Tirvakdhyam	Vertically and crosswise.	This sutra can be helpful when multiplication numbers by multipliers higher than 10. In addition, it can be utilized to multiply numbers near bases (10,100,1000, etc.).
Paravartya yojayet	Transpose and adjust	The division is achieved using this sutra. It makes division of any kind straightforward.
Shunyam saamyasamuccaye	The sum is 0 when the sum is the same.	This method is utilized when solving simultaneous linear equations using the same variable coefficients. The coefficients in the variables of these equations add up to 0 (or do not have a difference).
Anurpye sunyamanyat	The other one is zero if one is in ratio.	They are used for solving linear equations concurrently. When one variable's coefficient possesses the same ratio to one another as the independent terms, this sutra is applicable.
Sankalana vyavakalana bhyam	By adding and subtraction	The co-efficient "X" & "Y" are changed in the class of simultaneously linear equations, which are solved using this sutra.
Purana punara bhyam	By the completion or non-completion	This strategy helps us find all the data before solving the problem when the provided data is incomplete.
Chalana kalan bhyam	Differences and similarities	This sutra is primarily used to solve differentiation-related issues.
Yaavadunam	As much as the extend	This sutra is beneficial when performing "O" operations. It is advantageous when multiplying.
Vyashtisa manstih	Part and whole	This is used when one number is far from the base to get the product of two numbers. In addition, it helps with statistics and solutions of complicated numbers.
Shesanyankena charamena	The reminders by the last digit	This one helps in identifying division problem reminders. When we divide a number by 9, we use it.
Sopaantyada yamantyam	The ultimate and twice the penultimate	It helps multiply a number whose last digits add up to 10; the remaining digits are the same.
Ekanyunane purvena	One less than the previous one	This sutra is used when multiplying digits whose last digits sum up to 10 and the preceding numbers are the same.
Gunitasamu chyah	The sum is identical to the product of the sum.	Algebra and geometry difficulties are solved with this sutra.
Gunakasamu chyah	The factor of the sum is the sum of the factor.	Anyone can apply this sutra to factor and solve equations based on algebra.

Table 2. Comparison of different methods based on Vedic Sutras

Algorithm or method	Efficiency based on power	Efficiency based on delay	Efficiency based on area
Urdhva tiryagbhyam with different adders CLA, KSA, Carry Skip Adder	<b>180nm CSA</b> Power=15668.51nW <b>180nm CLA</b> Power=16926.73nW <b>180nm KSA</b> Power=33277.94Nw	<b>180nm CSA</b> Delay= 4417nS <b>180nm CLA</b> Delay= 3226nS <b>180nm KSA</b> Delay= 1063nS	-
Annurupye sutra	-	Delay=38.6ns (64X64 bit)	-
Karat suba algorithm and Urdhva tiryagbhyam algorithm	-	Delay=11.514ns (16X16 bit)	-
An adaptive multiplication unit was designed based on four sutras: Ekanyunena purvena, Anurupyena, Antyavor-Dasakepu, Urdhvatiyagbhyam, and a Control unit.	-	Delay= 17.59nS	-
UT with Carry Save Adder	-	Delay=47.156ns	Area=5547ns
UT algorithm with compressor & CLA adder	-	-	The area is represented by the number of LUTs used (679)
Nikhilam sutra with reversible logic	Power= 82 mW	Delay = 8.73ns	-
UT algorithm for multiplication and carry save adder for accumulation	-	Delay= 30.659 ns (16X16 bit)	-
Urdhva tiryagbhyam algorithm	-	Delay =6.4 65ns (32 X32 bit)	-
Urdhva tiryagbhyam with reversible adder	Power=82mW (8X8 bit)	Speed=145.03MHZ	Area=180

### 3 STRUCTURES AND IMPLEMENTATION OF VEDIC MULTIPLIERS

This MAC unit operates on the following principle. The inputs are retrieved from memory and sent into the multiplier block of the MAC, which multiplies the inputs and sends the result to the adder, which adds the result and stores it in memory. It is intended to complete this operation in one clock cycle. As seen from Fig., the MAC unit architecture comprises an 8-bit register, a 4-bit Vedic multiplier, and an 8-bit accumulator that uses a carry look-ahead air adder (CLA). The MAC unit's structure is depicted in Fig. 1.

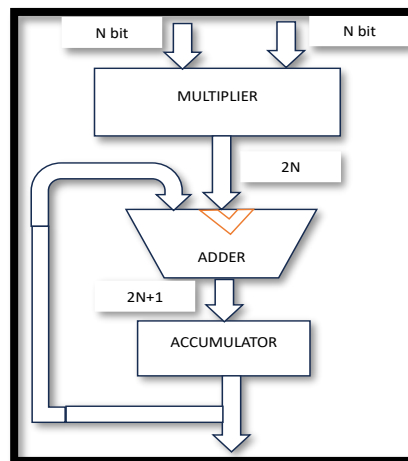


Fig. 1. Block Diagram of Proposed Vedic Multiplier-based MAC

### 3.1. 2x2 Vedic Multiplier

To understand this approach, let's look at two numbers, A and B, each having two bits. As can be seen in the line diagram below,  $A=a_0a_1$  and  $B=b_0b_1$ . The result's least significant bit (LSB) bit (vertical) is first determined by multiplying the two LSB bits of A and B together, which equals  $a_0b_0$ . The second step involves calculating the products crosswise so that the MSB of the second input and the LSB of the first input are multiplied and vice versa. Every multiplication produces two bits as output: a carry bit sent as Carry Input ( $C_{in}$ ) for each multiplication and a sum bit used in the result. The outcome is thus obtained, and the Output Carry creates the fourth bit. The 2x2 Vedic Multiplier's block diagram is displayed in Fig. 2.

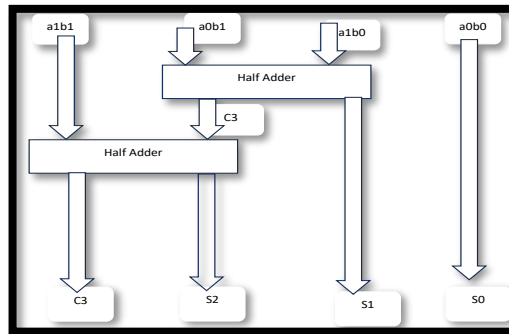


Fig. 2. Line diagram of 2x2 Vedic Multiplier

### 3.2. 4x4 Vedic Multiplier

The 4,2x2 multiplier blocks of this 4x4 bit multiplier are used to create the output, which is 8 bits in size, whereas the multiplicands are of bit size ( $n=4$ ). For both inputs, a and b, the input is divided into smaller pieces of size 2x2. These freshly created 2-bit chunks are fed into a 2x2 multiplier, producing 4-bit as the output. As shown in Fig. 3, the 4-bit output from 2x2 multiplier blocks is then delivered to an addition tree for addition. Similarly, the 32x32 design is carried out. Fig. 4 illustrates the equivalent 32-bit Vedic multiplier. The flowchart is shown in Fig. 5.

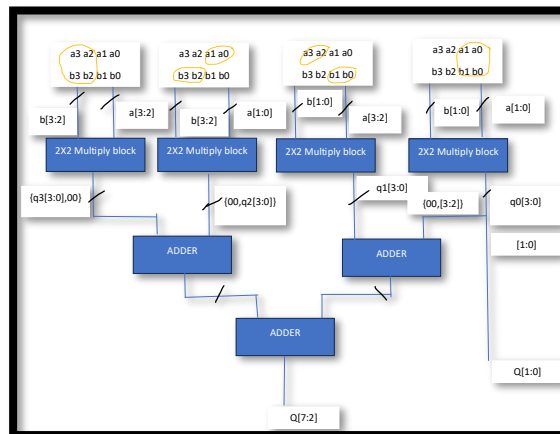


Fig. 3. 4x4 Vedic Multiplier

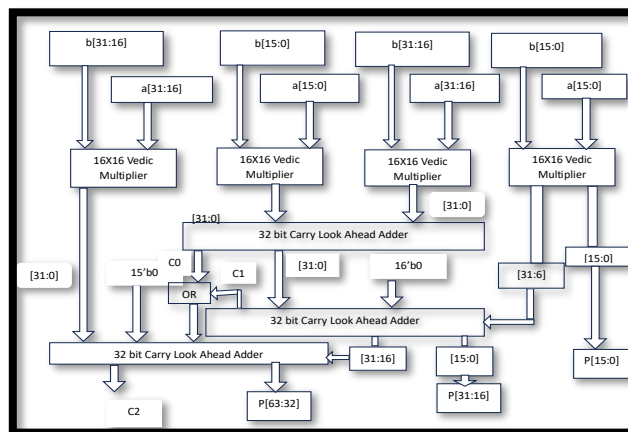


Fig. 4. 32-bit Vedic Multiplier

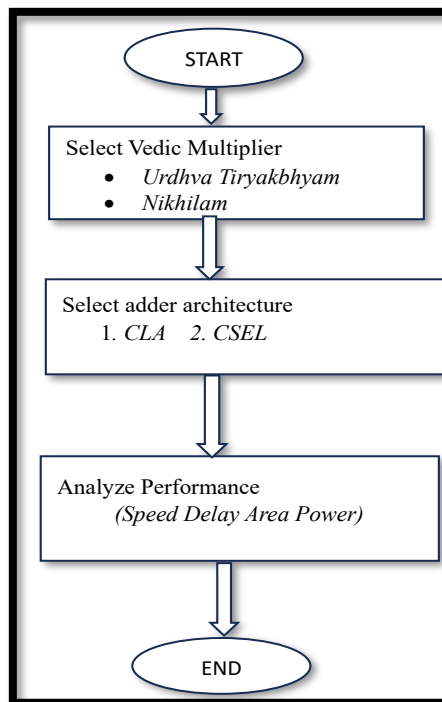


Fig. 5. Flow Chart of Generalized Vedic Multiplier

The comparison of the Nikhilam sutra and Urdhva triyakbhyam is given in Table 3.

Table 3. Performance analysis of Nikhilam sutra and Urdhva triyakbhyam

Model	Nikhilam sutra	Urdhva triyakbhyam
Number of slice registers	64	64
Number of slice LUTs	392	2547
Delay(s)	3.499	3.543
Power (w)	0.01652	0.01650

#### 4 CONCLUSIONS

The study concludes that Vedic multiplier and adder architecture in designing a 32-bit MAC unit depends on the application's priorities, balancing speed and resource efficiency. The Urdhva Tiryakbhyam multiplier combined with a Carry Look Ahead (CLA) adder achieved the fastest performance, with a minimal delay of 0.709 ns, though at the cost of higher resource utilization. On the other hand, the Nikhilam multiplier generally required fewer Logic LUTs, making it more resource-efficient, albeit slightly slower. CLA proved to be the most efficient adder regarding speed across both Vedic multipliers, while alternatives like the Kogge Stone Adder (KSA) offered a good balance between speed and area. The analysis highlights that the choice of multiplier and adder architecture should be guided by specific design requirements, whether prioritizing speed, power, or area in the given application. The configurations evaluated in this study are viable for practical implementation using Xilinx ISE tools, providing a foundation for further optimizations in future MAC unit designs for applications such as DSP and deep learning models.

#### FUNDING INFORMATION

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

#### ETHICS STATEMENT

This study did not involve human or animal subjects and, therefore, did not require ethical approval.

#### STATEMENT OF CONFLICT OF INTERESTS

The authors declare no conflicts of interest related to this study.

#### LICENSING

This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).

## REFERENCES

- [1] A. Malhotra and A. Singh, "Implementation of AI in the field of VLSI: A Review," *2022 Second International Conference on Power, Control and Computing Technologies (ICPC2T)*, Mar. 2022, doi: 10.1109/icpc2t53885.2022.9776845.
- [2] Vijay Pratap Sharma and Hemant Patidar, "Analysis and Implementation of MAC Unit for Different Precisions," *ICTACT Journal on Microelectronics*, vol.7, Issue.4, pp. 1260-1264, 2022, doi: 10.21917/ijme.2022.0217.
- [3] Y. Goh, D. Jung, G. Hwang, and J.-M. Chung, "Consumer electronics product manufacturing time reduction and optimization using AI-Based PCB and VLSI circuit designing," *IEEE Transactions on Consumer Electronics*, vol. 69, no. 3, pp. 240–249, Aug. 2023, doi: 10.1109/tce.2023.3240249.
- [4] J. Yadav, A. Kumar, S. Shareef, S. Bansal, and N. Rathour, "Comparative Analysis of Vedic Multiplier using Various Adder Architectures," *Journal of Physics Conference Series*, vol. 2327, no. 1, p. 012022, Aug. 2022, doi: 10.1088/1742-6596/2327/1/012022.
- [5] Sulbha Bhongale, Yogesh Khandare, Santosh Bobade, "Review on Recent Advances in VLSI Multiplier," *International Journal of Engineering Research & Technology*, vol. 10, Issue. 11, 2021, doi: 10.17577/IJERTV10IS110016
- [6] H. O. Ahmed, M. Ghoneima and M. Dessouky, "Concurrent MAC unit design using VHDL for deep learning networks on FPGA," *2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, Penang, 2018, pp. 31-36, doi: 10.1109/ISCAIE.2018.8405440.
- [7] G. Raut, S. Rai, S. K. Vishvakarma and A. Kumar, "RECON: Resource-Efficient CORDIC-Based Neuron Architecture," *IEEE Open Journal of Circuits and Systems*, vol. 2, pp. 170-181, 2021, doi: 10.1109/OJCAS.2020.3042743.
- [8] Savita Sonoli and Sahana Desai, "Design and Implementation of 4x4 Vedic Multiplier using Cadence," *International Journal of Engineering Research and Technology*, vol. 9, issue. 7, July 2020.
- [9] Akshay Savji, Shruti Oza, "Design and Implementation of Vedic Multiplier," *International Journal of Recent Technology and Engineering*, vol.8, issue.6, March 2020, doi: 10.35940/ijrte.F8808.038620.
- [10] Swathi Dayanand, Varshitha K R, Rohini T, Yasha Jyothi M Shirur, Jyoti R Munavalli, "Low Power High Speed Vedic Techniques in Recent VLSI Design – A Survey," *pices*, vol. 4, no. 6, pp. 147-156, Oct. 2020. doi: 10.5281/zenodo.4247825.
- [11] M. Akila, C. Gowribala and S. M. Shaby, "Implementation of high-speed Vedic multiplier using modified adder," *International Conference on Communication and Signal Processing*, Melmaruvathur, India, 2021. doi: 10.1109/ICCSP.2016.7754093.
- [12] Nusrat Jabeen M Anees, Ashish B. Kharate, "Design of area efficient digital FIR filter using MAC," *International Research Journal of Engineering and Technology*, vol.4, issue.9, Sep 2017, doi: 10.35940/ijrte.F8808.038620.
- [13] Satheesh K, P P Nagaraja Rao, "Implementation of High-performance 64-bit MAC Unit using Hybrid Carry Select Adder," *International Journal of Scientific Engineering and Technology Research*, vol.5, issue.7, pp. 1396-1399, Mar 2016.
- [14] D. Amuru *et al.*, "AI/ML algorithms and applications in VLSI design and technology," *Integration*, vol. 93, p. 102048, Nov. 2023, doi: 10.1016/j.vlsi.2023.06.002.
- [15] K Praveen Kumar Reddy, S. Aruna Mastani, "Implementation of High-Performance 64-bit MAC Unit for DSP Processor", *International Journal of Current Engineering and Scientific Research*, vol.2, issue.11, pp. 20-26, 2015.
- [16] Sumant Mukherjee, Saurabh Mitra, "Energy Efficient Approximate MAC Unit for High-Speed DSP Application," *International Journal of Science and Research*, vol.4, issue 6, pp. 423-428, June 2015.