

A Novel High Performance Architecture for Mac Unit Using Vedic Multiplier and Brent-Kung Adder

Dr. S. Ravi., M.S.N.V. Mohith., K. Yaswanth Simha., L. Alekhya., M. Maruthi Sriram

Department of Electronics and Communication Engineering, India

DOI: <https://doi.org/10.51244/IJRSI.2025.12040021>

Received: 01 April 2025; Accepted: 08 April 2025; Published: 30 April 2025

ABSTRACT

The DSP industries make use of Multiply and Accumulate (MAC) units in their systems unanimously. As the name implies, MAC unit performs both Multiplication and addition operations. The proposed MAC unit make use of parallel prefix adder, instead of ripple carry adder, and hence there is an improvement in the performance of the DSP processors. Further, this paper also examines the performance by using Brent-Kung adder, one of the high-speed adders which is used to reduce the delay of MAC units. To enhance the performance of multiplication process, the proposed design uses a Vedic multiplier, based on Urdhva Tiryagbhyam (UT) sutra. Verilog HDL is used to do the analysis of the MAC unit and Xilinx ISE 14.7 is used to simulate and synthesis the MAC unit.

Keywords: DSP, MAC, Parallel Prefix Adder, Brent Kung Adder, Vedic multiplier, Xilinx ISE 14.7.

INTRODUCTION

Power consumption of a DSP processor depends majorly on the power consumption of Mac unit in it [1]. A digital circuit that adds numbers is called an adder. Traditional adders, such as Ripple Carry Adder (RCA) and Carry Look-Ahead Adder (CLA), suffer from high propagation delay and increased complexity with larger bit widths [2]. Similarly, conventional multipliers like the Booth and Array multipliers have latency issues due to their sequential nature. To overcome these limitations, the Brent-Kung Adder (BKA) and Vedic Multiplier (VM) have been integrated into MAC units to enhance speed and reduce the delay. Two single bit binary values A and B can be "added" together using a simple binary adder circuit that can be created using regular AND and Ex-OR gates [3]. According to the rules for binary addition, the addition of these two digits results in two outputs are the sum and Carry-out (COUT) bit[12].

Compared to prior designs, a MAC unit utilizing Brent-Kung Adder and Vedic Multiplier exhibits reduced delay, lower power consumption, and higher throughput, making it ideal for real-time applications. In the existing network of MAC unit utilizes the Vedic multiplier with the carry look ahead adder. It provides the delay of 27.384ns. Therefore to reduce the delay we can move further design of MAC unit using Vedic multiplier with brent kung adder. Because of in now a days mostly designing of devices with low power and less area which improves the performance of system [5]. So therefore to reduce the delay it is very important parameter in the any device [2]. There is a high requirement of low power consuming devices. The significant adders and multipliers performs a very important role in the MAC unit [15].

Existing work:

By using the basic adders like half adders, full adders and ripple carry adders we can add at a time only two bits therefore it takes longer time for add more number of inputs. To overcome this problem then we can go to the parallel adders.

A Carry look-ahead adder is an advanced digital circuit designed to perform fast binary addition by reducing the delay in case of carry propagation [9]. Unlike a ripple carry adder, where the all full adder must wait for the previous carry to compute the next sum, a CLA calculates carry signals in advance using generate and propagate functions [4]. This significantly speeds up the addition process, making it ideal for high-speed arithmetic operations in processors and digital circuits.

The working of a CLA relies on the carry generation and carry propagation terms, which are derived from the binary inputs [4]. The generate term (G) indicates that a carry will be produced regardless of the previous carry, while the propagate term (P) shows that a carry will be passed to the next stage if a carry is received [3]. By using these terms, the CLA can compute carry values directly using combinational logic, eliminating the need for sequential carry propagation as seen in RCAs.

Proposed work:

To provide better performance in Very Large Scale Integration (VLSI) design we can use the parallel prefix adders [3]. The parallel prefix adders requires the execution of the operation can be done in parallel which can be produced via segmentation into smaller portions. The parallel Prefix Adder (PPA), which is used in Very Large Scale Integration (VLSI) circuits, is a very useful technology in the modern world. The VLSI chips heavily rely on accurate and quick processing of arithmetic operations [1]. PPA is able to contribute to these contributions. PPA comes in a variety of forms, including Brent-Kung, Kogge Stone, Ladner Fisher and Hans CarlsonI. In this we can consider the Brent Kung Adder.

They are employed to handle binary additions due of their adaptability. Keep Your Head Up It uses the Adder's (CLA) structure to obtain the parallel prefix adders. To accelerate arithmetic operations, tree structure algorithms are utilised. For better performance arithmetic circuits the parallel prefix adders are employed since they speed up operation. The brent kung adder is more advanced than other adders in parallel prefix adders. We can reduce the latency by utilising this adder.

Block diagram of Brent-kung adder:

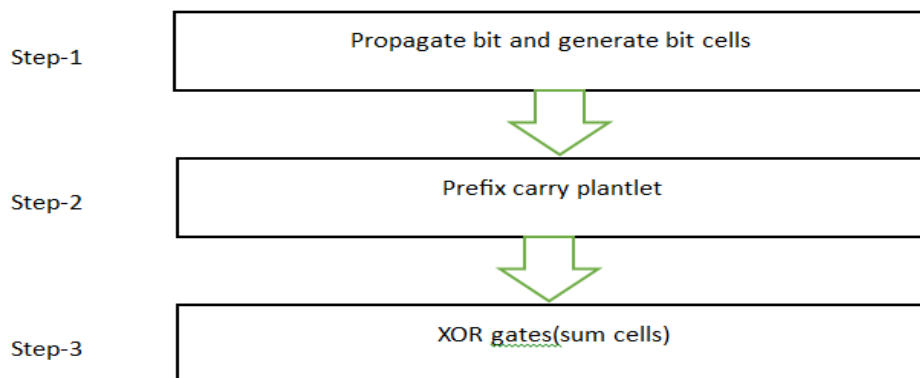


Fig:4.1.1 Block diagram of Brent-Kung adder

The three stages for to build the Brent Kung Adder

- pre-processing stage
- carry computation stage
- post-processing stage

Preprocessing Stage:

In this stage, the adder computes the Generates and Propagates signals for each bit of the input binary numbers. These signals help determine whether a bit position will produce or pass a carry. The formulas for these computations are:

- Generate (G): $G_i = A_i \cdot B_i$ (Carry is generated if both bits are 1)
- Propagate (P): $P_i = A_i \oplus B_i$ (Carry is propagated if at least one of the bits is 1)

Each bit position calculates its own G and P values independently in this step.

Prefix Stage (Carry Computation):

This stage uses a hierarchical structure to compute the carry signals efficiently. Instead of computing all carry signals sequentially in the Ripple Carry Adder, therefore in the Brent-Kung adder groups and processes them in a tree-like manner.

Instead of waiting for the previous carry to propagate, we compute all carries in parallel using the formulas:

$$C1=G0+(P0\cdot C0)$$

$$C2=G1+(P1\cdot C1)$$

$$C3=G2+(P2\cdot C2)$$

$$C4=G3+(P3\cdot C3)$$

Since these computations happen in parallel using combinational logic, the delay is significantly reduced compared to a ripple carry adder.

Post-Processing Stage (Sum Computation):

Once the carry signals are available from the prefix stage, the final sum bits are processed using the formula is expressed below:

$$S_i = P_i \oplus C_{i-1}$$

Since the carry values were computed efficiently in the previous step, the sum bits can now be determined in parallel, completing the addition process with minimal delay.

Architecture of brent-kung adder:

In comparison to the Kogge Stone adder in the brent kung adder it has less wiring congestion and a higher degree of regularity in this type of adder structure, which improves performance. Moreover it is more faster than the ripple carry adders. The architecture of brent kung adder is shown below.

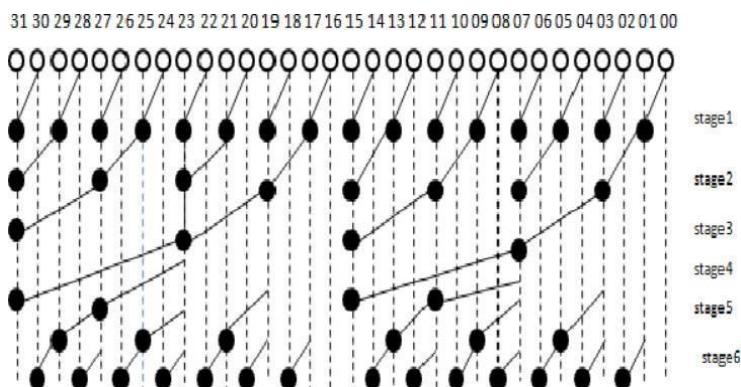


Fig: 4.2.1 Architecture of brent kung adder

Designing of MAC using brent kung adder:

A MAC unit is the basic and the fundamental component in digital signal processing and machine learning applications. It performs multiplication of two numbers followed by an addition operation, making it essential for applications requiring high-speed arithmetic computations [3]. The efficiency of a MAC unit heavily depends

on the design of the adder used for accumulation, as addition is a critical operation in the computation pipeline [4].

One of the most efficient adders for high-speed arithmetic unit is the Brent-Kung adder, a parallel prefix adder known for its minimal fan-out and structured carry propagation [2]. It significantly reduces the number of logic levels required for addition, leading to improved speed and reduced power consumption compared to conventional adders [10]. Integrating a Brent-Kung adder into a MAC unit enhances performance by optimizing the accumulation step [7]. The design involves a partial product generation phase, followed by the summation of these partial products using a fast adder [3]. The Brent-Kung adder helps in efficiently propagating carry signals, ensuring minimal delay in the accumulation process [3].

The design of a MAC unit using a Brent-Kung adder consists of three primary stages: multiplication, addition, and accumulation [1]. The multiplication stage uses an array or tree multiplier to generate partial products. These partial outputs of products are then summed using a combination of carry-save adders (CSAs) [2] and the Brent-Kung adder for final addition. This parallel prefix structure enables logarithmic time complexity for carry propagation, reducing the overall critical path delay. This results in a high-speed MAC unit suitable for applications requiring low latency and high throughput, such as image processing, neural networks, and real-time signal analysis [8].

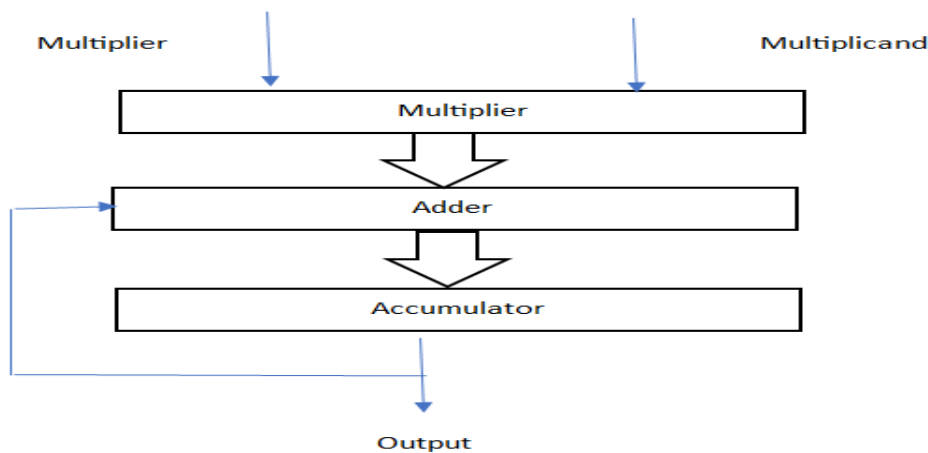


Fig:4.3.1 Architecture of basic MAC unit

For the 32-bit MAC unit's design we can use the 16 bit multipliers. It was created using 8-bit multipliers [14]. Similar to this, by using the 4 bit multiplier we can design the 8-bit multiplier, and a 2-bit multiplier is used to create a 4 bit multiplier [11].

Multiplier design by using vedic sutras:

It is possible to immediately apply Vedic mathematics, which is a very old method, to many other areas of mathematics, including algebra and arithmetic. By eliminating the stages that are not necessary for calculating any outcome, complexity is reduced. In vedic mathematics, there are 16 sutras. The various list of sutras is presented in below table.

Table-5.1 Vedic multiplier sutras

S. No	Sutras
1	(Anurupye) Shunyamanyat
2	Chalana-Kalanabhyam
3	Ekadhikena Purvena

4	Ekanyunena Purvena
5	Gunakasamuccayah
6	Gunitasamuccayah
7	Paraavartya-Yojayet
8	Puranapurāṇabhyam
9	Sankalana-vyavakalanabhyam
10	Shesanyankhena-Charamena
11	Sopantyadvayamantyam
12	Urdhva-Tiryakbhyam
13	Vyashtisamastih
14	Yavadunam

Only two of the 16 sutras mentioned above—Urdhva Tiryakbhyam(UT) and Gunitasamuccayah can be used to the multiplication of any two numbers [2]. In this MAC unit we caan utilized the UT sutra. Urdhva Tiryakbhyam (UT) sutra is defined as "vertically and cornerwise"[8]. Two figures are added base-neutrally using this method. Consider the partial product produced by the multiplication of two 3-bit values, such as U (20) and V (20), with C (30) acting as the carry and Y (20) as the output. Moreover, the following conduct must be followed [18].

Step 1: $C_0Y_0 = U_0V_0$

Step 2: $C_1Y_1 = (U_0*V_1) + (U_1*V_0) + C_0$

Step 3: $C_2Y_2 = (U_0*V_2) + (U_1*V_1) + (U_2*V_0)$

Step 4: $C_3Y_3 = (U_1*V_2) + (U_2*V_1) + C_1$ $C_4Y_4 =$

Step 5: $C_4Y_4 = (U_2*V_2) + C_3$

Hence, the final result is $C_4Y_4Y_3Y_2Y_1Y_0$.

Architecture of MAC unit designed by using Brent-Kung adder and vedic multiplier:

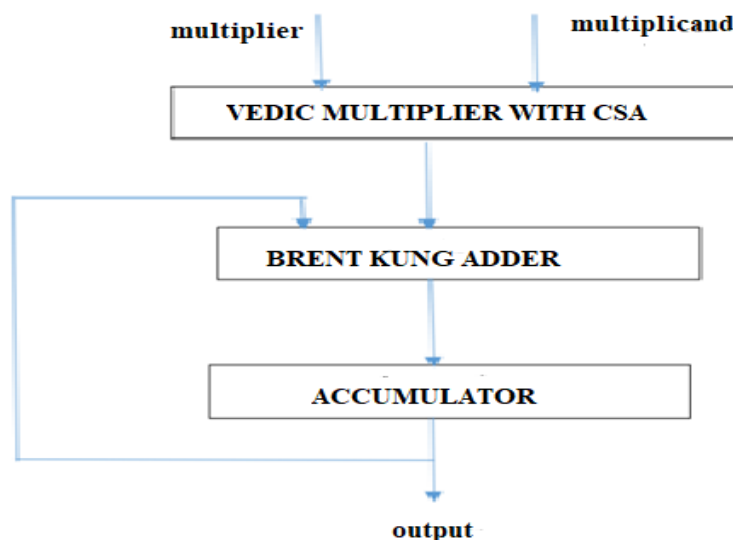


Fig:5.2.1 Architecture of MAC unit

The above block diagram shows the architecture of designed MAC unit. The MAC unit first performs multiplication using the Vedic Multiplier [4], which leverages the Urdhva-Tiryagbhyam (vertically and crosswise) algorithm to generate partial products in parallel, significantly reducing delay compared to conventional multipliers like the Booth or Wallace tree multipliers [9]. Once the multiplication is complete, the resultant product is fed into the accumulation stage, where it is added to a previously stored value using the Brent-Kung Adder.

The BKA, a parallel prefix adder, optimizes carry propagation by using a hierarchical tree structure that reduces the number of logic levels required to compute the final sum [7]. This approach minimizes propagation delay and improves throughput, making it superior to traditional adders like the Ripple Carry Adder (RCA) and Carry Look-Ahead Adder (CLA) [8]. By combining the Vedic Multiplier's fast multiplication technique with the Brent-Kung Adder's efficient addition. The implementation of 32-bit multiplier design is shown in the below figure.

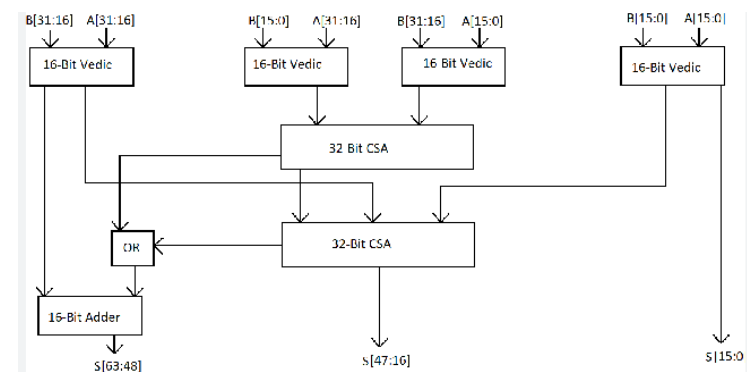


Fig: 5.2.2 Design of 32-bit multiplier based on UT sutra

The above block diagram shows the design of 32 bit multiplier. For the multiplication of two 32 bits we can use the 16 bit vedic multiplier [12]. In the first case we can multiply the two lower words of A and B. Then again multiply lower word of A with higher word of B bit [4]. Therefore like this perform another two cases also then we have four partial product outputs [9]. For the addition of product outputs we can use the 32 bit carry select adders by using this adders we can get the final 64 bit output [18].

RESULTS

Output Waveforms:

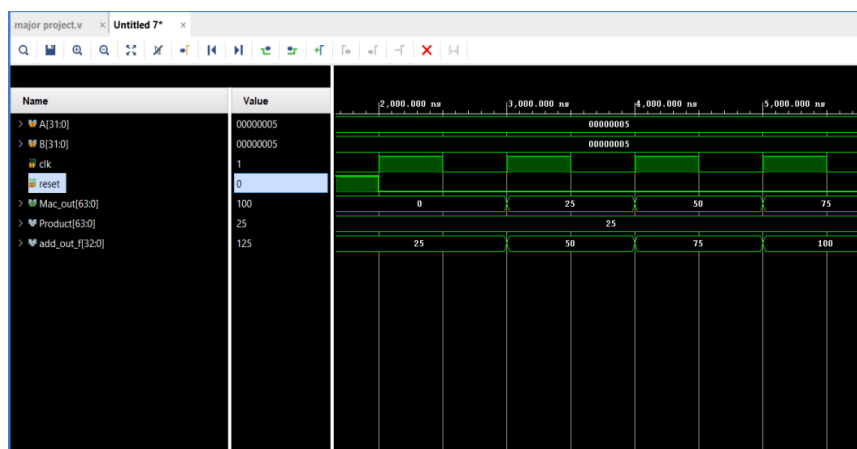


Fig:6.1.1 Output Waveforms for proposed MAC unit

The above figure shows the simulation result of MAC unit utilizes the Vedic multiplier and brent kung adder. We are giving the two inputs A and B then we get the result of product out and the add with the previous data called MAC output then we can get the final output.



Fig:6.1.2 Output waveform for Vedic Multiplier and carry look ahead adder

The above figure shows the simulation result of MAC unit utilizes the Vedic multiplier and carry look ahead adder. We are giving the two inputs A and B then we get the result of product out and the add with the previous data called MAC output then we can get the final output.



Fig:6.1.3 Output waveform for array multiplier and ripple carry adder

The above figure shows the simulation result of MAC unit utilizes the array multiplier and ripple carry adder. We are giving the two inputs A and B then we get the result of product out and the add with the previous data called MAC output then we can get the final output.

View RTL schematic:

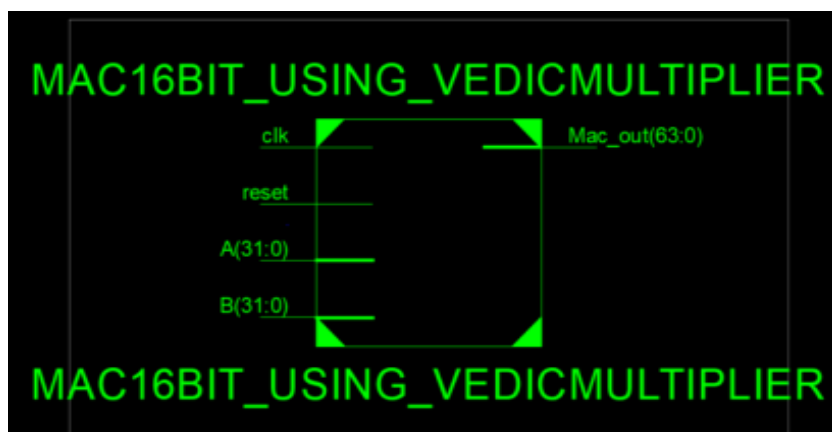


Fig:6.2.1 RTL schematic

Seeing the internal modules requires opening the window with the top module. simply choose the top module.

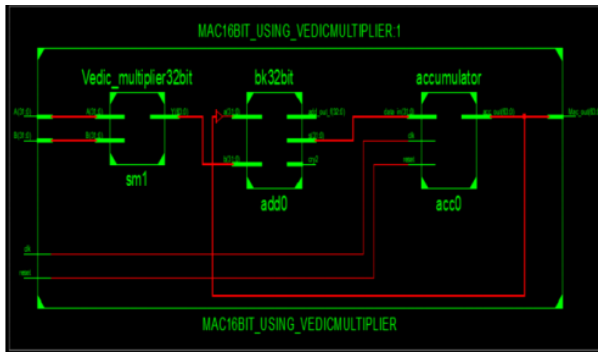


Fig: 6.2.2 RTL schematic internal module

Comparative Analysis of Different MAC Architectures in TERMS OF DELAY

S.NO	Architecture	Delay in ns
1.	32-bit MAC unit using vedic multiplier and carry look ahead adder	27.384
2.	32-bit MAC unit using vedic multiplier, Brent-Kung adder	16.094
3.	32-bit MAC unit using array multiplier, ripple carry adder	42.840

Fig:6.3.1 Table for different MAC architectures

The above table represents the various delays by using the different architectures. By using the array multiplier and ripple carry adder we can get the delay of 42.840ns. If any device run with high speed then the delay is can be very small. That's why we can go to further MAC unit that can be designed by using the Vedic multiplier with carry look ahead adder in this we can get the delay of 27.384ns. This delay is also very high amount that's why we can go to the design mac unit with the Vedic multiplier and brent kung adder. By using this architecture we can get the delay of only 16.094ns. Compared to above architectures the delay is less in this architecture therefore the speed is also can be improved.

CONCLUSION

In this study, the Vedic multiplier and the Brent-Kung adder were used to build a 32-bit (MAC unit). For this designed MAC unit we can get the delay of 16.094ns. Therefore compared to the other existing MAC unit the delay is reduced. That's why by using this architecture we can improve the system performance and utilized in high speed applications [3]. Vedic multipliers are created utilising carry save adders that are based on the Urdhva Tiryagbhyam(UT) sutra, and verilog HDL is used for simulation. Further we can design the MAC unit by using another multiplier for different sutras like named as the Ekanyunena purveena sutra and Vyashtisamastih sutra [16] then we can check the other parameters like speed and delay.

REFERENCES

1. K.Lilly, S.Nagaraj, B. Manvitha, K.Lekhya Analysis of 32-Bit Multiply and Accumulate unit (MAC) using Vedic Multiplier (2020).
2. K. Bharghava ram dinesh, R Vinoth, M.V.R. Kasyap Design and implementation of high speed 32 bit mac unit (2023).
3. B. Raghavaiah, M. Naga Priya, P. Dhanumjaya , P.Kumar Sai A Novel Architecture for Multiplier and Accumulator unit by using Hybrid Parallel Prefix Brent Kung Adder (2023).
4. Bittu, suman Dahiya Implementation of 64 bit mac unit with different adder circuits (2018).
5. N. R. Nagarajan, T. Muruganantham, and S. Rajapriya's "A New Architecture for Multiplier and Accumulator unit by Using Parallel Prefix Adders" (2019).

6. C. Liu, J. Han, and F. Lombardi's A Low-Power, High-Performance Approximate Multiplier with Configurable Partial Error Recovery (2014).
7. Honglan Jiang For Low-Power and High-Performance Operation, Approximate Radix-8 Booth Multipliers (2015).
8. K. Golda Hepzibha; C P. Subha A novel implementation of high speed modified brent kung carry select adder 2016 10th International Conference on Intelligent Systems and Control (ISCO)
9. Neethu johny, Divya rajan Design and implementation of a multiply and accumulate unit (2019).
10. J. Grad; J.E. Stine A hybrid Ling carry-select adder Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers, 2004.
11. Nagaraj, S; Reddy, GM Sreerama; Mastani, S Aruna; Analysis of different Adders using CMOS, CPL and DPL logic 2017 14th IEEE India Council International Conference (INDICON) 438362017 IEEE
12. S. Nagaraj, Dr.G.M. Sreerama Reddy and Dr.S. Aruna Mastani; A Comparative Study on Different Multipliers-Survey Journal of Advanced Research in Dynamical and Control Systems 14739-7522018 Institute of Advanced Scientific Research
13. M.Pushpa, S. Nagaraj, Design and Analysis of 8-bit Array, Carry Save Array, Braun, Wallace Tree and Vedic Multipliers, IEEE Sponsored International Conference on New Trends in Engineering & Technology (ICNTET 2018).
14. A. N. Gadakh and A. K. Khade's Design and Optimization of 1616 Bit Multiplier Using Vedic Mathematics (2016)
15. K. Bathija, R., S. Meena, R., S. Sarkar, and Rajesh Sahu, "Low Power High Speed 16x16 bit Multiplier Using Vedic Mathematics" (2012)
16. Levent Aksoy, Cristiano Lazzari, Eduardo Costa, Paulo Flores, and José Monteiro, "Design of Digit-Serial FIR Filters: Algorithms, Architectures, and a CAD Tool." (2012)
17. M. V. Durga Pavan, Ramesh S. R., "An Efficient Booth Multiplier Using Probabilistic Methodology" (2018)
18. L. Ranganath, D. Jay Kumar, and P. Shiva Nagendra Reddy's "Design of MAC Unit in Artificial Neural Network Architecture Using Verilog HDL" (2016)