*International Journal on Recent and Innovation Trends in Computing and Communication*
*ISSN: 2321-8169 Volume: 11 Issue: 5*
***Article Received:** 15 February 2023 **Revised:** 02 March 2023 **Accepted:** 30 April 2023*

# Design and Implementation of Hybrid Multiplier Using ZFC

**[1]Shaik Baba Fariddin, [2]Dr. Rahul Mishra**
[1]Research Scholar, Department of Electronics & Communication Engineering, Dr. A.P.J. Abdul Kalam University, Indore, M.P, India
[2]Research Guide, Department of Electronics & Communication Engineering, Dr. A.P.J. Abdul Kalam University, Indore, M.P, India
Corresponding Author Email : sbabafariddin@gmail.com

**ABSTRACT:** The field of research has recently been driven to build systems with low power consumption and high speed due to the increasing number of portable devices. The rapid development of semiconductor technology has contributed to a growing need for portable and embedded digital signal processing (DSP) devices. All DSP applications, multipliers are essential components. For high speed DSP, low power, high speed multipliers are therefore required. All current commercial DSP processors have at least one dedicated multiplier unit since the capacity to compute at a quicker pace is necessary to achieve excellent performance in many DSP and graphic processing algorithms. Numerous researchers have developed a number of multipliers, including modified Booth multipliers, array, Booth, carry save, and Wallace tree. However, today’s computational circuits such as high performance processors, digital signal processing, and cryptographic algorithms require highly effective and speed multipliers. Hence, In this work, Design and Implementation of Hybrid Multiplier using ZFC (Zero Finding Logic) is presented. This Hybrid Multiplier is the combination of Finite Field Multiplier and Modified Kogee Stone Multiplier. The Zero Finding Logic is used to identify the zeros from the resultant product.

**Keywords:** Multiplier, Digital Signal Processing, Power Consumption, Zero Finding Logic (ZFL).

## I. INTRODUCTION

The most crucial part in modern embedded computer systems is the arithmetic unit. Trigonometric functions and both fixed- and floating-point arithmetic operations are typically included in an arithmetic unit. The most crucial hardware components of a complex arithmetic unit are multiplier units [1].

The multiplier unit is capable to perform operations on the operands of different data types such as calculation of sum of products. The Multipliers are the fundamental building blocks in wide range of computing applications and currently found in many Digital Signal Processing (DSP) applications which includes computation intensive arithmetic functions, or CIFs, functions including convolution, filtering, Fast Fourier Transform (FFT), and multiplication-based operations like Multiply and Accumulate (MAC) and inner product are found in the arithmetic and logic units of microprocessors [2]. There are various algorithms that can be used to implement multiplication, including array, Booth, and modified Booth methods.

As the main building blocks in arithmetic unit, multipliers are used to process the scientific data, however the excessive delay and power consumption of multiplier has gained significant interest from research community [3]. Low power, area efficient, and high performance VLSI (Very Large Scale Integration) designs are essential elements of advanced digital processors. In the rapidly expanding mobile market, smaller dimensions and lower power consumption are becoming essential requirements for digital circuit design in addition to quicker units. In order to increase portability and battery life, mobile electronics must reduce space and power consumption [4].

There are many numerous types of multiplication algorithms accessible today, and they all follow three fundamental steps: final adding, partial product reduction, and partial product generation. Serial, parallel, and serial-parallel multiplications are a few examples of multiplication algorithms [5]. Serial multiplication operates at a lower rate and requires less hardware. High-speed applications use parallel multiplication, and the number of partial products determines the speed. There are various forms of parallel multipliers accessible nowadays, such as array and tree multipliers. Among the available multipliers, the Wallace tree multiplier is rather quick, and for quicker applications, they utilize the carry save technique [6].

In order to process large volumes of data with relatively minimal power and delay, many arithmetic cores operating in parallel are typically used [7]. However, the current multiplier requires more time to execute instructions in high-speed processors that run at higher clock frequencies. The processors used in wireless and portable devices are not designed to handle the current multiplier units, which have higher power consumption. Power conservation is therefore an essential field for development [8].

The multiplier that is used mostly determines the speed of a system, and a system's throughput is significantly increased when the multiplication delay is reduced. A significant amount of partial products are produced during multiplication. It is usually necessary to minimize the computational latency in generating these partial products while building a system. In the context of communication, speed and power consumption are the two primary factors taken into consideration during system design. Its speed is affected by the multipliers, or more accurately the adders, which made up the majority of these systems. A multiplier's or its associated adder's effect on speed increases with complexity [9]. Large bit additions are certain to increase the time needed to generate the output, even though addition operations are simple. The time required to finish addition increases with the amount of bits of addends, which eventually causes the multiplier's throughput to decrease [10].

The multipliers must be high-speed, low-power, and small-in-area in order to be suitable for VLSI implementations [11]. Adders and multipliers are essential for processing data in DSPs and communication systems. Current multiplier methods are not appropriate for DSP and communication systems since they need more energy and time [12]. Therefore, this work presents the Design and Implementation of Hybrid Multiplier using ZFC as an approach to these problems. The following is the structure of the research: The Literature Survey is described in Section II. The section III demonstrates the design and implementation of hybrid multiplier using ZFC. In section IV, the results analysis is assessed. In section V, the conclusion is provided.

## II. LITERATURE SURVEY

O. Maltabashi, Y. Kra and A. Teman et. al, [13] explains the implementation of an affinity-driven multiplier that is physically aware. An affinity-based physical-aware was designed to implement FDP (Fused Dot Product) based on the affinity between the logic gates that makes up the structure of gate level. The CDP (Clustered Dot Product) enabled the place and route tools to cluster gates with goof affinity which leads to less routing congestion and better placement usage. For the purpose of implementing DP calculations with up to 78 multipliers, a 65-nm CMOS (Complementary Metal Oxide Semiconductor) standard cell library was utilized. This resulted in power reductions of up to 63%, 60% lower area, and comparing similar methods based on commercial macros based on post-layout results, speed increases of up to 2.5× were achieved.

Z. Zhang and Y. He et. al., [14] provides a sign-digit-based conditional probability estimation for a low-error, energy-efficient fixed-width booth multiplier. A conditional probability estimate strategy based on sign-digits encoded using Booth code is suggested to increase the precision of the computation. By considering the sign bit of the Booth-encoded multiplier when applying the conditional probability, a symmetric error distribution is created. Consider the simulation results, the suggested multiplier has the highest computation accuracy and uses the least amount of energy each operation. Even better, it works with operand lengths that are not multiples of 4. In contrasting the energy-delay-error product with all of the others over a range of operand lengths, the highest decrease can be as high as 14.8%.

T. Alshawi, A. Bentrcia and S. Alshebeili et. al., [15] explains the Low-Complexity Matrix-Vector Multiplier Design and Implementation for Iterative Methods in Communication Systems. A new matrix-vector multiplier field programmable gate array design is provided, which may be effectively utilized to execute commonly used iterative techniques. The sparse matrix structure and the equal magnitude entries of spreading code matrices are both utilized by the suggested architecture. The timing analysis results and implementation details are supporting and demonstrate that they satisfy the majority of requirements for modern communication systems.

Tripathy S ., Omprakash L. B., Mandal S. K. and Patro B. S. et. al., [16] offers 45nm low power multiplier architectures for high speed computing utilizing vedic mathematics. This paper suggests multiplier architectures of four and eight bits that are based on the Urdhva Tiryakbhyam sutra. Using the Cadence EDA (Electronic Design Automation) tool, the 45 nm CMOS Process technology is used to develop these low power designs. To verify the robustness of the architectures, for every one of the five process corners, a process corner analysis of the recommended designs is offered. This multiplier showed growth in power consumption and propagation speed. The performance increased comes from the new UT (Urdhva Tiryakbhyam) based architecture, which has a shorter data path due to fewer adder units.

N. S. Kumar, S. S.D., S. N.G., S. V. Hande, P. K. Y.G. and K. B.S. et. al., [17] Design of an Area-Efficient Multiplier is presented. The 8*8 hybrid tree multiplier proposed in this research combines the Booth-encoding, Wallace, and Dadda techniques. Xilinx ISE (Integrated Synthesis Environment) 14.7 is used to simulate the design, while Cadence Virtuoso is used to analyze. Compared to the existing multiplier, the suggested multiplier requires 10.4% less area, according to the results.

Jetsdaporn Satansup, Worapong Tangsrirat et. al., [18] explains the CMOS Current Multiplier/Divider at 1.5 V. A CMOS circuit design technique for a small-sized low-voltage current-mode multiplier/divider circuit is described. Its operation at low supply voltage is predicated on the usage of a compact current quadratic cell. The suggested circuit is created and modelled to be utilized in TSMC (Taiwan Semiconductor Manufacturing Company limited) 0.25-µm CMOS technology, utilizing an only 1.5 V supply voltage. A maximum linearity error of 1.5%, a THD (Total

***Article Received:** 15 February 2023 **Revised:** 02 March 2023 **Accepted:** 30 April 2023*

Harmonic Distortion) of less than 2% at 100 MHz (Mega Hertz) a total power consumption of 508μW, and a -3dB small-signal frequency of approximately 245 MHz are also demonstrated by the PSPICE simulation results, which are in accurate accord with the theoretical ones.

D. Govekar and A. Amonkar et. al., [19] explains the creation and application of a hybrid adder-based high-speed modified booth multiplier. The multiplier design uses a novel hybrid adder design that has less delay and takes up less area. The suggested Multiplier design's area, delay, and power difficulties are presented. The suggested Modified Booth Multiplier design offers advantages over the traditional technique that uses a Carry LookAhead Adder, including lower area overhead and critical path delay. It also performs better. ModelSim15.7g was used to model and synthesize the suggested multiplier design using the Xilinx ISE 10.1 design tool. Verilog HDL (Hardware Description Language) is the programming language utilized.

G. G. Kumar and S. K. Sahoo et. al., [20] Using a high speed multiplier in low power, high performance applications In order to reduce the power-delay product of multipliers meant for high-performance and low-power applications, a design of 8 and 16-bit multipliers using fast adders (carry save adder, Brent-Kung adder, and carry-select adder) was used. When compared to conventional multipliers, the suggested Vedic multipliers with fast adders really achieve a significant improvement in delay and power-delay product, according to implementation results.

## III. DESIGN AND IMPLEMENTATION OF HYBRID MULTIPLIER

The Multiplication is included in all digital operations; as a result, the performance of a digital circuit is completely dependent on the performance of the multiplier. As one of the most important hardware structure in complex arithmetic circuits, the multiplier units perform multiplication operations such as one to one operand or one operand to many operands and produce the result as sum of products. The multiplication operations are one of the most crucial operations in digital computer systems, processors as the multiplier is one of the main building blocks in various computing operations. There are various types of multipliers, however the multiplier is need to be improved for the reduction of excessive delay and power consumption. To fulfill this, Design and Implementation of Hybrid Multiplier using ZFC is presented in this work. The Figure 1 shows the block diagram of presented hybrid multiplier.

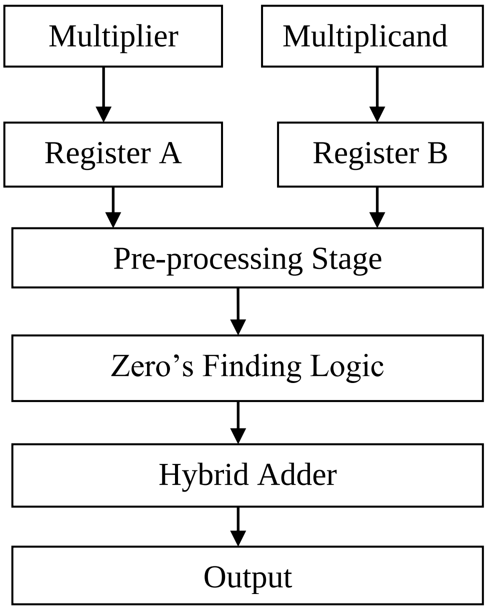

**Figure 1: Block Diagram of Hybrid Multiplier using ZFL**

Nowadays, multipliers are essential in digital signal processing and many other applications. Inputs are interpreted as multipliers and multiplicands in this system. It is possible to differentiate between 1s and 0s using the multiplier and multiplicand. Multiply each multiplier bit by the multiplicand bit position.

Pre-Processing Stage: In this phase, signals are generated and propagated to each pair of register inputs A and B by manipulation. Equations (1) and (2) show the following representations of the propagate signal and create signal:

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

$$G_I = A_i \, AND B_i \quad (2)$$

When the matching multiplier bit is set to "0," all 0s are produced; otherwise, only a portion of the multiplicand. The leftmost bit of each subsequent partial product is moved. By using multiplier bits, all multiples of the multiplicand are produced. The appropriate bit locations of the multiplier determine the variation in the partial product weights.

Finite Field Multiplier: In a finite filed, multiplication is defined as multiplication modulo an irreducible reducing polynomial that defines the filed (that is, division is performed after multiplication using the reducing polynomial as the divisor; the remainder is the product). Carry look-ahead adders with a parallel prefix form are called Kogge-Stone adders. Considered the fastest adder design feasible, it generates the carry signals in O(log2 N) time. In the industry, this architecture is the most widely used for high-performance adders. Due to the removal of unnecessary black cells, the modified Kogge-stone adder operates faster than the original design, making up for the delay. Reducing the unnecessary black cells and rerouting to account for the adder's functionality changed this adder.

***Article Received:** 15 February 2023 **Revised:** 02 March 2023 **Accepted:** 30 April 2023*

The initial zero searching for circuit generates 0 until it finds no zero in the input integer. It produces 1 after identifying the 0 in the input integer. The zeros in the obtained product are found using Zeros Finding Logic (ZFL). With hybrid, these goods are added. In extremely important processing units, the Kogge stone multiplier and finite filed multiplier are utilized in combination to reduce power and area usage.

## IV. RESULT ANALYSIS

In this section, design and implementation of hybrid multiplier using ZFC is implemented. The result analysis of presented adder are simulated and demonstrated here. The Figure 2 shows the layout diagram of Design and Implementation of Hybrid Multiplier using ZFC.

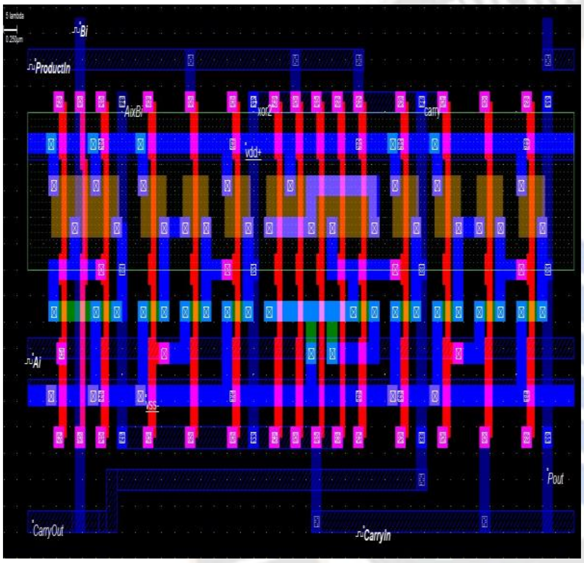

**Figure 2: Simulated Layout of Hybrid Multiplier using ZFC**

The Figure 3 shows the 3D-View of design and implementation of Hybrid multiplier using ZFC.

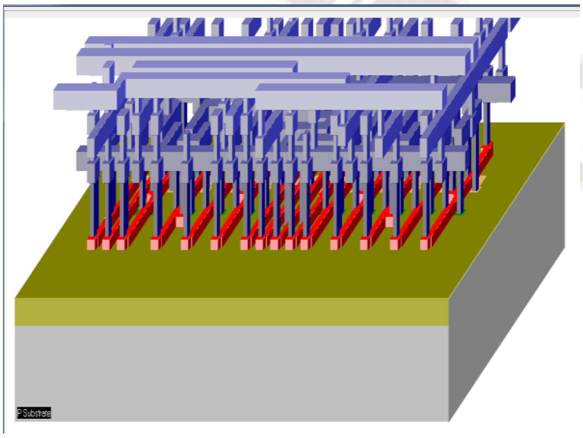

**Figure 3: 3D-View of Hybrid multiplier using ZFC**

The Figure 4 shows the simulated output waveforms of presented hybrid multiplier using ZFL.

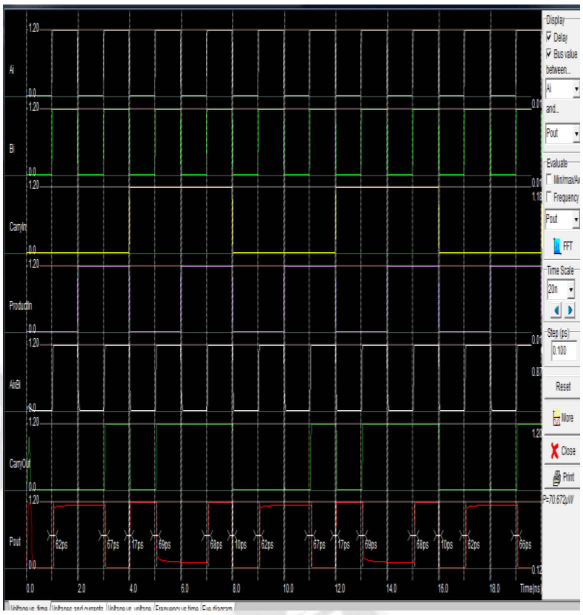

**Figure 4: Output Waveforms**

The Table 1 shows the power performance comparison.

| **Performance metric/Multiplier** | **Wallace Tree Multiplier** | **Hybrid Multiplier using ZFL** |
|---|---|---|
| Power | 82.36 $\mu$W | 70.672$\mu$W |

Compared to Wallace tree multiplier, presented adder has better power. The Figure 5 shows the power comparison.

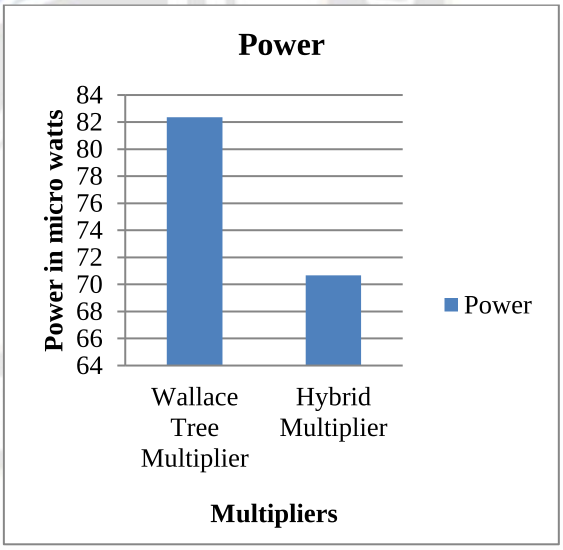

**Figure 5: Power Comparison**

The Hybrid multiplier has obtained better power than Wallace tree multiplier. The Figure 6 shows the delay comparison.

***Article Received:*** *15 February 2023* ***Revised:*** *02 March 2023* ***Accepted:*** *30 April 2023*

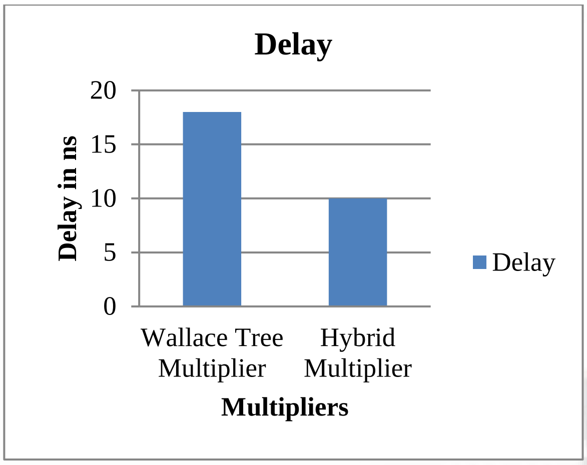

**Figure 6: Delay Comparison**

The delay performance is measured in nano seconds (ns). Compared to Wallace tree multiplier, presented hybrid multiplier has less delay. Therefore, presented Hybrid Multiplier has obtained better power and delay for multiplication operations.

## V. CONCLUSION

Processors are necessary for all digital circuits. Arithmetic and logic operations are carried out by all processors. Multiplication is one of the primary arithmetic operations, out of the four. Multipliers are utilized in multiplication operations. Although multipliers can be implemented using a number of techniques, DSP and other logical functions still require high speed and low power multipliers. Therefore, Design and Implementation of Hybrid Multiplier using ZFC is described in this analysis. A hybrid multiplier is created by combining a finite field multiplier with a modified Kogge stone multiplier. The Zero Finding Logic is used to identify the zeros from the resultant products. The performance presented multiplier is measured and compared in terms of Power and Delay. Compared to earlier multipliers, presented multiplier has obtained better results for multiplication operations.

## REFERENCES

[1] Uppugunduru Anil Kumar, Syed Ershad Ahmed, “Compressor based approximate multiplier architectures for media processing applications,” International Journal of Electrical and Computer Engineering (IJECE), vol. 11, no. 4, pp. 2953-2961, August 2021, doi: 10.11591/ijece.v11i4.pp2953-2961

[2] M. Heidarpur and M. Mirhassani, "An Efficient and High-Speed Overlap-Free Karatsuba-Based Finite-Field *Multiplier for FGPA Implementation," in* IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 29, no. 4, pp. 667-676, April 2021, doi: 10.1109/TVLSI.2021.3058509.

[3] Akshay Savji, Shruti Oza, “Design and Implementation of Vedic Multiplier,” International Journal of Recent Technology and Engineering (IJRTE), vol. 8, no. 16,pp. 3142-3145, March 2020, doi: 10.35940/ijrte.F8808.038620

[4] T. Sai Karthik, K. Manasa, Y.David Solomon Raju, “Design and Implementation of 64 Bit High Speed Vedic Multiplier,” International Journal of Advance Scientific Research And Engineering Trends, vol. 6, no. 1, January 2021, doi:10.51319/2456-0774.2021.0008

[5] V. Muralidharan , N.Sathish Kumar , “Design And Implementation Of Low Power And High Speed Multiplier Using Quaternary Carry Look-Ahead Adder,” Microprocessors and Microsystems, vol. 75, 2020, doi:10.1016/j.micpro.2020.103054

[6] E. Jagadeeswara Rao, K. Jayaram Kumar, Dr. T. V. Prasad, “Design of high speed Wallace tree multiplier using 8-2 and
4-2 adder compressors,” International Journal of Engineering & Technology, vol. 7, no. 4, pp. 2386-2390, 2018, doi: 10.14419/ijet.v7i4.12261

[7] A. N. Bahar and K. A. Wahid, "Design of QCA-Serial Parallel Multiplier (QSPM) With Energy Dissipation Analysis," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 67, no. 10, pp. 1939-1943, Oct. 2020, doi: 10.1109/TCSII.2019.2953866.

[8] Chaitanya CVS, Sundaresan C, P R Venkateswaran, Keerthana Prasad, “Design of modified booth based multiplier with carry pre-computation,” Indonesian Journal of Electrical Engineering and Computer Science, vol. 13, no. 3, pp. 1048-1055, March 2019, doi: 10.11591/ijeecs.v13.i3.pp1048-1055

[9] Ansiya Eshack, S. Krishnakumar, “Pipelined Vedic multiplier with manifold adder complexity levels,” International Journal of Electrical and Computer Engineering (IJECE), vol. 10, no. 3, pp. 2951-2958, June 2020, doi: 10.11591/ijece.v10i3.pp2951-2958

[10] Ansiya Eshack, S. Krishnakumar, “Reversible logic in pipelined low power vedic multiplier,” Indonesian Journal of Electrical Engineering and Computer Science, vol. 16, no. 3, pp. 1265-1272, December 2019, doi: 10.11591/ijeecs.v16.i3.pp1265-1272

[11] Chaitanya CVS, Sundaresan C, P R Venkateswaran, Keerthana Prasad, “ASIC design of low power-delay product carry pre-computation based multiplier,” Indonesian Journal of Electrical Engineering and Computer Science, vol. 13, no. 2, pp. 845-852, February 2019, doi: 10.11591/ijeecs.v13.i2.pp845-852

[12]Minal Keote, P. T. Karule, “Design and Implementation of Low Power Multiplier Using Proposed Two Phase Clocked Adiabatic Static CMOS Logic Circuit,” International Journal of Electrical and Computer Engineering (IJECE), vol. 8, no. 6, pp. 4959-4971, December 2018, doi: 10.11591/ijece.v8i6.pp4959-4971

[13] O. Maltabashi, Y. Kra and A. Teman, "Physically Aware Affinity-Driven Multiplier Implementation," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 39, no. 10, pp. 2886-2897, Oct. 2020, doi: 10.1109/TCAD.2019.2931185.

***Article Received:*** *15 February 2023* ***Revised:*** *02 March 2023* ***Accepted:*** *30 April 2023*

[14] Z. Zhang and Y. He, "A Low-Error Energy-Efficient Fixed-Width Booth Multiplier With Sign-Digit-Based Conditional Probability Estimation," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 65, no. 2, pp. 236-240, Feb. 2018, doi: 10.1109/TCSII.2017.2709801.

[15] T. Alshawi, A. Bentrcia and S. Alshebeili, "Design and Low-Complexity Implementation of Matrix-Vector Multiplier for Iterative Methods in Communication Systems," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 23, no. 12, pp. 3099-3103, Dec. 2015, doi: 10.1109/TVLSI.2014.2387476.

[16] S . Tripathy, L. B. Omprakash, S. K. Mandal and B. S. Patro, "Low power multiplier architectures using vedic mathematics in 45nm technology for high speed computing," 2015 International Conference on Communication, Information & Computing Technology (ICCICT), Mumbai, India, 2015, pp. 1-6, doi: 10.1109/ICCICT.2015.7045662.

[17] N. S. Kumar, S. S.D., S. N.G., S. V. Hande, P. K. Y.G. and K. B.S., "Design of an Area-Efficient Multiplier," 2017 International Conference on Recent Advances in Electronics and Communication Technology (ICRAECT), Bangalore, India, 2017, pp. 329-332, doi: 10.1109/ICRAECT.2017.75.

[18] Jetsdaporn Satansup, Worapong Tangsrirat, “1.5-V CMOS Current Multiplier/Divider,” International Journal of Electrical and Computer Engineering (IJECE), vol. 8, no. 3, pp. 1478-1487, June 2018, doi: 10.11591/ijece.v8i3.pp1478-1487

[19] D. Govekar and A. Amonkar, "Design and implementation of high speed modified booth multiplier using hybrid adder," 2017 International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2017, pp. 138-143, doi: 10.1109/ICCMC.2017.8282661.

[20] G. G. Kumar and S. K. Sahoo, "Implementation of a high speed multiplier for high-performance and low power applications," *2015 19th International Symposium on VLSI Design and Test*, Ahmedabad, India, 2015, pp. 1-4, doi: 10.1109/ISVDAT.2015.7208143.