

Energy-Efficient Multiplier Design Using Urdhva Tiryagbhyam Algorithm and Approximate Compressors

¹ SK MUZAMMIL, ² Dr.G.N.KODANDA RAMAIAH, ³ Mr.T SIVA KUMAR

¹ M.Tech Student, Department of Electronics and Communication Engineering, Kuppam Engineering College, KES Nagar, Kuppam, Andhra Pradesh 517425 India.

² Professor, Department of Electronics and Communication Engineering, Kuppam Engineering College, KES Nagar, Kuppam, Andhra Pradesh 517425 India. Email ID : gnk.ramaiah@gmail.com

³ Assistant Professor, Department of Electronics and Communication Engineering, Kuppam Engineering College, KES Nagar, Kuppam, Andhra Pradesh 517425 India.

Abstract— Efficient multiplier design is crucial for enhancing the performance of digital systems, particularly in applications requiring high-speed computations. This study focuses on implementing multipliers using the Vedic Urdhva Tiryagbhyam algorithm, incorporating approximate compressors for partial product addition in 5:3, 6:3, and 7:3 designs. The proposed method is compared with the existing method that employs stacker-based binary compressors. Both designs are synthesized and analyzed using the Xilinx Vivado tool, evaluating their performance based on area utilization (LUTs), static power, and dynamic power consumption. Experimental results reveal significant improvements in the proposed method. The area utilization is reduced from 596 LUTs to just 56 LUTs. Static power consumption drops from 0.823 W to 0.136 W, while dynamic power decreases from 48.008 W to 15.363 W. Overall, the proposed method demonstrates a remarkable reduction in power consumption and area, making it a superior choice for energy-efficient and compact digital multiplier designs. These advancements highlight the potential of integrating approximate compressors with the Vedic Urdhva Tiryagbhyam algorithm for high-performance digital arithmetic systems.

Keywords— Vedic Multiplier, Urdhva Tiryagbhyam Algorithm, Approximate Compressors, 5:3 Compressor, 6:3 Compressor, 7:3 Compressor, Partial Product Addition, Stacker-Based Binary Compressors etc.

I. INTRODUCTION

Multiplication is a fundamental operation in digital systems, forming the backbone of numerous applications ranging from arithmetic computation to signal processing. Over the decades, researchers have focused on developing efficient multiplier architectures to achieve better trade-offs among speed, area, and power consumption. The pioneering works of Wallace [1] and Dadda [2] laid the foundation for high-speed multipliers through the introduction of tree-based reduction techniques. These designs significantly reduced the delay caused by partial product addition, making them suitable for high-performance computing. Subsequent enhancements, such as column compression [3] and the use of multi-input counters and compressor circuits [4], further refined these architectures, addressing specific bottlenecks in multiplier design. However, challenges persist in the domain of multiplier design. Traditional methods, while effective in reducing delay, often result in increased area and power consumption [5]. Recent advances, such as the use of reversible logic [6] and algorithmic optimizations [7], aim to address these issues but still face limitations in scalability and integration with modern low-power systems.

Moreover, the growing demand for energy-efficient computing has highlighted the need for novel approaches that balance performance and power efficiency.

The limitations of existing methods, such as high power consumption and area overhead, motivate the need for alternative designs. Compressors, which play a crucial role in reducing partial products, offer an opportunity to optimize the trade-offs in multiplier design. By exploring novel architectures and algorithms, it is possible to develop multipliers that meet the demands of modern applications in terms of speed, power, and area.

The primary objectives of this work are:

- To design an optimized multiplier architecture using the Urdhva Tiryagbhyam algorithm combined with approximate compressors.
- To evaluate and compare the performance of the proposed design against existing methods, focusing on key metrics such as area, power, and speed.
- To validate the proposed approach using state-of-the-art simulation tools like Xilinx Vivado.

This paper makes the following contributions:

- A novel multiplier architecture leveraging approximate compressors for partial product reduction.
- A comparative analysis of the proposed design with traditional stacker-based binary compressors in terms of area, power, and performance.
- Implementation and validation of the proposed design using Xilinx Vivado, showcasing significant improvements in key metrics.

The organization of the paper divides different sections. The Literature review is presented in section 2. Further in section 3 & 4 shown the Existing Method and Proposed System, Experimental Results is shown in 5. Conclusion and future work are presented by last sections 6.

II. RELATED WORKS

[1] C. S. Wallace, This foundational work by Wallace introduces the Wallace tree, a method to optimize partial product reduction in multipliers. The Wallace tree uses a hierarchical structure of carry-save adders, reducing delay and enhancing speed in multiplication operations. This technique laid the groundwork for future high-speed multiplier designs by focusing on parallel processing and efficient partial product accumulation.

[2] L. Dadda, Dadda's work extends the concept of the Wallace tree by optimizing the number of reduction stages, leading to reduced hardware complexity. By

minimizing the number of adders in each stage, Dadda's algorithm achieves a more area-efficient design compared to Wallace's approach, making it suitable for resource-constrained applications.

[3] Z. Wang, G. A. Jullien, and W. C. Miller, This paper presents a novel column compression technique for multipliers, which improves upon existing reduction schemes by using a more compact representation of partial products. The proposed design reduces delay and power consumption, making it a critical contribution to high-performance computing.

[4] M. Mehta, V. Parmar, and E. Swartzlander, The authors introduce the use of multi-input counters and compressor circuits for high-speed multiplication. By aggregating multiple inputs simultaneously, these techniques enhance speed and reduce the critical path delay in digital multipliers, particularly in applications requiring high throughput.

[5] Krishnaveni, Dhulipala et al., This study focuses on reversible logic for designing an 8x8 Wallace tree multiplier. The use of reversible gates ensures energy efficiency and reduces power dissipation, aligning with modern demands for low-power computing. The design demonstrates improvements in both speed and resource utilization.

[6] S. Asif and Y. Kong, Asif and Kong propose an algorithmic approach to Wallace multiplier design using high-speed counters. Their method emphasizes reducing propagation delay and enhancing overall performance, particularly for large bit-width multipliers.

[7] S. Veeramachaneni and M. B. Srinivas, This work focuses on optimizing arithmetic circuits for multiplier implementation. By leveraging novel architectures, the authors achieve a balance between speed and power consumption, making their designs well-suited for embedded systems and portable devices.

[8] S. Veeramachaneni et al., This paper introduces innovative architectures for compressors used in multiplier designs. These compressors significantly reduce power and area, ensuring high-speed operation and making them integral components of modern multiplier circuits.

[9] C. Fritz and A. T. Fam, The authors propose a symmetric stacking method for designing fast binary counters, which are critical for multiplier operations. Their approach enhances the speed and efficiency of partial product reduction, aligning with the needs of high-performance VLSI systems.

[10] S. R. Huddar et al., This study integrates Vedic mathematics with compressor circuits to develop a high-speed multiplier. The proposed design highlights the synergy between ancient computation techniques and modern digital circuit design, achieving superior performance metrics.

[11] N. Pokhriyal et al., The authors present an area-efficient and low-power 8x8 Vedic multiplier using compressors. Their design minimizes resource usage while maintaining high computational accuracy, making it ideal for low-power applications.

[12] H. Kaur and N. R. Prakash, This work emphasizes the use of compressors for area-efficient and low-power-delay-product (PDP) Vedic multipliers. The proposed approach demonstrates significant improvements

in power and area metrics, addressing critical challenges in VLSI design.

[13] S. R. Huddar et al., By combining Vedic mathematics with modern compressor techniques, this study proposes a novel multiplier design that achieves high-speed operation and reduced complexity, targeting real-time applications.

[14] G. V. Nikhil et al., This paper introduces a low-power Vedic multiplier integrated with a Kogge-Stone adder and reversible logic gates. The proposed design achieves enhanced speed and reduced power consumption, catering to energy-efficient applications.

[15] DVS Chandrababu et al., This study implements a 16-bit multiplier using Vedic mathematics. The design focuses on achieving scalability and efficiency, demonstrating its applicability in large-scale arithmetic operations.

[16] Sunesh N. V and Sathishkumar P, This paper proposes a fast floating-point multiplier unit optimized for speed and accuracy. The design targets high-performance computing applications, emphasizing efficient handling of floating-point operations.

[17] S. Ramachandran and K. S. Pande, The authors present an integrated Vedic multiplier architecture that combines traditional techniques with modern optimizations. Their design achieves significant improvements in performance and efficiency.

[18] A. Raju and S. K. Sa, This work integrates Kogge Stone adders into multiplier designs, focusing on achieving high-speed operation and reduced power consumption. The proposed design is particularly effective in real-time processing environments.

[19] A. Jain and A. Jain, This comparative study evaluates the performance of Vedic multipliers against conventional designs. The results demonstrate the advantages of Vedic approaches in terms of speed and resource efficiency, highlighting their potential for diverse applications

III. EXISTING METHOD

A. 6:3 Compressor Design

The inputs of 6:3 compressor are given as $x_5, x_4, x_3, x_2, x_1, x_0$. These are divided into two groups of 3-bits. Set of 3-bits is given as the input of three-bit stacker and the "1s" are grouped together. h_0, h_1, h_2 and i_0, i_1, i_2 are the outputs of the three-bit stacker circuits used in the compressor. There are 64 input combinations for 6:3 compressor, out of which only 16 combinations are there after the stacking process. Hence, the equations for the compressor output are written in terms of the stacker outputs. The truth table of the output, where the inputs are stacker outputs, is shown in Table II. The schematic diagram of the proposed 6:3 compressor is shown in Fig. 1. The equations as per the truth table are given by.

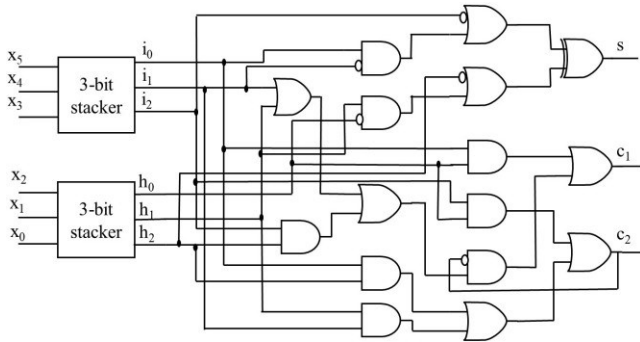


Fig. 1. 6:3 Compressor Design

B. 7:3 Compressor Design

The 7:3 compressor behaves as the 6:3 compressor when the Most Significant Bit (MSB) is 0. Hence, the equations remain the same as the 6:3 compressor when the MSB is 0. The equations for the output when MSB 1 are found out in terms of stacker output as done in the case of 6:3 compressor. The output is selected using a multiplexer keeping MSB as the select line. The schematic diagram of the proposed 7:3 compressor is shown in Fig. 4. The output equations when MSB equal to "1" are given by in tele monitoring of PD, offering a non-invasive method for continuous patient evaluation based on voice analysis.

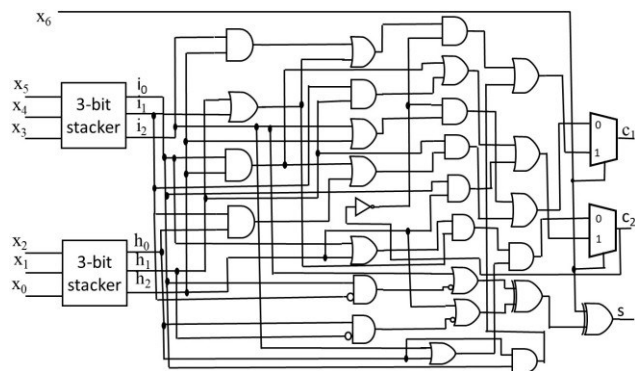


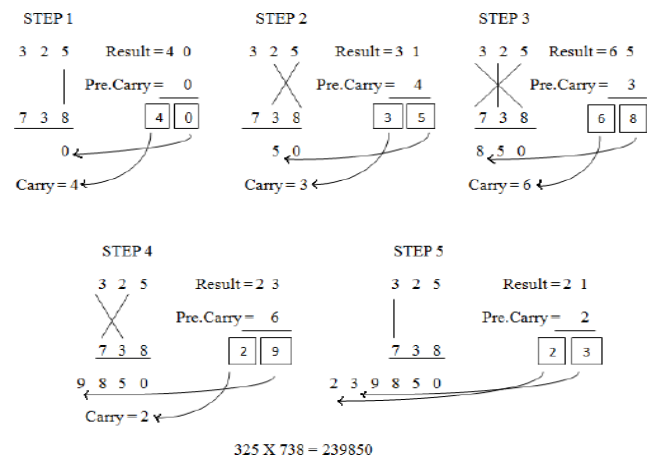
Fig. 2. 7:3 Compressor Design

IV. PROPOSED METHOD

The proposed method employs the Urdhva Tiryagbhyam algorithm, a Vedic multiplication technique, integrated with approximate compressors to achieve efficient partial product reduction. Unlike traditional stacker-based binary compressors, the proposed architecture uses 5:3, 6:3, and 7:3 compressor designs to optimize the addition of partial products. This approach significantly reduces the area and power consumption while maintaining competitive speed. The use of approximate compressors introduces minor computational inaccuracies, which are tolerable in many practical applications, particularly those involving multimedia and signal processing. The design is implemented and validated using the Xilinx Vivado tool, demonstrating substantial improvements over existing methods in terms of area utilization and power efficiency.

A. Urdhva Triyakbhyam Algorithm

The Urdhva Triyakbhyam algorithm represents a pivotal component of Vedic Mathematics, drawing upon the principles of vertical and crosswise multiplication. This ancient mathematical technique offers a general multiplication formula applicable to various scenarios in mathematics. Its literal translation, "Vertically and crosswise," encapsulates its fundamental approach. At its core, the Urdhva Triyakbhyam algorithm introduces a novel concept enabling the simultaneous generation and addition of partial products. Through this approach, all partial products necessary for multiplication are generated concurrently, streamlining the multiplication process. The algorithm facilitates parallelism in both the generation of partial products and their subsequent summation, optimizing computational efficiency. Moreover, the versatility of the Urdhva Triyakbhyam algorithm allows for its generalization to accommodate $n \times n$ bit numbers. This scalability underscores its applicability across a wide range of multiplication scenarios, further enhancing its utility in digital circuit design.



325 X 738 = 239850

Fig. 3. Example of Urdhva Triyakbhyam

B. Vedic Multiplier

Figure 4 illustrates the architecture of an 8-bit Vedic multiplier designed using the Urdhva Tiryagbhyam (vertical and crosswise) algorithm. The structure efficiently computes the product of two 8-bit operands by breaking the inputs into smaller chunks, leveraging parallelism inherent in the Vedic multiplication approach. The architecture involves generating partial products based on vertical and crosswise multiplication patterns and then summing them systematically to obtain the final result. This design's modular nature simplifies implementation and ensures scalability for higher-bit multiplications while maintaining computational efficiency.

improvements in area utilization and power savings compared to conventional designs.

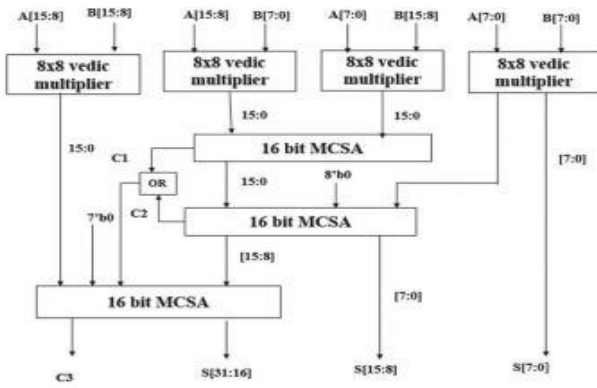


Fig. 4. 8-Bit Vedic Multiplier

C. Approximate Compressor

Figure 5 presents the schematic of a 3-1-1-2 compressor used in the partial product reduction stage. This compressor takes three primary inputs along with one carry-in bit and produces a single sum and two carry outputs. The design is optimized to approximate certain logic operations, thereby reducing the number of transistors and logic gates required. This approximation leads to lower area utilization and power consumption, albeit with a minimal trade-off in precision. The compressor's efficiency plays a critical role in the overall multiplier design, as it directly affects the hardware complexity and power profile.

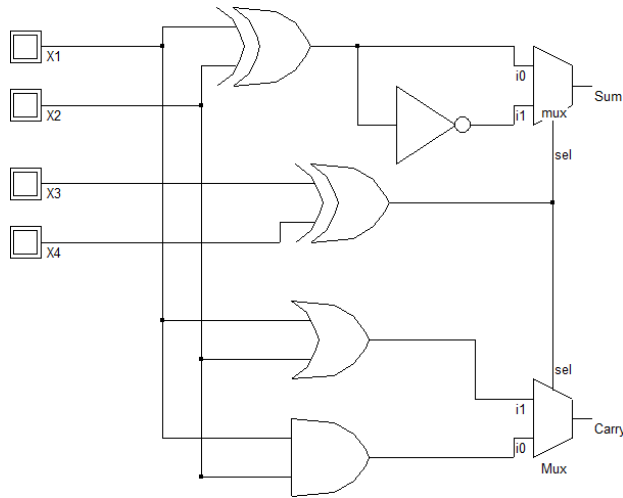


Fig. 5. 3-1-1-2 Compressor

D. Vedic Multiplier with Approximate Compressor

Figure 6 depicts the complete architecture of the proposed Vedic multiplier integrating approximate compressors. This design builds upon the traditional Vedic multiplier by incorporating 5:3, 6:3, and 7:3 approximate compressors in the partial product reduction phase. The approximate compressors replace exact compressors, simplifying the addition process and reducing the overall hardware resources needed. The figure outlines the flow of data from the generation of partial products to their reduction and final summation. This hybrid approach effectively balances performance and resource efficiency, resulting in significant

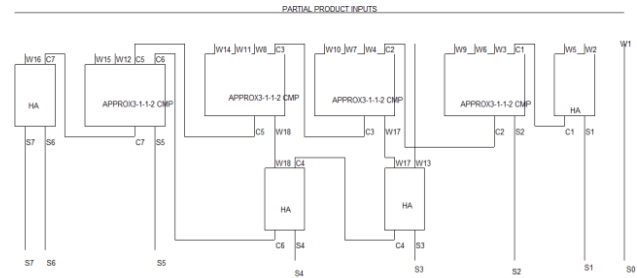


Fig. 6. Proposed Vedic Multiplier with Approximate Compressor

E. Components of Proposed method

The proposed method involves Vedic Urdhva Tiryagbhyam algorithm-based multipliers using approximate compressors for partial product addition. Below are equations that can represent the different components and performance metrics described:

1) Partial Product Generation (Vedic Urdhva Tiryagbhyam Algorithm)

The Vedic Urdhva Tiryagbhyam algorithm generates partial products PPP based on crosswise and vertical multiplications of the binary digits of two numbers A and B:

$$P_{ij} = A_i \cdot B_j \tag{1}$$

Where A_i and B_j are the i -th and j -th binary digits of the input operands A and B, respectively.

2) Compressor Logic for Partial Product Reduction

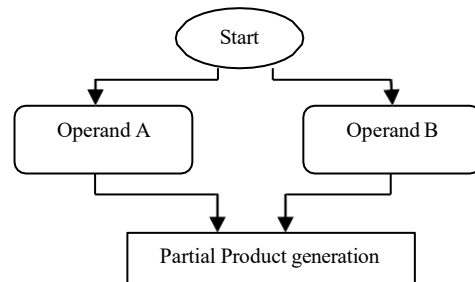
For approximate compressors (5:3, 6:3, and 7:3), the output sum and carry can be defined as follows:

$$\text{Sum} = \sum_{i=1}^n X_i + \text{Approximation Term} \tag{2}$$

$$\text{Carry} = \lfloor \frac{\sum_{i=1}^n X_i}{2} \rfloor \tag{3}$$

Where X_i are the input bits, n represents the number of inputs, and the approximation term accounts for simplifications in logic to reduce hardware complexity.

F. Implementation Flow chart



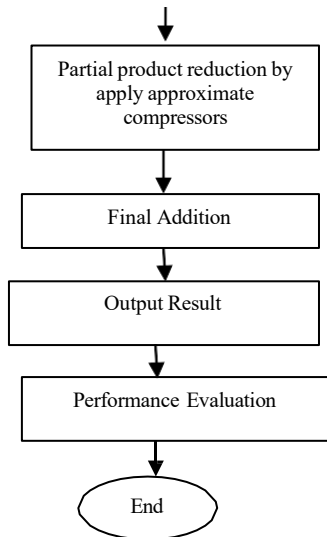


Fig. 7. Implimentation of the flow chart

Start

Begin the process of multiplier design.

-Input Operand A and Operand B

- Provide the binary inputs for multiplication.

-Partial Product Generation

- Use the Vedic Urdhva Tiryagbhyam algorithm to generate partial products.

-Partial Product Reduction

- Apply approximate compressors:
 - 5:3 Approximate Compressor
 - 6:3 Approximate Compressor
 - 7:3 Approximate Compressor

-Final Addition

- Sum the reduced partial products to generate the final result.

-Output Result

- Display the final multiplied value.

-Performance Evaluation

- Analyze key metrics:
 - Area Utilization (LUTs)
 - Static Power Consumption
 - Dynamic Power Consumption

-End

- Complete the process.

G. Performance Metrics

1) Performance Area Utilization (LUTs)

The number of Look-Up Tables (LUTs) used in the FPGA implementation of the design. Determines the compactness of the design. Lower LUT utilization reflects better optimization, making it suitable for resource-constrained applications.

Measured directly through synthesis tools like Xilinx Vivado.

$$\text{Reduction in LUTs} = \frac{\text{LUTs}_{\text{existing}} - \text{LUTs}_{\text{proposed}}}{\text{LUTs}_{\text{existing}}} \times 100\% \quad (4)$$

2) Static Power Consumption

The power consumed when the circuit is not actively switching. Leakage current due to transistor imperfections. Sub threshold leakage and gate-oxide tunneling contribute to static power. Important for applications requiring low standby power. Reflects the energy efficiency of the design.

$$P_{\text{static}} = V_{\text{dd}} \cdot I_{\text{leakage}} \quad (5)$$

Where

V_{dd} : Supply voltage.

I_{leakage} : Leakage current.

3) Dynamic Power Consumption

Power consumed during the active switching of logic gates in the circuit. Charging and discharging of capacitive loads. Short-circuit current during transitions. Dominates power consumption in high-speed applications. Directly related to the switching activity and operating frequency.

$$P_{\text{dynamic}} = C_{\text{load}} \cdot V_{\text{dd}}^2 \cdot f \quad (6)$$

Where

C_{load} : Load capacitance.

V_{dd} : Supply voltage.

f : Operating frequency.

4) Total Power

Consumption Sum of static and dynamic power.

$$P_{\text{total}} = P_{\text{static}} + P_{\text{dynamic}} \quad (7)$$

5) Energy Efficiency

Amount of energy consumed per operation.

$$\text{Energy Efficiency} = \frac{\text{Total Power}}{\text{Number of Operations per Second}} \quad (8)$$

6) Improvement in the proposed method

To emphasize the improvements in the proposed method:

$$\text{Improvement}_{\text{area}} = \frac{\text{LUTs}_{\text{existing}} - \text{LUTs}_{\text{proposed}}}{\text{LUTs}_{\text{existing}}} \times 100\% \quad (9)$$

$$\text{Improvement}_{\text{static power}} = \frac{P_{\text{static, existing}} - P_{\text{static, proposed}}}{P_{\text{static, existing}}} \times 100\% \quad (10)$$

$$\text{Improvement}_{\text{dynamic power}} = \frac{P_{\text{dynamic, existing}} - P_{\text{dynamic, proposed}}}{P_{\text{dynamic, existing}}} \times 100\% \quad (11)$$

V. RESULTS AND DISCUSSIONS

The simulation results of the proposed multiplier design highlight significant improvements over the existing methods, demonstrating its efficiency in terms of area

utilization and power consumption. The simulation, performed using Xilinx Vivado, validates these findings and confirms that the proposed design not only meets but surpasses the performance benchmarks of the existing method.

A. Existing Method

This figure 8 shows the Schematic architecture of the existing multiplier design. It uses stacker-based binary compressors for partial product addition. The design prioritizes accuracy but results in higher resource utilization and power consumption due to the complexity of the exact compression logic.

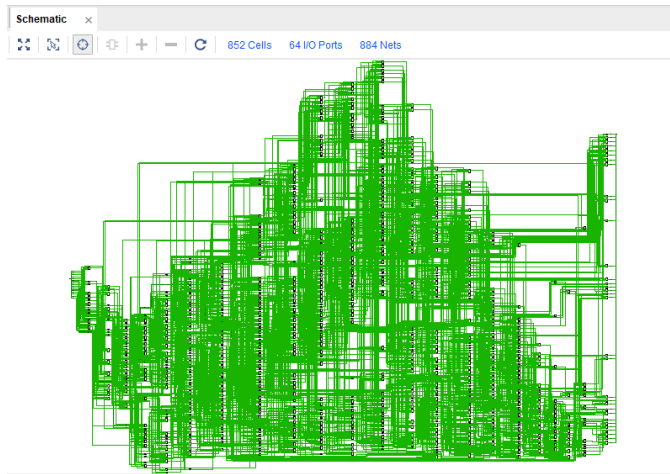


Fig. 8. Schematic Circuit for Existing Method

The area utilization analysis for the existing method demonstrates in the fig 9, the consumption of 596 LUTs in FPGA implementation. This indicates a significant hardware footprint, making it less suitable for resource-constrained applications.

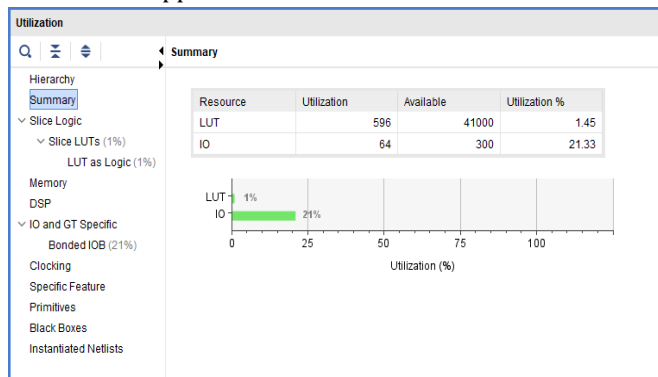


Fig. 9. Area Utilization of Existing method

The power analysis reveals high static and dynamic power consumption shown in fig.10. Static power is measured at 0.823 W, while dynamic power reaches 48.008 W, resulting in a total power consumption of 48.831 W. These values highlight the design's inefficiency in terms of energy consumption.

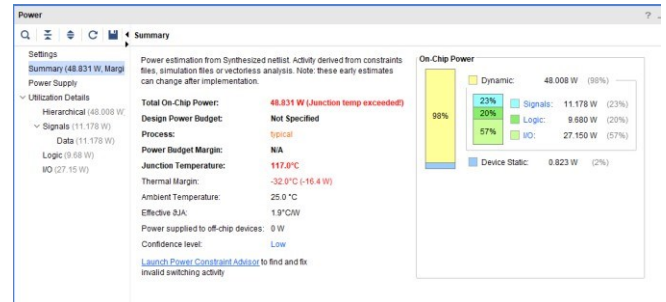


Fig. 10. Power Utilization of the proposed method

The functional simulation of the existing method confirms the correctness of its operation. The simulation also reflects the higher resource usage and power demands, reinforcing the need for optimization shown in fig.11.

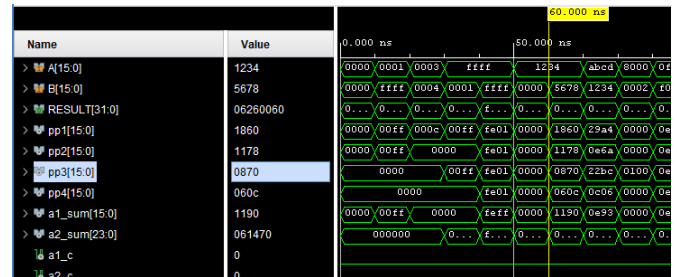


Fig. 11. Simulation of Existing method

B. Proposed Method

This schematic illustrates the architecture of the proposed multiplier, incorporating the Vedic Urdhva Tiryagbhyam algorithm with approximate compressors for partial product addition. The design simplifies the compression logic, reducing hardware complexity and enhancing efficiency.

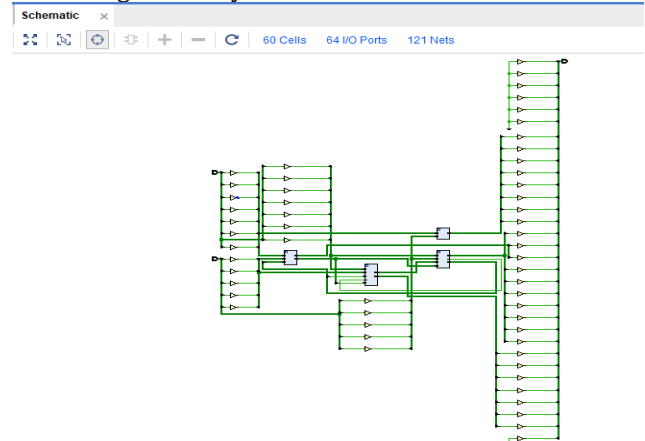


Fig. 12. Schematic Circuit for Proposed Method

The area utilization analysis shows in fig.13, a dramatic reduction in LUT usage, dropping from 596 to just 56 LUTs. This 90.6% improvement demonstrates the proposed

method's ability to minimize utilization resource consumption while maintaining functionality.

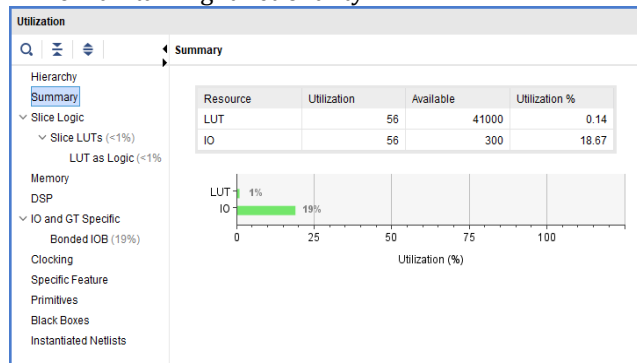


Fig. 13. Area Utilization of Proposed method

The power analysis of the proposed method shown in fig.14, highlights a substantial reduction in both static and dynamic power. Static power drops to 0.136 W, and dynamic power decreases to 15.363 W, leading to a total power consumption of 15.5 W. These values underscore the design's energy efficiency, making it ideal for low-power applications

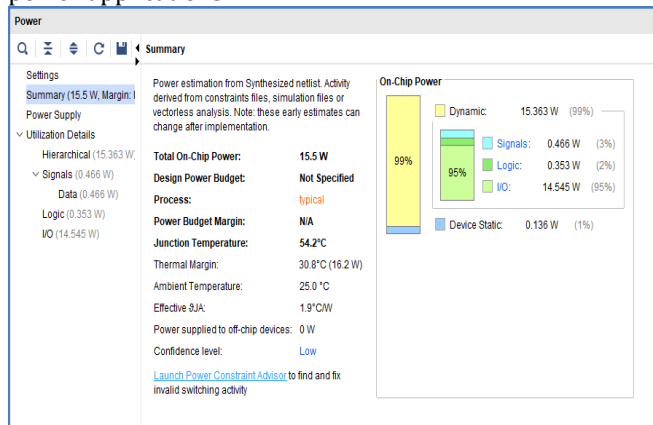


Fig. 14. Power Utilization of Proposed method

The simulation validates the proposed method's functionality and confirms its efficiency in reducing area and power consumption while maintaining accurate results shown in fig.15.

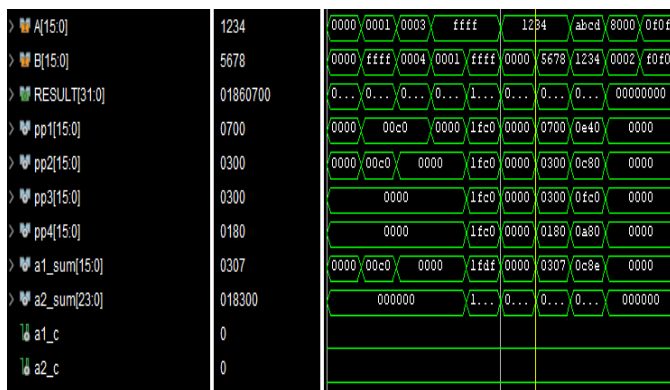


Fig. 15. Simulation of proposed method

C. Performance Comparison Table

TABLE I. PERFORMANCE COMPARISON OF EXISTING AND PROPOSED METHOD

S.N	Parameter	Existing method	Proposed method	Improvement (%)
1	Area (LUT)	596	56	90.6
2	Static Power in Watts	0.823	0.136	83.5
3	Dynamic Power in Watts	48.008	15.363	68.0
4	Total Power in Watts	48.831	15.5	68.3

Table I. Performance Comparison of Existing and Proposed Method. The performance comparison table consolidates the key metrics of both methods, demonstrating the proposed design's superiority: Area Utilization Reduced by 90.6%, from 596 LUTs to 56 LUTs. Static Power: Decreased by 83.5%, from 0.823 W to 0.136 W. Dynamic Power Lowered by 68.0%, from 48.008 W to 15.363 W. Total Power: Reduced by 68.3%, from 48.831 W to 15.5 W.

D. Performance Analysis

This graph visually compares the performance metrics of the existing and proposed methods, highlighting the significant improvements achieved by the proposed design. The graph emphasizes the drastic reductions in area utilization and power consumption, underscoring the efficiency of incorporating approximate compressors into the multiplier design

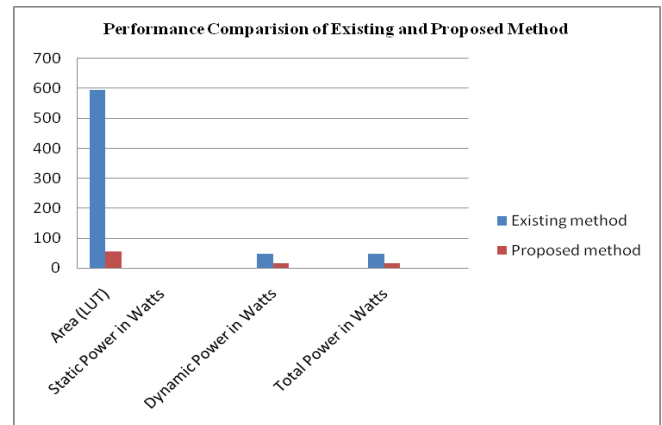


Fig. 16. Comparison Performance Graph

VI. CONCLUSION AND FUTURE SCOPE

The proposed design of a digital multiplier using the Vedic Urdhva Tiryagbhyam algorithm combined with approximate compressors (5:3, 6:3, and 7:3 designs) demonstrates significant advancements in resource efficiency and power consumption. Compared to the existing method, the proposed approach achieves a 90.6% reduction in area utilization (from 596 LUTs to 56 LUTs) and an 83.5% and 68.0% reduction in static and dynamic power, respectively.

These improvements validate the effectiveness of the proposed design for energy-efficient and compact digital arithmetic systems. By leveraging the simplicity of approximate compressors, the multiplier achieves an optimal balance between performance, accuracy, and resource utilization, making it highly suitable for applications in portable, low-power, and high-speed systems.

Future scope

In future the Approximate multipliers are ideal for error-tolerant applications like machine learning accelerators, image processing, and digital signal processing. Exploring these domains could unlock additional use cases.

REFERENCES

- [1] J C. S. Wallace, A suggestion for a fast multiplier, IEEE Trans. Electron.Comput., vol. EC-13, no. 1, pp. 14-17, Feb. 1964.
- [2] L. Dadda, Some schemes for parallel multipliers, Alta Freq., vol. 34, pp.349-356, May 1965.
- [3] Z. Wang, G. A. Jullien, and W. C. Miller, A new design technique for column compression multipliers, IEEE Trans. Comput., vol. 44, no. 8, pp. 962-970, Aug. 1995.
- [4] M. Mehta, V. Parmar, and E. Swartzlander, High-speed multiplier design using multi-input counter and compressor circuits, in Proc. 10th IEEE Symp. Comput. Arithmetic, Jun. 1991, pp. 43-50.
- [5] Krishnaveni, Dhulipala and Priya, M and Kaliaperumal, Baskaran. (2012). Design of an Efficient Reversible 8x8 Wallace Tree Multiplier. World Applied Sciences Journal. 20. 10.5829/idosi.wasj.2012.20.08.711.
- [6] S. Asif and Y. Kong, Design of an algorithmic Wallace multiplier using high speed counters, in Proc. IEEE Comput. Eng. Syst. (ICCES), Dec. 2015, pp. 133-138.
- [7] S. Veeramachaneni and M. B. Srinivas, "Design of optimized arithmetic circuits for multiplier realization," 2013 IEEE Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics (PrimeAsia), Visakhapatnam, 2013, pp. 219-224.
- [8] S. Veeramachaneni, K. M. Krishna, L. Avinash, S. R. Puppala, and M. B. Srinivas, Novel architectures for high-speed and low-power 3-2, 4-2 and 5-2 compressors, in Proc. 20th Int. Conf. VLSI Design Held Jointly 6th Int. Conf. Embedded Syst. (VLSID), Jan. 2007, pp. 324-329.
- [9] C. Fritz and A. T. Fam, "Fast Binary Counters Based on Symmetric Stacking," in IEEE Transactions on Very Large-Scale Integration (VLSI) Systems, vol. 25, no. 10, pp. 2971-2975, Oct. 2017.
- [10] S. R. Huddar, Kalpana M, Surbhi Mohan, Novel High-Speed Vedic Mathematics Multiplier using Compressors, Conference proceedings of IEEE, 2013.
- [11] N. Pokhriyal, H. Kaur, H. and D. N. R Prakash, 2013. Compressor Based Area-Efficient Low-Power 8x8 Vedic Multiplier. Int. Journal of Engineering research and Applications, 3(6), pp. 1469-1472.
- [12] H. Kaur and N. R. Prakash, "Area-efficient low PDP 8-bit vedic multiplier design using compressors," 2015 2nd International Conference on Recent Advances in Engineering and Computational Sciences (RAECS), Chandigarh, 2015, pp. 1-4.
- [13] S.R.Huddar, S.R. Rupanagudi, M.Kalpana and S.Mohan, "Novel high speed vedic mathematics multiplier using compressors," 2013 International Mutli-Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s), Kottayam, 2013, pp. 465-469.
- [14] G. V. Nikhil, B. P. Vaibhav, V. G. Naik and B. S. Premananda, "Design of low power barrel shifter and vedic multiplier with kogge-stone adder using reversible logic gates," 2017 International Conference on Communication and Signal Processing (ICCSP), Chennai, 2017, pp. 1690-1694.
- [15] DVS Chandrababu, T Ravindra, P Nagaraju, P Narsimha and Ch Neelima, 16 Bit Multiplier Implementation Using Vedic Mathematics. International Journal of Mechanical Engineering and Technology, 8(7), 2017, pp. 703-716.
- [16] Sunesh N. V and Sathishkumar P, "Design and implementation of fast floating-point multiplier unit," 2015 International Conference on VLSI Systems, Architecture, Technology and Applications (VLSI- SATA), Bangalore, 2015, pp. 1-5.
- [17] S. Ramachandran and Pande, K. S., Design, Implementation and Performance Analysis of an Integrated Vedic Multiplier Architecture, International Journal of Computational Engineering Research, vol. 2, pp. 697-703, 2012.
- [18] A. Raju and S. K. Sa, "Design and performance analysis of multipliers using Kogge Stone Adder," 2017 3rd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), Tumkur, Karnataka, India, 2017, pp. 94-99.
- [19] A. Jain and A. Jain, "Design, implementation and comparison of vedic multipliers with conventional multiplier," 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), Chennai, India, 2017, pp. 1039-1045..