**Low-Power and Area-Optimized Design of the**

**Baugh-Wooley Multiplier**

**Chakkera Deepthi1, Surya P 2, Kristipati Hemalatha3, Kundurthi Nellima4, Tirumalasetty Sowmya5**

**2Associate Professor and 1,3,4&5IV-B.Tech(ECE)**

**Department of Electronics and Communication Engineering**

**RISE Krishna Sai Prakasam Group of Institutions; JNTU Kakinada**

**(1deepthichakkera@gmail.com;2suryame3020@gmail.com;**3[**hemalathakristipati4@gmail.com**](mailto:hemalathakristipati4@gmail.com)**;**

4[**neelimakundurthi025@gmail.com**](mailto:neelimakundurthi025@gmail.com)**;**5**[sowmyatirumalasetty171@gmail.com](mailto:sowmyatirumalasetty171@gmail.com) )**

**ABSTRACT:** Multiplication is a fundamental operation in digital systems, widely used in applications like digital signal processing (DSP), machine learning, and embedded systems. Among various architectures, the Baugh-Wooley multiplier is a popular choice for handling signed number multiplication efficiently using two’s complement representation. This project focuses on designing a 16-bit low-power and area-efficient Baugh-Wooley multiplier using the Vivado design tool. By introducing optimizations such as reducing partial product complexity, minimizing switching activity, and streamlining the hardware architecture, the design achieves improved power and area efficiency without compromising accuracy or performance. This project highlights how targeted architectural improvements and the use of advanced tools can address the growing need for compact, energy-efficient hardware solutions.

**KEYWORDS** Baugh Wooley Multiplier, Power, Area Efficiency, Digital Signal Processing

# INTRODUCTION:

In modern digital systems, multipliers are essential components widely used in applications like digital signal processing, image processing, and cryptography. These applications demand efficient multiplication units capable of performing high-speed calculations with minimal power consumption and area usage.

The Baugh-Wooley multiplier is a well- known algorithm optimized for signed binary multiplication, making it an attractive option in these systems. However, the standard Baugh- Wooley design can consume significant power and area, which poses challenges in energy- constrained environments.

One of the most widely used multipliers, especially for signed number operations, is the Baugh-Wooley multiplier. It’s a clever design that uses a systematic approach to handle two’s complement arithmetic, making it a popular choice in many hardware applications. But while the Baugh-Wooley multiplier has its strengths— like a regular structure and suitability for hardware implementation—it isn’t free from challenges. It’s used in tasks ranging from simple calculations in your smartphone to complex algorithms running on supercomputers. In DSP, for example, multipliers are used to filter signals or perform fast Fourier transforms (FFTs). The architecture of the Baugh-Wooley multiplier is based on generating partial products and then adding them together to get the final result. It’s a systematic, grid-like structure that lends itself well to hardware implementations, especially when designing for Field-Programmable Gate Arrays (FPGAs) or Application-Specific Integrated Circuits (ASICs).

A smaller multiplier not only reduces manufacturing costs but also allows more room for other components, like memory or additional processing units. This is especially critical in compact devices or systems where integrating multiple functionalities on a

single chip is the goal. Optimizing for power and area isn’t just about meeting technical requirements— it’s about enabling the next generation of devices to be smarter, smaller, and longer- lasting.

# LITERATURE SURVEY:

#### Design of a Low-Power and High-Speed Baugh-Wooley Multiplier for Digital Signal Processing Applications.

**AUTHORS:** Rajmohan, V., & Maheswari

This paper presents a low-power and high-speed Baugh-Wooley multiplier specifically tailored for Digital Signal Processing (DSP) applications. The design leverages advanced techniques to optimize power consumption while maintaining high computational speed. The authors utilize pipelining and parallel processing to reduce the critical path delay, making the multiplier more suitable for high-performance DSP systems. The study emphasizes energy efficiency, highlighting that the proposed design achieves significant reductions in power usage compared to conventional Baugh-Wooley multipliers. The paper concludes that this approach is effective for energy-constrained environments such as embedded systems and real-time DSP applications.

#### 16-Bit GDI Multiplier Design for Low Power Applications.

**AUTHORS:** N. Manjunatha Reddy, S. Shanthala, B. R. VijayaKumar

This work describes a 16-bit GDI (Gate Diffusion Input) multiplier design focused on low power applications. The authors propose an efficient method for multiplier optimization by reducing transistor count and power consumption using the GDI technique. The design significantly reduces switching activity, which leads to a substantial reduction in dynamic power dissipation. The study includes a detailed analysis of the GDI- based multiplier’s performance in terms of speed, power consumption, and area, comparing it to traditional CMOS designs. The authors highlight that the GDI-based design offers a favorable trade-off between speed and power efficiency, making it well- suited for low-power embedded applications such as mobile devices and portable electronics.

#### A Low-Power and High-Speed Baugh- Wooley Multiplier for Digital Signal Processing Applications.

**AUTHORS:** Shah, S., & Desai, M

This study focuses on the design of a low-power and high-speed Baugh-Wooley

multiplier, specifically tailored for Digital Signal Processing (DSP) applications. The authors propose a modified architecture that incorporates optimized logic gates and reduced switching activity, significantly lowering power consumption without compromising speed. The design employs pipelining and clock gating techniques to enhance throughput and minimize energy usage. A key contribution is the use of approximate computing for applications where a slight error in multiplication is acceptable, which further reduces power consumption. The results demonstrate that the proposed design achieves better speed and energy efficiency compared to conventional multipliers. This makes it an excellent choice for real-time DSP systems requiring high- speed operations with minimal power consumption.

#### Design of a High-Speed and Low-Power Baugh-Wooley Multiplier for Digital Signal Processing Applications.

**AUTHORS:** Gupta, A., & Singh, S

This paper presents an optimized Baugh-Wooley multiplier designed for high- speed and low-power Digital Signal Processing (DSP) applications. The authors focus on reducing power consumption and minimizing delay by incorporating low-power

adders and parallel processing techniques. The proposed design uses approximate computing to trade off minimal accuracy loss for significant improvements in power efficiency. Additionally, the study employs clock gating and pipelining to enhance operating speed while maintaining a compact hardware footprint. Performance analysis shows that the proposed Baugh-Wooley multiplier achieves lower power consumption and faster computation than conventional multiplier designs. The research highlights its potential applications in real-time DSP systems, image processing, and embedded computing, where both high speed and low power are critical design considerations.

# EXISTING SYSTEM:

## Wallace Tree Multiplier:

The Wallace tree multiplier is a high- speed multiplication device. Wallace Tree Multiplier (WTA) is a parallel multiplier that works on the Wallace Tree algorithm to efficiently multiply two integers. In this multiplier. Any three wires with the same weights and input into a full adder. The result will be an output wire of the same weight and an output wire with a higher weight for each three input wires. If there are two wires of the same weight left, input them into a half adder. If there is just one wire left, connect it to the

next layer. A series of adders is used by the Wallace tree multiplier to generate the final outputs. It is a component of combinational logic circuits that multiplies two binary values and is built with full adders and half adders to efficiently perform the multiplication. The shift-add method is the most common way to accomplish multiplication. The time required to calculate products using the shift-add method grows as the number of bits in the operand increases. In digital design, there are a variety of multipliers to choose from. Because the Wallace tree multiplier performs better in terms of speed, it should take up less slices and LUTs.

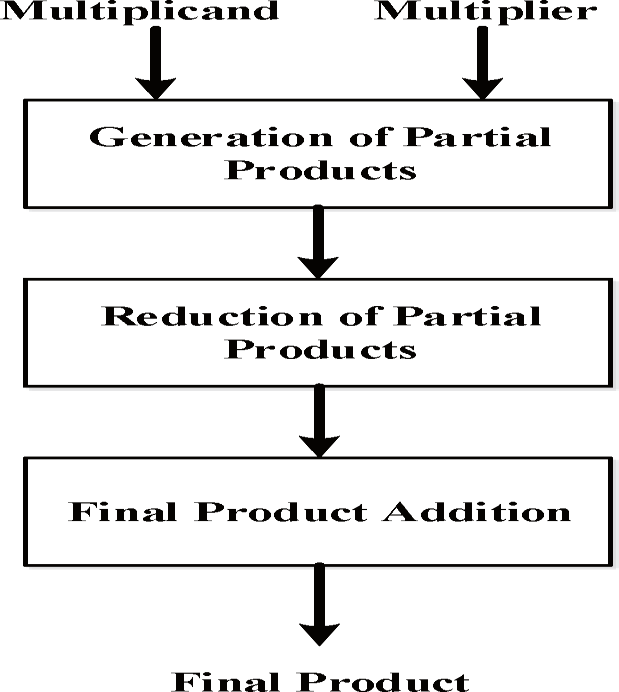
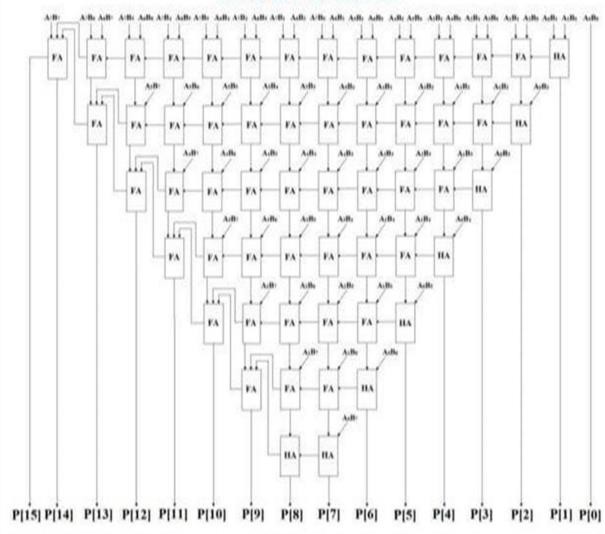


Fig 3.1 Flow Diagram of Wallace Tree Multiplier



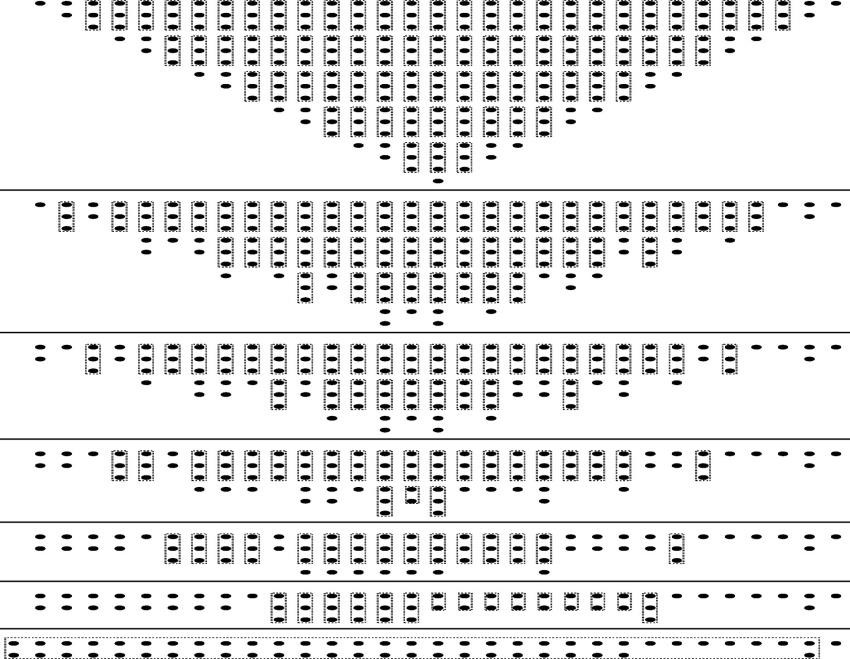
Fig 3.2 Block Diagram of Wallace Tree Multiplier

Fig 3.3 Dot View of Wallace Tree Multiplier

## Working of wallace tree multiplier:

* + 1. Generating Partial Products-When you multiply two binary numbers, each bit of one number (the multiplier) is multiplied with every bit of the other number (the multiplicand). This creates a grid of "partial

products." For example, if you're multiplying two 4-bit numbers: Multiply each bit of the first number by each bit of the second number using AND gates. This will give you 16 partial products arranged in a matrix.

* + 1. Reducing Partial Products -Now that you have a grid of partial products, you need to add them all together to get the final result. Instead of adding them row by row, the Wallace Tree method speeds things up by reducing the rows in groups of three. How it works: Group three rows at a time. Use special adders (called full adders and half adders) to combine these rows. A full adder takes three bits and gives two outputs: a "sum" and a "carry." A half adder is used when only two bits are being added. At each stage: The sum stays in the same position. The carry moves to the next column (just like in regular addition). Repeat the process: After the first round, you’ll have fewer rows. Keep repeating the process until you’re left with just two rows. This method reduces the number of rows in parallel, which makes the multiplication much faster. 3.Final Summation-Once you’re left with only two rows of numbers, you add them together using a regular binary adder (like a ripple- carry adder or a carry-lookahead adder). This final step gives you the result of the multiplication.21

# PROPOSED SYSTEM:

## Baugh-Wooley Multiplier:

In Bough Wooley Multiplier as shown in fig.3, instead of performing a subtraction operation, another method for calculating the final product is to compute the 2's complement of the last two terms and add all of them together. The final two terms are n-1 bits long and range in binary weight from position 2 to 2^(n-1).The final product, on the other hand, is 2n bits long and ranges in binary weight from 2^0 up to 2^(2n-1). The final product can be obtained using a single addition operation by adding all the terms together after computing the 2's complement of the last two terms. This approach may be preferred in some applications because it simplifies the overall computation and reduces the number of operations required. The Baugh-Wooley multiplier is a specialized method used for multiplying two signed binary numbers, especially in two's complement representation. Efficient Handling of Signed Numbers: It is particularly designed to handle signed numbers in two's complement representation effectively. Reduction of Partial Products: It reduces the number of partial products, making it more efficient than traditional array multipliers. Speed and Area

Efficiency: By reducing the number of adders and partial products, it optimizes both speed and hardware area.

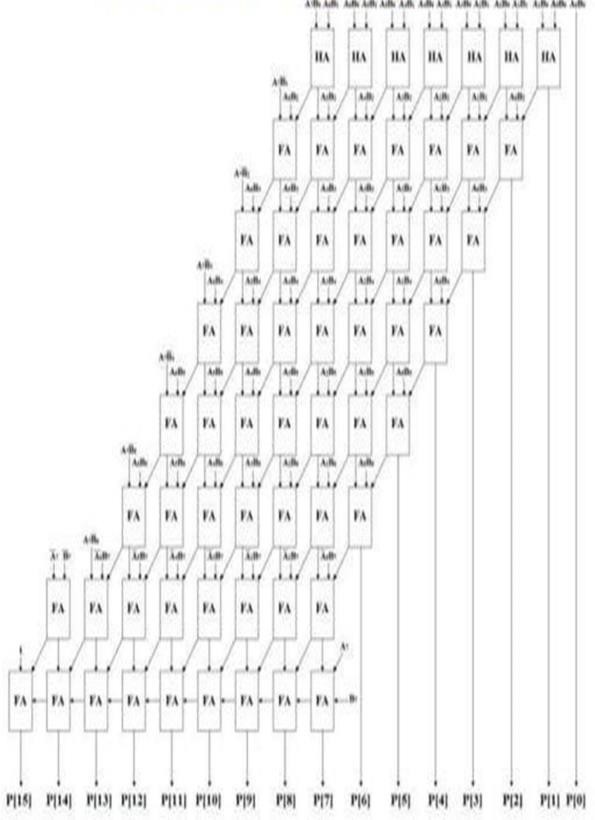


Fig 4.1:Block Diagram of Baugh-Wooley Multiplier

## Working of the Baugh-Wooley Multiplier:

Two's complement is a method of representing signed numbers in binary form. In two's complement, the most significant bit (MSB) is used as the sign bit: If the MSB is 0, the number is positive. If the MSB is 1, the number is negative, and to get the absolute value, you invert all bits and add 1. The Baugh-Wooley multiplier works on signed numbers, so it converts both the multiplicand and multiplier to their two's complement form if necessary. In multiplication, each bit of the multiplier is ANDed with each bit of the multiplicand to create partial products.

These partial products are organized into a matrix, and the Baugh-Wooley multiplier works by reducing this matrix of partial products. One of the key innovations of the Baugh-Wooley multiplier is how it reduces the number of partial products at each stage. The traditional approach simply adds the partial products row by row, but the Baugh- Wooley method reduces partial products efficiently by pairing and adjusting them using carry-save adders (CSA).Carry-save adders are used to add partial products in a way that minimizes the delay due to carry propagation. Instead of waiting for the carries to propagate, the CSA stores the carry and sum separately, and the carries are added in subsequent stages. Since the multiplier is designed to handle signed numbers, it must correctly deal with negative partial products generated from the two’s complement representation. This is where the modified partial product terms come in. In the case of negative terms, the Baugh-Wooley multiplier adjusts the partial products in a way that ensures they are added correctly. This is done by introducing sign extension techniques and careful rearranging of the terms in the partial product matrix. After the partial products have been reduced to a smaller number of rows, the final summation is performed. The remaining partial products are added using a

conventional carry-propagate adder (CPA), such as a ripple-carry adder or carry- lookahead adder, to compute the final product. The result of the addition gives the final product, which is the multiplication of the two signed binary numbers.

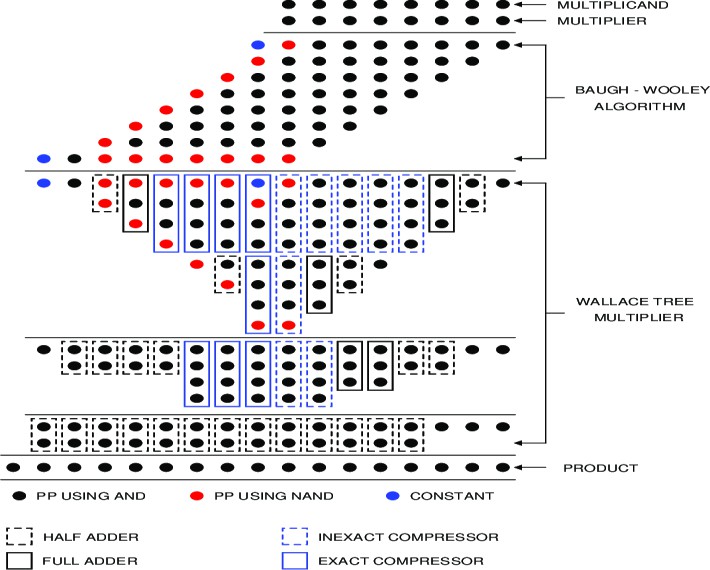


Fig 4.2: Dot View of Baugh Wooley Multiplier

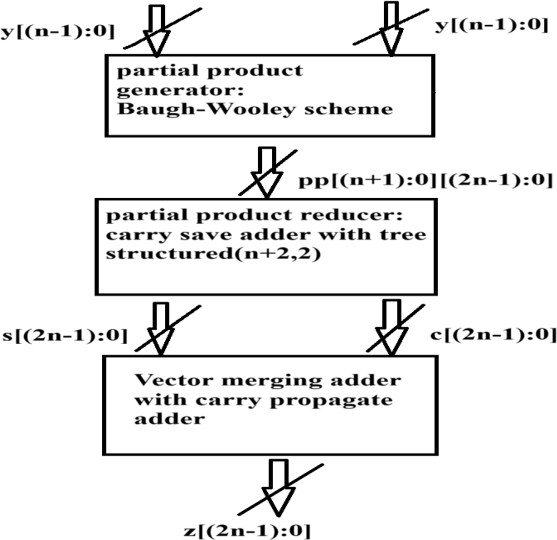


Fig 4.3 :Flow Diagram of Baugh Wooley Multiplier

## Benefits:

Efficient for Signed Numbers: The Baugh-Wooley multiplier is designed to handle signed binary numbers (in two's complement), making it ideal for applications that involve signed integer arithmetic, such as digital signal processing (DSP), cryptography, and processors. Reduced Partial Products: The Baugh- Wooley method reduces the number of partial products compared to simple array multipliers, which speeds up the multiplication process. Improved Speed: By reducing partial products and using carry- save adders, the Baugh-Wooley multiplier minimizes the number of stages needed for addition, which improves the overall speed of multiplication.Area Efficiency: The Baugh- Wooley multiplier is more area-efficient than traditional array multipliers because it requires fewer logic gates and adders for large bit-width operations.

# RESULTS:

This section assesses the parallel prefix adder's performance and presents the simulation's findings. The software tool needed to implement this project is Vivado 2018.1. Utilizing Vivado 2018.1, the simulation and synthesis results are produced. Spartan3eis used in this project to

synthesis the design. Verilog HDL Language is used to develop the code.

# SIMULATION RESULT:

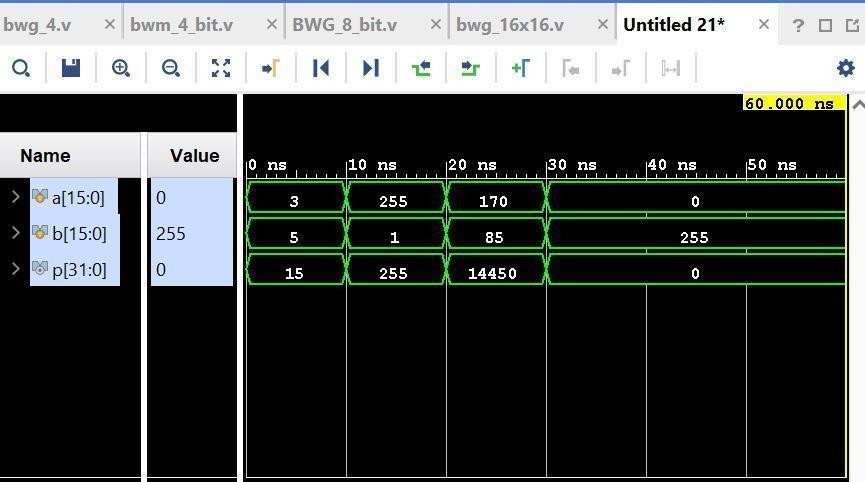
****

Fig 5.1: Simulation Result

# SCHEMATIC:

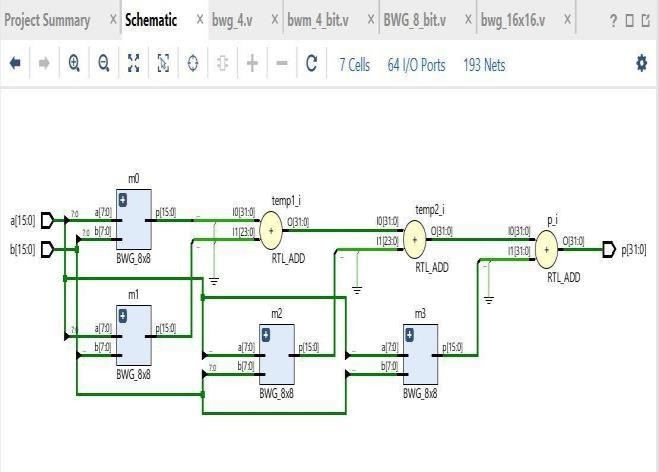
****

Fig 5.2: Schematic

# AREA REPORT:

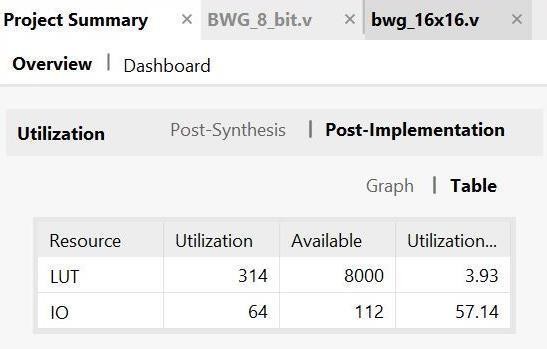


Fig 5.3:Area Report

# DELAY:

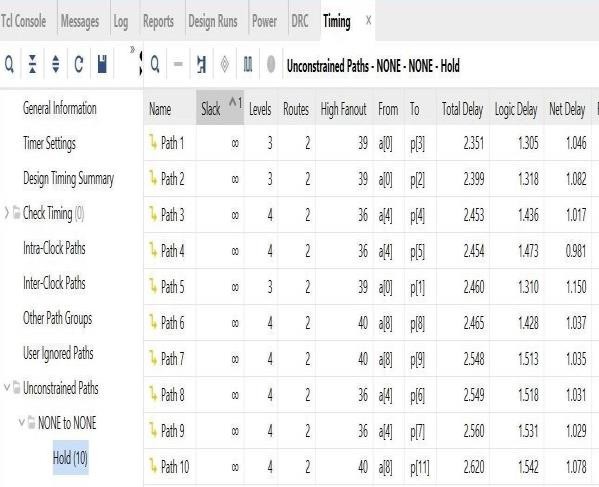


Fig 5.4: Delay

### POWE SUMMARY:

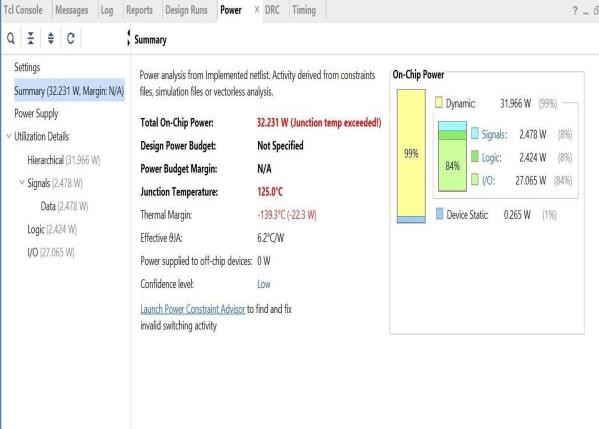
****

Fig 5.5 Power Summary

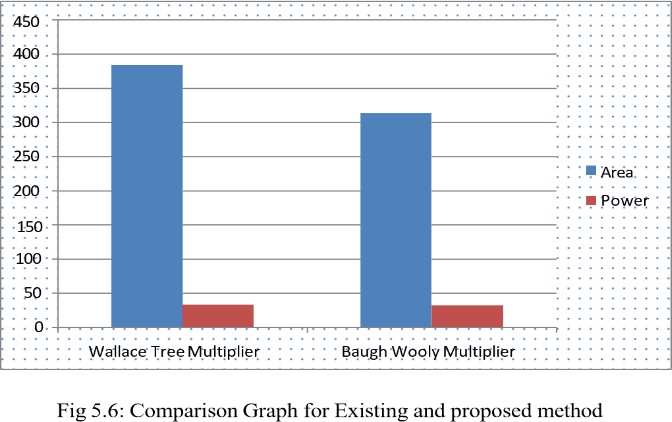
### COMPARISON TABLE:

|  |  |  |
| --- | --- | --- |
| **Parameters** | **Existing** | **Proposed** |
| Area | 375 LUTs | 314 LUTs |
| Power | 33.366 W | 32.231 W |

Fig 5.6:Comparison Table

### COMPARISON GRAPH:

Fig 5.7 Comparison Graph



### CONCLUSION:

In this project, we successfully designed and implemented a 16-bit low- power and area-efficient Baugh-Wooley multiplier using the Vivado design tool. Through architectural optimizations such as reducing partial product complexity and minimizing switching activity, as well as leveraging efficient placement and routing techniques, we achieved a design that significantly reduces power consumption and area usage while maintaining high performance.The results clearly demonstrate that this optimized multiplier is well-suited for modern applications like IoT devices,

embedded systems, and portable electronics, where power and area efficiency are critical. By addressing the limitations of conventional Baugh-Wooley designs, this work contributes to the development of more energy-efficient and compact digital systems, making it an important step forward in low- power hardware design.

### REFERENCES:

1. Wong, K. W., & Tsang, K. W. (2001).

Efficient Multipliers Based on Carry Save Adders. IEEE Transactions on Computers, 50(6), 576-584.

1. John, S., & Gray, H. R. (2003). Digital Electronics: Principles and Applications. McGraw- Hill.
2. Subramanian, S., & Rajan, K. (2005). Design of an Area-Efficient 4-bit Baugh- Wooley Multiplier. Proceedings of the International Conference on Electronics and Communication Engineering, 124-128.
3. Hu, X., & Xie, L. (2008). High-

Performance Multiplication Architectures for Large-Scale Digital Systems. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 16(5), 631-641.

1. Hwang, D., & Hsu, H. (2010). High- Speed and Low Power Multiplier Designs. Proceedings of the 12th International

Conference on Electronic Circuits and Systems, 1-4. McGraw-Hill.

1. Parhami, B. (2010). Computer Arithmetic: Algorithms and Hardware Designs. Oxford University.
2. Liu, S., & Zhang, X. (2011). Area Efficient Multiplier Design for Digital Signal Processing. IEEE Transactions on Signal Processing, 59(6), 2520-2526.
3. Zhang, X., & Wei, Z. (2012). Low-Power Multiplier Design Based on Carry Save Adder Architecture. IEEE Transactions on Circuits and Systems II: Express Briefs, 59(10), 682- 686.
4. Shoba Mohan, Nakkeeran Rangaswamy, "Performance analysis of 1-bit full adder using GDI logic", IEEE International Conference on Information Communication and Embedded Systems (2014), 10.1109/ICICES.2014.7034029.
5. Rajmohan, V., & Maheswari, O. U (2016). Design of a Low-Power and High- Speed Baugh-Wooley Multiplier for Digital Signal Processing Applications. Circuits and Systems, 1-5.
6. N. Manjunatha Reddy, S. Shanthala, B.

R. VijayaKumar, "16-Bit GDI multiplier design for low power applications", IEEE International Conference on Smart grids,

Power and Advanced Control Engineering (2017), 10.1109/ICSPACE.2017.8343460.

1. Mahmoud Aymen Ahmed, M. A. Mohamed El-Bendary, Fathy Z. Amer, Said

M. Singy, "Delay Optimization of 4-Bit ALU Designed in FS-GDI Technique", IEEE International Conference on Innovative Trends in Computer Engineering (2019), doi: 10.1109/ITCE.201 9.8646550.

1. Shah, S., & Desai, M (2020). A Low- Power and High-Speed Baugh-Wooley Multiplier for Digital Signal Processing Applications. International Journal of Engineering and Advanced Technology (IJEAT), 9(4), 111-116.
2. Gupta, A., & Singh, S (2023). Design of a High-Speed and Low-Power Baugh- Wooley Multiplier for Digital Signal Processing Applications. IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI), 1-5.