Efficient Integration of Wallace Tree and Baugh-Wooley Multiplier for High-performance Signed Multiply and Accumulate (MAC) Units

Anuraj V
National Institute of Technology Delhi

Jodu Aravind
Rajiv Gandhi University of Knowledge Technologies

Ankith
Rajiv Gandhi University of Knowledge Technologies

Dhandapani Vaithiyanathan
National Institute of Technology Delhi

他

https://doi.org/10.5109/7342450

出版情報：Evergreen. 12 (1), pp.217-226, 2025-03. Interdisciplinary Graduate School of Engineering Sciences, Kyushu University, Japan
バージョン：

Efficient Integration of Wallace Tree and Baugh-Wooley Multiplier for High-performance Signed Multiply and Accumulate (MAC) Units

Anuraj V[1,*], Jodu Aravind[2], Ankith[2], Dhandapani Vaithiyanathan[1], Preeti Verma[1]

[1]National Institute of Technology Delhi, Delhi-110036, India

[2]Rajiv Gandhi University of Knowledge Technologies, Basar-504101, Telangana, India

*Author to whom correspondence should be addressed:

E-mail: anuraj@nitdelhi.ac.in

(Received October 12, 2024; Revised December 19, 2024; Accepted January 27, 2025).

Abstract: In exploring ways to optimize Multiply and Accumulate (MAC) units, this study conducted a thorough experimental analysis focusing on four distinct signed multiplier architectures: Booth's Multiplier (BO), Modified Booth's Multiplier (MBO), Vedic Multiplier (VM), Baugh-Wooley Multiplier (BA), and an advanced Baugh-Wooley Multiplier integrated with the Wallace Tree (BAW). Evaluations cover 8-bit, 16-bit, and 32-bit operations, scrutinizing critical performance metrics such as area utilization, power consumption, and critical path delay. A central theme of the study is integrating the Wallace Tree algorithm with the Baugh-Wooley Multiplier, which demonstrates significant enhancements in efficiency. This integrated approach reduces hardware requirements and achieves notable reductions in power consumption and critical path delay. These features make it particularly suitable for applications requiring robust and energy-efficient multiplier designs, such as those used in digital signal processing and embedded systems. Validation and analysis were rigorously conducted using the Xilinx tool on the Artix-7 AC701 Evaluation platform for delay and area, using the Cadence tool for power analysis, ensuring the reliability and relevance of the findings. The proposed Baugh-Wooley multiplier integrated with the Wallace tree structure results in approximately 60% reduced area, 70% decreased delay, and 73% lower power consumption in 32-bit multipliers. These impressive results position the Baugh-Wooley-Wallace combination as a promising advancement for MAC unit technologies, opening doors for further optimizations and practical applications. This study contributes significantly to the ongoing development of high-performance multiplier architectures tailored to meet contemporary computational demands.

Keywords: DSP; MAC; Vedic Multiplier; Booth's Multiplier; Baugh Wooley Multipliers; Wallace Tree Multipliers

1. Introduction

Very Large-Scale Integration (VLSI) is designing integrated circuits by incorporating millions of transistors onto a single chip. This advanced technology facilitates the creation of highly complex and compact semiconductor devices, significantly enhancing the performance, efficiency, and functionality of modern electronic systems across diverse applications. Digital Signal Processing (DSP) employs advanced digital techniques to analyze, modify, and synthesize signals. Utilizing sophisticated algorithms and specialized hardware, DSP enhances the accuracy and reliability of communication systems, audio, and video processing, and various other applications that demand real-time signal manipulation and enhancement[1)]. The Multiplier-Accumulator (MAC) unit is a digital circuit that excels at performing multiplication and accumulation operations with great efficiency. It is extensively utilized in Digital Signal Processing (DSP)[2)] and other computational applications to enhance overall performance. The MAC unit achieves this by swiftly executing multiply-and-add instructions, which are crucial to the algorithms employed in tasks such as filtering, convolution, and real-time data processing[3)].

The multiplier is a crucial component in digital circuit design, playing a key role in the functionality of VLSI (Very Large-Scale Integration), DSP (Digital Signal Processing), FIR Filter[4,5)], and MAC (Multiplier-Accumulator) units[6)]. In VLSI, multipliers allow for the integration of complex computational tasks within compact circuits, thereby enhancing the processing capabilities of semiconductor devices. In DSP, multipliers are essential for executing operations such as filtering and convolution, which are fundamental to signal processing.

The multiplier in a MAC (Multiply-Accumulate)[7] unit is crucial for carrying out the multiply and accumulates[8,9] operations that are fundamental to many DSP (Digital Signal Processing)[10] algorithms. This enables efficient real-time processing of digital signals, which is essential for high-performance modern electronic systems. It is more efficient in terms of speed and resource usage compared to Vedic and array multipliers, which have higher delays and greater slice utilization[11].

Apart from ASICs, FPGAs have become increasingly popular for VLSI applications[12] due to their flexibility and rapid prototyping capabilities. FPGAs allow designers to implement and test logic designs quickly, enabling them to verify functionality before committing to a physical chip design[13]. This iterative process is invaluable, as it reduces the risk of costly errors and ensures that the logic will perform as intended in real-world scenarios. By using FPGAs for prototyping, engineers can efficiently assess various design parameters and make adjustments on-the-fly, significantly streamlining the development process. Once a design has been validated on the FPGA, it can then be transitioned to a full IC implementation, paving the way for production with greater confidence in its performance[14]. In digital design, factors such as cost, energy consumption, and time efficiency are paramount; thus, leveraging FPGAs not only expedites development but also enhances overall project viability[15].

This paper aims to design and optimize a Multiply-Accumulate (MAC) unit in VLSI using advanced multipliers, with a focus on creating a low-delay, signed multiplier[16]. The goal is to improve the performance and efficiency of digital signal processing applications by developing a robust multiplier that can accurately and efficiently handle signed numbers. To achieve this, various multiplier strategies will be explored, including the Booth multiplier (BO)[17], The effectiveness of the Modified Booth multiplier (MBO), Vedic Multiplier(VM), the Baugh-Wooley multiplier (BA), and the Baugh-Wooley with Wallace tree (BAW) are being evaluated to reduce computation time and power consumption. This is aimed at improving overall system performance in terms of speed and energy efficiency[18,19].

In this paper, we proposed a new method of signed multiplier for MAC architecture. In this work, we combined the Baugh-Wooley with the Wallace tree for the multiplication operation. The Baugh-Wooley Multiplier is particularly effective for signed number multiplication because it handles the sign bit directly during the multiplication process, avoiding the need for extra correction steps. By combining the Baugh-Wooley for the multiplication operation, this new combination provides an immense performance increment in the area delay and power in the higher bits.

In this paper, Section 2 provides a discussion of these multipliers. Session 3 described the proposed work, section 4 presents the results, And Section 5 is the Conclusion.

2. Discussion On Multipliers

Signed multipliers in digital systems process signed binary numbers, crucial for operations in digital signal processing and arithmetic units. They accurately handle both positive and negative values, enabling complex computations. These multipliers are designed to efficiently manage the nuances of signed arithmetic, ensuring correct results across various applications[20]. A good amount of investigation is going on the signed multiplication steps[21]. Getting an optimized design in the multiplier is still in progress[22]. Not only in signed unsigned multipliers are also important in the digital market[23].

In today's fast-paced world, the quest for faster multiplication techniques has become a critical area of research and development. Among the emerging approaches, the exploration of approximate quantum multipliers[24,25] stands out as a promising avenue. These multipliers are designed to trade off a certain degree of accuracy for significant gains in speed and efficiency, making them particularly suited for applications where approximate results are acceptable. However, a key challenge in designing such multipliers is the accurate calculation and evaluation of errors introduced by their approximate nature. Error calculation becomes a crucial factor and is inherently tied to the performance metrics of these designs. These metrics not only include traditional factors like speed, power, and area but also account for the degree of approximation and its impact on the functionality of the multiplier. While approximate quantum multipliers can deliver high-performance metrics and significantly enhance the efficiency of multiplication processes, it is important to note that they are inherently designed to provide only approximate results. This trade-off highlights the need for careful consideration of their application in scenarios where exact precision is not mandatory but high-speed performance is critical.

2.1 Booth's Multiplier

Booth's multiplier[26] is a sophisticated hardware algorithm designed for the efficient multiplication of binary numbers, particularly adept at handling signed numbers in two's complement representation. The flow chart is shown in Fig. 1. The algorithm significantly reduces the number of addition and subtraction operations needed during multiplication by examining the multiplier bit-pairs, encoding transitions, and employing optimized arithmetic operations. Booth's algorithm begins with initializing the multiplicand M, the multiplier Q, an extra bit Q_{-1} set to 0, and an accumulator A set to 0. It then iterates through the bit pairs of the multiplier. For each pair (Q_i, Q_{i-1}), specific operations are performed: adding M to A if the pair is 01, subtracting M from A if the pair is 10, and doing nothing if the pair is 00 or 11. After each

operation, the combined value of A, Q, and Q−1 undergoes an arithmetic shift right (ASR), maintaining the sign bit. This process continues until all bit pairs are processed, resulting in a final product held in the concatenated values of A and Q[27]. Booth's algorithm enhances speed and reduces hardware resource utilization by minimizing the number of operations through exploiting patterns in the multiplier, making it highly suitable for high-speed digital systems and applications requiring rapid and efficient arithmetic computations[28].

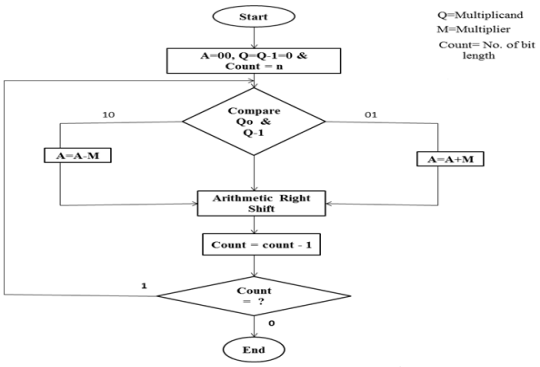

Fig. 1: Booth's multiplier[29]

2.2 Modified Booth Multiplier

The Modified Booth Multiplier reduces the number of partial products compared to the basic Booth Multiplier, enhancing efficiency. It performs signed and unsigned multiplications, with signed numbers in 2's complement form[30]. The algorithm is shown in Table 1. Consider two N-bit numbers, X and Y:

$$X = \sum_{i=0}^{N-1} x_i . 2^i + x_{N-1} . 2^{N-1} \quad (1)$$

$$Y = \sum_{i=0}^{N-1} y_i . 2^i + y_{N-1} . 2^{N-1} \quad (2)$$

where x_i and y_i denote the i^{th} bit of the multiplicand and multiplier, in this context, both X and Y are 16-bit numbers. If a negative number is used, it is represented in 2's complement form, whereas a positive number remains unchanged. The most significant bit (MSB) indicates the sign of the number[31].

The Modified Booth algorithm, introduced to address this, offers significantly faster multiplications[32]. One of its key advantages is that computation time is proportional to the length of the operands, enabling efficient processing of larger numbers. Additionally, it reduces the area of the multiplier block, making it more space-efficient. Other important features include its ability to handle both positive and negative numbers seamlessly and its improved efficiency in generating partial products, further contributing to its high-speed performance. These enhancements make the Modified Booth algorithm particularly advantageous for high-speed digital systems and applications requiring efficient arithmetic computations[33].

Table 1. Modified booth algorithm.

Y_{2i+1}	Y_{2i}	Y_{2i-1}	Recorded Digits	Operand Multiplication
0	0	0	0	0×Multiplicand
0	0	1	+1	+1×Multiplicand
0	1	0	+1	+1×Multiplicand
0	1	1	+2	+2×Multiplicand
1	0	0	-2	-2×Multiplicand
1	0	1	-1	-1×Multiplicand
1	1	0	-1	-1×Multiplicand
1	1	1	0	0×Multiplicand

2.3 Vedic Multiplier

In ancient India, Vedic mathematics was widely practiced as an advanced and efficient mathematical system, even before the 20th century. Among its numerous algorithms, the Urdhva Tiryagbhyam sutra stands out as an effective method for all types of multiplication. A fundamental building block of this approach is the 2x2 Vedic multiplier, which operates using a systematic sequence of logic. The process involves generating intermediate terms, such as

- C0 = x0 AND y0
- PP0 = x1 AND y0
- PP1 = x0 AND y1
- PP2 = x1 AND y1
- C1 = PP0 XOR PP1
- PP3 = PP0 AND PP1
- C2 = PP3 XOR PP2
- C3 = PP3 AND PP2

This logical structure ensures efficient computation of partial products and their simultaneous addition. The Vedic multiplier supports parallel processing, enabling faster and more efficient high-bit operations, making it a significant advantage over traditional multipliers. This technique not only simplifies complex calculations but also serves as a foundation for modern arithmetic circuits in digital systems[34].

2.4 Baugh Wooley Multiplier

The Baugh Wooley Multiplier is designed specifically for signed multiplication[35]. It takes inputs in 2's complement form and gives the output in 2's complement as well. The Baugh Wooley Multiplier can be used for both signed and unsigned multiplication. It is a simple algorithm involving a few steps for n*n bit multiplication[36]. The algorithm follows as:

- Invert the Most Significant Bits (MSB) of the first n-1 rows.
- Invert all partial products except for the Most Significant Bit (MSB) of the nth row.

- Add 1 to the n+1 column.
- Add 1 to the 2n column.

The Figure 2 is an example of a 4*4 Baugh Wooley Multiplier. The size of the Multiplicands (a, b) is 4-bit i.e., n=4. Since we represent the Multiplicands in the 2's Complement form, the minimum & maximum value it can handle for n*n Multiplier is -2^{n-1} to $2^{n-1} - 1$. This type of multiplier is commonly used in digital signal processing and other applications where fast multiplication of two numbers is required. The Baugh Wooley algorithm is known for its efficiency in performing signed binary multiplication, making it a popular choice in various computational tasks[37].

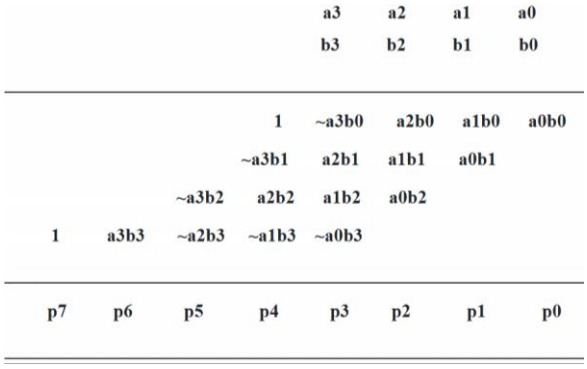

Fig. 2: Baugh wooley's algorithm.

The Baugh-Wooley Multiplier is specifically designed to handle signed number multiplication, a requirement in many digital signal processing and computing applications where negative numbers are common. The Baugh-Wooley approach simplifies the multiplication of signed binary numbers by modifying the partial products and arranging them in a way that inherently accounts for the sign bits. This method avoids the need for additional steps to correct the sign after the multiplication process, which is a significant advantage over traditional methods like Booth's algorithm. In Booth's multiplication, for instance, handling signed numbers requires additional operations that can increase the overall complexity and delay of the multiplier. The Baugh-Wooley multiplier addresses these challenges by directly incorporating the sign bit into the multiplication process, thereby streamlining the operation and reducing the overall latency. This makes it one of the fastest multipliers for signed binary numbers. Additionally, the Baugh-Wooley multiplier is efficient in terms of hardware usage. By organizing the partial products in a specific manner, it reduces the number of logic gates required, which in turn lowers power consumption and the area occupied by the circuit on a chip. This efficiency is particularly important in integrated circuit design, where space and power are often at a premium. However, while it excels in speed and efficiency, its structure can still be improved when dealing with larger bit-widths, where the complexity and delay can become more pronounced.

2.5 Wallace Tree Multiplier Algorithm

In 1964, C. S. Wallace introduced a binary multiplication technique known as the Wallace Tree multiplier[38], which significantly advanced the design of digital circuits by optimizing the way partial products are summed. This method arranges partial products in a hierarchical structure, reducing the number of sequential addition stages necessary to complete a multiplication operation. By systematically combining partial products using layers of full adders and half adders, the Wallace Tree minimizes the number of carry propagation steps, which are typically the bottleneck in traditional binary multipliers. Then, thereafter the different styles of modified Wallace tree algorithms are reported by the various researcher[39]. The reduction of these steps not only decreases the overall delay but also reduces the complexity of the hardware implementation, making the Wallace Tree multiplier particularly suitable for high-speed applications. In digital circuits, speed is often constrained by the critical path, the longest path that data must travel through a circuit. The Wallace Tree's architecture, by minimizing the depth of this critical path, allows for faster computation, which is crucial in performance-critical applications such as real-time data processing and digital signal processing. Despite these advantages, the Wallace Tree multiplier is primarily optimized for unsigned numbers, which limits its direct applicability in certain domains where signed number arithmetic is required. Nevertheless, its impact on the field of digital arithmetic cannot be overstated, as it laid the foundation for the development of faster and more efficient multiplication techniques that are still in use today[40].

The objective of this study is to develop a high-performance multiplier for signal processing applications. The findings indicate that while the Wallace tree multiplier excels in minimizing delay, it presents complexities when applied to signed multiplication techniques. In contrast, the Baugh-Wooley multiplier demonstrates a significant reduction in area compared to the Booth algorithm, which is widely recognized as one of the best methods for signed binary multiplication. Each approach offers distinct advantages: the modified Booth algorithm, for instance, optimizes both area and time delay, making it a compelling choice for efficient design. Overall, understanding the strengths and weaknesses of these various multiplier techniques is essential for advancing the performance of signal processing systems.

3. The proposed Baugh Wooley-Multiplier Integrated with Wallace Tree Algorithm (BAW)

The integration of the Wallace Tree and Baugh-Wooley Multiplier represents a powerful advancement in digital multiplication techniques by combining the strengths of both methods to create a more efficient and faster

multiplier. This hybrid approach leverages the hierarchical reduction of the Wallace Tree, which minimizes the number of sequential addition stages, with the Baugh-Wooley Multiplier's capability to efficiently handle signed numbers. The result is a multiplier that not only reduces the critical path delay but also decreases power consumption, making it ideal for modern digital applications that require both speed and efficiency. The integration works by first utilizing the Baugh-Wooley method to generate partial products that account for signed numbers and then applying the Wallace Tree's hierarchical reduction to sum these products quickly. By combining these two methods as shown in Fig. 3, the number of full adders and half adders required in the multiplication process is significantly reduced, which directly impacts the critical path length. A shorter critical path means that the multiplier can operate at a higher speed, which is crucial in applications where processing time is critical. Moreover, the reduction in the number of logic gates used translates to lower power consumption, which is increasingly important in today's energy-conscious world. This reduction in power consumption is not just a result of fewer active logic gates but also the decreased switching activity, which is a major contributor to dynamic power consumption in digital circuits. Additionally, the integration simplifies the overall design of the multiplier, making it more scalable and easier to implement in various digital systems.

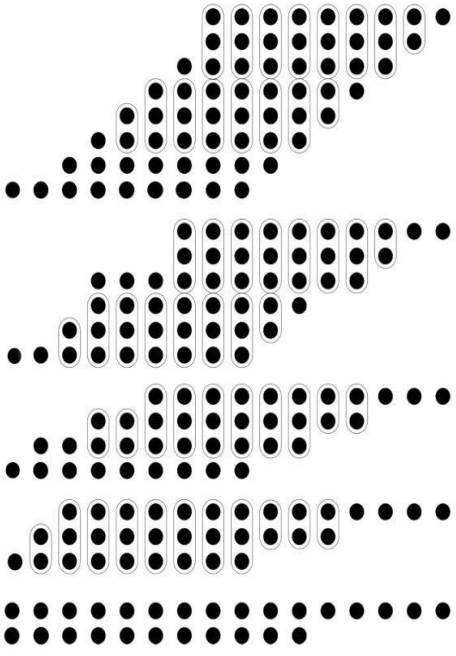

Fig. 3: Baugh wooley's integrated with wallace tree.

Overall, the proposed integration of the Wallace Tree and Baugh-Wooley Multiplier offers a significant improvement over the existing digital multiplication technique. The Baugh-Wooley algorithm efficiently handles signed number multiplication by generating partial products with inherent sign correction, simplifying the process. These partial products are hierarchically reduced using the Wallace Tree, which minimizes the critical path delay through parallel reduction stages with adders. At the core, NAND gates operate as universal logic gates, performing AND operations with inverted outputs, which are extensively used in constructing adders and other digital components in the architecture. By combining the speed and efficiency of the Wallace Tree with the robust handling of signed numbers provided by the Baugh-Wooley method, this hybrid multiplier sets a new standard for performance in digital circuits. It addresses the common trade-offs between speed, power consumption, and complexity that often limit the effectiveness of traditional multipliers. The reduction in critical path delay and power consumption, coupled with the simplified design, makes this integrated multiplier a compelling choice for a wide range of digital applications, particularly in fields like digital signal processing and computing, where rapid and accurate arithmetic operations are essential. The ability to efficiently handle large bit-widths and signed numbers without sacrificing performance further enhances the appeal of this approach. As digital systems continue to evolve, the demand for more efficient and faster multipliers will only increase, and the integration of the Wallace Tree and Baugh-Wooley Multiplier represents a forward-looking solution to meet these demands. Its application in modern digital design not only improves the performance of individual circuits but also contributes to the overall efficiency and effectiveness of complex digital systems. Therefore, this hybrid multiplier should be considered a superior and innovative solution for binary multiplication tasks, offering significant benefits in terms of speed, power efficiency, and design simplicity.

4. Result and Discussion.

The performance of the multipliers was evaluated based on their area, power consumption, and delay for different bit sizes 8-bit, 16-bit, and 32-bit. The comparison included four types of multipliers: Booth's Multiplier, Modified Booth's Multiplier, Vedic Multiplier, Baugh Wooley Multiplier, and the new Baugh Wooley integrated with Wallace Tree Multiplier. These multipliers were tested and analyzed using Verilog HDL and synthesized with the Xilinx tool on the Artix-7 AC701 Evaluation Platform. For the power analysis we use a cadence tool and using GPDK 45nm,90nm, and 180nm technology.

From Table 2 and Fig. 4, it is evident that the Baugh-Wooley with Wallace tree (BAW) only utilizes 46 % of the area reduction compare to Vedic multiplier. And compared to the Baugh-Wooley multiplier (BA), indicating a 36.6% reduction in slice utilization and thus occupying less area. Consequently, the BAW demonstrates superior area efficiency over the BA, providing crucial benefits for contemporary VLSI designs.

From Table 3 and Fig. 5, In the comparison of various 32-bit multipliers delay and the last column it compares

the frequency of 32-bit multiplier, it was observed that the Booth multiplier (BO) exhibited the highest delay at 66.730 ns. In contrast, the Modified Booth Multiplier (MBO) demonstrated a delay of 35.353 ns, which is 52.97% less compared to the Booth multiplier. The Baugh-Wooley Multiplier (BA) showed a delay of 25.813 ns, amounting to a reduction of 38.68% relative to the Booth multiplier. In the Vedic multiplier it shows 21.352 ns is also 3 ns higher than the proposed multiplier. Notably, the Baugh-Wooley Multiplier with Wallace Tree (BAW) exhibited the lowest delay at 18.641 ns, representing only 27.93% of the delay experienced by the Booth multiplier. Based on these results, it is evident that the Baugh-Wooley Multiplier with Wallace Tree configuration achieves the lowest delay among all the multipliers analyzed.

The core multiplier operates efficiently at a frequency of 53.63 MHz, which corresponds to the calculated latency of 18.641 ns. This indicates that the proposed multiplier is capable of handling this frequency seamlessly. Throughput, which refers to the amount of data processed per unit time, is directly related to this frequency. Since the design is a purely combinational circuit with no pipelining or intermediate storage, the latency is equivalent to the critical path delay of 18.641 ns. This demonstrates the multiplier's capacity to handle data at the specified rate without additional delays.

Table 2. Slices utilization.

Multiplier Type	8-bits	16-bits	32-bits
BO [33]	147	660	2423
MBO [33]	107	529	2021
VM[34]	117	511	2024
BA [36]	77	401	2587
BAW (Proposed)	77	407	1641

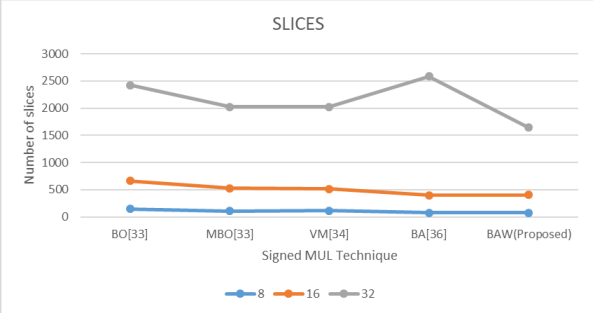

Fig. 4: Area Utilization comparison of multipliers.

Table 3. Delay in ns.

Multiplier Type	8-bits	16-bits	32-bits	Frq.of 32-bit (MHz)
BO[33]	16.24	34.465	66.730	14.99
MBO[33]	10.82	19.802	35.353	28.29
VM[34]	10.04	13.89	21.352	46.84
BA[36]	9.107	13.976	25.815	38.73
BAW (Proposed)	9.348	12.953	18.641	53.63

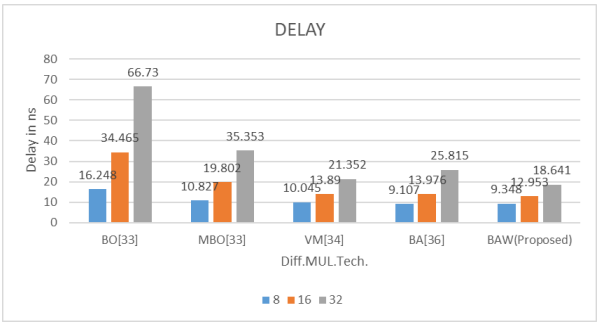

Fig. 5: Delay (ns) comparison of multipliers.

From Table 4 and Fig. 6, it is evident that for 32-bit multipliers using 180nm technology, the Booth multiplier exhibits the highest power consumption at 24.15309 milliwatts(mW). In contrast, the Baugh-Wooley Multiplier (BA) has a power consumption of 15.71274 mW, which is only 65.054% of the Booth multiplier's power consumption. The Modified Booth Multiplier (MBO) shows a power consumption of 12.7686 mW, representing only 50.82% of the Booth multiplier's power consumption.

Table 4. Power consumption of different signed multipliers across various technologies.

Multiplier	Bits	45nm (power mW)	90nm (power mW)	180nm (power mW)
BO[33]	8	0.01020	0.06272	0.32974
	16	0.59032	0.63621	3.01409
	32	3.92850	4.41629	24.1530
MBO[33]	8	0.08611	0.05356	0.24007
	16	0.53551	0.43234	1.91271
	32	3.58385	2.60545	12.2768
VM[34]	8	.047884	.078193	.301376
	16	.288403	.455505	1.75649
	32	1.53650	2.35837	9.23462
BA[36]	8	0.08432	0.06125	0.32514
	16	0.58905	0.37318	2.27608
	32	2.53762	3.67554	15.7127
BAW (Proposed)	8	0.10687	0.07247	0.40248
	16	0.34527	0.52591	2.09564
	32	1.87123	2.77970	11.3436

BO: Booth's Multiplier, MBO: Modified Booth's Multiplier, BA: Baugh-Wooley Multiplier, BAW: Baugh-Wooley Integrated with Wallace Tree

Additionally, the Baugh-Wooley Multiplier with Wallace Tree (BAW) has the lowest power consumption at 11.34361 mW, which is only 46.96% of the power consumption of the Booth multiplier. In 180 nm, 90 nm, and 45 nm are widely utilized to evaluate and optimize power efficiency. Among various implementations, the VM demonstrates significant advantages over other types of multipliers, offering superior performance in terms of power consumption. when comparing signed multipliers, the proposed design exhibits even better results, showcasing its effectiveness in handling signed operations with enhanced accuracy and efficiency.

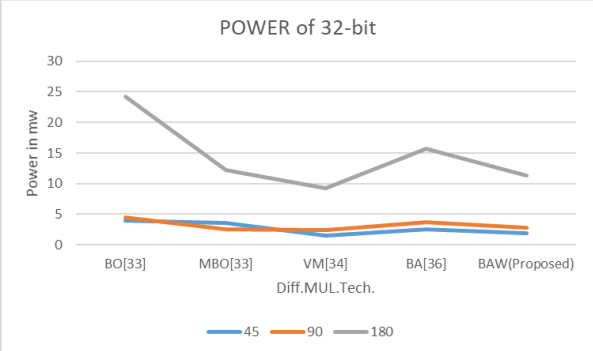

Fig. 6: Power (mW) comparison of 32-bit in 3 technology multipliers.

From these results, it is clear that the Baugh-Wooley Multiplier with Wallace Tree configuration achieves the lowest power consumption among all the multipliers analyzed. From Fig.7 and Fig.8, The 16-bit MAC (Multiply-Accumulate) unit demonstrates exceptional performance and efficiency. It operates with a delay of 15.377 nanoseconds, which is impressively fast for its bit-width. It utilizes 431 slice LUTs, reflecting its complex computational capabilities. Despite this complexity, the power consumption remains remarkably low at 0.714453 mW. The multiplier in this unit can handle values ranging from -2^{15} to $2^{15}-1$, and the constant can range from -2^{31} to $2^{31}-1$. The MAC unit produces outputs within the range of -2^{31} to $2^{31}-1$. However, it is important to note that the unit is susceptible to overflow if results exceed $2^{31}-1$ and underflow if results fall below -2^{31}. Overall, the 16-bit MAC unit offers high performance, efficient resource utilization, and low power consumption, making it a robust solution for complex computational tasks within its defined operational limits.

The integration of the Baugh-Wooley multiplier with the Wallace Tree can become less efficient for very large bit-widths due to increased design complexity and diminishing returns in delay and area optimization. The Baugh-Wooley algorithm generates partial products while efficiently handling signed numbers, avoiding additional correction steps. These partial products are then fed into the Wallace Tree, which hierarchically reduces them using adders to minimize the critical path delay and hardware complexity.

5. Conclusion.

In this paper, we have conducted a detailed analysis of various signed multiplier architectures, including Booth, Modified Booth, Baugh-Wooley, and Baugh-Wooley integrated with Wallace tree multipliers, focusing on key performance metrics such as area utilization, power consumption, and delay. Our findings demonstrate that the Baugh-Wooley multiplier offers superior performance, exhibiting the lowest delay, reduced power consumption, and minimal slice utilization compared to the other architectures analyzed. These results highlight the Baugh-

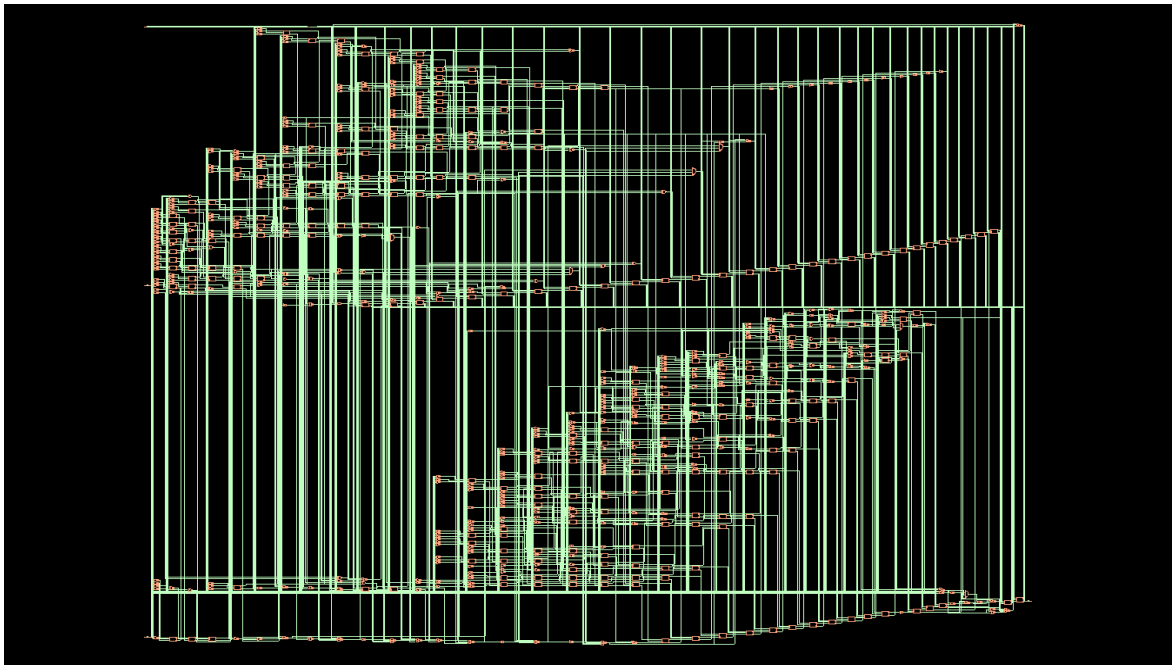

Fig. 7: 16-bit MAC BAW unit using 90nm GPDK slow corner.

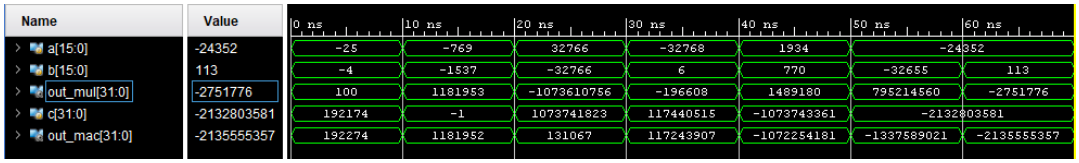

Fig. 8: Simulation of 16-bit MAC using BAW.

Wooley multiplier's efficiency and effectiveness, particularly when integrated with the Wallace tree, making it the most suitable choice for applications demanding high-speed, low-power, and resource-efficient multipliers. Future research will aim to further optimize these multipliers and explore their practical implementations like to corroborate our theoretical insights. Digital signal processing applications, such as speech processing, loudspeaker equalization, echo cancellation, adaptive noise cancellation, and various communication applications, including software-defined radio[41] need to done using this technique.

Acknowledgments

The authors acknowledge the Chips to Startup (C2S) grant from the Ministry of Electronics and Information Technology (MeitY), Govt. of India for EDA tools support.

Declarations

• Funding: Not Applicable.

• Conflicts of Interest: The authors declare no conflict of interest.

• Ethical Approval: Not Applicable.

• Data Availability: The manuscript has no associated data

References

1) M. Gnanasekaran and M. Manikandan, "Low delay-high compact FIR filter using reduced Wallace multiplier," *Second International Conference on Current Trends In Engineering and Technology - IC-CTET, Coimbatore, India,* 400-404 (2014). doi:10.1109/IC-CTET.2014.69662.

2) S. Saranya, G. Sudha, S. Subbiah, D. P. P, R. B, and S. S, "Design of an Efficient Mac Unit for DSP Applications," *Intelligent Computing and Control for Engineering and Business Systems (ICCEBS), Chennai, India,* 1-6 (2023). doi:10.1109/ICCEBS58601.2023.10448585.

3) S. U. Siddamshetty, S. Nambi, S. Boppu and D. Ghosh, "Efficient Multiplication and Accumulation of Signed Numbers," *IEEE International Symposium on Smart Electronic Systems (iSES), Warangal, India,* 130-135 (2022). doi:10.1109/iSES54909.2022.00036.

4) M. Sreenivasa Reddy, Mahesh K. Singh, "FIR Filter Complexity Analysis Using Multi-Rate Signal Processing Approach," *Evergreen,* 11 (2) 1263-1267 (2024). doi:10.5109/7183433

5) A. H. A. Bayoumi, S. A. Ibrahim and H. A. H. Fahmy, "Fast Parallel Multiple Access Distributed Arithmetic (FPMA-DA) Reconfigurable FIR Filter," *International Conference on Microelectronics (ICM), Abu Dhabi, United Arab Emirates,* 1-6 (2023). doi:10.1109/ICM60448.2023.10378942.

6) H. S. O. Migdadi, R. A. Abd-Alhameed, H. A. Obeidat, J. M. Noras, E. A. A. Qaralleh and M. J. Ngala, "FIR implementation on FPGA: Investigate the FIR order on SDA and PDA algorithms," *Internet Technologies and Applications (ITA), Wrexham, UK,* 417-421(2015), doi: 10.1109/ITechA.2015.7317439.
7) E. Kalali and R. van Leuken, "Near-Precise Parameter Approximation for Multiple Multiplications on a Single DSP Block," *in* IEEE Transactions on Computers*,* 71 (9) 2036-2047 (2022). doi:10.1109/TC.2021.3119187.
8) V. N. Kumar, K. R. Nalluri and G. Lakshminarayanan,"Design of area and power efficient digital FIR filter using modified MAC unit,"*2nd International Conference on Electronics and Communication Systems (ICECS), Coimbatore, India,* 884-887 (2015). doi:10.1109/ECS.2015.7125041.
9) A. Chauhan and P. Sathish Kumar, "Implementation of FIR Filter & Mac Unit by Using Neural Networks in FPGA," *International Conference on Advances in Computing, Communications and Informatics (ICACCI), Bangalore, India,* 2496-2501 (2018). doi:10.1109/ICACCI.2018.8554523.
10) G. S, E. S, C. Bhavya and Y. J. M. Shirur, "Design and Implementation of Arithmetic based FIR Filters for DSP Application," *International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE), Bengaluru, India,* 782-787 (2023). doi:10.1109/IITCEE57236.2023.10090953.
11) B. Zhou, G. Wang, G. Jie, Q. Liu and Z. Wang, "A High-Speed Floating-Point Multiply-Accumulator Based on FPGAs," *in IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* 29 (10) 1782-1789 (2021). doi:10.1109/TVLSI.2021.3105268.
12) A. K. Jameil and H. Al-Raweshidy, "CNN Architecture on FPGA Using High Level Module for Healthcare Devices," *in* IEEE Access*,* 10 60486-60495 (2022). doi:10.1109/ACCESS.2022.3180829.
13) S. Moradi Cherati, G. Jaberipur and L. Sousa, "Sparse Matrix-Vector Multiplication Based on Online Arithmetic," *in* IEEE Access*,*12 87653-87664 (2024). doi: 10.1109/ACCESS.2024.3416395.
14) S. Rooban, M. Nagesh, M. V. S. L. Prasanna, K. Rayudu and G. D. Sai, "Implementation of 128-bit Radix-4 Booth Multiplier," *International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India,* 1-7 (2021). doi:10.1109/ICCCI50826.2021.9457004.
15) P. Sanjana, M. Ramesh, A. Kale, A. A. Anita and P. Sasipriya, "Design and Evaluation of Error Tolerant Booth Multipliers for Image Processing Applications," *4th International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India,* 1414-1418 (2022). doi:10.1109/ICSSIT53264.2022.9716363.
16) J.Christopher Vishal, et al., "Design of Reconfigurable Multiply-Accumulate Unit with Computational Optimization," *Microelectronic Devices, Circuits and Systems: Second International Conference, ICMDCS, Vellore, India,* (2021).
17) S. Radhakrishnan, R. K. Karn, M. A. Meerasha and T. Nirmalraj, "Design of Low Power and High-Speed MAC-based FIR Filter using Hybrid Adder and Modified Booth Multiplier," *5th IEEE International Conference on Emerging Electronics (ICEE), New Delhi, India,* 1-4 (2020). doi:10.1109/ICEE50728.2020.9776857.
18) R. S. Mishra, P. Gour, S. Dhariwal, G. Kumar and A. Anand, "Design and Analysis of Low Power MAC for DSP Processor," *International Conference on Artificial Intelligence and Applications (ICAIA) Alliance Technology Conference (ATCON-1), Bangalore, India,* 1-3 (2023). doi:10.1109/ICAIA57370.2023.10169461.
19) H.S.R Vamsi, K.S Reddy, C. Babu, N.S Murty, "Design of reversible logic based 32-bit mac unit using radix-16 booth encoded Wallace tree multiplier,"*International Conference on Computer Communication and Informatics (ICCCI),* 1–6 (2018). doi.org/10.1109/ICCCI.2018.8441263.
20) Immareddy, Srikanth, and Aunmetha Sundaramoorthy, "A survey paper on design and implementation of multipliers for digital system applications," 55 4575–4603 (2022). doi.org/10.1007/s10462-021-10113-0.
21) L. Du *et al*., "A Low-Power DNN Accelerator With Mean-Error-Minimized Approximate Signed Multiplier," *in* IEEE Open Journal of Circuits and Systems*,* 5 57-68 (2024). doi:10.1109/OJCAS.2023.3279251.
22) Y. Zhang *et al*., "An Energy-Efficient Binary-Interfaced Stochastic Multiplier Using Parallel Datapaths," *in* IEEE Transactions on Very Large Scale Integration (VLSI) Systems*,* 31 (9) 1439-1443 (2023). doi:10.1109/TVLSI.2023.3247739.
23) L. Sayadi, S. Timarchi and A. Sheikh-Akbari, "Two Efficient Approximate Unsigned Multipliers by Developing New Configuration for Approximate 4:2 Compressors," *in* IEEE Transactions on Circuits and Systems I: Regular Papers*,* 70 (4) 1649-1659 (2023). doi:10.1109/TCSI.2023.3242558.
24) A. Crimmins and S. L. Alarcon, "Approximate Quantum Array Multiplier," *researchgate.net* (2024).
25) S. Mariserla, T. V. Murthy and K. Pradeep, "High Speed VLSI Architecture of Approximate Multiplier Using Majority Logic," *IEEE International Conference on Quantum Computing and Engineering (QCE), Montreal, QC, Canada,* 58-66 (2024). doi:10.1109/QCE60285.2024.00017.
26) L. P. Rubinfield, "A Proof of the Modified Booth's Algorithm for Multiplication," *in* IEEE Transactions

on Computers*, C-24,* (10) 1014-1015 (1975). doi:10.1109/T-C.1975.224114.

27) B. V. Mahesh and T. Srivasarao, "Performance Evaluation of FFT through Adaptive Hold Logic (AHL) Booth Multiplier," *International Conference for Advancement in Technology (ICONAT), Goa, India,* 1-6 (2023). doi:10.1109/ICONAT57137.2023.10080290.
28) S. Phuse and P. P. Tasgaonkar, "Design and Implementation of Different Multiplier Techniques and Efficient MAC Unit on FPGA," *International Conference on Signal and Information Processing (IConSIP), Pune, India,* 1-5 (2022). doi:10.1109/ICoNSIP49665.2022.10007482.
29) C. V. S. Chaitanya and P. Kumar, "Design and Analysis of Booth Multiplier with Optimised Power Delay Product," *International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India,* 1-6 (2018). doi:10.1109/ICCCI.2018.8441236.
30) T. Gaurav, K. Patel and R. Parekh, "RTL to GDSII Implementation of RADIX-4 Booth Multiplier," *IEEE International Conference on Nanoelectronics, Nanophotonics, Nanomaterials, Nanobioscience & Nanotechnology (5NANO), Kottayam, India,* 1-5 (2022). doi:10.1109/5NANO53044.2022.9828885.
31) V. Manjunath, K. Harikiran, S. Manikanta, Sivanantham and K. Sivasankaran, "Design and implementation of 16×16 modified booth multiplier," *Online International Conference on Green Engineering and Technologies (IC-GET), Coimbatore, India,* 1-5 (2015). doi:10.1109/GET.2015.7453817.
32) Y. S. J, S. M. G, M. G, and K. Mariammal, "Design of Efficient Booth Multiplier based Polyphase FIR Filters,"*Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS), Trichy, India,* 1834-1841 (2023). doi:10.1109/ICAISS58487.2023.10250454.
33) M.R. Shakya, P. Jindal, " Comparative analysis of 8-bit and 16-bit array multiplier, modified booth multiplier," *Proceedings of the 3rd International Conference on Contents, Computing & Communication (ICCCC-2022).*
34) S.BANDI. "RoBA Multiplier-Driven FIR Filter Synthesis: Uniting Efficiency and Speed for Enhanced Digital Signal Processing." *Journal of VLSI circuits and systems*, 6(2) 23-30 (2024).
35) P. Kishore, B. Akash, G. Aditya and N. Harika, "Design and Analysis of Low Power and High-Speed Baugh Wooley Multiplier using Modified Gate Diffusion Input Technique,"*3rd International Conference for Innovation in Technology (INOCON), Bangalore, India,* 1-7 (2024). doi:10.1109/INOCON60754.2024.10512080.
36) V. Ponugoti, S. Oruganti, S. Poloju, "Design of baugh-wooley multiplier using full swing gdi technique," *In:" Soft Computing and Signal Processing", Springer Nature,* 769– 779 (2022).
37) V. Pala, V. Makhe, K. Bhuva and R. Parekh, "RTL to GDSII Flow Implementation of 8-bit Baugh-Wooley Multiplier," *IEEE International Conference on Nanoelectronics, Nanophotonics, Nanomaterials, Nanobioscience & Nanotechnology (5NANO), Kottayam, India,* 1-6 (2022). doi:10.1109/5NANO53044.2022.9828876.
38) S. Wallace,"A Suggestion for a Fast Multiplier,"in *IEEE Transactions on Electronic Computers, EC-13,* (1) 14-17 (1964). doi:10.1109/PGEC.1964.263830.
39) S. Raveendran, P. J. Edavoor, Y. B. N. Kumar and M. H. Vasantha, "In-exact Signed Wallace Tree Multiplier Design Using Reversible Logic," *in IEEE Access,*9 108119-108130 (2021). doi:10.1109/AC-CESS.2021.3100892.
40) S. Madhavi, K. Rasagna, N. Kavya, M. Sindhu and V. Kaveri, "Implementation of Programmable FIR Filter Using Dadda Multiplier and Parallel Prefix Adder," *International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India,* 585-589 (2018). doi:10.1109/ICIRCA.2018.8597249.
41) S. Arivalagan, B. P. James and M.F. Leung. "Reconfigurable Frequency Response Masking Multi-MAC Filters for Software Defined Radio Channelization"*, Electronics*,13(21) 4211 (2024).