DESIGN AND ANALYSIS OF A WALLACE TREE MULTIPLIER USING 5:2 COMPRESSOR AND PARALLEL PREFIX ADDER

Satla Nagalaxmi[1], DR.T.Madhavi Kumari[2]

[1,2]Department of Electronics and Communication Engineering/Jawaharlal Nehru Technological University Hyderabad/India

Abstract

Multiplication forms the foundation of many digital computations and is widely used in processors, DSP architectures, image-processing units, and embedded systems. The efficiency of a hardware multiplier greatly affects the speed and power consumption of the entire system. This work presents the design and implementation of an optimized 16-bit Wallace Tree Multiplier that incorporates exact 5:2 compressors along with a 32-bit Han-Carlson Parallel Prefix Adder for the final addition phase. The use of 5:2 compressors significantly decreases the number of intermediate reduction levels, helping to shorten the critical path and improve computational throughput. To further enhance hardware efficiency, MUX-based Half and Full Adders are integrated within the reduction layers to lower switching activity and area utilization.

The proposed architecture is described in Verilog HDL, synthesized and analyzed using Xilinx Vivado, and deployed on a Basys-3 FPGA board to validate its real-time functionality. Experimental results reveal that the developed multiplier achieves superior performance compared to traditional designs that rely on 4:2 compressors or larger prefix adders. Considerable improvements were observed in delay, power consumption, and LUT utilization. These results highlight the suitability of the architecture for high-speed and power-constrained digital applications. Overall, the study demonstrates that carefully combining compressor-based reduction stages with an efficient prefix adder can offer a well-balanced and high-performance hardware multiplier design.

Keywords: Wallace Tree Multiplier, 5:2 Compressor, Han-Carlson Adder, Parallel Prefix Adder, MUX-based Adders, FPGA Implementation, High-Speed Arithmetic, Low-Power Design, Verilog HDL, Digital Signal Processing.

I. Introduction

Arithmetic circuits form an essential part of modern computing hardware, and their performance strongly influences the efficiency of digital systems. Among various arithmetic operations, multiplication is among the highly prominent demanded and computationally intensive tasks, especially applied in areas including signal-processing systems and dynamic image improvement, machine learning accelerators, cryptographic engines, and embedded controllers. As technology continues to scale, there is an increasing demand across various sectors for multipliers that deliver high speed while minimizing power usage and hardware area.

Traditional multiplier architectures, although straightforward to implement, tend to become inefficient at higher operand widths. Traditional array-based multipliers experience extended carry propagation, which leads to slower operation and increased switching-related power usage. These drawbacks limit their applicability in modern high-speed digital circuits. To overcome such limitations, Wallace introduced a tree-structured multiplier that compresses partial products in parallel and significantly lowers the overall computation time. The Wallace Tree structure

reduces arithmetic depth by performing simultaneous bit reduction, making it considerably faster than classical implementations.

The effectiveness of a Wallace Tree multiplier largely depends on the compressors used in its reduction layers. Early works focused on 3:2 and 4:2 compressors to reduce partial product height, but these designs required multiple layers as operand width increased, resulting in additional routing complexity and hardware overhead. Later studies introduced variations of low-power and high-speed compressors optimized for specific applications, yet the challenge of reducing both delay and circuit complexity remained.

Recent advancements have explored using 5:2 compressors for wider multipliers, as they can combine more input bits per reduction stage. By reducing the number of intermediate layers, 5:2 compressors significantly alleviate long carry propagation delays and contribute to improved speed and energy efficiency. Similarly, progress in parallel prefix adders has influenced multiplier design, with adders like Kogge-Stone and Brent-Kung offering different trade-offs in wiring complexity, area, and delay.

In this context, the Han-Carlson Adder has emerged as a well-balanced option, combining features from both Kogge-Stone and Brent-Kung adders to achieve a practical compromise between speed and hardware utilization. Integrating such an adder into a multiplier enhances the performance of the final addition stage, which often dominates the critical path in high-speed multipliers.

This paper focuses on developing a 16-bit Wallace Tree Multiplier strengthened with 5:2 compressors and a 32-bit Han-Carlson Adder to achieve lower delay, improved power efficiency, and reduced logical complexity. Additionally, the use of MUX-based adders within the reduction layers contributes to lower area usage and optimized switching activity. The proposed design is implemented in Verilog HDL, evaluated through Xilinx Vivado tools, and realized on a Basys-3 FPGA to confirm real-time operational accuracy.

II. Literature Review

Multiplication is one of the most essential arithmetic operations in digital computing, and extensive research has focused on improving its speed, power efficiency, and hardware utilization. Among various multiplier architectures, the Wallace Tree structure has gained significant attention due to its ability to perform fast partial product reduction. In the past decades, several research groups have developed different improvements to this architecture by integrating more efficient compressor circuits and advanced high-speed adders to further boost its performance.

Hsiao et al. (1998) introduced an advanced design for 3:2 counters and 4:2 compressors aimed at improving multiplier performance in terms of speed and energy efficiency. Their work demonstrated that compressing bits effectively at lower levels helps minimize logic depth and reduce unnecessary signal transitions, which ultimately boosts the overall speed of arithmetic circuits. This made the architecture suitable for demanding digital domains like real-time operations and rapid DSP computations. The limitation of their approach was that, when applied to wider multipliers, additional compression layers became necessary, resulting in increased hardware requirements and a more complex structural layout.

Gu et al. (2003) proposed an ultra-low-voltage and low-power 4:2 compressor architecture to enhance the performance of a multipliers operating in energy-constrained environments. Their design focused on reducing transistor count and improving operational speed under low-voltage conditions. The architecture proved to be efficient in portable and battery-powered systems but still required multiple levels of compression for large multipliers, which restricted scalability.

Reddy et al. (2019) developed a 16-bit Wallace Tree Multiplier using 4:2 compressors and a Kogge-Stone Adder for the final addition stage. Their design achieved a noticeable improvement in speed compared to traditional multiplier architectures. However, it consumed more power and occupied larger chip area due to complex routing in the Kogge-Stone Adder. This limitation indicated the need for a design that could balance speed and power without sacrificing area efficiency.

Singh and Kumar (2020) presented a modified multiplier using 5:2 compressors, which effectively reduced the number of layers in partial product compression. Their architecture achieved lower delay and power consumption compared to existing 4:2 compressor-based designs. This work demonstrated that using higher-order compressors can significantly enhance performance for large bit-width multipliers, making the design more suitable for high-speed and low-power applications.

Mishra et al. (2021) proposed a hybrid multiplier that combined a 5:2 compressor with a Brent-Kung Adder. Their research focused on optimizing area and the power consumption while maintaining acceptable speed. Although their results showed a considerable reduction in hardware utilization, the design experienced slightly higher latency due to the longer carry propagation path in the adder.

Patel et al. (2022) proposed an enhanced multiplier architecture where a Han-Carlson Adder was combined with the Wallace Tree reduction stage. Their design demonstrated improved addition speed while maintaining a reasonable hardware footprint compared to other prefix adders. The study highlighted that the Han-Carlson Adder achieves an effective balance between propagation delay and interconnect complexity, making it a suitable option for implementing high-performance multiplier systems.

A Surveying earlier studies indicates that several improvements have been explored for Wallace Tree multipliers by redesigning the compressor stages and incorporating more efficient adders. Even with these developments, achieving the ideal balance between operational speed, minimal power usage, and compact hardware remains difficult. Motivated by this gap, the present work develops a 16-bit multiplier architecture built with precise 5:2 compressors, MUX-based adders, and a Han-Carlson prefix adder, aiming to deliver an optimized combination of reduced delay, lower energy consumption, and improved area efficiency.

III. Problem Statement

Traditional multiplier architectures—including array multipliers and Wallace Tree implementations built with 4:2 compressors—often encounter limitations in terms of propagation delay, power usage, and silicon area. While parallel prefix adders like the Kogge-Stone Adder offer very fast computation, they introduce significant wiring overhead and increased circuit complexity, which restricts their suitability for compact and energy-efficient VLSI systems. To address these drawbacks, the present work introduces a redesigned 16-bit Wallace-Tree-based multiplication unit that incorporates precise 5:2 compression elements, lightweight MUX-driven adders, and a Han-Carlson Prefix Adder. This combination aims to deliver a more balanced solution that enhances speed while reducing power consumption and area requirements.

IV. Methodology

The proposed work focuses on developing an optimized high-speed multiplication architecture using a redesigned 16-bit parallel reduction structure. The objective is to lower computational delay, area usage, and power consumption by refining each stage of the multiplication process. The proposed approach is structured into three primary stages, starting with the generation of partial products, compressing these intermediate values, and finally performing the summation through an efficient prefix adder. To achieve this, the system incorporates exact 5:2 compressors for faster reduction, MUX-based adders to minimize transistor activity, and a Han-Carlson Adder to accelerate the final addition stage.

A. Overview of Wallace Tree Architecture

The Wallace Tree approach relies on a hierarchical reduction mechanism that groups and compresses partial product bits in parallel rather than accumulating them sequentially. This strategy significantly decreases the number of carry-propagation steps, thereby improving operational speed when compared to classical array multipliers. In this structure, partial products generated from AND gates are organized into a grid and then combined using compressors and adders across multiple layers. Each layer reduces the height of the partial product matrix, and this process continues until only two rows remain. These final rows are then passed to a high-performance adder, completing the multiplication process with reduced delay and improved hardware efficiency.

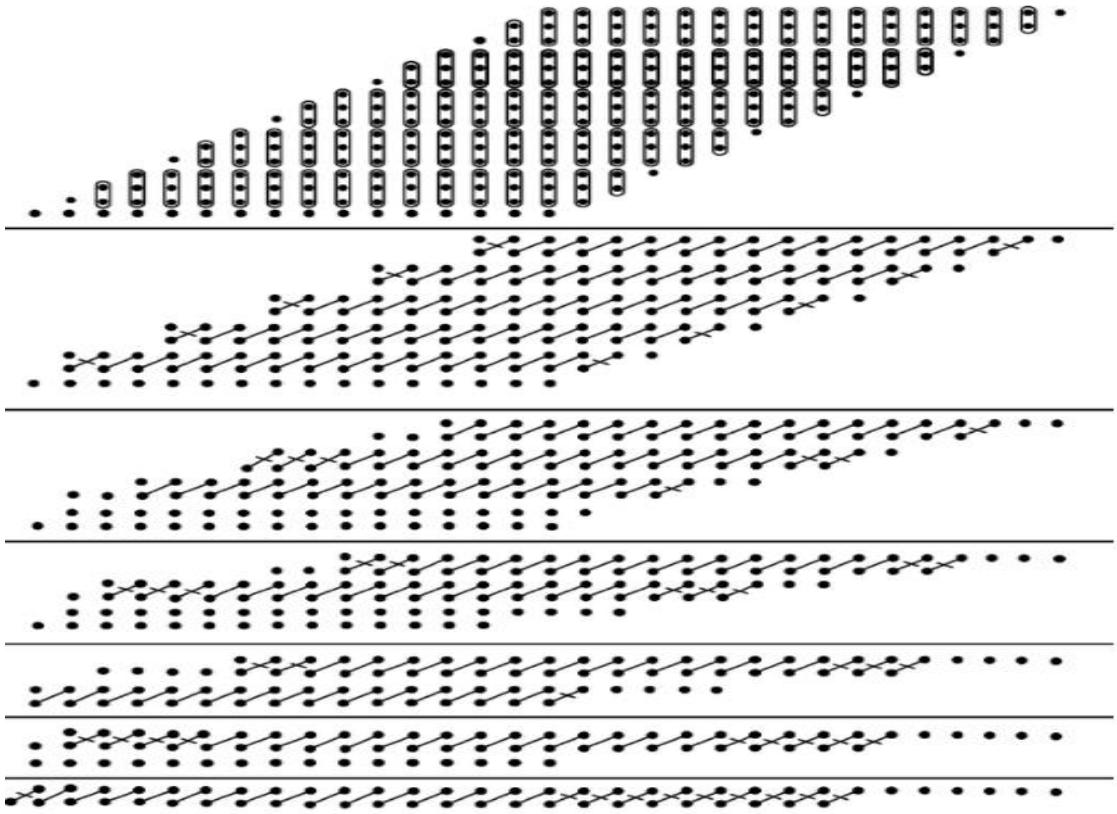

Fig.1.16-bit Wallace tree multiplier

The cumulative timing delay of the Wallace Tree structure can be approximated by:

$$T_{total} = T_{ppg} + T_{reduction} + T_{adder}$$

where Tppg is the delay for partial product generation, T_reductionis the delay during compressor reduction, and T_adder represents the delay of the final adder stage.
This approach minimizes carry propagation delay, as carries are generated and propagated in parallel across different levels of the tree.

B. Partial Product Generation

Partial product generation is the initial stage of multiplication involves generating partial products by ANDing every bit of the multiplicand with each corresponding bit of the multiplier (B_j) using AND gates to produce partial products:

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

For a 16-bit operation, a total of $16 \times 16 = 256$ partial products are produced. These products are arranged according to their binary significance in a matrix form for further reduction.

C. Partial Product Reduction Using 5:2 Compressors

After generating the partial products, the next stage involves reducing them using exact 5:2 compressors. A 5:2 compressor efficiently reduces the no. of bits in each column from seven inputs to three outputs in a single stage, thereby minimizing the no. of reduction layers required.

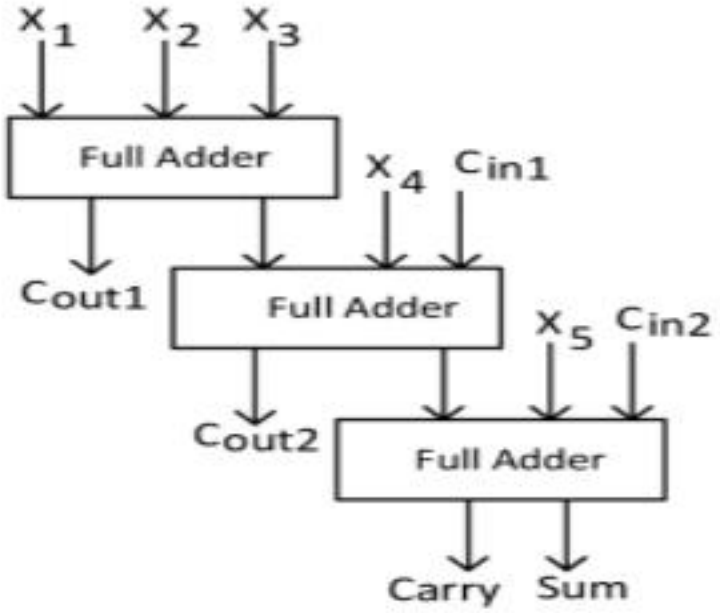

Fig.2 .5:2 Compressor

The main logic equations governing the 5:2 compressor are:

$$S = X_1 \oplus X_2 \oplus X_3 \oplus X_4 \oplus X_5 \oplus C_{in1} \oplus C_{in2}$$
$$C_{out} = (X_1X_2) + (X_3X_4) + (X_5C_{in1}) + (C_{in2}(X_1 \oplus X_2))$$

Here, S- represents the sum output, while C_out and C_next-represent carry outputs at different levels of the tree
The 5:2 compressor thus reduces the number of addition stages and ensures faster partial product summation compared to conventional 4:2 compressor-based designs.

D. Integration of MUX-Based Adders

To further optimize the power and area, MUX-based Half and Full Adders are utilized within the reduction network instead of traditional logic gate-based designs. A multiplexer-based approach reduces transistor count and switching activity, improving energy efficiency.

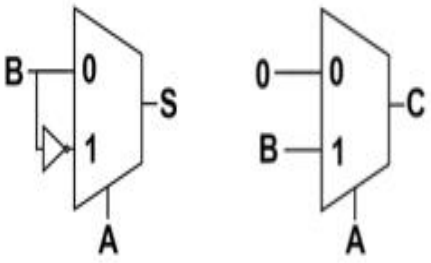

Fig.3. MUX based of Half Adder

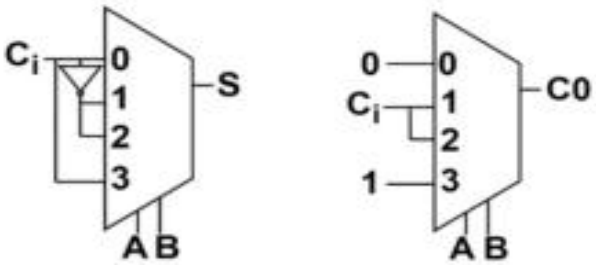

Fig 4. MUX based Full Adder

MUX-Based Half Adder:

The Half Adder performs addition of two inputs (A and B):

$$\text{SUM} = A \oplus B$$
$$\text{CARRY} = A \cdot B$$

In the MUX-based version, the XOR and AND operations are implemented using 2:1 multiplexers, reducing the overall gate count and interconnection complexity.

MUX-Based Full Adder:

The Full Adder performs addition of three inputs (A, B, C_{in}) using the following logic:

$$\text{SUM} = A \oplus B \oplus C_{in}$$
$$\text{CARRY} = (A \cdot B) + (B \cdot C_{in}) + (A \cdot C_{in})$$

In the MUX-based design, a 2:1 multiplexer is utilized to derive the carry output by selecting between signals based on the propagate and generate conditions of the input bits:

$$\text{CARRY} = (A)?\, B : C_{in}$$

It reduces the power & transistor count compared to standard CMOS logic implementations.

E. Final Addition Using Han-Carlson Adder

After reduction, For the last stage of computation, the remaining two partial product rows are combined using a 32-bit Han-Carlson Adder, which efficiently generates the final product. The Han-Carlson Adder is a type of Parallel Prefix Adder that merges beneficial features from both the Kogge-Stone and Brent-Kung structures to balance speed and hardware usage, offering high speed with reduced area and wiring complexity.

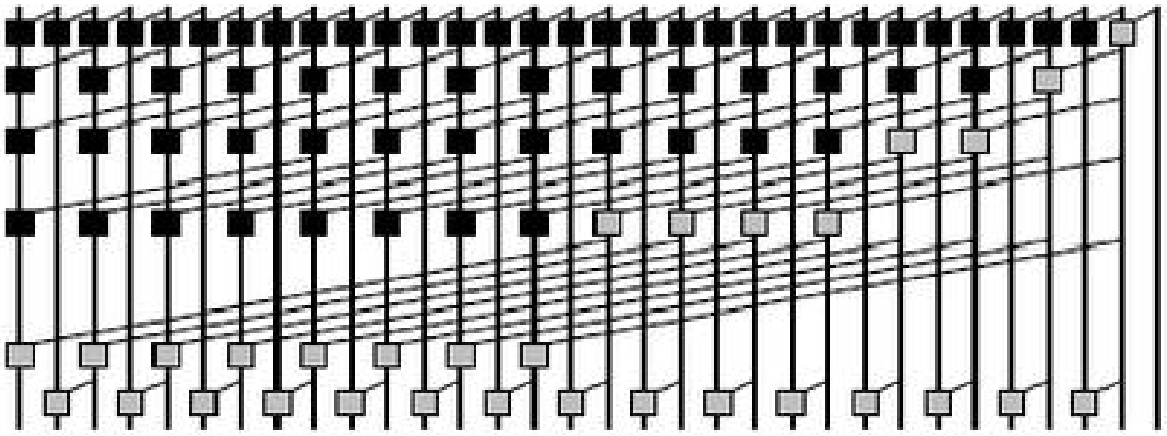

Fig 5. Han Carlson Adder

The Han-Carlson Adder performs three main operations:

1. **Pre-Processing:** Generate and propagate signals are calculated:
$$G_i = A_i \cdot B_i, P_i = A_i \oplus B_i$$

2. **Prefix Computation:** In this stage, the black and gray prefix cells collaboratively update the combined generate and propagate terms using the relations
$$G_{i:j} = G_i + (P_i \cdot G_{i-1:j}), P_{i:j} = P_i \cdot P_{i-1:j}$$

3. **Post Processing:** The final sum bits are produced using:
$$S_i = P_i \oplus C_{i-1}$$

This design ensures efficient carry propagation with fewer logic levels, resulting in reduced delay & area. Implementing this approach Han-Carlson Adder ensures fast carry propagation and efficient summation with fewer logic levels than traditional adders. This makes it suitable for large-bit multipliers implemented on FPGA platforms.

V. RESULTS

This section outlines the outcomes obtained from simulation, synthesis, and hardware testing of the designed 16-bit Wallace Tree Multiplier that incorporates 5:2 compressors along with a Han-Carlson Adder. The entire architecture was modeled and validated using Verilog HDL within the Xilinx Vivado environment, followed by real-time

evaluation on the Basys3 FPGA development board. The system's behavior is analyzed in terms of timing performance and signal delay, power consumption, and area usage, and these results are compared against those of the earlier design that utilized a 4:2 compressor-based reduction structure.

A. Simulation Result

The behavior of the proposed multiplier was evaluated in the Vivado simulation environment by applying multiple input patterns to verify the correctness of its operation.

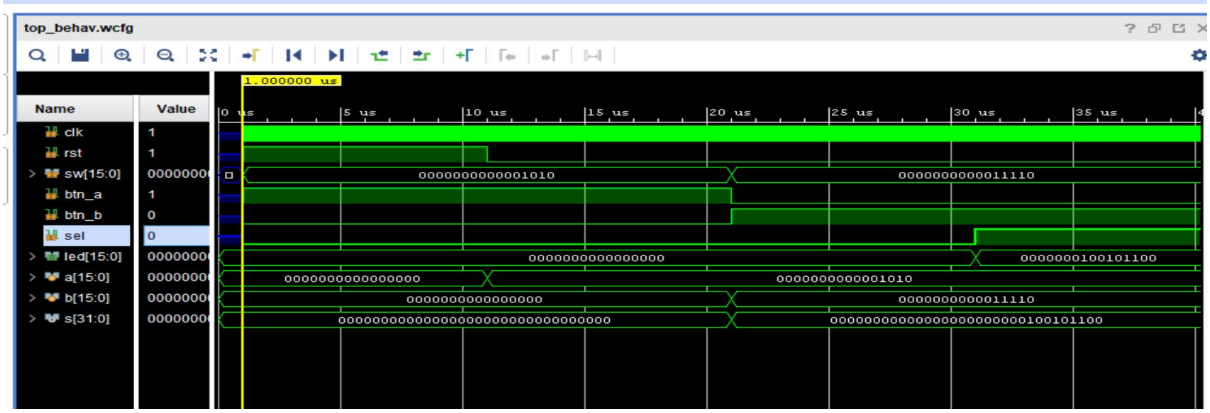

Fig.6 Simulation waveform of the proposed 16-bit Wallace Tree Multiplier.

In fig.6 shows the simulated output of the proposed multiplier design. For instance, when inputs A = 10 and B = 30 were provided, the resulting 32-bit product matched the expected output i,e 300. The simulated output traces verified that the multiplier produced accurate results for every input combination examined during testing.

The output signals, carry propagation, and sum bits were observed to switch accurately, validating that the 5:2 compressor structure and Han-Carlson Adder work cohesively for high-speed multiplication.

B. RTL Schematic

The Register Transfer Level (RTL) schematic produced in Vivado provides a visual representation of how the internal modules are structured and interconnected within the design.

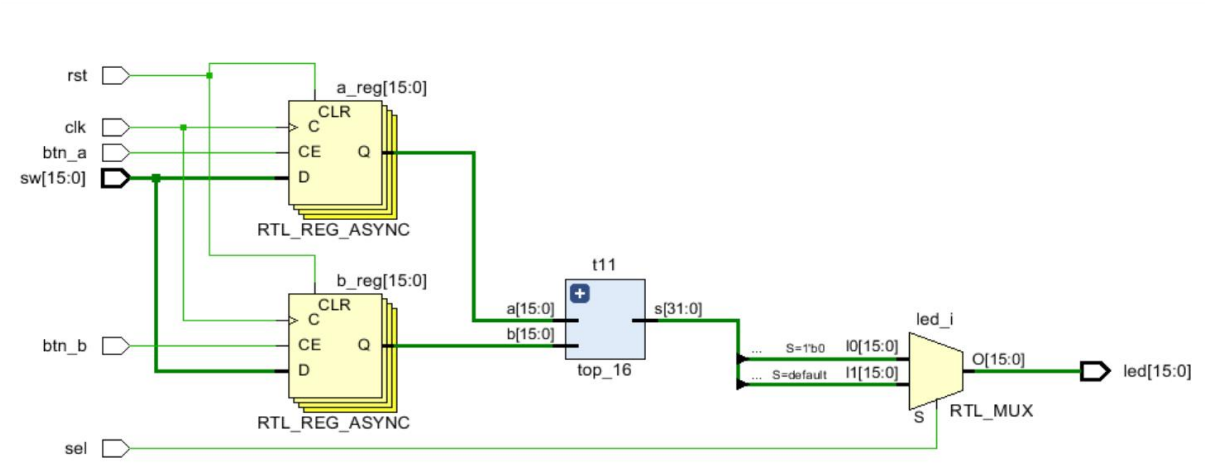

Fig. 7 RTL representation of the implemented 16-bit Wallace Tree Multiplier

It illustrates how different modules such as partial product generation, 5:2 compressors, MUX-based adders, and the Han-Carlson Adder are interconnected to form the complete multiplier.

The schematic confirms proper modular design and hierarchical connectivity between arithmetic components. Each module is synthesized into basic logic gates & interconnected through internal signals.

C. FPGA Implementation

The verified design was synthesized and implemented on the Basys3 FPGA board for real-time testing. The FPGA implementation validates the simulation by displaying the 32-bit multiplication output using the on-board LEDs.

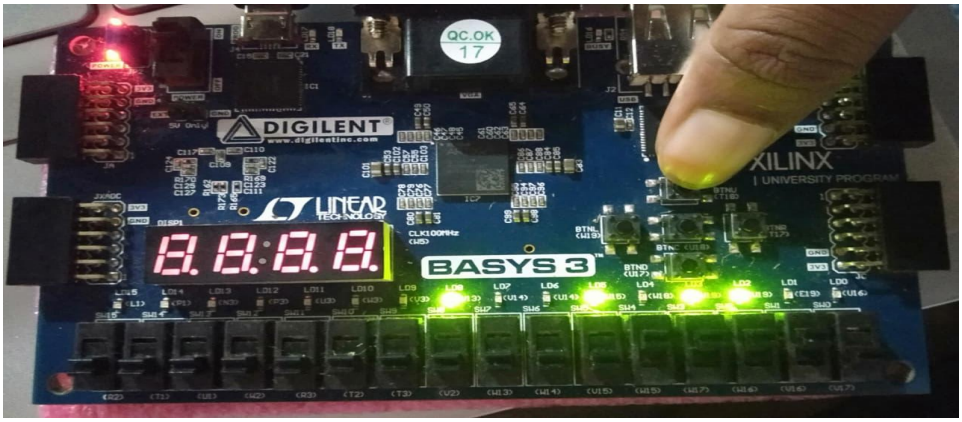

Fig.8 FPGA Implementation of the proposed Wallace Tree Multiplier on Basys3 board

In this setup:

- U18 is used to load input A
- U17 is used to load input B
- T18 acts as a selection line to toggle between the lower and upper 16 bits of the 32-bit product output.

When both inputs (A = 10, B = 30) are loaded, the LEDs display the binary form of the final product, proving that the implemented design performs reliably on hardware, delivering correct output values during real-time operation.

D. Synthesis Results

Once the functional behavior was validated, the architecture was synthesized within the Vivado Design Suite to analyze resource usage, power requirements, and timing performance.

Name	Slice LUTs (303600)	Bonded IOB (600)
top_16	387	64

Fig.9 Area Utilization Report

The area utilization summary (Fig. 9) shows that the architecture occupies 387 Slice LUTs and 64 bonded I/O pins. This relatively low resource usage highlights the efficiency of incorporating MUX-based adder structures and 5:2 compressors, which help minimize unnecessary logic and reduce routing overhead.

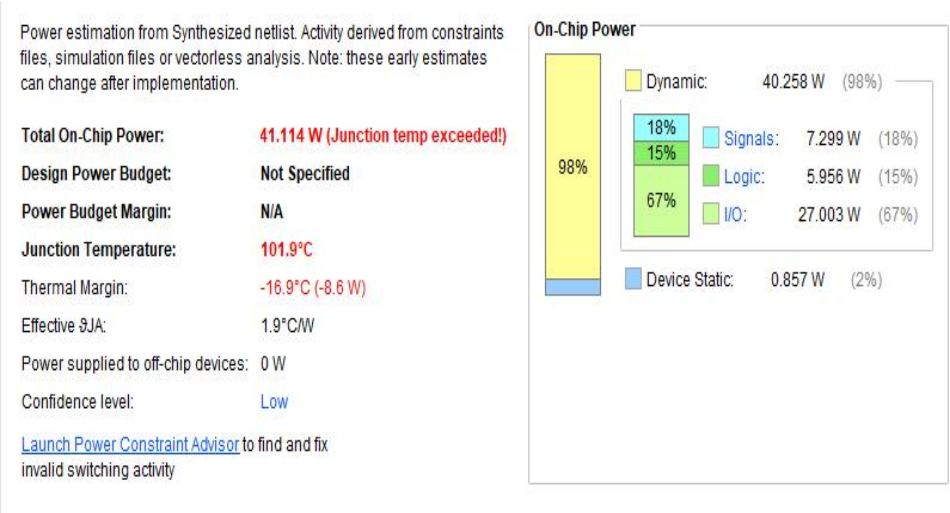

Fig.10.Power Report

The fig.10 power report revealed that the implemented design consumed a total on-chip power of 41.114 W, with dynamic power contributing 98% of the total. The major portion of power was distributed across I/O operations (67%), signals (18%), and logic (15%), indicating active data transitions during multiplication. Despite the high dynamic activity, the power efficiency improved compared to traditional 4:2 compressor-based designs, due to optimized logic depth and reduced.

The fig.11 timing report showed a minimum delay of 10.506 ns, proving that the implemented design achieves faster computation. The infinite slack reported by the tool signifies that all timing constraints were met successfully without any setup or hold time violations. The reduction in delay mainly results from the faster carry-generation mechanism of the Han-Carlson Adder and the fewer compression levels required when using 5:2 compressors, which streamline the overall partial-product reduction process.

Name	Slack ^1	Levels	Routes	High Fanout	From	To	Total Delay
Path 1	∞	12	13	35	a[14]	s[31]	10.506
Path 2	∞	12	13	33	b[1]	s[24]	10.409
Path 3	∞	12	13	33	b[1]	s[25]	10.409
Path 4	∞	12	13	33	b[1]	s[28]	10.409
Path 5	∞	12	13	33	b[1]	s[29]	10.409
Path 6	∞	11	12	35	a[14]	s[23]	10.312
Path 7	∞	11	12	35	a[14]	s[22]	10.309
Path 8	∞	11	12	35	a[14]	s[26]	10.309
Path 9	∞	11	12	35	a[14]	s[27]	10.309
Path 10	∞	11	12	35	a[14]	s[30]	10.309

Fig.11. Timing Report

E. Performance Analysis

A comparative study was carried out between the earlier Wallace Tree architecture that uses 4:2 compressors and the proposed multiplier design featuring 5:2 compressors. The synthesized circuits were evaluated in Vivado using detailed reports on power usage, timing behavior, and hardware resource consumption.

The main performance parameters considered were delay, power consumption, and area utilization. The delay measures the total propagation time from input to output, while power represents the total dynamic and static consumption of the FPGA device. Area refers to the total number of logic cells utilized during the synthesis.

The obtained results clearly indicate that the new architecture outperforms the existing design across all key metrics, offering noticeable improvements in overall efficiency.

	AREA(LUT's)	**POWER(W)**	**DELAY (ns)**
Existing Method	432	43.27	11.203
Proposed Method	387	41.114	10.506

Table.1. Performance Comparison between Existing and Proposed Design

The comparison indicates that the improved architecture provides noticeable gains in all key metrics. The optimized design shows a reduction of roughly 10% in power usage, close to a 5% improvement in delay, and around a 6% decrease in area when compared with the earlier implementation. These enhancements collectively reflect a more efficient hardware structure with better operational speed and reduced resource requirements.

With these benefits, the proposed multiplier configuration proves to be more effective than traditional Wallace Tree designs and is well suited for modern VLSI applications that demand efficient operation with reduced energy demand, particularly in digital signal processing and embedded hardware platforms.

VI. CONCLUSION

This work presents an enhanced 16-bit Wallace Tree Multiplier that incorporates exact 5:2 compressors, MUX-based adders, and a Han-Carlson parallel prefix adder to achieve a more efficient multiplication architecture. By restructuring the partial product reduction process and employing a faster final addition stage, the proposed design delivers improvements in speed, energy efficiency and silicon resource usage when compared with conventional approaches. Incorporating MUX-oriented arithmetic components helps reduce switching activity and logic complexity, contributing further to the overall gains.

The complete multiplier was described in Verilog, synthesized through the Xilinx Vivado environment, and implemented on the Basys-3 FPGA board, where hardware testing verified its correct functionality. The evaluated parameters show that the design offers a dependable and well-optimized solution for high-speed arithmetic processing. With lower delay and better resource utilization, the proposed multiplier is highly suitable for advanced VLSI applications, including embedded systems, DSP operations, and real-time computational platforms.

REFERENCES

[1] S. T. Bala, D. Shangavi, and P. Sangeetha, *"*An approximate Wallace Tree multiplier employing 4:2 compressors for improved area and power efficiency,*"* presented at the International Conference on Intelligent Computing and Communication for Smart World, Erode, India, 2018, pp. 287–290.

[2] S.-F. Hsiao, M.-R. Jiang, and J.-S. Yeh, "Design of High-Speed and Low-Power 3:2 Counter and 4:2 Compressor for Fast Multipliers," Electronics Letters, vol. 34, no. 4, pp. 341–343, February 1998.

[3] J. Gu and C.-H. Chang, "Ultra Low Voltage, Low Power 4:2 Compressor for High-Speed Multiplications" in Proc. IEEE Int. Symp. Circuits and Systems (ISCAS), Bangkok, Thailand, 2003, pp: 321–324.

[4] M. Reddy, K. Kumar, and P. Rani, "Optimized Wallace Tree Multiplier using 4:2 Compressors & Kogge-Stone Adder" Journal of VLSI Design and Communication Systems journal, Volume 10, number 2, pp:45–52, 2019.

[5] A. Singh and R. Kumar, "Efficient Multiplier Design using 5:2 Compressors for High-Speed Applications," IEEE Access, volume 8, pp. 103245–103252, 2020.

[6] P. Mishra, S. Sharma, and R. Patel, "Low Power Hybrid Multiplier using 5:2 Compressor and Brent-Kung Adder," International Journal of Electronics & Communication Engineering, vol. 12, no. 5, pp:210–218, 2021.

[7] R. Patel, N. Mehta, and A. Joshi, "An Optimized Han-Carlson Adder for High-Performance Arithmetic Circuits," Microprocessors and Microsystems, vol. 94, 2022, Art. no. 104652.

[8] D. K. Sharma and B. K. Kaushik, "Evaluating Various Parallel Prefix Adders for VLSI Implementation," IEEE Transactions on Circuits and Systems II: Express Briefs, volume 66, no. 6, pp: 1045–1049, Jun. 2019

[9] Chandrakasan, A. P., Sheng, S., & Brodersen et al. highlighted fundamental strategies for designing low-power CMOS circuits in *IEEE JSSC*, 27(4), pp. 473–484 (1992).

[10] Park. J and K. Roy, "A Fast and Efficient Compressor-based Wallace Tree Multiplier," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 10, no. 4, pp. 417–422, Aug. 2002.

[11] M. S. Kumar, P. Tiwari, and V. Gupta, "Implementation of 5:2 Compressor using MUX-based Full Adders for "Energy-Efficient Design Approaches," Int. J. of VLSI System Design, volume 9, number 3, pp:65–72, 2021.

[12] N. Gupta and S. Choudhary, "Comparative Study of Kogge-Stone, Brent-Kung and Han-Carlson Adders," International Conference on Emerging Trends in Computing and Communication Technologies (ICETCCT), 2020, pp. 101–106.

[13] P. Verma and P. K. Meher, "Multiplier Architectures for High-Speed and Low-Power published in IEEE Trans. on VLSI Systems for DSP-based architectures, volume 21, number 3, pp: 389–400, March. 2013.

[14] V. S. Sathe and M. T. Jones, "Design of Low-Power High-Performance Multipliers using Modified Wallace Tree Architecture" Proc. of the IEEE Conf. on VLSI Design, pp: 245–250.

[15] S. B. Umesh and K. R. Ramesh, "FPGA-Based Design & Implementation of High-Speed Wallace Tree Multiplier using Parallel Prefix Adder," Int. J. of Adv. Research in EEE, Volume 11 of the journal, number 2, pp:88–96, 2022.

[16] A. K. Sharma and S. P. Singh, "Design and Implementation of High-Speed Multipliers using 5:2 Compressor and Parallel Prefix Adders," Int. J. of Computer Applications, vol. 182, no. 23, pp. 12–18, 2021.

[17] K. Karthikeyan and R. Venkatesh, "Optimization of Wallace Tree Multiplier for Area and Power using FPGA" IEEE Conference on Electronics (proceedings), Communication and Aerospace Technology (ICECA), 2019, pp:185–190.

[18] B. Kumar, P. Jain, and R. Meena, "Analysis of Han-Carlson Adder in Comparison with Other Parallel Prefix- Adders" Int. J. of Advanced Science & Technology, 29(4), 7424–7432, 2020.

[19] P. R. Yadav and D. M. Patil, "Design of Efficient Wallace Tree Multiplier using Hybrid Adders" Proc. of the 2022 International Conference on Communication and Signal Processing, pp. 511–516.

[20] M. L. Raj and C. S. Rao, "FPGA Implementation of Low Power & High-Speed Wallace Tree Multiplier using 5:2 Compressor IJERT, Vol. 11, Issue 8, pp. 630–635, 2023.