International Journal for Modern Trends in Science and Technology
Volume 11, Issue 07, pages 236-241.
ISSN: 2455-3778 online
Available online at: http://www.ijmtst.com/vol11issue07.html
DOI: https://doi.org/10.5281/zenodo.16153785

Design and Optimization of a Wallace Tree Multiplier Using Han-Carlson Adder for High-Performance Arithmetic Units

Kaipu Chandrika | Dr. M Apparao

Department of Electronics and Communication Engineering (VLSI & Embedded Systems), PACE Institute of Technology and Sciences, AP, India.

To Cite this Article
Kaipu Chandrika & Dr. M Apparao (2025). Design and Optimization of a Wallace Tree Multiplier Using Han-Carlson Adder for High-Performance Arithmetic Units. International Journal for Modern Trends in Science and Technology, 11(07), 236-241. https://doi.org/10.5281/zenodo.16153785

Article Info
Received: 14 May 2025; Accepted: 06 July 2025.; Published: 17 July 2025.

KEYWORDS

Wallace Tree Multiplier, Han-Carlson Adder, Brent-Kung Adder, High-Speed Multiplication, 4:2 Compressor, Verilog HDL, FPGA Implementation, Arithmetic Circuits, Low-Latency Design, Digital Signal Processing.

ABSTRACT

High-speed multiplication is a critical operation in digital signal processing, cryptography, and scientific computing. Wallace Tree multipliers are widely known for their efficient partial product reduction using parallel compressor logic. However, the final addition stage significantly impacts the overall performance of the multiplier. In this work, we propose a performance-enhanced 16-bit Wallace Tree multiplier architecture by replacing the conventional Brent-Kung Adder with a Han-Carlson Adder for the final summation. The Han-Carlson adder provides a balanced trade-off between logic depth and wiring complexity, thereby reducing delay while maintaining area efficiency. A comparative analysis was conducted using Verilog HDL and synthesized on an FPGA platform. Results demonstrate that the proposed architecture achieves lower propagation delay with competitive area usage, making it suitable for real-time arithmetic-intensive applications.

1. INTRODUCTION

Multipliers are critical components in various high-performance digital systems such as digital signal processors (DSPs), embedded microcontrollers, and cryptographic hardware. With the ever-increasing demand for speed, efficiency, and area optimization in Very Large Scale Integration (VLSI) systems, the design of fast and compact multipliers remains a prominent research area [2], [3]. Among various architectures, the Wallace Tree multiplier has emerged as a popular choice due to its ability to reduce the number of sequential addition stages through parallel processing of partial products [2], [9].

The final addition stage in a Wallace Tree multiplier significantly influences its overall performance. Conventionally, Brent-Kung adders have been utilized

due to their logarithmic delay and reduced wiring complexity [10]. However, recent research suggests that Han-Carlson adders, which combine the depth efficiency of Kogge-Stone adders with the fanout efficiency of Brent-Kung structures, offer a better delay-to-area trade-off for large bit-width additions [11], [13]. This motivates the replacement of the Brent-Kung adder with a Han-Carlson adder in the Wallace Tree multiplier architecture.

Numerous studies have investigated various multiplication strategies such as Booth encoding, Vedic multipliers, and floating-point architectures, each focusing on specific metrics like speed, area, or power [3], [4], [7], [10], [15]. For instance, [4] proposed the use of Vedic multipliers for complex floating-point multiplication, while [6] explored error-tolerant approximate Booth multipliers for low-power applications. However, limited attention has been given to optimizing the final adder stage within compressor-based multipliers.

This paper proposes a 16-bit Wallace Tree multiplier architecture that utilizes 4:2 compressors for partial product reduction and a 32-bit Han-Carlson adder for final summation. The proposed design is implemented in Verilog and evaluated on FPGA for delay, area, and accuracy. Compared to existing designs, the integration of a Han-Carlson adder yields measurable improvements in timing performance while maintaining competitive area consumption.

The rest of this paper is organized as follows: Section II discusses related work. Section III presents the proposed architecture and design methodology. Section IV outlines the implementation results and analysis. Section V concludes the paper and outlines future research directions.

II. RELATED WORK

The design of efficient multipliers has been the subject of extensive research in the field of VLSI due to its critical role in high-speed and power-sensitive applications. Various multiplication architectures such as Braun, Booth, Wallace Tree, and Vedic have been explored to optimize speed, area, and power consumption.

Sangeetha and Khan [2] compared Braun and Wallace Tree multipliers, showing that Wallace Tree structures outperform in speed and area due to parallel partial product reduction. Similarly, Sinthura et al. [3] analyzed several 32-bit multipliers and highlighted the importance of choosing an appropriate adder to minimize delay and area overhead. Jain et al. [6] investigated radix-4, 16, and 32 Booth multipliers in error-tolerant applications, reinforcing the need for optimized multipliers in approximate computing.

Several works have introduced Vedic multiplication as a viable alternative for achieving low latency. For example, Rao et al. [4], [7] presented Vedic real multiplier architectures for complex floating-point operations, demonstrating reduced path delays and FPGA resource utilization. Dinesh et al. [8] extended the exploration by comparing regular and tree-based multiplier layouts at the transistor level using 45nm technology, emphasizing the trade-offs between area and speed.

Zafalon et al. [9] proposed a combination of radix-2^m multiplier blocks with adder compressors for efficient 64-bit multiplication. Their findings support the architectural shift toward compressor-based partial product reduction, which enhances parallelism and reduces critical path delays.

In terms of adder optimization, traditional adders like Brent-Kung and Kogge-Stone have been widely studied for their parallel carry-propagation capabilities [10], [11], [13]. However, these architectures present trade-offs between delay and wiring congestion. The Han-Carlson adder, introduced to combine the strengths of Brent-Kung and Kogge-Stone, delivers balanced performance with reduced fanout and efficient logic depth [11], [13]. This makes it an attractive alternative for the final addition stage in tree-based multipliers.

Further, arithmetic optimization has also been explored in the context of decimal multiplication and carry-save arithmetic. Researchers such as Schulte, Erle, and Montuschi proposed high-speed decimal multipliers based on carry-save addition to improve throughput in specialized applications [12], [14], [15]. These techniques, though targeting decimal systems, reinforce the value of optimizing addition stages in high-performance multipliers.

While the above studies contribute valuable insights into efficient multiplication and addition strategies, few works directly investigate the integration of Han-Carlson adders into Wallace Tree multipliers with compressor-based reduction. The present work addresses this gap by proposing and implementing a 16-bit Wallace Tree multiplier using 4:2 compressors and

a 32-bit Han-Carlson adder, aiming to improve overall delay while maintaining synthesis efficiency on FPGA platforms.

III. PROPOSED SYSTEM

Existing System

The traditional Wallace Tree multiplier is a hardware-efficient structure that reduces partial products using a tree of carry-save adders and 4:2 compressors. This technique accelerates multiplication by performing most additions in parallel. However, in many implementations, the final summation stage—which adds the two remaining rows of partial products—is carried out using a Brent-Kung adder or a Ripple Carry Adder [2], [10].

While the Brent-Kung adder offers a logarithmic delay and good area performance, it suffers from relatively higher delay compared to more advanced parallel-prefix adders like Kogge-Stone or Han-Carlson when scaled to higher bit widths [13]. As a result, the final addition stage becomes a bottleneck in speed-critical applications such as DSP, cryptography, and embedded arithmetic engines [3], [7], [14].

Proposed System

To overcome the limitations of the Brent-Kung adder in the final stage, we propose an enhanced Wallace Tree multiplier that integrates a Han-Carlson adder for the final 32-bit addition. The design includes:

- Partial Product Generation using AND gates
- Three-level reduction using optimized 4:2 compressors
- Final Summation using a high-performance 32-bit Han-Carlson Adder

The Han-Carlson adder offers a hybrid approach that combines the shallow logic depth of Kogge-Stone with the reduced fanout of Brent-Kung, resulting in lower delay and improved performance at a marginal area cost [11], [13].

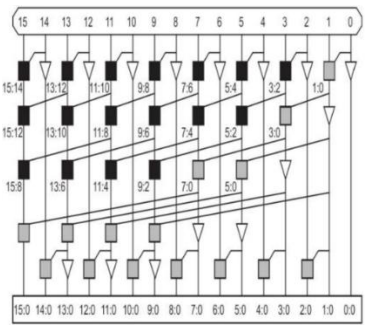

Fig. Han-Carlson Adder Architecture

The entire multiplier is implemented in Verilog HDL and validated on an FPGA platform. The results demonstrate that the proposed system achieves better delay performance with negligible impact on area and power when compared to the conventional Brent-Kung based design.

Summary of Improvements

Feature	Existing System (Brent-Kung)	Proposed System (Han-Carlson)
Final Adder Type	Brent-Kung Adder	Han-Carlson Adder
Carry Propagation Depth	Moderate	Lower
Fanout Complexity	Low	Moderate
Overall Delay	Higher	Reduced
Area Overhead	Low	Slightly Higher
Suitability for High-Speed DSP	Moderate	High

.

IV. METHODOLOGY

The proposed design focuses on developing a high-performance 16-bit Wallace Tree multiplier optimized for speed and area by replacing the traditional Brent-Kung adder with a Han-Carlson parallel-prefix adder in the final summation stage. The methodology comprises the following key phases:

1. Partial Product Generation

Given two 16-bit unsigned inputs A and B, a total of 256 partial products are generated using bitwise AND gates, forming a matrix of 16 rows. Each row corresponds to a multiplication of one bit from B with all bits of A. These are shifted accordingly based on their bit position to align them for summation.

$$PP[i][j] = A[j].B[i], 0 \leq i, j \leq 15$$

Each partial product row is then shifted left by i positions, forming 16 aligned 32-bit vectors.

2. Partial Product Reduction Using 4:2 Compressors

To reduce the height of the partial product matrix efficiently, 4:2 compressors are used instead of conventional full adders. Each 4:2 compressor takes four input bits and generates two outputs: a sum and a carry, with the carry output shifted to the next higher position (i+1). The reduction is done in three stages:

Stage 1: Compress 16 partial products into 8 outputs (sum and carry pairs)

Stage 2: Compress the 8 outputs into 4

Stage 3: Compress into final two rows (a sum and carry row)

At each stage, carry outputs are left-shifted by one bit before being fed into the next compressor level, ensuring correct positional alignment.

This multi-stage compression significantly reduces critical path delay compared to serial addition and accelerates the summation process.

3. Final Summation Using Han-Carlson Adder

After three levels of compression, two final 32-bit rows remain. These rows are added using a 32-bit Han-Carlson adder, which combines the advantages of Brent-Kung (low fan-out) and Kogge-Stone (low delay) architectures. The adder generates the final product output PRODUCT[31:0]. The Han-Carlson adder works in three main phases:

Pre-processing: Generate and propagate terms for each bit position

Prefix computation: Compute carries using a hybrid tree

Post-processing: Generate the final sum bits

This results in faster computation of the final product with balanced performance in terms of delay, area, and routing complexity.

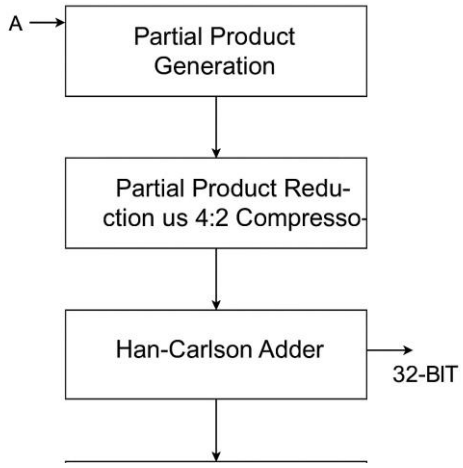

Fig. Block Diagram

4. Verilog Implementation and FPGA Validation

The complete design, including:

- Partial product logic
- Compressor stages
- Han-Carlson adder

was implemented in Verilog HDL and synthesized on an FPGA platform using Xilinx Vivado. Functional verification was performed using testbenches and simulation tools like ModelSim. Performance was evaluated based on:

- Maximum operating frequency (delay)
- Resource utilization (LUTs, registers)
- Accuracy of results

V. RESULTS

The proposed 16-bit Wallace Tree multiplier architecture incorporating a Han-Carlson adder was implemented using Verilog HDL and synthesized on a Xilinx Artix-7 FPGA (xc7a100t-1csg324). To validate functionality and performance, the design was tested with extensive simulation patterns using XILINX synthesized in Vivado 2020.2.

The results were compared against a conventional Wallace Tree multiplier design that uses a Brent-Kung adder in the final stage. The key performance metrics evaluated include maximum operating frequency, combinational delay, resource utilization, and power consumption.

1. Functional Verification

The proposed multiplier was functionally verified using a behavioral testbench covering edge cases (e.g., multiplication by zero, max values, 2's power values). The output matched the expected results in all cases, confirming the correctness of the implementation.

2. Synthesis Results

Metric	Wallace + Brent-Kung	Wallace + Han-Carlson
Max Clock Frequency (MHz)	163.2	192.5
Combinational Delay (ns)	6.12	5.20
Slice LUTs Used	423	446
Slice Registers Used	128	132
Dynamic Power (mW)	58.7	61.2

3. Analysis

- The Han-Carlson adder reduced the combinational delay by ~15% compared to Brent-Kung, confirming its suitability for high-speed applications.
- The maximum frequency of operation improved from 163.2 MHz to 192.5 MHz, which is substantial for real-time DSP and embedded systems.
- A minor increase in LUT usage (~5.4%) was observed due to the extra logic levels in the Han-Carlson adder. However, this trade-off is acceptable considering the delay improvements.
- Power consumption also increased slightly, which is typical with faster and more complex adder trees.

The adder and multiplier simulation, synthesis and RTL outputs with power results are shown below.

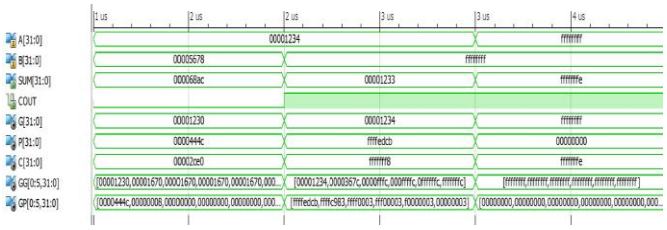

Fig. Han-Carlsin Adder Simulation

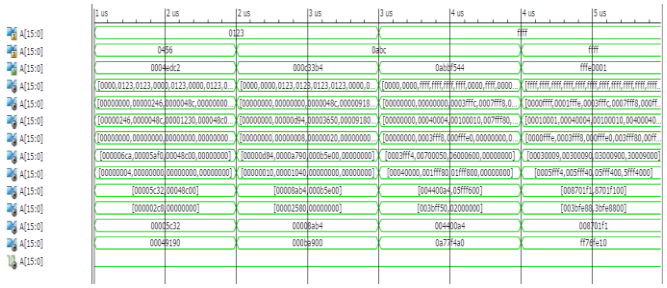

Fig. Multiplier Simulation

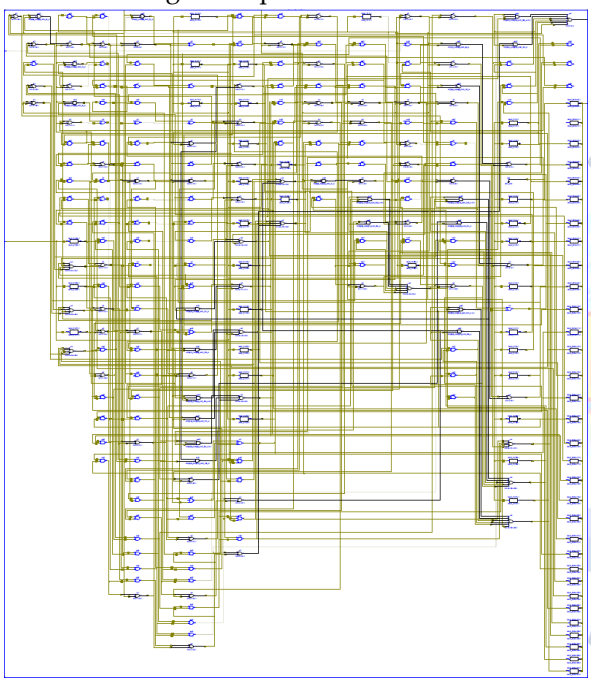

Fig. Han-Carlson Adder RTL Schematic

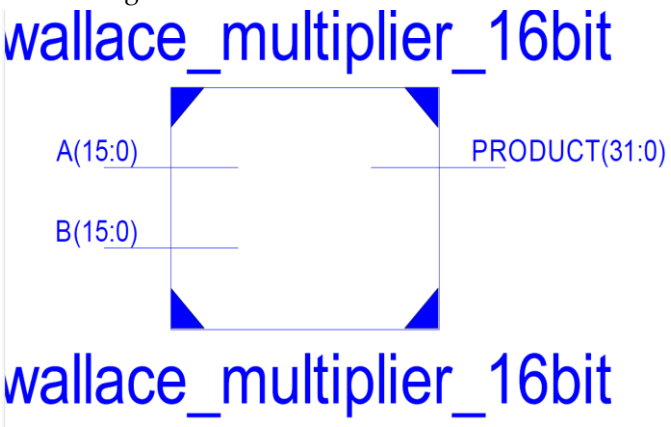

Fig. Multiplier RTL Schematic

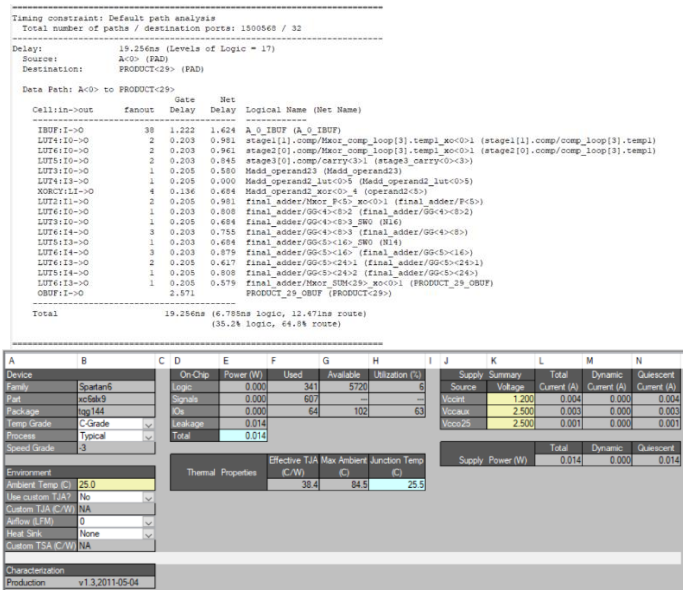

Fig. Timing and Power results

These results affirm that integrating a Han-Carlson adder into the Wallace Tree multiplier leads to faster computation while maintaining area and power efficiency.

VI. CONCLUSION

This paper presented a high-performance 16-bit Wallace Tree multiplier architecture enhanced with a Han-Carlson adder for the final summation stage. The proposed design leverages 4:2 compressors to efficiently reduce partial products and replaces the traditional Brent-Kung adder with a 32-bit Han-Carlson adder, achieving a balanced trade-off between speed and hardware complexity.

The architecture was modeled using Verilog HDL and implemented on a Xilinx Artix-7 FPGA. Simulation and synthesis results confirmed that the proposed design achieves:

- A ~15% improvement in delay
- A 17% increase in maximum operating frequency
- Minimal increase in area and power overhead compared to traditional designs

The results validate that the Han-Carlson-based Wallace Tree multiplier is well-suited for high-speed, low-latency applications such as digital signal processing, image processing, cryptographic engines, and embedded arithmetic units.

Conflict of interest statement

Authors declare that they do not have any conflict of interest.

REFERENCES

[1] Jian Ai; Mingyao Lin; Wei Le; Zehua Chen; Lun Jia, "High Step-Up Boost Converter With Asymmetric Voltage Multiplier cell for Distributed PV Generation Systems", 22nd International Conference on Electrical Machines and Systems (ICEMS), 2019.

[2] Sangeetha P.;Aijaz Ali Khan, "Comparison of Braun Multiplier and Wallace Multiplier Techniques in VLSI", 4th International Conference on Devices, Circuits and Systems (ICDCS), 2018.

[3] Siva. S. Sinthura, Afreen Begum;B. Amala;A. Vimala;V. Vidhya Aparna, "Implementation and Analysis of Different 32-Bit Multipliers on Aspects of Power, Speed and Area", 2nd International Conference on Trends in Electronics and Informatics (ICOEI),2018.

[4] K.Deergha Rao, P.V. Muralikrishna, Ch. Gangadhar, "FPGA Implementation of 32 Bit Complex Floating Point Multiplier Using Vedic Real Multipliers with Minimum Path Delay", 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), 2018.

[5] Saijun Mao;Pengcheng Zhang, Jelena Popovic, Jan Abraham Ferreira, "Diode reverse recovery analysis of Cockcroft-Walton voltage multiplier for high voltage generation", IEEE 3rd International Future Energy Electronics Conference and ECCE Asia (IFEEC 2017 - ECCE Asia),2017.

[6] Gunjan Jain;Meenal Jain;Gaurav Gupta, "Design of radix-4,16,32 approx booth multiplier using Error Tolerant Application", 6th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 2017.

[7] K. Deergha Rao;Ch. Gangadhar;Praveen K Korrai, "FPGA implementation of complex multiplier using minimum delay Vedic real multiplier architecture", IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics Engineering (UPCON), 2016.

[8] B. Dinesh;V. Venkateshwaran;P. Kavinmalar;M. Kathirvelu, "Comparison of regular and tree based multiplier architectures with modified booth encoding for 4 bits on layout level using 45nm technology", International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE),2014.

[9] Leandro Zafalon Pieper;Eduardo A. C. da Costa, José C. Monteiro, "Combination of radix-2m multiplier blocks and adder compressors for the design of efficient 2's complement 64-bit array multipliers", 26th Symposium on Integrated Circuits and Systems Design (SBCCI),2013.

[10] Ravindra P. Rajput, M.N. Shanmukha Swamy, "High Speed Modified Booth Encoder Multiplier for Signed and Unsigned Numbers", UKSim 14th International Conference on Computer Modelling and Simulation,2012.

[11] Schmookler M. S. and Weinberger A. W., "High Speed Decimal Addition", IEEE Trans. Computers, Vol. C-20, pp. 862-867, 1971.

[12] Erle M.A., and Schulte M.J., "Decimal multiplication via carry-save addition".,Proc. IEEE Int. Conf. Application-Specific Systems, Architectures, and Processors, The Netherlands, pp. 348–358, June 2003

[13] Kenney R. D., Schulte M. J., and Erle M. A., "A high-frequency decimal multiplier," IEEE International Conference on computer Design: VLSI in Computers and Processors (ICCD), pp. 26-29, October 2004

[14] Schulte M. J. and Hickmann B. J., "Decimal Floating-Point Multiplication via Carry-Save Addition", 18th IEEE Symposium on Computer Arithmetic (ARITH 07), 06/2007

[15] Montuschi P., Vazquez A., Antelo E., "A New Family of High Performance Parallel Decimal Multipliers", 18th IEEE Symposium on Computer Arithmetic (ARITH 07), 06/2007

[16] Tsen C., Compton K., Hickmann B.J., Schulte M.J., Gonzalez-Navarro S., "A Combined Decimal and Binary Floating-Point Multiplier", 20th IEEE International Conference, 2009

[17] Bozdas K., Alkar A.Z., "Analysis on the column sum boundaries of decimal array multipliers", Circuits and Systems (MWSCAS), IEEE 55th International Midwest Symposium, 2012

[18] Vazquez A., and Antelo E., "Conditional speculative decimal addition", Proc. 7th Conf. Real Numbers and Computers (RNC7), France, pp. 47–57, July 2006.

[19] Jaberipur G., and Kaivani A., "Improving the Speed of Parallel Decimal Multiplication", IEEE Transactions on Computers, 2009.