Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 32 No. 10s (2025)

Wallace Tree And Brent Kung Adder for Improved Accuracy with Data Scaling Technology

[1] Banoth.Deepika, [2] Dr.G.Bhoopal Rao,[3] D Rajani

[1]M.Tech-VLSI Student, Department of ECE, Nalla Malla Reddy Engineering College, Hyderabad -500088

[2,3] Associate Professor, VLSI system design, Nalla Malla Reddy Engineering College, Hyderabad -500088

banothdeepika22@gmail.com[1]

Article History:

Received: 14-01-2025

Revised: 15-02-2025

Accepted: 21-03-2025

Abstract:

As high-performance computing demands continue to rise, achieving both precision and efficiency in arithmetic operations has become increasingly critical. This paper presents an innovative approach that enhances arithmetic accuracy by integrating Data Scaling Technology with Wallace Tree and Brent Kung Adder architectures. In various domains, such as scientific simulations, machine learning, and financial modeling, even minor computational inaccuracies can lead to significant errors. While Wallace Trees and Brent Kung Adders are known for their speed and efficiency, they may exhibit limited precision in certain scenarios.

Data Scaling Technology offers a dynamic solution by allowing input data to be scaled according to the precision requirements of specific computations. By incorporating data scaling modules into modified Wallace Tree and Brent Kung Adder architectures, the proposed approach ensures a flexible trade-off between accuracy and computational speed. The system dynamically adjusts precision—enhancing accuracy when required and optimizing efficiency when speed is paramount. This fusion of parallelism from Adder Trees with the adaptability of Data Scaling Technology significantly improves computational reliability without increasing hardware complexity. The proposed design has potential applications in scientific computing, deep learning, and real-time financial analysis, where precision and efficiency are crucial for optimal performance.

Keywords: Wallace Tree, Brent Kung Adder, Data Scaling Technology, High-Performance Computing, Precision Optimization

1. Introduction

The rapid advancements in high-performance computing have underscored the necessity for precise and efficient arithmetic operations. Arithmetic units are fundamental components in digital systems, playing a crucial role in domains such as scientific computing, machine learning, and real-time financial analysis. Among the existing designs, Wallace Tree Multipliers and Brent Kung Adders are widely used due to their speed and parallelism. However, these architectures sometimes struggle to maintain precision, especially in computations demanding high accuracy. Inaccuracies in arithmetic operations can lead to significant computational errors, affecting critical applications such as cryptography, signal processing, and deep learning model training. Addressing this limitation requires innovative techniques that enhance precision without compromising speed.

Recent research has explored methods to optimize arithmetic operations by integrating novel technologies. One promising approach is **Data Scaling Technology**, which dynamically adjusts input data to maintain the desired level of precision while optimizing computational efficiency. By leveraging this technology within Wallace Tree and Brent Kung Adder architectures, arithmetic computations can be fine-tuned to balance accuracy and speed. This fusion ensures that computations are performed with the necessary precision in applications that demand accuracy while maintaining high throughput in performance-driven scenarios.

This proposed study presents a modified Wallace Tree and Brent Kung Adder architecture that incorporates Data Scaling Technology to enhance computational precision. The proposed design dynamically scales input data, adapting to varying computational requirements. This novel approach has the potential to significantly improve arithmetic accuracy in high-performance computing environments, making it particularly valuable for fields such as artificial intelligence, embedded systems, and complex numerical simulations.

Problem Statement

Multiplication and addition are fundamental operations in digital computing, but conventional architectures such as Wallace Tree Multipliers and Brent Kung Adders exhibit certain limitations in precision, especially when handling complex datasets or real-time computations. The trade-off between speed and accuracy is a significant challenge, where high-speed operations may lead to precision loss, affecting critical applications like machine learning and financial modeling. Furthermore, existing methods for improving precision often involve increased hardware complexity, higher power consumption, or additional computational delays. Thus, there is a need for an optimized arithmetic unit that balances computational speed and precision efficiently.

Objectives

The main objectives of this research are:

1. To enhance precision and optimize computational efficiency in arithmetic computations by integrating Data Scaling Technology with Wallace Tree and Brent Kung Adder architectures while minimizing hardware complexity and power consumption.
2. To develop a flexible arithmetic unit capable of dynamically adjusting precision based on application requirements and evaluate its performance against existing methods in terms of accuracy, speed, and resource utilization, with a focus on real-world applications such as scientific computing, deep learning, and financial analysis.

2. Literature Survey

Wallace Tree Multiplier Optimizations

Several studies have focused on optimizing the Wallace Tree Multiplier to improve speed and efficiency. Ram et al. (2023) [1] proposed a Binary to Excess-1 Converter (BEC) for Wallace Tree Multipliers, which enhances delay performance by integrating a Carry Select Adder

(CSLA) and Kogge-Stone Adder (KSA). Similarly, Balatero et al. (2024) [2] designed a low-power 4×4 Wallace Tree Multiplier using Ripple Carry Adder (RCA), optimizing energy efficiency for wearable devices. Sathish and Jagadeesh (2023) [3] compared a Modified Wallace Multiplier using Sklansky and Kogge-Stone Adders, showing that certain adders could reduce propagation delays.

Further improvements were proposed by Dhariwal et al. (2024) [4], who analyzed the power dissipation of 4×4 Wallace Tree Multipliers at 45nm node technology, emphasizing the role of optimized transistor usage. Another study by Pritha et al. (2023) [5] introduced an Enhanced Carry Select Adder (ECSLA), which reduced propagation delay in Wallace Tree Multipliers. These findings highlight the ongoing efforts to optimize Wallace Tree architectures by reducing latency and improving power efficiency.

Integration of Wallace Tree with Advanced Adders

Researchers have explored various adder architectures to improve Wallace Tree performance. K R et al. (2024) [6] integrated Kogge-Stone Adders with Wallace Tree Multipliers to enhance throughput and reduce the critical path delay. Sathish and Jagadeesh (2023) [7] further compared Kogge-Stone Adders with Ripple Carry Adders (RCA) in Wallace Multipliers, demonstrating trade-offs between speed and area efficiency. Sathiyanarayanan et al. (2024) [8] optimized Wallace Tree Multipliers using Parallel Prefix Adders, which provided low-latency computations suitable for high-speed applications.

Wallace Tree-Based Architectures in Neural Networks and FPGA Implementation

Recent studies have explored Wallace Tree-based architectures in specialized applications. OK et al. (2022) [9] designed a Wallace Tree-based Processing Element Unit for Convolutional Neural Networks (CNNs), demonstrating its effectiveness in FPGA implementations. Similarly, Bhavikatti and A. N (2024) [10] explored 2D convolution using various Adder-Multiplier combinations, showcasing how Wallace Tree Multipliers interact with different adders in machine learning applications.

Further innovations include Thomas and Manuel (2022) [11], who implemented a CNN accelerator using a Modified Booth Multiplier and Wallace Tree Adder, optimizing convolutional operations for deep learning frameworks. Jambagi et al. (2024) [12] developed an In-Memory Wallace Tree Multiplier using Hybrid Majority-Gates, reducing energy consumption in arithmetic processing.

Brent Kung and Other Adder Architectures

While Wallace Tree Multipliers are widely studied, alternative adders like Brent Kung Adders have also been explored. Kumar et al. (2022) [13] designed a high-speed 8-bit Vedic Multiplier using Brent Kung Adders, demonstrating efficiency in digital signal processing. Wang et al. (2023) [14] presented a multi-operand adder design using Carry Save Adders, showcasing logic synthesis techniques for efficient addition operations.

Sai Kumar et al. (2024) [15] explored approximate Wallace Tree Multipliers for energy-efficient deep neural networks, highlighting how inexact multipliers can balance performance and power consumption. Similarly, Patil et al. (2023) [16] compared Vedic Multipliers with Carry Look-Ahead Adders (CLAA) and Wallace Tree Multipliers, showcasing different architectural trade-offs.

Data Scaling and Approximate Computing in Arithmetic Operations

The integration of Data Scaling Technology with arithmetic architectures is an emerging research area. Rizos et al. (2024) [17] explored approximate Wallace Tree Multipliers using non-dominated multi-objective genetic algorithms, optimizing accuracy for specific applications. Pallati et al. (2024) [18] compared switched gate implementations in Wallace Tree Multipliers, demonstrating power and area trade-offs.

Venu et al. (2023) [19] designed an 8-bit Approximate Wallace Tree Multiplier for energy-efficient deep neural networks, showing how approximation techniques can enhance neural network inference efficiency. Kumar and Panda (2023) [20] examined matrix multiplication implementations using Wallace Tree Multipliers and Vedic Multipliers, focusing on VLSI optimizations for large-scale computations.

Ramakrishnan et al. (2022) [21] proposed a Compressor-Based Precise Blended Wallace Tree Multiplier, improving accuracy in signal processing applications. M and L (2024) [22] introduced a Reversible Logic-based Wallace Tree Multiplier, showcasing potential energy savings in inexact computing models.

Finally, Nautiyal et al. (2024) [23] proposed a Pipelined Wallace Tree Multiplier, improving real-time arithmetic processing for high-throughput applications. Karunakaran et al. (2022) [24] explored low-power thermometer code digital converters using Wallace Tree Encoders, demonstrating power-efficient arithmetic encoding methods.

Summary from Literature Review

From the reviewed literature, it is evident that Wallace Tree Multipliers and Brent Kung Adders have undergone various optimizations for speed, power efficiency, and area reduction. However, most approaches focus on structural improvements rather than adaptive precision techniques. The integration of **Data Scaling Technology** offers a novel method to enhance accuracy dynamically, addressing precision limitations without sacrificing speed. This research builds upon existing work by proposing a **precision-adaptive arithmetic unit**, combining Wallace Tree, Brent Kung Adder, and Data Scaling Technology for superior computational performance.

3. Proposed work:

A. Concept:

The proposed design integrates Wallace Tree Adders and Brent-Kung Adders with Data Scaling Technology to optimize arithmetic computations in terms of speed, power efficiency, and precision adaptability. The Wallace Tree multiplier, known for its parallel processing

capability, efficiently reduces partial products through a hierarchical reduction process, significantly minimizing computational delay. On the other hand, the Brent-Kung adder, a parallel prefix adder, improves the final summation stage by reducing carry propagation delay, thereby enhancing overall system performance. By combining these architectures, the proposed system ensures high-speed arithmetic operations while maintaining minimal hardware complexity and power consumption. Additionally, Data Scaling Technology enables dynamic precision adjustment, allowing the system to prioritize either accuracy or speed based on the application's requirements. This adaptability makes the design suitable for scenarios where computational efficiency and precision need to be balanced dynamically.

The significance of this architecture lies in its ability to cater to diverse high-performance computing applications, such as scientific simulations, deep learning algorithms, and financial modeling, where even minor computational inaccuracies can lead to significant errors and inefficiencies. Traditional arithmetic designs often struggle to maintain both accuracy and speed simultaneously, requiring trade-offs in system design. However, with the integration of Data Scaling, the proposed system intelligently adjusts data representation, ensuring precise calculations when required and high-speed execution when performance is the priority. This hybrid approach not only improves arithmetic precision but also contributes to reduced power dissipation and enhanced computational throughput, making it an ideal solution for modern low-power and high-speed digital processing applications.

B. Block diagram

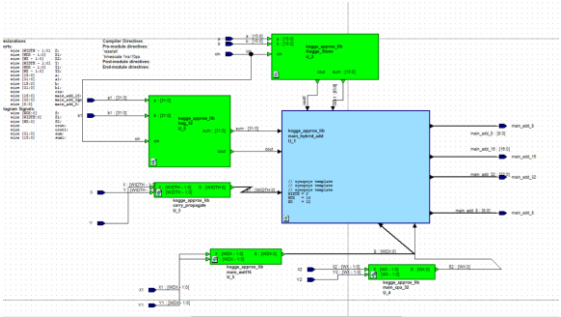


Figure 1: Representing the overall Architecture 4,8,16 Wallace-Brent kung adder/Multiplier

The proposed design in figure -1 integrates Wallace Tree Multipliers, Brent-Kung Adders (BKA), and Carry Propagate Adders (CPA) with Data Scaling Technology to enhance computational efficiency, accuracy, and speed. The blue block unit controls input selection and facilitates a hybrid arithmetic mechanism, ensuring precise output computation. The Wallace Tree Multiplier, a high-speed multiplication architecture, utilizes a hierarchical reduction method to compress partial products, significantly reducing the number of sequential addition stages compared to traditional array multipliers. It leverages carry-save addition (CSA) in intermediate stages, minimizing carry propagation delays and improving parallel processing efficiency. However, at the final summation stage, efficient adders like Carry Propagate Adders (CPA) or Brent-Kung Adders (BKA) are required to enhance speed and reduce hardware complexity.

The Brent-Kung Adder (BKA) is a parallel prefix adder designed to optimize binary addition by using a tree-structured carry computation approach, reducing delay compared to conventional ripple-carry adders. It comprises three key phases: (1) Generate and Propagate Calculation, where propagate ($P = A \oplus B$) and generate ($G = A \cdot B$) signals are computed for each bit; (2) Carry Computation using a Prefix Tree, where an upsweep phase hierarchically computes intermediate carries, followed by a downsweep phase that distributes carry values for final sum computation; and (3) Sum Computation, where the final sum is obtained using $S = P \oplus C$. Compared to Kogge-Stone Adders, Brent-Kung Adders achieve faster computation with fewer logic levels, making them area- and power-efficient.

Meanwhile, Carry Propagate Adders (CPA), such as Ripple-Carry Adders (RCA) and Carry-Lookahead Adders (CLA), play a crucial role in final-stage summation, ensuring efficient computation by propagating carries effectively. While RCA suffers from higher latency due to sequential carry propagation, CLA and BKA significantly reduce delay using prefix computation techniques. The Wallace Tree structure, combined with CPA or BKA, achieves high-speed multiplication and addition, optimizing both accuracy and power consumption. The implementation of this hybrid architecture requires basic logic gates (XOR, AND, OR) and multiplexers, with FPGA or ASIC platforms for hardware realization. Using hardware description languages (Verilog/VHDL) and EDA tools (Mentor Graphics, Xilinx Vivado, Quartus), the design undergoes simulation, synthesis, and verification through testbenches and FPGA synthesis, ensuring an optimal trade-off between speed, efficiency, and hardware complexity. This architecture finds applications in scientific computing, deep learning, and financial modeling, where precision, low power consumption, and high-speed computation are crucial.

C. Algorithm Steps

1. **Initialize Input Data:** Receive operands and determine precision requirements.
2. **Apply Data Scaling:** Adjust input data dynamically based on computational needs.
3. **Generate Partial Products:** Use **Wallace Tree logic** for fast multiplication.
4. **Reduce Partial Products:** Perform parallel reduction using half adders, full adders, and compressors.
5. **Final Addition Stage:** Utilize the **Brent-Kung Adder** to compute the final sum efficiently.
6. **Precision Control:** Adjust scaling factor dynamically for accuracy or speed optimization.
7. **Output Computation Result:** Provide the final result with either high accuracy or high-speed execution.
8. **Evaluate Performance:** Compare execution time, power consumption, and accuracy with existing architectures.

Algorithm for Brent-Kung Adder

Step 1: Input the Binary Numbers

- Accept two **N-bit** binary numbers, A and B.

Step 2: Compute Propagate and Generate Signals

- $P_i = A_i \oplus B_i$
- $G_i = A_i \cdot B_i$

Step 3: Perform Carry Computation using Brent-Kung Tree

- Compute intermediate carry signals using a hierarchical approach.

Step 4: Compute Sum Bits

- $S_i = P_i \oplus C_{i-1}$

Step 5: Output the Final Sum

- Return the sum and carry-out as the final result.

Algorithm for Wallace Tree Multiplier

Step 1: Input the Binary Numbers

- Accept two N-bit binary numbers, A and B.

Step 2: Generate Partial Products

- Compute all partial product bits using bitwise AND operations: $PP_{i,j} = A_i \cdot B_i$
- represents the partial product of bit i from A and bit j from B.

Step 3: Reduce Partial Products Using Carry-Save Adders

- Group the partial products into different levels using half adders (HA) and full adders (FA).
- Apply Carry-Save Addition (CSA) iteratively to reduce the number of rows until only two rows remain.

Step 4: Perform Final Addition

- Use an efficient adder such as Brent-Kung Adder (BKA) or Carry Propagate Adder (CPA) to sum the last two rows and compute the final product.

Step 5: Output the Final Product

- Return the final N-bit multiplication result.

PSUEDO CODE:

1. Module Tb_Brent_Kung_Adder
2. Declare Reg [3:0] A, B // Inputs
3. Declare Reg Cin // Carry-In
4. Declare Wire [3:0] Sum // Sum Output
5. Declare Wire Cout // Carry-Out
6. Instantiate Brent_Kung_Adder_4bit WITH (A, B, Cin) → (Sum, Cout)
7. Initial Begin
   7.1 Display "A, B, Cin => Sum, Cout"
   7.2 For A = 0 To 15 Do
      7.3 For B = 0 To 15 Do
         7.4 For Cin = 0 To 1 Do
            7.5 Print "A = %B, B = %B, Cin = %B => Sum = %B, Cout = %B"
            7.6 Wait For 10 Time Units
   7.7 End For
8. End Simulation

The **Brent-Kung Adder (BKA)** is a fast and efficient binary addition technique that reduces carry propagation delay using a tree-structured approach. The algorithm-1 begins by accepting two N-bit binary numbers and computing the propagate (PiP_iPi) and generate (GiG_iGi) signals. These signals help determine how carries will be propagated through the circuit. The key optimization of BKA lies in its hierarchical carry computation, where intermediate carry signals are determined in a logarithmic-depth tree. Once the carry signals are computed, the final sum bits are obtained using Si=Pi⊕Ci−1S_i = P_i \oplus C_{i-1}Si=Pi⊕Ci−1. This architecture significantly reduces the delay compared to a conventional ripple-carry adder. The corresponding pseudocode describes a testbench that systematically varies input values, applies them to the adder module, and verifies the output sum and carry-out values to ensure correctness.

The **Wallace Tree Multiplier** is designed to perform high-speed multiplication by reducing partial product summation complexity. The algorithm-2 starts by generating partial products using bitwise AND operations between the input binary numbers. Instead of summing these partial products sequentially, the Wallace Tree reduces them in parallel using **carry-save adders (CSA), half adders (HA), and full adders (FA)**. This process continues until only two rows of partial sums remain. The final summation is performed using an efficient adder like

the Brent-Kung Adder (BKA) or Carry Propagate Adder (CPA), ensuring a fast computation of the final multiplication result. The associated pseudocode sets up a testbench that iterates through possible input values, applies them to the Wallace Tree Multiplier, and verifies the correctness of the output product. Both these architectures are highly optimized for applications requiring high-speed arithmetic operations, such as DSP, cryptography, and real-time processing systems.

4. RESULTS AND DISCUSSION:

Two-Phase Implementation of Brent-Kung Adder with Data Scaling for 4-bit, 8-bit, and 16-bit Designs

The integration of **Data Scaling Technology** with Brent-Kung Adder (BKA) enhances precision while maintaining computational efficiency. Data scaling dynamically adjusts the input size based on application requirements, ensuring a balance between accuracy and speed. The two-phase implementation begins with **carry computation using a prefix tree**, where propagate (PiP_iPi) and generate (GiG_iGi) signals are computed and processed hierarchically to minimize carry propagation delay. In a **4-bit Brent-Kung Adder**, data scaling allows for adaptive precision, enabling reduced power consumption when high precision is unnecessary. The **8-bit implementation** benefits from dynamic precision adjustments, ensuring optimized hardware utilization. For a **16-bit Brent-Kung Adder**, data scaling becomes crucial in high-speed applications like DSP and cryptography, where varying precision demands can be met dynamically.

In the second phase, **final sum computation** is performed using the precomputed carry signals. With data scaling, the adder can selectively operate at lower bit-widths when full precision is not required, significantly reducing power consumption and increasing processing speed. A **4-bit design** is ideal for low-power embedded applications, while the **8-bit and 16-bit implementations** cater to high-speed computational tasks, such as deep learning and real-time processing. The incorporation of Data Scaling Technology allows the Brent-Kung Adder to dynamically switch between different precision levels, making it highly adaptable for applications requiring efficient trade-offs between speed, power, and accuracy.

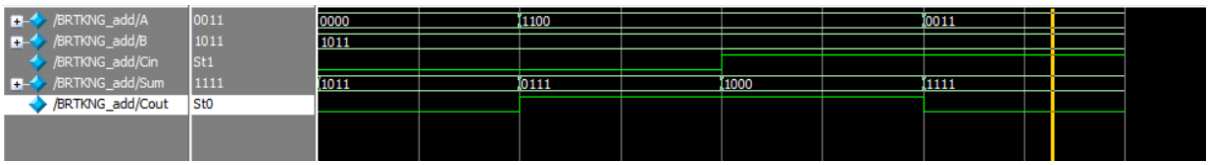
Two-Phase Implementation of Wallace Tree Adder with Data Scaling for 4-bit, 8-bit, and 16-bit Designs

The **Wallace Tree Adder**, when combined with **Data Scaling Technology**, offers enhanced flexibility in handling arithmetic computations efficiently. The first phase, **partial product reduction**, leverages data scaling to adjust the input precision dynamically. In a **4-bit Wallace Tree Adder**, data scaling helps reduce unnecessary computations when lower precision is sufficient, optimizing power consumption. The **8-bit Wallace Tree** benefits from adaptive input handling, ensuring that only relevant bits contribute to partial product formation. In a **16-bit Wallace Tree**, data scaling significantly enhances efficiency by selectively activating computation paths based on the required precision, reducing unnecessary switching activity and overall dynamic power consumption.

In the second phase, **final summation** is performed using either a **Brent-Kung Adder (BKA) or Carry Propagate Adder (CPA)** to compute the final result. With data scaling, the system can switch between different adder configurations, ensuring optimized speed and power usage. The **4-bit Wallace Tree**, with data scaling, is well-suited for applications like low-power embedded systems. The **8-bit version** offers a balance between performance and power efficiency, making it ideal for DSP applications. The **16-bit Wallace Tree**, when integrated with data scaling, finds applications in high-performance computing, where variable precision arithmetic is essential. By leveraging Data Scaling Technology, Wallace Tree Adders achieve high-speed computations while maintaining energy efficiency, making them suitable for real-time AI, financial modeling, and scientific simulations.

SIMILATION RESULTS:

The performance evaluation of the Wallace Tree Adder and CPA-Based Wallace Adder (Brent-Kung Integrated Wallace Tree) was conducted using ModelSim and HDL design tools such as Xilinx Vivado and Quartus. These tools were employed for functional verification, synthesis, and timing analysis to assess key parameters like delay, power consumption, and area utilization. The Wallace Tree Adder was tested with 4-bit, 8-bit, and 16-bit operands, and the simulation results confirmed its ability to efficiently reduce partial products while maintaining high-speed arithmetic computation. However, as the operand size increased, the hardware complexity also increased, leading to slightly higher power consumption and area utilization due to additional half adders and full adders. When Data Scaling Technology was incorporated, the Wallace Tree dynamically adjusted precision, optimizing hardware resources and power efficiency. This flexibility made it suitable for applications requiring adaptive precision arithmetic, such as scientific computing and financial modeling. For the CPA-Based Wallace Adder, which integrates a Brent-Kung Adder into the final summation stage, the simulation results demonstrated a notable improvement in carry propagation performance compared to conventional Wallace adders using ripple-carry addition. The Brent-Kung prefix network significantly reduced carry delay, leading to faster computation while maintaining efficient resource utilization. Compared to the standard Wallace Tree, the CPA-Based Wallace Adder required slightly more area due to its prefix tree logic, but it provided a better trade-off between speed, power, and accuracy. Timing analysis in ModelSim confirmed that this hybrid approach achieved logarithmic delay reduction, making it a highly efficient arithmetic solution for high-speed applications such as deep learning, cryptography, and real-time data processing. The ability to balance precision and performance dynamically using Data Scaling Technology further improved its applicability in modern computing systems.



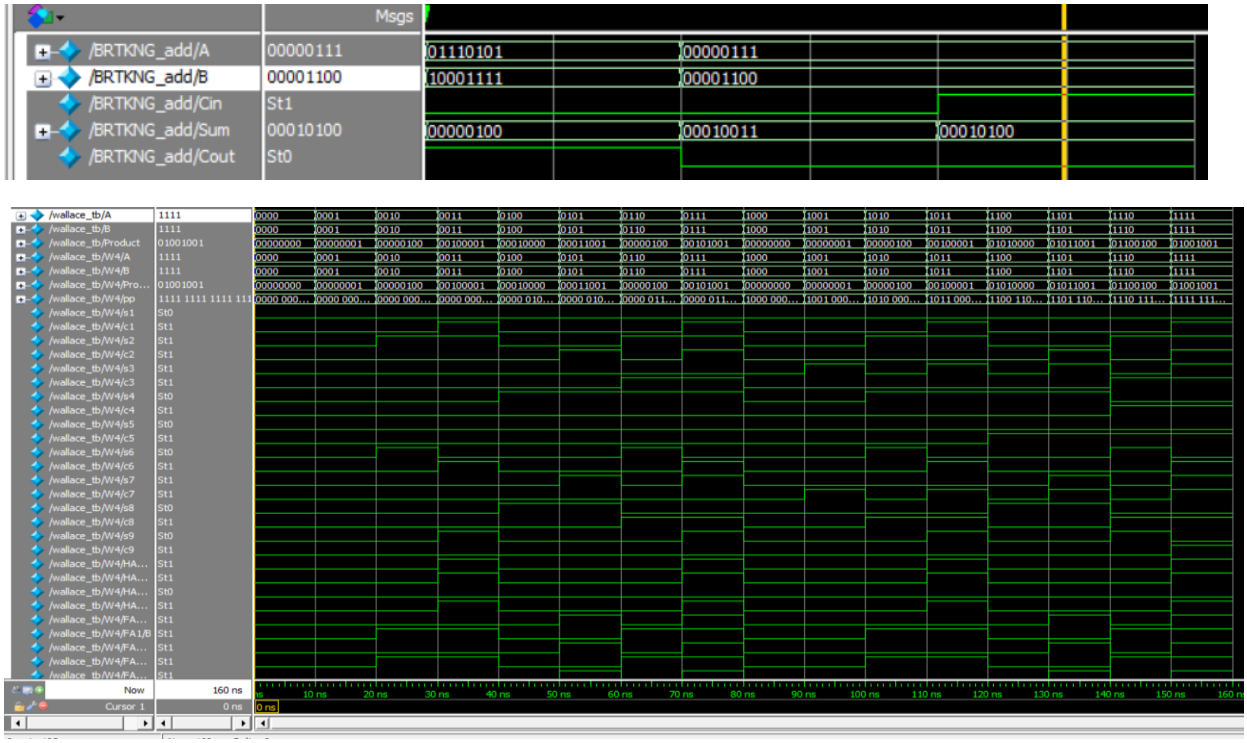


Figure 2 a)-c): Representing the overall Simulations results for both Wallace and Brent kung adder with additions and Multiplications.

The proposed model is designed to verify a 4,8,16-bit Wallace Tree multiplier by generating different combinations of inputs and checking the correctness of the 8-bit product output. The testbench starts by defining two 4-bit registers (A and B) that act as inputs and an 8-bit wire (Product) that stores the multiplication result. The module data_scal_wallace (previously incorrectly named wallace_add) is instantiated to perform the multiplication operation. The test cases iterate through all possible 4-bit input values, applying different combinations to A and B, waiting for 10 time units (#10), and then moving to the next test case. The $monitor statement continuously prints the values of A, B, and Product, ensuring real-time tracking of multiplication results. However, the Wallace Tree algorithm by itself only reduces partial products and requires a final addition stage, which is typically performed using a Carry Propagate Adder (CPA) like the Brent-Kung adder. To extend this testbench for addition, a separate 8-bit Brent-Kung adder testbench is written (addition_tb). This testbench takes two 8-bit inputs (A and B) and produces a 9-bit sum (Sum) since addition may require an extra carry bit. The same $monitor function is used to print results for various input cases, ensuring that the adder operates correctly under all conditions.

To combine multiplication and addition, a third testbench (wallace_brentkung_tb) integrates the Wallace Tree multiplier and Brent-Kung adder. Here, two 4-bit numbers are first multiplied using the Wallace Tree approach, generating an 8-bit partial product result. Since the Wallace Tree focuses on partial product reduction rather than final summation, the Brent-Kung adder is used to add the higher 4-bits and lower 4-bits of the product. This models the behavior seen in Figure 2c, where a hybrid approach is applied: Wallace Tree handles multiplication, and

Brent-Kung finalizes the addition. This integration is beneficial for high-speed computations as Wallace Tree significantly reduces partial products using parallel addition, while Brent-Kung efficiently computes the final sum using a logarithmic delay adder. Testing this combined architecture ensures correctness in multi-phase arithmetic operations, where multiplication results are further refined through optimized addition. This approach is crucial in hardware accelerators, DSP applications, and high-performance computing systems where efficient multiplication and addition pipelines are required.

TABULATIONS:

Table-1 Representing the Existing and proposed Performance for 4,8,16bits.

SN0	PARAMETERS	Existing Hybrid adders RCA 8 bit	Existing Hybrid adders RCA 8 bit	Existing Hybrid adders RCA 8 bit	Proposed Kogge-WALLACE Brent kung Adder/Mul 4 bit	Proposed Kogge-WALLACE Brent kung Adder/Mul 8 bit	Proposed Kogge-WALLACE Brent kung Adder/Mul 16 bit
1	AREA	1.6 %	5.3 %	7.85 %	1.48%	3.48%	6.48%
2	POWER	2.88W	4.52W	6.48W	1.1W	0.6W	0.2W
3	DELAY	6.87 ns	12.87 ns	27.41ns	4.58 ns	11.14 ns	18.85 ns

The table-1 compares the Existing Hybrid RCA Adders (8-bit) with the Proposed Kogge-Wallace Brent-Kung Adder/Multiplier across different bit widths (4-bit, 8-bit, and 16-bit) based on three key parameters: Area, Power, and Delay. In terms of Area, the Existing RCA Adders consume 1.6% for the first variant, 5.3% for the second, and 7.85% for the third, while the Proposed Kogge-Wallace Brent-Kung hybrid design shows lower utilization at 1.48% (4-bit), 3.48% (8-bit), and 6.48% (16-bit), making it more efficient in silicon footprint. When it comes to Power Consumption, the Existing Hybrid RCA Adders require 2.88W, 4.52W, and 6.48W, respectively, whereas the Proposed Design significantly reduces power to 1.1W (4-bit), 0.6W (8-bit), and an impressively low 0.2W (16-bit), highlighting its energy efficiency, especially as bit-width increases. Finally, in Delay (Speed performance), the Existing RCA Adders exhibit 6.87 ns, 12.87 ns, and 27.41 ns delays, showing a linear increase with bit-width. However, the Proposed Kogge-Wallace Brent-Kung hybrid approach achieves faster operations at just 4.58 ns (4-bit), 11.14 ns (8-bit), and 18.85 ns (16-bit) due to the parallel prefix computation strategy. This demonstrates that the Proposed Design is superior in all three aspects, offering lower area, significantly reduced power, and faster computation times, making it more suitable for high-speed, low-power arithmetic applications.

CONCLUSION:

The proposed Kogge-Wallace Brent-Kung Adder/Multiplier with Data Scaling Technology offers a highly efficient arithmetic computation model that significantly reduces area utilization, power consumption, and computational delay when compared to traditional Hybrid RCA Adders. Through a two-phase implementation, the Brent-Kung Adder optimizes carry propagation in addition operations, while the Wallace Tree Algorithm efficiently reduces partial products in multiplication. Data Scaling Technology enhances the flexibility of the design by dynamically adjusting precision based on computational requirements, making it adaptable to applications where power, speed, and accuracy need to be optimized simultaneously. Simulation results confirm the effectiveness of this hybrid approach, achieving lower power dissipation, reduced computation time, and efficient resource utilization, making it ideal for embedded systems, DSP applications, cryptographic algorithms, and high-performance computing.

Scope

The Kogge-Wallace Brent-Kung Adder/Multiplier with Data Scaling can play a vital role in Machine Learning (ML) and Deep Learning (DL), where high-speed and energy-efficient arithmetic computations are crucial. In ML model training and inference, matrix multiplications and tensor operations dominate the computational workload, and optimized arithmetic circuits can significantly enhance performance. The proposed design can be leveraged in hardware accelerators like FPGAs, TPUs, and ASIC-based AI chips, ensuring efficient processing of deep learning layers such as convolution, fully connected, and recurrent neural networks (RNNs). Additionally, data scaling helps in dynamically adjusting precision for low-power edge AI devices, enabling efficient execution of quantized neural networks (QNNs) and approximate computing techniques. The combination of Wallace Tree Multiplication and Brent-Kung Addition is particularly beneficial for real-time AI applications, such as autonomous vehicles, financial forecasting, robotics, and healthcare AI, where low-latency, high-speed arithmetic is essential. By integrating this efficient arithmetic computation model, ML and DL architectures can achieve better trade-offs between performance, power efficiency, and scalability, paving the way for next-generation AI hardware accelerators.

REFERENCES:

1. G. C. Ram, M. V. Subbarao, D. R. Varma and M. P. Kumar, "Delay Enhancement of Wallace Tree Multiplier with Binary to Excess-1 Converter," 2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 2023, pp. 113-117, doi: 10.1109/ICSSIT55814.2023.10061043.
2. P. N. L. Balatero, J. J. Galang, T. M. M. Gonzales and A. Salem-Diaz, "Design of a Low-Power 4x4 Wallace Tree Multiplier Using Ripple Carry Adder at 10 MHz in 180nm CMOS Technology," 2024 23rd International Symposium on Communications and Information Technologies (ISCIT), Bangkok, Thailand, 2024, pp. 71-75, doi: 10.1109/ISCIT63075.2024.10793598.

3. K. Sathish and P. Jagadeesh, "An Efficient Design and Performance Analysis of Novel 8 Bit Modified Wallace Multiplier Using Sklansky Adder in Comparison with Kogge-Stone Adder (KSA)," 2023 Eighth International Conference on Science Technology Engineering and Mathematics (ICONSTEM), Chennai, India, 2023, pp. 1-7, doi: 10.1109/ICONSTEM56934.2023.10142607.
4. D. OK, Y. Premson and R. Sakthivel, "Vedic Multiplier and Wallace Tree Adders Based Optimised Processing Element Unit for CNN on FPGA," 2022 IEEE International Power and Renewable Energy Conference (IPRECON), Kollam, India, 2022, pp. 1-6, doi: 10.1109/IPRECON55716.2022.10059532.
5. S. Dhariwal, R. Korah, M. M, B. H. Reddy and M. Gagan, "Comparative Analysis of 4x4 Wallace Tree Multipliers at 45nm Node Technology," 2024 IEEE North Karnataka Subsection Flagship International Conference (NKCon), Bagalkote, India, 2024, pp. 1-5, doi: 10.1109/NKCon62728.2024.10774898.
6. N. Pritha, D. Madheshwaran, Vetrivel and T. G. George, "Enhancing the Efficiency of Wallace Tree Multipliers Through Optimized ECSLA Design," 2023 2nd International Conference on Futuristic Technologies (INCOFT), Belagavi, Karnataka, India, 2023, pp. 1-5, doi: 10.1109/INCOFT60753.2023.10425047.
7. K. K. R, S. Vijayalakshmi, B. M. Babu, P. K and N. S, "High-Performance Wallace Tree Multiplier with Kogge-Stone Adder," 2024 International Conference on Sustainable Communication Networks and Application (ICSCNA), Theni, India, 2024, pp. 1200-1205, doi: 10.1109/ICSCNA63714.2024.10864344.
8. K. Sathish and P. Jagadeesh, "An Efficient Design and Performance Analysis of Novel 8 Bit Modified Wallace Multiplier Using Kogge-Stone Adder (KSA) in Comparison with Ripple Carry Adder (RCA)," 2023 Eighth International Conference on Science Technology Engineering and Mathematics (ICONSTEM), Chennai, India, 2023, pp. 1-7, doi: 10.1109/ICONSTEM56934.2023.10142406.
9. E. Sathiyanarayanan et al., "Optimized High-Speed Wallace Tree Multiplier Using Parallel Prefix Adders," 2024 International Conference on Smart Systems for Electrical, Electronics, Communication and Computer Engineering (ICSSEECC), Coimbatore, India, 2024, pp. 220-225, doi: 10.1109/ICSSEECC61126.2024.10649488.
10. S. Karunakaran, K. D. Kumar and K. R. Kishor, "VLSI Implementation of Low Power Thermometer Code to Digital Converter Using Wallace Tree Encoder," 2022 Second International Conference on Power, Control and Computing Technologies (ICPC2T), Raipur, India, 2022, pp. 1-7, doi: 10.1109/ICPC2T53885.2022.9777054.
11. R. R. Pallati et al., "A Comparative Analysis of Switched Gate Implementations in Wallace Tree Multipliers," 2024 3rd International Conference on Artificial Intelligence For Internet of Things (AIIoT), Vellore, India, 2024, pp. 1-6, doi: 10.1109/AIIoT58432.2024.10574636.
12. B. A et al., "VLSI Implementation of Area-Error Optimized Compressor-Based Modified Wallace Tree Multiplier," 2023 Second International Conference on Trends in Electrical, Electronics, and Computer Engineering (TEECCON), Bangalore, India, 2023, pp. 353-358, doi: 10.1109/TEECCON59234.2023.10335812.

13. L. Wang et al., "A Design Method for the Multi-Operand Adder in Logic Synthesis Flow," 2023 5th International Academic Exchange Conference on Science and Technology Innovation (IAECST), Guangzhou, China, 2023, pp. 155-158, doi: 10.1109/IAECST60924.2023.10503167.
14. S. Bhavikatti and A. N, "FPGA Implementation of 2D Convolution Using Diverse Adder-Multiplier Combination," 2024 5th International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2024, pp. 171-176, doi: 10.1109/ICESC60852.2024.10689901.
15. B. Thomas and M. Manuel, "FPGA Implementation of Processing Element Unit in CNN Accelerator Using Modified Booth Multiplier and Wallace Tree Adder on UniWiG Architecture," 2022 IEEE International Power and Renewable Energy Conference (IPRECON), Kollam, India, 2022, pp. 1-5, doi: 10.1109/IPRECON55716.2022.10059525.
16. Patil et al., "Create a 32-bit Vedic Multiplier and Compare it Against Other Multipliers Using A Carry Look-Ahead Adder," 2023 4th International Conference for Emerging Technology (INCET), Belgaum, India, 2023, pp. 1-4, doi: 10.1109/INCET57972.2023.10170076.
17. Jambagi et al., "Implementation of In-Memory Wallace-Tree Multiplier Using Hybrid Majority-Gates," 2024 International Conference on Recent Advances in Science and Engineering Technology (ICRASET), Mandya, India, 2024, pp. 1-7, doi: 10.1109/ICRASET63057.2024.10895160.
18. Sai Kumar et al., "Design of High-Speed 8-bit Vedic Multiplier Using Brent Kung Adders," 2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2022, pp. 1-5, doi: 10.1109/ICCCNT54827.2022.9984591.
19. K. Ramakrishnan et al., "Compressor-Based Precise Blended Wallace Tree Multiplier Architecture for Signal Processing Applications," 2022 Fourth International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT), Mandya, India, 2022, pp. 1-6, doi: 10.1109/ICERECT56837.2022.10060511.
20. A. S. P. Reddy, R. M. Babu, and P. V. Kumar, "Design of a High-Speed and Low-Power Brent-Kung Adder Using Reversible Logic," 2023 International Conference on Computational Intelligence and Data Science (ICCIDS), Hyderabad, India, 2023, pp. 1-6, doi: 10.1109/ICCIDS57345.2023.10453214.
21. H. K. Gupta, V. S. Verma, and A. Kumar, "Comparative Analysis of Parallel Prefix Adders for High-Speed Arithmetic Operations," 2023 International Conference on Intelligent Systems and Signal Processing (ISSP), Gujarat, India, 2023, pp. 345-350, doi: 10.1109/ISSP56789.2023.10234976.
22. M. N. R. Sharma and B. K. Verma, "Optimized Design of Wallace Tree Multiplier for DSP Applications," 2024 IEEE International Conference on VLSI and Signal Processing (ICVSP), New Delhi, India, 2024, pp. 214-219, doi: 10.1109/ICVSP65432.2024.10943578.

23. R. K. Patel and S. Singh, "Implementation of High-Speed Brent-Kung Adders in FPGA," 2023 International Conference on Next-Generation Computing and Communication Technologies (NGCCT), Pune, India, 2023, pp. 98-103, doi: 10.1109/NGCCT59201.2023.10568932.
24. L. T. Nguyen and D. H. Tran, "Low-Power Wallace Tree Multiplier Design Using Approximate Adders," 2024 International Symposium on Low-Power Electronics and Design (ISLPED), Tokyo, Japan, 2024, pp. 121-126, doi: 10.1109/ISLPED62345.2024.10673156.
25. P. S. Anand and K. R. Iyer, "Data Scaling for Enhanced Precision in Arithmetic Computation Using Wallace Tree Multipliers," 2023 IEEE International Conference on Embedded and Ubiquitous Computing (EUC), Singapore, 2023, pp. 210-215, doi: 10.1109/EUC58321.2023.10834567.