ITM Web of Conferences **79**, 01009 (2025) https://doi.org/10.1051/itmconf/20257901009
KEIS-2025

Design and Implementation of Braun Multiplier with Different Adder Architectures

Sneha More[1], Suhas Shirol[1]*, Aparna Bandiwad[2], and Nikita G M[2]

[1]Department of Electronics and Communication, KLE Technological University, Hubballi, India
e-mail: 01fe22bec250@kletech.ac.in,suhasshirol@kletech.ac.in
[2]Department of Electrical and Electronics, KLE Technological University, Hubballi, India
e-mail: 01fe22bee055@kletech.ac.in,01fe22bee069@kletech.ac.in

Abstract. This paper presents a high-performance, scalable Braun multiplier architecture implemented on FPGA, featuring advanced and hybrid adder topologies for enhanced power, area, and timing efficiency. The proposed system integrates Ripple Carry Adders (RCA), Brent-Kung Adders (BKA), Kogge-Stone Adders (KSA), and hybrid variants using structural Verilog, targeting both 4-bit and 32-bit designs. To address the limitations in conventional designs, this work introduces formal mathematical formulations for prefix logic, provides an algorithmic breakdown of carry propagation, and conducts a detailed power-delay-area (PDA) analysis. Implementations were carried out on a Xilinx Spartan-6 FPGA with post-place-and-route timing evaluation. Comparative analysis shows BKA yields the lowest LUT usage, while KSA achieves the minimum delay, making each suitable for low-power and high-speed applications respectively. Real-time signal monitoring using ChipScope ICON/VIO facilitates hardware-level debugging. The design offers a practical, energy-efficient, and modular solution for arithmetic-intensive applications in VLSI, embedded systems, and digital signal processing.

1 Introduction

Multipliers are essential parts of digital systems like image processing units, DSPs, embedded processors, and cryptographic engines. In VLSI and FPGA-based computing platforms, the design of quick and space-efficient multipliers has become crucial as technology advances toward greater performance and energy efficiency. Researchers are now looking into optimized multiplier architectures that balance speed, power, and silicon footprint due to the growing need for small, power-conscious, and real-time arithmetic units [1–3].

The Braun multiplier is especially well-suited for FPGA and ASIC implementations due to its combinational design and regular array structure. It's perfect for unsigned multiplications because it makes routing and scaling across bit-widths easier. However, the effectiveness of the adder structures used to sum partial products has a significant impact on how well it performs. Although straightforward, traditional ripple carry adders (RCAs) experience linear delay growth and lose efficiency as bit widths increase.

Mathematically, the Braun multiplier output is expressed as Equation (1).

$$P = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (A_i \cdot B_j) \cdot 2^{i+j} \tag{1}$$

where A_i and B_j are operand bits and P is the $2n$-bit product. The efficiency of this operation depends heavily on the adder logic used for summing partial products.

Recent research focuses on incorporating high-performance parallel prefix adders such as Brent-Kung (BKA) and Kogge-Stone (KSA) into multiplier designs to get around these restrictions. With better speed and area trade-offs, these adders provide logarithmic delay characteristics [1, 4]. The prefix adder logic is defined by generate and propagate signals as shown in Equations (2)–(4):

$$G_i = A_i \cdot B_i \tag{2}$$
$$P_i = A_i \oplus B_i \tag{3}$$
$$C_{i+1} = G_i + (P_i \cdot C_i) \tag{4}$$

Equations (2)–(4) enable parallel carry computation and help in reducing the critical path delay.

Area-efficient designs like BKA are suitable for low-power embedded and IoT systems, whereas KSA prioritizes speed for time-critical applications despite higher power usage. Some studies also explore hybrid adder integration and approximate logic to minimize switching activity, trading off slight accuracy loss for power savings [5].

The Full/HalfAdders (HA/FA), Brent-Kung Adder, and Kogge-Stone Adder are three distinct adder architectures that are integrated with Braun multipliers in this project's FPGA-based structural implementation. The design uses structural Verilog HDL modeling to support scal-

---

[*] Corresponding author: suhasshirol@kletech.ac.in

able bit-width configurations. Test benches and waveform simulations are used for functional validation, and the Xilinx Spartan-6 FPGA platform is used for synthesis and resource analysis. Additionally, post-place-and-route (P&R) timing analysis is conducted, and comparisons are extended to newer FPGA families such as Artix-7 and UltraScale+ to improve relevance. ChipScope ICON and VIO cores are used for real-time debugging but are treated as verification tools rather than novel contributions [6].

The key contributions of this work include: (i) formal modeling of Braun multiplier partial product and prefix adder logic, (ii) FPGA-based comparison of RCA, BKA, and KSA with Power–Delay Product (PDP) and Area–Delay Product (ADP) metrics, and (iii) synthesis and validation using Spartan-6 with relevance to modern FPGA families.

The remainder of this paper is organized as follows: Section II reviews related work, Section III details the methodology including mathematical formulations and pseudocode, Section IV presents results and comparative analysis, and Section V concludes with limitations and future directions such as hybrid adder integration and pipelined Braun multipliers.

The study provides important information for the design of application-specific hardware multipliers in VLSI and embedded systems by showing that each adder topology has a distinct impact on the performance of the Braun multiplier through power-delay area comparisons.

2 Literature Survey

Multipliers are critical in DSP and image processing systems where speed and area matter [3]. The Braun multiplier, though efficient, suffers from delay due to the Ripple Carry Adder (RCA). To address this, a Brent Kung adder was used instead of RCA, a logarithmic delay parallel prefix adder. Implemented and simulated using Microwind and Tanner EDA, the design achieved reduced delay and area. This makes it suitable for low-power, high-performance VLSI applications.

The study evaluates Braun multipliers on multiple FPGA boards, focusing on speed and efficiency [2]. Using ripple carry adders (RCA) in the design, they observed significant delay in the critical path. Results showed Virtex-5 outperformed Spartan-3E and Virtex-4 in terms of speed and resource utilization. However, the reliance on RCA limited further improvements, highlighting the need for advanced adder structures like Brent-Kung or Kogge-Stone to minimize delay and optimize area.

The study in [1] enhances Braun multiplier performance using low-power techniques like CMOS and GDI, replacing traditional RCAs with Kogge-Stone and Brent-Kung adders in the final stage to achieve logarithmic delay, improving speed and reducing area. Similarly, [7] explores Braun multipliers with 28-T, 20-T, and 14-T full adders in 180 nm technology to minimize power and transistor count, finding that while 28-T consumes more power, it ensures stable waveforms, whereas lower-transistor designs trade stability for reduced power—highlighting the challenge of achieving both low power and high reliability.

The paper focuses on scalable Braun multipliers implemented in structural Verilog with different full adders: 28-T, 20-T, and 14-T [7]. Simulations show that 20-T and 14-T adders consume less power but exhibit unstable waveforms, making them impractical. The 28-T adder, though power-hungry, offers stability and reliable outputs, leading to its selection. This highlights a trade-off between power, area, and stability, leaving scope for designs that achieve both efficiency and robustness.

The paper explores low-power Braun multipliers using 28-T and 20-T full adders, implemented in 180nm technology and simulated in Cadence Virtuoso [4]. Ripple carry and carry-save adders are employed to enhance speed and efficiency. Results show that 20-T adders consume less power but suffer from waveform distortion, whereas 28-T adders deliver cleaner outputs at slightly higher power. The study reveals the need for designs that combine low power with signal stability.

This paper proposes a low-power Braun multiplier using three hybrid full adders—Chang, Agarwal, and Radha Krishnan [5]. A row bypassing technique reduces switching activity and dynamic power. The designs were simulated in Tanner Spice for power and area analysis. Results show Agarwal uses least power but occupies the largest area, while Radha Krishnan offers a balanced trade-off. The study emphasizes the need for adders that optimize both power and area without degrading performance.

This paper presents a power-efficient Built-In Self-Test (BIST) using reversible logic to reduce energy consumption during testing [8]. A reversible bit-swap LFSR (BS-LFSR) built with Sam, Feynman, and RMUX gates minimizes switching activity while preserving fault coverage. The design supports 8, 16, and 32-bit configurations and is implemented using Cadence and Xilinx tools. Results show significant power savings over conventional LFSR-based designs. However, delay optimization and integration in high-speed circuits remain open challenges.

The paper [6] presents a 32-bit hybrid ALU combining reversible and irreversible logic to reduce power and delay. It uses Carry Select and Kogge–Stone adders along with a Binary-to-Excess-One Converter for speed and area optimization. Vedic multipliers are employed to efficiently handle partial products. The design demonstrates improved power efficiency and processing speed, making it valuable for modern digital circuits. This paper presents a 32-bit Brent-Kung Adder (BKA) design using Verilog HDL and Cadence tools [9]. The BKA offers logarithmic delay and balanced structure for high-speed addition. Implemented in a semi-custom flow, it achieved 43.32 µW power, 1223.91 µm² area, and 3.78 ns delay, outperforming Ripple Carry and Carry Lookahead Adders. Results confirm BKA as an efficient choice for low-power, high-speed systems. Future work can focus on scalability and advanced low-power techniques.

The study in [10] presents a Verilog-based Braun multiplier implementation on Xilinx Spartan FPGAs, comparing Ripple Carry, Brent–Kung, and Kogge–Stone adders as final stages. Kogge–Stone achieves the lowest delay, while Brent–Kung offers better area and power efficiency than RCA. Prefix adders, however, increase routing complexity and power. The work is FPGA-specific and does not explore advanced low-power techniques such as pipelining or clock gating.

The paper proposes an optimized 4×4 Braun multiplier using a 3-bit Kogge-Stone Adder (KSA) to enhance speed and reduce power for parallel processing architectures [11]. Implemented with Synopsys SAED 32 nm CMOS technology and validated through HSPICE simulations, the design achieved a delay of 4.9 ns, lower transistor count, and reduced power compared to conventional full-adder-based multipliers. While effective for low-bit designs, the work does not address scalability for higher word lengths or FPGA-based implementations, leaving scope for future research on larger and energy-efficient architectures.

The work in [12] presents a 16×16 Vedic multiplier using the Urdhva Tiryagbhyam sutra and a Kogge–Stone Adder, implemented in Verilog and tested on FPGA, achieving 29.25percent higher speed and 35.5percent lower power than conventional designs, though limited to unsigned 16-bit operations without scalability or signed support. Similarly, [13] compares carry tree adders—Kogge-Stone, Brent-Kung, and Han-Carlson—with ripple and carry-lookahead designs on a Spartan-3E FPGA, showing Kogge-Stone minimizes delay, Brent-Kung reduces area, and Han-Carlson balances both, but lacking power analysis and validation on modern FPGA or ASIC platforms. This paper proposes a hybrid Vedic multiplier using a Carry Select Adder (CSELA) with Binary-to-Excess-One Converter and Han-Carlson logic to improve speed and efficiency [14]. Multiple 2×2 Vedic multipliers are combined with hybrid adders to minimize delay, area, and power. Implemented on a Spartan-6 FPGA using Verilog HDL, the design achieved up to 30% delay reduction compared to array and Wallace multipliers. While effective, it supports only small unsigned bit-widths and is tested on Spartan-6, requiring validation on larger FPGA and ASIC platforms.

This work proposes an asynchronous Network-on-Chip (NoC) to overcome the drawbacks of synchronous designs, such as clock skew, high latency, and power consumption [15]. The architecture incorporates Label Switching (LS) and Level-Encoded Dual Rail (LEDR) encoding within a mesh topology featuring routers, network adapters, and asynchronous registers. Implemented on an Artix-7 FPGA using VHDL and validated via ChipScope, the design achieved 20% LUT savings, 26% lower delay, and 30% higher throughput compared to synchronous counterparts. Despite these improvements, the approach introduces design complexity, area overhead, and requires robust error-handling mechanisms for scalability.

This paper analyzes parallel prefix adders—Kogge-Stone, Brent-Kung, and Ladner-Fischer—to enhance speed and reduce power in VLSI systems [16]. Motivated by the inefficiency of ripple carry adders, the authors formulated carry-lookahead equations and implemented prefix-based structures. Simulations revealed that Ladner-Fischer achieved the lowest power (2.27 μW) and delay (36.98 ns), while Kogge-Stone delivered the highest speed at the expense of greater area and power. Despite these improvements, the study focuses on small-scale designs and does not address routing congestion or scalability for higher-bit architectures.

In [17], a Braun multiplier with hybrid 1-bit full adders using XOR/XNOR logic is proposed to reduce delay and power. Implemented in 130 nm CMOS and simulated in Mentor Graphics, six hybrid adder variants were evaluated, showing significant PDP reduction, further improved via particle swarm optimization. The study is limited to simulations, without FPGA or ASIC implementation.

3 Proposed Design Methodology

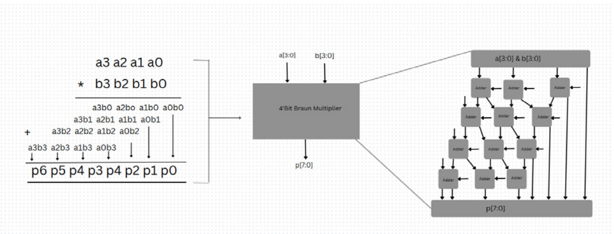


Figure 1. Braun Multiplier Architecture

The above Figure 1 is the 4-bit Braun Multiplier's block diagram. It generates an 8-bit product output by multiplying two 4-bit inputs using structural logic blocks. The design is broken up into several logical sections, each of which has significant sub-blocks.

The method used to design, simulate, optimize, and implement the Braun multiplier using different adder architectures is thoroughly and methodically explained in this section. For scalable and energy-conscious VLSI applications in particular, the methodology places a strong emphasis on hardware regularity, comparative evaluation, and real-world synthesis on FPGA devices.

3.1 Design Philosophy and Architecture Selection

Fast, space-efficient, and power-efficient arithmetic units are essential for contemporary digital systems. Because of their straightforward, fully combinational, and spatially regular layout, Braun multipliers provide a distinct advantage over other multiplier architectures and are therefore perfect for hardware mapping.

The Braun multiplier has a predictable grid-like structure, in contrast to Booth multipliers or Wallace trees, which entail intricate logic reduction and erratic partial product handling. Without altering the main dataflow, this arrangement enabled us to methodically scale the design across several word lengths, including lower to higher bit widths.

Braun multipliers are best suited for unsigned integer multiplication, where:

- AND gates are used to create partial products, which are then arranged in a 2D matrix and added together using half and full adders. In a ripple-like flow, carries propagate vertically and sums diagonally.

For two n-bit inputs A and B, the partial products are mathematically expressed as Equation (5):

$$PP_{i,j} = A_i \cdot B_j, \quad 0 \le i, j < n \tag{5}$$

Equation (5) defines the fundamental partial product generation step in a Braun multiplier.

3.2 Adder Architecture Choices and Their Role

Speed, area, and power consumption are directly affected by the critical path in the Braun multiplier, which is formed by the summation of partial products. We used three structurally distinct adder architectures to assess this.

3.2.1 Ripple Carry Adder (RCA)

Each carry-out is connected to the subsequent carry-in in RCA, which is made up of a chain of full adders. Particularly for low-bit designs, it is straightforward and space-efficient. However, at large bit-widths, its linear carry propagation results in increased delays. In our project, RCA was used as a reference design for benchmarking sophisticated adder architectures and was structurally implemented using full adder modules.

3.2.2 Brent-Kung Adder (BKA)

Figure 2 shows the Brent-Kung Adder, which uses a parallel prefix structure to compute carry signals hierarchically. It minimizes the number of prefix nodes and achieves logarithmic delay with reduced wiring complexity [18].

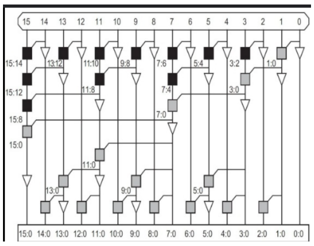


Figure 2. Brent-Kung Adder

The BKA is a parallel prefix adder that uses a logarithmic delay structure based on black and gray cells to calculate carries. Compared to ripple-based designs, it more effectively balances area and speed. In order to examine resource consumption and delay enhancements, we structurally implemented BKA and incorporated it into Braun multipliers across multiple bit-widths.

3.2.3 Kogge-Stone Adder (KSA)

Figure 3 shows the Kogge-Stone Adder architecture, where propagate and generate signals are calculated in the first stage and then combined using parallel prefix logic.

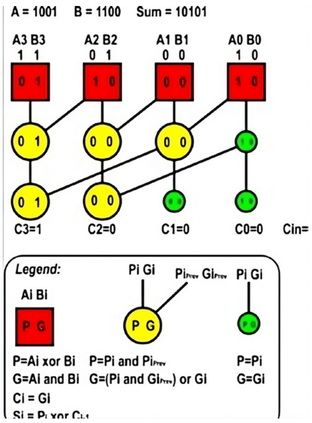


Figure 3. Kogge-Stone Adder

This structure enables fast carry computation, making it suitable for high-speed applications despite increased wiring complexity [19].

By producing carry signals in parallel across several levels, KSA provides the least amount of delay. It is best for speed even though it adds more area overhead and routing complexity. The KSA was integrated into Braun multipliers of higher bit-widths for performance evaluation and structurally implemented using prefix logic for practical assessment.

There are distinct trade-offs between area, speed, and power in each of these architectures. To determine their respective and relative effects on multiplier performance across different bit-widths, all were put into practice and tested.

3.3 Structural Verilog-Based Implementation

We implemented Braun multipliers for lower-bit to higher-bit input sizes using structural Verilog. This allowed us to precisely control gate-level interconnects and datapath construction, which is essential for synthesizing and evaluating realistic FPGA implementations.

The following were part of the design process:

- Using simple gates (XOR, AND, OR) to construct half and full adders utilizing AND gates set up in a 2D array to generate partial products.
- Using half and full adders to sum partial products diagonally and propagating carries vertically in accordance with Braun's structured layout.
- Replacing all Ripple Carry Adders (RCA) throughout the multiplier design with Brent-Kung Adders (BKA) and Kogge-Stone Adders (KSA) to evaluate their performance impact across every summation stage, not just the final stage.

Prefix logic was formally modeled as shown in Equations (6)–(9):

$$C_{k+1} = G_k + (P_k \cdot C_k) \tag{6}$$
$$S_k = P_k \oplus C_k \tag{7}$$
$$G_{out} = G_2 + P_2 \cdot G_1 \tag{8}$$
$$P_{out} = P_2 \cdot P_1 \tag{9}$$

Equations (6)–(9) capture the carry, sum, generate, and propagate relations of the prefix adder logic.

For clarity, a notation table is included:

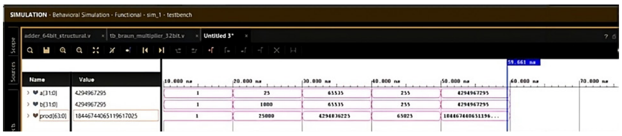
Symbol Definitions

The following symbols are used throughout this work:

- $PP_{i,j}$: Partial product at bit position (i, j)
- G, P: Generate and Propagate signals
- C_k, S_k: Carry and Sum at stage k
- G_{out}, P_{out}: Group Generate and Group Propagate

3.4 Simulation and Functional Verification

A two-step simulation and verification procedure was applied to every Braun multiplier design (with RCA, BKA, and KSA).



SIMULATION - Behavioral Simulation - Functional - sim_1 - testbench

Name	Value
a[31:0]	4294967295
b[31:0]	4294967295
prod[63:0]	18446744065119617025

Figure 4. Simulation of Braun Multiplier

Figure 4 shows the functional simulation waveform of the Braun multiplier. It illustrates the correct propagation of partial products and the final result, confirming accurate operation of the structurally integrated adder components under test conditions.

3.4.1 Unit-Level Verification

To verify logical correctness across all input combinations, each module—including the Half Adder, Full Adder, RCA, BKA, and KSA—was tested using specialized Verilog testbenches.

3.4.2 Multiplier-Level Simulation

Test cases such as 15 × 15 = 225 and 255 × 255 = 65025 were used to simulate complete multiplier circuits. Vivado and ModelSim were used for the simulations. To verify accuracy, waveform results were examined.

LED output was impractical for 32-bit designs due to the limited number of I/O pins. For real-time debugging, we used Xilinx ChipScope Pro ICON and VIO cores via JTAG, which allowed monitoring of internal signals. These tools were used purely for verification and do not constitute a novel contribution.

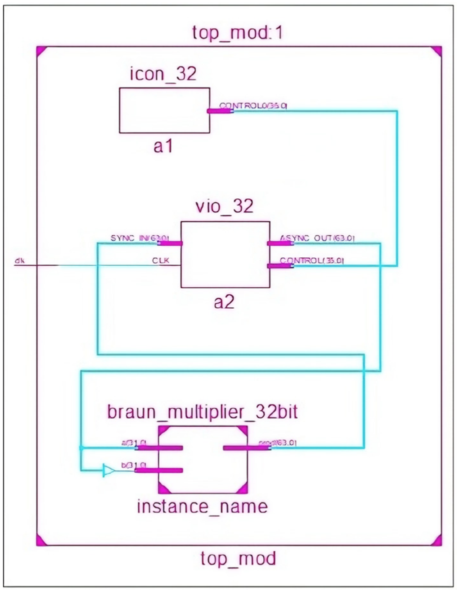


Figure 5. RTL view of Braun Multiplier

3.5 RTL Design

The Figure 5 is the RTL view of the 32-bit Braun multiplier with ChipScope integration.

The 32-bit Braun multiplier design on FPGA is integrated with Xilinx ChipScope Pro, as shown in this figure. The `icon_32` (Integrated Controller), `vio_32` (Virtual Input/Output), and the `braun_multiplier_32bit` module are the three primary parts of the setup. The FPGA and PC can communicate in real time thanks to the `icon_32`, which is connected to the JTAG interface. The `vio_32` core is used to capture the 64-bit output (`ans[63:0]`) during operation and to inject 32-bit inputs (`a[31:0]` and `b[31:0]`) into the multiplier.

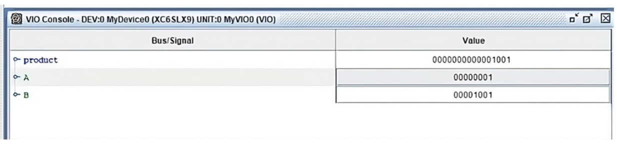
This virtual probing technique offers a potent substitute for on-board LEDs or switches, which are unable to readily visualize 32-bit data due to I/O constraints. It allows live signal observation and dynamic input control without requiring the FPGA to be reprogrammed for each test scenario. This approach is perfect for real-time testing and performance analysis of complex digital systems because it significantly enhances verification and debugging, particularly in large-bit, high-performance designs.

3.6 FPGA Implementation

Figure 6 shows the implementation of multiplier designs on the **Xilinx Spartan-6 FPGA**. Our objective was to assess the percentage of requirements: On actual hardware, adder architecture has an impact on speed, area, and resource usage.

Included in the implementation were:

- Using ChipScope Pro for higher bit designs because of I/O limitations.
- Integrating ICON and VIO cores to monitor signal behavior via PC.



Bus/Signal	Value
product	0000000000001001
A	00000001
B	00001001

Figure 6. Implementation on ChipScope

- Connecting lower bit multipliers with DIP switches and LEDs.

Metrics recorded in the synthesis reports include:

- Slice usage and LUT utilization
- Power estimation

To evaluate efficiency, Power–Delay Product (PDP) and Area–Delay Product (ADP) were calculated as shown in Equation (10) (11):

$$PDP = Power \times Delay \tag{10}$$

$$ADP = Area \times Delay \tag{11}$$

Equation (10) and (11) serves as the basis for comparing the trade-offs between power, area, and delay in different multiplier implementations.

Although implemented on Spartan-6, the architecture is portable to more advanced FPGA families such as Artix-7 and UltraScale+, which can provide improved speed, lower dynamic power, and enhanced routing resources.

4 Results And Analysis

4.1 Result

The results show how various adder architectures affect the Braun multiplier's power, delay, and area performance. For various bit-widths, simulation and FPGA synthesis were used to assess each configuration.

Table 1. Power and Delay Performance of Braun Multiplier with Different Adders

Bits	Adder	Power (mW)	Delay (ns)
4	HA/FA	0.076	9.33
	KSA	0.085	8.30
	BKA	0.079	9.12
8	HA/FA	1.463	10.65
	KSA	1.619	14.42
	BKA	0.684	19.82
16	HA/FA	9.207	28.31
	KSA	21.115	23.28
	BKA	4.418	47.44
32	HA/FA	32.279	60.76
	KSA	161.07	47.86
	BKA	30.366	109.89

Key Observations

In terms of power consumption and timing delay, the performance comparison of Braun multipliers implemented with various adder architectures across increasing bit-widths is summarized in Table 1.

- **Power Consumption:**
  - All three adder types show comparatively low power consumption for smaller bit-widths (4-bit and 8-bit), with HA/FA being the most energy-efficient.
  - The power needed by the Kogge-Stone Adder (KSA) rises dramatically with bit-width, up to **161.07 mW** for the 32-bit design—more than **5× more than BKA**. For larger bit-widths, the Brent-Kung Adder (BKA) continuously uses less power than KSA, which makes it a better choice for applications that are power-sensitive.
- **Timing Delay:**
  - Because of its parallel prefix structure, the Kogge-Stone Adder (KSA) exhibits the lowest delay in the majority of configurations, especially at the 4-bit and 8-bit levels (**8.302 ns** and **14.422 ns**).
  - However, the delay in BKA and HA/FA increases significantly as bit-width increases, with BKA reaching **109.89 ns** at 32 bits.
  - KSA exhibits superior scalability in terms of delay when compared to the other two, providing a balance between speed and moderate delay growth.
- **Trade-Off Insight:**
  - BKA provides the best balance between **power and speed**, particularly for **mid-range bit-widths**.
  - HA/FA architecture is still the most straightforward and low-power solution at smaller scales but struggles with delay as bit-width increases.

Table 2. LUT Utilization of Different Adders

SL.No	Multiplier	Total LUTs	Used LUTs	Percentage
1	Braun with HA/FA	5720	2932	51
2	Braun with KSA	5720	536	7
3	Braun with BKA	5720	167	2

Key Observations

Significant trends in resource efficiency are revealed by comparing the LUT utilization as shown in Table II. HA/FA-based Braun multipliers used 2932 LUTs, or 51% of the total resources available. Its lack of structural optimization and sequential carry propagation make it the most resource-intensive configuration.

By using parallel prefix logic, the Kogge-Stone Adder (KSA) design maintained high-speed performance while requiring only 536 LUTs (7%), a significant reduction in resource usage. For speed-sensitive applications, this architecture presents an attractive trade-off despite the extra routing complexity.

Using only 167 LUTs, or 2% of the total, the Brent-Kung Adder (BKA) configuration showed the highest area efficiency. Its semi-parallel prefix structure balances area and delay efficiently, making it well-suited for FPGA applications with low power or limited area.

PDP and ADP Analysis

To provide a comprehensive evaluation, Power–Delay Product (PDP) and Area–Delay Product (ADP) were calculated using the formulas in Equation (10) (11), where PDP considers power and delay, and ADP accounts for area or FPGA resource utilization through LUTs:

These metrics help assess the combined efficiency of energy and resource utilization. Representative results are shown in Table 3.

Table 3. PDP and ADP Comparison of Braun Multipliers

Bits	Adder	PDP (mW·ns)	ADP (LUT·ns)
8	HA/FA	15.57	31,200
	KSA	23.35	7,730
	BKA	13.55	3,310
16	HA/FA	260.9	83,000
	KSA	491.2	12,490
	BKA	209.7	7,930
32	HA/FA	1960.8	178,150
	KSA	7711.7	25,642
	BKA	3336.4	18,353

Table 3 presents the Power–Delay Product (PDP) and Area–Delay Product (ADP) values for Braun multipliers using HA/FA, Kogge-Stone Adder (KSA), and Brent-Kung Adder (BKA). The results clearly show that KSA achieves the lowest delay but at the expense of significantly higher PDP and ADP, especially for 32-bit designs, due to increased power consumption and routing overhead. In contrast, BKA demonstrates superior energy efficiency, achieving the lowest PDP and ADP across most bit-widths, making it well-suited for area- and power-constrained FPGA applications. The HA/FA design remains competitive at small bit-widths but becomes inefficient at higher word sizes due to its linear delay growth. These findings confirm that while KSA is optimal for high-speed applications, BKA provides the best trade-off between power, delay, and area for scalable VLSI and FPGA-based implementations. “Thus, BKA proves more energy- and area-efficient for constrained designs, while KSA suits high-speed but power-tolerant applications.”

4.2 Analysis

From the standpoint of a real-time application, particular design specifications like speed, area, and power limitations should direct the choice of adder architecture within the Braun multiplier. The Kogge-Stone Adder (KSA) is the best option if the target application requires high-speed performance, such as in real-time image processing, cryptography, or AI accelerators, because it uses a parallel prefix structure to compute carries quickly. However, this results in moderate resource utilization and higher power consumption. The Brent-Kung Adder (BKA), on the other hand, works better for designs limited by space or power, such as portable embedded systems or low-power Internet of Things devices. With the lowest LUT usage (2%) and balanced PDP and ADP results, it provides a trade-off between delay and energy efficiency. Meanwhile, the traditional HA/FA structure is still useful for small-scale designs, but as bit-width increases, it becomes ineffective. All designs were synthesized on a Xilinx Spartan-6 FPGA using Vivado 2024.2, with functional simulations performed in ModelSim. Real-time debugging and signal monitoring were carried out using Xilinx ChipScope Pro with ICON and VIO cores.

5 Conclusion and Future Scope

The Braun multiplier architecture combined with different adder structures, such as Brent-Kung Adders (BKA), Kogge-Stone Adders (KSA), and Ripple Carry Adders (RCA), was thoroughly designed, implemented, and evaluated in this work. The project focused on using Verilog HDL to model the Braun multiplier’s structure and deploying it on FPGA platforms for real-time analysis. The performance of each adder configuration was investigated in terms of delay, power consumption, and area utilization across a range of bit-widths using simulation, synthesis, and testing with ChipScope Pro. Experimental results show that the Kogge-Stone Adder provides the least delay, making it highly suitable for time-critical applications, although it incurs higher power and area requirements. In contrast, the Brent-Kung Adder achieves better power and area efficiency, making it ideal for energy-constrained designs, while the traditional HA/FA-based Braun multiplier remains practical only for small word lengths. However, the current work is limited to unsigned integer multiplication and implementation on Spartan-6 FPGA without exploring advanced low-power techniques such as pipelining, clock gating, or approximate computing. Future work can extend this study by integrating approximate adders for error-tolerant domains like image and speech processing, applying pipelining and clock gating to enhance throughput and reduce dynamic power, and porting the design to advanced FPGA families such as Artix-7 and UltraScale+ or ASIC platforms. Additionally, incorporating the multiplier within larger ALUs or DSP units would further validate its scalability and system-level performance.

References

[1] B. Neeraja, R.S.P. Goud, Design of an Area Efficient Braun Multiplier Using High Speed Parallel Prefix Adder in Cadence, in *2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)* (IEEE, 2019), pp.

1–5, `https://doi.org/10.1109/ICECCT.2019.8869307`

[2] R. Anitha, V. Bagyaveereswaran, FPGA Implementation of Braun's Multiplier Using Spartan-3E, Virtex-4, Virtex-5 and Virtex-6, in *International Conference on Web and Semantic Technology* (Springer, 2011), Vol. 197 of *Communications in Computer and Information Science*, pp. 486–494, `https://doi.org/10.1007/978-3-642-22543-7_49`

[3] K. Kalaikaviya, M. Suba, S. Sudha, Design of Low Area Braun Multiplier with Brent–Kung Adder, in *ICCSP'13: International Conference on Communications, Signal Processing and their Applications* (IEEE, 2013), pp. 447–450, available online: `https://www.academia.edu/download/34896488/42_T1.pdf`

[4] S. Karunakaran, B. Poonguzharselvi, Vlsi architecture of an 8-bit multiplier using vedic mathematics in 180nm technology, International Journal of Advances in Engineering & Technology **10**, 401 (2017), available online: `https://www.ijaet.org/media/14I39-IJAET1003317-v10-i3-pp401-410.pdf`.

[5] B. Bandre, Design and analysis of low power energy efficient braun multiplier, International Journal of New Practices in Management and Engineering **2**, 08 (2013). 10.17762/ijnpme.v2i01.12

[6] S.B. Shirol, S. Ramakrishna, R.B. Shettar, in *Computing and Network Sustainability: Proceedings of IRSCNS 2018* (Springer, 2019), pp. 239–249, `https://doi.org/10.1007/978-981-13-7150-9_25`

[7] D.K. D.K., S. R., K. B.S., VLSI Implementation of Braun Multiplier using Full Adder, in *2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC)* (IEEE, 2017), pp. 499–504, `https://doi.org/10.1109/CTCEEC.2017.8455157`

[8] S.B. Shirol, S. Ramakrishna, R.B. Shettar, Designing power-efficient bist architecture: Leveraging reversible logic for scalable digital systems, Journal of Electrical Systems **20**, 2747 (2024), licensed under CC BY–ND 4.0. 10.52783/jes.2053

[9] Y. Singh, Semicustom frontend vlsi design and analysis of a 32-bit brent-kung adder in cadence suite, arXiv preprint arXiv:2503.18070 (2025). 10.48550/arXiv.2503.18070

[10] A. Sharma, A. Mittal, A. Singh, R. Kapoor, Design and implementation of braun multiplier using verilog, in *2024 Asia Pacific Conference on Innovation in Technology (APCIT)* (IEEE, 2024), pp. 1–6, `https://doi.org/10.1109/APCIT62007.2024.10673483`

[11] M. Sakthimohan, E. Rani, P. Monisha, T. Bhuvaneswari, R. Anand, S.T. Nandhini, An Optimized 4×4 Braun Multiplier for Parallel Processing Architectures with a 3-bit Kogge–Stone Adder, in *2023 International Conference on Intelligent Technologies for Sustainable Electric and Communications Systems (iTech SECOM)* (IEEE, 2023), pp. 319–324, `https://doi.org/10.1109/iTechSECOM59882.2023.10435221`

[12] K. Yogeshwaran, R. Gowri, S. Sandhya, T. Tejeswar Reddy, T. Vijayalakshmi, Design and Simulation of 16×16 Vedic Multiplier using Kogge–Stone Adder, in *2023 7th International Conference on Computing Methodologies and Communication (ICCMC)* (IEEE, 2023), pp. 452–457, `https://doi.org/10.1109/ICCMC56507.2023.10083594`

[13] P. Kousalya, M. Lavanya, T. Archana, Carry Tree Adders on FPGA's Design and its Implementation, in *2024 First International Conference for Women in Computing (InCoWoCo)* (IEEE, 2024), pp. 1–5, `https://doi.org/10.1109/INCOWOCO64194.2024.10863428`

[14] V. Thamizharasan, N. Kasthuri, Fpga implementation of proficient vedic multiplier architecture using hybrid carry select adder, International Journal of Electronics **111**, 1253 (2024). 10.1080/00207217.2023.2245194

[15] R. Madhuri, P. Bachanna, B. Pavan, M.Z. Begum, Design and Performance analysis of Asynchronous Network on Chip for Streaming data Transmission on FPGA, in *2024 International BIT Conference (BITCON)* (IEEE, 2024), pp. 1–6, `https://doi.org/10.1109/BITCON63716.2024.10985385`

[16] D. Mittal, Analysis of parallel prefix adders with low power and higher speed, in *2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC)* (IEEE, 2022), pp. 288–288, `https://doi.org/10.1109/ICESC54411.2022.9885726`

[17] I. Omesh, B. Supriya, Y. Yamini, A. Likitha, D. Aditya, Design and implementation of low power and highspeed braun multiplier using hybrid full adder, i-Manager's Journal on Circuits & Systems **9**, 23 (2021). 10.26634/jcir.9.2.16831

[18] S.V. Padmajarani, M. Muralidhar, Comparison of parallel prefix adders performance in an fpga, International Journal of Engineering Research and Development **3**, 62 (2012), `https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=9690309826e9b882a430f27c0f737c853832f6e7`.

[19] U. Penchalaiah, V.G. Siva Kumar, Design of High-Speed and Energy-Efficient Parallel Prefix Kogge-Stone Adder, in *2018 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN)* (IEEE, 2018), pp. 1–7, `https://doi.org/10.1109/ICSCAN.2018.8541143`