Optimized 4:2 Compressor based Approximate Multiplier for Low Power Digital Signal Processing

K. Ezhilarasan
Associate Professor
Department of Electronics and Engineering
Communication Engineering Technology
SJC Institute of Technology, Chikkaballapura

Kaveri S.
Department of Electronics and Communication Engineering
SJC Institute of Technology, Chikkaballapura

K. Bhavya Sree
Department of Electronics and Communication
SJC Institute of Technology, Chikkaballapura

Shreyas M.
Department of Electronics and Communication Engineering
SJC Institute of Technology
Chikkaballapura

Vidyashree K.S.
Department of Electronics and Communication Engineering,
SJC Institute of Technology, Chikkaballapura

ABSTRACT
In many digital signal processing systems, multiplication is the workhorse and also the biggest drain on power, area, and latency. This paper introduces an elegant compressor an enhanced multiplier built around approximate 4:2 compressors that trims energy and hardware cost while preserving the fidelity that matters. The design uses a hybrid compression strategy; the most significant bits are handled by exact 4:2 compressors to protect numerical accuracy where it counts, while the least significant bits are handled by lightweight approximate 4:2 compressors. By selectivity relaxing correctness only where the impact on the overall result is small, the architecture dramatically reduces switching activity, shortens critical – path delay, and shrink resource usage. Partial products are aggregated in a structured compression tree and finished off with a fast final adder to produce the product. To quantify trade-offs, the design is evaluated on power, propagation delay, area, and accuracy metrics (Mean Error Distance and Error Rate). Results show notable improvements in power–delay product and area efficiency compared with exact multipliers, while introducing only modest, generally acceptable errors. The result is a compelling multiplier option for DSP contexts that can tolerate slight inaccuracies in exchange for lower power and smaller silicon footprint.

Keywords
Approximate Computing 4:2 Compressor, Approximate Multiplier, Hybrid Compression, Low Power Design Digital Signal Processing (DSP), Partial Product Reduction, Error Tolerant Systems, Power-Delay Product (PDP), VLSI Architecture

1. INTRODUCTION
The rapid advancement of digital technology has created an increasing demand for high-performance and energy-efficient computing systems. Modern Very Large-Scale Integration (VLSI) circuits are expected to perform complex arithmetic operations at high speed while consuming minimal power and chip area [1], [2]. Arithmetic units are the fundamental building blocks of digital systems and are extensively used in Digital Signal Processing (DSP), image processing, video processing, wireless communication, artificial intelligence, machine learning, and multimedia applications. Among these arithmetic units, multipliers play a significant role because multiplication is one of the most frequently executed and computationally intensive operations [2], [3].

In DSP applications, multiplication is required in filtering, convolution, correlation, Fast Fourier Transform (FFT), Discrete Cosine Transform (DCT), image enhancement, compression, and feature extraction. Similarly, machine learning algorithms and neural network accelerators perform millions of multiplication operations during training and inference. Therefore, the efficiency of the multiplier directly affects the overall performance, power consumption, and hardware utilization of the complete system [3], [4].

Conventional multiplier architectures are designed to generate exact results with high computational accuracy. Although these multipliers provide error-free outputs, they require complex logic circuits, larger silicon area, higher transistor count, and increased power consumption. As CMOS technology continues to scale and portable electronic devices become more common, reducing power consumption while maintaining high processing speed has become one of the primary objectives of VLSI designers [4], [5]. Consequently, researchers are exploring alternative arithmetic architectures that can improve hardware efficiency.of producing mathematically exact results, approximate computing intentionally introduces small computational errors to reduce hardware complexity, propagation delay, and power consumption [5], [6]. This technique is particularly effective for error-resilient applications where small inaccuracies do not noticeably affect the output quality. Applications such as image processing, video processing, multimedia systems, speech recognition, and machine learning naturally possess error tolerance, making approximate arithmetic circuits highly suitable for these domains [6], [7].

Approximate arithmetic has attracted considerable attention in recent years because it offers significant improvements in energy efficiency without severely affecting application-level accuracy. Among various approximate arithmetic units, approximate multipliers have become an important research topic because multiplication contributes a major portion of the computational complexity in many digital systems. By simplifying arithmetic

logic, approximate multipliers achieve lower power consumption, reduced hardware area, shorter critical paths, and higher operating frequency [7], [8].

A multiplier generally consists of three major stages: partial product generation, partial product reduction, and final addition. Among these stages, partial product reduction contributes significantly to the overall delay and hardware complexity. Therefore, optimizing this stage is essential for improving multiplier performance [8], [9]. Compressor circuits are widely used to reduce the number of partial products efficiently. Compared to conventional full-adder-based reduction methods, compressors reduce more input bits simultaneously, resulting in lower delay and improved hardware utilization.

The 4:2 compressor is one of the most widely adopted compressor architectures because it compresses four input bits into two output bits along with carry signals. This significantly reduces the height of the partial product tree, minimizes the number of reduction stages, and improves multiplication speed [9], [10]. Conventional 4:2 compressors provide accurate computation but require relatively complex logic implementation, leading to higher power dissipation and larger hardware area.

To overcome these limitations, approximate 4:2 compressors have been proposed. These compressors simplify the logic equations by removing non-critical logic paths and shortening carry propagation. As a result, they reduce transistor count, propagation delay, switching activity, and power consumption while maintaining acceptable computational accuracy [10], [11]. However, excessive approximation may increase computational errors, making it essential to achieve an optimal balance between hardware efficiency and output accuracy.

Several approximate compressor architectures have been reported in the literature, each focusing on different trade-offs among power, delay, area, and error characteristics. Although many existing designs improve one performance parameter, they often compromise another. Therefore, designing an approximate compressor that simultaneously achieves low power consumption, small hardware area, reduced delay, and acceptable error remains an active research challenge [11], [12].

In this work, an optimized approximate multiplier based on 4:2 compressors is proposed for low-power digital signal processing applications. The proposed architecture employs approximate compressors in the Least Significant Bit (LSB) positions while retaining exact compressors in the Most Significant Bit (MSB) positions. Since approximation errors generated in LSBs have minimal impact on the final multiplication result, while errors in MSBs significantly affect computational accuracy, this hybrid approach provides an effective trade-off between performance and accuracy [12], [13].

Furthermore, the simplified compressor architecture reduces The VLSI circuits must be tested, as testing VLSI circuits presents numerous difficulties in terms area overhead, power, and speed. One way for testing a complex architecture of VLSI circuits is the generations of low-power LFSR test patterns [13],[14].

Patel et al. [8] (2025) proposed an optimized compressor-based multiplier architecture that enhances hardware utilization and computational speed. By reducing the number of reduction stages and improving parallelism, the design achieves better performance in DSP systems. The architecture also minimizes area and power consumption.

Sharma et al. [9] (2025) introduced a high-speed approximate

High-speed computation using multiple 4-bit ALU architectures the work focuses on improving important VLSI parameters such as speed, power-consumption, area and computational efficiency.[14]

2. LITERATURE SURVEY

Rashidi et al. [1] (2026) proposed an area-efficient and high-performance approximate multiplier based on approximate 4:2compressor. The design introduces multiple compressor structures to reduces to critical path delay and hardware complexity. By eliminating unnecessary carry propagation paths, the results indicate that the proposed multiplier is highly suitable for image processing applications where minor errors can be tolerated while achieving significant improvements in speed and energy efficiency.

Chakraborty et al. [2] (2025) developed optimized approximate 4:2 compressor architectures aimed at reducing logic complexity. The proposed compressors utilize simplified Boolean expressions to minimize gate count and switching activity. This results in improved computational speed and reduced power consumption. The study demonstrates that compressor optimization plays a crucial role in enhancing multiplier performance in VLSI arithmetic circuits.

Kumar et al. [3] (2025) presented an area-efficient approximate multiplier architecture using compressor-base reduction techniques. The design focuses on minimizing hardware utilization by replacing exact components with approximate ones in non-critical paths. Simulation results show a reduction in power consumption and delay while maintaining acceptable output accuracy for DSP applications.

Jaswal et al. [4] (2025) introduced a low-power approximate multiplier tailored for neural network and machine learning applications. The architecture significantly reduces energy consumption by simplifying arithmetic operations and limiting switching activity. Despite approximation, the design maintains acceptable accuracy levels, making it suitable for AI accelerators and embedded systems.

Krishna et al. [5] (2025) proposed an approximate signed multiplier architecture that enhances performance in image processing systems. The design reduces circuit delay and hardware complexity by optimizing the partial product reduction stage. The results show improved processing speed and efficiency without significantly

Darabi et al. [6] (2025) developed an inverted compressor-based approximate multiplier that balances accuracy and hardware efficiency. The architecture modifies traditional compressor structures to reduce logic depth and improve performance. This approach achieves better trade-offs between error rate and computational efficiency in VLSI systems.

Singh et al. [7] (2025) designed a low-power and high-speed Approximate 4:2c compressor specifically for multiplier applications. The proposed compressor improves partial product reduction efficiency and reduces propagation delay. The study highlights that optimized compressor design directly impacts multiplier performance and energy consumption.

multiplier for VLSI applications. The design reduces the critical path delay by optimizing arithmetic operations and compressor placement. The results demonstrate improved throughput and efficiency in high-speed processing systems.

Mehta et al. [10] (2025) developed an efficient approximate multiplier architecture that reduces hardware area and power consumption. The design focuses on optimizing the reduction

stage using approximate compressors, leading to improved performance in signal processing applications.

Li et al. [11] (2024) presented an energy-efficient approximate multiplier using simplified compressor circuits. The design reduces switching activity and dynamic power consumption while maintaining acceptable accuracy levels. The results show improved performance compared to traditional multiplier designs.

Zuhair et al. [12] (2024) proposed optimized approximate 4:2 compressor circuits that reduce logic complexity and enhance computational speed. The design improves multiplier efficiency by minimizing delay and hardware requirements.

Vakili et al. [13] (2024) introduced compact approximate compressors for Dadda multipliers. The architecture reduces hardware area and delay by optimizing compressor structures. The results demonstrate improved power efficiency and performance.

Bhandari et al. [14] (2024) conducted a comprehensive analysis of approximate compressor architectures for multipliers. The study evaluated power consumption, delay, and accuracy metrics, highlighting the advantages of approximate designs in arithmetic circuits.

Ren et al. [15] (2024) proposed an approximate multiplier for deep neural network accelerators. The design reduces energy consumption while maintaining computational efficiency, making it suitable for AI-based hardware systems.

Gupta et al. [16] (2024) developed a low-power approximate compressor architecture that improves propagation delay and reduces hardware complexity. The design enhances performance in multimedia and signal processing systems.

Zhang et al. [17] (2024) introduced a probability-based approximate compressor design that reduces error distance in multiplier circuits. The architecture improves arithmetic performance while maintaining acceptable accuracy levels.

Yang et al. [18] (2024) proposed optimized approximate multiplier architectures that improve computational speed and power efficiency. The design is suitable for high-performance VLSI systems requiring efficient arithmetic operations.

Guntur et al. [19] (2024) developed an optimized compressor selection algorithm for multiplier design. The algorithm improves partial product reduction efficiency and enhances overall multiplier performance.

Rehman et al. [20] (2023) proposed a novel approximate multiplier architecture focused on reducing hardware complexity and delay. The design improves efficiency in signal processing applications by optimizing arithmetic operations.

Jiang et al. [21] (2023) developed approximate compressor circuits for multiplier architectures. The design reduces power consumption and hardware utilization while improving arithmetic performance.

Gupta et al. [22] (2023) introduced a low-power approximate multiplier design for DSP applications. The architecture reduces energy consumption while maintaining computational speed and acceptable accuracy

3. THE PROPOSED METHEDOLOGY

The proposed image processing methodology involves a systematic sequence of steps to acquire, enhance, analyze, and interpret digital images. Initially, the image is acquired using a camera, sensor, or other imaging device and converted into a suitable digital format. Preprocessing is then performed to remove noise, correct distortions, and improve image quality through techniques such as filtering, contrast enhancement, and resizing.

Next, image segmentation is used to separate the important regions or objects from the background.

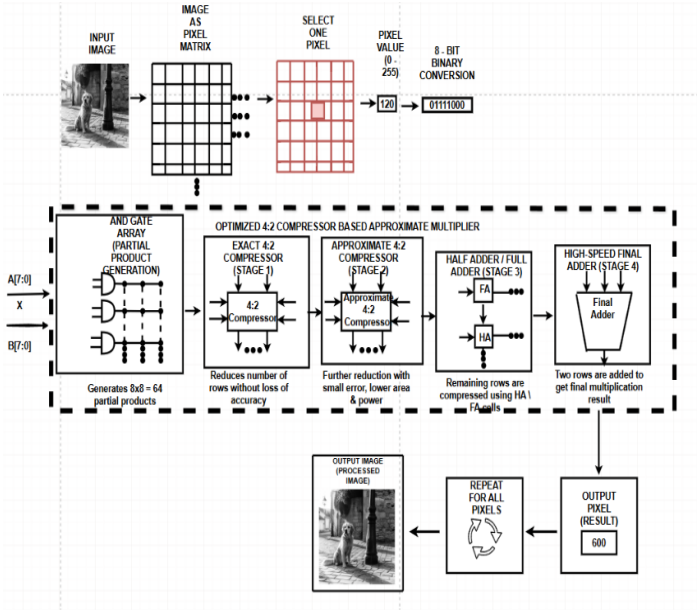

Figure 1: Project based promote method of low power 4:2 compressor-based approximate multiplier for dsp

The relevant features such as edges, shapes, textures, and intensity levels are extracted from the segmented image. These features are subsequently analyzed using suitable image processing or machine learning techniques for object detection, classification, or recognition. Finally, the processed results are evaluated to determine the accuracy and effectiveness of the proposed image processing method. This methodology enables efficient extraction of meaningful information from digital

image for various applications such as medical imaging, surveillance, remote sensing, and industrial, surveillance, remote sensing, and industrial inspection.

By leveraging advanced digital image processing techniques and computer vision algorithms, this methodology transforms raw pixel data into structured, actionable insights with high precision. rather than relying on manual analysis, it automates complex visual recognition tasks to enhance efficiency and accuracy across diverse domains. In healthcare, t assists medical imaging by detecting subtle structural anomalies and facilitating accurate diagnostic segmentation. Within security and surveillance, it powers real-time threat identification, object tracking, and crowd monitoring. In satellite remote sensing, it enables multi-spectral land-use classification, environmental assessment, and geographical mapping. Finally, in high-throughput smart manufacturing, it drives automated industrial inspection systems to perform real-time quality control, defect detection, and assembly verification. In high-throughput smart manufacturing environments where thousands of components move across production lines every minute, this methodology serves as the foundation for automated industrial inspection systems. By integrating high-resolution digital imaging with advanced computer vision models, it continuously monitors assembly processes in real time to detect sub-millimeter structural defects, surface blemishes, or component misalignments that human inspectors might easily miss. This automated oversight streams continuous quality control data back to the factory floor, minimizing costly manufacturing downtime.

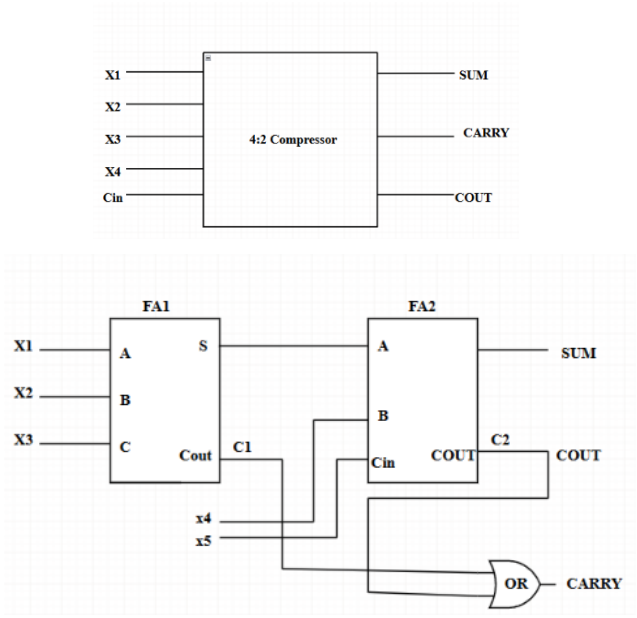

Figure 2: Symbol and circuits of the conventional 4:2 compressor

The proposed optimized 4:2 compressor-based on the approximate multiplier is designed to achieve low power consumption and high-speed operation for digital signal processing applications by combining exact and approximate computation techniques. Multiplication is a fundamental operation in DSP systems, but conventional multipliers consume significant power and hardware resources due to the large number of partial products and addition stages. To address these challenges, the proposed architecture introduces a hybrid approach that balances accuracy and efficiency by dividing the computation into exact and approximate regions. The multiplication process begins with the generation of the partial product matrix, where two n-bit input operands are multiplied at the bit level using AND operations.

Each bit of one operand is combined with every bit of the other operand to produce an n x n matrix of partial product arranged in a structured format. This matrix contains a large number of bits that must be reduced efficiently to obtain the final result. Since this stage forms the foundation of multiplication, it is implemented without approximation to preserve the correctness of the initial data. Following the partial product generation, the reduction of these products is performed using a compressor tree structure.

The compressor tree plays a critical role in minimizing the number of rows in the partial product matrix, thereby reducing the complexity and delay of the overall multiplication process. In the proposed design, this compressor tree is divided into two regions: the exact compression region and the approximate compression region. This division enables selective application of approximation, which is the key concept behind the design. In the exact compression region, which corresponds to the most significant bits, accurate computation is performed using exact 4:2 compressors along with full adders and half adders.

A 4:2 compressor takes multiple input bits and compressors them into fewer outputs, there by reducing the number of intermediate signals. The use of full adders and half adders further assists in combining partial products efficiently. Since errors in the most significant bits can significantly affect the final output, this region maintains full accuracy to ensure reliable results. In contrast, the approximate compression region handles the least significant bits, where small errors have minimal impact on the overall output. In this region, approximate 4:2 compressors are employed, which use simplified logic expressions to reduce the number of logic gates and switching activity. This simplification leads to reduced power consumption, smaller hardware area, and faster computation.

Although approximation introduces slight in accuracies, these errors are confined to the least significant bits and are acceptable for many DSP applications such as image and audio processing, where perfect precision is not always required. The combination of exact and approximate compression forms a hybrid reduction strategy that effectively balances accuracy and performance. By applying exact computation to critical bits and approximate computation to non-critical bits, the proposed design achieves significant improvements in efficiency without compromising the overall quality of the result. This hybrid approach is the main strength of the architecture, enabling it to meet the demand so flow-power and high-speed systems. After passing through multiple levels of compression in the compressor tree, the partial product matrix is reduced to two rows consisting of sum bits and carry bits. This reduction simplifies the final stage of computation and minimizes propagation delay.

The final addition stage is then performed using a fast adder, such as a ripple carry adder or a carry look-ahead adder, which combines the two rows to produce the final product. The output of this stage is a 2n-bit result representing the multiplication of the input operands. Overall, the proposed optimized 4:2 compressor-based approximate multiplier provides an efficient solution for modern digital systems by reducing power consumption, improving speed, and minimizing hardware complexity. The use of approximation in the least significant region, combined with accurate computation in the most significant region, ensures that the design maintains acceptable

accuracy while achieving significant performance gains. Therefore, this architecture is highly suitable for low-power DSP applications, where energy efficiency and speed are critical considerations.

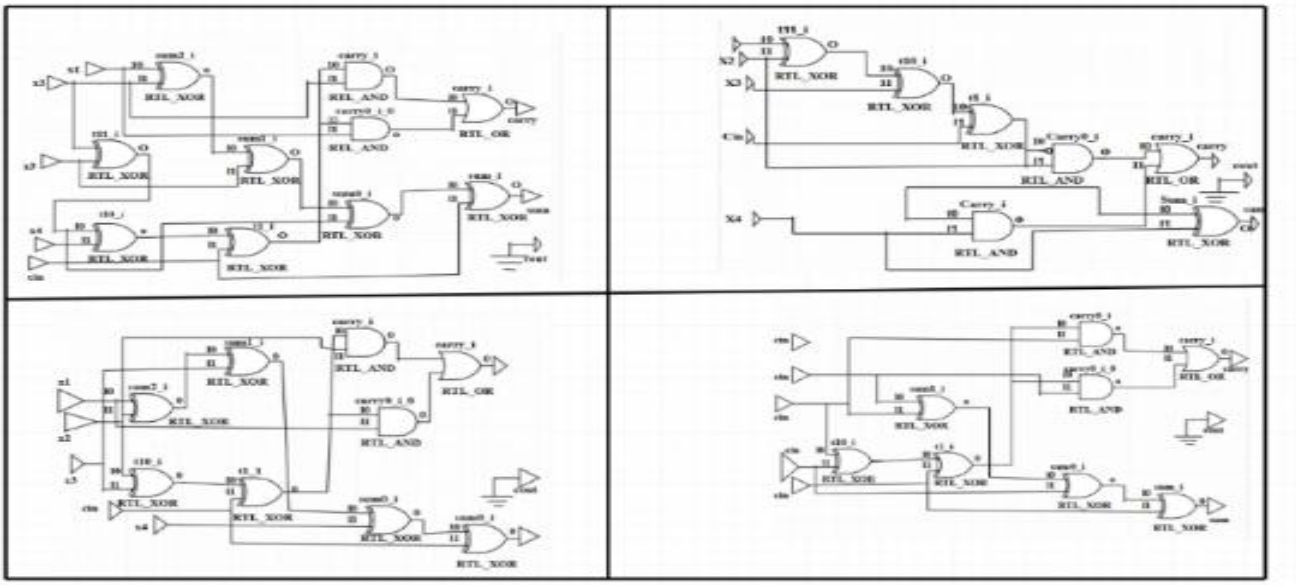

Figure3: Proposed approximate 4:2 compressors: a) Type-1 b) Type- 2 c) Type-3 d) Type-4

Table 4: Truth table of the proposed approximate 4:2 compressor

INPUTS				TYPE 1				TYPE 2			TYPE 3			TYPE 4		
X1	X2	X3	X4	Cin	S	C	Cout	S	C	Cout	S	C	Cout	S	C	Cout
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	1	0	1	0	1	0	0	0	1	0
0	0	1	0	1	0	1	0	1	0	0	1	0	0	1	0	0
0	0	1	1	0	0	0	0	1	0	0	0	0	1	1	0	0
0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0
0	1	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0
0	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0
0	1	1	1	1	1	0	1	0	0	1	0	0	0	0	1	0
1	0	0	0	1	0	0	1	0	0	1	0	0	0	1	0	0
1	0	0	1	0	0	1	0	1	0	1	1	0	0	1	0	0
1	0	1	0	1	0	0	1	0	0	1	0	0	0	1	0	0
1	0	1	1	0	1	0	0	1	0	1	0	1	0	1	0	0
1	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0
1	1	0	1	0	0	1	0	0	0	0	0	1	0	1	0	0
1	1	1	0	1	0	0	1	0	0	1	1	0	0	1	0	0
1	1	1	1	0	1	0	0	0	0	0	1	0	0	1	0	0
1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0
1	0	0	1	0	1	0	0	0	0	0	0	1	0	1	0	0
1	0	1	0	0	0	1	0	1	0	0	1	0	0	1	0	0
1	0	1	1	1	0	0	1	0	0	1	0	0	0	1	0	0
1	1	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0
1	1	0	1	1	0	0	1	0	0	1	0	0	0	1	0	0
1	1	1	0	0	0	1	0	1	0	0	1	1	0	1	0	0
1	1	1	1	0	1	0	0	1	0	0	1	0	0	1	0	0

Type 1:

Sum = x1 + x2 + x3 + x4 + Cin Cout = 0

Carry = (x2 + x3 + x4 + Cin) x2 + (x2 + x3 + x4 + Cin) sum=x1 + x2 + x3 + x4 + Cin

Cout = 0

Carry = (x1 + x2 + x3 + Cin) x2 + (x1 + x2 + x3 + Cin) x4

Type 3:

$Sum = x1 + x2 + x3 + x4 + C_{in}$ $Cout = 0$

$Carry = (x2 + x3 + C_{in})\ x2 + (x2 + x3 + C_{in})\ x1$ Type 4:

$Sum = x1 + x2 + x3 + x4$ $Cout = 0$

$Carry = (x2 + x3 + x4)\ x2 + (x2 + x3 + x4)\ x1$

The figure 3 illustrates four optimized approximate 4:2 compressor architectures, namely type 1, type 2, type 3, and type 4, which are designed as low-complexity building blocks for an approximate multiplier. A 4:2 compressor accepts four input bits, $x1$, $x2$, $x3$, and $x4$, together with an input carry Cin, and generates Sum (S), Carry (C), and Carry-out (Cout) signals. In the proposed approximate $X1$,$X2$, $X3$, and C_{in}. Type 4 further simplifies the arithmetic operation

by generating the Sum from $x1$, $x2$, $x3$, and $x4$. The elimination of the Cout signal and simplification of the carry-generation network reduce the critical computation path and switching activity. The figure 4 truth table shown below the architecture verifies the outputs of S, C, and Cout for different combinations of the five input signals. It is used to evaluate the functional behavior and approximation introduced by each compressors type. Although approximate compressors may produce errors for some input combinations, the resulting error can be controlled and tolerated in applications such as image processing, multimedia processing, machine learning, and DSP, where exact arithmetic is not always necessary. These optimized compressors can therefore be integrated into the partial-product reduction stage of an approximate multiplier. By replacing conventional 4:2 compressors with the proposed low-complexity designs, the overall multiplier can achieve significant improvements in power efficiency, area utilization, operating speed, and energy-delay product, while maintaining acceptable computational accuracy. Hence, the proposed approximate 4:2 compressor forms an important optimization technique for developing low-power and high- performance approximate multipliers for digital signal processing systems. The optimized compressor is integrated into a other tree-based multiplier architecture to reduce the number of partial-product rows efficiently. During multiplication, the input operands first generate partial products; these products are then compressed through successive reduction stages, and the remaining Sum and Carry vectors are processed by the final addition stage. By reducing the number of logic levels and switching nodes, the architecture can shorten the critical path and improve operating speed while lowering dynamic power consumption, importantly, approximation does not mean uncontrolled error: the compressor can be designed so that errors remain within an acceptable range for image processing, multimedia, machine learning, convention, filtering, and other error-tolerant DSP applications. Research has demonstrated that optimized approximate 4:2 compressors can provide substantial improvements in hardware metrics; for example, one published design reported reductions of 35% in power, 36% in delay, and 17% in area compared with an exact multiplier. A further advanced techniques is error balancing, in which the approximate compressor is designed to generate both positive and negative errors so that errors from different compressor stages can partially cancel each other. This approach can improve output accuracy without requiring a large increase in hardware complexity. One reported unbiased approximate 4:2 compressor-based multiplier achieved reductions of 22% in delay, 28% in power, and 26% in area, with approximately 39% improvement in PDP and 46% improvement in EDP for one design. More recent research is also exploring technologies such as Fanfest and CNTFET for further reduction of power and delay in approximate compressor-based multipliers. Overall, the proposed architecture can be considered a low-power, area-efficient and high-speed multiplier design in which accuracy is treated as an additional design parameter rather than an absolute requirement. The major contribution is the optimization of the 4:2 compressor logic and its effective integration into the partial-product reduction network. This makes the architecture particularly suitable for next-generation DSP processors, image and video processing hardware, edge-computing devices, machine-learning accelerators, and energy-constrained VLSI systems, where a small numerical error can be tolerated in exchange for significant improvements in energy efficiency and computational performance. The approximation mainly simplifies the carry-generation mechanism and, in some configurations, eliminates the conventional carry-out path by setting Cout = 0.

This reduces the number of logic gates, transistor count, switching activity, critical-path delay, and silicon area. Although approximation introduces a small amount of computational error, it can provide substantial improvements in power efficiency, speed, and hardware utilization. Such compressors are particularly suitable for error-tolerant applications such as image processing, multimedia systems, machine learning, computer vision, and digital signal processing.

designs, the conventional carry-out path is simplified by setting Cout = 0, thereby reducing the number of logic gates and internal switching activities. This simplification helps decrease power consumption, transistor count, silicon area, and propagation delay, making the compressor suitable for energy-efficient digital signal processing applications. The four compressor types employ different approximate logic configurations to obtain a suitable trade-off between accuracy and hardware efficiency. Type 1 and Type 2 use simplified logic structures to generate the Sum and Carry outputs with reduced circuit complexity. Type 3 generates the Sum according to the combined input condition, represented as $Sum = x1 + x2 + x3 + x4 + C_{in}$, while its carry generation is simplified using the intermediate combination.

4. COMPREHENSIVE ANALYSIS & RESULTS

A 4:2 compressor (Type 1) is an optimized combination block engineered to sum five binary inputs-four primary partial product bits (x1, x2, x3, x4) and one incoming carry bit (Cin) from a lower-significant stage. It outputs three binary values: Sum, Carry, and an outgoing carry bit (Cout)destined for the adjacent higher-significant column. In conventional implementations, a Type 1 4:2 compressor cascades to standard 3:2 compressors (Full Adders) in series. The primary advantage of these structures over simple full-adder chains lies in its internal carry routing. The intermediate carry Cout generated by the first 3:2 block depends solely on the primary input signals (x1 through x4) rather than Cin. By remoting Cin from the Cout calculation, horizontal carry propagation through the tree stage is completely eliminated, heading on overall system delay that scales efficiently from a transistor – level dynamic perspectives, standard CMOS logic implementations of compressor stages often suffer from high switching activity. And intermediate glitches due to XOR gate propagation delays., Type 1 architectures frequently optimize this by utilizing transmission-gate logic, pass- transistor logic (PTL), or XOR -XNOR dual-rail structures. These circuit – level techniques minimized the critical path delay down to three XOR-XNOR gate delays for the SUM signal and to gate delays for the carry output, achieving uniform signal arrival times that reduced

dynamic power consumption. The simulation interface in the waveform trace reveals functional verification across conjugative test vectors. By observing signals such as SUM, CARRY, and Cout transitioning along side primary inputs, the simulation checks both static logic correctness (Truth-table integrity) and dynamic timing setup / hold conditions modern electronic design automation (EDA flows execute such unit- level simulations to ensure no static or dynamic hazards occur before synthesis into physical library cells in full – scale digital arithmetic in engines - such as Wallace tree or Dadda tree multi-bit multipliers-these compressors serve has the primary workhorse for partial product accumulation. Instead of adding bits row-by-row linearly, array trees use 4:2 compressor matrices to reduce partial product height exponentially per stage. This accelerates high-performance.

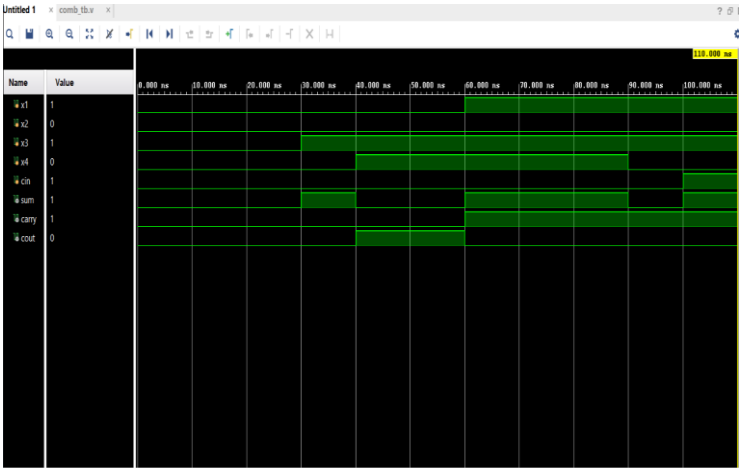

Figure 5: Simulation waveform of 4:2 compressor type 1

compressor used in the approximate multiplier. The waveform displays the variation of the input and output signals with respect to simulation time. When the input combinations change, the compressor output signals-such as Sum and Carry-change accordingly. This confirms that the designed 4:2 compressor responds correctly to different input conditions. In the proposed approximate multiplier, the compressor reduces the number of partial-product reduction stages, which helps decrease power consumption, area, and propagation delay. The waveform therefore provides evidence that the optimized compressor can performs the required arithmetic operation with acceptable approximation error. This makes the architecture suitable for low-power digital signal-processing applications, including image processing, multimedia processing, and other error-tolerant DSP systems. The simulation results demonstrate that the optimized 4:2 compressor-based approximate multiplier can effectively perform multiplication operations while reducing hardware complexity and computational requirements. By using an optimized compressor structure, the multiplier can reduce the number of logic elements, switching activity, and partial-product reduction stages, which contributes to lower power consumption, reduced circuit area, and improved propagation delay. This makes the proposed architecture particularly beneficial for energy-efficient digital signal processing (DSP) systems, where multiplication is a frequently repeated operation. In image processing, the small approximation error can be tolerated in applications such as filtering, image enhancement, and convolution.

The type2 waveform illustrates the functional behavior of the optimized 4:2 compressor employed in the proposed approximate multiplier for low-power digital signal processing. The simulation is carried out over a time range of 0-110ns, during which the input signals x1, x2, x3, x4, and Cin are systematically varied to evaluate the corresponding Sum, Carry, and Cout outputs. Initially, when the input signals are inactive. The compressor produces stable low outputs.

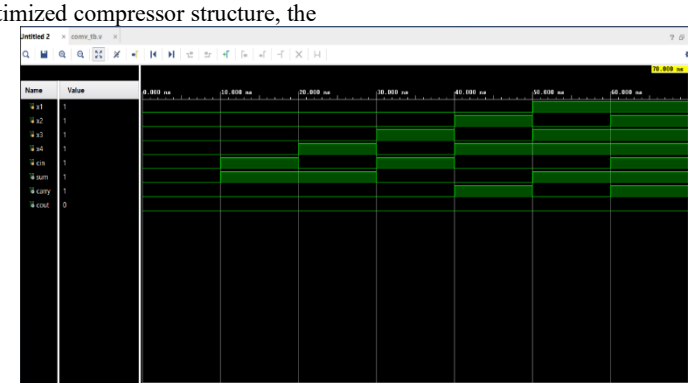

Figure 6: Simulation waveform of 4:2 compressor type 2

As different combinations pf partial-product inputs are applied, the waveform demonstrates the generation of appropriate sum and carry signals, confirming the compressor's ability to efficiently combine multiple input bits. In particular, when several inputs become active simultaneously, the Sum and Carry outputs change according to the approximate compression logic, demonstrating effective reduction of partial-product bits. The transition of Cin further verifies the compressors response to carry propagation and its ability to maintain proper output behavior under different input conditions. Overall, the waveform confirms the functional operation and timing response of type2, indicating that the optimized compressor can reduced logic complexity and switching activity while maintaining acceptable computational

accuracy.

This reduction in hardware complexity contributes to lower power consumption, reduced propagation delay, and improved area efficiency, making the proposed approximate multiplier suitable for energy-efficient DSP architectures, image and video processing, multimedia applications, machine-learning accelerators, and other error-tolerant VLSI systems where power and performance are more critical than exact arithmetic accuracy.

The Cin transition further demonstrates the compressor's ability to handle carry propagation under different input conditions while maintaining the expected output response. The observed waveform transistors occur according to the applied input patterns, indicating correct functional and timing behavior of type2. Overall, the simulation confirms that the optimized 4:2 compressor can reduce logic complexity, switch activity, propagation delay, and power consumption while maintaining acceptable computational accuracy. Therefore, type2 is well suited for low-power approximate multipliers and energy-efficient DSP applications, particularly where a controlled amount of computational error can be traded for improved hardware efficiency. By employing an optimized 4:2 compressor structure, the design reduces the complexity of partial-product reduction can contribute to lower power consumption.

The type-3 optimized 4:2 compressor-based approximate multiplier, designed for low-power digital signal-processing applications. The simulation includes four primary inputs x1, x2, x3, and x4, an additional carry-in (Cin), and three outputs: sum, carry, and carry-out (Cout). The waveform verifies the logical behavior of the compressor by applying different combinations of input signals over the 0-110ns simulation interval.

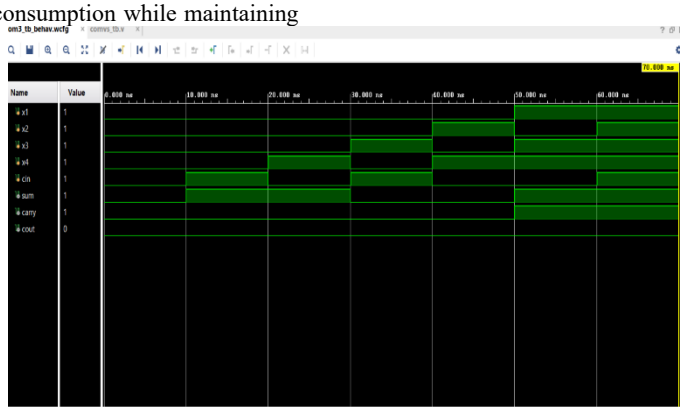

Figure 7: Simulation waveform of 4:2 compressor type 3

The green transitions indicate changes in binary logic levels, while the stable regions demonstrate the corresponding steady-state output responses. In the initial portion of the waveforms the input signals remain predominantly at logic-low levels, establishing the initial operating condition of the compressor. Around 30ns, the x3 input becomes logic 1, producing a corresponding transition in the sum output.

This behavior demonstrates the XOR-like contribution of the compressors approximate sum-generation logic. Because only a limited number of input bits are active during this interval, no significant carry propagation is required. This confirms that the type-3 architecture can generate the required output with reduced combinational complexity. At approximate 40 ns, the x4 input transitions to logic 1 while x3 remains active. With two input bits asserted simultaneously, the compressor generates a carry-related response, which is visible through the Cout transition. This interval is particularly important because it demonstrates how the compressor handles multiple active partial-product bits while separating the arithmetic information into sum and carry components.

Such compression of partial products is essential in multiplier architecture. Because it reduces the number of addition stages required the number of addition stages required before the final product is generated. At around 60ns, x1 becomes logic1 while x3 and x4 remain active. Consequently, the waveforms shown changes in both the sum and carry outputs, this represents a higher-density input condition in which several partial-product bits are present simultaneously, the type3 approximate compressor processes these signals using simplified structure.

This reduction in logic complexity can decrease transistor count, switching activity propagation delay, and power consumption making the architecture suitable for energy-constrained DSP systems. Between approximately 60ns and 90ns, the waveform maintains a stable combination of active inputs, and the sum and carry output remain stable. This stable region demonstrate that the compressor reaches a valid steady-state response without unnecessary transitions.

The type-4 implementation of a approximate of an optimized 4:2 compressor, which is used as a fundamental building blocks in the proposed approximate multiplier for low-power digital signal processing. The simulation waveform contains for primary input signals, x1, x2, x3, and x4, along with the carry-in (Cin) signal. The corresponding outputs are sum, carry, and carry-out (Cout). The waveform is observed over a simulation period od approximately 0-110ns, demonstrating the functional behavior of the compressor for different combinations of input signals.

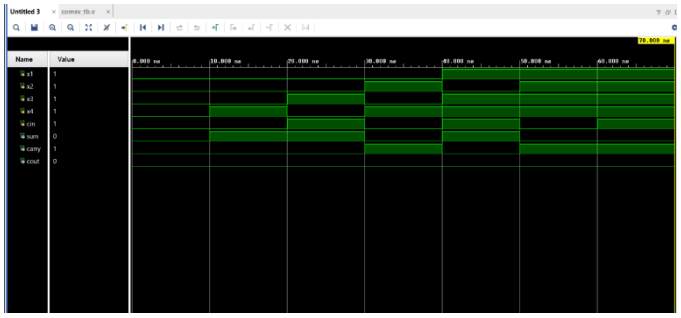

Figure 8: Simulation waveform of 4:2 compressor type 4

In the waveform, the input signals are applied at different time intervals to verify the operation of the type-4 compressor under multiple input combinations. The signal x1, x2, x3, and x4, represent the four partial- product bits that are normally generated during multiplication. The Cin signal represents the carry received from the preceding compressor stage. By changing these inputs at different time intervals, the simulation evaluates whether the compressor can correctly generate the required sum and carry.

The sum output indicates the lower-order results generated by the compressor, while the carry output represents the higher-weight contribution produced from the input combination. In the displayed waveform, the sum and carry signals change according to the applied input patterns, showing that the compressor responds dynamically to variations in the input data. The Cout signal provides the carry information that can be propagated to the next stage of the multiplier. This separation of sum and carry signals is essential for constructing efficient partial-product reduction trees in multiplier architectures. A major objective of the type-4 approximate compressor is to reduce hardware complexity, switching activity, power consumption, and propagation delay compared with an exact compressor. Instead of implementing all possible arithmetic operations.

With conventional full-adder-based logic, an optimized approximate structure simplifies selected logic paths. This reduction in circuit complexity makes the architecture particularly suitable for error-tolerant DSP applications, where a small amount of computational error can be accepted in exchange for significant improvements in power and performance. The waveform also demonstrates the importance of the carry-in signal in determining the final compressor outputs. When the input combination changes, the generated sum and carry values are correspondingly modified, indicating the propagation of arithmetic information through the compressor. The transitions visible around different simulation intervals confirms that the circuit is being tested with several combinations rather than with only a single static input.

The “type-4” structural designation highlights a custom gate configuration turned for target performance matrices such as dynamic power reduction, reduced transistor count, or minimal delay. In approximate computing frame works type – 4 implementations frequently simplified internal XNOR / MUX trees to trade off minor accuracy under rare input conditions in exchange for sustainable savings in chip area and latency.

By integrating 4:2 compressors into Wallace or Dadda tree reduction networks, digital designers significantly compressor the critical path compared to traditional full- adder networks. replacing two stages of conventional 3:2 adders with a signal 4:2 compressor stage reduces total gate delay from 4 XOR gate delays down two or three

This architectural optimization makes type-4 compressors highly effective for real-time digital signal processing, computer vision hardware, and low -power embedded processors. carry signals change according to the applied input patterns, showing that the compressor responds dynamically to variations in the input data. The Cout signal provides the carry information that can be propagated to the next stage of the multiplier. This separation of sum and carry signals is essential for constructing efficient partial-product reduction trees in multiplier architectures. A major objective of the type-4 approximate compressor is to reduce hardware complexity, switching activity, power consumption, and propagation delay compared with an exact compressor.

Instead of implementing all possible arithmetic operations with conventional full-adder-based logic, an optimized approximate structure simplifies selected logic paths. This reduction in circuit complexity makes the architecture particularly suitable for error-tolerant DSP applications, where a small amount of computational error can be accepted in exchange for significant improvements in power and performance. The waveform also demonstrates the importance of the carry-in signal in determining the final compressor outputs. When the input combination changes, the generated sum and carry values are correspondingly modified, indicating the propagation of arithmetic information through the compressor.

The transitions visible around different simulation intervals confirms that the circuit is being tested with several combinations rather than with only a single static input. Conventional exact 4:2 compressors utilize to cascade full adders (FAs) resulting in a critical path delay of 4 XOR. Advanced type -4 architectures alter the underlying logic using specialized XOR-XNOR gates and multiplexers (MUXes) to compte Cout independently of Cin. This eliminates carry propagation delay within the compressor stage, caping the critical path at just three XOR or two XOR. standard CMOS implementations require up to 52 transistors for an exact 4:2 compressors.

Carry signals change according to the applied input patterns, showing that the compressor responds dynamically to variations in the input data. The Cout signal provides the carry information that can be propagated to the next stage of the multiplier. This separation of sum and carry signals is essential for constructing efficient partial-product reduction trees in multiplier architectures. A major objective of the type-4 approximate compressor is to reduce hardware complexity, switching activity, power consumption, and propagation delay compared with an exact compressor.

Instead of implementing all possible arithmetic operations with conventional full-adder-based logic, an optimized approximate structure simplifies selected logic paths. This reduction in circuit complexity makes the architecture particularly suitable

for error-tolerant DSP applications, where a small amount of computational error can be accepted in exchange for significant improvements in power and performance. The waveform also demonstrates the importance of the carry-in signal in determining the final compressor outputs. When the input combination changes, the generated sum and carry values are correspondingly modified, indicating the propagation of arithmetic information through the compressor. The transitions visible around different simulation intervals confirms that the circuit is being tested with several combinations rather than with only a single static input.

5. PERFORMANCE ANALYSIS OF THE PROPOSED COM 1-COM 4 APPROXIMATE 4:2 COMPRESSORS

Table 8: FPGA Resource Utilization of the Proposed COM1-COM4 Approximate 4:2 Compressors

Compressor	Slice LUTs	Slices	LUT as Logic	Bonded IOBs
COM 1	1	1	1	8
COM 2	1	1	1	8
COM 3	1	1	1	8
COM 4	1	1	1	7

Table 8 presents the comparative FPGA resource utilization of the proposed COM1, COM2, COM3, and COM4 approximate 4:2 compressor architectures. The results indicates that COM1, COM2, and COM3 each require one Slice LUT, one Slice, one LUT configured as logic, and eight bonded I/O resources COM4 exhibits the same logic-resource requirement but utilizes seven bonded I/O resources. The identical LUT and Slice utilization across the four architectures indicates that all the proposed compressors achieve a highly compact hardware realization under the evaluated implementation conditions. The utilization of only one LUT as logic demonstrates that the combinational functionality of each compressor can be mapped efficiency onto the available programmable logic resources. The bonded I/O utilization differs slightly for COM4, which requires seven I/O resources compared with eight for COM1-COM3. This different reflects the respective external signal requirements of the synthesized modules and does not directly represent the internal computational complexity of the compressor. The low resource utilization of all four architectures demonstrates their suitability for area-efficient approximate arithmetic implementation. However, resource utilization alone is not sufficient to identify the best compressor. Therefore, the results should be considered together with the timing and power measurements presented in the subsequent sections to establish the overall hardware-efficiently trade-off among COM1-COM4

TABLE 9: Timing Performance of the proposed COM1-COM4 Approximate 4:2 Compressors

Compressor	Logic Delay (ns)	Net Delay (ns)	Total Delay (ns)
COM 1	3,099	2,675	5,774
COM 2	3,099	2,676	5,775
COM 3	3,099	2,675	5,774
COM 4	3,098	2,448	5,546

Table 9 presents the comparative timing performance of the proposed COM1-COM4 approximate 4:2 compressors. The timing characteristics are evaluated in terms of logic delay, net delay, and total propagation delay, thereby providing a quantitative assessment of the temporal efficiency of each compressor architecture.

The total propagation delay is determined by the sum of the internal logic delay and the interconnection net delay:

$$TTotal = TLogic + TNext$$

For COM1, the logic delay is 3.099 ns and the net delay is 2.675ns, resulting in a total delay of 5.774 ns. COM 2 exhibits a logic delay of 3.099 ns and a net delay of 2.676 ns, giving a total delay of 5.775 ns. COM3 achieves a total delay of 5.774 ns. With logic and net delays of 3.099 ns and 2.675 ns, respectively. Among the four architectures, COM4 provides the lowest total propagation delay of 5.546 ns, comprising a logic delay of 3.098 ns and a net delay of 2.448 ns. The lower net-delay contribution of COM 4 results in an overall timing improvement compared with the other proposed compressors. The results indicate that COM1, COM2, and COM3 have closely comparable timing characteristics, whereas COM4 demonstrates a comparatively better temporal response. The reduced propagation delay of COM4 is advantages for high-speed arithmetic operations and can contribute to improved performance when the compressor is incorporated into the partial-product reduction network of the proposed approximate multiplier.

TABLE 10: Power Consumption of the Proposed COM1-COM4 Approximate 4:2 Compressors

Compressor	Total Power (W)	Device Static Power (W)	Static Power (%)	Hierarchical Power (W)	Signal Power (W)
COM 1	1.497	0.254	17	1.243	0.036
COM 2	1.497	0.254	17	1.243	0.036
COM 3	1.497	0.253	17	1.189	0.035
COM 4	1.349	0.252	17	1.096	0.026

Table 10 presents the comparative power characteristics of the proposed COM1-COM4 approximate 4:2 compressors. COM1 and COM2 exhibit a total power consumption of 1.497 W, whereas COM3 reduces the total power to 1.442 W. COM4 achieves the lowest total power consumption of 1.349 W. The device static power decreases slightly from 0.254 W for COM2 to 0.252 W for COM4. Similarly, the hierarchical power decreases from 1.243 W to 1.096 W, while the signal power decreases from 0.036 W to 0.026 W. This reduction indicates a lower estimated power requirement for the later compressor configurations.

The total power can be represented as:

$$PTotal = PStatic + PDynamic$$

Among the evaluated architectures, COM4 provides the most favorable power characteristic, with a total estimated power of

1.349 W. Therefore, COM4 demonstrates improved power efficiency under the evaluated Vivado implementation conditions.

6. CONCLUSION

The proposed optimized 4:2 compressor-based approximate multiplier provides an effective solution for reducing power consumption, area utilization, and computational complexity in digital signal processing applications. By incorporating approximate arithmetic using optimized 4:2 compressor structures, the design achieves a suitable balance between hardware efficiency and computational accuracy. The reduction in the number of logic operations and critical-path complexity contributes to improved power and area performance while maintaining acceptable output accuracy. This makes the proposed multiplier particularly suitable for error-tolerant DSP applications where small arithmetic errors can be tolerated without significantly affecting overall system performance.

The optimized multiplier can be effectively applied in image processing, multimedia processing, machine-learning accelerators, filtering, convolution operations, MAC units, and other low-power VLSI systems. Compared with conventional exact multipliers, the proposed architecture demonstrates the potential for reduced hardware overhead and improved energy efficiency. Therefore, the design provides a promising architecture for next-generation low-power and high-performance digital systems. As a future enhancement, the proposed architecture can be further optimized by exploring different approximation techniques, and error-compensation mechanism to achieves a better trade-off between power, area, delay, and accuracy.

The design can also be extended to higher-bit multipliers such as 16-bit, 32-bit, and 64-bit architectures and integrated into larger DSP data paths. Further improvements can focus on dynamic voltage scaling, power-gating, pipeline optimization, and technology scaling for advanced CMOS technologies. The multiplier can also be evaluated using practical applications such as image filtering, neural-network inference, edge-AI processing, and real-time multimedia systems. Future work may additionally investigate machine-learning-assisted approximation and application-specific error control, enabling the multiplier to dynamically select an appropriate accuracy level according to the requirements of the target applications. Thus, the proposed optimized 4:2 compressor-based approximate multiplier can serve as a strong foundation for developing energy-efficient, area-optimized, and high-performance VLSI systems.

7. REFERENCES

[1] M. Rashidi, H. Saeidi, and A. Afzali-Kusha, “Area-efficient and high-performance approximate multiplier based on approximate 4:2 compressor, “Integrated, the VLSI Journal, vol. 96, pp. 1-10, 2026.

[2] B. Rashidi, “Area-efficient and high-performance approximate multiplier based on approximate 4:2 compressors,” Integration, the VLSI Journal, vol. 109, pp. 102693, 2025.

[3] A. Chakraborty, S. DasandP.Banerjee, “Optimized approximate compressor architectures for Dadda multipliers,” Microelectronics Journal, vol. 146, pp. 106018, 2025.

[4] P. Jaswal, L. Krishna and B. Srinivasu, “Low-power approximate multipliers for neural network applications,” IEEE Access, vol. 13, pp. 32112-32122, 2025.

[5] L. Krishna, S.Bodapati and S. Veeramachaneni, “Approximate signed multipliers for image processing system,” Integration, the VLSI Journal, vol. 110, pp. 115-124,2025.

[6] R. Singh and P. Sharma, “Low-power approximate 4:2 compressor designed for circuits,” IEEE Access, vol. 13, pp. 45876-45886,2025.

[7] S. Patel and R. Shah, “Optimized compressor – based multiplier architectures for DSP applications,” Microelectronics Journal, vol. 147, pp. 106095,2025.

[8] L. Li Y. Jiang and X. Wang, “Energy-efficient approximate multiplier using compressors circuits,” IEEE Access, vol. 12,pp. 24810-24820,20204.

[9] Z. Zuhair, A. Mahmood and M. Tariq, “Design of optimized approximate 4:2 compressors for multiplier circuits,” Integration, the VLSI Journal, vol. 97, pp. 215-224, 2024.

[10] S. Vakili and M. Taheri, “Compact approximate

compressors for efficient Dadda multipliers," Microelectronics Journal, vol. 140, pp. 105831, 2024

[11] A. Bhandari and P. Kumar, "Analysis of approximate compressor architectures for arithmetic circuits," IEEE Access, vol. 12, pp. 32517-32527, 2024.

[12] Z. Ren, X. Liu and J. Chen, "Approximate multipliers for deep neural network accelerators, "IEEE Transaction Circuits and Systems, vol. 71, no. 2, pp. 755-768,2024.

[13] P. Gupta and R. Mehta, "Low-power approximate compressor circuits for multimedia systems," Microelectronics Journal, vol. 141, pp. 105905, 2024.

[14] Y. Zhang and H. Yang, "Probability-based approximate compressor architecture for arithmetic circuits," IEEE Access, vol. 12, pp. 58711-58720, 2024.

[15] H. Yang and X. Li, "Optimized approximate multiplier architectures for VLSI systems," Integrated, the VLSI Journal, vol. 98, pp. 301-309, 2024.

[16] A. Guturuand S. Gupta, "Compressor selection techniques for efficient multiplier design," IEEE Access, vol. 12, pp. 91820-91830, 2024.

[17] A. Rehman and M. Shafique, "Architecture for approximate multipliers with low error and high efficiency," IEEE Access, vol. 11, pp. 22418-22429, 2023.

[18] Y. Jiang, J. Han and F. Lombardi, "Low-power approximate compressors for arithmetic circuits," IEEE Transactions on Circuits and Systems 1, vol. 70, no. 3, pp. 1105-1115, 2023

[19] P. Gupta and R. Kumar, "Low-power approximate multiplier design for signal processing applications," Microelectronics Journal, vol. 135, pp. 105622,2023.

[20] J. ParkandH.Kim, "Approximate multiplier architecture for image processing system," IEEE Access, vol. 11, pp, 76321-76330, 2023.

[21] A. Ali, M. Ahmed and S. Khan, "Area-efficient approximate multipliers for VLSI applications," Integration, the VLSI Journal, vol. 92, pp. 112-121, 2023.

[22] S. Singh and R. Patel, "Compressor-based approximate multiplier design for DSP applications," International Journal of Electronics, vol. 110, no. 7, pp. 1185-1198, 2023

[23] M. Akbari, A. Eslami and M. Pedram, "Approximate multipliers for energy-efficient digital signal processing," IEEE Access, vol. 10, pp. 12567-12578, 2022.

[24] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," IEEE Transactions on Emerging Topics in Computing, vol. 10, no. 1, pp. 1-15, 2022.

[25] A. Momeni, J. Han, P. MontuschiandF. Lombardi, "Design and analysis of approximate compressors for multiplication, "IEEE Transactions Compressors, vol. 71, no. 3, pp. 621-633, 2022.

[26] H. Kim and S. Lee, "High-performance approximate multiplier design for DSP applications," Microelectronics Journal, vol. 122, pp. 105432, 2022.

[27] Y. Jiang, L. Liu and J. Han, "Approximate arithmetic circuits: A survey, characterization and recent applications," IEEE Design & Test, vol. 39, no. 2, pp. 8-21, 2022.

[28] V. Lakshmi, K. Ezhilarasan and S. N, "FPGA Implementation of efficient and low power test pattern generator," 2022 2nd Asian conference on innovation in technology (ASIANCON), Ravet, India, 2022, pp. 1-5, doi: 10.1109/ASIANCON55314. 2022. 9908689.

[29] Bhavyashree H, Dr. K Ezhilarasan, High-speed carry look-ahead decimal adder design: a Verilog and FPGA-Based implementation and comparative study with hybrid and majority-logic decimal adders, computer science journal, volume 16, issue 9 page no: 21-38, 2026. Doi: 10.14118.CSJ.2026.V1619.003 .

[30] Shylaja V, Dr. K Ezhilarasan, Insights of performance enhancement techniques on high-speed computation multiple 4-bit ALU, page no: 427-444. Doi: 20.18001. Jot.2023.V11111.23.2944.