

Original Article

Design and Implementation of Energy Efficient Area Optimized High Speed Approximate Multipliers Using Higher-Order Compressors

Talla Srinivasa Rao¹, Ch Srinivasu², K. Babulu³

^{1,3}Department of Electronics and Communication Engineering,

Jawaharlal Nehru Technological University Gurajada Vizianagaram (JNTU-GV), Andhra Pradesh, India.

²Department of Electronics and Communication Engineering, Raghu Engineering College, Andhra Pradesh, India.

¹Corresponding Author : srinivasarao.t@adityauniversity.in

Received: 13 May 2024

Revised: 11 June 2024

Accepted: 12 July 2024

Published: 27 July 2024

Abstract - Multipliers play an important role in Digital Signal Processing (DSP) applications, but their traditional design has significant limitations, such as high power consumption, huge area requirements, and long critical path delays. While inexact multipliers have emerged as a potential solution, they are best suited to applications where a small variation from absolute accuracy is acceptable. These inexact multipliers provide significant power savings and reduced area requirements while retaining acceptable levels of precision. Approximation is introduced at both the compressor and multiplier levels, resulting in significant reductions in area and power use. This study proposes an efficient approach to multiplier design that uses higher-order compressors rather than standard 4:2 compressors. The aim is to optimize hardware usage and power consumption. We propose five types of approximate compressors (4:2, 5:2, 6:2, 7:2, and 8:2) and develop four versions of the 8X8 approximate multiplier. These versions use approximation at various points of partial products with either exact full adders or Carry Skip Adders (CSKA) to improve efficiency when adding partial products at the end. Our innovations, which include two stages of partial product integration, result in significant reductions in power and hardware consumption, with average power savings of 62% and hardware reductions of up to 48%. The article also includes a thorough comparison of power, area, latency, and error analysis, emphasizing advances in hardware efficiency.

Keywords - Approximate Multiplier Design, Carry Skip Adder, Compressor optimization, Digital Signal Processing, Higher-Order Compressors.

1. Introduction

Multipliers hold a pivotal role in various application domains, including Digital Signal Processing (DSP), computer vision, multimedia processing, image recognition, and artificial intelligence. These applications often necessitate a substantial number of multiplication operations, leading to a significant power drain. High power utilization poses a formidable challenge, particularly for mobile devices, where energy efficiency is paramount. In response to this challenge, numerous research endeavors have sought to mitigate the power utilization of multiplier circuits. One promising strategy to achieve this is by introducing approximation in multiplication operations. This approach is particularly relevant in cases where the applications in question can tolerate a certain degree of error, typically those related to human sensory perception. Given the inherent limitations in human sensory perception, such as restricted visual or auditory ranges, the imperative for pinpoint accuracy in computation results is relaxed. In these contexts, inexact

multiplication emerges as a valuable means to conserve power while still delivering outcomes that align with the human perceptual thresholds. Inexact multipliers make a trade-off by sacrificing a degree of precision to gain advantages in terms of cell area reduction, latency minimization, and power utilization reduction. These inexact multipliers can be classified into two main categories. One of them focuses on controlling the timing of the multiplier, and this control is often achieved through adjustable voltage scaling. By applying a lower voltage to the multiplier, the critical path's delay increases.

Consequently, when there is a breach in the timing path, errors emerge, leading to the generation of inexact results. The second type of inexact multipliers involves modifying the functional behavior of multipliers themselves. This entails redesigning conventional accurate multiplier circuits, such as the Wallace Tree Multiplier and the Dadda Tree Multiplier. Notably, many prior research efforts have centered on the



adaptation of inaccurate $m:n$ compressors, where 'm' represents the number of inputs, and 'n' signifies the number of outputs. These modified compressors play a pivotal role in the multiplication process by compressing the partial products. This is particularly significant because the compression of partial products typically accounts for a major portion of the multiplier's energy consumption and contributes to prolonged path delays. In approximate multiplier design, researchers are currently exploring and using 4:2 or smaller approximate compressors across different stages of reduction. However, there is still room for improvement in multiplier circuitry performance. This work addresses this need by using higher-order compressors rather than the standard 4:2 or lower ones. The goal is to reduce hardware utilization and power consumption, resulting in even greater efficiency and performance in multiplier design. We are also working on establishing the best location of the partial product order (from LSB to MSB) for introducing approximation compressors to combine partial products and improve performance. This entails using two designs with distinct layouts of partial product orders. The primary contributions of our study can be stated as follows:

1. Five alternative approximate compressors are proposed as fundamental construction pieces, with sizes ranging from 4:2 to 8:2.
2. Four variations of an 8x8 approximation multiplier are created utilizing these fundamental building elements.
 - Version I introduce approximation at the 7th position from the Least Significant Position (LSP) of the partial product order utilizing an exact full adder during the last step of partial product addition.
 - Version II introduces approximation at the 9th place from the Least Significant Position (LSP) of a partial product order utilizing an accurate full Adder during the final stage of partial product addition.
 - Version III introduces approximation at the 7th place from the Least Significant Position (LSP) of a partial product order utilizing the Carry Skip Adder during the last stage of partial product addition.
 - Version IV adds approximation at the 9th place from the Least Significant Position (LSP) of a partial product order utilizing the Carry Skip Adder during the last stage of addition.

All variants were created with only two phases of partial product mixing to improve speed. In Versions I and II, precise full adders are used to add partial products in the second stage, whereas Carry Skip Adder (CSKA) is utilized in Versions III and IV to reduce the latency. When compared to prior works, our technique demonstrates substantial advantages. The average power consumption is reduced by 49%, with exceptional examples showing an astonishing reduction of up to 62%. The average hardware utilization has been lowered by 21%, with some instances showing a fall of up to 48%, indicating a significant improvement in hardware efficiency.

The remainder of this paper is structured as follows: Section II offers a comprehensive review of relevant literature, providing a contextual backdrop for our research that includes the design, implementation, and evaluation of inexact computing components and Carry Skip Adder (CSKA). Focusing on inexact multipliers, 4:2 compressors, and performance measures, this discussion establishes the current landscape, setting the stage for our contributions to advancing the field of inexact computing. Section III presents our innovative approach to designing higher-order exact and inexact compressors. Four versions of 8X8 approximate multipliers are designed in Section IV. Section V conducts a comprehensive comparison across multiple dimensions, including Power, area, delay, and error analysis. Section VI describes conclusions based on our findings and contributions.

2. Related Work

In this section, a comprehensive overview of the basic design elements of the inexact multiplier is provided and detailing the roles of exact and inexact compressors and the carry skip adder are discussed. By leveraging these components, the inexact multiplier achieves an optimal balance between performance, efficiency, and resource utilization, making it a valuable innovation in the field of digital arithmetic design.

2.1. Literature Review

Farshchi et al. introduce a new approximate multiplier for low-power digital signal processing that uses the Broken-Array Multiplier method to reduce power consumption by up to 58% while slightly decreasing output accuracy, as demonstrated by simulations and comparisons with accurate multipliers and other approximate designs [1]. Jiang et al. provide a review of Approximate Arithmetic Circuits, which gives a complete evaluation of several approximation strategies for improving performance and energy economy while reducing accuracy loss. It focuses on the optimization strategies for adders, multipliers, and divisions under various design constraints, demonstrating how error rates affect computational correctness in a variety of applications [2]. Momeni et al. discuss the design and analysis of approximate compressors for multiplication, with a focus on inexact computing for digital processing at nanometric scales, and present two novel approximate 4-2 compressors and their application in inexact multiplication schemes, as well as extensive simulation results and image processing applications [3]. Zervakis et al. present a novel hardware-level approximation technique, called partial product perforation, for designing efficient approximate multiplication circuits, demonstrating significant reductions in power consumption, area, and delay while maintaining acceptable error levels, and evaluates its effectiveness in real-world applications from the image processing and data analytics domains [4]. Venkatachalam et al. describe a novel design strategy for approximate multipliers that saves significant power and area while maintaining higher precision than previous designs,

with applications in error-tolerant multimedia signal processing and data mining [5]. Li et al. present a comprehensive assessment of approximate computing approaches, emphasizing the opportunities, constraints, and future potential of applying approximation tactics to maximize performance and energy efficiency in computing systems [6]. Ansari et al. introduce low-power approximation multipliers based on encoded partial products and approximate compressors, which show enhanced accuracy-performance trade-offs in a variety of applications, including picture sharpening, JPEG compression, and MIMO communication systems [7].

Sabetzadeh et al. offer an ultra-efficient imprecise 4:2 compressor and multiplier based on majority logic for approximate computation, with considerable advantages in transistor count, power consumption, and delay over earlier solutions [8]. Strollo et al. provide a comprehensive analysis and comparison of approximation 4-2 compressors for low-power approximate multipliers, emphasizing the trade-offs between power, precision, and error metrics for building efficient digital circuits [9].

Edavoor et al. discuss the design and analysis of two approximation compressors with reduced area, delay, and power and their implementation in 8×8 and 16×16 Dadda multipliers [10]. Ullah et al. discuss the need for high-performance and resource-efficient soft multiplier IP cores for FPGAs, and they provide generic area-optimized, low-latency accurate, and approximate softcore multiplier architectures [11]. Ahmadinejad et al. suggested energy- and quality-efficient approximation multipliers based on new approximate compressors that used NAND gates to generate complementing partial products and evaluated the proposed designs in neural networks and image processing applications [12]. Prashanth et al. show that utilizing approximate computing to construct multiplier blocks with AND-OR re-coded compressors and fast adders dramatically improves power, footprint, and delay. This method is especially appropriate for error-tolerant applications where tiny mistakes are tolerated.

Furthermore, he emphasizes the relevance of VLSI implementation of arithmetic functions for hardware in picture and digital signal processing, particularly in autonomous applications [13]. Munawar et al. present a modified-Dadda algorithm-based multiplier that uses a novel half-adder-based carry-select adder with a binary to excess-1 converter and an improved RCA. In simulations across various technologies and frequencies, it demonstrates superior speed, power efficiency, and transistor count, making it a promising solution for low-power, low-cost digital controllers [14]. Minaeifar et al. offer two multipliers, "M00" and "M01," that use error correction techniques and approximate 4:2 compressors to improve energy efficiency. According to evaluations, M00 outperforms in terms of energy consumption

and power-delay product, whereas M01 achieves high precision and is particularly good in image processing tasks such as sharpening, smoothing, and Discrete Cosine Transform [15].

2.2. Exact and Inexact 4:2 Compressors

A precise 4:2 compressor is typically implemented by Chang et al. using two full adders, as illustrated in Figure 1. In a multiplication operation, X_1 to X_4 represent the partial products within the same column, and C_{i-1} denotes the carry-in originating from the previous column's compressor. This accurate 4:2 compressor generates three outputs: C_i , carry, and sum. [16].

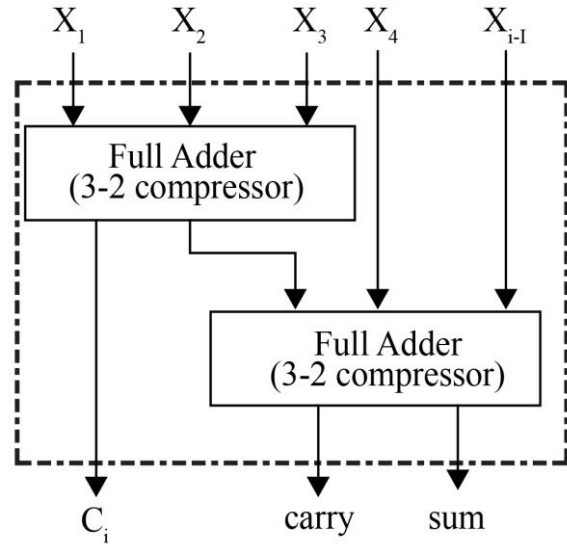


Fig. 1 Exact 4:2 compressor

In contrast, Figure 2 depicts the block diagram for most inexact 4:2 compressors. In comparison to the precise 4:2 compressor in Figure 1, the inexact 4:2 compressor omits the use of C_{i-1} and does not produce C_i . As a result, it has four inputs and two outputs, significantly simplifying the partial product compression process provided by Momeni et al. and Akbari et al. [3,17]. However, it's important to note that errors can occur when all four inputs are set to 1, resulting in a binary output of '100', which necessitates at least three output ports. The efficiency of a circuit design using approximate 4:2 compressors heavily depends on the computational logic within these compressors.

Researchers have proposed various designs for approximate 4:2 compressors, each with its unique strengths and weaknesses, reflecting a trade-off between accuracy and resource utilization. These aforementioned inexact 4:2 compressors collectively offered a means to rapidly reduce partial products while delivering lower timing delays, reduced cell area and decreased power utilization compared to the precise 4:2 compressor. The choice of an approximate 4:2 compressor design depends on the specific requirements of the

application. High-speed, low-power applications may benefit from simpler, less accurate designs, while applications requiring precise calculations may need more complex and accurate compressors. The ongoing research in this field continues to explore new ways to balance these trade-offs, striving to create designs that offer the best possible combination of speed, power efficiency, and accuracy. Recent literature studies investigate compressors of sizes 4:2, 5:3, and 6:3, which achieve satisfactory levels of hardware reduction. However, further improvements may be attained by increasing the compressor ratio to 5:2, 6:2, 7:2, and 8:2, as is done in the proposed work.

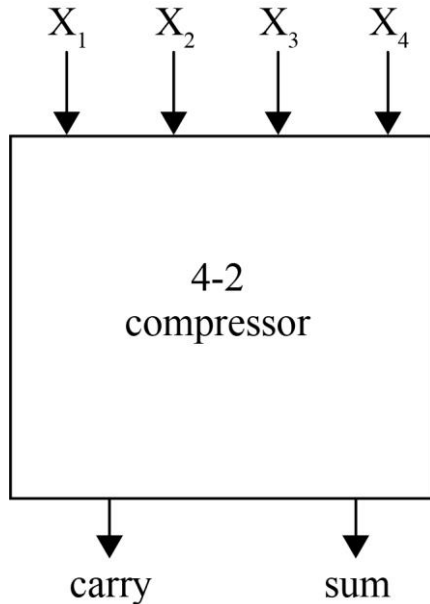


Fig. 2 Approximate 4:2 compressor

2.3. Design of Inexact Multiplier

An approximate multiplier is a specialized digital circuit designed to perform multiplication with controlled inaccuracies, aiming to optimize power, speed, and area efficiency in digital signal processing applications. The approximate multiplier architecture typically involves novel techniques in operand encoding, approximate partial product generation, and error-tolerant accumulation [10].

The multiplication procedure entails creating partial products by multiplying each bit of the multiplicand by each bit of the multiplier, stacking these partial products into two rows, and then computing the final binary result with a carry propagate adder, as shown in Figure 3. Optimizing the second step is significantly responsible for the multiplier's performance. Its design incorporates approximation compressors, and the effect of doing so is measured using an 8x8 unsigned Dadda multiplier [17].

Conventional Dadda multipliers make use of precise 4:2 compressors; however, since the approximate compressors do

not have Cin and Cout connections, the design has to be changed to produce an approximate multiplier. Figure 2 shows the reduction circuitry, which consists of half-adders, full adders, and approximate 4:2 compressors. Each dot in the circuitry represents a partial product from AND gates.

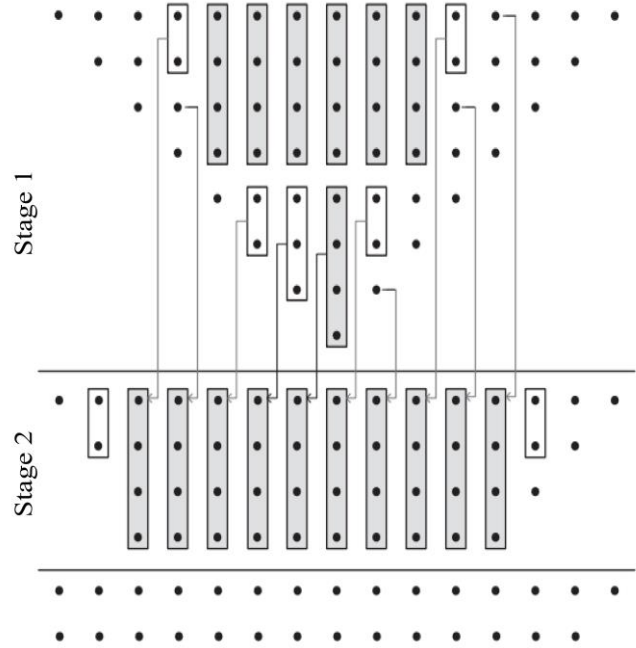


Fig. 3 Schematic diagram of 8X8 approximate multiplier

2.4. Carry Skip Adder (CSKA)

A 16-bit carry skip adder, also referred to as a carry-select adder, enhances the efficiency of binary addition in digital circuit designs by strategically bypassing carry propagation delays inherent in a conventional ripple carry adders. It operates across four stages, each handling four bits, for a total of 16 pairs of input bits (A and B) representing the binary numbers to be added. Within each group, a basic binary adder computes the sum of the corresponding bits along with the carry-in from the prior stage.

Generate (G) and Propagate (P) signals are then derived to indicate the possibility of carry-out and carry-in, respectively, for each group. Through careful assessment of these signals, the carry-skip mechanism efficiently determines whether to propagate or skip the carry within each group. A carry-select circuit subsequently chooses the appropriate carry values for propagation to the subsequent stage.

Finally, the last stage aggregates the bits from the last group with the carry-out from the preceding stage to yield the final sum. This approach significantly accelerates addition by circumventing unnecessary carry propagation, thereby minimizing overall propagation delay and enhancing computational speed compared to traditional ripple carry adders.

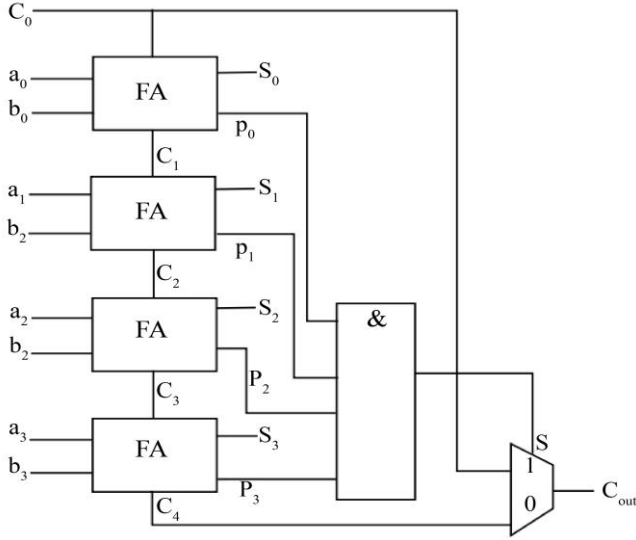


Fig. 4 Carry Skip Adder (CSKA)

3. Proposed Compressors Design

In approximate multiplier design, researchers are exploring 4:2 or smaller compressors. This paper introduces higher-order compressors (5:2, 6:2, 7:2, and 8:2) to reduce hardware usage and power consumption, enhancing efficiency. We focus on optimizing partial product positions (from LSB to MSB) using two designs with different arrangements. Version I introduce approximation at the 7th position from the LSP, while Version II does so at the 9th position. Both versions use two stages of partial product combination for speed enhancement and employ a Carry Skip Adder (CSKA) to reduce delay in the second stage. This paper introduces higher-order compressors (5:2, 6:2, 7:2, and 8:2), along with our proposed inexact 4:2 compressors, to reduce hardware usage and power consumption, thereby enhancing efficiency. The exact designs utilize exact full adders, exact half adders, and exact 4:2 compressors as building blocks. In contrast, the approximate designs employ approximate adders, approximate half adders, and approximate 4:2 compressors, offering a balance between performance and resource optimization.

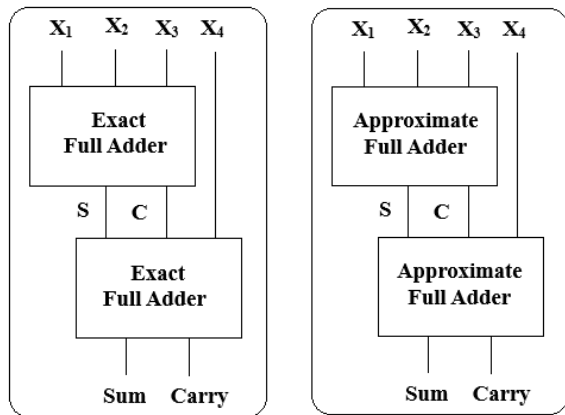


Fig. 5(a) Exact 4:2 Compressor Fig. 5(b) Inexact 4:2 Compressor

3.1. Exact and Inaxt 4:2 Compressor

Let us begin our discussion with the precise exact 4:2 compressor and the inexact 4:2 compressor, as seen in Figure 5. The exact 4:2 compressor was created using two exact full adders, while the approximate 4:2 compressor was designed using two approximate full adders.

An approximate full adder functions similarly to an exact full adder but with a difference in the sum expression where an OR gate replaces the XOR gate. Additionally, in the carry expression, the input carry (C_{in}) is directly assigned to the carry output. This modification changes the behavior of the sum output and the carry output compared to a conventional full adder.

$$\begin{aligned} Sum &= A \text{ or } B \\ C_{out} &= C_{in} \end{aligned}$$

3.2. Exact and Inaxt 5:2 Compressor

Figure 6 depicts the exact 5:2 compressor designed using one exact 4:2 compressor and one exact full adder, as well as the approximate 5:2 compressor designed with one approximate 4:2 compressor and one approximate full adder.

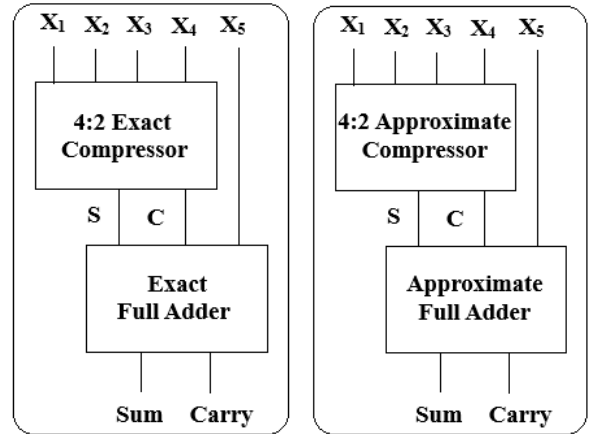


Fig. 6(a) Exact 5:2 Compressor Fig. 6(b) Inexact 5:2 Compressor

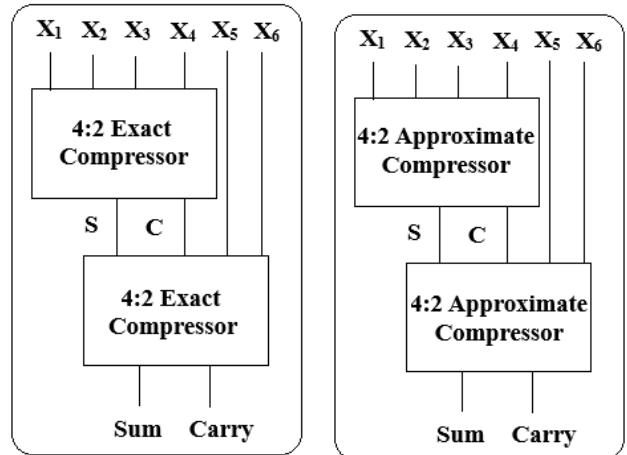


Fig. 7(a) Exact 6:2 Compressor Fig. 7(b) Inexact 6:2 Compressor

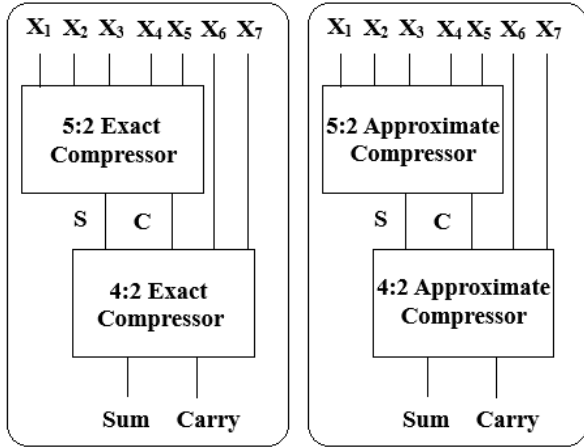


Fig. 8(a) Exact 7:2 Compressor Fig. 8(b) Inexact 7:2 Compressor

3.3. Exact and Inexact 6:2 Compressor

The exact 6:2 compressor was designed using two exact 4:2 compressors, whereas the approximate 6:2 compressor was designed using two approximate 4:2 compressors and one approximate full adder, as illustrated in Figure 7. Two compressors of 6:2 and 4:2 are sufficient, and no need for a full adder.

3.4. Exact and Inexact 7:2 Compressor

The exact 7:2 compressor was created by combining one exact 5:2 compressor and one exact 4:2 compressor, whereas the approximate 7:2 compressor was created by combining one approximate 5:2 compressor and one approximate 4:2 compressor, as illustrated in Figure 8.

3.5. Exact and Inexact 8:2 Compressor

The exact 8:2 compressor was designed by using one exact 6:2 compressor and one exact 4:2 compressor, whereas the approximate 8:2 compressor was designed by using one approximate 6:2 compressor and one approximate 4:2 compressor as shown in Figure 9.

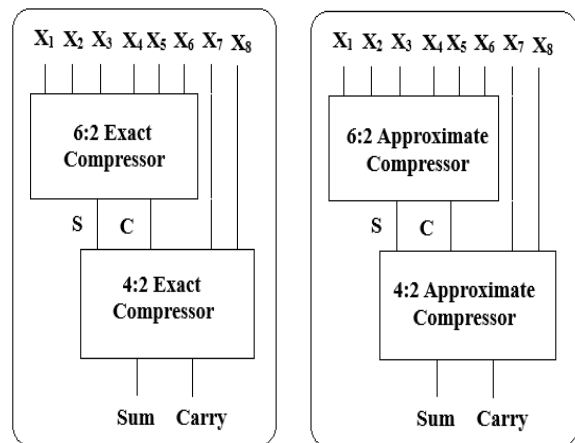


Fig. 9(a) Exact 8:2 Compressor Fig. 9(b) Inexact 8:2 Compressor

4. Proposed Approximate Multiplier

4.1. Version-I: Introducing Approximation at the 7th Position from the Least Significant Position (LSP) of the Partial Product Order by Using Exact Full Adder at the Last Stage Addition of Partial Products (Version-I-7th-FA)

In our innovative multiplier design, we streamline the computation process, achieving a notable reduction in the number of stages required to generate the output compared to conventional methods. Remarkably, our design necessitates only two stages to compute the final output. In traditional Dadda multipliers, five stages are typically employed for computation. Conversely, in approximate multipliers utilizing 4:2 compressors, three stages are typically needed. However, our design breaks new ground by condensing the process even further. Moreover, we strategically leverage approximate adders and compressors in the least significant part of our design, optimizing for speed, area, and power efficiency. Conversely, in the most significant part, we capitalize on exact adders and compressors, ensuring precision where it is most critical. By implementing this approach, we not only enhance computational speed but also significantly reduce the overall area and power consumption of the multiplier circuit. Our design represents a breakthrough in multiplier architecture, offering a compelling balance between performance and resource utilization. Four versions of an 8x8 approximate multiplier were developed using half adders, full adders, and higher order compressors like 4:2,5:2,6:2,7:2,8:2 compressors (both Exact and Approximate. Version I & III introduce approximation at the 7th position from the Least Significant Position (LSP) of the partial product order, while Version II & III do so at the 9th position. All the versions are designed with only two stages of partial product combination to enhance speed and utilize a Carry Skip Adder to reduce delay in the second stage. In Version-I approximate multiplier, approximate adders and compressors are used in the Lower Significant Position (LSP) up to the 7th column to save on hardware and power, as shown in Figure 10. For the Most Significant Part (MSP), exact adders and compressors are employed to maintain accuracy. Specifically, for adding two partial products, a half adder is used; for three partial products, a full adder is used; and for four partial products, a 4:2 compressor is utilized. This pattern continues up to 8:2 compressors for designing an 8-bit multiplier. In the final stage, exact full adders are used to combine the partial products to ensure precision in the result.

4.2. Version-II: Introducing Approximation at the 9th Position from the Least Significant Position (LSP) of the Partial Product Order by Using EXACT FULL ADDER at The Last Stage Addition of Partial Products (Version-I-9th-FA)

This design is identical to Version I, but approximate adders and compressors are used up to the 9th column instead of the 7th column, as shown in Figure 11. The usage of approximation is increased to decrease the hardware utilization and power consumption.

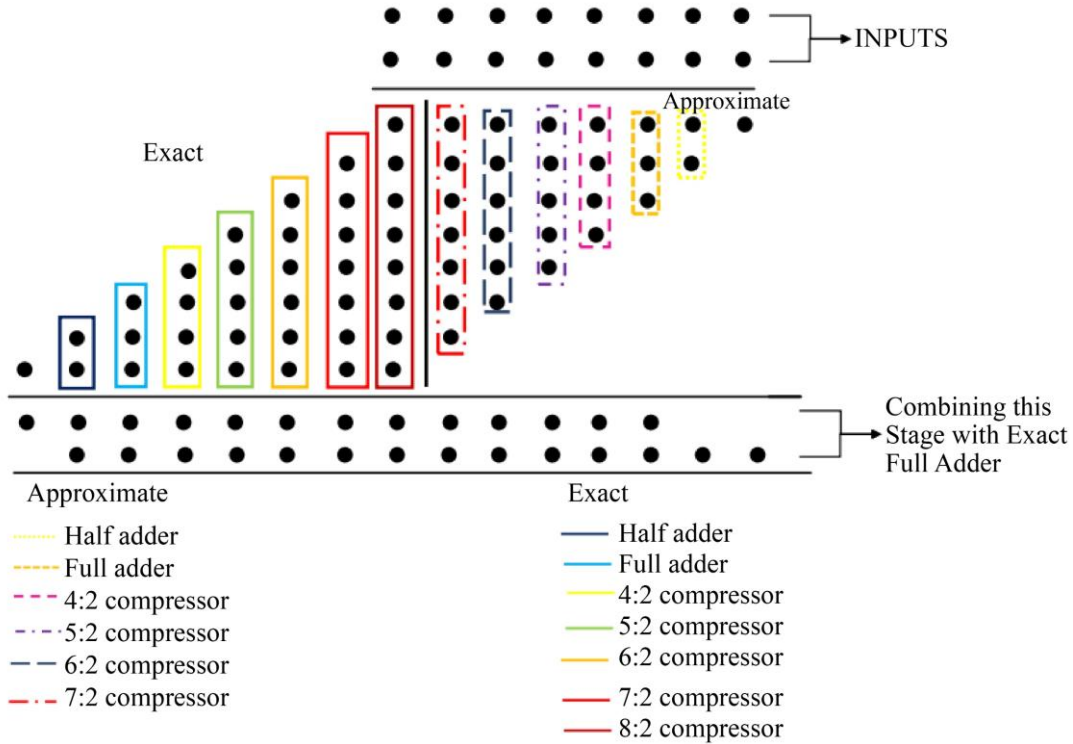


Fig. 10 Schematic representation of 8X8 approximate multiplier (version-I)

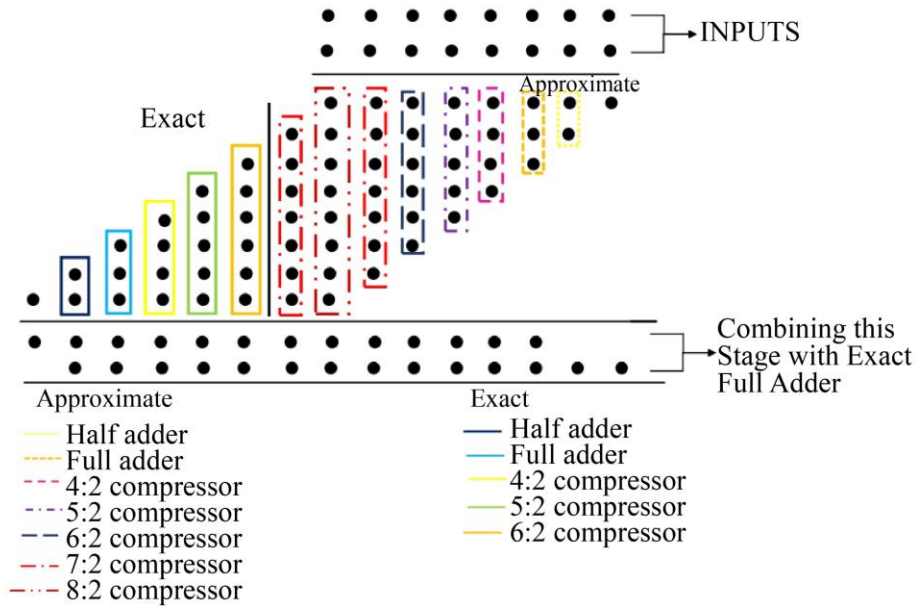


Fig. 11 Schematic representation of 8X8 approximate multiplier (version-II)

4.3. Version-III: Introducing Approximation at the 7th Position from the Least Significant Position (LSP) of the Partial Product Order by Using Carry Skip Adder at the Last Stage Addition of Partial Products (Version-I-7th-CSKA)

The proposed design adheres to the operational principles outlined in Version I, as shown in Figure 12, with a notable enhancement introduced in the final stage. Here, a 16-bit carry-skip adder is integrated to elevate operational speed.

While retaining the functionality of the 8-bit design, this adaptation capitalizes on the improved efficiency and rapidity offered by the 16-bit carry-skip adder, thereby expediting computations during the final processing phase. By maintaining the fundamental methodology of the approximate multiplier, this modification optimizes both speed and efficiency through the strategic incorporation of advanced adder architecture.

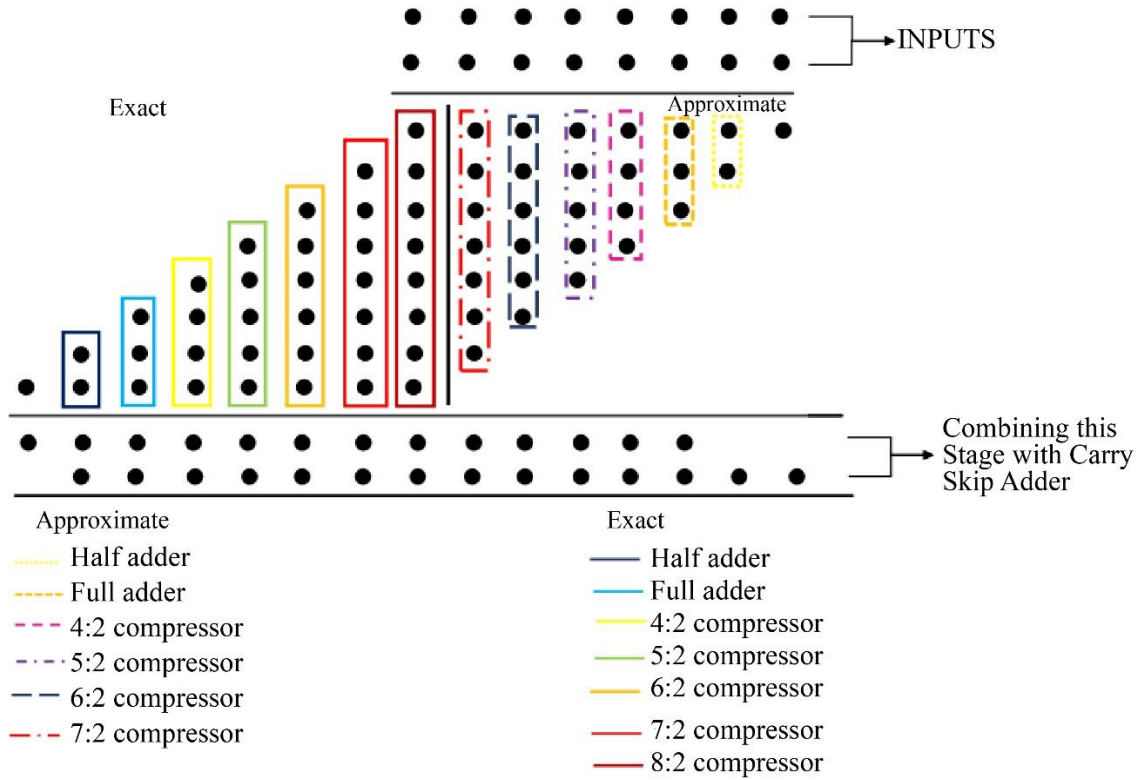


Fig. 12 Schematic representation of 8X8 approximate multiplier (version-III)

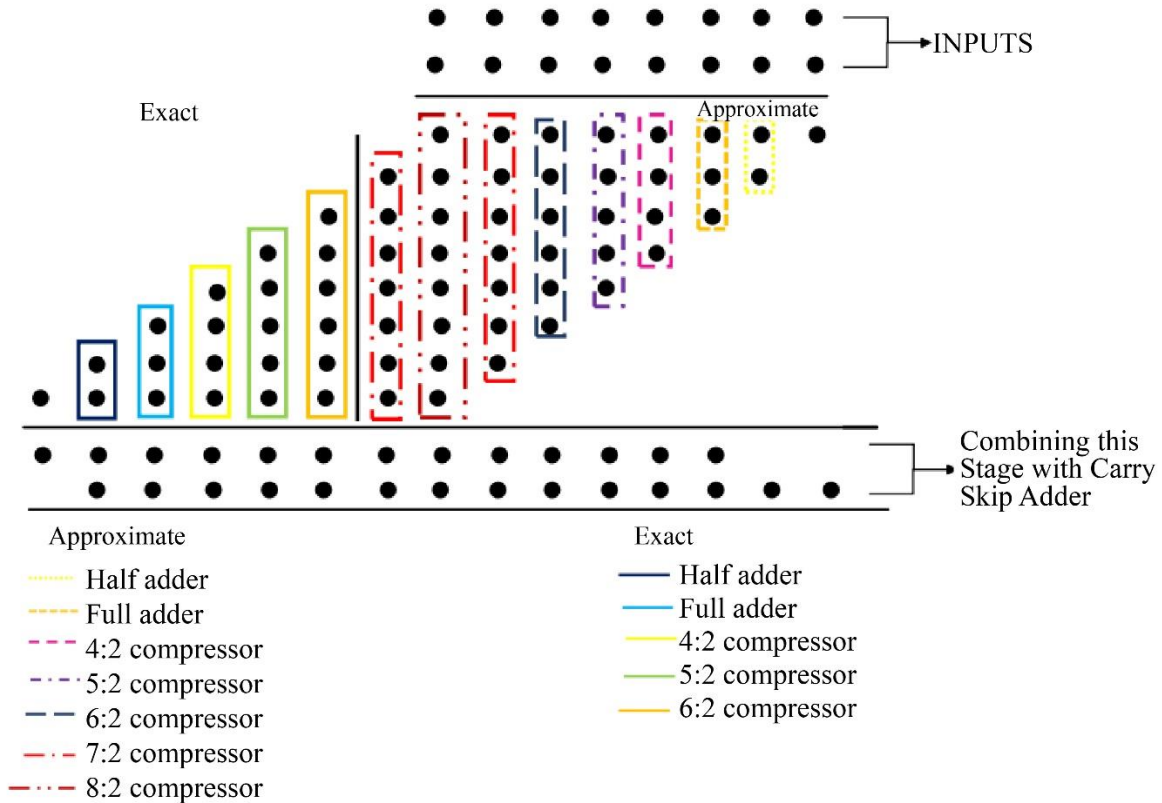


Fig. 13 Schematic representation of 8X8 approximate multiplier (version-IV)

4.4. Version-IV: Introducing Approximation at the 9th Position from the Least Significant Position (LSP) of the Partial Product Order by Using Carry Skip Adder at the Last Stage Addition of Partial Products (Version-I-9th-CSKA)

This design closely mirrors Version III, with a key difference in the use of approximate adders and compressors. In Version IV, as shown in Figure 13, these approximate components are employed up to the 9th column, compared to the 7th column in Version III. By extending the usage of approximate adders and compressors further into the significant part of the multiplier, Version IV aims to reduce hardware utilization and power consumption more effectively. Despite this increased approximation, the design maintains accuracy in the Most Significant Part (MSP) and ensures precision in the final result by using exact adders and compressors where necessary.

5. Results and Discussion

The proposed approximate multiplier designs are realized in both FPGA and ASIC design environments.

5.1. Comparative Analysis on FPGA - Xilinx Vivado

The proposed approximate multipliers are modeled using Verilog HDL, implemented and validated on Xilinx Vivado with Aritix-7, Nexus-4 FPGA board for estimating the hardware utilization at the operating frequency of 200MHz. The obtained results, comprising hardware utilization, power consumption, and maximum operating frequency, are meticulously organized and presented in Table 1, and the

results are plotted in Figure 14. The proposed design methods show significant improvements in hardware utilization and power consumption compared to existing methods. Version-I-7th-FA, Version-II-9th-FA, Version-III-7th-CSKA, and Version-IV-9th-CSKA use fewer LUTs and less power than all existing methods.

Table 1. Performance comparison of the proposed design on FPGA, Aritex-7 device

Design Methods	Hardware Utilization (LUTs)	Power (Watts)
DM-FA (Existing)	73	13.60
DM-CSKA (Existing)	78	13.65
AMC4-FA (Existing)	59	10.43
AMC4-CSKA (Existing)	75	10.24
Version-I (Proposed)	49	6.11
Version-II (Proposed)	40	5.14
Version-III (Proposed)	61	7.44
Version-IV (Proposed)	52	6.91

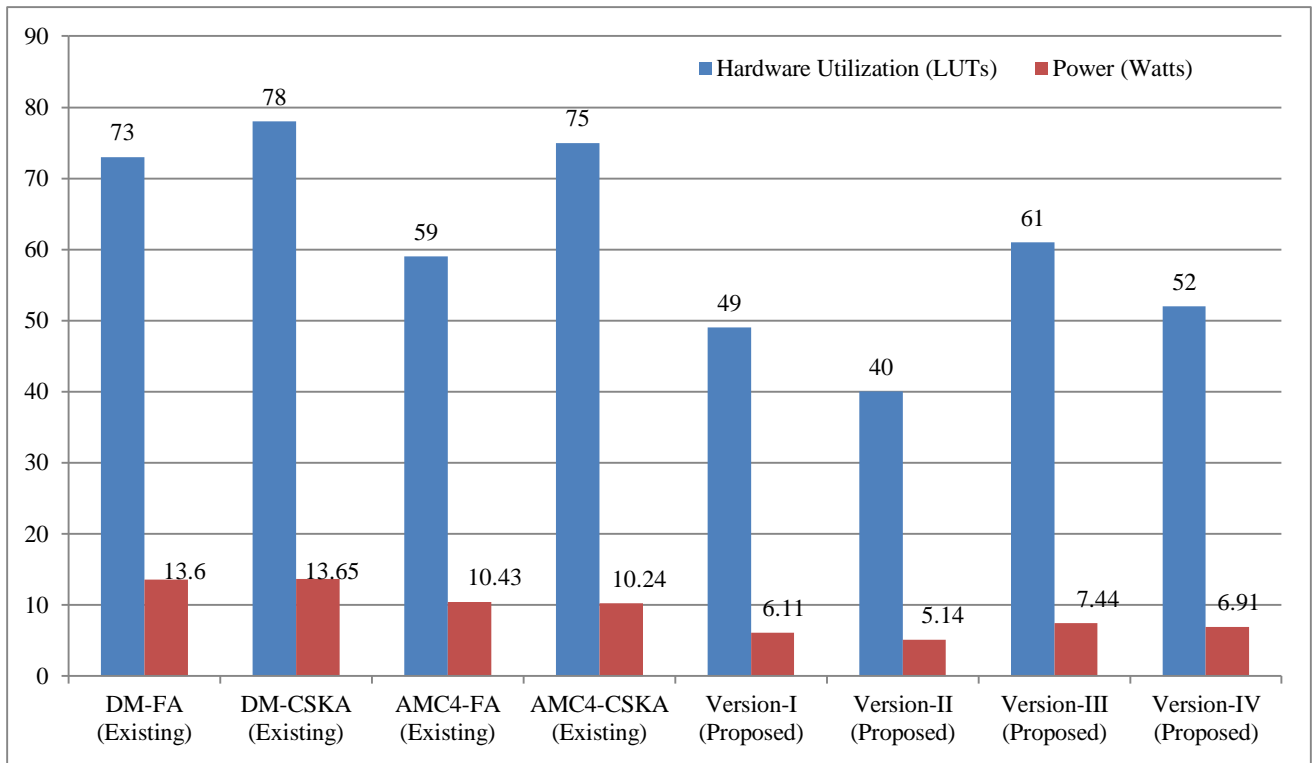


Fig. 14 Performance Analysis on FPGA-Aritex-7 board

Specifically, Version-I-7th-FA uses only 49 LUTs and 6.11 watts, while Version-II-9th-FA is even more efficient with 40 LUTs and 5.14 watts. Among the CSKA designs, Version-III-7th-CSKA and Version-IV-9th-CSKA also show better performance with 61 LUTs and 7.44 watts and 52 LUTs and 6.91 watts, respectively. In contrast, the existing methods like DM-FA and DM-CSKA require 73 and 78 LUTs, consuming 13.60 and 13.65 watts. AMC4-FA and AMC4-CSKA, while slightly more efficient than DM-FA and DM-CSKA, still lag behind the proposed designs, with AMC4-FA using 59 LUTs and 10.43 watts, and AMC4-CSKA using 75 LUTs and 10.24 watts. All methods operate at the same frequency of 200 MHz. The proposed designs consistently demonstrate superior hardware efficiency, showcasing reductions ranging from approximately 21% to 48% compared to existing designs. Similarly, these proposed designs exhibit substantial power efficiency improvements, with reductions in power consumption ranging from about 49% to 62%.

5.2. Comparative Analysis of ASIC Design

The proposed approximate multiplier architectures design entry through Verilog underwent synthesis through Cadance Design Compiler, leveraging a 90-nm CMOS standard cell library. The obtained results, comprising Area utilization, power requirements, and delay time, are meticulously organized and presented in Table 2, and the results are plotted in Figure 15. The proposed design methods show significant improvements in core area, power consumption, delay, and cell count compared to existing methods. Version-I-7th-FA, Version-II-9th-FA, Version-III-7th-CSKA, and Version-IV-9th-CSKA use less core area and power while having lower delays and cell counts than all existing methods. Specifically, Version-I-7th-FA has a core area of 398.45 μm^2 and consumes 8.53 μW , while Version-II-9th-FA is even more efficient with a core area of 380.13 μm^2 and 7.11 μW power consumption.

Among the CSKA designs, Version-III-7th-CSKA and Version-IV-9th-CSKA also show better performance with core areas of 320.24 μm^2 and 290.65 μm^2 and power consumptions of 10.46 μW and 9.21 μW , respectively. In contrast, the existing methods like DM-FA and DM-CSKA have larger core areas of 510.92 μm^2 and 525.83 μm^2 and higher power consumptions of 15.60 μW and 17.65 μW . AMC4-FA and AMC4-CSKA, while slightly more efficient than DM-FA and DM-CSKA, still lag behind the proposed designs, with AMC4-FA having a core area of 380.51 μm^2 and consuming 13.43 μW , and AMC4-CSKA having a core area of 410.68 μm^2 and consuming 14.24 μW .

Table 2. Performance comparison on ASIC design

Design Methods	Core Area (μm^2)	Power (μW)	Delay (ps)	Cell Count (k)
DM-FA (Existing)	510.92	15.60	187	116
DM-CSKA (Existing)	525.83	17.65	138	122
AMC4-FA (Existing)	380.51	13.43	157	80
AMC4-CSKA (Existing)	410.68	14.24	125	102
Version-I (Proposed)	398.45	8.53	115	64
Version-II (Proposed)	380.13	7.11	115	64
Version-III (Proposed)	320.24	10.46	84	78
Version-IV (Proposed)	290.65	9.21	84	74

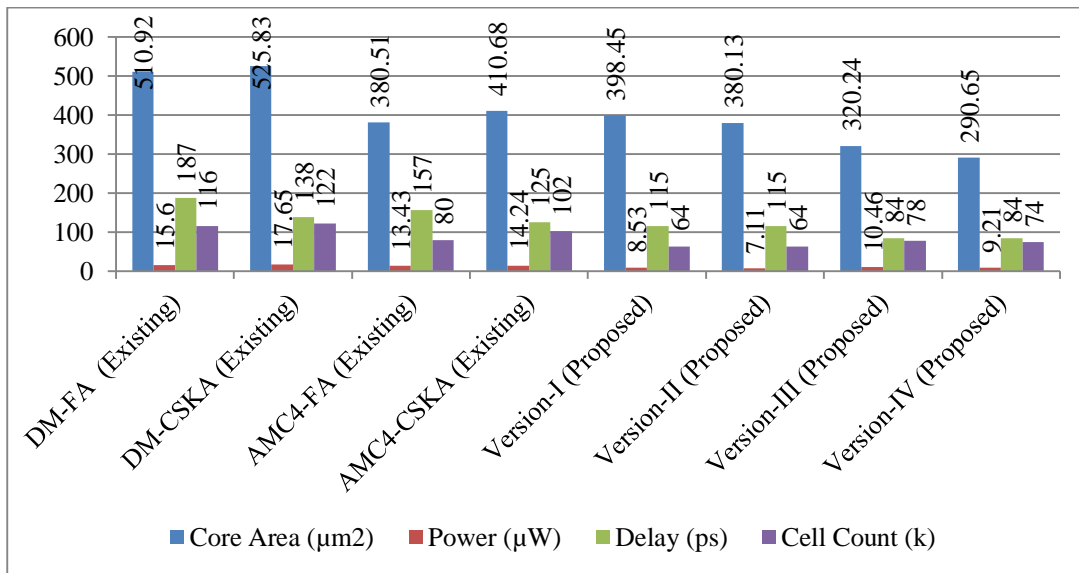


Fig. 15 Performance analysis of proposed design on ASIC environment

Table 3. Error analysis on proposed architectures

Design Methods	Error rate	ED	MED	MRED	NED
Version I (Proposed)	0.0070	14331	0.22	0.22	1.538e ⁻⁵
Version II (Proposed)	0.0070	14331	0.22	0.22	1.538e ⁻⁵
Version III (Proposed)	0.0042	23611	0.36	0.36	1.538e ⁻⁵
Version IV (Proposed)	0.0042	23995	0.36	0.36	1.538e ⁻⁵

All methods also show improvements in delay and cell count. The proposed designs are much more efficient in terms of area and power compared to existing ones. They reduce hardware usage by 24% to 44% and cut power consumption by 40% to 59%. The multiplication speed is improved by 38% to 55%, and cell usage is reduced by 36% to 47%. All proposed versions show significant improvements in area, speed and power efficiency, though there may be trade-offs in performance, complexity, or cost.

The differences between the proposed designs are minor, giving flexibility in choosing the best one based on specific needs. These comparisons highlight the importance of evaluating and optimizing designs for better performance and efficiency in modern integrated circuits, helping designers make better decisions for higher-quality products. Such insights empower designers to make informed decisions, aiming for enhanced performance, efficiency, and overall product quality in integrated circuits.

5.3. Error Analysis

All designs exhibit a uniform Normalized Error Distance (NED) of $1.538e^{-5}$, indicating consistent error normalization and similar absolute deviations between expected and actual outputs.

However, the Error rate (E) and Error Distance (ED) vary in Version-I-7th-FA and Version-II-9th-FA: Error rate of 0.0070, ED of 14331, with Mean Error Distance (MED) and Mean Relative Error Distance (MRED) of 0.2204. Version-III-7th-CSKA and Version-IV-9th-CSKA: Lower error rate of 0.0042 but higher EDs of 23611 and 23995, leading to MED and MRED values of 0.3631 and 0.3690. These metrics in Table 3 show that while NED remains consistent across all designs, Full Adder-based designs (Versions I and II) have lower error distances and consistent MED/MRED. In contrast, CSKA-based designs (Versions III and IV) feature a lower

error rate but higher error distances and MED/MRED, offering advantages in area optimization and power efficiency. Despite uniform NED, FA-based designs have lower error distances, while CSKA-based designs have lower error rates but higher error distances, benefiting from area and power efficiency.

6. Conclusion

In conclusion, this paper presents a novel approach to designing efficient multipliers for Digital Signal Processing (DSP) applications by leveraging higher-order approximate compressors. The proposed 4:2, 5:2, 6:2, 7:2, and 8:2 compressors, combined with various strategies for partial product approximation and final addition using exact full adders or Carry Skip Adders (CSKA), demonstrate significant improvements in power and area efficiency. Specifically, the development of four versions of 8x8 approximate multipliers introduces approximation at different positions of the partial products, optimizing the balance between accuracy and resource utilization. The introduction of approximation at strategic positions within the multipliers leads to substantial reductions in power consumption, achieving average savings of 62% and a noteworthy decrease in hardware utilization by up to 48%.

These results highlight the potential of inexact multipliers to maintain acceptable accuracy levels while optimizing resource usage, making them highly suitable for power-sensitive and area-constrained applications. While error normalization remains consistent, the FA-based designs show lower error distances, whereas CSKA-based designs offer area and power efficiency. The comprehensive comparison of power, area, delay, and error metrics underscores the advancements in hardware efficiency, positioning the proposed designs as a promising solution for enhancing the performance of digital signal processing systems.

References

- [1] Farzad Farshchi, Muhammad Saeed Abrishami, and Sied Mehdi Fakhraie, "New Approximate Multiplier for Low Power Digital Signal Processing," *The 17th CSI International Symposium on Computer Architecture and Digital Systems*, Tehran, Iran, pp. 25-30, 2013. [CrossRef] [Google Scholar] [Publisher Link]

- [2] Honglan Jiang et al., “Approximate Arithmetic Circuits: A Survey, Characterization and Recent Applications,” *Proceedings of the IEEE*, vol. 108, no. 12, pp. 2108-2135, 2020. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [3] Amir Momeni et al., “Design and Analysis of Approximate Compressors for Multiplication,” *IEEE Transactions on Computers*, vol. 64, no. 4, pp. 984–994, 2015. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [4] Georgios Zervakis et al., “Design-Efficient Approximate Multiplication Circuits through Partial Product Perforation,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 10, pp. 3105–3117, 2016. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [5] Suganthi Venkatachalam, and Seok-Bum Ko, “Design of Power and Area Efficient Approximate Multipliers,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 5, pp. 1782–1786, 2017. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [6] Jie Li et al., “Networked Human Motion Capture System Based on Quaternion Navigation,” *Proceedings of the 11th EAI International Conference on Body Area Networks*, pp. 38-44, 2017. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [7] Mohammad Saeed Ansari et al., “Low-Power Approximate Multipliers Using Encoded Partial Products and Approximate Compressors,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 8, no. 3, pp. 404–416, 2018. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [8] Farnaz Sabetzadeh, Mohammad Hossein Moaiyeri, and Mohammad Ahmadinejad, “A Majority-Based Imprecise Multiplier for Ultra-Efficient Approximate Image Multiplication,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 11, pp. 4200–4208, 2019. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [9] Antonio Giuseppe Maria Strollo et al., “Comparison and Extension of Approximate 4-2 Compressors for Low-Power Approximate Multipliers,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 9, pp. 3021–3034, 2020. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [10] Pranose J. Edavoor, Sithara Raveendran, and Amol D. Rahulkar, “Approximate Multiplier Design Using Novel Dual-Stage 4:2 Compressors,” *IEEE Access*, vol. 8, pp. 48337–48351, 2020. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [11] Salim Ullah et al., “High-Performance Accurate and Approximate Multipliers for FPGA-Based Hardware Accelerators,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 2, pp. 211–224, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [12] Mohammad Ahmadinejad, and Mohammad Hossein Moaiyeri, “Energy- and Quality-Efficient Approximate Multipliers for Neural Network and Image Processing Applications,” *IEEE Transactions on Emerging Topics in Computing*, vol. 10, no. 2, pp. 1105–1116, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [13] H.C. Prashanth et al., “Design and Evaluation of In-Exact Compressor Based Approximate Multipliers,” *Proceedings of the ACM Great Lakes Symposium on VLSI 2022, GLSVLSI*, pp. 431–436, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [14] Muteen Munawar, Zain Shabbir, and Muhammad Akram, “Area, Delay, and Energy-Efficient Full Dadda Multiplier,” *Journal of Circuits, Systems and Computers*, vol. 32, no. 15, pp. 1–16, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [15] Atefeh Minaeifar et al., “A High-Accuracy Low-Power Approximate Multipliers with New Error Compensation Technique for DSP Applications,” *Circuits, Systems, and Signal Processing*, vol. 43, pp. 526–544, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [16] Chip-Hong Chang, Jiangmin Gu, and Mingyan Zhang, “Ultra Low-Voltage Low-Power CMOS 4-2 and 5-2 Compressors for Fast Arithmetic Circuits,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 51, no. 10, pp. 1985–1997, 2004. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [17] Omid Akbari et al., “Dual-Quality 4:2 Compressors for Utilizing in Dynamic Accuracy Configurable Multipliers,” *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, vol. 25, no. 4, pp. 1352–1361, 2017. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]