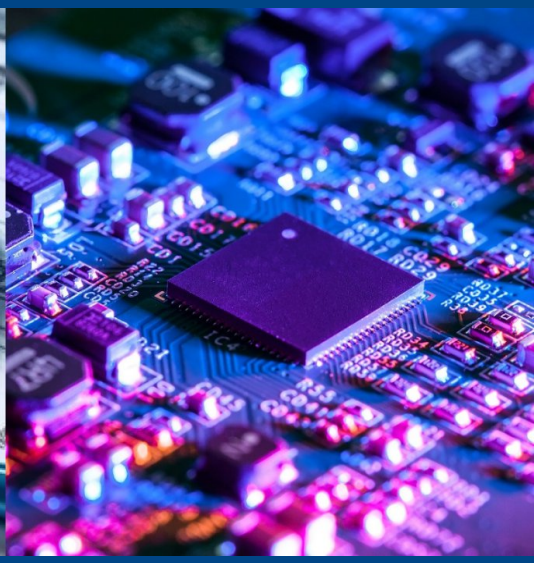
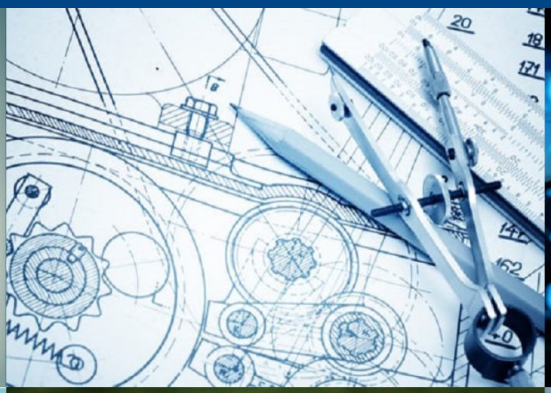
ISSN: 2582-7219

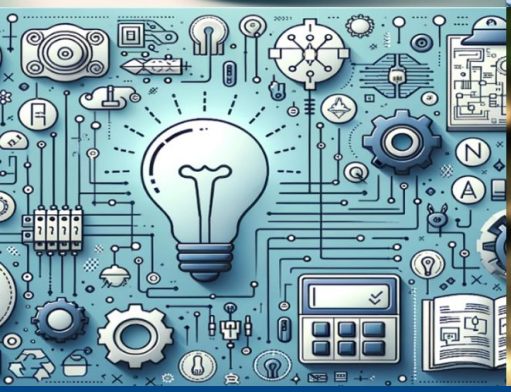




International Journal of Multidisciplinary Research in Science, Engineering and Technology

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.206 **Volume 9, Issue 4, April 2026**

© 2026 IJMRSET | Volume 9, Issue 4, April 2026 | DOI:10.15680/IJMRSET.2026.0904290

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

High-Performance Approximate Unsigned Multipliers Employing an Optimized 4:2 Compressor Configuration

T. Maheswaran, S.Mekanthika, R.Narmatha, S.Rithika

Assistant Professor of ECE, M.P.NachimuthuM.Jaganathan Engineering College, Erode, Tamil Nadu, India

Final Year ECE, M.P.NachimuthuM.Jaganathan Engineering College, Erode, Tamil Nadu, India

Final Year ECE, M.P.NachimuthuM.Jaganathan Engineering College, Erode, Tamil Nadu India

Final Year ECE, M.P.NachimuthuM.Jaganathan Engineering College, Erode, Tamil Nadu India

ABSTRACT : Approximate computing has emerged as an efficient paradigm for improving performance and reducing power consumption in error-tolerant applications such as image processing, machine learning, and multimedia systems. Among arithmetic units, multipliers are one of the most critical and resource-intensive components in digital systems. This work presents a high- performance approximate unsigned multiplier using an optimized 4:2 compressor configuration. The proposed design focuses on reducing critical path delay and hardware complexity while maintaining acceptable accuracy. By introducing an optimized approximate 4:2 compressor in the partial product reduction stage, the multiplier achieves significant improvements in speed and area efficiency. The design selectively approximates lower significant bits to minimize overall error impact while maximizing performance gains. Simulation results demonstrate that the proposed multiplier achieves reduced delay, lower power consumption, and improved area efficiency compared to conventional exact and existing approximate multipliers. Error metrics such as Mean Error Distance (MED) and Error Rate (ER) are analyzed to validate computational accuracy. The architecture is implemented using HDL and evaluated through synthesis and simulation tools.

KEYWORDS: High performance approximate unsigned multipliers employing an optimized 4:2 compressor configuration.

I. INTRODUCTION TO THE HIGH PERFORMANCE APPROXIMATE UNSIGNED MULTIPLIERS

EMPLOYING AN OPTIMIZED 4:2 COMPRESSOR CONFIGURATION.

In modern digital systems, multipliers are fundamental building blocks widely used in applications such as digital signal processing (DSP), image and video processing, cryptography, and machine learning. As the demand for high-speed and low-power devices continues to grow, the design of efficient multiplier architectures has become a critical area of research in VLSI system design.

However, conventional exact multipliers often involve significant hardware complexity, high power consumption, and longer propagation delays, which limit their performance in resource-constrained environments. To address these challenges, approximate computing has been introduced as a promising solution that allows controlled inaccuracies in computation to achieve substantial improvements in performance, power efficiency, and area utilization.This approach is particularly suitable for applications where perfect accuracy is not mandatory, such as multimedia processing and artificial intelligence, where minor errors do not significantly affect output quality. One of the key techniques used to enhance multiplier performance is the use of compressor circuits, especially the 4:2 compressor, which plays a crucial role in partial product reduction. Traditional 4:2 compressors, although effective, still contribute to delay and power overhead due to complex carry propagation mechanisms. Therefore, optimizing the compressor design is essential to improving overall multiplier efficiency.In this work, a high-performance approximate unsigned multiplier is proposed using an optimized 4:2 compressor configuration.

The proposed compressor reduces logic complexity and minimizes carry chain delay, leading to faster computation and lower power consumption. By integrating the optimized compressors into the multiplier architecture, the overall system

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

achieves enhanced speed and reduced hardware cost while maintaining acceptable computational accuracy.The remainder of this paper is organized as follows: Section II reviews existing literature and related work. Section III describes the proposed optimized compressor and multiplier architecture. Finally, Section V concludes the paper and discusses future research directions.In modern digital systems, multipliers are one of the most essential arithmetic components and are widely used in applications such as digital signal processing (DSP), image and video processing, machine learning, and communication systems. Since multiplication is a computation-intensive operation, the overall performance of these systems largely depends on the efficiency of the multiplier design. However, conventional exact multipliers require a large number of logic elements, resulting in high power consumption, increased delay, and larger chip area, which are critical limitations in modern VLSI design.

To overcome these challenges, approximate computing has gained significant attention as an emerging design paradigm. This approach relaxes the requirement for exact computation and allows small, controlled errors in exchange for improvements in speed, power efficiency, and hardware complexity. Approximate multipliers are particularly suitable for error-tolerant applications such as multimedia processing, artificial intelligence, and data analytics, where minor inaccuracies do not significantly affect the overall output quality.A key factor in improving multiplier performance is the efficient reduction of partial products. Compressor circuits, especially the 4:2 compressor, play a vital role in this process by reducing multiple input bits into fewer output bits, thereby decreasing the number of computation stages. Although traditional 4:2 compressors improve performance compared to basic adders, they still suffer from issues such as carry propagation delay and increased power consumption due to complex internal logic structures. To address these limitations, this paper proposes a high-performance approximate unsigned multiplier employing an optimized 4:2 compressor configuration.

The proposed compressor is designed to simplify internal logic, reduce critical path delay, and minimize power consumption while maintaining acceptable accuracy levels. By integrating the optimized compressors into the multiplier architecture, the overall system achieves improved speed, reduced area, and enhanced energy efficiency. The rest of the paper is organized as follows: Section II presents the literature survey of existing multiplier designs and approximation techniques. Section III explains the proposed optimized 4:2 compressor and multiplier architecture. Section IV discusses the simulation results and performance evaluation.Finally, Section V concludes the paper and outlines future work. This makes the design highly suitable for modern VLSI applications that require high speed, low power, and efficient area utilization. The remainder of this paper is organized as follows: Section II presents the background and literature survey. Section III describes the proposed optimized compressor and Finally, SectionV concludes the paper and outlines future research directions.

II. BACKGROUND AND LITERATURE SURVEY

Multipliers are fundamental components in digital systems and are extensively used in applications such as digital signal processing (DSP), image processing, machine learning, and communication systems. The performance of these systems largely depends on the efficiency of multiplier architectures. Conventional multipliers, including array multipliers, Wallace tree multipliers, and Booth multipliers, are designed to produce accurate results but often suffer from high power consumption, increased propagation delay, and large hardware complexity. These limitations make them less suitable for modern applications that demand high speed and energy efficiency.To overcome these challenges, advanced techniques such as partial product reduction using compressor circuits have been introduced. Among these, the 4:2 compressor is widely used due to its ability to reduce multiple input bits into fewer outputs, thereby improving computational speed. However, traditional 4:2 compressor designs still involve complex carry propagation paths, which contribute to higher delay and power consumption.

In recent years, approximate computing has emerged as an effective solution to improve the performance of arithmetic circuits by allowing controlled inaccuracies in computation. Various research works have proposed approximate multipliers, approximate adders, and optimized compressor designs to reduce power, delay, and area. Approximate 4:2 compressors simplify internal logic and reduce hardware complexity, resulting in faster and more energy-efficient designs.Additionally, techniques such as truncation and hybrid multiplier architectures have been developed to achieve a balance between accuracy and performance. While these approaches offer significant improvements, they often introduce higher error rates or increased design complexity. Moreover, many existing designs focus on optimizing a single parameter, such as delay or power, without achieving an optimal balance among all performance metrics.

Therefore, there is a need for an improved multiplier design that simultaneously optimizes speed, power consumption, and area while maintaining acceptable accuracy. This motivates the proposed work, which employs an optimized 4:2 application.The efficiency of these applications depends heavily on the speed, power consumption, and area of the multiplier circuits. Traditional multiplier architectures, such as array multipliers, Wallace tree multipliers, and Booth multipliers, are widely used due to their ability to produce highly accurate results. his motivates the proposed work, which introduces an optimized 4:2 compressor-based approximate unsigned multiplier that aims to achieve reduced delay.

To improve multiplier efficiency, partial product reduction techniques have been introduced, where compressor circuits play a vital role in minimizing the number of intermediate addition stages. Among various compressor architectures, the 4:2 compressor is extensively used due to its capability to efficiently combine multiple input bits and reduce computation time. Despite these advantages, conventional 4:2 compressors still suffer from complex internal logic and long carry propagation paths, which adversely affect delay and power performance.

TABLE I SUMMARY OF THE LITERATURE SURVEY

Ref. No	Method	Outcomes	Challenges
[1]	Array Multiplier	Simple structure and accurate results	High delay and large area
[2]	Wallace Tree Multiplier	Faster reduction of partial products	Complex wiring and layout
[3]	Booth Multiplier	Reduces number of partial products	Complex encoding logic
[4]	Modified Booth Multiplier	Further reduction in partial products	Increased design complexity
[5]	Braun Multiplier	Regular structure suitable for VLSI	High power consumption
[6]	Conventional 4:2 Compressor	Improves reduction efficiency	Carry propagation delay
[7]	Exact 4:2 Compressor	High accuracy in computation	Larger hardware complexity
[8]	Approximate Adder-based Multiplier	Reduces power and increases speed	Loss of accuracy
[9]	Error-Tolerant Adder (ETA)	Faster computation with low power	Reduced precision
[10]	Approximate 4:2 Compressor	Lower delay and reduced area	Higher error rate

The literature survey highlights the evolution of multiplier design techniques aimed at improving performance in terms of speed, power, and area. Conventional multipliers such as array and Wallace tree structures provide accurate results but suffer from increased delay, power consumption, and hardware complexity. To address these limitations, advanced techniques such as Booth multiplication and compressor- based architectures were introduced to reduce the number of

© 2026 IJMRSET | Volume 9, Issue 4, April 2026 | DOI:10.15680/IJMRSET.2026.0904290

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

partial products and improve computational speed.

In recent years, approximate computing has gained significant attention as an effective approach for enhancing multiplier efficiency. Approximate adders and compressors reduce circuit complexity and power consumption by allowing controlled inaccuracies in computation. However, these designs often introduce higher error rates, which must be carefully managed. Hybrid and truncated multipliers attempt to balance accuracy and performance but increase design complexity or reduce precision.

III PROPOSED METHOD

The proposed scheme consists of four sections. The first two sections describe the proposed design technique of compressors and three approximate 4:2 compressors based on the proposed design technique. In the next two sections, a new approach for multiplier structure as well as two new approximate unsigned multiplier structures are presented.

A. Proposed Technique for Designing Approximate Compressors

This section attempts to analyze the parameters that influence the accuracy of an approximate compressor and then proposes an approach to explore a new design. The efficiency of an approximate compressor can be analyzed by applying it to a multiplier structure. Fig. 2 is commonly used to represent an 8-bit unsigned multiplier. Two stages are depicted in Fig. 2: stage 1 and stage 2.

Assuming uniform distribution for input operands of the multiplier, the probability of compressors input in stage 1 can be calculated. However, it is not clear for compressors in stage 2, because it is contingent on the approximate compressor's structure. So, this section focuses on the probability of the output derived from stage 1.

Let's assume P_0 be the probability that the compressor output is exact also, $P+$ and $P-$ signify the probabilities of compressor output with positive and negative approximation, respectively. In this scenario, placing two compressors in one

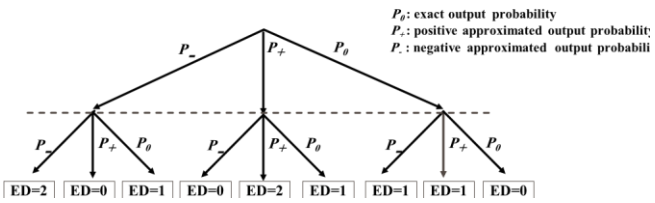


Fig. 3. States of absolute ED resulting of two compressors in one column at stage1.

column, for example in columns 7 through 10, results in three different Error Distance (ED) with absolute values of 0, 1, and 2, where ED is the difference between the correct and incorrect output. Fig. 3 summarizes all conceivable output states' combinations of two compressors in one column, and their corresponding absolute ED.

According to the EDs, shown at the bottom of the Fig. 3, under three conditions, the sum of EDs of two compressors in the same column becomes zero. One of the conditions is when both compressor's outputs are exact, where the probability of this condition equals $P_0 \times P_0 = P_0^2$. Besides, if the probability of the first (and the second) compressor's outputs has opposite signs ($P+$ and $P-$), the generated ED will be zero with the probability of $2 \times (P+ \times P-)$. Here, the probability of the correct output for a column at stage 1 is given by P_0 (ED=0), which equals the sum of all the three conditions' probabilities, where ED is zero. Equation (4) defines P_0 P(ED = 0). According to the probability concepts, (5) is also established.

$$P_0(ED = 0) = P_0^2 + 2 \times (P_+ \times P_-) \quad (4)$$

$$P_0 + P+ + P- = 1 \quad (5)$$

A well-designed approximate multiplier maximizes P_0 P(ED = 0). This means that P_0 must be as high as possible or $P+=P-$ in order to maximize P_0 P(ED = 0).

Let's assume that the probability of a positive or negative outcome is denoted using the notation p_i+ and p_i- for each of

© 2026 IJMRSET | Volume 9, Issue 4, April 2026 | DOI:10.15680/IJMRSET.2026.0904290

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

the 16 input patterns of 4:2 compressor. The ED for each pattern is represented by ED_i. So, the positive and negative approximation probabilities of the compressor output can be determined using equation (6) and (7), respectively. Equations

(6) and (7) must be equal to set the condition P+=P–.
X

$$(p_i^+ \times ED_i) = P+ \qquad (6)\ X$$
$$(p_i^- \times ED_i) = P- \qquad (7)$$

Consequently, it is necessary to know the probability of the compressor's inputs to reduce the multiplication error. The compressor's inputs at stage 1 are partial products, which are generated by an AND gate. By assuming a uniform distribution of the multiplier's inputs, the probability of the partial product to be '0' or '1' is 3/4 and 1/4, respectively.

For example, the probability of '0001' is $(\frac{3}{4}\times\frac{3}{4}\times\frac{3}{4}\times\frac{1}{4} = \frac{27}{256})$. A categorization based on the probability of each of the 16 input patterns labeled as x1x2x3x4 is illustrated in Table I. The input patterns that have the same compressor's output are grouped together, as explained in [20].

Since each group of inputs has the same probability,
P+=P– can be established by considering of equal positive

TABLE I

PROBABILITY OF OCCURRENCE OF INPUTS AND GROUPING OF INPUTS

Grouping No.	$x_1x_2x_3x_4$	Probability of each input in the group
first	0000	81/256
second	0001,0010,0100,1000	27/256
third	0011,1100,0101,1010,0110,1001	9/256
fourth	0111,1110,1011,1101	3/256
fifth	1111	1/256

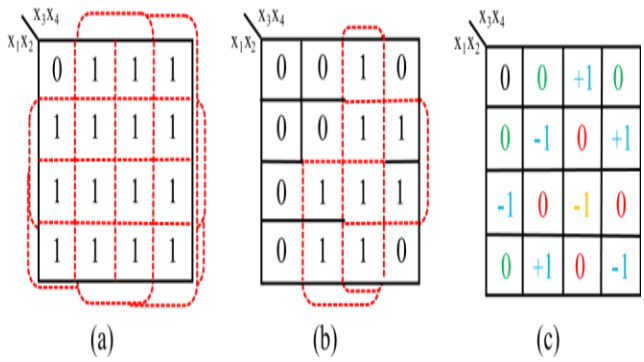


Fig. 4. Karnaugh map for AC6G-12. a) sum output. b) carry output. c) the approximation used for input patterns.

© 2026 IJMRSET | Volume 9, Issue 4, April 2026 | DOI:10.15680/IJMRSET.2026.0904290

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

and negative ED_i in each input group. So, an equal negative and positive approximations to each input group in the proposed approximate 4:2 compressor will be applied. Further information about the importance of using both positive and negative approximation in the truth table for compressors development can be found in [13] and [18]. However, the AND gates in stage 1 does not generate equal probability for the compressor inputs, as explained in [18]. The probabilities of the input are not equal, as shown in Table I, which contrasts with [18].

B. Three Proposed Approximate Compressors

The proposed schemes do not include the C_{in} and C_{out}, like existing approximate compressor. The proposed compressors are categorized according to the number of gates that they have. They are proposed as follows:

1) Approximate Compressors with 6 Gate (AC6G)
2) Approximate Compressors Free Gate I (ACFGI)
3) Approximate Compressor Free Gate II (ACFG II)
4) AC6G design

1) AC6G Design: An important design consideration for this compressor is assuming that P+=P– as mentioned in section A. The compressors in the same category of the proposed AC6G design are listed in Table II. Using input permutation, 16 compressors in this category can be created. The compressors are denoted by the notation AC6G-n, where n is the compressor number.

2)

As an example, AC6G-12 Karnaugh map has been shown in Fig. 4, where Fig. 4(a) and (b) show Karnaugh map of output sum and carry, respectively and Fig. 4c shows the used approximation for input patterns. The first to fifth groups of the inputs are shown in black, green, blue, red, and yellow color, respectively. Seven input patterns from third and fifth group are approximated. P+=P– is set for the third group of the inputs, while the fifth group is approximated due to the absence of the C_{in} and C_{out}. The compressors in AC6G category have the highest accuracy compared with the two other proposed category due to the well-established condition

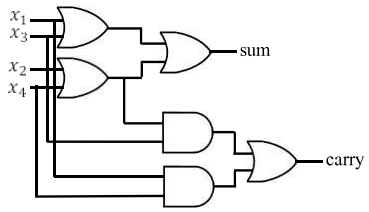


Fig. 5.The gate schematic of the AC6G-12.

TABLE II

OUTPUTS OF ALL ACF6G COMPRESSORS

AC6Gi-2	$(x_1 + x_2) + (x_3 + x_4)$	$(x_1 \cdot (x_3 + x_4)) + (x_2 \cdot x_4)$
AC6Gi-3	$(x_1 + x_2) + (x_3 + x_4)$	$(x_1 \cdot (x_3 + x_4)) + (x_3 \cdot x_4)$
AC6Gi-4	$(x_1 + x_2) + (x_3 + x_4)$	$(x_2 \cdot (x_3 + x_4)) + (x_1 \cdot x_3)$
AC6Gi-5	$(x_1 + x_2) + (x_3 + x_4)$	$(x_2 \cdot (x_3 + x_4)) + (x_1 \cdot x_4)$
AC6Gi-6	$(x_1 + x_2) + (x_3 + x_4)$	$(x_2 \cdot (x_3 + x_4)) + (x_3 \cdot x_4)$
AC6Gi-7	$(x_1 + x_2) + (x_3 + x_4)$	$(x_3 \cdot (x_1 + x_2)) + (x_1 \cdot x_2)$
AC6Gi-8	$(x_1 + x_2) + (x_3 + x_4)$	$(x_4 \cdot (x_1 + x_2)) + (x_1 \cdot x_2)$
AC6Gi-9	$(x_1 + x_3) + (x_2 + x_4)$	$(x_1 \cdot (x_2 + x_4)) + (x_2 \cdot x_3)$
AC6Gi-10	$(x_1 + x_3) + (x_2 + x_4)$	$(x_1 \cdot (x_2 + x_4)) + (x_3 \cdot x_4)$
AC6Gi-11	$(x_1 + x_3) + (x_2 + x_4)$	$(x_3 \cdot (x_2 + x_4)) + (x_1 \cdot x_2)$
AC6Gi-12	$(x_1 + x_3) + (x_2 + x_4)$	$(x_3 \cdot (x_2 + x_4)) + (x_1 \cdot x_4)$
AC6Gi-13	$(x_1 + x_4) + (x_2 + x_3)$	$(x_1 \cdot (x_2 + x_3)) + (x_2 \cdot x_4)$
AC6Gi-14	$(x_1 + x_4) + (x_2 + x_3)$	$(x_1 \cdot (x_2 + x_3)) + (x_3 \cdot x_4)$
AC6Gi-15	$(x_1 + x_4) + (x_2 + x_3)$	$(x_4 \cdot (x_2 + x_3)) + (x_1 \cdot x_2)$
AC6Gi-16	$(x_1 \quad x_4) \quad (x_2 \quad x_3)$	$(x_4 \cdot (x_2 \quad x_3)) \quad (x_1 \cdot x_3)$

© 2026 IJMRSET | Volume 9, Issue 4, April 2026 | DOI:10.15680/IJMRSET.2026.0904290

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

TABLE III

OUTPUTS OF ALL ACFGI COMPRESSORS

Compressors No.	sum	carry
ACFGI-1	1	x_1
ACFGI-2	1	x_2
ACFGI-3	1	x_3
ACFGI-4	1	x_4

P+ = P–. According to (2), Critical Path Delay (CPD) of exact compressor is "2tXOR+tAND+tOR", where tXOR, tAND and tOR is the delay of 2-input XOR, AND and OR gates, respectively. CPD of AC6Gs equals to "2×tOR+tAND".

This paper focuses on the design of compressors with nearly equal positive and negative approximations for the input groups. There is a trade-off between designing an efficient hardware and verifying P+=P–. Since reducing the cost of the hardware is a fundamental design objective, it is aimed to achieve a free gate compressor, where condition P+=P– is met for as many inputs as possible for the two proposed compressors, named: ACFGI and ACFGII.

3) ACFGI Design: The ACFGI outputs' equations are tabulated in Table III. Carry output corresponds to one of the inputs, whereas its sum output is equal to '1'. Four distinct compressor designs can be generated which tabulated in Table III by permutation input.

4)

As an example, the Karnaugh map of the ACFGI-4 is shown in Fig. 6.

In the design of this compressor, achieving low hardware is a priority, so P+=P– is confirmed only for the third input

Compressors No.	sum	carry
AC6G-1	$(x_1 + x_2) + (x_3 + x_4)$	$(x_1 \cdot (x_3 + x_4)) + (x_2 \cdot x_3)$

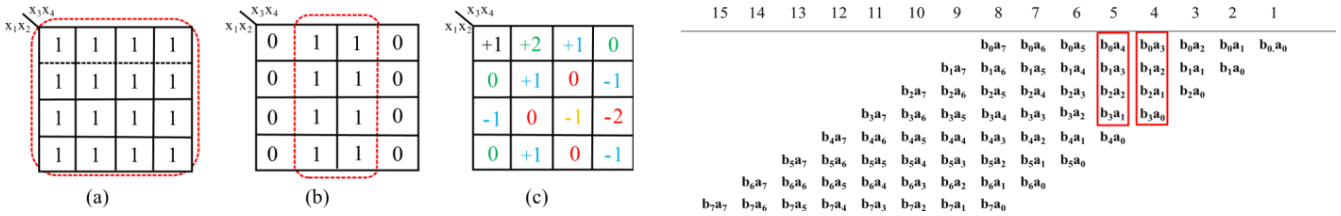


Fig. 6. Karnaugh map of ACFGI-4. a) sum output. b) carry output. c) the approximation used for the input patterns.

© 2026 IJMRSET | Volume 9, Issue 4, April 2026 | DOI:10.15680/IJMRSET.2026.0904290

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

TABLE IV

OUTPUTS OF ALL ACFGII COMPRESSORS

Compressors No.	sum	carry
ACFGII-1	x_1	x_2
ACFGII-2	x_1	x_3
ACFGII-3	x_1	x_4
ACFGII-4	x_2	x_1
ACFGII-5	x_2	x_3
ACFGII-6	x_2	x_4
ACFGII-7	x_3	x_1
ACFGII-8	x_3	x_2
ACFGII-9	x_3	x_4
ACFGII-10	x_4	x_1
ACFGII-11	x_4	x_2
ACFGII-12	x_4	x_3

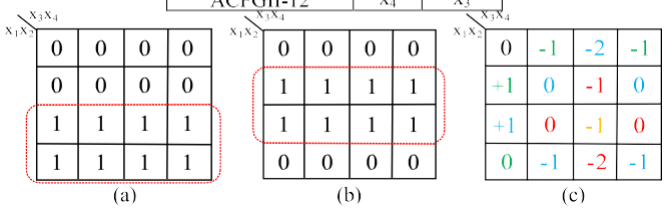


Fig. 7. Karnaugh map for ACFGII-1. a) sum output. b) carry output. c) approximation used for input patterns.

group. Despite its lack of high accuracy, this compressor is the most efficient candidate for low-cost implementation. Since sum output is a constant value of '1', the final addition step in the multiplier is simplified. This compressor is appropriate for applying to the least significant partial product columns.

5) ACFGII Design: The output equations of the ACFGII designs are tabulated in Table IV. From this table, twelve distinct compressors with a slightly different set of input permutations, can be developed. However, one of the possible compressors have been reported in [14]. Hence, this paper explores all possible ACFGII compressors and their usage in multipliers.

ACFGII-1's compressor's Karnaugh map is shown in Fig. 7. From Fig. 7(a) and (b), the sum and carry have equal ones in their Karnaugh maps. From Fig. 7(c), the ACFGII-1 represents ten inputs' approximations, where the input '0000' has the highest probability values and it is not approximated. Therefore, ACFGII generates higher performance in terms of accuracy in comparison to ACFGI.

C. Proposed Technique for Designing Approximate Multiplier

In this paper different compressors for adjacent partial product column are used. The input probability of the two compressors in the two columns of the partial products isFig. 8.Stage 1 of the 8-bit multiplier.first studied. The proposed design approach for creating the multiplier's structure is then explained.Fig. 8 shows the first stage of the 8-bit multiplier with its two operands: A and B. Two compressors in columns 4 and 5 have been highlighted by red color to analyze their behavior as an example of adjacent compressors.

Regardless of their operand's lengths, there is a relation between two adjacent columns in multipliers, e.g., if both b_0a_3 and b_1a_2 in column 4 are equal to '1', it implies that both a_3 and b_1 have value of 1. So, it is reasonable to assume that b_1a_3 in column 5 must have value of '1'. Hence, it can be concluded that the inputs of the compressor in neighboring columns are dependent.

Let's assume having two compressors, one in column i and another in column i + 1. If one of the 16 input patterns occurs

© 2026 IJMRSET | Volume 9, Issue 4, April 2026 | DOI:10.15680/IJMRSET.2026.0904290

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

in column i, the probability of some inputs occurring in column i + 1 is zero. Additionally, the probability of some of the input patterns is higher than others due to their mentioned dependencies. As a result, if one of the input patterns occurs in column i, the probabilities cannot be obtained using given information in Table I. Hence, the conditional probabilities for patterns occurrences have been calculated from the multiplier simulation using MATLAB software and tabulated in Table V.

For instance, if the compressor input in column i is ”0001”, then P(i+1|i=0001) gives the probability of all possible input patterns in column i+1. Based on presented information in Table V, if an input pattern from the nth ($1 \leq n \leq 5$) input group occurs in column i of a compressor, the same input pattern is most probably among inputs of the nth group in column i +1. For example, if the compressor input in column i is equal to group belongs to ’0001’. It worth to mention that Table I representsprobabilities of all input patterns in a group that have the same probability, while Table V represents, conditional occurrence probabilities of the input patterns and if a pattern occurs in column i, the groups do not have the same probability in column i + 1. ’0001’ (an input from the second group), then in column i + 1, the highest probability input compared to other inputs of second.

The design principle deduced from Table V is that if the compressor input in column i is approximated, then that input in column i +1 must be calculated accurately or approximated by the opposite sign. It is possible to get close to approximations with opposite signs by using proposed compressors with input permutations for columns i and i + 1.
The proposed multiplier structures utilize a range of compressors with different approximations following the result of Table V. Since the proposed design guideline isany size is the same, the probabilities given in Table V for a multiplier of size n are true and proposed design can be easily generalized to n-bit multipliers. The first four columns of proposed multipliers are truncated. For proposed-mul1 and proposedmul2, respectively, ACFGIs and ACFGIIs are utilized, while AC6Gs are used in the upper columns for both proposed multipliers. As a result, the algorithm is started by figuring out which AC6Gs should be in the upper columns, then considering the interdependency between columns, appropriate compressors for the middle columns are chosen. To begin, exact compressors are arranged in a multiplier structure, and then the compressor for column 11 is chosen by swapping in all 16 compressors and comparing their NMED parameters to get the lowest value. The same process is repeated for another upper column. The procedure has been also used to establish the best arrangement of compressors for the middle columns.

TABLE V

OCCURRENCE PROBABILITY OF COLUMN I+1 INPUTS ACCORDING TO COLUMN I INPUTS

Probabilities	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
P(i+1|i=0000)	$\frac{178}{324}$	$\frac{42}{324}$	$\frac{32}{324}$	0	$\frac{30}{324}$	$\frac{6}{324}$	0	0	$\frac{26}{324}$	$\frac{6}{324}$	$\frac{4}{324}$	0	0	0	0	0
P(i+1|i=0001)	$\frac{26}{108}$	$\frac{26}{108}$	$\frac{16}{108}$	$\frac{16}{108}$	$\frac{6}{108}$	$\frac{6}{108}$	0	0	$\frac{4}{108}$	$\frac{4}{108}$	0	$\frac{2}{108}$	0	0	0	0
P(i+1|i=0010)	$\frac{30}{108}$	$\frac{10}{108}$	$\frac{22}{108}$	0	$\frac{18}{108}$	$\frac{6}{108}$	$\frac{12}{108}$	0	$\frac{6}{108}$	0	$\frac{4}{108}$	0	0	0	0	0
P(i+1|i=0011)	0	0	$\frac{10}{36}$	$\frac{10}{36}$	0	0	$\frac{6}{36}$	$\frac{6}{36}$	0	0	$\frac{2}{36}$	$\frac{2}{36}$	0	0	0	0
P(i+1|i=0100)	$\frac{32}{108}$	$\frac{8}{108}$	$\frac{8}{108}$	0	$\frac{22}{108}$	$\frac{4}{108}$	0	0	$\frac{16}{108}$	$\frac{4}{108}$	$\frac{4}{108}$	0	$\frac{10}{108}$	0	0	0
P(i+1|i=0101)	$\frac{6}{36}$	$\frac{4}{36}$	$\frac{4}{36}$	$\frac{4}{36}$	$\frac{4}{36}$	$\frac{4}{36}$	0	0	$\frac{2}{36}$	$\frac{2}{36}$	$\frac{2}{36}$	$\frac{2}{36}$	0	$\frac{2}{36}$	0	0
P(i+1|i=0110)	0	0	0	0	$\frac{12}{36}$	$\frac{4}{36}$	$\frac{8}{36}$	0	0	0	0	0	$\frac{6}{36}$	$\frac{2}{36}$	$\frac{4}{36}$	0
P(i+1|i=0111)	0	0	0	0	0	0	$\frac{4}{12}$	$\frac{4}{12}$	0	0	0	0	0	0	$\frac{2}{12}$	$\frac{2}{12}$
P(i+1|i=1000)	$\frac{42}{108}$	$\frac{10}{108}$	$\frac{8}{108}$	0	$\frac{10}{108}$	$\frac{2}{108}$	0	0	$\frac{26}{108}$	$\frac{6}{108}$	$\frac{4}{108}$	0	0	0	0	0
P(i+1|i=1001)	$\frac{6}{36}$	$\frac{6}{36}$	$\frac{4}{36}$	$\frac{4}{36}$	$\frac{2}{36}$	$\frac{2}{36}$	0	0	$\frac{4}{36}$	$\frac{4}{36}$	$\frac{2}{36}$	$\frac{2}{36}$	0	0	0	0
P(i+1|i=1010)	$\frac{6}{36}$	$\frac{2}{36}$	$\frac{4}{36}$	0	$\frac{6}{36}$	$\frac{2}{36}$	$\frac{4}{36}$	0	$\frac{6}{36}$	$\frac{2}{36}$	$\frac{4}{36}$	0	0	0	0	0
P(i+1|i=1011)	0	0	$\frac{2}{12}$	$\frac{2}{12}$	0	0	$\frac{2}{12}$	$\frac{2}{12}$	0	0	$\frac{2}{12}$	$\frac{2}{12}$	0	0	0	0
P(i+1|i=1100)	0	0	0	0	0	0	0	0	$\frac{16}{36}$	$\frac{4}{36}$	$\frac{4}{36}$	0	$\frac{10}{36}$	$\frac{2}{36}$	0	0
P(i+1|i=1101)	0	0	0	0	0	0	0	0	$\frac{2}{12}$	$\frac{2}{12}$	$\frac{2}{12}$	$\frac{2}{12}$	$\frac{2}{12}$	$\frac{2}{12}$	0	0
P(i+1|i=1110)	0	0	0	0	0	0	0	0	0	0	0	0	$\frac{6}{12}$	$\frac{2}{12}$	$\frac{4}{12}$	0
P(i+1|i=1111)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\frac{2}{4}$	$\frac{2}{4}$

© 2026 IJMRSET | Volume 9, Issue 4, April 2026 | DOI:10.15680/IJMRSET.2026.0904290

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

generic, a trialand-error method is required to determine the appropriate position for the compressors to reduce the total error of multiplier.

Algorithm 1 describes the proposed method in multipliers implementation. It is also notable that since the process of creating partial products in a multiplier ofFor general n-bit multiplier, like the proposed 8-bit multiplier described above, the columns are partitioned into three sections. The truncated section includes column 1 to n/2. The middle columns are ((n/2)+1) to ((3n/2)−1) and the upper columns include column (3n/2) to (2n-3).

D. Two Proposed Approximate 8-Bit Multipliers'Structures

This section describes the structures of the two proposed approximate multipliers, called proposed-mul1 and proposedmul2. In the multipliers, four least significant columns of partial products are eliminated. The AC6Gs are used for eleventh to thirteenth column because of their great precision. The ACFGIs and ACFGIIs are used for columns 5 to 10 in the proposed-mul1 and proposed- mul2.

The proposed-mul1 structure is depicted in Fig. 9. The compressors are numbered in Fig. 9, which are according to Table III for ACFGI and Table II for AC6G compressors.

Since sum output of ACFGI compressors is constant and equals to '1' in the middle columns, one of the final output vectors is equal to '1'. Therefore, modified and simpler full-adder and half-adder are used for sum of the two final vectors as shown in Fig. 9. In [16], the structure of full adder and half adder, with a constant '1' as input have been proposed. According to [16], the implementation of the half adder only requires one NOT gate, and the implementation of the full adder only requires one XOR and one OR gate. According Algorithm 1 specifying Approximate Compressors for Different Columns of Approximate Multiplier

Let's assume
First, all of compressor (i,j) are exact i is number of stage and j is number of column k is the type of compressor
N is the operand length output: Assigning appropriate approximate compressors for compressor (i,j)
–specifying appropriate AC6G compressors for upper columns
1:for i in [stage1, stage2, ...,stage($\log_2 n-1$)]
2: for j in [column ($\frac{2\times(2^{n-1})}{3}+1$) to column ($2n-3$)]
3: for k in [AC6G_1 to AC6G_16] 4: compressor(i,j) ← k
5: if NMED(k) < NMED (k-1)
6: compressor(i,j) == k
7: else
8: compressor(i,j) == k-1
9: end if
10: end for
11: end for
12: end for
–specifying appropriate ACFGI and ACFGII compressors for middle columns–
13: for i in [stage1, stage2, ..., stage($\log_2 n-1$)] 14: for j in [column ($^{n}+1$) to column ($\frac{2\times(2^{n-1})}{3}$)]

15: for k in [ACFGI_1 to ACFGI_4] –proposed-mul1 15: (for k in [ACFGII_1 to ACFGII_12]) –proposed- mul2
16: compressor(i,j) → k 17: if NMED(k) < NMED (k-1)
18: compressor(i,j) == k
19: else
20: compressor(i,j) == k-1
21: end if
22: end for

© 2026 IJMRSET | Volume 9, Issue 4, April 2026 | DOI:10.15680/IJMRSET.2026.0904290

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

23: end for
24: end for

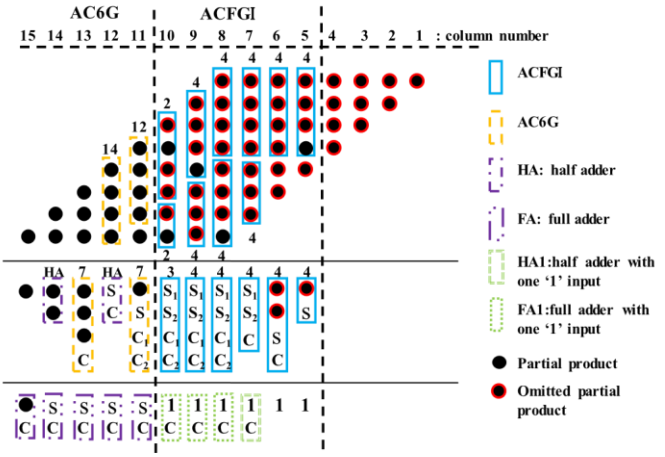


Fig. 9. Structure of the proposed-mul1, which consists of ACFGI and ac6g whose numbers according to table III and table II.

to Fig. 9, CPD of proposed-mul1 is "$t_{AND}+2t_{AC6G}+5t_{FA}$" which t_{AND}, t_{AC6G} and t_{FA} refers to delay of partial product generation step and delay of AC6G compressor and delay of full-adder, respectively.
From Fig. 9, the partial products are shown with a black dot covered by a red circle symbol, which are omitted in the proposed-mul1 approximate multiplier based on the definition of the ACFGI, see Table III. Elimination of partial products has a significant impact on the hardware optimization. Some

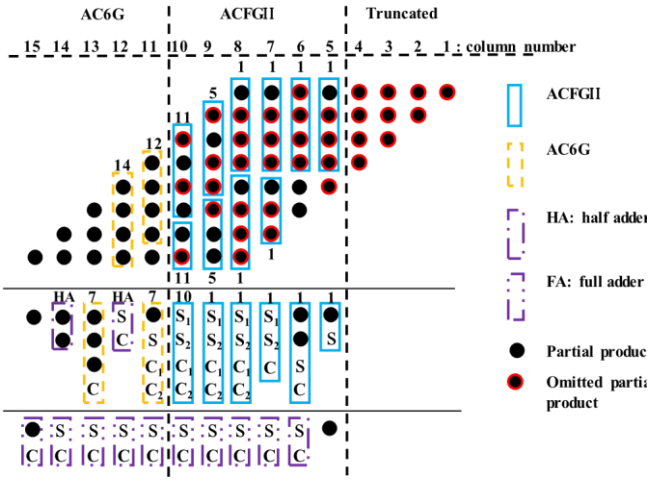


Fig. 10. Structure of Proposed-Mul2 Which Consists of ACFGII and AC6G Whose Numbers According to Table IV And Table II.

of its compressors have less than four inputs. Although earlier work has utilized exact half-adder and full-adder instead of approximate compressors with fewer inputs, the proposed multiplier uses approximate compressors with one or two inputs equal to '0' to achieve an efficient hardware multiplier.

Fig. 10 depicts the proposed-mul2 structure. ACFGII is used for the middle columns, which are numbered according to Table IV. According to Fig. 10, CPD of proposed-mul2 is

"$t_{AND} + t_{HA} + 9t_{FA}$".

© 2026 IJMRSET | Volume 9, Issue 4, April 2026 | DOI:10.15680/IJMRSET.2026.0904290

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

From Fig. 10, many bits of the partial products have not been used and included in the computations. These bits are highlighted by black dots with a red circle and labeled as the omitted partial product bits. Like proposed-mul1 method, the proposed-mul2 uses AC6G in its columns 11 to 13.

IV. EVALUATION AND COMPARISON

In this section, the proposed-mul1 and proposed-mul2 are evaluated and compared to previous studies. To demonstrate the accuracy and implementation metrics of the proposed-mul1 and 2, a FOM is introduced, which could highlight the tradeoff between precision and hardware efficiency of the designs. Moreover, to assess and verify the efficiency of the proposed designs, they were evaluated via image multiplication, sharpening, smoothing and edge detection.

A. Hardware Analysis

Table VI shows the synthesized results for circuit area, critical path delay, power consumption and the power- delay product (PDP) of the proposed-mul1, proposed- mul2, Mul_1 [13], Mul_2 [13], Multiplier1 [15], M8_1[21], M8_2 [21], Majority_based [16], C_full [19], M8 [18], ALM_SOA [26] and Exact designs. In this table, the delays are the minimum delay at which the circuits can be synthesized. The simulation condition is typical (25°C and supply voltage is 0.9V). All designs were described by VHDL (VHSIC Hardware Description Language). Implementation metrics were synthesized by the Synopsys Design Compiler (DC) using 28nm TSMC technology.

TABLE VI

HARDWARE RESULTS OF 8-BIT APPROXIMATE MULTIPLIERS

Multipliers	Delay (ns)	Power (uw)	Area (um^2)	PDP (fj)	ADP
Mul_1[13]	0.13	163.4	578.9	21.2	75.3
Mul_2[13]	0.13	155.8	554.0	20.3	72
Multiplier1[15]	0.15	247.7	845.8	37.1	126.9
M8_1[21]	0.18	217.3	688.7	39.1	124
M8_2[21]	0.16	208.9	646.6	33.4	103.5
Majority_based [16]	0.15	148.4	580.1	22.3	87
C_full[19]	0.21	275.2	804.0	57.8	168.8
M8[18]	0.22	283.5	829.8	62.4	182.6
ALM_SOA(M=5)[26]	0.14	158.4	618.1	22.1	86.53
Proposed-mul1	0.09	76.1	331.4	6.8	29.8
Proposed-mul2	0.10	73.4	318.5	7.3	34.4
Exact	0.24	313.1	922.2	75.1	221.3

From the results presented in Table VI, it is obvious that both proposed-mul1 and proposed-mul2 designs have superior implementation metrics than other existing designs. This is due to the simplicity of the proposed partial product generation and reduction method that they use.

From Table VI, Mul_2 [13], demonstrates the highest performance amongst existing methods reported in the literature and both proposed -mul1 and mul2 achieve higher performance than Mul_2 [13], as follows.

Moreover, the proposed-mul1 outperforms the mul_2[13] in terms of the delay, power, area, PDP, and ADP by 31%, 53%, 43%, 67%, and 59%, respectively. In addition, the proposed-mul2 also demonstrates higher performance to that of Mul_2 [13] in terms of the delay, power, area, PDP, and ADP by 23%, 53%, 42%, 64%, and52%, respectively.

B. Accuracy Analysis

To assess and compare the performance of the proposed approximate circuits with the existing methods, some assessment metrics such as: error rate (ER), mean relative error distance (MRED) and normalized mean error distance (NMED) are used [28]. The definition of these metrics are as follows:

Error rate (ED) metric is the difference between the exact and approximate output values, as shown in (9)

$$ED = M - M' \quad (8)$$

© 2026 IJMRSET | Volume 9, Issue 4, April 2026 | DOI:10.15680/IJMRSET.2026.0904290

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

where M, M′ denote the exact and the approximate output values and | | is the absolute value.
For an n-bit multiplier, the NMED can be calculated using (11). To calculate NMED, the Mean ED (MED) is first computed using (10). The NMED is the MED normalized by the maximum output of the exact design. The gap between exact and approximate outputs is more substantial than their relative differences in many approximate applications involving the human senses [16].

$$MED = \frac{1}{22N} \sum_{i=1}^{22N} ED_i \quad (9)$$

$$N\,MED = \frac{1}{22N(2N-1)} \sum_{i=1}^{22N} ED_i \quad (10)$$

ACCURACY RESULTS OF 8-BIT APPROXIMATE MULTIPLIERS

Multipliers	ER(%)	NMED	MRED	Max ED
Mul_1[13]	93.50	0.019	0.088	9620
Mul_2[13]	93.48	0.018	0.082	8056
Multiplier1[15]	81.79	0.025	0.079	4096
M8_1[21]	72.59	0.019	0.060	6936
M8_2[21]	72.60	0.028	0.083	1156
Majority_based [16]	99.81	0.007	0.438	1950
C_full[19]	19.02	0.003	0.007	8264
M8[18]	82.61	0.002	0.041	568
ALM_SOA(M=5)	98.80	0.013	0.055	7670
Proposed-mul1	99.93	0.018	0.509	7120
Proposed-mul2	98.86	0.017	0.151	7148

TABLE VIII

ACCURACY RESULTS OF 16-BIT APPROXIMATE MULTIPLIERS

Multipliers	ER(%)	NMED	MRED
Proposed-mul1	100	0.010	0.119
Proposed-mul2	99.98	0.009	0.066

The MRED metric represents the average relative error and can be calculated using (12):

$$MRED = \frac{1}{22N} \sum_{i=1}^{22N} \frac{ED_i}{M_i} \quad (11)$$

where M_i is the exact value of the multiplication.
The ER represents the percentage of multiplications for which the approximate design differs from its exact design counterpartMRED, ER, NMED and maximum ED metrics were calculated for 8-bit multipliers by simulating all 65536 input patterns using MATLAB software and tabulated in Table VII.

From Table VII, M8 [18] demonstrates the lowest NMED. It can be explained by the fact that the M8 uses exact compressors for most significant columns of its multiplier.

The C_full [19] method's MRED and ER values are the lowest amongst all methods. This can be explained by the fact that there is no truncation in its design.

The proposed_mul1 demonstrates the highest ER and NMED as some of its compressors generate a non-zero output

© 2026 IJMRSET | Volume 9, Issue 4, April 2026 | DOI:10.15680/IJMRSET.2026.0904290

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

when their inputs are zero. The proposed-mul1 and proposed-mul2 have nearly similar NMED value compared to Mul_2 [13].

The proposed Algorithm 1 was used to design the 16bit approximate multiplier structure. The results of both proposed multipliers for 16-bit input operands are reported in Table VIII. The results are obtained with 10 million operands with uniform distribution. In comparison to 8-bit approximate multiplier, it is found that employing compressors according to Algorithm 1 is effective in reducing the amount of error metrics.

C. Image Multiplication, Sharpening, Smoothing and Edge Detection

The quality of the proposed multipliers is examined by applying them in fault-tolerant applications. The two proposed multipliers are used to multiply two images, sharpening,

$$SSIM = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + c_2)}{(\mu_{2x} + \mu_{2y} + c_1)(\sigma_{x2} + \sigma_{y2} + c_2)} \quad (13)$$

In most image processing applications, 30 dB is considered as an acceptable value for the PSNR of the resulting images. As the difference between the exact and approximate is not distinguishable by the human eyes. The PSNR values are reported for nine image multiplication examples in Table IX.

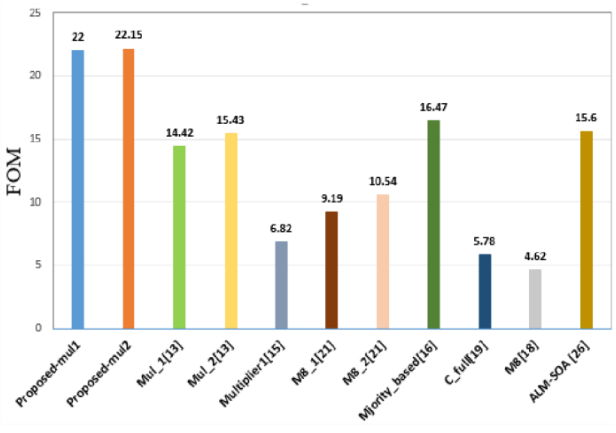
From Table IX, it can be seen that the resulting PSNRs of the images of the proposed approximation methods are above 30dBs, while the proposed designs have significantly lower hardware cost. The approximate 8-bit

Multipliers	Lake × bridge		Peppers-color × Peppers-color		Lena × female		Peppers × fruits		Female × tree		Tank × truck		House × tree		mandril × mandril		Lake-color × airplane	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Mul_1[13]	30.48	0.918	34.01	0.866	31.39	0.835	30.24	0.827	29.94	0.927	36.15	0.929	29.97	0.915	28.64	0.930	29.76	0.844
Mul_2[13]	31.68	0.920	34.47	0.891	33.19	0.896	31.11	0.837	31.64	0.865	37.17	0.942	30.55	0.932	29.20	0.935	29.43	0.836
Multiplier1[15]	27.87	0.930	29.47	0.748	30.26	0.828	28.56	0.784	27.67	0.936	35.19	0.940	24.85	0.881	24.60	0.889	23.20	0.869
M8_1[21]	32.20	0.937	34.02	0.853	31.93	0.847	32.03	0.828	31.90	0.852	36.02	0.930	31.19	0.931	30.34	0.917	31.10	0.874
M8_2[21]	29.49	0.929	28.38	0.808	28.72	0.812	28.44	0.788	27.16	0.915	33.29	0.913	28.53	0.854	24.40	0.848	28.42	0.850
Majority_based[16]	41.13	0.977	41.37	0.970	40.17	0.971	40.10	0.967	40.85	0.969	41.10	0.958	40.90	0.970	39.51	0.988	40.42	0.974
C_full[19]	44.90	0.997	38.62	0.980	44.84	0.974	42.96	0.963	42.41	0.997	55.96	0.998	42.51	0.984	38.94	0.984	41.36	0.992
M8[18]	50.15	0.998	49.54	0.997	49.87	0.997	50.54	0.996	49.91	0.997	50.97	0.997	50.20	0.997	46.53	0.999	50.72	0.997
ALM_SOA(M=5) [26]	35.01	0.958	30.42	0.858	31.38	0.859	30.52	0.825	34.60	0.908	35.38	0.905	31.57	0.942	31.52	0.957	32.13	0.886
Proposed-mul1	32.72	0.922	33.44	0.917	32.92	0.904	32.62	0.856	31.90	0.870	36.16	0.913	31.08	0.910	31.49	0.950	30.97	0.871
Proposed-mul2	32.39	0.923	33.80	0.907	32.89	0.884	32.79	0.875	31.78	0.898	35.18	0.915	31.10	0.916	31.03	0.951	30.45	0.874

smoothing and edge detection as the essential operations in image processing. MATLAB programs is used to perform image applications using the approximate multipliers. The Peak Signal to Noise Ratio (PSNR) and Structural Similarity (SSIM) of the resulting image is used as a measure to assess the performance of the multipliers. The PSNR metric is defined as:

$$PSNR = 10log\left(\frac{m \times p \times MAX_I^2}{\sum_{i=0}^{m-1}\sum_{j=0}^{p-1}(I(i \cdot j) - k(i \cdot J))^2}\right) \quad (12)$$

Where m and p are the dimensions of the images. MAX_I is the maximum possible value of image pixels and, $I(i,j)$ and $k(i,j)$ represents exact and approximate multiplication of two image pixel at location of i, j. According to [35], the SSIM is defined by (14).



multiplier [18] exhibit the highest PSNRs. This can be explained by the fact that this method uses exact compressors for most significant columns of its multiplier bits.

© 2026 IJMRSET | Volume 9, Issue 4, April 2026 | DOI:10.15680/IJMRSET.2026.0904290

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

In [13], [14], and [19], image smoothing and sharpening have been used as applications to evaluate the effectiveness of approximate multipliers. Table IX displays the minimum, maximum, and average PSNR and SSIM values. Nine images (House, Fruits, Pirate, Blonde Woman, Lena, Boat, Livingroom, Tank, Peppers) are examined to provide these results.
The Sobel operator, utilized for edge detection also implemented using the proposed multipliers. The Sobel operator has several uses in computer vision for extracting basic features [19].To perform a more comprehensive study and a fair comparison between hardware costs and error measurements, thisFig. 11.The values of FOM for approximate multipliers.paper introduces a figure of merit (FOM) by multiplying PSNR by the average of all hardware parameters delay, power and area, as shown in (15):

FOM = PSN R delay saving + area saving + power saving
× 3
(14)

The PSNR value used in the FOM calculation represents the average over all case studies. In [16], authors introduced a FOM, which combines PSNR, delay saving and power saving of the hardware. However, this FOM does not consider the area saving of the hardware. The proposed FOM, defined by (15), of the two proposed designs and references are depicted in Fig. 11.
From Fig. 11, the proposed_Mul1 and 2 methods have the highest FOM value, imply their superiority to other methods.

D. Neural Network Application

Another application of approximate multipliers is the hardware implementation of neural networks. In the following, the efficiency of the proposed multipliers in the hardware

TABLE X

THE PSNRS AND SSIM ACHIEVED FROM SHARPENING, SMOOTHING AND EDGE DETECTION WITH 8-BIT APPROXIMATE MULTIPLIERS

Multipliers	Sharpening						Smoothing						Edge detection					
	PSNR			SSIM			PSNR			SSIM			PSNR			SSIM		
	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max
Mul-1 [13]	29.12	31.23	32.44	0.981	0.987	0.984	29.81	30.62	31.97	0.939	0.941	0.962	42.39	42.92	43.97	0.945	0.948	0.957
Mul-2 [13]	32.12	33.95	35.40	0.983	0.988	0.984	34.78	33.95	37.89	0.974	0.984	0.981	41.74	42.67	44.31	0.912	0.947	0.957
Multiplier1 [15]	26.37	28.85	30.81	0.973	0.988	0.983	26.07	28.86	30.49	0.916	0.936	0.949	41.20	42.37	44.07	0.910	0.939	0.949
M8-1 [21]	29.02	32.10	33.56	0.965	0.991	0.983	27.19	31.36	35.79	0.930	0.950	0.962	43.25	44.05	44.25	0.961	0.962	0.969
M8-2 [21]	26.16	30.05	31.46	0.957	0.986	0.981	37.42	39.81	41.86	0.981	0.984	0.989	40.37	41.24	41.96	0.917	0.928	0.940
Majority-based [16]	36.10	37.18	38.66	0.988	0.994	0.992	27.25	29.54	32.85	0.956	0.961	0.969	40.55	41.02	41.34	0.797	0.818	0.861
C-full [19]	39.70	47.26	51.76	0.991	0.999	0.998	42.89	46.64	56.89	0.991	0.993	0.995	50.93	53.21	54.00	0.991	0.994	0.995
M8 [18]	47.64	48.13	49.09	0.992	0.999	0.998	39.42	40.41	41.37	0.996	0.996	0.997	58.73	59.19	60.23	0.998	0.998	0.999
ALM-SOA(M=5) [26]	32.14	36.33	38.38	0.969	0.989	0.983	32.09	35.75	37.81	0.953	0.963	0.974	43.49	44.90	45.73	0.946	0.952	0.966
Proposed-mul1	26.74	28.09	29.56	0.945	0.984	0.972	25.19	30.04	31.39	0.929	0.951	0.959	39.44	39.98	40.35	0.796	0.810	0.824
Proposed-mul2	27.51	29.32	31.28	0.967	0.986	0.980	25.60	26.24	27.71	0.951	0.956	0.962	45.62	46.36	47.68	0.965	0.973	0.979

© 2026 IJMRSET | Volume 9, Issue 4, April 2026 | DOI:10.15680/IJMRSET.2026.0904290

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

TABLE XI

ACCURACY OF 8-BIT APPROXIMATE MULTIPLIERS EMPLOYED IN HARDWARE IMPLEMENTATION OF NEURAL NETWORKS

Multipliers	Accuracy	
	MLP	CNN
Mul_1[13]	91.0%	81.5%
Mul_2[13]	90.9%	82.9%
Multiplier1[15]	91.7%	84.7%
M8_1[21]	91.1%	81.6%
M8_2[21]	91.2%	84.5%
Majority_based [16]	88.7%	82.4%
C_full[19]	91.7%	87.3%
M8[18]	89.9%	83.2%
ALM_SOA(M=5)	81.7%	79.8%
Proposed-mul1	91.1%	87.7%
Proposed-mul2	91.3%	88.1%
Exact	92.1%	88.6%

implementation of neural networks has been examined and compared with similar works. For this purpose, a multi-layer perceptron (MLP) neural network and a convolution neural network (CNN) have been implemented and the results of the implementations are given in Table X.

The implemented MLP neural network consists of two layers with 700 neurons in the hidden layer and 10 neurons in the output layer. This network is trained using the MNIST dataset [36]. The implemented CNN also includes two convolution layers and a fully connected layer in the output. This network is trained using the SVHN dataset [37].

As Table X shows, the proposed multipliers have a less than one percent drop in accuracy compared to the exact multiplier. Although the proposed multipliers are less accurate than the existing multipliers in some cases, it is noteworthy that the proposed multipliers provide high accuracy in the implementation of both networks, while some previous multipliers perform better in implementing one of the two networks and worse in one another.

V. CONCLUSION AND FUTURE SCOPE

In this work, a high-performance approximate unsigned multiplier using an optimized 4:2 compressor configuration has been proposed and analyzed. The design effectively reduces critical path delay, lowers power consumption, and optimizes hardware area, while maintaining acceptable computational accuracy. By applying selective approximation to the least significant bits, the multiplier achieves a significant improvement in speed and energy efficiency compared to conventional and existing approximate multiplier designs.

Simulation results confirm that the proposed system provides a balanced trade-off between accuracy and performance, making it suitable for a wide range of VLSI applications, including digital signal processing, image and video processing, and machine learning accelerators. Moreover, the scalable nature of the design allows it to be extended to higher bit-width operands, and the methodology can be integrated into larger DSP or AI processing pipelines for system-level efficiency improvements.

Future work can focus on incorporating more advanced error-tolerant techniques to further reduce power without significantly affecting accuracy, exploring hybrid architectures combining exact and approximate multipliers to optimize performance for specific application needs, and implementing the design on FPGA or ASIC platforms to validate real-world hardware metrics.

Additionally, the proposed multiplier can be leveraged in neural network accelerators and AI processors where minor computational errors are tolerable, thereby enhancing throughput and energy efficiency. Overall, the proposed design demonstrates a scalable, energy-efficient, and high-speed solution for unsigned multiplication in modern VLSI systems and opens avenues for further optimization and application- specific adaptations in future research.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Furthermore, the scalable architecture of the proposed multiplier allows it to be extended to higher bit- width operands, making it adaptable for more complex VLSI applications and future system-level integrations. The design methodology is also flexible enough to support hybrid architectures, performance for application-specific requirements. Future work can focus on incorporate advanced error-tolerant techniques, such as probabilistic computing or adaptive precision control, to further reduce power consumption while maintaining high computational reliability. Additionally, implementation on FPGA or ASIC platforms will allow for the evaluation of real-world hardware metrics such as dynamic power, silicon area, and maximum operating frequency, providing insights for industrial applications. Integration with pipelined processing architectures and multi-core VLSI systems can further enhance performance and efficiency, paving the way for next- generation low-power, high-speed computing platforms.

REFERENCES

[1] J. Gu and Y. Kim, "Design and Analysis of Approximate 4-2 Compressor for Efficient Multiplication," IEIE Trans. Smart Processing & Computing, vol. 11, no. 3, pp. 162–168, 2022.
[2] H. Seok et al., "Design Optimization of a 4-2 Compressor for Low-Cost Approximate Multipliers," IEIE SPC, vol. 11, no. 6, pp. 455–461, 2022.
[3] Y. Xu, Y. Guo, and S. Kimura, "Approximate Multiplier Using Reordered 4-2 Compressor with OR-Based Error Compensation," in Proc. IEEE ASICON, 2019.
[4] S. Venkatachalam and S. Ko, "Design of Power and Area Efficient Approximate Multipliers," IEEE Trans. VLSI Systems, vol. 25, no. 5, pp. 1782–1786, 2017.
[5] X. Wang and W. Qian, "MinAC: Minimal-Area Approximate Compressor Design Based on Exact Synthesis," in Proc. IEEE ISCAS, 2022.
[6] V. Zanandrea et al., "Ultra-Compact Approximate 4:2 Compressor for Power-Efficient Multipliers," in Proc. IEEE LASCAS, 2025.
[7] L. S. Koneru et al., "Optimised Power-Efficient Design of Approximate Multiplier Using Approximate Compressor," in Proc. IEEE ICSC, 2025.
[8] L. S. Koneru et al., "Compressor-Based Approach for Power- Optimized Error-Tolerant Approximate Multipliers," in Proc. IEEE ICSC, 2025.
[9] F. Ranjbar et al., "High Performance 8-bit Approximate Multiplier Using Novel 4:2 Compressors," Int. J. Integrated Engineering, 2018.
[10] U. A. Kumar and S. E. Ahmed, "Compressor-Based Approximate Multiplier Architectures for Media Processing Applications," IJECE, 2020.
[11] M. K. Sukla et al., "A High-Speed and Low-Leakage Inexact Compressor-Based Approximate Multiplier with Error Encoding Logic," IETE Journal of Research, 2025.
[12] "Design and Analysis of Approximate 4–2 Compressors for High- Accuracy Multipliers," IEEE Xplore, 2021.
[13] M. A. Shafieabadi et al., "An Approximate 4-2 Compressor Based on Spintronic Devices," Int. J. Modern Electronics and Computer Science, 2019.
[14] "Energy and Area Efficient Imprecise Compressors for Approximate Multiplication," AEU - Int. J. Electronics and Communications, 2019.
[15] "Area-Efficient and High-Performance Approximate Multiplier Based on 4:2 Compressors," Integration, VLSI Journal, 2026.
[16] B. Bagheralmoosavi et al., "Power-Area Efficient IMPLY-Based 4:2 Compressor for Multipliers," 2024.
[17] P. Jaswal et al., "Low Power Approximate Multiplier Architecture for Deep Neural Networks," 2025.
[18] L. H. Krishna et al., "Approximate Signed Multiplier with Sign- Focused Compressor," 2025.
[19] E. Farahmand et al., "ScaleTRIM: Scalable Approximate Multiplier with Compensation," 2023.
[20] S. Gupta et al., "Low-Power and High-Accuracy Approximate Multiplier with Reconfigurable Truncation," 2021.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH

IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com