Design of Error Efficient Inaccurate 3:2 Compressor for Approximate 8×8 Multiplier

Srinivasa Gupta Noolu, M. Satya Anuradha, and Krishna Veni Sahukara

***Abstract*—Energy-efficient arithmetic units are critical for modern error-resilient computing systems. This work introduces a novel inaccurate 3:2 compressor that exploits input probability characteristics to reduce complexity by allowing a single, low-probability error condition. The proposed approximation alters only the sum output for the all-one input combination, while maintaining an exact carry output. The compressor is integrated into a radix-4 methodology based 8×8 multiplier realized through a hierarchical composition of 2×2 and 4×4 multiplier modules. Three alternative architectures are developed for intermediate partial product accumulation using the proposed inexact 3:2 compressor. Error behavior is analyzed in MATLAB and benchmarked against existing inaccurate designs using Error Rate (ER), Mean Error Distance (MED), and Normalized Mean Error Distance (NMED). Hardware implementations are carried out in Cadence Spectre with GPDK 45-nm technology to evaluate area, power consumption, and delay. Experimental results indicate that the proposed multiplier provides favorable accuracy–efficiency trade-offs. Application-level validation using image multiplication and image convolution further demonstrates improved perceptual quality measured through PSNR and SSIM.**

***Index Terms*—Approximate Multiplier, Error Metrics, inaccurate 3:2 compressor, Image Processing.**

Original Research Paper
DOI: 10.53314/ELS2630074N

I. Introduction

THE widespread adoption of multimedia applications on portable electronic platforms has created a strong demand for efficient digital signal processing (DSP) architectures. DSP blocks constitute the fundamental components of such applications, enabling computational tasks including image and video processing. Since the human visual system cannot detect every subtle variation in images or videos, outputs that are nearly accurate are often acceptable [1]. This tolerance opens the door to approximate computing, a design paradigm where slight inaccuracies in results can lead to substantial power savings.

Although Digital Signal Processing (DSP) and multimedia applications remain significant beneficiaries of approximate computing, the rapid expansion of Deep Learning has become the principal catalyst for the adoption of approximate arithmetic. Modern architectures, such as Convolutional Neural Networks (CNNs) and Large Language Models (LLMs), require the execution of millions of Multiply–Accumulate (MAC) operations, resulting in substantial computational demands [2]. Owing to the inherent resilience of neural networks to minor arithmetic inaccuracies, approximate multipliers can be employed to achieve significant reductions in area and energy consumption while maintaining high predictive accuracy and inference quality [3].

Binary arithmetic is fundamental to DSP operations, with addition, subtraction, multiplication, and division relying heavily on adders [4]. Over recent years, various forms of inexact adders have been introduced in the literature. One of the earlier designs, known as the Lower-part OR Adder (LOA), described in [5], applies exact addition for higher-order bits while employing simple OR gates for the lower bits. In order to generate the carry-in for the accurate higher significant part, an AND gate has been introduced. While LOA offers low power consumption and reduced area, it suffers from a high error rate.

Another architecture, the Error Tolerant Adder-1 (ETA-1) proposed in [6], improves upon LOA by using modified XOR gates in the lower-bit section and standard adders for higher bits. Despite enhancements in performance, ETA-1 still exhibits notable inaccuracies. Napoli *et.al.* [7] provides an extensive evaluation of 86 approximate full adders, categorizing them into four distinct families based on their logic functions and transistor-level implementations. Using 45-nm CMOS technology, the researchers analyzed each design's trade-offs between hardware efficiency (area, power, delay) and error metrics to identify the most effective circuits for specific use cases. The analysis concludes by recommending the best-performing adders for different optimization goals, such as the overall most efficient designs and those best suited for high-precision or low-power applications.

Multiplication, another key operation in DSP algorithms, can also be approximated at different design levels. For wide-width operands, hybrid radix Booth encoding is often utilized to produce partial products [8]. Consequently, several works have focused on designing approximate Booth decoders [9]–[12]. For smaller-width multiplications, partial products are usually summed using various compressor structures [13]–[16], which have also been targeted for approximation to enhance power and area efficiency.

The proposed methodology of [17] optimizes Mitchell's logarithmic multiplication for convolutional neural network

Manuscript received on January 26th, 2026. Received in revised form on April 30th and June 28th, 2026. Accepted for publication on July 19th, 2026.

Srinivasa Gupta Noolu, ECE Department, Andhra University College of Engineering, Andhra University, Visakhapatnam 530003, India (email: srinivasgupta.486@gmail.com, ORCiD: 0000-0001-9747-0292).

M. Satya Anuradha, ECE Department, Andhra University College of Engineering, Andhra University, Visakhapatnam 530003, India (email: radhamsa@gmail.com, ORCiD: 0000-0001-8314-2132).

Krishna Veni Sahukara, Department of ECE, GVPCDPGC (A), Visakhapatnam, Andhra Pradesh, India (email: drkrishnaveni@gvpcdpgc.edu.in, ORCiD: 0000-0001-6151-2059).

(CNN) inference by utilizing a customizable truncation parameter (w) to reduce mantissa bit processing and a one's complement approximation for energy-efficient handling of negative numbers. These hardware optimizations achieve significant energy savings with minimal impact on accuracy. Liu *et.al.* [18] introduces a methodology that enhances the standard Mitchell logarithmic multiplication by using a piecewise linear approximation and data truncation to balance hardware efficiency with computational accuracy. The primary outcome that achieves lower MRED and NMED and better visual quality in image processing tasks compared to state-of-the-art approximate multipliers, though it occupies a slightly larger circuit area and consumes more power than the simplest existing alternatives.

In this work, an inaccurate 3:2 compressor architecture is introduced by inserting errors in the lower probable input patterns and systematically analyzed. A detailed error characterization is performed using MATLAB and the results are compared against several previously published inexact compressor designs. The proposed compressor is subsequently integrated into a radix-4–based 8×8 multiplier for the accumulation of intermediate partial product (IPP) bits. Error behavior of the resulting approximate multiplier is evaluated and benchmarked against existing inexact compressor–based multiplier implementations through MATLAB simulations. To examine circuit-level performance, the 8×8 multiplier is realized in Cadence Spectre, where area, average power consumption and critical path delay are extracted using GPDK 45-nm CMOS technology. In addition, the practical applicability of the proposed inexact multiplier is demonstrated through an image multiplication and image convolution experiments, with output quality assessed using Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM). The main contribution of the paper is summerized below:

- The probability of occurrence of the input bit patterns in the partial product column are determined.
- Probability of each pattern (000 to 111) are estimated and discussed. Based on lower probable input pattern errors are introduced in the truth table of the proposed inaccurate 3:2 compressor.
- Errors are intentionally introduced only in the Sum output, since any inaccuracy in the Carry would not only corrupt the Sum but also propagate to subsequent stages, thereby amplifying the overall error.
- Proposed inaccurate 3:2 compressor is used in Radix-4 methodology based 8-bit multiplier to add the partial product bits. Error metrics and Circuit parameters are estimated.
- Proposed inaccurate 3:2 compressor is used in Radix-4 methodology based 8-bit multiplier and validated through image multiplication and convolution application.

The remainder of this paper is organized as follows: Section II summerizes different state-of-the-art designs. Section III discusses the design methodology of the proposed inaccurate 3:2 compressor. In Section IV, the design of the 8×8 multiplier using 4×4 and 4×4 using 2×2 is discussed, followed by the integration of the proposed compressor within the 8×8 multiplier architecture. Section V discusses the estimation of simulation results and comparison of performance metrics. Section VI focuses on validating the proposed inaccurate 3:2 compressor based Approximate 8×8 multiplier through an application— Image Multiplication, and analyzes the results using image quality metrics. Finally, Section VII concludes the paper.

II. Related Work

In literature, Beura *et.al.*, [19] proposes multi-bit inexact adders (2-bit, 4-bit, 8-bit) using K-map simplification and logic reduction to improve power and delay. Results show reduced error metrics, lower PDP, and improved performance in 45 nm technology. The design is validated through image de-noising, achieving better PSNR, NCC, and SSIM. Yang *et.al.*, [20] proposes low-power approximate adders using XOR/XNOR and pass transistor logic, achieving reduced area, delay, and PDP. AXA1 is fastest, AXA2 smallest, and AXA3 most accurate. However, designs suffer from voltage degradation and are limited to single-bit analysis. [21] proposes inexact adders (InXA1 – InXA3) using cell replacement to reduce power and delay. InXA2 shows the best performance with low error and high efficiency. However, use of pass logic creates swing issue.

In [22] authors presents a reversible logic-based Haar wavelet and lifting scheme using approximate full adders (RAFA-1, RAFA-2) to reduce quantum cost, gate count, and power. It achieves efficient hardware performance with acceptable image quality validated on FPGA. Ramaswamy *et.al.*, [23] proposes carry-based approximate full adders (CBAA) by simplifying logic and avoiding complex XOR operations, replacing them with basic AND/OR gates to reduce area and power. Simulation results show significant improvements. Multiple variants (CBA1–CBA4) offer different trade-offs between error and performance. However, the design lacks validation in real application scenarios like image processing or DSP systems.

Mehrabani *et.al.*, [24] introduces four novel approximate full adders (DAFA1 – DAFA4) that utilize dynamic logic and Carbon Nanotube Field-Effect Transistors (CNFETs). By iteratively simplifying the circuit architecture, the designs shows improvement in the Power-Delay-Area Product (PDAP) compared to existing state-of-the-art models. Simulation results demonstrate that these compact adders maintain high performance in image sharpening tasks while significantly reducing energy consumption. [25] introduces eight low-cost approximate full adders (LCAFAs) designed to optimize hardware performance in error-tolerant applications like image processing. The designs use a novel clock-control signal approach to minimize transistor counts, resulting in average area and delay reductions. Ziad *et.al.*, [26] introduces a static approximate adder that balances energy efficiency and accuracy by using a segmented Least Significant Module (LSM). The architecture utilizes constant truncation for the lowest bits and half-adders for the upper LSM bits to effectively predict carries and reduce hardware overhead. Simulation results show the design achieves the lowest energy consumption among its peers while maintaining high-quality image reconstruction.

Several approximate multipliers based on different inaccurate compressors have been proposed in the literature during the last 10 years. In [13], Momeni *et al.* introduced two approximate 4:2 compressor architectures in which the carry-in (Cin) and carry-out ($Cout$) signals were deliberately omitted to achieve lower propagation delay and reduced power dissipation. These compressor designs were subsequently incorporated into a Dadda multiplier framework to evaluate their effectiveness. Yang *et al.* [14] proposed three approximate 4:2 compressors using Karnaugh map simplification. Among them, ACCI1 offers the highest accuracy, producing erroneous outputs only when all inputs are high, albeit with greater area overhead due to its higher gate count. Akbari *et al.* in [16], have introduced four approximate 4:2 compressors with dual operating modes: inexact and supplementary. The inexact mode minimizes power consumption by activating only the approximate circuitry, whereas the supplementary mode enhances accuracy by enabling additional exact logic. A power-gating mechanism is employed to switch between the two modes.Strollo *et al.* [27] proposed three approximate compressors based on the stacking-circuit methodology. Among them, the second design exhibits only two erroneous output cases, offering superior error performance; however, its power consumption and area overhead remain relatively high.

Approximate computing primarily aims to minimize circuit area, energy consumption, and processing latency by allowing a controlled level of inaccuracy, making it suitable for error-resilient applications. Accordingly, the key objective of the proposed design is to achieve an optimal trade-off between computational accuracy and hardware overhead. Based on this principle, an inaccurate 3:2 compressor has been developed, as elaborated in the subsequent section.

III. Design of Proposed Inaccurate 3:2 compressor

Over the past decade, extensive research has been conducted in the field of inexact computing to develop low-power, compact circuit designs. However, maintaining high accuracy and operational speed continues to be a significant challenge. In this work, the proposed compressor-based multiplier, designed using the Radix-4 algorithm, demonstrates improved speed of operation and power efficiency, while also offering improved accuracy and faster performance compared to existing designs reported in the literature [28].

A. Conventional 3:2 Compressor

The conventional 3:2 compressor, also referred to as a 1-bit full adder, accepts three input signals and produces two outputs: Sum and $Carry$, as defined in Eqs. (1) and (2). The logic implementation of the exact 3:2 compressor, illustrated in Fig. 1, involves a significant number of circuit elements, which leads to relatively high power consumption, as noted in [29].

$$Sum = A \oplus B \oplus Cin \tag{1}$$

$$Carry = AB + BCin + ACin \tag{2}$$

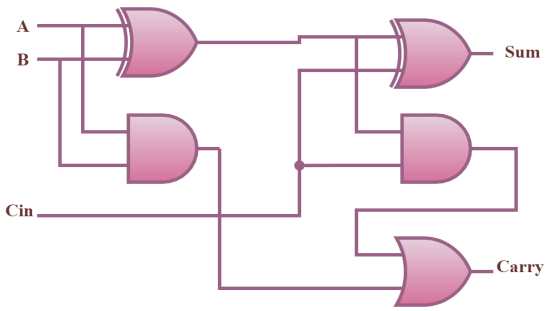

Fig. 1. Conventional 3:2 Compressor.

B. Proposed Inaccurate 3:2 Compressor

The proposed inaccurate 3:2 compressor is derived from the conventional mirror adder structure, with a controlled modification introduced to reduce circuit complexity and power consumption. Let the three binary inputs be denoted as A, B, and Cin, and the corresponding outputs be the Sum (S_M) and $Carry$ (C_M). For an exact 3:2 compressor (1-bit mirror adder), the outputs are defined as:

$$C_M = AB + BCin + ACin \tag{3}$$

$$S_M = (A + B + Cin)\overline{C_M} + ABCin \tag{4}$$

In the proposed inaccurate 3:2 compressor, the carry output is preserved exactly as in the conventional expression, whereas the sum output is approximated by modifying only a single input combination to reduce hardware complexity. Statistical observations indicate that the probability of the all-zero input pattern is the highest, while the probability of the all-one input pattern is the lowest. Motivated by this distribution, the approximation is introduced only for the least probable case. Specifically, when all three inputs are logically high ($A = B = Cin = 1$), the exact implementation yields a Sum output of '1', but the proposed design forces the Sum output to '0'. It can be noted that the Sum output is dependent on the $Carry$ output. Any inaccuracy in the $Carry$ will consequently introduce an error in the Sum. Furthermore, errors in the $Carry$ may propagate to subsequent stages, thereby significantly degrading the overall error performance metrics. Accordingly, the inaccurate Sum (S_{ix}) and $Carry$ (C_{ix}) functions are defined as follows:

$$C_{ix} = AB + BCin + ACin \tag{5}$$

$$S_{ix} = (A + B + Cin)\overline{C_{ix}} \tag{6}$$

$$S_{ix} = \begin{cases} 0, & if A = B = Cin = 1 \\ A \oplus B \oplus Cin, & otherwise \end{cases} \tag{7}$$

Only one error is introduced in the Sum output (highlighted in yellow), occurring in the input combination (1,1,1), as shown in Table I.

TABLE I
TRUTH TABLE OF PROPOSED INACCURATE 3:2 COMPRESSOR

A	B	Cin	C_M	S_M	C_{ix}	S_{ix}
0	0	0	0	0	0	0
0	0	1	0	1	0	1
0	1	0	0	1	0	1
0	1	1	1	0	1	0
1	0	0	0	1	0	1
1	0	1	1	0	1	0
1	1	0	1	0	1	0
1	1	1	1	1	1	0

C. Probability-Based Justification

Assuming the input bits A, B, and Cin are independent and identically distributed with probability $P(A = 1) = P(B = 1) = P(Cin = 1) = 0.5$, the probability that all three inputs are high is:

$$P(A = 1, B = 1, Cin = 1) = (1/2)^3 = 1/8$$

Thus, the probability of an erroneous output is:

$$P_{error} = \tfrac{1}{8} = 12.5\%$$

This low error probability ensures that the proposed modification has negligible impact on overall accuracy, while significantly reducing design complexity of the sum generation circuitry. Since the carry output remains exact, error propagation in higher-order arithmetic stages is minimized, making the design suitable for approximate multipliers and DSP applications where full precision is not mandatory.

Table II describes the probability of occurrences of 3-bit input patterns in the columns of the partial product bit matrix. Group A contains only one pattern where all the bits are '0's. Group B has 3 patterns where there are two '0's and one '1' in a column of 3 bits. Group C has 3 patterns where one '0' and two '1's are there. Group D has only one pattern with all '1's. Group-A contains only 1 input pattern (000) with a probability of occurrence of 37.5%. Group-B contains three input patterns (001, 010, 100) and combinedly their probability is 45.31%. Similarly, Group-C contains three input patterns (011, 101, 110) with a net probability of occurrence of 15.62%. The last Group-D has only one input pattern of all '1's having probability of occurrence of 1.56%. This analysis shows that the probability of occurrence of maximum number of one's in the input pattern is lowest and it increases as the number of one's decreases. Therefore, insertion of errors in Group D or C could offer improvement in the error metrics.

D. Error Metrics

The concept of approximate computing is based on the principle that substantial improvements in performance pa-

TABLE II
PROBABILITY OF OCCURRENCE OF INPUT BIT PATTERNS

Groups	Patterns	Number of occurrences*	Probability of occurrence x 100
A	000	24576	37.5
B	001, 010, 100	29696	45.3125
C	011, 101, 110	10240	15.625
D	111	1024	1.5625

*Refers to the number of occurrences in 65536 possible input conditions responsible for generating a 3-bit column of partial product matrix.

the intentional introduction of errors [30]. To quantify this uncertainty, suitable error metrics are employed. The gate-level architectures of both the previously reported 3:2 compressors [19]–[25] and the proposed design are modeled and simulated in MATLAB. Existing state-of-the-art studies have introduced multiple design variants; for benchmarking purposes, the most optimal configurations reported in those works have been selected for comparison. The outputs generated by the proposed inaccurate 3:2 compressor are subsequently evaluated and benchmarked against those of existing architectures to assess relative performance variations.

The accuracy of the proposed 3:2 compressor and its deviation from exact arithmetic are quantified using the error metrics defined in Eqs. (8)–(12).

- Error Distance (ED): It is the arithmetic difference between the accurate output (p) and the inaccurate output (q) [30].

$$ED = |p - q| = \left| \sum p[j] \times 2^j - \sum q[k] \times 2^k \right| \quad (8)$$

- Error Rate (ER): It is defined as the number of incorrect outputs generated with respect to all possible outcomes [30].

$$ER = \frac{Number\,of\,incorrect\,outputs}{Total\,number\,of\,outputs} \times 100 \quad (9)$$

- Total Error Distance (TED): It is the absolute sum of error distance [30].

$$TED = \left| \sum ED \right| \quad (10)$$

- Mean Error Distance (MED): It is the average ED for a set of outputs [30].

$$MED = \frac{TED}{Total\,number\,of\,outputs} \quad (11)$$

- Normalized Mean Error Distance (NMED): It is the normalized MED [30].
- Number of effective Bits (NoEB): To check whether the number of bits is free from error, NoEB is used instead of the root mean square of the errors (ERMS) [15].

$$NoEB = 2n - log_2(1 + ERMS) \quad (12)$$

Table III shows that the proposed inaccurate 3:2 compressor offers ER of 12.5% with only one error. Reported designs [19]–[22], [25] offer 25% ER with two errors. The proposed

TABLE III
ERROR METRICS COMPARISON OF PROPOSED AND REPORTED INACCURATE 3:2 COMPRESSORS

Designs	Error	TED	ER	MED	NMED
Proposed	1	1	12.5	0.125	0.0416
Beura [19]	2	2	25	0.25	0.0833
Yang3 [20]	2	2	25	0.25	0.0833
Almurib2 [21]	2	2	25	0.25	0.0833
Raveendran1 [22]	2	4	25	0.5	0.1667
Raveendran2 [22]	2	4	25	0.5	0.1667
Ramasamy [23]	5	5	62.5	0.625	0.2083
Mehrabani4 [24]	3	3	37.5	0.375	0.1250
Yan4 [25]	2	4	25	0.5	0.1667

IV. DESIGN OF APPROXIMATE 8×8 MULTIPLIER USING PROPOSED INACCURATE 3:2 COMPRESSOR

After evaluating the performance of the proposed 3:2 compressor, it is further integrated into a higher-level arithmetic unit to demonstrate its practical applicability. The approximate 8×8 multiplier is designed using the radix-4 approach for performing multiplication between two numbers. In this design, multiple 2×2 multipliers are combined to construct a 4×4 multiplier, as illustrated in Fig. 2. Subsequently, four such 4×4 multipliers are utilized to realize an 8×8 multiplier architecture. Hence, a total of sixteen 2×2 multiplier units are needed to design the complete 8×8 multiplier. All 2×2 multipliers operate concurrently, enabling parallel computation and thus minimizing propagation delay.

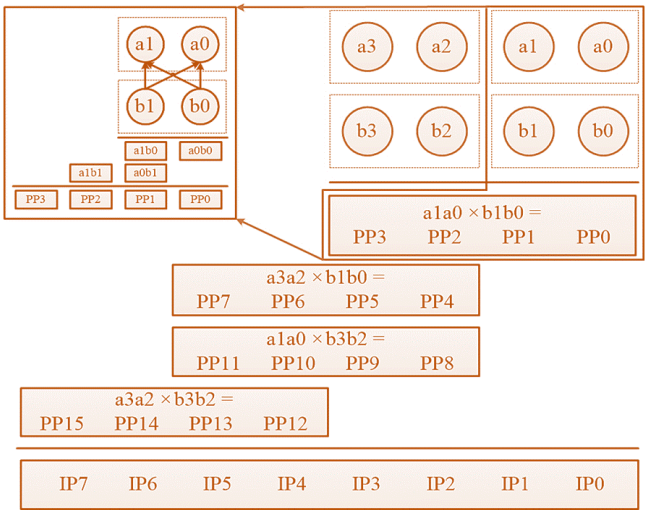

Fig. 2. Design of 4-bit Multiplier Using Four 2-bit Multipliers.

In the 2×2 multiplier structure, the input bits a1, a0 serve as the multiplicand, while b1, b0 represent the multiplier. Partial product (PP) is generated using vertical and crosswise multiplication of these bits. These partial products are then combined through addition to produce PP3, PP2, PP1, and PP0. In a similar manner, multiplying a3 and a2 with b1 and b0 yields PP11 through PP8, and the process continues for higher-order bits. The complete set of partial products from PP15 to PP0 is subsequently added to form the intermediate results IP7 through IP0, as illustrated in Fig. 2.

The 8×8 multiplier is realized using four parallel 4×4 multiplier blocks, as shown in Fig. 3. Each of these blocks produces its corresponding intermediate partial product (IPP) bits (IP31 to IP0) concurrently. The final multiplication result is obtained by combining these intermediate values through an addition stage that employs the proposed inaccurate 3:2 compressors. In this architecture, IP0 to IP3 are directly truncated to generate output bits P0 to P3, while IP4 to IP27 are summed using inaccurate 3:2 compressor units.

Since P0 corresponds to the least significant position and the significance increases progressively up to P15, incorporating inexact circuits within the multiplier requires careful consideration to limit error propagation. Fig. 4 presents three architectural options for integrating the inaccurate 3:2 compressor. In Architecture-1, the IPPs enclosed within the dotted region are combined using inaccurate 3:2 compressors. Architecture-2 and Architecture-3 follow a similar methodology, employing inexact compressors to sum the partial products located within their respective dotted regions.

V. PERFORMANCE ESTIMATION AND COMPARISON OF APPROXIMATE 8×8 MULTIPLIER

The approximate 8×8 multiplier is modeled and simulated in MATLAB. For comparative evaluation, the exact 3:2 compressors used in the partial product accumulation stage are replaced with the proposed design as well as previously reported inaccurate compressors [19]–[25]. The resulting error metrics obtained for the different multiplier architectures discussed in the preceding section are summarized in Tables IV–VI.

A. Estimation of Error Metrics and Comparison

The proposed and reported [19]–[25] inaccurate 3:2 compressors are utilized to add the intermediate partial products as shown in Fig. 4 (Architecture-1). The estimated error metrics is shown in Table IV.

TABLE IV
ESTIMATED ERROR METRICS OF APPROXIMATE 8×8 MULTIPLIER FOR ARCHITECTURE-1

Designs	ER	MED	NMED	NoEB
Proposed	21.0151	14.2	0.0003	12.3112
Beura [19]	39.8438	23.5	0.0008	10.4388
Yang3 [20]	71.9116	63.75	0.0022	9.4625
Almurib2 [21]	52.1851	48.75	0.0017	9.6782
Raveendran1 [22]	73.4619	128.5176	0.0044	8.4688
Raveendran2 [22]	63.5010	89.0332	0.0031	8.8918
Ramasamy [23]	95.4163	96.2954	0.0033	9.1239
Mehraban4i [24]	77.6134	74.5942	0.0028	9.8787
Yan4 [25]	61.2287	87.4952	0.0030	9.6622

Table IV shows the estimated error metrics for Architecture-1. It is observed that the proposed inaccurate 3:2 compressor based approximate 8-bit multiplier offers 46.15%

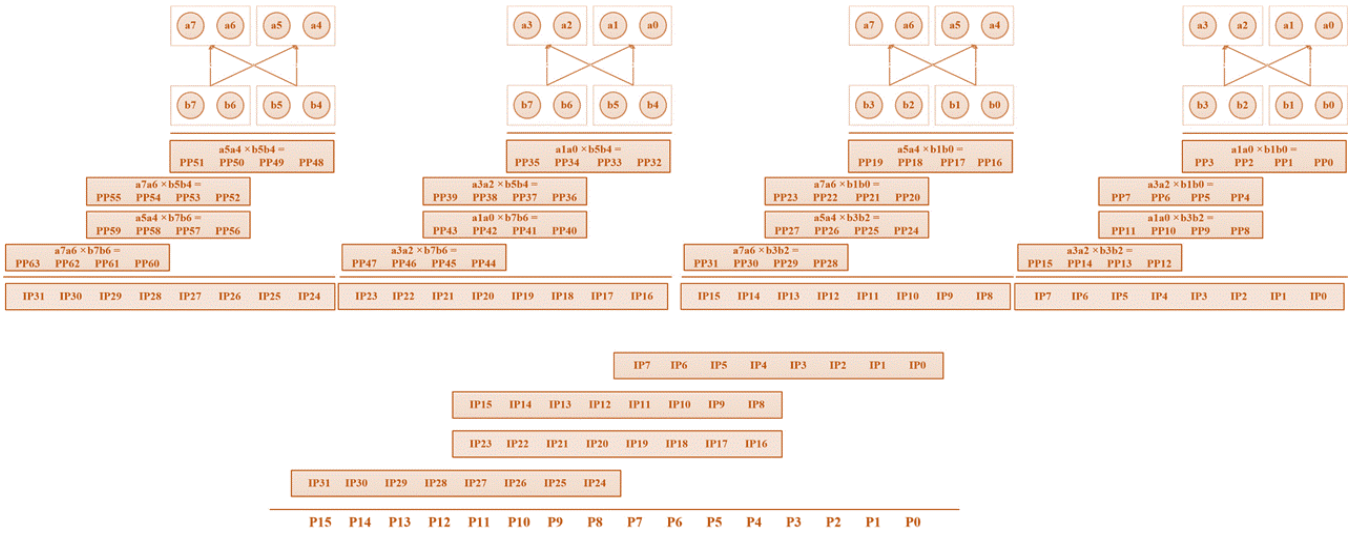

Fig. 3. Detailed Architecture of 8×8 Multiplier.

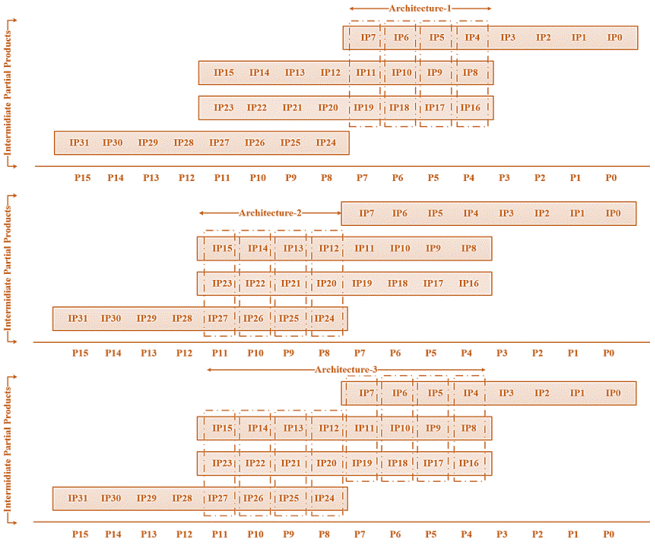

Fig. 4. Placement of Inaccurate 3:2 Compressors in the IPP-bit Array.

improvement in ER, 39.5% improvement in MED, and 62.5% improvement in NMED over best reported counter-part [19]. The proposed design based approximate multiplier also offers 2-bits improvement in NoEB.

Table V shows the estimated error metrics for Architecture-2. The error metrics summarized in Table V indicate that the design reported in [19] achieves the lowest ER, along with better MED and NMED values compared to the other existing approaches. In contrast, the proposed inaccurate 3:2 compressor–based multiplier offers further improvement across these parameters, outperforming the design in [19].

The error parameter evaluation for Architecture-3 is presented in Table VI. This is the most erroneous architecture as all possible IPs are added using inaccurate 3:2 compressors. The approximate 8×8 multiplier designed using the proposed inaccurate 3:2 compressor achieves significantly better accuracy metrics compared to the other reported architectures, with the design in [19] ranking next in performance.

TABLE V
ESTIMATED ERROR METRICS OF APPROXIMATE 8×8 MULTIPLIER FOR ARCHITECTURE-2

Designs	ER	MED	NMED	NoEB
Proposed	26.2122	127.82	0.0007	9.4211
Beura [19]	42.0776	390.19	0.0134	6.4241
Yang3 [20]	66.9434	672.19	0.0231	5.9805
Almurib2 [21]	44.0338	520.56	0.0179	6.1080
Raveendran1 [22]	78.1052	2747.50	0.0945	4.1686
Raveendran2 [22]	55.7312	1078.90	0.0371	4.7484
Ramasamy [23]	95.4254	1532.20	0.0527	5.1319
Mehrabani4 [24]	81.1346	873.248	0.0248	5.7455
Yan4 [25]	53.2782	1059.45	0.0310	4.6841

TABLE VI
ESTIMATED ERROR METRICS OF APPROXIMATE 8×8 MULTIPLIER FOR ARCHITECTURE-3

Designs	ER	MED	NMED	NoEB
Proposed	37.7451	255.14	0.0033	8.2535
Beura [19]	60.5469	413.69	0.0142	6.3860
Yang3 [20]	89.2944	735.94	0.0253	5.9172
Almurib2 [21]	68.4647	569.31	0.0196	6.0592
Raveendran1 [22]	92.4118	2853.70	0.0982	4.1291
Raveendran2 [22]	82.5089	1120.50	0.0385	4.7364
Ramasamy [23]	99.7284	1556.20	0.0535	5.1083
Mehrabani4 [24]	92.3261	1082.92	0.0333	4.7482
Yan4 [25]	80.2181	1101.48	0.0380	4.6216

B. Estimation and Comparison of Circuit Parameters of Approximate 8×8 Multipliers

Architecture-3 is chosen for the evaluation of circuit-level performance parameters, namely area, average power consumption, and minimum propagation delay. Architectures 1 and 2 employ four inaccurate 3:2 compressors for processing the input operands (as shown in Fig. 4), resulting in identical area and power characteristics. Architecture 3, in contrast, replaces all eight accurate 3:2 compressors with inaccurate counterparts, thereby achieving the lowest area and power consumption relative to Architecture-1 and Architecture-2. A detailed Pareto analysis comparing all three architectures is presented in the subsequent subsection. As depicted in Fig. 4, Architecture-3 utilizes eight exact 3:2 compressors for intermediate partial product accumulation; these units are substituted with the proposed inaccurate 3:2 compressors to assess performance impact. In addition, several previously reported inexact compressor designs [19]–[25] are also integrated for comparative analysis, allowing the generation of corresponding approximate outputs of the 8×8 multiplier. All configurations are implemented and simulated in Cadence Spectre using the GPDK 45-nm CMOS technology to extract the relevant performance metrics.

TABLE VII
ESTIMATED CIRCUIT PARAMETERS OF APPROXIMATE 8×8 MULTIPLIER FOR ARCHITECTURE-3

Designs	Area (μM^2)	Power (μW)	Delay (ns)	PDP (pJ)	ADP (fM^2s)
Proposed	122	8.3	14.3	118.7	1744.6
Beura [19]	115	7.3	13.8	100.74	1587
Yang3 [20]	131	8.2	14.8	121.36	1938.8
Almurib2 [21]	127	7.8	14.5	113.12	1841.5
Raveendran1 [22]	152	8.7	15.3	133.11	2325.6
Raveendran2 [22]	155	8.6	15.1	129.86	2340.5
Ramasamy [23]	178	9.1	15.6	141.96	2776.8
Mehrabani4 [24]	129	8.1	14.6	118.26	1883.4
Yan4 [25]	118	7.8	14.1	109.98	1663.8

From Table VII, it is observed that the reported inaccurate 3:2 compressor-based multiplier [19] exhibits 5% less area, 12% lower power dissipation and 3.5% faster operation compared to the proposed inaccurate 3:2 compressor based approximate 8×8 multiplier. Owing to the compact 6-transistor implementation of the inexact 3:2 compressor in [25], it achieves approximately 3.2% reduction in area, 6.02% lower power consumption, and a 1.4% decrease in computational delay compared to the proposed inexact 3:2 compressor-based 8×8 approximate multiplier. However, the proposed design offers improved circuit parameter results over other state-of-the-art designs reported in [20]–[24].

C. Pareto Analysis

The Pareto plot in Fig. 5 evaluates various architectures used in the radix-4 methodology based approximate 8×8 multipliers by considering PSNR as the accuracy metric and PDP as the performance indicator. Architecture-1 and Architecture-2 are also implemented in Cadence Spectre for extracting circuit parameters. Designs located toward the top-left region of the plot are more desirable, as they exhibit higher accuracy along with lower PDP. The analysis demonstrates that, [19] located in the left side of the plot, but due to less PSNR value it resides in the middle. In comparison with existing state-of-the-art approaches, the proposed inexact 3:2 compressor-based 8-bit approximate multiplier occupies top-middle position which provides an improved balance between computational accuracy and energy efficiency.

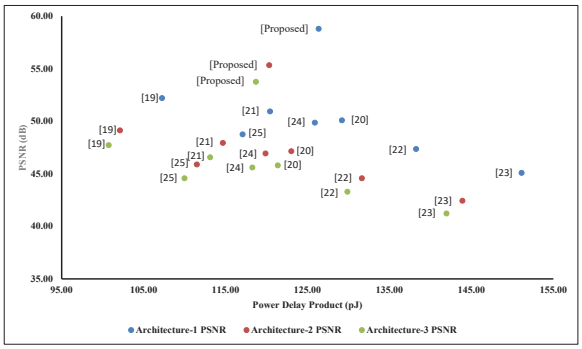

Fig. 5. Pareto Diagram of PSNR Versus PDP of Approximate 8×8 Multipliers.

VI. APPLICATION: IMAGE PROCESSING

The validation of the approximate 8×8 multiplier using proposed inaccurate 3:2 compressor can be demonstrated through several error-tolerant applications. Prior works [27], [31], [32] have employed a variety of image-processing benchmarks to evaluate and validate their inexact computing architectures.

In this section, the architecture-3 based 8×8 multiplier is utilized for image multiplication and image convolution. The proposed inaccurate 3:2 compressors are substituted in place of the exact compressors for performing the operations. For comparative analysis, the state-of-the-art inexact designs reported in [19]–[25] are also incorporated.

A. Image Multiplication

A MATLAB script has been developed to carry out the multiplication process. Two 256×256 grayscale images, apple.jpg and rice.png, are used for pixel-wise multiplication, where each pair of corresponding pixels is multiplied to generate the resultant image shown in Fig. 6. As discussed earlier, the human visual system has limited perceptual ability to detect minor inaccuracies across a set of approximate images. Therefore, quantitative metrics are required to assess the quality improvement of the approximate output images. Two widely used image quality metrics—Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM)—are employed for this purpose. The corresponding results are presented in Table VIII.

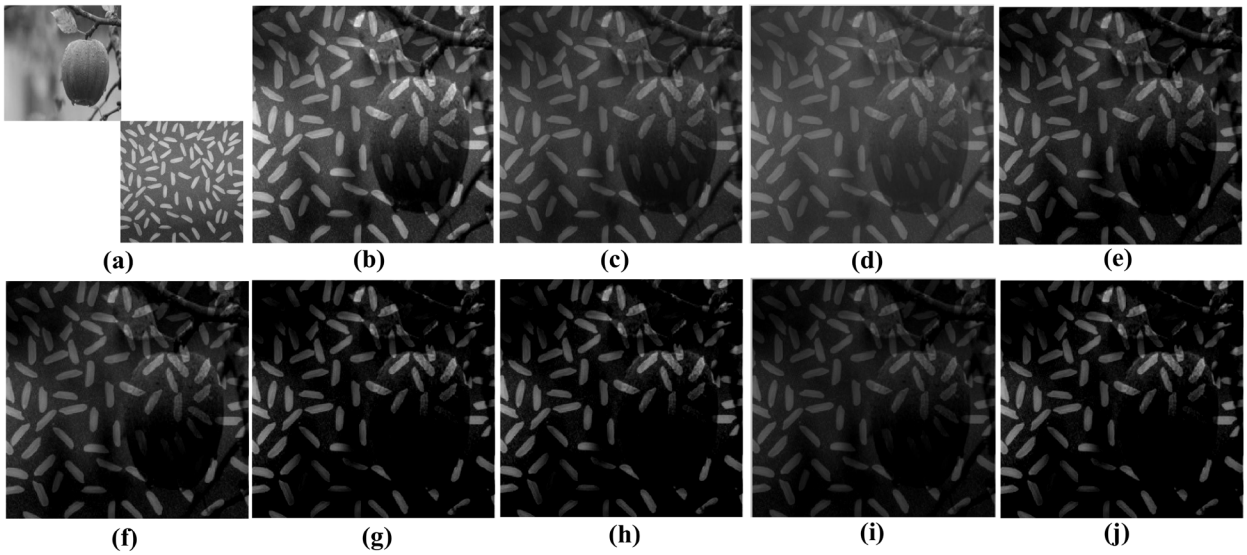

Fig. 6. Multiplication of Two Images Using Exact and Inaccurate 3:2 Compressors Based 8-bit Multiplier (a) Apple, Rice (b) Exact Multiplication (c) Inexact Multiplication by Proposed design (d) [19] (e) [20]-3 (f) [21]-2 (g) [22]-2 (h) [23] (i) [24]-4 (j) [25]-4

TABLE VIII
ESTIMATED PSNR AND SSIM OF APPROXIMATE 8×8 MULTIPLIER FOR ARCHITECTURE-3

Designs	Image Multiplication		Image Convolution	
	PSNR	SSIM	PSNR	SSIM
Proposed	53.76	0.993	54.67	0.994
Beura [19]	47.73	0.986	48.54	0.987
Yang3 [20]	45.8	0.977	46.58	0.978
Almurib2 [21]	46.57	0.983	47.36	0.984
Raveendran2 [22]	43.3	0.972	44.03	0.973
Ramaswamy [23]	41.23	0.968	41.93	0.969
Mehrabani4 [24]	45.6	0.975	46.37	0.976
Yan4 [25]	44.58	0.973	45.34	0.974

B. Image Convolution

Image convolution is a fundamental operation in image processing. It involves a spatial filter (kernel) that slides across an image, performing a sum-of-products operation between the kernel weights and the pixel intensities in the local neighborhood. In this task, the standard exact multipliers are replaced with approximate ones.

The convolution of an input image I with a kernel K of size $m \times n$ is defined as:

$$G(x,y) = \sum_{i=1}^{m} \sum_{j=1}^{n} I(x+i-1, y+j-1) \cdot K(i,j) \quad (13)$$

where:

- $I(x,y)$ is the pixel intensity at coordinates (x,y).
- $K(i,j)$ represents the weights of the convolution kernel.
- $G(x,y)$ is the resulting output pixel.

The multiplication $I \cdot K$ is performed using proposed and reported approximate 8×8 multipliers. The error introduced is evaluated by comparing the resulting $G(x,y)$ with the output of an exact multiplier. For the analysis cameraman.tif image is considered. A 3×3 Laplacian kernel, which highlights areas of rapid intensity change, is defined as:

$$K = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (14)$$

The output images obtained after inexact multiplication, along with their corresponding PSNR and SSIM values, are presented in Fig. 6 and Table VIII, respectively. Although the multiplied images exhibit minor differences, these variations are generally imperceptible to the human eye. Nevertheless, the computed PSNR and SSIM metrics clearly indicate that the inexact multiplication using the proposed inaccurate 3:2 compressor achieves superior image quality compared to the existing designs reported in [19]–[23]. The image convolution resultant images are shown in Fig. 7. Estimated PSNR and SSIM values are tabulated in Table VIII. By incorporating a single low-probability error-based inaccurate 3:2 compressor, our approximate 8×8 multiplier achieves improved PSNR and SSIM performance compared to existing counterparts.

VII. CONCLUSION

This paper presented a novel inaccurate 3:2 compressor designed by introducing a single, low-probability error in the sum output while maintaining an exact carry output. By exploiting the statistical rarity of the all-one input combination, the proposed design achieves reduced hardware complexity with minimal impact on accuracy. The compressor was successfully integrated into a radix-4 methodology based 8×8

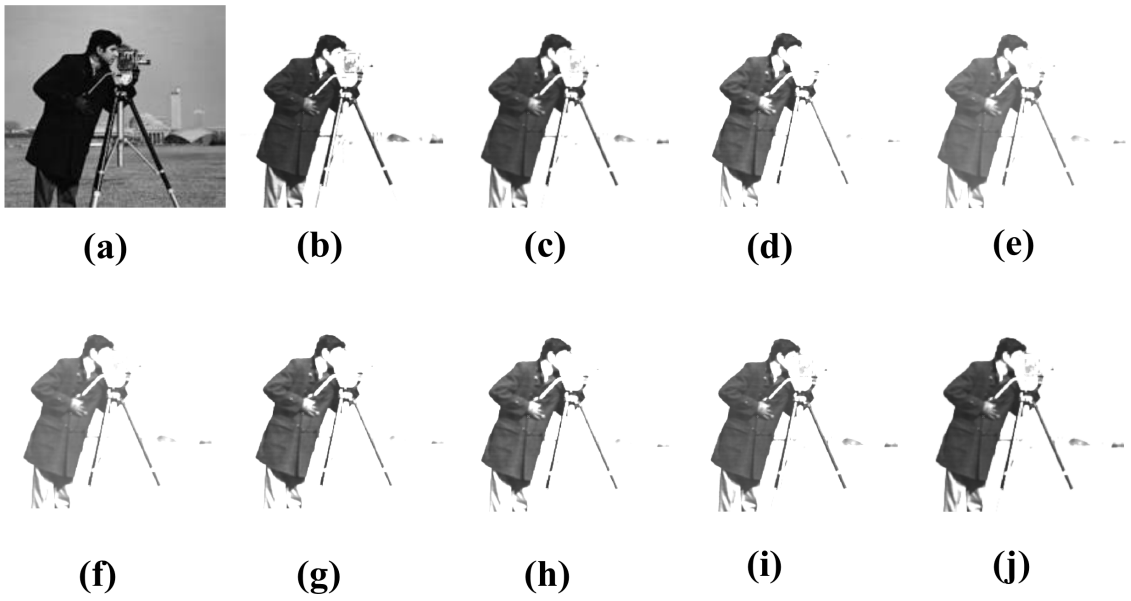

Fig. 7. Image Convolution Using Inaccurate 3:2 Compressors Based Approximate 8-bit Multiplier (a) Cameraman.tif (b) using Exact multiplier (c) using Proposed design (d) [19] (e) [20] (f) [21] (g) [22]-2 (h) [23] (i) [24]-4 (j) [25]-4.

multiplier architecture constructed hierarchically from smaller multiplier units. Three different architectures were investigated for partial product accumulation using the proposed inexact compressor. Comprehensive error analysis revealed that the proposed multiplier significantly improves ER, MED, and NMED compared to existing inexact designs. Circuit-level simulations using Cadence Spectre with GPDK 45-nm technology further demonstrated that the proposed architecture delivers competitive area, power, and delay characteristics relative to the best reported state-of-the-art designs. The practical applicability of the proposed approach was validated through an image multiplication and image convolution task, where improved PSNR and SSIM values confirmed superior perceptual quality. Overall, the proposed inaccurate 3:2 compressor and multiplier architecture offer an effective solution for low-power and error-tolerant signal processing applications.

REFERENCES

[1] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-power digital signal processing using approximate adders," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 1, pp. 124–137, 2013. DOI: 10.1109/TCAD.2012.2217962

[2] M. Chintagunta and V. R. Kota, "Compressor based approximate multipliers for neural network accelerators," in *2024 International Conference on Advancements in Power, Communication and Intelligent Systems (APCI)*, 2024, pp. 1–6. DOI: 10.1109/APCI61480.2024.10616425

[3] P. Jaswal, L. H. Krishna, and B. Srinivasu, "Low power approximate multiplier architecture for deep neural networks," *arXiv preprint arXiv:2509.00764*, 2025. DOI: 10.48550/arXiv.2509.00764

[4] S. Dutt, S. Dash, S. Nandi, and G. Trivedi, "Analysis, modeling and optimization of equal segment based approximate adders," *IEEE Transactions on Computers*, vol. 68, no. 3, pp. 314–330, 2019. DOI: 10.1109/TC.2018.2871096

[5] H. R. Mahdiani, A. Ahmadi, S. M. Fakhraie, and C. Lucas, "Bio-inspired imprecise computational blocks for efficient vlsi implementation of soft-computing applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 4, pp. 850–862, 2010. DOI: 10.1109/TCSI.2009.2027626

[6] N. Zhu, W. L. Goh, W. Zhang, K. S. Yeo, and Z. H. Kong, "Design of low-power high-speed truncation-error-tolerant adder and its application in digital signal processing," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 8, pp. 1225–1229, 2010. DOI: 10.1109/TVLSI.2009.2020591

[7] E. Napoli, E. Zacharelos, A. G. M. Strollo, and G. Di Meo, "Approximate full-adders: A comprehensive analysis," *IEEE Access*, vol. 12, pp. 136 054–136 072, 2024. DOI: 10.1109/ACCESS.2024.3463182

[8] V. Leon, G. Zervakis, D. Soudris, and K. Pekmestzi, "Approximate hybrid high radix encoding for energy-efficient inexact multipliers," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 3, pp. 421–430, 2018. DOI: 10.1109/TVLSI.2017.2767858

[9] L. Qian, C. Wang, W. Liu, F. Lombardi, and J. Han, "Design and evaluation of an approximate wallace-booth multiplier," in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2016, pp. 1974–1977. DOI: 10.1109/ISCAS.2016.7538962

[10] W. Liu, L. Qian, C. Wang, H. Jiang, J. Han, and F. Lombardi, "Design of approximate radix-4 booth multipliers for error-tolerant computing," *IEEE Transactions on Computers*, vol. 66, no. 8, pp. 1435–1441, 2017. DOI: 10.1109/TC.2017.2672976

[11] S. Venkatachalam, H. J. Lee, and S.-B. Ko, "Power efficient approximate booth multiplier," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2018, pp. 1–4. DOI: 10.1109/ISCAS.2018.8351708

[12] S. Venkatachalam, E. Adams, H. J. Lee, and S.-B. Ko, "Design and analysis of area and power efficient approximate booth multipliers," *IEEE Transactions on Computers*, vol. 68, no. 11, pp. 1697–1703, 2019. DOI: 10.1109/TC.2019.2926275

[13] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, "Design and analysis of approximate compressors for multiplication," *IEEE Transactions on Computers*, vol. 64, no. 4, pp. 984–994, 2015. DOI: 10.1109/TC.2014.2308214

[14] Z. Yang, J. Han, and F. Lombardi, "Approximate compressors for error-resilient multiplier design," in *2015 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFTS)*, 2015, pp. 183–186. DOI: 10.1109/DFT.2015.7315159

[15] D. Esposito, A. G. M. Strollo, E. Napoli, D. De Caro, and N. Petra, "Approximate multipliers based on new approximate compressors," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 12, pp. 4169–4182, 2018. DOI: 10.1109/TCSI.2018.2839266

[16] O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, "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. DOI:

10.1109/TVLSI.2016.2643003

[17] M. S. Kim, A. A. D. Barrio, L. T. Oliveira, R. Hermida, and N. Bagherzadeh, "Efficient mitchell's approximate log multipliers for convolutional neural networks," *IEEE Transactions on Computers*, vol. 68, no. 5, pp. 660–675, 2019. DOI: 10.1109/TC.2018.2880742

[18] H. Liu, M. Wang, L. Yao, and M. Liu, "A piecewise linear mitchell algorithm-based approximate multiplier," *Electronics*, vol. 11, no. 12, p. 1913, 2022. DOI: 10.3390/electronics11121913

[19] S. K. Beura, A. A. Jawale, B. P. Devi, and P. Saha, "On the implementation of multi-bit inexact adder cells and application towards image de-noising," *ELECTRONICS*, vol. 24, no. 1, pp. 33–42, 2020. DOI: 10.7251/els2024033b

[20] Z. Yang, A. Jain, J. Liang, J. Han, and F. Lombardi, "Approximate xor/xnor-based adders for inexact computing," in *2013 13th IEEE International Conference on Nanotechnology (IEEE-NANO 2013)*, 2013, pp. 690–693. DOI: 10.1109/NANO.2013.6720793

[21] H. A. Almurib, T. N. Kumar, and F. Lombardi, "Inexact designs for approximate low power addition by cell replacement," in *2016 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2016, pp. 660–665.

[22] S. Raveendran, P. J. Edavoor, N. Y. B. Kumar, and M. H. Vasantha, "An approximate low-power lifting scheme using reversible logic," *IEEE Access*, vol. 8, pp. 183 367–183 377, 2020. DOI: 10.1109/ACCESS.2020.3029149

[23] M. Ramasamy, G. Narmadha, and S. Deivasigamani, "Carry based approximate full adder for low power approximate computing," in *2019 7th International Conference on Smart Computing & Communications (ICSCC)*, 2019, pp. 1–4. DOI: 10.1109/ICSCC.2019.8843644

[24] Y. Safaei Mehrabani and R. Faghih Mirzaee, "Dafa: Dynamic approximate full adders for high area and energy efficiency," *Integration*, vol. 97, p. 102191, 2024. DOI: 10.1016/j.vlsi.2024.102191

[25] A. Yan, H. Bao, W. Jiang, J. Cui, Z. Huang, and X. Wen, "Efficient design approaches to cmos full adder circuits," *Microelectronics Journal*, vol. 149, p. 106235, 2024. DOI: 10.1016/j.mejo.2024.106235

[26] H. A. Ziad and A. A. Sakib, "Haloc-axa: an area/-energy-efficient approximate adder for image processing application," in *2025 IEEE 68th International Midwest Symposium on Circuits and Systems (MWSCAS)*. IEEE, 2025, pp. 64–68. DOI: 10.48550/arXiv.2510.20137

[27] A. G. M. Strollo, E. Napoli, D. De Caro, N. Petra, and G. D. Meo, "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. DOI: 10.1109/TCSI.2020.2988353

[28] B. S. Mithil and C. Ramesh, "Approximate parallel prefix adders for image processing applications," in *2025 IEEE 5th International Conference on VLSI Systems, Architecture, Technology and Applications (VLSI SATA)*, 2025, pp. 1–6. DOI: 10.1109/VLSISATA65374.2025.11070135

[29] S. Venkatachalam and S.-B. 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. DOI: 10.1109/TVLSI.2016.2643639

[30] J. Liang, J. Han, and F. Lombardi, "New metrics for the reliability of approximate and probabilistic adders," *IEEE Transactions on Computers*, vol. 62, no. 9, pp. 1760–1771, 2013. DOI: 10.1109/TC.2012.146

[31] S. K. Beura, S. M. Mahanta, B. P. Devi, and P. Saha, "Inexact radix-4 booth multipliers based on new partial product generation scheme for image multiplication," *Integration*, vol. 94, p. 102096, 2024. DOI: 10.1016/j.vlsi.2023.102096

[32] H. Pei, X. Yi, H. Zhou, and Y. He, "Design of ultra-low power consumption approximate 4–2 compressors based on the compensation characteristic," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 1, pp. 461–465, 2021. DOI: 10.1109/TCSII.2020.3004929