Received: September 26, 2024. Revised: November 10, 2024. 422

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Design and Evaluation of High-Speed Approximate Multipliers Based on Improved Error Distance 4:2 Compressors for Error Resilient Image Applications

Zahraa A. Zuhair[1]* **Emad A. Al-Sabawi[1]**

[1]*Computer and Information Engineering Department, College of Electronics, Ninevah University, Mosul Iraq*
* Corresponding author's Email: zahraa.ahmed@uoninevah.edu.iq

Abstract: Approximate Computing (AC) have widely adopted in designing large-scale logic circuits. In particular, approximate adder and multiplier circuits have been considerably targeted in the realm of image processing due to their high energy-saving while preserving proper level of computing accuracy. Nevertheless, intensive research and development are maintained in the means of seeking more matured designs that effectively prioritize design overheads over error resilience. In this paper, low error-distance approximate 4:2 compressor circuits are proposed to construct high-speed approximate adders. The developed compressors realize high logic computing and incur competitive area and power consumption, and therefore, they were leveraged to configure an approximate 8×8 multiplier designs. To achieve a favorable trade-off between computational accuracy and hardware resource usage, we develop a simulation framework that evaluates the accuracy of the proposed multiplier designs at the gate-level (measuring error distance) and at the application-level (evaluating SNR and SSIM) of an image. The framework truncates specific propagated carry bits, i.e., least significant bits (LSBs), to realize profitable area- and power-saving. Furthermore, two main high-speed multiplier designs are proposed herein, namely High Computing Performance Approximate Multiplier (HCP-AMUL) and HCP Low Error Approximate Multiplier (HCPLE-AMUL). Matlab R2022b along with VS Code are used for running simulations and accuracy evaluation, while Vivado 2018.2 is utilized for HDL reconfigurable logic design and implementation and evaluation of area, power, and speed, configured on an FPGA Xilinx Nexys 4 Artix-7 (XC7A100T-1CSG324) trainer board. The experimental results demonstrate the efficacy of the developed multipliers as the developed HCPLE-AMUL delivers 54.26%, 11.72%, and 449.85 of speedup, power saving, and Power-Delay-Area-Error-Product (PDAEP) improvement, respectively. On the other hand, the presented HCP-AMUL realizes an improved saving of area and power at the expense of an acceptable lowering of computation accuracy. It achieves 9.66%, 505.40, and 53.73% of power saving, PDAEP, and speedup respectively, Thus, the proposed compressor and multiplier circuits potentially can be promising approximate computing modules for image processing applications to provide improved trade-off between computation accuracy and logic utilization complexity.

Keywords: Approximate computing, Approximate multiplier, 4:2 compressors, Accuracy improvement, Energy-efficiency, Image processing, High-performance computing, Low error distance.

1. Introduction

Modern systems and large-scale applications necessitate a massive amount of resources, such as energy-storage modules, memory, and compute units [1]. As systems and applications of big data processing have been extensively evolved, high-performance and energy efficient electronic circuits have become necessity, especially for portable devices with limited battery capacity. Interestingly, it is often not possible to meet all demands and enhance the entire system parameters simultaneously [2]. In contrast, low-power circuits have greater delay, while high-power structures are faster. This is where circuit and system designers attempt to strike a balance [2]. In the recent decade, various approaches have been introduced that deal with resource constraints on the one hand, and maintain high data processing performance on the other [2]. To address these

challenges, approximate computing has emerged as an alternative low-power design approach. It is considered an effective model for increasing power efficiency and enhancing embedded system performance [3]. It allows for some error tolerance in the circuit outputs in order to simplify logical expressions, resulting in area saving at the transistor level, dynamic power dissipation, and reduced propagation delay of computation [3].

Approximate logic computing can be applied at various levels, from circuits up to architecture [4]. Many applications, such as image processing, digital signal processing, and neural networks, do not demand a high level of computing accuracy as human vision can recognize and distinguish images within an acceptable level of quality [5]. For instance, approximate logic computing can be employed in image-based applications, such as image sharpening, smoothing, and classification [6-8]. Therefore, approximate computing has been necessitated to target favourable attributes of minimizing computational complexity, power consumption, and processing delay while achieving acceptable computing accuracy [8-11]. In approximate computing, logic circuits are designed to slightly tolerate errors while preserving acceptable accuracy of the computed outputs [6]. In recent years, researcher has focused on approximating arithmetic circuits, including adder and multiplier circuits. Due to the large area and high-power consumption of multipliers imposes significant overheads on the system design. Thus, applying approximate computing to these circuits can offer benefits of power and performance. Practically, approximate approaches can be applied to any stage of a multiplier, nevertheless, partial products (PPs) accumulation consumes the majority of the hardware resources and execution time. Therefore, optimizing partial product accumulation can significantly reduce area-power overheads [3].

Adder circuits are considered the computational pillar of large-scale logic upon which significantly sophisticated arithmetic circuits, such as multiplication, division, square root, etc., can be constructed [8, 10]. However, the multiplication circuit is the most used among the arithmetic computational circuits, embedded in the processor for data processing [12]. Besides, it incurs high logic area and power overheads and takes longer execution time. In particular, a multiplier circuit is primarily composed of addition circuits that take place at two stages, i.e., partial products reduction or compression and final output product calculation [6]. Approximate compressors and multipliers are mainly designed to reduce power consumption, aiming at trading off accuracy for improved efficiency via reducing the number of operands or their bits width [13]. This implies that an approximate multiplier scarifies a small amount of combinational logic, required to perform computation, to achieve low power consumption and/or speed up execution performance. However, this is delivered at the expense of impacting the computational accuracy of multiplication [14]. Thus, it has become essential to perform multiplication operations using compressed multipliers that are configured based on approximate adder circuits. This can assist in achieving an effective trade-off between error resilience and design overheads in terms of logic area, propagation delay, and power to deliver error-tolerant and energy-efficient multipliers [6].

Several approximate compressor circuits were proposed with various ranges of bits compression including 3:2, 4:2, and 5:2 [15, 16]. However, the approximate 4:2 compressor is widely used to approximate the intermediate addition operations, required to accumulate the partial products and other multiplication steps [17, 18]. Consequently, to further address this research topic more maturely, this paper introduces two designs of 4:2 approximate compressor, utilized to accumulate the partial products of multiplication. Afterward, these proposed compressors were leveraged to develop two main approximate multiplier designs, namely High Computing Performance Approximate Multiplier (HCP-AMUL) and HCP Low Error Approximate Multiplier (HCPLE-AMUL). The multipliers are 8×8 bits width designs that are intended to be deployed for approximate computing in the realm of image processing and/or neural networks. They realize a confidence level of computing accuracy in terms of Mean Error Distance (MED) and Mean Relative ED (MRED). Meanwhile, they offer additional favourable attributes such as competitive area, power, delay, and Power Delay Product (PDP) compared to cutting-edge prior works. The manuscript contributes the following:

1. Propose two low error distance and energy-efficient 4:2 compressors based on trading off error resiliency over the logic area, power, and propagation delay overheads.
2. Leveraged the proposed compressors to configure two developed 8×8 multipliers on an FPGA platform using Verilog as an HDL language with Xilinx Vivado for reconfigurable hardware design and implementation. Followed by evaluating favorable attributes of parallel high computing performance while maintaining

acceptable computing accuracy of multiplication operations.

3. Introduce an accurate framework architecture that simulates the developed multipliers at two levels including:
   - Building a software module of the multiplier and evaluating the error distance of all possible combinations of 8×8 multiplier by measuring the error distance between the accurate and approximated values of the multipliers at the gate-level using C language with VS Code as an IDE to build and run software codes.
   - Moreover, employing the design that achieves reduced error distance at the application-level for image smoothing and evaluating its accuracy in terms of PSNR and SSIM, which have been the most crucial standard metrics used to evaluate and demonstrate an image quality with MATLAB environment.

In the field of low-power design for logic circuits, approximate computing has been an emerging approach that prioritizes reducing the power consumption of adder and multiplier circuits over error resilience at gate and circuit levels. In the literature, several scholarly studies [6, 8, 16, 19] have investigated the trade-off of sacrificing a small amount of computing accuracy to minimize power and area overheads of adder and multiplier circuits, relying on employing approximate computing paradigms.

Compressors are vastly used to build multiplier structures to simplify and accelerate the process of partial reduction of the product. They are built from cascaded full adder circuits, that configure a variety of structures, including 5:2, 3:2, and 4:2. Herein, we focus on the 4:2 compressor as is considered the most common [8], due to 8:4 and 16:4 compressors can be formed using it, where the latter are used in constructing 8×8 and 16×16 multiples. Thus, here, a wide range of approximate adder, compressor, and multiplier designs are reviewed at different abstraction levels of approximation. Several paradigms of computational approximation are presented in the last decade. For instance, the authors in [11] suggest that approximating the computation using software- and application-level algorithms is a suitable approach. An adder circuit takes the inputs (x_1, x_2, and x_3) and produces the outputs (sum and carry). Approximate techniques for designing adder circuits concentrate on ignoring or truncating specific computation bits of sum and carry according to their computational weight importance while preserving high accuracy via realizing a minimal Error Distance (ED). Therefore, the outcome of conducting logic reduction in the design of an adder circuit leads to designing approximate adders. For instance, in [20] a full adder circuit is proposed that ignores the input C_{in} and output C_{out}, the logic expression is provided in Table 1. To reduce the design complexity and achieve energy-efficiency, the authors in [21] omitted the input/output carry (C_{in} and C_{out}), thus the signals C_{in} and C_{out} are truncated. Similarly, in 2023 [22], the carry signals (C_{in} and C_{out}) are truncated to lower the power consumption.

Various studies aimed at reducing computing overhead while maintaining acceptable accuracy. A study in [6] involves replacing certain types of logic gates with alternatives to achieve approximate computation, the logic expression of FA and HA are provided in Table 1. Minaeifar et al. [6] presented three approximate multipliers (Mul-1, Mul-2, and Mul-3) by truncating the LSBs to reduce design complexity and power consumption. The authors employed exact 4:2 coprocessors, approximate half-adders (HA), and exact and approximate full-adder (FA) to accumulate the PPs of the proposed approximate multipliers. In the approximate of HA, they computed the sum using an OR-gate instead of an XOR-gate and maintained the carry computed based on an AND gate, thus 12 transistors are required to construct a HA circuit. Whereas in the approximate of FA, the sum is computed directly as x_1 and the carry is produced by ORing (x_1, x_2, and C_{in}), therefor, 8 transistors are utilized to configure an approximate FA circuit. Additionally, two to four LSBs are assigned to logic 0 in both multiplication operands, i.e., multiplicand and multiplier, which results in 28 to 32 AND gates reduction in the generation of PPs. Although the proposed approximate multipliers achieve significant PDP reduction (50%, 53%, and 23%, respectively), they incur relatively considerable MRED (6.5×10^{-2}, 25×10^{-2}, and 11.8×10^{-2}, respectively). In another study [7], the circuit outputs are considered either (0) or (1). Three methods were developed including method (1) presents the *sum* $=x_2 \oplus x_3$ and the *carry* $=x_1$, whereas method (2) sets the *sum =0* and the *carry* $=x_1$, and in method (3) *sum =0* and the *carry* = $x_2 \oplus x_3$. To incorporate such approximate suggestions into the design of approximate compressor circuits. The logic circuit of their proposed designs are listed in Fig. 1. Other studies [23, 24] aim to reduce the complexity of the exact compressor via proposing an approximate 4:2 compressor that ignores the C_{in} and C_{out}. In 2023 [25], 2-methods proposed an approximate 4:2

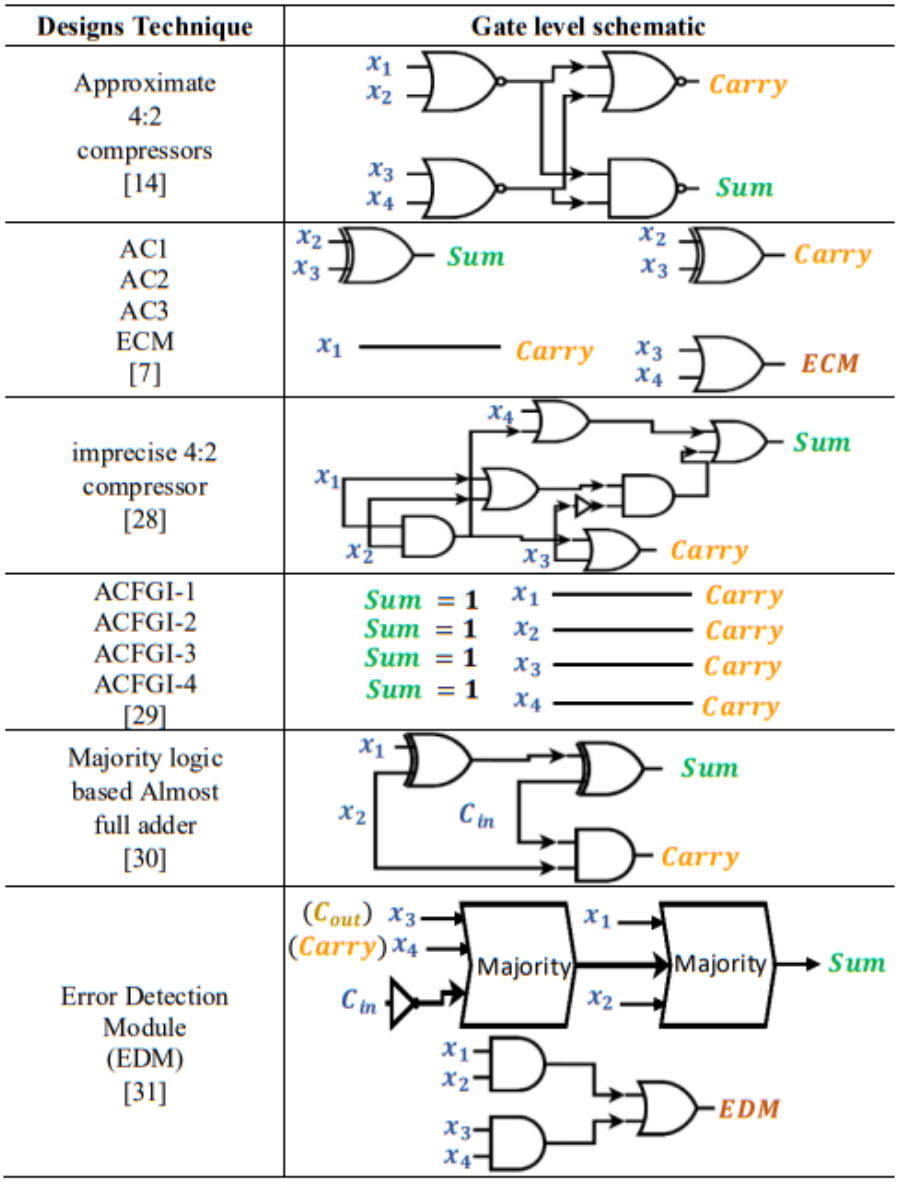


Figure. 1 Gate level schematic for Previously proposed approximate adder circuits

compressor by using decomposition and recombination techniques to approximate the sum and carry expressions. Also, they omitted the propagated carry-in and carry-out (C_{in} and C_{out}) to shrink the complexity of the compressor, thus speeding up the accumulation process of partial products.

Furthermore, the authors in [21], proposed a 5-method of 4:2 approximate compressors with high accuracy and shorter delay paths to achieve a suitable trade-off by recombination methods and decomposition, with an introduced 2-error. Likewise, in 2024 [26], an approximate 4:2 compressor is proposed to be implemented for power-efficient multiplier, the logic equations of their proposal are listed in Table 1. The authors in [27], presented approximate 4:2 compressor by using two NOR gates, an XOR gate, and an OR gate, which makes the compressor incurs reduced number of transistors, as listed in Table 1. Van Toan et al. [3], introduced four approximate 4:2 compressors including CP1, CP2, CP3, and CP4, where their equations are presented in Table 1. The trade-off between accuracy loss and transistor count is always a critical issue when designing approximate compressors. Logic circuits designers attempt to produce the most accurate approximate compressors with the lowest number of transistors, which is essential due to the increasing demand of approximate applications, these methods are illustrated in Fig. 1. Table 1 also provides equations for other previous works. Both proposed compressors offer superior trade-off between error resilience and the number of transistors compared to

Table 1. Recent selected related prior works of approximate 4:2 compressor design

Design Technique	Year	Logic Equations
Approximate Half-adder and Full-adder [6]	2024	$sum_{ha} = x_1 + x_2$ $carry_{ha} = x_1 \cdot x_2$ $sum_{fa} = x_1$ $carry_{fa} = x_1 + x_2 + C_{in}$
Approximate Full-adder [32]	2023	$sum_{fa} = \overline{(A + B) \oplus C}$ $carry_{fa} = (A + B) \cdot C$
AFAC-1 AFAC-2 AFAC-3 AFAC-4 AFAC-5 [16]	2022	$sum_{AFAC-1} = x_2 \cdot \bar{x}_3 + x_1 \cdot \bar{x}_3 + \bar{x}_1 \cdot x_3$ $carry_{AFAC-1} = x_3 \cdot (x_1 + x_2)$ $sum_{AFAC-2} = x_3 + x_1 \oplus x_2$ $carry_{AFAC-2} = x_3(x_1 \oplus x_2) + (\overline{x_1 \oplus x_2}) \cdot x_1$ $sum_{AFAC-3} = x_1(\bar{x}_2 + x_3) + \bar{x}_2 \cdot x_3 + \bar{x}_1 \cdot x_2 \cdot \bar{x}_3$ $carry_{AFAC-3} = x_2 \cdot (x_3 + x_1)$ $sum_{AFAC-4} = x_1 \oplus x_2 \oplus x_3$ $carry_{AFAC-4} = x_3$ $sum_{AFAC-5} = x_1 + x_2 + x_3$ $carry_{AFAC-5} = x_3(x_1 + x_2)$
Approximate Half-adder and Full-adder [33]	2020	$sum_{ha} = x_1 + x_2$ $carry_{ha} = x_1 \cdot x_2$ $sum_{fa} = (x_1 + x_2) \oplus x_3$ $carry_{fa} = (x_1 + x_2) \cdot x_3$
Almost full adder based 4:2 adder compressors [30]	2022	$C_{out} = (x_1 \oplus x_2) \cdot x_3 + (x_1 \oplus x_2) \cdot x_1$ $S = (x_1 \oplus x_2 \oplus x_3)$ $sum = S \oplus x_4 \oplus C_{in}$ $carry = (S \oplus x_4) \cdot C_{in} + S \cdot x_4$
Approximate 4:2 compressor [27]	2023	$sum = \overline{(p_1 \oplus p_2) + (\overline{p_3 + p_4})}$

Design Technique	Year	Logic Equations
		$carry = p_1 + p_2$
AC6G ACFGI ACFGII [29]	2023	$sum_{AC6G} = (x_1 + x_2) + (x_3 + x_4)$ $carry_{AC6G} = (x_1 \cdot (x_3 + x_4)) + (x_2 \cdot x_3)$ $sum_{AC6GI} = 1$ $carry_{AC6GI} = x_1$ $sum_{AC6GII} = x_1$ $carry_{AC6GII} = x_2$
Approximate 4:2 compressor [31]	2017	$sum = (x_1 \oplus x_2) \oplus (x_3 + x_4)$ $carry = x_1 \cdot x_2 + x_1 \cdot x_3 + x_1 \cdot x_4 + x_2 \cdot x_3 + x_2 \cdot x_4$
Approximate 4:2 compressor [28]	2023	$sum = x_1 \cdot x_2 + x_4 + \bar{x}_3(x_1 + x_2)$ $carry = x_1 \cdot x_2 + x_3$
Approximate 4:2 compressor [20]	2019	$\overline{sum} = (x_1 \oplus x_2 \oplus x_3)\bar{x}_4 + (x_3 \cdot x_2 + x_3 \cdot x_1 + x_2 \cdot x_1)x_4$ $\overline{carry} = (x_3 \cdot x_2 + x_3 \cdot x_1 + x_2 \cdot x_1) \cdot \bar{x}_4 + x_4$
inexact 4:2 compressor [10]	2024	$sum = \overline{\overline{(x_1 + x_2 + x_3)} \cdot \overline{x_4}}$ $carry = \overline{\overline{(x_1 + x_2 + x_3)} + \overline{x_4}}$
Approximate compressor [26]	2024	$sum = ((x_1 + x_2 + x_3) \oplus x_2)$ $carry = ((x_4 + x_3) \cdot (x_2 + x_1)) + (x_4 \cdot x_3)$
CP1 CP2 [3]	2020	$sum = (x_4 \oplus x_3 \oplus x_2 \oplus x_1) + x_4 \cdot x_3 \cdot x_2 \cdot x_1$ $carry = (x_4 + x_3) \cdot (x_2 + x_1) + (x_4 \cdot x_3) + (x_2 \cdot x_1)$ $sum = (x_2 + x_1) + (x_4 \oplus x_3)$ $carry = (x_4 + x_3) \cdot (x_2 + x_1) + x_4 \cdot x_3$

counterparts. They realize an average of 81.25% and 62.5% higher accuracy, respectively, and reducing the number of transistors (32 and 28, respectively). On the other hand, multiplies are commonly used for performing computation, but they significantly contribute to the overall computational power consumption due to incurring substantial data processing at the gate-level. In this regard, many approximate designs of multiplication operations have been introduced in prior works, to achieve

partial product minimization using a tree of approximate compressors to save hardware costs without significant loss of accuracy. An approximate 8×8 multiplier was proposed in [8], to achieve an improved logic utilization of the hardware, the authors truncated the first three columns of the partial products and compressed the 4-column via the proposed approximate 4:2 compressor circuit, while the remaining columns were compressed using an exact compressor circuit. In their work [6], they focused on using only approximate compressors in their design (1) of the approximate multiplier. While design (2) focuses on minimizing the error distance by using 8 exact compressors in the most significant column of the PPs and 9 approximate compressors in the less significant column. To obtain the final product, they used half and full adder circuits. The authors in [23], proposed an approximate multiplier (design 1), where it was designed in three stages. The first stage includes a three-half adder, a seven 4:2 approximate compressor, and a two-full adder. The second stage includes a one-half adder and a ten 4:2 approximate compressor. In design (2), the lower half of columns utilizes the suggested approximate 4:2 compressors, while the columns in the upper half utilize an accurate 4:2 compressor to minimize error distance. In unsigned 8-bit Dadda tree multipliers [16], the process involves two steps. In step (1), two half-adders, two 3:2 compressors, and eight 4:2 compressors are used. In step (2), one half-adder, one 3:2 compressor, and ten 4:2 compressors are used. The final product is obtained by carrying out adder propagation from the previous stage. To construct the 8x8 Dadda Multiplier, the authors in [22], suggested using the 4:2 compressor. In stage (1), they used three half adders, two full adders, four approximate 4:2 compressors, and three exact 4:2 compressors. In stage (2), they utilized one-half adder, one full adder, five approximate 4:2 compressors, and five exact 4:2 compressors. The exact 4:2 compressors are used in the seven most significant columns, and the proposed approximate 4:2 compressors are used in the eight least significant columns. Compared with the previous works of different methods, the proposed design of an eight-bit approximate multiplier circuit, realized acceptable results; however, further improvements can be met. In this paper, we propose two approximate 8×8 multiplier circuit designs based on two proposed 4:2 compressors with the aim of achieving an effective trade-off between energy efficiency and fault resilience. Our proposed designs characterized significantly lower Mean Error Distance (MED) and Mean Relative Error Distance (MRED) values (329.50, 0.11) for design 1 and (304.84, 0.11) for design (2) than previous designs. Additionally, they achieve a significant reduction of area by (39.81, 39.81) for design 1 and 2, respectively, compared to their equivalents mentioned previously. The primary key that we delivered reduced power-area overheads is that we only employed NAND gates in the design of carry for the approximate compressor. It is known that NAND gates fall under the category of universal gates since they offer high flexibility to implement any logical functions used to compute complex logical expressions. The proposed designs consume less power by (131mW and 128mW), respectively and a saving rate of 9.66% and 11.72%, respectively, therefore, they can be effective competitors in the realm of approximate logic computing.

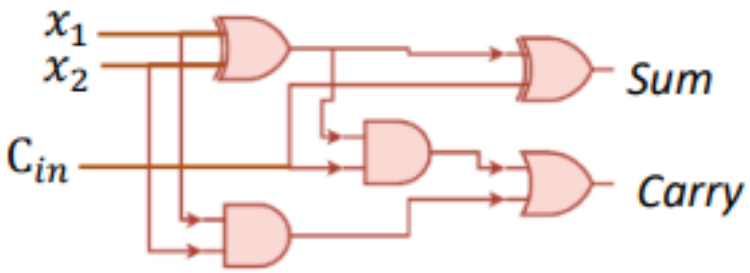
The remaining of the manuscript is structured as follows. Section 2 presents the circuit theory and background of an exact full adder, 4:2 compressor, and 5:2 compressor circuit. The methodology of the proposed two designs of approximate 4:2 compressors is covered in Section 3. The developed approximate 8×8 multiplier designs are also elaborated in Section 3. The simulations and application results are discussed in Section 4, and finally, Section 5 drives the conclusion and paves the path for possible future work to extend this research study.

2. Theory and background

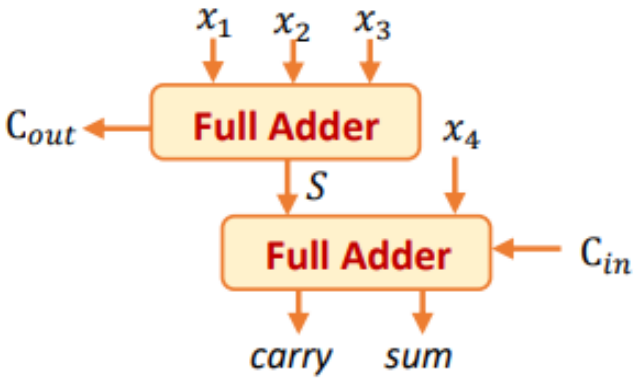
Generally, an exact full adder (FA) circuit encompasses three inputs (x_1, x_2, C_{in}) and two outputs (sum, carry) as illustrated in Fig. 2. Two exclusive-OR (XOR) gates are used to compute the sum whereas two AND gates and an OR gate are required to compute the carry. The function of an exact FA circuit is expressed in Eqs. (1) and (2). In the Early designs of array multipliers, full and half adders have been widely employed to accumulate partial products. However, their high logic intensity, energy consumption, and low-speed performance led to the emergence and development of low-power alternative designs [18, 32]. Thus, to minimize the latency in the PPs accumulation stage of multiplication operations, the 4:2 and 5:2 compressor circuits have been extensively utilized in designing high-speed multipliers [18]. Next, the schematic circuits of 4:2 and 5:2 compressors are discussed.

$$Sum = x_1 \oplus x_2 \oplus C_{in} \tag{1}$$

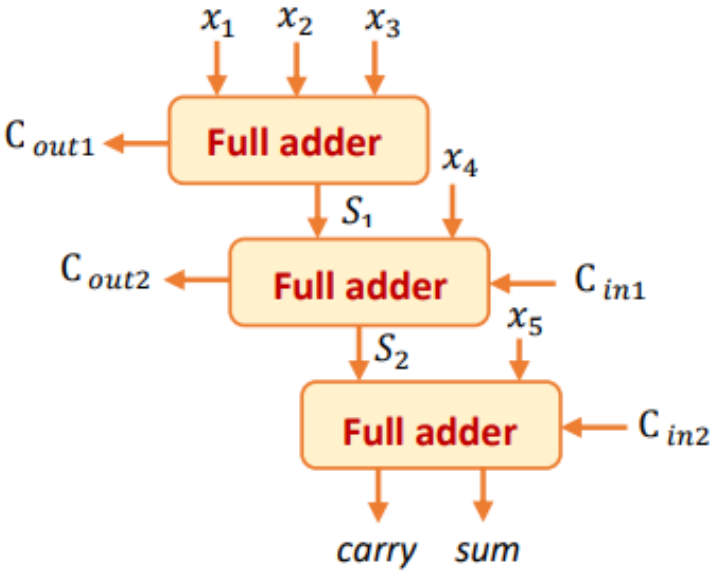
$$C_{out} = (x_1 \oplus x_2) \cdot C_{in} + (x_1 \cdot x_2) \tag{2}$$



(a)



(b)



(c)

Figure. 2 Schematic circuit diagram of adders: (a)Full Adeer Circuit, (b)Exact 4:2 Compressor, and (c) Exact 5:2 Compressor

2.1 Exact 4:2 compressor

For a 4:2 compressor design, the full adder circuit can be modified to have five entries (x_1, x_2, x_3, x_4, and C_{in}) and three outputs (C_{out}, sum, and carry), as expressed in Eqs. (3) to (6) [18], and illustrated in Fig. 2.

$$Sum = x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus C_{in} \quad (3)$$

$$C_{out} = (x_1 \oplus x_2) \cdot x_3 + (x_1 \cdot x_2) \quad (4)$$

$$S = x_1 \oplus x_2 \oplus x_3 \quad (5)$$

$$Carry = (S \oplus x_4) \cdot C_{in} + (S \cdot x_4) \quad (6)$$

Table 2. Notation list

Symbol	Description
AC	Approximate Computing
RISC-V	Reduce Instruction Set Computing-Five
PPs	Partial Products
LSBs	Least significant bits
CSA	Carry Ripple Adder
CRA	Carry Save Adder
CS	Carry, Sum
Mul	Multiplier
HA	Half adder
FA	Full adder
VS	Visual Studio
HDL	Hardware Description Language
FPGA	Field Programable Gate Array
IDE	Integrated Development Environment
2D	2-Dimaentional
MAX_i	Image's maximum pixel value
M_{exc}	Exact value of partial product
M_{appx}	Approximate value of partial product
ER	Error Rate
ED	Error Distance
AME	Absolute Mean Error
NED	Normalize Error Distance
MED	Mean Error Distance
MSE	Mean Squared Error
NMED	Normalize Mean Error Distance
MRED	Mean Relative Error Distance
PDP	Power Delay Product
PDAEP	Power Delay Area Error Product
FOM	Figure Of Merit
SNR	Signal-to-Noise Ratio
PSNR	Peak Signal-to-Noise Ratio
SSIM	Structural Similarity Index Measure
LUT	Look Up Table
ns	nano second
mW	milli Watt
PJ	Pico Joule
N	Number of bits
AC_1	Approximate Computing 1
AC_2	Approximate Computing 2
HCP-AMUL	High Computing Performance Approximate Multiplier
HCPLE-AMUL	High Computing Performance Low Error Approximate Multiplier

2.2 Exact 5:2 compressor

The exact 5:2 compressor circuit is configured by cascading three full adder circuits. A 5:2 compressor has seven inputs, where five are the primary inputs (x_1, x_2, x_3, x_4, x_5) and the other two are carry input bits (C_{in1}, and C_{in2}) come from the previous lower stage. It produces two output bits (sum, and carry) and two carry bits (C_{out1}, C_{out2}) that are connected to the next module stage, as formulated in Eqs. (7) to (12) [18], and depicted in Fig. 2.

$$Sum = x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5 \oplus C_{in1} \oplus C_{in2} \quad (7)$$

$$C_{out1} = (x_1 \oplus x_2) \cdot x_3 + (x_1 \cdot x_2) \quad (8)$$

$$S_1 = x_1 \oplus x_2 \oplus x_3 \quad (9)$$

$$C_{out2} = (S_1 \oplus x_4) \cdot C_{in1} + (S_1 \cdot x_4) \quad (10)$$

$$S_2 = x_4 \oplus S_1 \oplus C_{in1} \quad (11)$$

$$Carry = (S_2 \oplus x_5) \cdot C_{in2} + (S_2 \cdot x_5) \quad (12)$$

3. Methodology of the proposed approximate logic circuits

The design of an approximate multiplier primarily involves two paradigms. They are tailored to either utilize approximate adder and compressor circuits or modify the structure of the multiplier by truncating some specific bits, thereby achieving favorable attributes in terms of energy efficiency and area overhead. In this scholarly study, we proposed two approximate compressor circuits, which are later in this section leveraged to construct two high-speed approximate 8×8 multipliers. In the proposed multipliers, we adopted the approximate compressors in developing the structure of the approximate multipliers. Furthermore, a deep analysis of carry-bits propagation in the multiplier's structure was conducted to achieve an efficient approximation approach of the multiplication. To demonstrate the efficacy of the proposed approximate compressor and multiplier circuits, they are employed in an image processing application, i.e., image smoothing. We evaluated the proposed multipliers in terms of SNR, PSNR, and SSIM to show their superiority while maintaining high computational accuracy. Finally, the structure that provides an improved trade-off between energy efficiency and error resilience is synthesized and implemented to further evaluate the design challenges in terms of power, delay, and area (resource utilization).

3.1 Framework architecture of the proposed approximate multiplier circuit

Fig. 3 illustrates the framework architecture used to implement the proposed designs including two

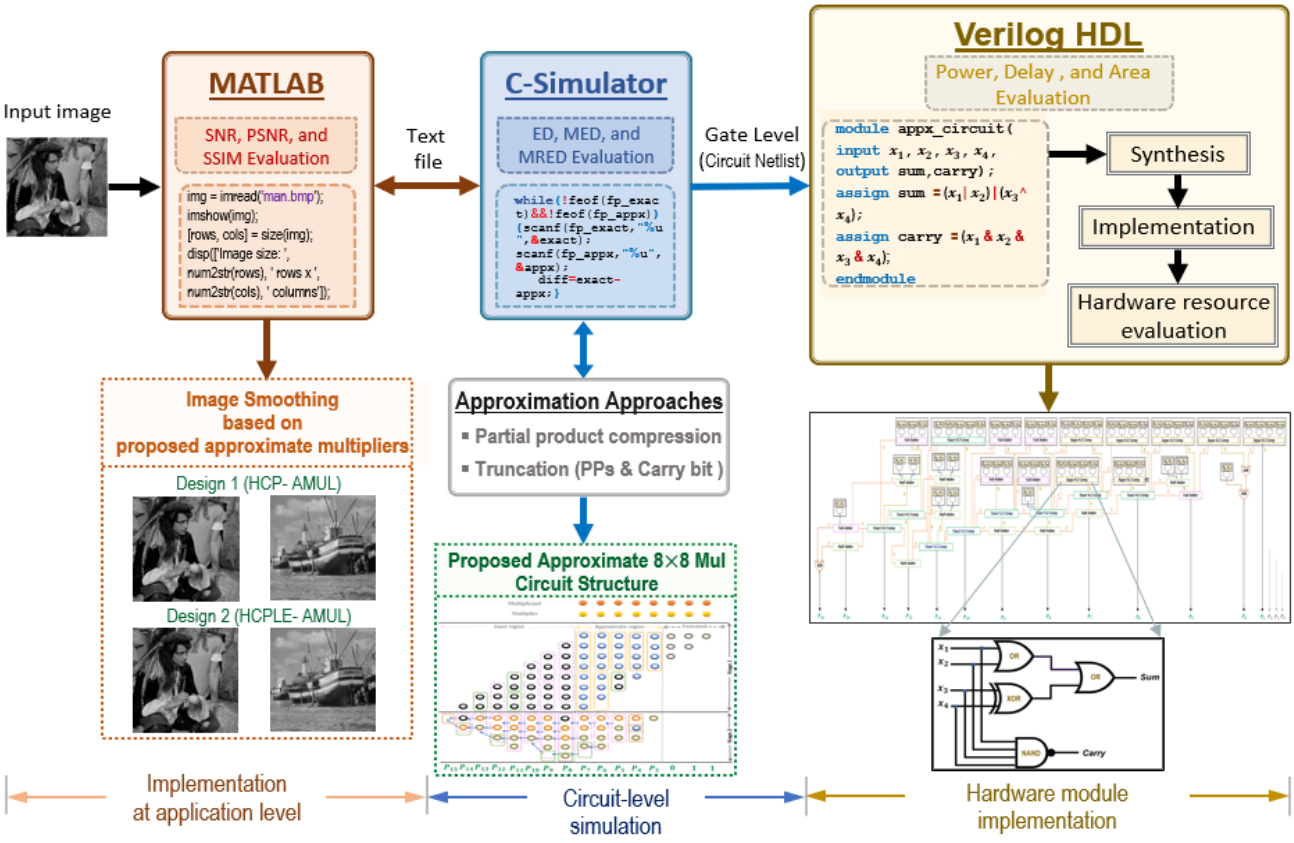


Figure. 3 The framework architecture of the developed approximate logic design and evaluation

approximate 4:2 compressors and two approximate multipliers. The framework shows three main steps to develop and evaluate the proposed multiplier circuits. First, we developed a C-simulator for the 8×8 multiplier based on approaches that compress the partial products and/or truncate the propagated carry bits. The error distance is evaluated to measure the accuracy of the proposed multiplier. When the design realizes low error distance, its structure is used to participate in performing 2D convolution operations on a gray image for image smoothing and evaluate the accuracy at the application level. Subsequently, to measure the accuracy of the resulting image using the proposed multiplication circuit, the multiplication results are stored in a text file as pixel values. This file was used as the input for the multiplier circuit. The output of the multiplication is also stored in a text file that is transferred to Matlab to be converted into an image and evaluated using the image quality metrics including SSIM, SNR, and PSNR. In case the objective evaluation of the results does not show promising results, it returns to step 1 (circuit-level simulation) to modify the structure of the proposed multiplier by using exact adders, lowering the truncation of the carry bits, or a combination of both techniques. In step 3, the structure of the developed multiplier is configured using Verilog as HDL language to evaluate the proposed approximate multiplier in terms of propagation delay, power, and area overhead (hardware resources utilization). Therefore, three main programming languages and IDE editors (C language, Matlab scripting code, and Verilog) were used in this study.

3.2 Proposed approximate 4:2 compressor

In the digital logic circuits design, adders and multipliers components incur a considerable amount of area and power overhead, and therefore, equivalent approximated circuits are becoming increasingly sought. In order to achieve high efficiency for addition operations, we develop two compressor circuits. These compressors perform compression on partial products of the multiplication to provide compressed partial products and reduce the critical path of downstream logic, which is the most critical stage in multiplication operations, thus increasing their overall energy efficiency [12]. Here, we concentrate on designing high-speed and energy-efficient 4:2 compressors while maintaining an acceptable level of accuracy for logic computing. Compressors are designed to minimize the propagation delay by summing up partial products through a reduction tree.

Table 3. Truth table of the proposed approximate 4:2 compressors (AC_1 & AC_2)

#	x_4	x_3	x_2	x_1	CS_{AC1}	ED_{AC1}	CS_{AC2}	ED_{AC2}	$Probability_{error}$
0	0	0	0	0	10	+2	10	+2	81/256
1	0	0	0	1	11	+2	11	+2	27/256
2	0	0	1	0	11	+2	11	+2	27/256
3	0	0	1	1	10	0	11	+1	9/256
4	0	1	0	0	11	+2	11	+2	27/256
5	0	1	0	1	10	0	11	+1	9/256
6	0	1	1	0	10	0	11	+1	9/256
7	0	1	1	1	11	0	11	0	3/256
8	1	0	0	0	11	+2	11	+2	27/256
9	1	0	0	1	10	0	11	+1	9/256
10	1	0	1	0	10	0	11	+1	9/256
11	1	0	1	1	11	0	11	0	3/256
12	1	1	0	0	10	0	10	0	9/256
13	1	1	0	1	11	0	11	0	3/256
14	1	1	1	0	11	0	11	0	3/256
15	1	1	1	1	00	-4	01	-3	1/256
Error Rate (ER):					$P_{error}(+2,-4)$ $= 190/256$ $= 74.22\%$		$P_{error}(+2,+1,-3)$ $= 235/256 = 91.80\%$		

However, this can introduce errors in specific outputs [7]. We proposed two compressor designs (AC_1 and AC_2) that reduce errors in compressing the outputs (carry and sum). An error rate (ER) of (18.75% and 37.5%), respectively is achieved, i.e., ER indicates number of erroneous computed outputs among all possible combinations with their probability of error, (6 and 12), respectively are erroneous out of a total of 32 outputs. Meanwhile, approximate approaches significantly shrink the area and lower the complexity of computing logic.

3.2.1. First proposed approximate 4:2 compressor design

The logic operations of the proposed approximate 4:2 compressor design 1 (AC_1) are expressed in Eqs. (13) and (14), where it consists of four inputs (x_1, x_2, x_3, x_4) and two outputs (sum, carry). Fig. 4 shows the gate level of the AC_1. As can be seen, the adder is formed with XOR gates while the carry circuit is further approximated to lower the complexity of computational logic. On the other hand, the probability of error for carry and sum outputs of the presented AC_1 compressor compared to the exact compressor is evaluated and listed in Table 3. The error distance is calculated based on considering all input combinations. Notably, 6 possible input combinations, i.e., (0000, 0001, 0010,0100,1000,1111), provide erroneous output for the carry bit.

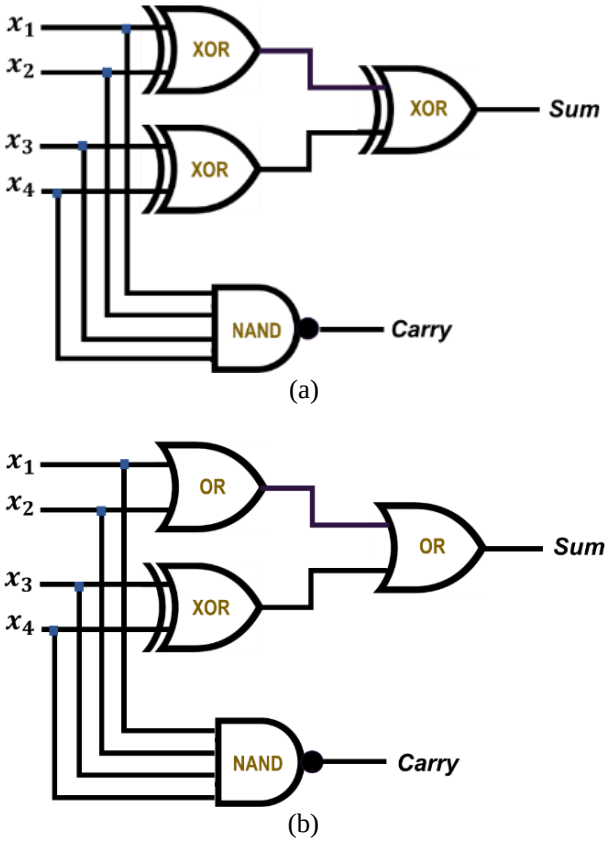


Figure. 4 Circuit schematic of the proposed approximate 4:2 compressors: (a) AC_1 and (b) AC_2

This is composed of an error probability of $+2$ and -4, which is $190/256 = 0.74$. For instance, the probability of input combination 0000 is 81/256. This

is due to the output of an AND gate having a probability of ¼ of logic '1' and a probability of ¾ of logic '0' for the initial partial product. Therefore, considering the probability of inputs, the error rate (ER) of design AC_1 is considered high as compared to that of the exact 4:2 compressor circuit. This is due to 190 input combinations out of 256 possible combinations might produce a faulty carry bit, leading to causing 74.22% probability of error. Furthermore, the error distance (ED) is evaluated by subtracting the approximated output from the exact one as expressed in Eq. (15).

$$Sum = (x_1 \oplus x_2) \oplus (x_3 \oplus x_4) \tag{13}$$

$$Carry = \overline{x_1 \cdot x_2 \cdot x_3 \cdot x_4} \tag{14}$$

$$ED = M_{exc} - M_{apx} \tag{15}$$

$$ER = \sum_i |ED_i| \times P_i \tag{16}$$

Where M_{exc} and M_{apx} indicate the exact and approximate output, respectively, and P_i is the probability of the i-th input pattren having a faulty output. As noticed, the sum is obtained from three 2-input XOR gates that incur 24 transistors while the carry incurs 8 transistors, which is required to construct a single 4-input NAND gate. The logic circuit of compressor AC_1 incurs reduced logic area compared to the exact 4:2 compressor, however, it imposes 74.22% probability of error. However, it achieved improved latency and excellent power consumption efficiency.

3.2.2. Second proposed approximate 4:2 compressor design

The proposed compressor circuit (AC_2) achieves fewer transistors and reduces power consumption. Fig. 4 illustrates the gate-level schematic of the proposed AC_2 compressor, where the carry is generated from a single 4-input NAND gate, thereby incurring 8 transistors. While the sum is obtained from two 2-input OR gates and a single 2-input XOR gate, thus it incurs 20 transistors. Eq. (17) provides the logical expression of the sum, while the logic computation of carry remains the same as in AC_1. The analysis of error probabilities and their locations for the proposed AC_2 compressor is presented in Table 3. As observed, the input combination patterns of (0000,0001,0010,0011,0100,0101,0110,1000,1001,1010,1111), which indicates that 12 bits of 32 possible output bits are inaccurate. However, the AC_2 imposes an error probability of roughly 91.80%, which implies that 235 out of 256 output combinations might be erroneous output. Although there is a slight accuracy sacrifice in this design, it is still acceptable to save area and energy.

$$Sum = (x_1 + x_2) + (x_3 \oplus x_4) \tag{17}$$

3.3 Proposed approximate 8×8 multiplier modules

Approximate multipliers compute approximate results by dividing multiplication into one accurate and one approximate part. The approximation approach is mainly performed based on compressing partial products, truncating Least Significant Bits (LSBs), or/and truncating the propagation of the carry bits (c_{in} and c_{out}) depending on their locality and importance. This can lead to significantly reducing logic design complexity and increasing energy efficiency while maintaining acceptable accuracy.

In this section, we present two approximate 8×8 multiplier designs by dividing the operation into truncated, precise, and approximate portions and employing the proposed approximate compressors in the approximate region. Thus, the partial products are split into three areas: truncated, approximate, and accurate area, where approximate compressors are leveraged at the least significant bits. However, at the most significant bits, we employ exact compressors and full and half adder circuits. Generally, the multiplication process is divided into three parts [32]: a) generating partial products (PPs) by multiplying each bit of the multiplier with the bits of the multiplicand, b) accumulating partial products using adder and compressor circuits, and c) performing carry ripple adder to obtain the final multiplication result. Therefore, there are two possible major operations for approximating multiplication. However, the second step, which involves accumulating the partial product bits, is the most crucial and complex part of the multiplication process. This is due to incurring considerable computational logic, thereby causing long propagation delays [14]. Hence, it has been sought to compress these bits efficiently and improve the energy efficiency and performance of multipliers while maintaining a high level of computing accuracy. Thus, we leveraged the proposed 4:2 compressor circuits to build 8×8 multiplier circuits. These proposed approximate multipliers perform trade-offs between design overheads (power, area, delay, and computing accuracy) to achieve desirable properties that meet low-power and high-speed requirements. Next, we discuss our two proposed approximate multiplier circuits.

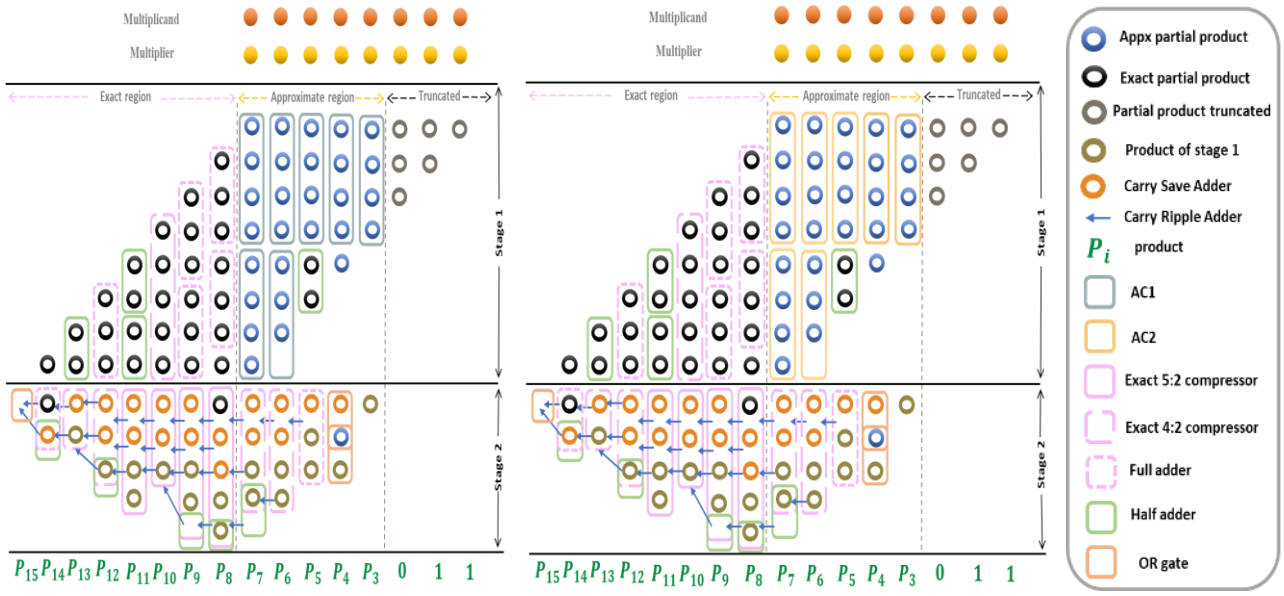
3.3.1. Proposed high computing performance approximate 8×8 multiplier

Here, we propose a very high-speed multiplier, namely the High Computing Performance Approximate Multiplier (HCP-AMUL) circuit, which concentrates on truncating and approximating the partial products in two stages. First, a thorough analysis was performed to examine how many least significant bits can be truncated while preserving an acceptable computing accuracy. As can be noticed in Fig. 5, the proposed 4:2 AC_1 approximate compressor is utilized to accumulate partial products at the lower portion, where AC_1 is placed in columns 3 through 7. Thus, we aim to minimize the potential error resulting from approximation by adapting the proposed AC_1 compressor at least significant bits to prevent a significant impact on precision. In contrast, approximations in the second half of the partial product columns can lead to significant loss of accuracy. Therefore, we used exact compressors (4:2 and 5:2) and full and half adder circuits in the second portion. In stage 1, seven AC_1 components, and one half-adder circuit are used in the approximate region to compress the PPs of the presented HCP-AMUL. On the other hand, three half-adder circuits, five full-adder circuits, and one exact 4:2 compressor are deployed in the precise region. Moreover, the full and half adder circuits are utilized to add residual partial products from column 9 to column 14 and to maintain high computing accuracy of addition operations.

Furthermore, five half-adders, two full adders, three OR gates, four exact 4:2 compressors, and four exact 5:2 compressors are utilized for summing up PPs in stage 2. In this stage, rather than using approximate circuits, we utilized OR gates in columns 5 and 15. This has saved an area reduction of 48 transistors to slightly lowering aera-power overheads while preserving acceptable computing accuracy. However, the key point that has made the proposed approximate multipliers deliver high computing performance is employing the Carry Save Adder (CSA) tree in stage 1 whereas the Carry Ripple Adder (CRA) circuit is used in stage 2 to compute the final bits of multiplication. This is due to the CSA addition circuit delays the computations of carry bits, therefore, all PPs bits can be accumulated concurrently, realizing high parallelism in stage 1. In this context, the delay of the carry bits propagation can only be seen in stage 2 with the CRA, as shown in the architecture of the proposed HCP-AMUL multiplier in Fig. 5. Theremore, major contribution to reducing the delay was made by considering the input sequence (PPs and c_{in}) of all compressors and adder circuits used.

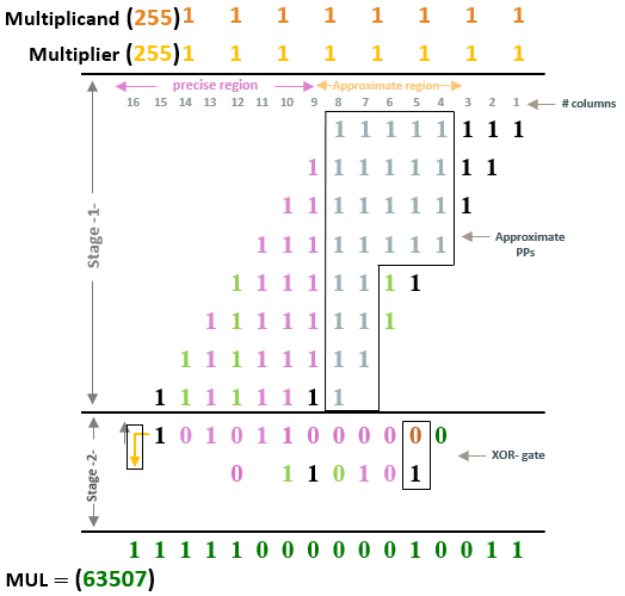
This sequence indicates a positive or negative effect on the performance of the design, i.e. the performance improves the later the inputs which is produced from the previous stage are added. Additionally, we perform a comprehensive analysis based on the proposed C simulator with Matlab image processing validation using the proposed HCP-

AMUL circuit. It has been observed that truncating the first three bits (bit0 to bit2) of the PPs increases the absolute mean error (AME) by 0.76%, due to which we set the first three bits to 011, resulting in a reduction of the AME to 0.65%.

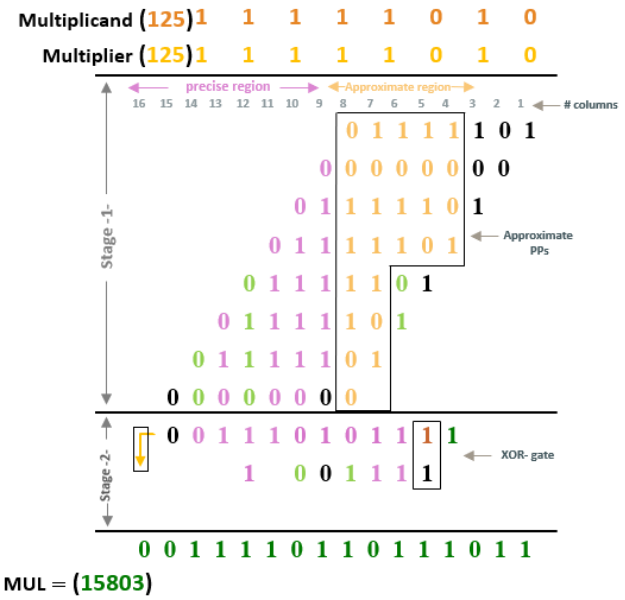


HCP-AMUL HCPLE-AMUL

Figure. 5 The structure of the proposed approximate multipliers



(a)



(b)

Figure. 6 A case study of the proposed multipliers: (a) HCP-AMUL and (b)HCPLE-AMUL

It does not considerably affect the quality of the processed image. Likewise, it slightly improves the speed up, i.e., eliminating the propagation delay of two adder circuits for columns 2 and 3, in stage 2 and reduces area-power overheads, saving six 2-input AND gates, two 2-input XOR gates, and one 4:2 compressor circuit. Fig. 6 illustrates a case study example of HCP-AMUL where the multiplier is equal to 255 and the multiplicand is 255, i.e., the area of approximation is indicated with a rectangular. The approximated result is 63507 which is achieving 1518 error distance in this case. However, an overall average error, Mean Error Distance (MED), of 329.50 is achieved. Therefore, to improve the computing accuracy of the proposed HCP-AMUL with higher energy and area efficiency, we proposed an improved version of design 1 using the proposed AC_2 compressor, which we discuss next.

3.3.2. Proposed high computing performance low error approximate 8×8 multiplier

To enhance the prior design's (HCP-AMUL) computational precision and energy consumption, we proposed design (2), namely High Computing Performance Low Error Approximate Multiplier (HCPLE-AMUL), as depicted in Fig. 5. This design is similar to (HCP-AMUL), however, it replaces the AC_1 with the AC_2 in partial product columns (3 to 7) of stage 1. Therefore, in the first stage of this design, seven AC_2 circuits and one half-adder circuit are used in the approximate region, while three half-adders, five full-adders, and one exact 4:2 compressor are utilized in the precise region. Moreover, in the second stage, we employ CRA to sum up the propagated carry bits with the output bits of stage 1 and compute the final production bits, thus stage 2 is identical to that of the proposed approximate multiplier design 1. Fig. 6 provides a case study example of HCPLE-AMUL where the multiplier is equal to 125 and the multiplicand is 125. The approximated result is 15803 which produces an error distance of 178; however, the MED for this design is 304.84. It is evident that this design achieves lower MED since the proposed approximate compressor (AC_2) provides less MED, which compensates for errors resulting from truncating the first three LSBs. In both proposed multipliers (HCP-AMUL, and HCPLE-AMUL), we used the CSA in stage 1 and the CRA in stage 2. This assists in implementing the proposed multipliers to incur two stages, leading to shrinking the propagation delay and lowering the power consumption. Unlike our proposed multipliers, approximate multipliers presented in the prior works [3, 16], require multiple stages to sum up partial products, which incurs additional exact components to compute the propagated carry bits and produce the output bits of the multiplication.

The general schematic diagram of the proposed approximate 8×8 multiplier is depicted in Fig. 7. It illustrates where the proposed approximate compressors have been employed. Furthermore, it shows where the precise compressors and full and half adder circuits are placed. As seen, the first three LSBs are truncated with setting it to (011), while other logic computing blocks each has two output bits, where the sum is indicated by (s) and the carry by (c). Moreover, the proposed compressors were deployed in the lower portion to accumulate the sum of partial

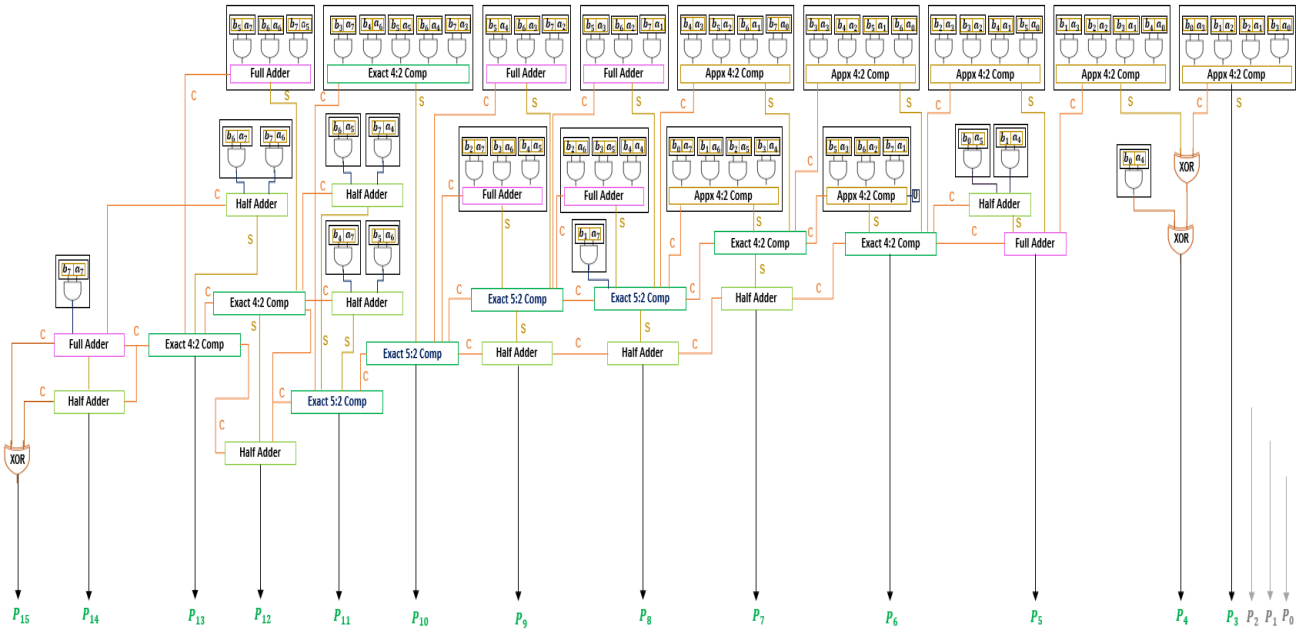


Figure. 7 General diagram of the proposed high computing performance approximate multiplier circuits

products. Also, the addition of carry bits propagation has been delayed to stage 2 to perform it using the carry ripple adder. This has improved the speedup of the presented multiplier, boosted 7% as compared to the exact design. Meanwhile, the computing accuracy has been maintained to realize acceptable error tolerance, realizes 0.47×10^{-2} for NMED, Additionally, when compared to earlier studies, it obtained the superior balance of 449.85 PDAEP between energy expenditure, area, and error. Thus, the proposed HCPLE-AMUL circuit can be employed in error-tolerant applications where a small number of errors can be tolerated, such as image processing and neural network applications.

4. Experimental results setup and analysis

The proposed framework was simulated using VS Code as an Integrated Development Environment (IDE) for C language programming to evaluate the accuracy of NED, MED, and MRED terms of the proposed multipliers, whereas Matlab R2022b was used to evaluate the accuracy of the proposed multipliers by deploying them to perform multiplication operations in 2-Dimentional (2D) with filler kernel of (3×3) for image smoothing at application-level. We calculated image processing evaluation metrics such as SNR, PSNR, and SSIM. For hardware implementation of the proposed approximate 4:2 compressors and 8×8 multipliers, we synthesized them based on using Xilinx Nexys 4 Artix-7 (XC7A100T-1CSG324) training board using Verilog Hardware Description Language (HDL) with Vivado 2018.2 to evaluate the power and delay of proposed compressors and multipliers. As a notice, in order to ensure the accuracy and consistency of the results, all previous works with which we compared our proposals were simulated and implemented under identical hardware and environmental conditions. This approach guarantees the reliability of all measurements mentioned in this paper. Next, we discuss the evaluation of the proposed approximate 4:2 compressors and 8×8 multipliers.

4.1 Performance evaluation of proposed compressors

To assess the efficiency of the approximate 4:2 compressors, in terms of delay, power consumption, and area, we compared with the precise design and previously published compressors. Table 4 presents a comparison between our proposed compressors AC_1 and AC_2, and previous works, in terms of power, delay, area, and PDP. Compressor CP1 [3] achieves high accuracy at the expense of high-power consumption and delay. Also, Compressor [10] achieves higher delay, high power consumption, and lower accuracy. While our proposed approximate 4:2 compressors (AC_1 and AC_2) outperform previous proposes and the precision compressor in terms of delay, achieving a minimum delay of (0.938ns). Furthermore, our proposed compressors show improvements in area by (50% and 50%), respectively, compared to the precision 4:2 compressor, and they achieve power savings, refer to Eq. (23), of (20.17% and 24.37%), respectively, as

Table 4. Performance evaluation of the proposed compressor circuits

Design	Possible ED (out of 32)	NED	# of LUT+ Flipflop	Propagation Delay (ns)	Total Power (mW)	Power Delay Product (PDP (*PJ*))	FOM (1-NED/PDP)
Exact	–	–	4	1.230	119	146.37	–
Sayadi AC6G-12 [29]	11	0.34375	2	1.099	116	127.484	0.005148
Ghanatabadi Design 1 [32]	8	0.25	2	1.099	117	128.583	0.005833
Beura [10]	9	0.28125	2	1.099	117	128.583	0.00559
Thakur [26]	10	0.3125	2	1.099	95	104.405	0.006585
Zhang [27]	10	0.3125	2	1.099	115	126.385	0.00544
VanToan CP1 [3]	2	0.0625	2	1.099	117	128.583	0.007291
VanToan CP2	8	0.25	2	1.099	116	127.484	0.005883
Proposed AC_1	6	0.1875	2	0.938	95	89.11	0.009118
Proposed AC_2	12	0.375	2	0.938	90	84.42	0.007403

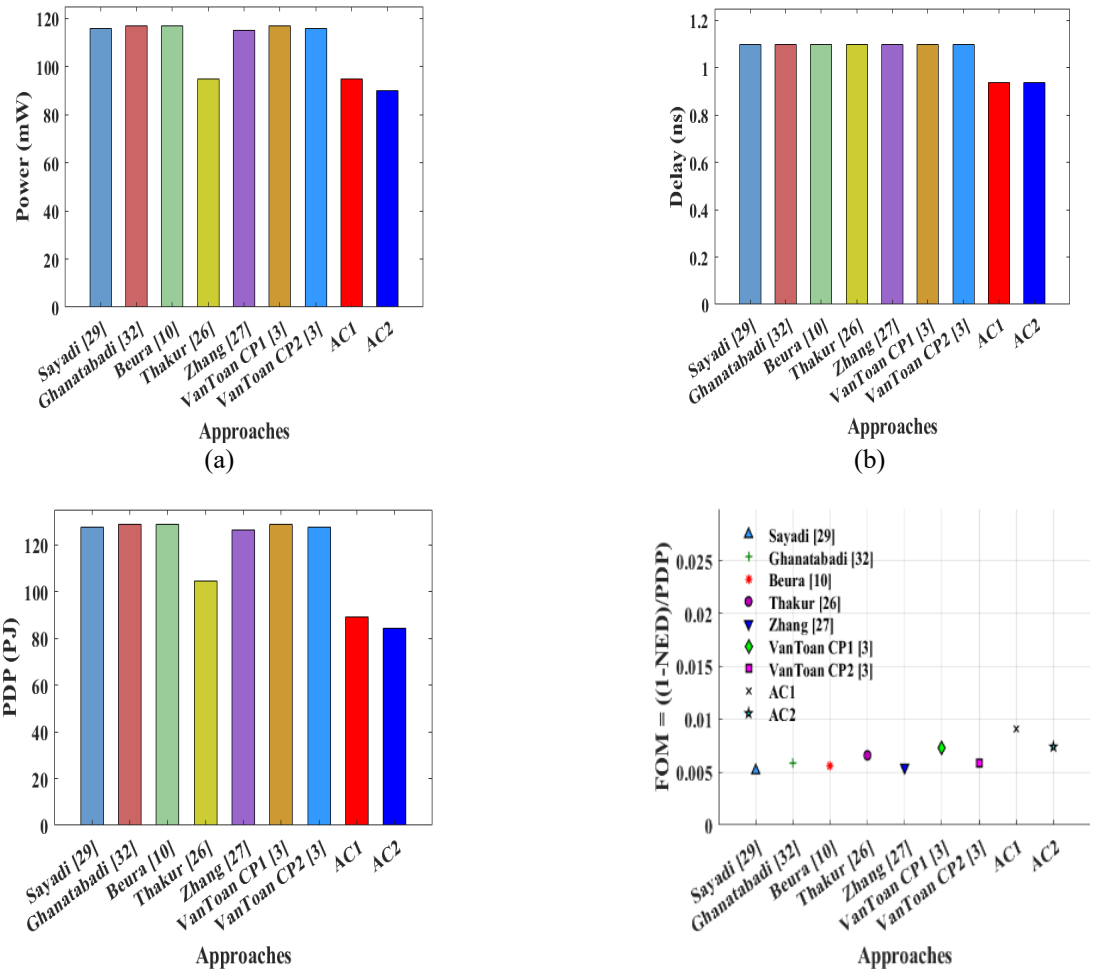


Figure. 8 Performance analysis of the different approximate compressors: (a) Power Evaluation, (b) Propagation Delay Evaluation, (c) Power Delay Product, and (d) Figure of Merit

Table 5. Performance evaluation of the proposed compressor circuits compared to selected prior works

Design	Area saving (%)	Speedup (%)	Power saving (%)	Accuracy (%)
Exact	baseline	baseline	baseline	baseline
Sayadi AC6G-12 [29]	50	10.65	2.521008	65.625
Ghanatabadi Design 1 [32]	50	10.65	1.680672	75
Beura [10]	50	10.65	1.680672	71.875
Thakur [26]	50	10.65	20.16807	68.75
Zhang [27]	50	10.65	3.361345	68.75
VanToan CP1 [3]	50	10.65	1.680672	93.75
VanToan CP2	50	10.65	2.521008	75
Proposed AC_1	50	23.74	20.16807	81.25
Proposed AC_2	50	23.74	24.36975	62.5

shown in Table 5. Additionally, Fig. 8 demonstrates an improvement in power for the proposed compressors, the one shown in Fig. 8 achieves the favorable improvement in delay compared to the presented works and the exact compressor. While our proposed compressors (AC_1 and AC_2) deliver the benefit of PDP, see Fig. 8 in, compared to the other approximate 4:2 compressors. They demonstrate effective PDP, achieving (89.11 and 84.42), for design 1 and 2, respectively, as represented in Eq. (24) [2]. However, AC_2 is conserved the improved approximate 4:2 compressor which achieves a superior balance between acceptable accuracy and hardware efficiency.

4.2 Accuracy evaluation of proposed compressors

To calculate the error rate of AC_1 and AC_2. Based on Table 3, |ED| for each error occurrence is equal to $+2, -3, and\ -4$, respectively, considering Eq. (16), the sum of all probabilities of AC_1 and AC_2 in Table 3 determines the error rate of the proposed compressors. Thus, the error rate of AC_1 and AC_2 is equal to 74.22% and 91.80%, respectively. This is because 190 and 235, respectively input combinations result in a faulty carry bit out of 256 possible combinations. In order to thoroughly compare the various approximate compressors that have been studied, used a figure of merit (FOM) parameter. The favorable FOM expresses to understanding the tradeoff between different parameters of approximate compressors. Eq. (18) formulates the FOM expression of compressors relying on the PDP and accuracy depending on NED. A higher FOM value indicates better performance. As depicted in Fig. 8, our proposed approximate compressors (AC_1 and AC_2) outperform prior works in the literature, achieving high FOM values of 0.009118 and 0.007403, respectively. This demonstrates an effective trade-off between error tolerance and energy efficiency.

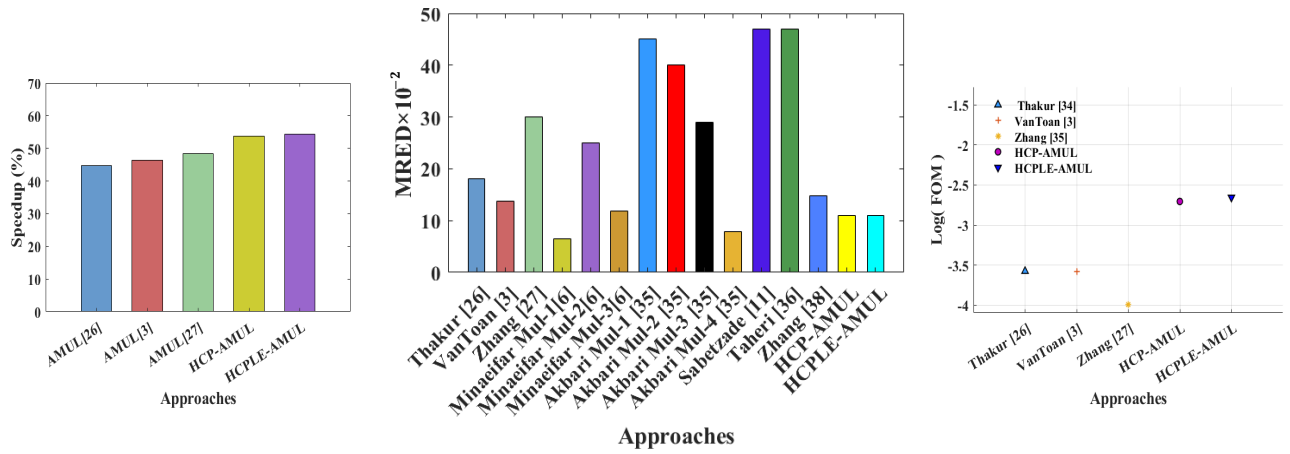
$$FOM = \frac{(1-NED)}{PDP} \tag{18}$$

4.3 Performance evaluation of proposed approximate multipliers

To assess the hardware efficiency of our designs, we implement the proposed multipliers at various precisions, compressing the least significant columns of partial products with the proposed approximate 4:2 compressors and compressing the high significant columns with precise compressors and full- and half-adders. The suggested multipliers are compared to exact multipliers on the precise compressors, half adder and full-adder circuits, and earlier works in terms of delay, area, power consumption, PDP, and PDAEP as shown in Eq. (25) [2]. Notably, the propagation delay is determined by the Register Transfer Level (RTL) of the reconfigurable approximate multipliers. Table 6 summarizes the comparison between our 8×8 multipliers and prior works in terms of electrical performance (i.e., delay, power, area), PDP, and PDAEP. Our proposed 8×8 multipliers (HCP-AMUL and HCPLE-AMUL) outperform previous works and the exact multiplier in terms of delay, it achieved shorter delay by (5.819ns and 5.752ns) respectively, while comparative works achieve superior in terms of area reduction, refer to Eq. (23), while our proposals improve in area by (39.81% and 39.81%) respectively compared to the exact multiplier. The power saving efficiency is estimated to be (9.66 and 11.72) respectively compared to exact design and previous researches. The PDP and PDAEP gains can be respectively, (762.29 and 505.40) for design 1 and (736.26 and 449.85) for design 2. Additionally, Fig. 9 demonstrates a significant improvement in speedup for the proposed approximate multipliers, refer to Eq. (23), while our proposed multipliers in Fig. 9 provide a superior improvement in MRED (0.112 and 0.110) respectively, compared to the approximate multiples provided.

Table 6. Performance evaluation of the proposed multiplier circuits compared to selected prior works

Multiplier Design	# of LUT + Flipflop	Delay (ns)	Power (mw)	Area Reduction (%)	Speedup (%)	Power Saving (%)	Power Delay Error Product (PDP (*PJ*))	Power Delay Area Error Product (PDAEP)
Exact	216	12.576	145	baseline	baseline	baseline	1823.52	baseline
Thakur [26]	92	6.959	138	57.407	44.66	4.827586	960.342	3622.410
VanToan [3]	88	6.737	140	59.259	46.43	3.448275	943.18	3635.393
Zhang [27]	98	6.489	123	54.630	48.40	15.17241	798.147	8729.174
Proposed HCP-AMUL	130	5.819	131	39.815	53.73	9.655172	762.289	505.398
Proposed HCPLE-AMUL	130	5.752	128	39.815	54.26	11.72414	736.256	449.852



(a) (b) (c)

Figure. 9 Performance analysis of the different approximate multipliers: (a) Speedup Evaluation, (b) Mean Relative Error Distance, and (c) Figure Of Merit

Measurements show that there must be a trade-off between power consumption, delay, area, and accuracy, but achieving a balance between these measures is critical to keeping up with current technology. It is clear that our proposed approximate multipliers achieve an effective balance between accuracy and hardware efficiency compared to previous work.

4.4 Accuracy evaluation of approximate multipliers

Here, we present a comparative study between our proposed multipliers and other comparable works about the multipliers' accuracy. The accuracy measures that we employed were the error rate (ER), mean error distance (MED), mean relative error distance (MRED), and normalized error distance (NED). The accuracy measurements of the proposed approximate multipliers and previous works are outlined in Table 7. The approximate multipliers [3, 6, 11, 26, 27, 35, 36, 38] have the lowest quality in terms of MED and MRED, while the Minaeifar Mul-1 multiplier [6] demonstrates the best quality in terms of MED and MRED compared to the previous works listed in Table 7, but at the cost of using up more hardware resources. The accuracy improves as the number of most significant partial product columns compressed using accurate compressors and adders increases. While our proposals (HCP-AMUL and HCPLE-AMUL) exhibit the highest quality and accuracy compared to previous works, although the error probability of HCPLE-AMUL is (91.80%), theremore, the greatest importance should be given to the quality and accuracy in terms of MED, NMED and MRED, which are distinguished by (304.84, 0.47×10^{-2}, 11.05×10^{-2}) respectively as illustrate in Table 7.

Table 7. Evaluation of mean error distance analysis of the proposed multiplier circuits compared to selected prior works

Approximate Design	MED	NMED× 10^{-2}	MRED× 10^{-2}
Thakur [26]	2664.49	4.1	18.01
VanToan [3]	2845.83	4.38	13.79
Zhang [27]	7258.49	11.16	30.09
Minaeifar Mul-1[6]	260.10	0.4	6.5
Minaeifar Mul-2	2731.05	4.2	25
Minaeifar Mul-3	845.32	1.3	11.8
Akbari Mul-1 [35]	3966.53	6.1	45
Akbari Mul-2	3511.35	5.4	40
Akbari Mul-3	2601.00	4	29
Akbari Mul-4	1365.53	2.1	7.8
Sabetzade [11]	520.20	0.8	47
Taheri [36]	3251.25	5	47
Zhang [38]	1040.40	1.6	14.8
Proposed HCP-AMUL	329.50	0.51	11.28
Proposed HCPLE-AMUL	304.84	0.47	11.05

This is attributed to the use of AC_2 in the least significant partial product columns, and which realizes a low error rate compared to previous researches. Eqs. (20) to (22) are used to evaluate the error distance analysis. While Table 7 lists the analysis of mean error distance of the proposed multipliers compared to selected previously ones introduced. In order to give a thorough comparison of the many examined approximate multipliers with respect to their efficacy, a figure of merit (FOM) parameter is needed. The suitable FOM refers to comprehending the tradeoff between various parameters of approximate multipliers. Eq. (19) computes the FOM of multipliers in terms of PDAEP, refer to Eq. (25) [2] and accuracy. A higher FOM value indicates superior performance. As shown in Fig. 9, our proposed designs (HCP-AMUL and HCPLE-AMUL) outperform the previously presented approximate multipliers, where they achieved high FOM by (505.40 and 449.85) respectively, achieving an effective trade-off between fault tolerance, area and energy efficiency. Although other designed multipliers consume less transistor count, while our proposed multiplier has the lowest energy consumption and superior accuracy.

$$FOM = \frac{1-NED}{PDAEP} \quad (19)$$

$$MED = \frac{1}{2^{2N}}\sum_{i=1}^{2^{2N}}\left|M_{exc} - M_{apx}\right| \quad (20)$$

$$MRED = \frac{1}{2^{2N}}\sum_{i=1}^{2^{2N}}\frac{|ED_i|}{M_{exc}} \quad (21)$$

$$NED = \frac{MED}{(2^N-1)^2} \quad (22)$$

$$Change(\%) = \frac{V_{exc}-V_{appx}}{V_{exc}} \times 100 \quad (23)$$

$$PDP = Power \times Delay \quad (24)$$

$$PDAEP = Power \times Delay \times Area \times Error \quad (25)$$

The percentage change formula is used to calculate the relative difference between the exact value (V_{exc}) and an approximate value (V_{appx}). Similarly, M_{exc} and M_{apx} represent the exact and approximate values of partial products, respectively. ED_i represents the error distance of i-th combination and NED (Eq. (22)), indicates the normalized value of error distance relative to the possible maximum value of error (Eq. (20)), [8], Where MED is defined as the mean absolute error distance between the exact and approximate outputs when all possible inputs are applied to the multiplier (Eq. (20)) [35]. Where N is the bit length of the multiplier. While MRED is the mean absolute distance between the exact and approximate outputs divided by the exact output for all inputs (Eq. (21)) [35]. Considering all possible combinations (65,536), the proposed HCPLE-AMUL provides the best reduced MED and MRED of 304.84 and 11.05×10^{-2} respectively, compared to other proposed approximate multipliers and selected previously published multipliers. it provides the best tradeoff between design complexity in terms of area, power, and delay and error resilience.

4.5 Accuracy evaluation of image smoothing

Here, to evaluate the processing quality of the presented AMULs, they are leveraged in real-world applications. They were employed to perform image smoothing as an important image-processing application and evaluate their accuracy. The formula used to perform 2D convolution for image smoothing on the input image is defined in Eq. (26) [35].

$$II'(i,j) = \sum_{k=-1}^{1}\sum_{l=-1}^{1} kernel_{smoothing} \cdot I(i+K, j+l) \quad (26)$$

Table 8. Evaluation of image smoothing based on PSNR, SNR, and SSIM metrics

Multiplier Design	SNR (dB)	PSNR (dB)	SSIM
Thakur [26]	14.5093	22.2129	0.99941
VanToan [3]	18.0591	25.7627	0.9998
Zhang [27]	16.1634	23.867	0.9996
Proposed HCP-AMUL	18.0311	25.7347	0.9998
Proposed HCPLE-AMUL	18.0311	25.7347	0.9998

Where $I(i+k, j+l)$ represents pixels' values of $k \times l$ window taken from the original image, while the smoothing mask matrix is expressed as in Eq. (27).

$$Kernel_{smoothing} = \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix} \quad (27)$$

It is worth noting that all other operations, such as addition, subtraction, and division, are regarded precise. Peak signal-to-noise ratio (PSNR) and the Structural Similarity Index Measure (SSIM) are calculated to evaluate the accuracy of the resulting images [14]. The SNR for a single image quantifies the ratio of the desired signal to the background noise, and the PSNR metric is defined as in Eq. (29) [26]:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2 \quad (28)$$

$$PSNR = 10\ log_{10}\left(\frac{MAX_i^2}{MSE}\right) \quad (29)$$

In this equation, MAX_i represents the image's maximum pixel value, m and n are its dimensions, and $I(i,j)$ and $K(i,j)$ are the precise and estimated values for each pixel [39], respectively. PSNR is a widely used metric for image quality objective assessment. The SSIM is another metric for image quality measurement that evaluates the structural similarity of the exact and approximate images based on the point in which a human visual system is capable of extracting information based on the image structure [39]. The SSIM and SNR are defined in Eqs. (31) and (32) [2], concurrently.

$$SSIM = \sum_{j=1}^{M} \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (30)$$

$$SNR = 10\ log_{10}\left(\frac{Signal\ Power}{MSE}\right) \quad (31)$$

(a) (b) (c)

(d) (e) (f)

Figure. 10 Image smoothing based on utilizing different multipliers: (a)Exact Multiplier, (b)AMUL [26], (c)AMUL [3], (d)AMUL [27], (e)HCP-AMUL, and (f) HCPLE-AMUL

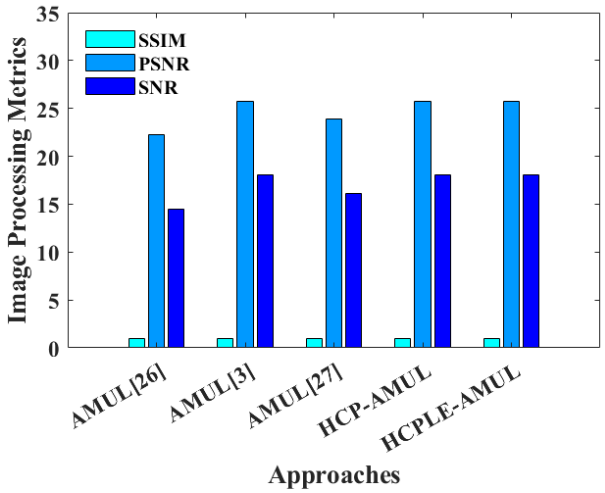


Figure. 11 Image smoothing based on utilizing different approximate multipliers

Where the signal Power is the power of the original image signal. For a grayscale image, this can be computed as the mean of the squared pixel intensities. While Mean Squared Error (MSE) is the average of the squares of the differences between the original and reconstructed processed image with approximate multipliers. The values of PSNR, SNR, and SSIM for image smoothing are listed in Table 8. As observed, the proposed HCP-AMUL and HCPLE-AMUL provide improved values of PSNR and SNR. Both proposed approximate designs achieve (18.03 and 25.73) for SNR and PSNR, respectively. While the designs presented in [26] realize (14.509 and 22.21). Similarly, the design proposed in [27] provides (16.16 and 23.86) for SNR and PSNR concurrently. This confirms the validity of the proposed design to deliver improved error tolerance at the application level of image smoothing. Notice that the approximate multiplier presented in [3] provides marginally improved metrics, however, it incurs a higher propagation delay. Fig. 10 summarizes the image processing metrics for the proposed approximate multipliers compared to recent relative designs. It can be concluded that the difference between the outputs of the exact and the approximate multipliers can be ignored.

Finally, Fig. 11 depicts all image processing evaluation metrics for the proposed multipliers compared to recently published relative ones. The proposed AMULs deliver competitive results, thereby they can be adapted to provide an improved tradeoff between computing accuracy and error resilience for error-tolerant applications.

Although our proposed compressors have a high error probability, their accuracy is acceptable, particularly when included in our multipliers, as evidenced by the accuracy measurements listed in Table 7 at the gate level and Table 8 at the application level, which are both high and acceptable in comparison to previous research.

In summary, the proposed framework architecture, compressors, and the developed approximate multiplier circuits can potentially be used with neural networks to perform classification tasks based on approximate adder and multiplier circuits while evaluating prediction accuracy for deep neural network-based applications. This is due to the large-scale of technology node size scaling and the intensive development of big data processing leads to making energy efficiency the biggest drawback. Thus, in the design of digital logic circuits for image-based applications, it has been advantageous to approximate adder and multiplier circuits at the gate level. Hence, in this paper, we have employed approximate computing approaches to address the energy concern and achieve favorable energy saving while preserving a high level of computing accuracy.

5. Conclusion and future directions

Energy-efficiency and complexity of computational accuracy have been investigated intensively in the design of approximate compressors and multipliers. However, approximate multipliers require careful analysis for trading off accuracy over efficiency based on the demanded quality constraints of specific applications. Hence, in this scholarly study, a framework architecture that evaluates the design of approximate 8×8 multiplier is proposed. The framework utilizes Matlab along with VS Code for accuracy evaluation, meanwhile, Vivado was used for design implement on an FPGA platform (Xilinx Nexys 4 Artix-7. Furthermore, two main high-speed multipliers are developed (HCP-AMUL and HCPLE-AMUL). Their approximation design was tailored by compressing the PPs using the proposed 4:2 compressors. Moreover, to realize energy-efficiency in designing the developed multiplier circuits, a truncation scheme that truncates the PPs along with the propagation of carry bits was leveraged using the proposed 4:2 compressors. Integrating the developed high computing performance approximate multiplier that delivers improved error tolerance (HCPLE-AMUL) into an image smoothing achieves significant improvement in performance metrics such as high-speed performance, energy-efficiency, and resource utilization while preserving competitive image computation quality (MRED of 11.05×10^{-2}). Finally, we aim at extending the presented 8×8 HCPLE-AMUL circuit to construct a parallel 16×16 array multiplier that can be leveraged into a lightweight artificial neural network classifier and

evaluate the predicting accuracy versus energy-efficiency, or the proposed approximate multipliers are included in the lightweight, open-source RISC-V processor allowing it to be used in devices that require low power while maintaining adequate accuracy.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptual, methodology, design, simulation, implementation, analysis, and writing draft manuscript have been done by Zahraa A. Zuhair. The supervisor and manuscript review have been done Emad A. Al-Sabawi. Both authors approve the final version.

References

[1] S. Mittal, "A survey of techniques for approximate computing", *ACM Computing Surveys (CSUR)*, Vol. 48, No. 4, pp. 1-33, 2016.

[2] F. Karimi, R. Faghih Mirzaee, A. Fakeri-Tabrizi, and A. Roohi, "Design and evaluation of ultra-fast 8-bit approximate multipliers using novel multicolumn inexact compressors", *International Journal of Circuit Theory and Applications*, Vol. 51, No. 7, pp. 3454-3479, 2023.

[3] N. Van Toan and J. Lee, "FPGA-based multi-level approximate multipliers for high-performance error-resilient applications", *IEEE Access*, Vol. 8, pp. 25481-25497, 2020.

[4] S. Venkataramani, V. Chippa, S. Chakradhar, K. Roy, and A. Raghunathan, "Quality programmable vector processors for approximate computing", In: *Proc. of the 46th Annual IEEE/ACM International Symposium on Microarchitecture*, pp. 1-12, 2013.

[5] M. Masadeh, O. Hasan, and S. Tahar, "Machine-learning-based self-tunable design of approximate computing", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 29, No. 4, pp. 800-813, 2021.

[6] A. Minaeifar, E. Abiri, K. Hassanli, M. Karamimanesh, and F. Ahmadi, "Energy efficient approximate multipliers compatible with error-tolerant application", *Computers and Electrical Engineering*, Vol. 114, p. 109064, 2024.

[7] B. Vakili, O. Akbari, and B. Ebrahimi, "Efficient approximate multipliers utilizing compact and low-power compressors for error-resilient applications", *AEU-International Journal of Electronics and Communications*, Vol. 174, p. 155039, 2024.

[8] A. Sadeghi, N. Shiri, and M. Rafiee, "High-efficient, ultra-low-power and high-speed 4: 2 compressor with a new full adder cell for bioelectronics applications", *Circuits, Systems, and Signal Processing*, Vol. 39, pp. 6247-6275, 2020.

[9] W. Liu, F. Lombardi, and M. Shulte, "A retrospective and prospective view of approximate computing [point of view", In: *Proc. of the IEEE*, Vol. 108, No. 3, pp. 394-399, 2020.

[10] S. Beura, B. Devi, P. Saha, and P. Meher, "Design of a Novel Inexact 4: 2 Compressor and Its Placement in the Partial Product Array for Area, Delay, and Power-Efficient Approximate Multipliers", *Circuits, Systems, and Signal Processing*, Vol. 43, No. 6, pp. 3748-3774, 2024.

[11] F. Sabetzadeh, M. Moaiyeri, and M. Ahmadinejad, "A majority-based imprecise multiplier for ultra-efficient approximate image multiplication", *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol. 66, No. 11, pp. 4200-4208, 2019.

[12] 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, 2020.

[13] M. Ahmadinejad and M. Moaiyeri, "Energy- and quality-efficient approximate multipliers for neural network and image processing applications", *IEEE Transactions on Emerging Topics in Computing*, Vol. 10, No. 2, pp. 1105-1116, 2021.

[14] M. Ahmadinejad, M. Moaiyeri, and F. Sabetzadeh, "Energy and area efficient imprecise compressors for approximate multiplication at nanoscale", *AEU-International Journal of Electronics and Communications*, Vol. 110, p. 152859, 2019.

[15] A. Gorantla and P. Deepa, "Design of approximate compressors for multiplication", *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, Vol. 13, No. 3, pp. 1-17, 2017.

[16] K. Naresh, Y. Sai, and S. Majumdar, "Design of 8-bit dadda multiplier using gate level approximate 4: 2 compressor", In: *Proc. of 2022 35th International Conference on VLSI Design and 2022 21st International Conference on Embedded Systems (VLSID)*, pp. 269-274, 2022.

[17] D. Esposito, A. 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.

[18] C. Chang, J. Gu, and M. Zhang, "Ultra low-voltage low-power CMOS 4-2 and 5-2 compressors for fast arithmetic circuits", *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol. 51, No. 10, pp. 1985-1997, 2004.

[19] T. Roshini, R. Krishna, P. Reddy, and M. Vinodhini, "Improved High speed approximate multiplier", In: *Proc. of 2020 4th International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech)*, pp. 1-5, 2020.

[20] K. Reddy, M. Vasantha, Y. Kumar, and D. Dwivedi, "Design and analysis of multiplier using approximate 4-2 compressor", *AEU-International Journal of Electronics and Communications*, Vol. 107, pp. 89-97, 2019.

[21] T. Kong and S. Li, "Design and analysis of approximate 4-2 compressors for high-accuracy multipliers", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 29, No. 10, pp. 1771-1781, 2021.

[22] Y. Zhang, X. Chen, P. Guo, and G. Xie, "Design and Analysis of Approximate Multiplier of Majority-Based Imprecise 4-2 Compressor for Image Processing", In: *Proc. of 2023 IEEE 23rd International Conference on Nanotechnology (NANO)*, pp. 1-5, 2023.

[23] L. Krishna, J. Rao, S. Ayesha, S. Veeramachaneni, and S. Mahammad, "Energy efficient approximate multiplier design for image/video processing applications", In: *Proc. of 2021 IEEE International Symposium on Smart Electronic Systems (iSES)*, pp. 210-215, 2021.

[24] U. Kumar, P. Bikki, S. Veeramachaneni, and S. Ahmed, "Power efficient approximate multiplier architectures for error resilient applications", In: *Proc. of 2022 IEEE 19th India Council International Conference (INDICON)*, pp. 1-5, 2022.

[25] J. Kokkiligadda, D. Jhariya, and N. Paras, "Analysis and Implementation of Approximate 4-2 Compressors for Approximate Multipliers", In: *Proc. of 2023 2nd International Conference on Paradigm Shifts in Communications Embedded Systems, Machine Learning and Signal Processing (PCEMS)*, pp. 1-6, 2023.

[26] G. Thakur, H. Sohal, and S. Jain, "Power-Area-Optimized Approximate Multiplier Design for Image Fusion", *Circuits, Systems, and Signal Processing*, Vol. 43, No. 4, pp. 2288-2319, 2024.

[27] M. Zhang, S. Nishizawa, and S. Kimura, "Area efficient approximate 4-2 compressor and probability-based error adjustment for approximate multiplier", *IEEE Transactions on Circuits and Systems II: Express Briefs*, Vol. 70, No. 5, pp. 1714-1718, 2023.

[28] F. Bahrami, N. Shiri, and F. Pesaran, "An efficient Imprecise 4: 2 Compressor Using Gate Diffusion Input Supplemented with Dynamic Threshold", *Journal of Southern Communication Engineering*, Vol. 13, No. 50, pp. 1-10, 2023.

[29] L. Sayadi, S. Timarchi, and A. Sheikh Akbari, "Two Efficient Approximate Unsigned Multipliers by Developing New", *IEEE Transactions on Circuits and Systems*, Vol. 70, No. 4, pp. 1649-1659, 2023.

[30] K. Pathak, A. Darji, and J. Sarvaiya, "Low power Dadda multiplier using approximate almost full adder and Majority logic based adder compressors", In: *Proc. of 2022 IEEE Region 10 Symposium (TENSYMP)*, pp. 1-6, 2022.

[31] M. Ha and S. Lee, "Multipliers with approximate 4-2 compressors and error recovery modules", *IEEE Embedded Systems Letters*, Vol. 10, No. 1, pp. 6-9, 2017.

[32] M. Ghanatabadi, B. Ebrahimi, and O. Akbari, "Accurate and compact approximate 4: 2 compressors with GDI structure", *Circuits, Systems, and Signal Processing*, Vol. 42, No. 7, pp. 4148-4169, 2023.

[33] V. Arulkarthick and A. Rathinaswamy, "Delay and area efficient approximate multiplier using reverse carry propagate full adder", *Microprocessors and Microsystems*, Vol. 74, p. 103009, 2020.

[34] P. Foroutan and K. Navi, "A Novel Current Mode Approximate Multiplier Scheme Based on 4: 2 and 5: 2 Compressors with Low Power Consumption and High Speed in CNTFET Technology", *Circuits, Systems, and Signal Processing*, Vol. 43, No. 5, pp. 3042-3072, 2024.

[35] 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.

[36] M. Taheri, A. Arasteh, S. Mohammadyan, A. Panahi, and K. Navi, "A novel majority based imprecise 4: 2 compressor with respect to the current and future VLSI industry",

Microprocessors and Microsystems, Vol. 73, p. 102962, 2020.

[37] E. Zacharelos, I. Nunziata, G. Saggese, A. Strollo, and E. Napoli, "Approximate recursive multipliers using low power building blocks", *IEEE Transactions on Emerging Topics in Computing*, Vol. 10, No. 3, pp. 1315-1330, 2022.

[38] Y. Zhang, X. Chen, C. He, and G. Xie, "Energy-efficient multipliers using imprecise compressors for image multiplication", *International Journal of Circuit Theory and Applications*, Vol. 50, No. 11, pp. 3875-3890, 2022.

[39] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity", *IEEE Transactions on Image Processing*, Vol. 13, No. 4, pp. 600-612, 2004.