Research Article

Error efficient LOB-based approximate multipliers for error-tolerant applications

E. Jagadeeswara Rao[1*] and P. Samundiswary[2]
Research Scholar, Department of Electronics Engineering, Pondicherry University, Kalapet, Puducherry, India[1]
Professor, Department of Electronics Engineering, Pondicherry University, Kalapet, Puducherry, India[2]

Abstract
Approximate computing (AC) was positioned at the forefront of research in the field of error-tolerant applications. One key facet of AC was the use of approximate arithmetic functions, which offered significant reductions in delay, area, and power consumption at the expense of accuracy. Among these arithmetic functions, multiplication was extensively employed and played a pivotal role in error-tolerance applications. However, as the bit width increased, the design metrics and accuracy of existing multiplication designs tended to reduce. In this paper, novel architectures for leading one-bit-based approximate multipliers (LOBAMs) were proposed, aimed at improving both accuracy and design metrics. This paper focused on 8 × 8 and 16 × 16 approximate multipliers (AMs) designed in 90 nm complementary metal oxide semiconductor technology. The simulation results confirmed that LOBAMs outperformed existing AMs, reducing mean relative error distance, mean error distance, worst-case of error, normalized error distance, and error distance by an average of 74.59%, 80.75%, 41.06%, 84.19%, and 72.3%, respectively. Furthermore, when the proposed LOBAMs were embedded into an image smoothing filter, they demonstrated superior performance in terms of peak signal-to-noise ratio and structural similarity index metric compared to prior AMs. Finally, the proposed LOBAMs were exhibited remarkable advancements in both accuracy and design metrics when compared to existing AMs. This work underscored the potential of LOBAMs to revolutionize AC and contribute to more efficient and accurate error-tolerant systems.

Keywords
Approximate computing, Error metrics, Quality metrics, Design metrics, LOB-based approach.

1.Introduction
The ever-increasing demand for higher computing performance while conserving energy resources has been a constant challenge for emerging applications. Approximate computing (AC) has emerged as a promising approach to address this challenge. AC has sought to replace complex, traditional, power-hungry data processing blocks with simpler, low-gate counts. These approximations have introduced inaccuracies into processed data, but, in return, they have offered substantial reductions in power consumption and chip area. This paradigm shift has had the potential to create more energy-efficient systems that are tailored to current and future application trends [1, 2].

The ubiquity of electronic devices in our daily lives has made digital data computation a transformative force in society.

Contemporary computing platforms have executed computations with precision tailored to the specific requirements of future applications. In recent years, these computing platforms' performance has witnessed exponential growth [3]. However, not all computations have been equally important from an application standpoint. Human perception has often been unable to detect inaccuracies in processing images or videos, especially in applications with inefficient algorithms and exact models. In such contexts, introducing minor computational errors into digital logic circuits has significantly reduced complexity while boosting performance, albeit at the cost of some accuracy [4].

The input data has inherently contained noise in many real-world scenarios, such as image and video processing, multimedia systems, and data processing for recognition and clustering. The methods engaged in these data processing tasks have often been probabilistic or statistical. Given the probabilistic

---
*Author for correspondence

nature and statistical characteristics of these computations, minor errors have typically had a negligible effect on performance. Consequently, AC has found practical application in scenarios where a certain degree of accuracy loss has been acceptable [4].

The demand for computing stages has been estimated to continue growing persistently, with workloads involving extensive data processing to achieve optimal results. AC has presented a promising approach to enhance energy efficiency in this context. By introducing controlled errors into computations, AC has been able to reduce the power consumption and complexity of digital circuits while maintaining acceptable accuracy levels [5].

The process of a multiplier has been divided into three main stages: partial product (PP) generation, PP addition, and final summation. The first two stages, in particular, have significantly impacted design metrics (DMs) compared to the remaining stages [6]. While existing approximate multipliers (AMs) have excelled in optimizing DMs, they have often struggled to improve error metrics (EMs), particularly with regard to the few partial products (PPs).

To address this challenge, researchers have introduced advanced AM methods, and a modified static approach has been applied to unsigned and floating-point AMs, which have aimed to design area-efficient AMs [7–9]. While these area-efficient AMs have delivered improved DMs, they have done so at the expense of output accuracy, rendering them unsuitable for error-tolerant applications.

Efforts have also been made to improve accuracy through rounding and truncation methods [10–13]. However, these approaches have tended to compromise DMs. Moreover, for larger-sized multipliers, these AMs have exhibited limited accuracy improvements. In response to these challenges, rounding methods have been employed to design AMs. These AMs have reduced errors for larger input operand sizes but at the cost of increased complexity [14–19].

Additionally, the truncation method has been used to design Booth, Wallace, and Dadda multipliers. Moreover, different logic reduction circuits based on 4:2 compressors have been used at the PP reduction stage of these multipliers [20–24]. Later, error-optimized different compressors have been used at

the PP reduction stage of Dadda, Booth, and Wallace multipliers [25–29]. In addition, truncation approaches with different error-efficient compressors have been suggested for PP reduction of Wallace and Dadda structures [30–34]. Furthermore, combinations of hybrid compressors and counters have been designed for PP reduction of these multipliers [35–39]. Later, recursive-based AMs have also been developed using smaller AMs to better balance accuracy and DMs [40–45]. However, the area of these higher-order AMs has increased significantly with the input bit size. As a result, while existing AMs have often offered superior DMs, they have not necessarily achieved improved EMs.

In light of these challenges, leading one-bit-based approximate multipliers (LOBAMs), specifically LOBAM0 and LOBAM1, are being proposed in this work to enhance both DMs and EMs. These LOBAMs select the first four n/2 bits from the two n-bit inputs and place them in the leading one-bit (LOB) position. The final result is being obtained through a combination of shifting and addition operations involving LOB values. In this paper, EMs are being assessed [46], including mean relative error distance (MRED), worst-case of error (WCE), mean error distance (MED), normalized error distance (NED), and error distance (ED), for both the proposed LOBAM0, LOBAM1, and state-of-the-art AMs. Furthermore, the image smoothing filter (ISF) [47] is being embedded with the proposed LOBAM0, LOBAM1, and existing AMs, employing standard images [48] to evaluate quality metrics (QMs) in terms of the structural similarity index metric (SSIM) and peak signal-to-noise ratio (PSNR) [49].

The major contributions of this work can be summarized as follows:
1. Introduction of two novel AMs, LOBAM0 and LOBAM1, based on the LOB approach, achieving better EMs and DMs.
2. Inclusion of gate-level designs for the proposed LOBAM multipliers to assess EMs and DMs.
3. The trade-off between energy efficiency and accuracy is demonstrated by the figure of merits (FoM), which are denoted as FoM1 and FoM2.
4. Design of ISF embedded with both existing and proposed AMs, facilitating the evaluation of QMs.

The subsequent sections of this work are ordered as follows: Section 2 reviews prior studies on AMs, section 3 explains the proposed AM algorithms, and section 4 outlines the simulation results and presents

an analysis of the corresponding results. Finally, section 5 offers the conclusion of this work.

2.Literature review

This section reviews a few of the research efforts in designing AMs, mainly focusing on the trade-offs between DMs and EMs. Static Segmentation-based AM involves truncating half of the feed operands and utilizing an unsigned multiplier [7]. However, truncation introduces errors, impacting the accuracy of the AM. A low-power and area-efficient floating-point AM based on the static segmentation method, which truncates input operands, leading to a reduction in area. However, the accuracy of the suggested AM depends on the segmentation output [8]. Next, a low-power AM is suggested based on multiplexer logic, and an effective correction technique is introduced, which truncates input operands, leading to improved accuracy compared to static segmentation-based AM [9]. Next, the truncation and rounding-based AM method combines truncation and rounding of PP bits to reduce the number of half and full adders during PP reduction. However, increasing the bit width leads to reduced accuracy [10]. Truncation-based AM aims to improve DMs through truncation but at the cost of accuracy, which depends on the truncation length [11].

Moreover, the combination of truncation and rounding that AM employs incorporates an add-and-shift logic. This method performs approximate multiplication, and accuracy is influenced by the rounded and truncated numbers [12]. Later, more error-efficient AMs are suggested using the leading one/zero bit approach [13]. However, the area of the suggested AMs is dependent on the size of the leading one/zero bit. Next, an AM with reciprocal error compensation is designed to strike a balance between energy consumption and accuracy. However, this AM may not offer better accuracy for higher-order AMs [14]. Subsequently, the dynamic range unbiased AM design sets the LSB of the truncated value to '1' to mitigate errors, with the precision and quality determined by the truncation length 'm' [15]. Hardware rounding-based AMs feed operands to the closest power of two, simplifying the hardware rounding process but introducing more significant errors as accuracy relies on the rounded input operand values [16]. Reconfigurable rounding-based AMs round feed operands to the nearest power of 2, with accuracy contingent on the chosen rounded values [17]. Furthermore, error-efficient AMs are suggested using rounding and rounding with the

Karatsuba algorithm [18, 19]. However, the accuracy of the suggested AMs depends on the rounding size. Later, compressors with minimum number of logic gates and majority-based compressors are used to design AMs for optimizing the EMs, power, and delay [20–23]. Nevertheless, the area of the suggested AMs is more.

Furthermore, low-power majority logic-based AM design improves accuracy but depends on the size of the compressors for its accuracy [24]. Next, majority-based and compressors-based AM provide better EMs. This AM may not exhibit satisfactory DMs for higher-order AMs [25]. Next, AM with reduced logic in compressors improves DMs but sacrifices accuracy, especially for higher-order AMs [26]. Next, the AM utilizes a modified booth multiplier with a truncation approach to enhance DMs, but experiences reduced accuracy with larger bit sizes [27]. Next, the error-efficient AMs entirely rely on the performance of suggested 4-2 compressors, with their accuracy and DMs depended on the compressor's size [28]. Next, AMs employing various PP reduction methods achieve better latency and accuracy compared to existing AMs [29]. However, accuracy remains a concern for high-input widths. Next, area and power-efficient AMs utilize low-power devices in compressors to enhance performance but have accuracy contingent on compressor size [30].

Furthermore, the AM designed with half and full adder to improve DMs may not excel in accuracy, especially for higher-order AMs [31]. Next, the suggested AM utilized an unbiased 4:2 compressor, generating negative and positive sign errors in balance to enhance accuracy. However, it may compromise accuracy for higher-order bit widths [32]. Next, suggested Hardware-Efficient AM hardware-efficient AMs depend on minimum error 4-2 compressors, with the final product's accuracy contingent on the proposed 4-2 compressor [33].

Also, compressor-based, recursive-type, and truncation-type AMs are currently under review. For small-sized AMs, 4-2 compressors and modified 2-bit AMs are considered sufficient. Some of the well-known 4-2 compressors and recursive-type AMs are under review, and their methods, pros, and cons are listed in *Table 1*. From *Table 1*, it is evident that the suggested AMs are better in terms of DMs and EMs. However, the accuracy of larger AMs depends on the approximation of the smaller ones.

Table 1 Literature review of recent works

S. No.	Authors	Year	Methods	Pros	Cros
1	Ejtahed and Timarchi [34]	2022	Approximate compressor	Improved area and power consumption values	Higher error rates for higher-order AMs
2	Minaeifar et al. [35]	2023	Error compensation techniques	Enhanced power delay product and energy consumption values	More suitable for low-order AM designs
3	Sayadi et al. [36]	2023	Approximate compressors with positive and negative approximations	Better area and power consumption values	Suitable for unsigned AM designs
4	Rahmani et al. [37]	2023	Truncating method with 4:2 inexact compressors	Improved MED and dynamic power consumption values	Suitable for unsigned AM designs
5	Esmaeili et al. [38]	2023	Imprecise full adder by the gate diffusion input strategy	Enhanced normalization of the mean error distance (NMED), delay, and dynamic power consumption values	Area efficiency is better for lower-order designs
6	Yongxia et al. [39]	2023	Low-power computing units	Positive errors from Booth encoder, negative errors from approximate Wallace tree structure	Suitable for unsigned AM designs
7	Perri et al. [40]	2022	AMs with Encoding logics	Improved energy consumption	Accuracy is dependent on AM size
8	Deepsita et al. [41]	2022	AMs with low-power compressors	Better PDP and MRED	Accuracy is dependent on compressor logic
9	Zacharelos et al. [42]	2022	Carry manipulation method	Provides a better error-performance trade-off	More significant area usage for higher-order AMs
10	Waris et al. [43]	2021	Double-sided error distribution	Improved area and delay	Accuracy is dependent on AM size
11	Sk [44]	2022	Compressors used at the column reduction stage	Better error rate and MED	More suitable for low-order AM designs
12	Karthikeyan and Sk [45]	2023	Compressors used at PP reduction stage	Improved MRED and MED	Suitable for unsigned AM designs

In the realm of AC, compressor-based, truncation, and recursive-type AMs have consistently delivered decent results in both DMs and EMs. The suggested compressor-based AMs exhibit higher accuracy; however, they come with a larger area requirement. On the other hand, truncation-based AMs outperform others in terms of DMs, but the suggested ones have a larger area. Meanwhile, recursive-type AMs excel in terms of EMs, but their area requirements increase with bit size. In conclusion, the literature survey reveals that existing AMs either offer better DMs or EMs. As a result, LOB-based AMs are proposed to maintain better DMs and EMs, and their details will be explained in the following section.

1282

3.Methods
3.1Proposed designs methodology
The proposed LOBAMs aim to reduce EMs by multiplying large-width input operands of multipliers using half-width input operand LOB-based multipliers. The small size of LOB-based multipliers provides better EMs than existing static and dynamic type AMs. To execute this, the significant $n/2$-bits represented by XH, XL, YH, and YL are chosen from n-bit input operands X and Y. These $n/2$-bits of XH, XL, YH, and YL are treated as the LOBs that make use of the leading block. Further, the multiplication is completed utilizing shift, addition, and difference

operations, and thereby, the result is achieved by shifting the multiplication output by n bits.

In this paper, two LOBAMs (LOBAM0, LOBAM1) are presented, and the final approximate product (X × Y) is calculated using three ($XH \times YH$, $XH \times YL$, and $XL \times YH$) and four ($XH \times YH$, $XH \times YL$, $XL \times YH$, and $XL \times YL$) PPs, respectively. The n-bit multiplier's mathematical expressions are listed below:

Let X represents the n-bit multiplicand, and Y represent the n-bit multiplier of the proposed AMs, and are calculated using Equations 1 and 2.

$$X = XL + 2^{\frac{n}{2}}XH \qquad (1)$$

where XH and XL are the $n/2$-bit MSBs and LSBs of X.

$$Y = YL + 2^{\frac{n}{2}}YH \qquad (2)$$

where YH and YL are the $n/2$-bit MSBs and LSBs of Y.

3.2 Proposed LOBAM0 architecture

The foremost aim of the suggested LOBAM0 is to amplify the accuracy in contrast to that of the existing AMs. *Figure 1* portrays the design of the proposed LOBAM0. It includes of four units: a. $n/2$-bit Extractor, b. $n/2$-bit LOB, c. Arithmetic Units, and d. Adders. First, the $n/2$-bit Extractor unit determines the XH, XL, YH, and YL. Thereby, the $n/2$-bit

Extractor unit outputs are fed to the $n/2$-bit LOB units [12]. The LOB position of each input operand calculates the following Equation 3.

$$A_{ld} = \left(\prod_{k=j+1}^{n-2} \overline{A(k)}\right) \bullet A(j) \text{ for } 0 \le j \le n-2 \quad (3)$$

where A can be either XH or XL or YH or YL.

The output values of the $n/2$-bit Extractor and LOB units are further given to arithmetic units (AU-1, AU-2, and AU-3). Each AU consists of Barrel Shifters, Adders, and Subtractors. Finally, the product output Z_0 received at the outputs of Arithmetic Units are applied to adders (A1 and A2) and is given by Equation 4.

$$
\begin{aligned}
Z_0 &\simeq (XH \times YH)2^n + (XH \times YL)2^{\frac{n}{2}} + (XL \\
&\quad \times YH)2^{\frac{n}{2}} \\
&\simeq ((XH_{ld} \times YH) + (XH \times YH_{ld}) - (XH_{ld} \\
&\quad \times YH_{ld}))2^n + ((XH_{ld} \times YL) \\
&\quad + (XH \times YL_{ld}) - (XH_{ld} \\
&\quad \times YL_{ld}))2^{\frac{n}{2}} \\
&+((XL_{ld} \times YH) + (XL \times YH_{ld}) - (XL_{ld} \times YH_{ld}))2^{\frac{n}{2}}
\end{aligned}
$$
$$(4)$$

where XL_{ld} = LOB position of XL and YL_{ld} = LOB position of YL, XH_{ld} = LOB position of XH and YH_{ld} = LOB position of YH.
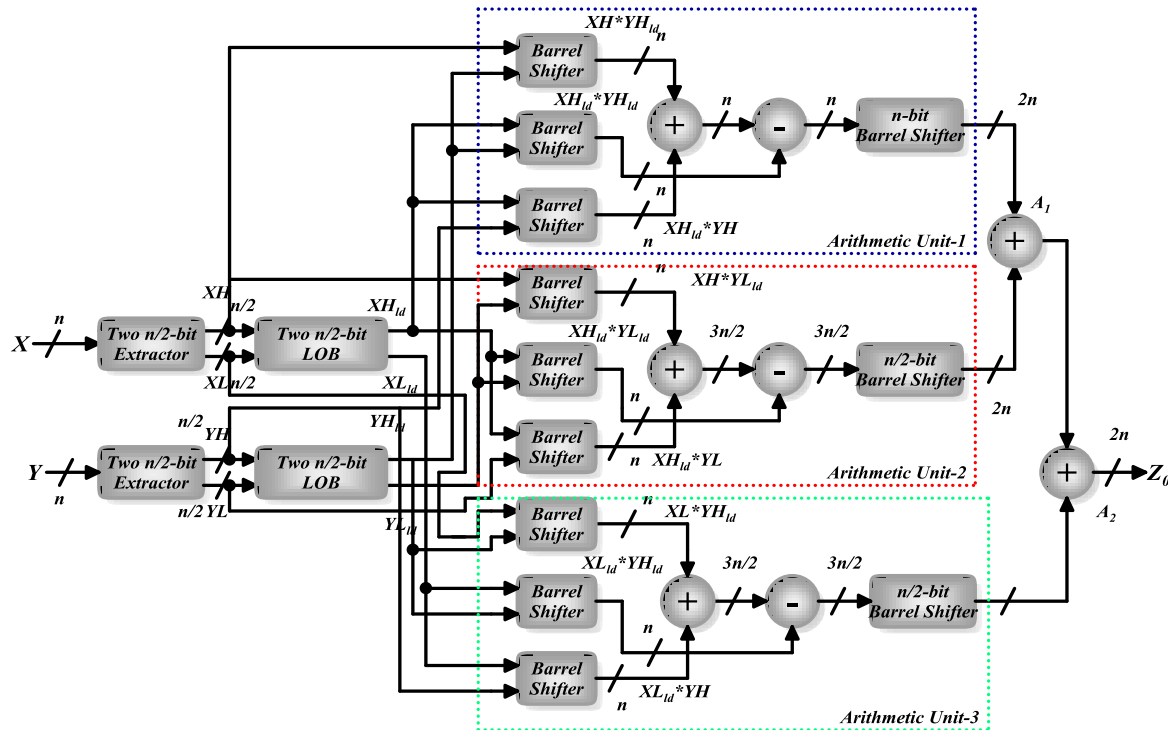

Figure 1 Block diagram of proposed LOBAM0

The proposed LOBAM0 design process is described in the form of four steps. The inputs X and Y are the n-bit multiplicand and multiplier of the proposed AM. The algorithm of the proposed LOBAM0 consists of four steps:

- The n/2-bit MSB and LSBs of X and Y are selected using an extractor unit.
- The n/2-bit LOBs (leading blocks) of XH, XL, YH, and YL are found using LOB units.
- The n/2-bit multiplication is achieved using AUs.
- The n-bit final multiplication product is generated using adders.

The novelty of the proposed LOBAM0 was the utilization of LOB units, marking a departure from other designs. The proposed design also offered improved DMs, as the LOB units occupied less area compared to other AC techniques.

3.3 Proposed LOBAM1 architecture

The proposed LOBAM1 architecture minimizes the EMs and improves the QMs. Moreover, it also raises a little circuit complexity compared to that of LOBAM0. The block diagram of the suggested AM is projected in *Figure 2*. It contains four units: a. *n/2*-bit Extractor, b. *n/2*-bit LOB, c. Arithmetic Units, and d. Adders. First, the *n/2*-bit Extractor unit determines the *XH, XL, YH*, and *YL*. Later, the *n/2*-bit Extractor unit's outputs are inserted into the *n/2*-bit LOB units [17]. The output values of the *n/2*-bit Extractor and LOB units are further applied to the Arithmetic Units, which consist of Barrel Shifters, Adders, and Subtractors. Lastly, on applying the outputs of Arithmetic Units to the adders (A1, A2, and A3), the product output Z_1 received and is given by:

$$Z_1 \simeq (XH \times YH)2^n + (XH \times YL)2^{\frac{n}{2}} + (XL \times YH)2^{\frac{n}{2}} + (XL \times YL)$$
$$\simeq ((XH_{ld} \times YH) + (XH \times YH_{ld}) - (XH_{ld} \times YH_{ld}))2^n + ((XH_{ld} \times YL) + (XH \times YL_{ld}) - (XH_{ld} \times YL_{ld}))2^{\frac{n}{2}}$$

$$+((XL_{rd} \times YH) + (XL \times YH_{rd}) - (XL_{rd} \times YH_{rd}))2^{\frac{n}{2}} + ((XL_{ld} \times YL) + (XL \times YL_{ld}) - (XL_{ld} \times YL_{ld}))$$

(5)

The design process of the proposed LOBAM1 is analogous to that of the proposed LOBAM0. But, the only difference is that the final multiplication product is obtained by considering the one *n/2*-bit $XL \times YL$ product. The proposed architecture can even choose both the greater-order, lower-order $\frac{n}{2}$-bits of *X* and *Y*. Immediately, once the divisions of lower-order and higher-order are nominated, the essential AU-1, AU-2, AU-3, AU-4 and adders are utilized to generate the final multiplication product. The novelty of the proposed design lay in the consideration of all four lower and higher units for finding the final multiplication product, which was achieved with LOB units and led to better EMs than existing AM designs.

3.4 Hardware units of proposed designs

The main hardware units of the proposed designs are the LOB unit, barrel shifter, and adder unit. The gate-level logic diagram of the LOB unit for 8-bit input operands is shown in *Figure 3*. The LOB unit consists of and, NOR and NOT gates. The LOB unit input is denoted as 8-bit A, and the output is denoted as 8-bit Y.

If the bit size of the AM is increased, the LOB unit only increases the number of logic gates. Therefore, the area occupation of the LOB unit is less compared to other AC techniques. A barrel shifter uses the last k-bit multiplication using the k/2-bit segment and LOB value. The gate-level diagram of the barrel shifter is completely explained in [50]. Finally, a Han-Carlson adder is used in the adder unit. It is a high-speed adder compared to other parallel prefix adders [51].
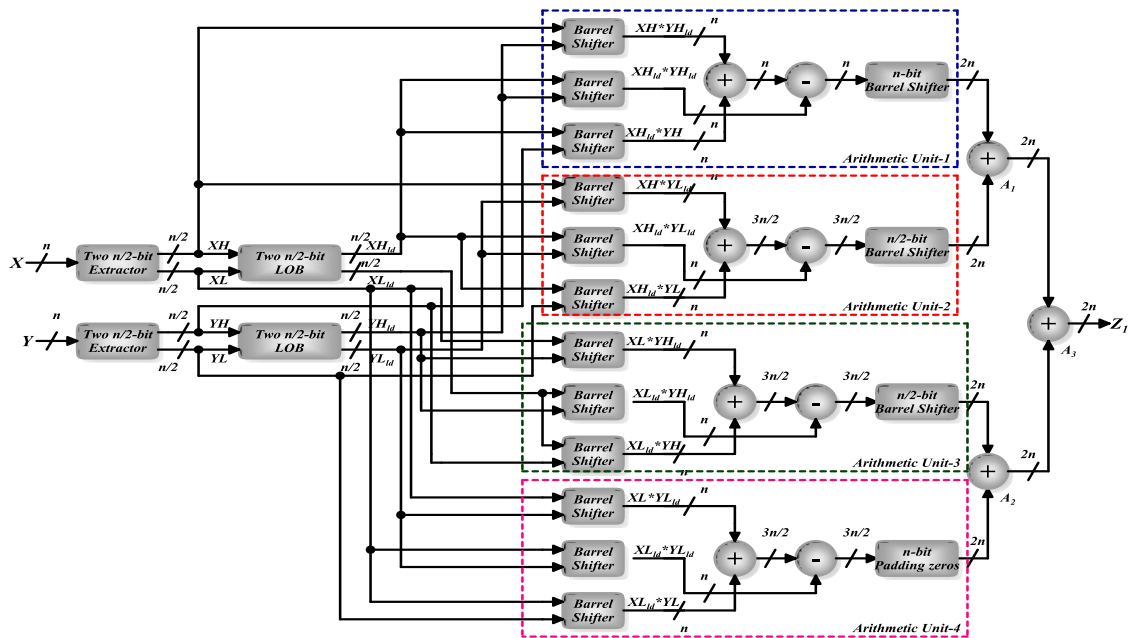
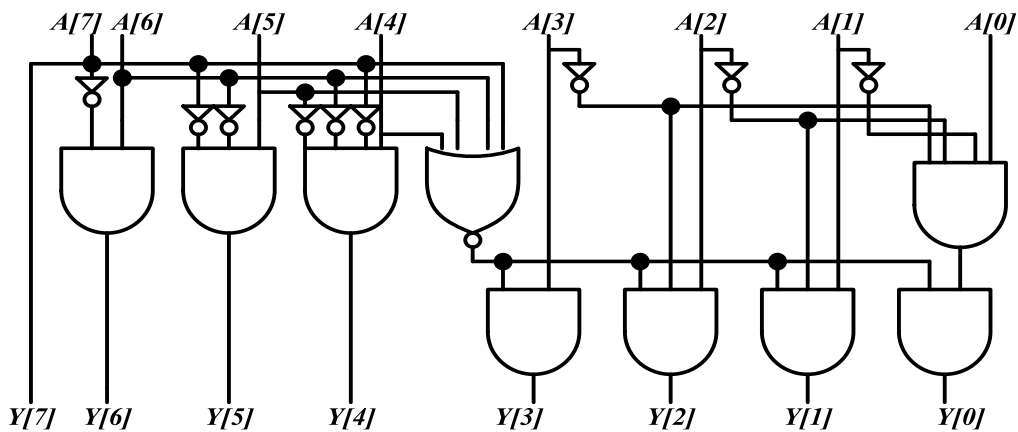**Figure 2** Block Diagram of Proposed LOBAM1

Figure 3 Gate-level LOB logic diagram for 8-bit input operands [12]

4.Results and discussion

All the proposed designs are simulated using Xilinx Vivado 2016.4. The simulation outputs of the 8-bit proposed designs are shown in *Figures 4* and *5*. The simulation outputs show that the proposed designs provide better multiplication products for higher order input values than the exact multiplication products.
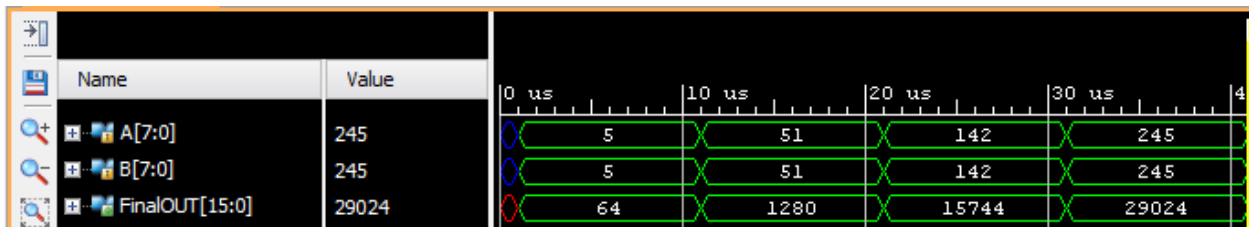

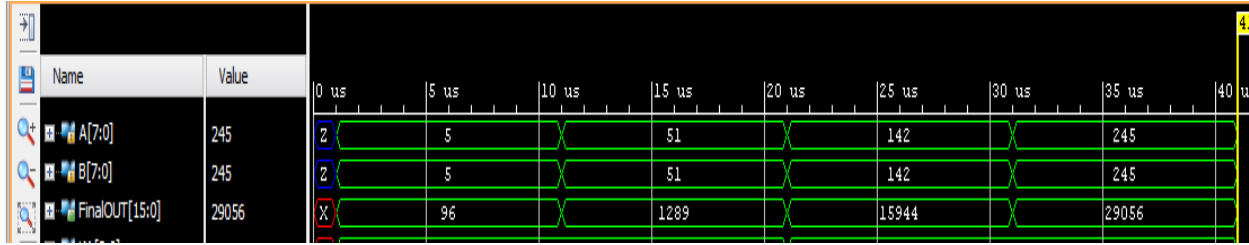**Figure 4** Simulation output of proposed LOBAM0

Figure 5 Simulation output of proposed LOBAM1

This section presents the performance analysis of the proposed LOBAM0 and LOBAM1 in terms of EMs such as MED, MRED, NED, WCE, and ED, as well as DMs such as area, power, delay, power-delay product (PDP), and energy-delay product (EDP). The EMs and DMs of the proposed LOBAM0 and LOBAM1 are also evaluated against the existing AMs. The features of n-bit existing AM designs in terms of rounding length (RL) are tabulated in *Table 2*.

Table 2 Existing AMs design features

AM Design	Features (RL)
[16]	n
[17]	n
[32]	---

The discussion section begins with an analysis of the EMs of the proposed and existing AMs. Next, the DMs of the proposed and existing AMs are clearly examined. Finally, a QM analysis is also performed in terms of PSNR and SSIM for the proposed LOBAM0, LOBAM1, and existing AMs in the ISF, demonstrating the applicability of the filter for image smoothing applications.

4.1Analysis of accuracy
The EMs are used to validate the proposed LOBAM0 and LOBAM1 accuracy. The EMs of the proposed multipliers are compared to the existing AMs. To derive the EMs, the AMs are simulated using 1 lakh random input patterns in Verilog code, and the EMs are computed in MATLAB. The following steps describe the simulation procedure for calculating the EMs:
- 1 lakh random input patterns are chosen.

- The random input patterns are converted into a .hex file and simulated using MATLAB code.
- The .hex file is applied to the Verilog test bench program for the proposed and existing AMs. The approximate output is converted into a .text file and simulated using Verilog code.
- The EMs (accuracy measures) are extracted and simulated using MATLAB code.

Several metrics are used to measure the error analysis of AMs, some of which are ED, WCE, NED, MRED, and MED [44], defined as Equation 6, Equation 7, Equation 8 and Equation 9.

$$ED = |C_{Exact} - C_{approximate}| \qquad (6)$$

where C_{exact}= output of exact multiplier and $C_{approximate}$= Output of AM.

$$MED = \frac{1}{2^{2n}}\sum_{k=0}^{2^{2n}}|ED_k| \qquad (7)$$

$$NMED = \frac{MED}{C_{max}} \qquad (8)$$

$$MRED = \frac{1}{2^{2n}}\sum_{k=0}^{2^{2n}}|RED_k| \qquad (9)$$

Where, $RED = \frac{ED}{C}$ and C = Exact output multiplier.

The maximum error of the AM of one million sample values is used as the WCE, which helps multiply more significant quantities.

These EMs are computed for the proposed and existing AMs and are presented in *Tables 3* and *4*. The results of the simulation show that the proposed multipliers achieve an average of 72.3% reduction in ED compared to the existing AMs. Moreover, it is also clear from *Table 2* that the proposed AMs achieve an average of 74.6%, 80.8%, 84.2%, and 41.1% improvement in MRED, MED, NED, and WCE, respectively, compared to the existing AMs.

Table 3 The EMs of various 8-bit AMs

Multiplier	ED	WCE	NED	MED	MRED
[16]	823636708	62158	0.2038	1.2666e+04	2.34×10^{-5}
[17]	921245872	63258	0.1524	29121	0.33×10^{-5}
[32]	458567992	16376	0.4315	1984.5	0.51×10^{-5}
LOBAM 0	302771238	22848	0.0240	1089.3	0.17×10^{-5}
LOBAM 1	290295950	22840	0.0224	987.2	0.16×10^{-5}

Table 4 The EMs of various 16-bit AMs

Multiplier	ED	WCE	NED	MED	MRED
[16]	1.332e+11	4.2949e+09	0.1184	5.0841e+05	0.37×10^{-5}
[17]	2.222e+13	4.3512e+09	0.8612	52345	0.39×10^{-5}
[32]	7.033e+14	9.8156e+04	1.0000	10812	1.55×10^{-5}
LOBAM 0	**411256351**	**38820**	**0.2982**	**1014.5**	**0.11×10^{-5}**
LOBAM 1	**403345621**	**36257**	**0.1436**	**824.3**	**0.10×10^{-5}**

4.2 Design metrics(DMs) analysis

The DMs of the proposed AMs are analyzed by coding them in Verilog and then simulating them using the Cadence register-transfer-level (RTL) Compiler. The 90nm complementary metal oxide semiconductor technology was used in the experimental setup, which had an operating frequency of 1GHz, a supply voltage of 1V, and a temperature of 27oC. The synthesis results of the 8-bit AMs are shown in *Table 5*. The proposed LOBAM0 and LOBAM1 AMs are the smallest PDP and EDP values compared to the existing AMs. Additionally, the proposed AMs have reduced area,

delay, and power by 49.7% to 17.4%, 35.7% to 8.5%, and 53.8% to 14.8%, respectively, compared to the existing AMs. These results demonstrate that the proposed AMs can significantly reduce DMs while maintaining good accuracy.

The synthesis results of the 16-bit AMs are shown in *Table 6*. The proposed LOBAM0 and LOBAM1 AMs are the smallest PDP and EDP values compared to the existing AMs. Additionally, the proposed AMs are reduced area, delay, and power by 75.7% to 47.6%, 34.3% to 7.9%, and 64.9% to 11.2%, respectively, compared to the existing AMs.

Table 5 Performance estimation of various 8-bit AMs

Multiplier	AREA (μm^2)	Delay (ns)	Power (mw)	PDP (fJ)	EDP(fj.ns)
[16]	1461	6.54	0.092	601.68	3934.99
[17]	2011	6.52	0.115	749.8	4888.70
[32]	1987	3.88	0.162	628.56	2438.81
LOBAM 0	**1208**	**4.21**	**0.075**	**315.75**	**1329.31**
LOBAM 1	**1699**	**5.01**	**0.098**	**490.98**	**2459.81**

Table 6 Performance estimation of various 16-bit AMs

Multiplier	Area (μm^2)	Delay (ns)	Power (mw)	PDP (fJ)	EDP (fj.ns)
[16]	6739	8.54	0.152	1298.08	11085.60
[17]	4106	9.22	0.314	2895.08	26692.64
[32]	8836	7.37	0.189	1392.93	10265.89
LOBAM 0	**1752**	**6.06**	**0.110**	**666.6**	**4039.60**
LOBAM 1	**2154**	**6.79**	**0.210**	**1425.9**	**9681.86**

4.3 Comprehensive analysis

Many AMs are less accurate and have power consumption similar to existing AMs (see *Tables 5* and *6*). Therefore, to achieve a balance between energy efficiency and accuracy in terms of NED and MRED, a balanced FoM such as FoM1 and FoM2 are determined based on Equations 10 and 11.

$FoM1 = NED \times PDP$ (10)
$FoM2 = MRED \times PDP$ (11)

The FoM1 metric is the ratio of accuracy to energy efficiency, while the FoM2 metric is the ratio of accuracy to power. The FoM1 metric of the 16×16 AMs is shown in *Figure 6*. The proposed AMs with the smallest FoM1 value offer a more efficient compromise between energy efficiency and accuracy

in terms of NED. The value of FoM1 is less through the proposed AMs, which is 19 times lesser than the average FoM1 value of the other AMs.

The FoM2 metric for each of the 16×16 AMs is shown in *Figure 7*. The proposed AMs with the smallest FoM2 values are offered a more efficient compromise between energy efficiency and accuracy in terms of MRED than that of the existing AMs. The proposed AMs achieve the smallest FoM2, which is 16 times lesser than the average FoM2 of the other AMs. It is noted that the proposed AMs achieve better accuracy than the other AMs while consuming less energy. The results demonstrate that the proposed AMs are used to improve the performance of error-tolerant applications such as ISF.
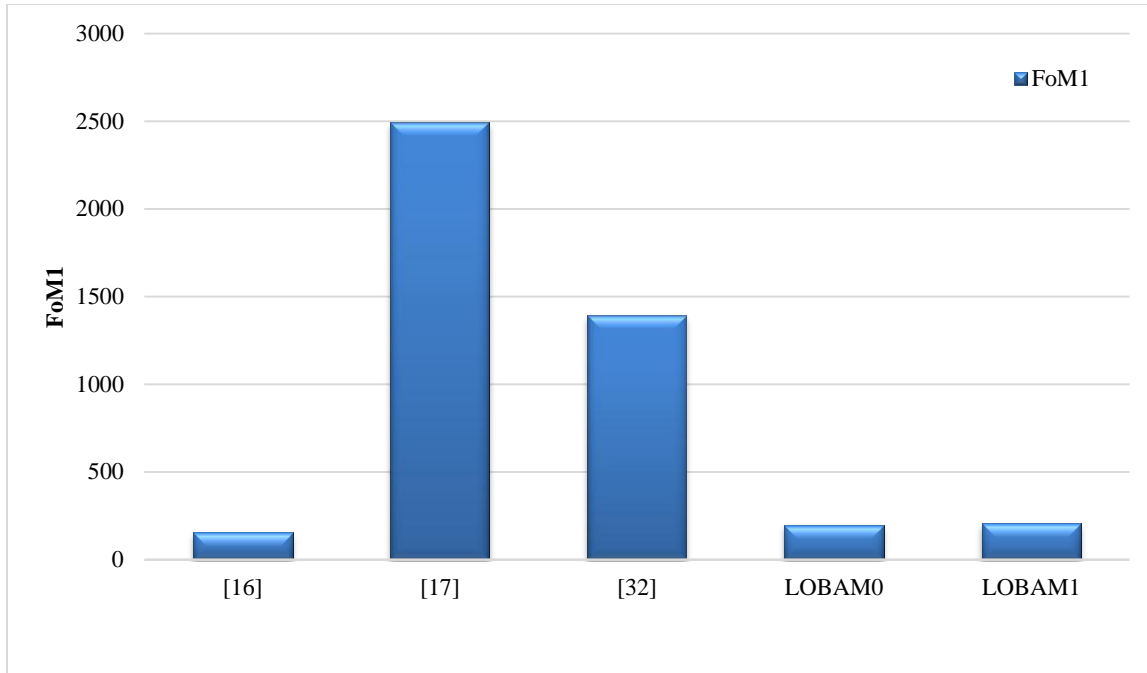
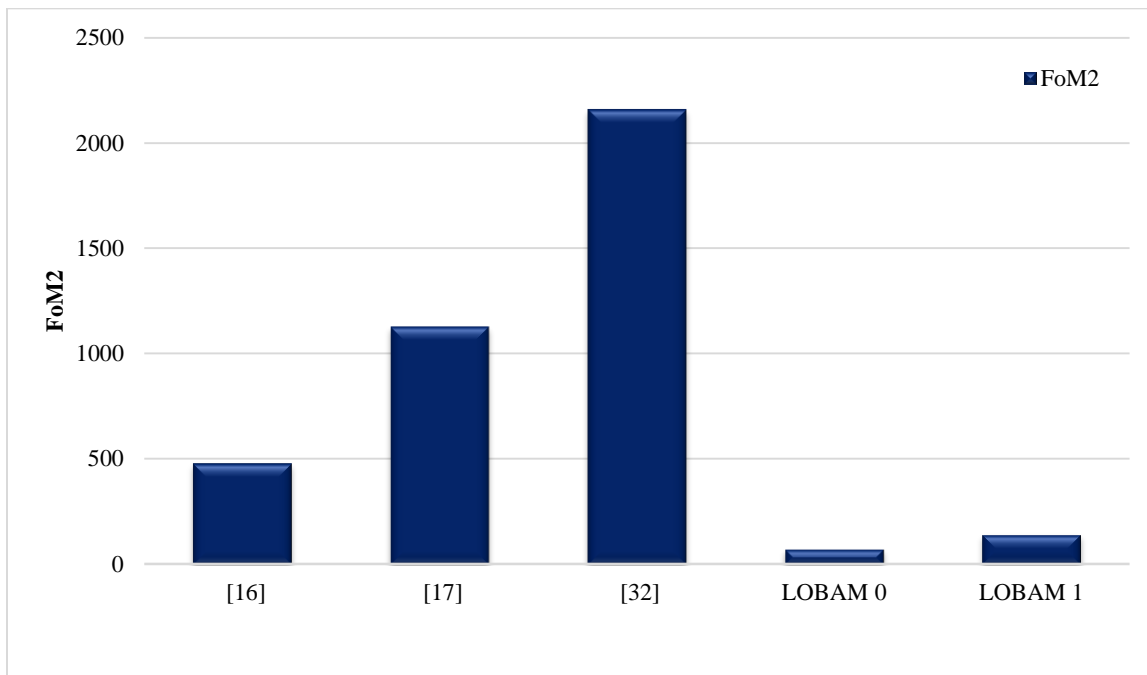**Figure 6** FoM1 factor for the considered 16×16 AMs

Figure 7 FoM2 factor for the considered 16×16 AMs

The following subsection discusses the analysis of the ISF with proposed LOBAM0 and LOBAM1 multipliers with respect to performance over the ISF with existing AMs in terms of QMs.

4.4 ISF quality metrics measures

In addition, the evaluation of the proposed AMs are incorporated into the ISF in terms of PSNR and SSIM using benchmark images. In ISF, the image sub-matrix is convolved with the standard mask to produce the finest pixel [47].

The convolution is performed using the proposed and existing AMs. The performance of the ISF embedded with the AMs is evaluated in terms of PSNR and SSIM. The proposed AMs are evaluated for their performance in the ISF application using six benchmark images with a pixel size of 256×256 [48]. The QMs of the ISF with the proposed AMs are compared to the QMs of the ISF with the existing AMs.

- The standard test image is selected.
- The standard test image is converted into a .hex file and simulated using MATLAB code.
- The .hex file is applied to the Verilog test bench program for the proposed and existing AMs. The approximate output is converted into a .text file and simulated using Verilog code.
- The QMs (quality measures) are extracted and simulated using MATLAB code.

The performance of ISF embedded with AMs is evaluated in terms of PSNR and SSIM [49]. They are defined in Equations 12 and 13.

$$PSNR = 20\,log_{10}\left(\frac{I_{max}}{\sqrt{MSE}}\right) \qquad (12)$$

where MSE- Mean Square Error.

$$SSIM(m,n) = \frac{(2\mu_a\mu_b+A_1)(2\sigma_{ab}+A_2)}{(\mu_a^2+\mu_b^2+A_1)(\sigma_a^2+\sigma_b^2+A_2)} \qquad (13)$$

where μ_a, and μ_b are the mean values.
where σ_a , σ_b and σ_{ab} are variances while A_1, and A_2 are the constants measured to keep the finite value of the metric.

Table 7 shows the QMs in terms of PSNR and SSIM. From the simulation outcomes, it is transparent that ISF with the recommended LOBAM0 and LOBAM1 achieves better PSNR and SSIM lying in the span of 3.7 %- 26.7 % and 11.8 % - 88.5 %, respectively, compared to the ISF embedded with existing AMs.

Table 7 PSNR (db) and SSIM values for ISF using various AMs

MultiplieRS	LENA		Cameraman		Girl		House		Lake		Couple		Average	
	PSNR (db)	SSIM	PSNR (db)	SSIM	PSNR (db)	SSIM	PSNR (db)	SSIM	PSNR (db)	SSIM	PSNR (db)	SSIM	PSNR (db)	SSIM
[16]	35.4	0.807	35.7	0.789	34.4	0.869	31.4	0.719	32.9	0.760	36.4	0.760	33.9	0.780
[17]	30.3	0.795	30.4	0.752	31.1	0.799	30.1	0.725	31.0	0.745	30.1	0.751	30.5	0.761
[32]	28.1	0.081	27.4	0.191	27.3	0.137	27.4	0.024	27.5	0.112	27.2	0.122	33.0	0.111
LOBAM 0	35.6	0.895	35.8	0.891	34.3	0.899	32.9	0.772	33.8	0.783	36.9	0.813	34.9	0.842
LOBAM 1	36.1	0.901	36.2	0.911	35.1	0.899	33.2	0.814	34.7	0.811	37.1	0.823	35.4	0.860

4.5Limitations

The proposed LOBAM0 and LOBAM1 have different strengths and weaknesses. LOBAM0 performs best in terms of DMs when the higher and lower order bits of the multiplicand and the higher order bits of the multiplier are used. However, LOBAM1 provides better EMs when the higher and lower order bits of both the multiplicand and the multiplier are used. A complete list of abbreviations is shown in *Appendix I.*

5.Conclusion and future work

In this work, the LOB-based AMs were proposed, which significantly reduction of EMs while maintaining DMs. LOBAMs utilized a LOB unit to extract the most significant n/2 bits of the input operands. These n/2 bits were then used to perform the multiplication, which was less error-prone than using the entire input operands. The proposed LOBAMs achieved an average reduction of 72.3% in ED, 74.6% in MRED, 80.8% in MED, 84.2% in
1289

NED, and 41.1% in WCE compared to existing AMs. They also attained an average reduction of 49.7% in area, 35.7% in delay, and 53.8% in power. The proposed LOBAMs were verified with the ISF, and the results demonstrated that the LOBAMs achieved better QMs than existing AMs. In the future, the proposed LOBAMs will be further improved by incorporating the reconfigurable approach. This will lead to even better DMs and EMs, enhancing the performance of error-resilient applications.

Author's contribution statement
E. Jagadeeswara Rao: Conceptualization, investigation, writing –original draft, analysis and interpretation, and study. **P. Samundiswary**: Editing, analysis and interpretation, study, and supervision.

E. Jagadeeswara Rao and P. Samundiswary

References

[1] Mittal S. A survey of techniques for approximate computing. ACM Computing Surveys. 2016; 48(4):1-33.

[2] Gupta V, Mohapatra D, Raghunathan A, Roy K. Low-power digital signal processing using approximate adders. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 2012; 32(1):124-37.

[3] Yang Z, Jain A, Liang J, Han J, Lombardi F. Approximate XOR/XNOR-based adders for inexact computing. In 13th international conference on nanotechnology 2013 (pp. 690-3). IEEE.

[4] Han J, Orshansky M. Approximate computing: an emerging paradigm for energy-efficient design. In 18th European test symposium 2013 (pp. 1-6). IEEE.

[5] Venkataramani S, Chakradhar ST, Roy K, Raghunathan A. Approximate computing and the quest for computing efficiency. In proceedings of the 52nd annual design automation conference 2015 (pp. 1-6).

[6] Reddy KM, Vasantha MH, Kumar YN, Dwivedi D. Design and analysis of multiplier using approximate 4-2 compressor. AEU-International Journal of Electronics and Communications. 2019; 107:89-97.

[7] Narayanamoorthy S, Moghaddam HA, Liu Z, Park T, Kim NS. Energy-efficient approximate multiplication for digital signal processing and classification applications. IEEE Transactions on Very Large Scale Integration Systems. 2014; 23(6):1180-4.

[8] Di MG, Saggese G, Strollo AG, De CD, Petra N. Approximate floating-point multiplier based on static segmentation. Electronics. 2022; 11(19):1-23.

[9] Strollo AG, Napoli E, De CD, Petra N, Saggese G, Di MG. Approximate multipliers using static segmentation: error analysis and improvements. IEEE Transactions on Circuits and Systems I: Regular Papers. 2022; 69(6):2449-62.

[10] Ko HJ, Hsiao SF. Design and application of faithfully rounded and truncated multipliers with combined deletion, reduction, truncation, and rounding. IEEE Transactions on Circuits and Systems II: Express Briefs. 2011; 58(5):304-8.

[11] Vahdat S, Kamal M, Afzali-kusha A, Pedram M. LETAM: a low energy truncation-based approximate multiplier. Computers & Electrical Engineering. 2017; 63:1-7.

[12] Vahdat S, Kamal M, Afzali-kusha A, Pedram M. TOSAM: an energy-efficient truncation-and rounding-based scalable approximate multiplier. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2019; 27(5):1161-73.

[13] Du Y, Chen Z, Cheng B, Shan W. Design and analysis of leading one/zero detector based approximate multipliers. Microelectronics Journal. 2023; 136:105783.

[14] Lingamneni A, Basu A, Enz C, Palem KV, Piguet C. Improving energy gains of inexact DSP hardware through reciprocative error compensation. In proceedings of the 50th annual design automation conference 2013 (pp. 1-8).

[15] Hashemi S, Bahar RI, Reda S. DRUM: a dynamic range unbiased multiplier for approximate applications. In IEEE/ACM international conference on computer-aided design 2015 (pp. 418-25). IEEE.

[16] Zendegani R, Kamal M, Bahadori M, Afzali-kusha A, Pedram M. RoBA multiplier: a rounding-based approximate multiplier for high-speed yet energy-efficient digital signal processing. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2016; 25(2):393-401.

[17] Garg B, Patel S. Reconfigurable rounding based approximate multiplier for energy efficient multimedia applications. Wireless Personal Communications. 2021; 118:919-31.

[18] Rao EJ, Samundiswary P. Error-efficient approximate multiplier design using rounding based approach for image smoothing application. Journal of Electronic Testing. 2021; 37:623-31.

[19] Rao EJ, Rao KT, Ramya KS, Ajaykumar D, Trinadh R. Efficient design of rounding-based approximate multiplier using modified Karatsuba algorithm. Journal of Electronic Testing. 2022; 38(5):567-74.

[20] Gorantla A, P D. Design of approximate compressors for multiplication. ACM Journal on Emerging Technologies in Computing Systems. 2017; 13(3):1-7.

[21] Angizi S, Jiang H, Demara RF, Han J, Fan D. Majority-based spin-CMOS primitives for approximate computing. IEEE Transactions on Nanotechnology. 2018; 17(4):795-806.

[22] Moaiyeri MH, Sabetzadeh F, Angizi S. An efficient majority-based compressor for approximate computing in the nano ERA. Microsystem Technologies. 2018; 24:1589-601.

[23] Shirinabadi FS, Reshadinezhad MR. A new twelve-transistor approximate 4: 2 compressor in CNTFET technology. International Journal of Electronics. 2019; 106(5):691-706.

[24] Liu W, Zhang T, Mclarnon E, O'neill M, Montuschi P, Lombardi F. Design and analysis of majority logic-based approximate adders and multipliers. IEEE Transactions on Emerging Topics in Computing. 2019; 9(3):1609-24.

[25] Anusha G, Deepa P. Design of approximate adders and multipliers for error tolerant image processing. Microprocessors and Microsystems. 2020; 72:102940.

[26] Zhu Y, Liu W, Yin P, Cao T, Han J, Lombardi F. Design, evaluation and application of approximate-truncated booth multipliers. IET Circuits, Devices & Systems. 2020; 14(8):1305-17.

[27] Strollo AG, Napoli E, De CD, Petra N, Di MG. Comparison and extension of approximate 4-2 compressors for low-power approximate multipliers. IEEE Transactions on Circuits and Systems I: Regular Papers. 2020; 67(9):3021-34.

[28] Yang Z, Li X, Yang J. Power efficient and high-accuracy approximate multiplier with error correction. Journal of Circuits, Systems and Computers. 2020; 29(15):2050241.

[29] Khaleqi QJM, Ahmadinejad M, Moaiyeri MH. Ultraefficient imprecise multipliers based on innovative 4: 2 approximate compressors. International Journal of Circuit Theory and Applications. 2021; 49(1):169-84.

[30] Sudharani B, Sreenivasulu G. Design of high speed approximate multipliers with inexact compressor adder. International Journal of Advanced Technology and Engineering Exploration. 2021; 8(80):887-902.

[31] Fang B, Liang H, Xu D, Yi M, Sheng Y, Jiang C, et al. Approximate multipliers based on a novel unbiased approximate 4-2 compressor. Integration. 2021; 81:17-24.

[32] Chandaka S, Narayanam B. Hardware efficient approximate multiplier architecture for image processing applications. Journal of Electronic Testing. 2022; 38(2):217-30.

[33] Kumar UA, Bharadwaj SV, Pattaje AB, Nambi S, Ahmed SE. CAAM: compressor based adaptive approximate multiplier for neural network applications. IEEE Embedded Systems Letters. 2022; 15(3):117-20.

[34] Ejtahed SA, Timarchi S. Efficient approximate multiplier based on a new 1-gate approximate compressor. Circuits, Systems, and Signal Processing. 2022:1-20.

[35] Minaeifar A, Abiri E, Hassanli K, Darabi A. A high-accuracy low-power approximate multipliers with new error compensation technique for DSP applications. Circuits, Systems, and Signal Processing. 2023:1-9.

[36] Sayadi L, Timarchi S, Sheikh-akbari A. Two efficient approximate unsigned multipliers by developing new configuration for approximate 4: 2 compressors. IEEE Transactions on Circuits and Systems I: Regular Papers. 2023; 70(4):1649-59.

[37] Rahmani M, Babaeinik M, Ghods V, Khalesi H. Designing of an 8×8 multiplier with new inexact 4: 2 compressors for image processing applications. Circuits, Systems, and Signal Processing. 2023:1-31.

[38] Esmaeili E, Pesaran F, Shiri N. A high-efficient imprecise discrete cosine transform block based on a novel full adder and wallace multiplier for bioimages compression. International Journal of Circuit Theory and Applications. 2023; 51(6):2942-65.

[39] Yongxia S, Huaguo L, Bao F, Cuiyun J, Zhengfeng H, Maoxiang Y, et al. Design of approximate booth multipliers based on error compensation. Integration. 2023; 90:183-9.

[40] Perri S, Spagnolo F, Frustaci F, Corsonello P. Designing energy-efficient approximate multipliers. Journal of Low Power Electronics and Applications. 2022; 12(4):1-17.

[41] Deepsita SS, Kumar DM, Mahammad NS. Energy efficient error resilient multiplier using low-power compressors. ACM Transactions on Design Automation of Electronic Systems. 2022; 27(3): 1-26.

[42] Zacharelos E, Nunziata I, Saggese G, Strollo AG, Napoli E. Approximate recursive multipliers using low power building blocks. IEEE Transactions on Emerging Topics in Computing. 2022; 10(3):1315-30.

[43] Waris H, Wang C, Xu C, Liu W. AxRMs: approximate recursive multipliers using high-performance building blocks. IEEE Transactions on Emerging Topics in Computing. 2021; 10(2):1229-35.

[44] Sk NM. Low power, high speed approximate multiplier for error resilient applications. Integration. 2022; 84:37-46.

[45] Karthikeyan T, Sk NM. Energy efficient multiply-accumulate unit using novel recursive multiplication for error-tolerant applications. Integration. 2023; 92:24-34.

[46] Liang J, Han J, Lombardi F. New metrics for the reliability of approximate and probabilistic adders. IEEE Transactions on Computers. 2012; 62(9):1760-71.

[47] Kavand N, Darjani A, Rai S, Kumar A. Design of energy-efficient RFET-based exact and approximate 4: 2 compressors and multipliers. IEEE Transactions on Circuits and Systems II: Express Briefs. 2023; 70(9): 3644-8.

[48] Garg B, Sharma GK. A quality-aware energy-scalable gaussian smoothing filter for image processing applications. Microprocessors and Microsystems. 2016; 45:1-9.

[49] Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing. 2004; 13(4):600-12.

[50] Kumar RK, Joe DA. Behavioral level simulation of vedic multiplier for ALU. Journal of Advanced Research in Dynamical and Control Systems. 2017; 9(16):1231-49.

[51] Han T, Carlson DA. Fast area-efficient VLSI adders. In IEEE 8th symposium on computer arithmetic 1987 (pp. 49-56). IEEE.

E. Jagadeeswara Rao received his B.Tech. degree and M.Tech degree in Electronics and Communication Engineering, as well as VLSI Design, from JNTU, Kakinada, and AU, Visakhapatnam, India, in 2010 and 2015, respectively. He is currently a research scholar at Pondicherry University in Pondicherry. His areas of research include Approximate Multipliers and the Efficient Design of Arithmetic Elements. He has seven years of teaching experience and one year of industrial experience. Furthermore, he has supervised and guided various projects at both the undergraduate and postgraduate levels.
Email: emandi.jagadeesh@gmail.com

Dr. P. Samundiswary obtained her B.Tech degree and M.Tech. Degree in Electronics and Communication Engineering from Pondicherry Engineering College affiliated to Pondicherry University, Pondicherry, India, in 1997 and 2003. She received her Ph.D. degree from Pondicherry

E. Jagadeeswara Rao and P. Samundiswary

Engineering College affiliated with Pondicherry University, Pondicherry, India, in 2011. She has nearly 25 Years of teaching experience. Presently, she is working as a Professor in the Department of Electronics Engineering, School of Engineering and Technology, Pondicherry University, India. She has published more than 150 papers in National and International Conference Proceedings and Journals. She has been one of the book's authors published by LAMBERT Academic Publishing, Germany. Also, she has co-authored ten book chapters published by INTECH and Springer Publishers. She has received the Best Teacher Award thrice in the Department of Electronics Engineering from Pondicherry University. She is a recipient of the Best Woman Researcher in Science and Technology Award obtained from JNTU Kakinada. She is a member of IEEE, IETE, and IACSIT. Her areas of interest are Wireless Communication and Networks, Computer Networks, Optical Communication, and VLSI Design.
Email: samundiswary_pdy@yahoo.com

Appendix I

S. No.	Abbreviation	Description
1	AC	Approximate Computing
2	AMs	Approximate Multipliers
3	DMs	Design Metrics
4	EMs	Error Metrics
5	EDP	Energy-Delay Product
6	ED	Error Distance
7	FoM	Figure of Merits
8	ISF	Image Smoothing Filter
9	LOBAM	Leading One-Bit-Based Approximate Multiplier
10	LOB	Leading One-Bit
11	MED	Mean Error Distance
12	MRED	Mean Relative Error Distance
13	NED	Normalized Error Distance
14	NMED	Normalization of the Mean Error Distance
15	PPs	Partial Products
16	PDP	Power-Delay Product
17	PSNR	Peak Signal-to-Noise Ratio
18	QMs	Quality Metrics
19	RL	Rounding Length
20	RTL	Register-Transfer-Level
21	SSIM	Structural Similarity Index Metric
22	WCE	Worst-Case of Error