IEEE TRANSACTIONS ON
EMERGING TOPICS
IN COMPUTING

Received 19 August 2020; revised 2 March 2021; accepted 8 April 2021.
Date of publication 13 April 2021; date of current version 7 June 2022.

Digital Object Identifier 10.1109/TETC.2021.3072666

Energy- and Quality-Efficient Approximate Multipliers for Neural Network and Image Processing Applications

MOHAMMAD AHMADINEJAD AND MOHAMMAD HOSSEIN MOAIYERI, (Senior Member, IEEE)

Faculty of Electrical Engineering, Shahid Beheshti University, Tehran 1983969411, Iran
CORRESPONDING AUTHOR: MOHAMMAD HOSSEIN MOAIYERI (h_moaiyeri@sbu.ac.ir)

ABSTRACT Approximate computing is a new trend that trades off computational accuracy for lower energy dissipation and design complexity in various applications, where high precision is not a critical need. In this paper, energy- and quality- efficient approximate multipliers based on new approximate compressors are proposed. We use NAND gates for generating the complemented partial products, which reduces the number of transistors. Furthermore, new approximate compressors with different accuracy and performance characteristics are designed. Accordingly, three hybrid approximate multipliers offering different trade-offs between accuracy and hardware efficiency are proposed. The proposed designs are simulated using HSPICE with the 7nm FinFET model as a modern technology. Furthermore, the efficacies of the approximate multipliers in the neural network and image processing applications are evaluated using MATLAB. According to the results, the proposed designs provide far better compromises between the quality and energy metrics in comparison with the previous designs and can be considered as efficient alternatives for the exact multipliers in neural network and image processing applications.

INDEX TERMS Approximate computing, approximate multiplier, compressor, energy efficiency, neural network, image processing

I. INTRODUCTION

Due to the widespread applications of portable electronic devices, and the increasing complexity of digital circuits, power dissipation, has become a crucial issue in low-power modern integrated circuits. Approximate computing has emerged as a practical solution to reduce the power consumption of digital circuits in applications, where accuracy is not a critical necessity [1]–[3]. Indeed, functional accuracy is a primary requisite in any digital integrated circuits, especially arithmetic blocks. However, various applications such as multimedia processing, data mining, computer vision, and pattern recognition can exploit approximate computing to reduce the energy consumption and complexity of digital circuits [2]. These applications can tolerate a certain amount of error while producing acceptable output results. For instance, more than a certain quality for an image or video is intangible for a human, because of the perceptual limitations. Furthermore, since the external input data to a digital system are usually noisy and discrete, there is already a limit in the accuracy in showing output data. Furthermore, machine learning-based applications are based on the iterative refinement that can mitigate or buffer trivial errors [3]. By using approximate computing, designers can reduce preciseness to the extent that the outputs are still usable for the intended purposes, while achieving considerable reductions in energy and cost [4], [5].

On the other hand, scaling down the transistor dimensions in complementary metal-oxide-semiconductor (CMOS) technology has resulted in some critical challenges, including reduced gate control and short channel effects [6], [7]. The FinFET device, with a 3D structure, offers excellent gate control, lower short channel effect, smaller subthreshold slope, and higher Ion/Ioff, making it a good substitute for the planar MOSFET. Besides, the intrinsic body of FinFET has eliminated the random dopant fluctuations as the main cause of threshold voltage variation in nanoscale transistors [8]. However, the self-heating effect is a challenging issue in FinFETs. This issue can be addressed by using low-power design methods such as approximate computing in higher design levels [9].

As one of the most common arithmetic blocks in many digital systems, a multiplier is a suitable candidate for approximate design. Multiplier is one of the most useful but power-hungry blocks, which is often located on the critical path of digital systems. Therefore, designing efficient approximate multipliers can considerably improve the energy-efficiency of digital systems [10].

Multiplication is typically made up of three steps: 1) an array of AND gates generating the partial products; 2) partial product reduction; 3) computing the final output using a carry ripple adder [11]. The AND gates contribute to a considerable number of transistors in a multiplier. Hence, using NAND gates for generating the partial products can reduce the hardware complexity of a multiplier [12]. Furthermore, the reduction step unfavorably increases power consumption and design complexity. Therefore, reducing the complexity of this step by approximate computing can have a significant effect on the performance and energy-efficiency of a multiplier [13]. Compressors are the computing modules required to reduce partial products. The simplest structure for a compressor is a full adder, which was used classically in multipliers. Higher-order compressors such as 4:2 and 5:2 compressors can be employed efficiently to improve the multiplier's performance [14]. Several approximate compressors with different accuracy and hardware characteristics, as well as some innovative configurations for the reduction stage, have already been proposed in the literature [2], [10], [13]–[23].

In this paper, first, we use NAND gates instead of AND gates to generate the complemented partial products efficiently. Furthermore, new approximate 5:2 and 4:1 compressors are proposed. Utilizing the proposed approximate 4:1 compressor as a new approach for designing energy- and quality-efficient approximate multipliers significantly reduces the complexity of the reduction stage and makes the final addition much simpler. This simplification leads to a considerable reduction in the total delay and power consumption of the approximate multipliers. By utilizing the combinations of the proposed compressors, three approximate multipliers offering different trade-offs between accuracy and hardware characteristics are proposed. The HSPICE tool and 7nm FinFET technology are used to evaluate the proposed approximate multipliers. Moreover, the efficacy of each multiplier is evaluated in the neural network and image processing applications. The results indicate that the first proposed multiplier using only the proposed 5:2 compressor offers excellent accuracy and quality metrics by sacrificing the circuit characteristic such as power consumption and delay to some extent. However, the second and third proposed multipliers, which also use the proposed approximate 4:1 compressors, lead to significant reductions in the power dissipation, delay, and transistor count compared to their previous counterparts. To make a fair and more illustrative judgment on the efficacy of the approximate multipliers, Pareto diagrams and figures of merit considering both accuracy and circuit characteristics are taken into consideration. According to the results, despite the different characteristics and accuracy metrics provided by the proposed multipliers, they achieve better trade-offs between accuracy and circuit characteristics compared to their counterparts.

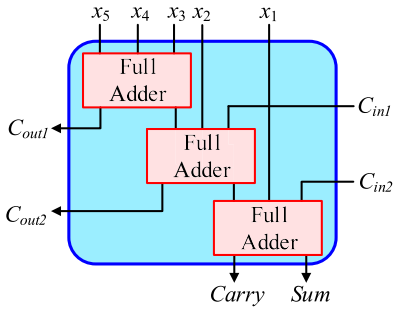


FIGURE 1. Schematic of the exact 5:2 compressor.

The rest of this paper is structured as follows: In Section II, the backgrounds of the study are briefly reviewed. Section III describes the proposed approximate compressors and multipliers. The simulation results and comparisons are presented in Section IV, and finally, Section V concludes the study.

II. BACKGROUNDS

A. RELATED WORKS

The partial product reduction stage uses many compressors with different orders and arrangements to reduce the partial products into two rows. As this stage contributes most to the power consumption, and transistor count in a multiplier, proposing efficient methods for partial product reduction based on compressors is of great importance. The schematic of the 5:2 compressor, as one of the widely used compressor structures, is illustrated in Figure 1. In this structure, x_1, x_2, x_3, x_4, and x_5 are the primary inputs, and *Sum* and *Carry* are the primary outputs. Moreover, the C_{in1} and C_{in2} carry inputs come from the preceding block of lower significance, and the C_{out1} and C_{out2} carry outputs go to the next block of higher significance.

Several exact compressors with different orders and structures have already been suggested in the literature [11], [24]–[26]. However, by emerging the approximate computing as a practical approach for reducing energy consumption, several approximate multipliers using different imprecise compressors have recently been suggested in the literature [2], [10], [13]–[23], [27]. Most of these approximate compressors ignore C_{in} and C_{out} signals to improve energy efficiency [2].

Two pioneering approximate 4:2 compressors and two types of conventional Dadda multiplier were proposed in [13]. The second multiplier that uses the approximate compressor, ignoring C_{in} and C_{out} signals, is more efficient than the first design and offers a better compromise between accuracy and performance parameters. 4:2 compressors capable of switching between the exact and approximate modes were proposed and used in the structure of the Dadda multiplier in [16]. The approximate 4:2 compressor suggested in [19] was designed by manipulating the truth table of the approximate

compressor presented in [20] and was used in the Dadda multiplier. In [14] and [15], approximate 4:2 and 5:2 compressors and modified Dadda multipliers based on these compressors were presented. In [2], a simple imprecise 4:2 compressor, using only one majority gate, was suggested. The *Sum* output of this design is equal to '1', and its *Carry* output is the majority of inputs, which significantly simplifies the structure of the approximate multiplier. In [18], the partial products of the multiplier were altered to change the probability terms. Two types of approximate multipliers based on the altered partial products were proposed in [18]. Two imprecise 4×4 multipliers were proposed and used to construct high-order multipliers in [17]. In [23], an approximate multiplier, using OR-based compressors for lower bit positions and approximate half adders and full adders for the higher bit positions, was proposed.

B. ACCURACY METRICS

This section discusses both application-independent and application-dependent metrics for evaluating the accuracy of approximate multipliers.

Two widely used application-independent metrics for quantifying the accuracy of the approximate multipliers are mean relative error distance (*MRED*) and normalized mean error distance (*NMED*) [28], [29]. The *MRED* parameter, which is the average value of all possible relative error distances, is expressed by

$$MRED = \frac{1}{2^{2N}} \sum_{i=1}^{2^{2N}} \frac{|ED_i|}{M_i} \tag{1}$$

where M_i is the exact output for each input operand, N is the bit length of the multiplier, and ED_i is the error distance between the exact and imprecise results for each input.

In the approximate applications that deal with the human senses, the relative error distance between the exact and approximate outputs is less important than the error distance. For instance, in an image, the difference between the exact and inexact pixels is more tangible for human eyes than their relative difference [2]. The *NMED* metric, which is defined based on the error distance parameter, is defined as

$$NMED = \frac{1}{(2^N - 1)^2} \sum_{i=1}^{2^{2N}} \frac{|ED_i|}{2^{2N}} \tag{2}$$

The peak signal to noise ratio (*PSNR*) and the mean structural similarity index (*MSSIM*) [30] are two application-dependent metrics for evaluating the quality of the output images of the approximate multipliers in comparison with the output images computed by the exact multipliers. The *PSNR* metric is defined as

$$PSNR = 10\log_{10}\left(\frac{m \times p \times {MAX_I}^2}{\sum_{i=0}^{m-1}\sum_{j=0}^{p-1}[I(i,j) - K(i,j)]^2}\right) \tag{3}$$

where, MAX_I is the maximum value of each pixel, m and p are the image dimensions, and $I(i, j)$ and $K(i, j)$ are the exact and approximate values for each pixel.

PSNR is a widely used metric for image quality assessment because of its clear physical meaning and simplicity of evaluation. Nevertheless, it is not based on the human visual system. Another metric for image quality evaluation is the mean structural similarity index metric (*MSSIM*), which computes the structural similarity of the exact and approximate images based on the point that the human visual system is able to extract information based on the image structure [30]. *MSSIM* is defined in Eq. (4). The explanations for these parameters were described in detail in [30].

$$MSSIM(X, Y) = \frac{1}{M} \sum_{j=1}^{M} \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{4}$$

III. PROPOSED APPROXIMATE DESIGNS

Recent efforts on designing approximate multipliers have focused on the arrangement of the partial product reduction stage and its compressors. A significant number of transistors in a multiplier belongs to the partial product generation stage, which is composed of an array of AND gates (54 AND gates in an 8-bit multiplier with four truncated rows). Accordingly, we use NAND gates to generate the complemented partial products, which leads to the elimination of 54 inverters. This will lead to fewer transistors and lower power dissipation, as the inverter gate has a high switching activity. However, this point should be considered for designing the other required modules in the next stages of the multiplier.

A. APPROXIMATE 5:2 COMPRESSOR

The proposed approximate 5:2 compressor ignores C_{in} and C_{out} signals, and its outputs are obtained as

$$S = \overline{(x_1 \oplus x_2)} \oplus \overline{(x_3 \oplus x_4)} \tag{5}$$

$$C = \overline{(x_1 + x_2) \cdot (x_3 + x_4) \cdot (\overline{x_1 \oplus x_2} + \overline{x_3 \oplus x_4})} \tag{6}$$

$$Sum = x_5 \oplus S \tag{7}$$

$$Carry = \overline{(x_5 + S)} + C \tag{8}$$

The truth table of the proposed approximate 5:2 compressor is shown in Table 1. To design this compressor, we place a greater emphasis on accuracy, and energy efficiency is considered as a secondary issue. Assuming the input bits to a multiplier are distributed uniformly in general (considering an equal probability for the input bits to be '0' or '1') [18]–[21], the probability that a partial product (an input bit to a compressor), generated by a 2-input NAND gate, equals to '0' ('1') is ¼ (¾). The probability of each input pattern is

TABLE 1. Truth table of proposed approximate 5:2 compressor.

x_5	x_4	x_3	x_2	x_1	Carry	Sum	Error distance
	(NANDs' outputs)						
0	0	0	0	0	1	1	-2
0	0	0	0	1	1	0	-2
0	0	0	1	0	1	0	-2
0	0	0	1	1	1	1	0
0	0	1	0	0	1	0	-2
0	0	1	0	1	1	1	0
0	0	1	1	0	1	1	0
0	0	1	1	1	1	0	0
0	1	0	0	0	1	0	-2
0	1	0	0	1	1	1	0
0	1	0	1	0	1	1	0
0	1	0	1	1	1	0	0
0	1	1	0	0	1	1	0
0	1	1	0	1	1	0	0
0	1	1	1	0	1	0	0
0	1	1	1	1	0	1	0
1	0	0	0	0	1	0	-2
1	0	0	0	1	1	1	0
1	0	0	1	0	1	1	0
1	0	0	1	1	1	0	0
1	0	1	0	0	1	1	0
1	0	1	0	1	1	0	0
1	0	1	1	0	1	0	0
1	0	1	1	1	0	1	0
1	1	0	0	0	1	1	0
1	1	0	0	1	1	0	0
1	1	0	1	0	1	0	0
1	1	0	1	1	0	1	0
1	1	1	0	0	1	0	0
1	1	1	0	1	0	1	0
1	1	1	1	0	0	1	0
1	1	1	1	1	0	0	0

calculated using the specific multiplication rule, as the input bits of the compressor (x_5-x_1) are independent of each other. For instance, the probability of occurrence of the input pattern "11111" is 243/1024 (the maximum possible value). Accordingly, to design an approximate compressor with less inaccuracy, the outputs corresponding to the inputs with a high probability of occurrence such as "11111", "11110", "11101", "11011", "10111", and "01111" should be avoided to be modified.

One of the most important accuracy parameters in designing approximate compressors is error distance (*ED*), which is the arithmetic distance between an erroneous output and its correct value [31]. It is worth pointing out that *ED* should be minimized, especially for the outputs corresponding to the input patterns with higher probabilities of occurrence [19].

According to Table 1, our proposed approximate 5:2 compressor has only six incorrect outputs out of 32 outputs (shaded in Table 1). Moreover, the sum of the probabilities of the input patterns with erroneous outputs (error rate) is only 1.6 percent (16/1024). Although the error distance corresponding to the false outputs is -2, it is negligible due to the very low probability of occurrence.

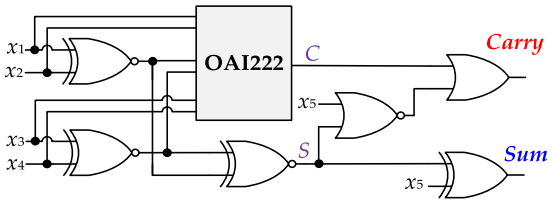


FIGURE 2. Schematic of the proposed approximate 5:2 compressor.

The structure of the proposed approximate 5:2 compressor is shown in Figure 2. The proposed approximate compressor includes three 6-transistor full-swing transmission gate-based XNOR gates [32], an XOR gate, a CMOS NOR gate, a CMOS OR gate, and a 12-transistor CMOS OAI222 (OR-AND-INV) compound gate. It is worth mentioning that increasing the number of cascaded transmission gate-based XNOR/XOR gates can reduce driving capability, which increases the propagation delay [33]. In order to minimize the propagation delay, in the multiplier structure, the XOR gate generating the *Sum* output is designed using an XNOR gate and a CMOS inverter gate. This design can be implemented at the transistor level using 44 transistors.

B. APPROXIMATE 4:1 COMPRESSOR

Unlike the proposed approximate 5:2 compressor, which considers accuracy as the primary concern, the proposed approximate 4:1 compressor is primarily designed to reduce transistor count, and energy consumption of the multiplier, and accuracy is taken into account as a secondary concern. Hence, in the proposed approximate 4:1 compressor the *Sum*, C_{in}, and C_{out} signals are ignored, and the *Carry* output is obtained as

$$Carry = \overline{x_1 \cdot x_2 \cdot (x_3 + x_4)} \tag{9}$$

Table 2 shows the truth table of the proposed 4:1 compressor. In this design, nine outputs are erroneous, which lead to an error rate of 47 percent. However, the output corresponding to the "1111" input pattern with the highest probability of occurrences is correct. Moreover, as mentioned earlier, the output error distances and their signs are also of importance, which can affect the accuracy of the multipliers. Error distances with opposite signs (such as -1 and +1) can neutralize each other's impact in the structure of an approximate multiplier [14]. In our design, the probabilities of the input patterns leading to the positive and negative error distances are 21 percent and 26 percent, respectively, which are very close to each other.

The proposed 4:1 compressor, shown in Figure 3, has a straightforward structure and is designed with only eight transistors based on the complementary style. Moreover, by using the 4:1 compressors in the second reduction stage (before the final addition), in the multiplier structure, the bit length of the final adder and the total delay of the multiplier are significantly reduced (see Figures 4c and 4d).

TABLE 2. The truth table of the proposed approximate 4:1 compressor.

x_4	x_3	x_2	x_1	Carry	Error distance
	(NANDs' outputs)				
0	0	0	0	1	-2
0	0	0	1	1	-1
0	0	1	0	1	-1
0	0	1	1	1	0
0	1	0	0	1	-1
0	1	0	1	1	0
0	1	1	0	1	0
0	1	1	1	0	-1
1	0	0	0	1	-1
1	0	0	1	1	0
1	0	1	0	1	0
1	0	1	1	0	-1
1	1	0	0	1	0
1	1	0	1	1	+1
1	1	1	0	1	+1
1	1	1	1	0	0

C. APPROXIMATE MULTIPLIERS

Figure 4 shows the schematic of the reduction circuitries of the proposed 8 × 8 approximate multipliers. Due to the widespread use of low-bit multipliers in machine learning and image processing applications, with no loss of generality, we concentrate on an 8-bit Dadda multiplier [34]. Furthermore, studies have shown that approximate compressors are more convenient for approximate unsigned multipliers [35]. Accordingly, in this work, we focus on unsigned approximate multipliers.

As a common method to trade-off accuracy for lower energy, the four least significant columns of the partial products are truncated [2]. In these designs, besides the proposed compressors, the conventional CMOS half adder and CMOS mirror full adder circuits [32] are also used to reduce the partial products and compute the product.

As illustrated in Figure 4, three combinations of the proposed approximate compressors are considered for designing the approximate multipliers. In the first case (see Figure 4b), the proposed approximate 5:2 compressor, offering higher accuracy, is used in the reduction circuitry. The second case (see Figure 4c) employs the proposed approximate 4:1 compressor, which drastically reduces the delay and power of the approximate multiplier while it still provides a reasonable accuracy. In the third case (see Figure 4d), the approximate 4:1 compressor is used in the eight least significant columns. The other seven most significant columns in the reduction circuitry use the proposed approximate 5:2 compressor. The objective of the third design is to provide a more effective compromise between the hardware and accuracy characteristics.

As mentioned before, in the proposed multipliers, NAND gates replace AND gates in the partial product generation stage. Accordingly, the partial products, which are the main inputs to the modules in the reduction stage, are inverted. Therefore, it is necessary to make some changes in the modules used in the reduction stage to provide regular inputs for the final adder.

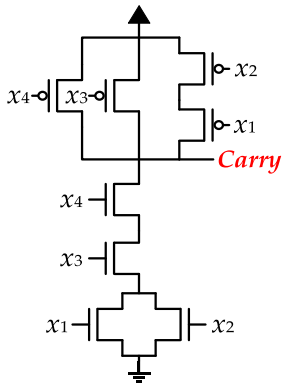


FIGURE 3. Schematic of the proposed approximate 4:1 compressor.

As shown in Figure 4, some of the partial products go straight to the second step. Accordingly, the outputs of the modules used in the first step, which are the inputs to the modules in the second step, should be complemented. It is well known that the complement of a logic function $f(a_1, a_2, \ldots, a_n)$ can be derived as the dual of $f(\overline{a_1}, \overline{a_2}, \ldots, \overline{a_n})$.

The inputs to the half and full adders used in the first step are complemented. Thus, the first reduction step uses dual forms of these modules and generates complemented outputs. As the proposed compressors are designed based on complemented inputs, an inverter gate is used to complement the output of each of the compressors used in the first step. As the second step provides the inputs of the final adder, the modules used in the second step should transform the complemented partial products to their regular form. As a result, the second step uses the proposed compressors and the dual of a half adder with an inverter gate at its output. Moreover, as the second and third proposed multipliers (Figs. 4 (c) and (d)) use the proposed 4:1 compressor in the first step, some partial product columns are reduced from 4 to 3 and 2 bits in the second step. Therefore, the structure of the approximate 4:1 compressors used in these columns can be simplified to 3:1 and 2:1 compressors. By considering unused inputs as '1' in Eq. (9), the *Carry* outputs for the 2:1 and 3:1 compressors can be obtained from Eqs. (10) and (11), respectively. Furthermore, the bit lengths of the final adder in the second and third multipliers are reduced to 2 and 7, respectively, which drastically reduces the total delay of the multiplier.

$$Carry\_Comp31 = \overline{x_1 \cdot (x_2 + x_3)} \tag{10}$$

$$Carry\_Comp21 = \overline{(x_1 + x_2)} \tag{11}$$

IV. SIMULATIONS AND COMPARISONS

In this section, the multipliers are simulated, evaluated, and compared comprehensively, considering various aspects of performance and accuracy.

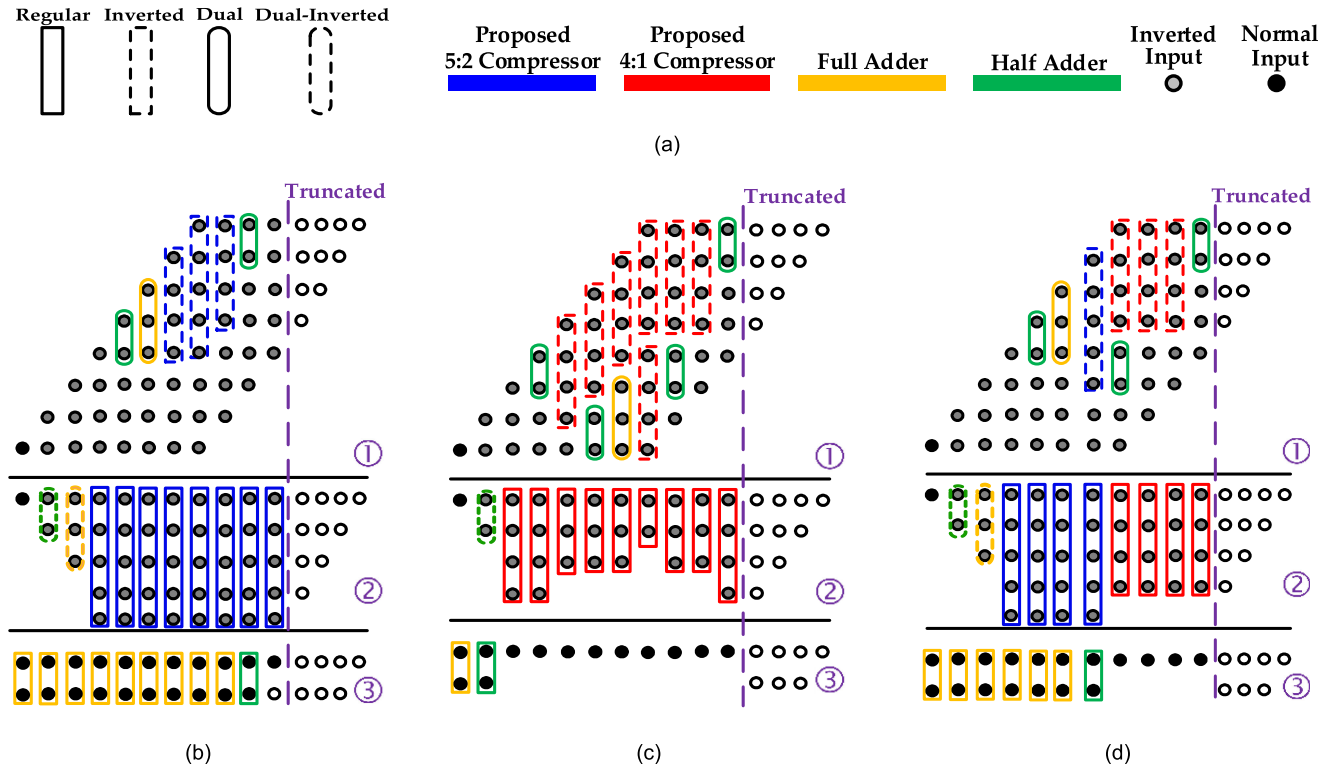


FIGURE 4. Partial product reduction circuitries of proposed approximate multipliers (a) Elements (b) First proposed design (PM1) (c) Second proposed design (PM2) (d) Third proposed design (PM3).

A. HARDWARE ANALYSIS

The exact 8-bit multiplier using an efficient exact compressor [24], and the previous and proposed approximate multipliers are simulated using HSPICE with the 7nm FinFET technology [36]. Table 3 gives the important parameters of the utilized model for the n-type and p-type FinFETs. The simulations are conducted at 0.7V supply voltage and 2GHz frequency.

Although the driving current of a FinFET can increase by increasing the number of parallel fins, considering the significant fin pitch overhead, it also considerably enlarges the cell area and increases the switching capacitance [2], [8]. Therefore, as the power consumption and hardware efficiency are two critical factors in approximate circuits, and given that the holes and electrons can have near mobilities in FinFETs, single-fin devices are used.

TABLE 3. The parameters of the 7nm FinFET model [36].

Parameters	n-type	p-type
Physical fin thickness (T_{fin})	6.5 nm	6.5 nm
Fin height (H_{fin})	32 nm	32 nm
Fin pitch (P_{fin})	27 nm	27 nm
Gate length (L)	21 nm	21 nm
Equivalent oxide thickness (EOT)	1.25 nm	1.25 nm
Body doping	10^{16} cm^{-3}	1016 cm^{-3}
Source/Drain doping	2×10^{20} cm^{-3}	2×10^{20} cm^{-3}
Low field mobility ($\mu 0$)	240 cm^2/V.s	200 cm^2/V.s
Gate Work Function (Φ_M)	4.372 eV	4.812 eV

It is notable that for a fair comparison, we have designed all of the compressors and the other components used in the previous multipliers efficiently at the transistor level according to the details given in the corresponding references using 7nm FinFET technology. Moreover, all of the previous multipliers have been redesigned efficiently based on NAND partial products. Furthermore, the four least significant columns of all approximate multipliers have been truncated. For more clarity, the descriptions of the multipliers under investigation are given in Table 4.

Table 5 gives the simulation results. The worst-case critical path delay of each multiplier is reported as the delay. A long stream of random input combinations, considering switching activity, have been generated and used as stimuli to estimate the power consumption of the multipliers. The power-delay product (*PDP*) and energy-delay product (*EDP*) are reported to assess the energy-efficiency of the multipliers. Moreover, the area of each multiplier is estimated as the total area occupied by its transistors based on the FinFET geometries described in detail in [37] (λ = 7.5nm) and Table 3.

According to the results, the second proposed multiplier (PM2), using the proposed approximate 4:1 compressor, drastically improves the performance parameters, compared to the other multipliers given in Table 5. This is due to use of the proposed 4:1 compressor, which has a straightforward structure that simplifies the reduction stage and reduces the bit length of the final adder. The design with the closest

TABLE 4. The Descriptions for the multipliers under investigation.

Dadda Multiplier	Description
M1	Multiplier based on the compressor of [2]
M2	Multiplier based on the second compressor of [13]
M3	Mul_2 multiplier proposed in [14]
M4	Multiplier based on the third 4:2 compressor of [15]
M5	Multiplier based on the DQ4:2C4 of [16]
M6	Multiplier based on the compressor of [17]
M7	M8_1 multiplier proposed in [17]
M8	M8_2 multiplier proposed in [17]
M9	Multiplier based on the compressor of [18]
M10	Multiplier1 proposed in [18]
M11	Multiplier based on the compressor of [19]
M12	2StepTrunc multiplier proposed in [21]
M13	Modified Dadda multiplier proposed in [22]
M14	Multiplier based on the compressor of [27]
M15	Multiplier proposed in [23]
M16	Multiplier based on the exact compressor of [24]

TABLE 6. Accuracy metrics of the approximate multipliers.

Multipliers	NMED	MRED
PM1	0.003	0.021
PM2	0.045	0.329
PM3	0.004	0.044
M1	0.055	4.171
M2	0.054	4.227
M3	0.018	0.082
M4	0.014	0.070
M5	0.020	0.080
M6	0.021	0.094
M7	0.017	0.065
M8	0.028	0.088
M9	0.019	0.078
M10	0.024	0.083
M11	0.011	0.046
M12	0.012	0.048
M13	0.008	0.079
M14	0.003	0.011
M15	0.002	0.030

performance parameters to the proposed design is M1. However, PM2 outperforms this multiplier in terms of the accuracy metrics (see Table 6). It is worth mentioning that the second proposed circuit improves the delay, power, *PDP*, *EDP*, transistor count, and area, on average by 79 percent, 64 percent, 92 percent, 98, percent56 percent, and 56 percent, respectively, as compared to the previous approximate designs listed in Table 5. The third proposed approach (PM3), which is among the four most accurate designs, achieves the third position considering all of the performance parameters. Although the first proposed design (PM1) is not ranked among the top designs regarding the performance parameters, it is between the three best accurate designs, as shown in Table 6.

Robustness to process variations is one of the most critical challenges in deep nanoscale circuits. Performance variations raised by the inevitable process variations can degrade the efficacy and robustness of the integrated circuits. Accordingly, we have conducted Monte Carlo simulations considering Gaussian distribution and variations at the 3σ level to assess the impacts of the process variations. The variation for fin thickness (T_{fin}), fin height (H_{fin}), fin pitch (P_{fin}), gate length (L), and equivalent oxide thickness (EOT) as the critical process parameters of FinFETs is considered as 15 percent [10]. To have a more illustrative assessment of the

TABLE 5. Comparison of the multipliers.

Multipliers	Delay (ps)	Power (μW)	PDP (fJ)	EDP ($\times 10^{-28}$ Js)	Transistors	Area (μm^2)
PM1	229.6	10.95	2.51	5769	1068	8.89
PM2	47.2	3.61	0.17	80	484	4.03
PM3	178.4	6.46	1.15	2056	718	5.98
M1	135.4	4.91	0.66	900	642	5.34
M2	218.9	8.77	1.92	4204	978	8.14
M3	219.1	6.94	1.52	3331	848	7.06
M4	233.0	11.40	2.66	6187	1096	9.12
M5	220.0	9.34	2.05	4519	978	8.14
M6	223.0	7.6	1.68	3756	864	7.19
M7	237.9	13.50	3.21	7641	1248	10.39
M8	237.9	12.07	2.87	6831	1206	10.04
M9	229.2	10.46	2.40	5496	1156	9.62
M10	233.8	10.02	2.34	5476	1112	9.26
M11	235.7	11.38	2.68	6322	1042	8.67
M12	234.8	10.74	2.52	5920	1036	8.62
M13	246.2	11.82	2.91	7167	1140	9.49
M14	245.1	12.80	3.14	7687	1174	9.77
M15	243.2	12.97	3.15	7671	1286	10.71
M16	252.8	18.79	4.75	12008	1530	12.74

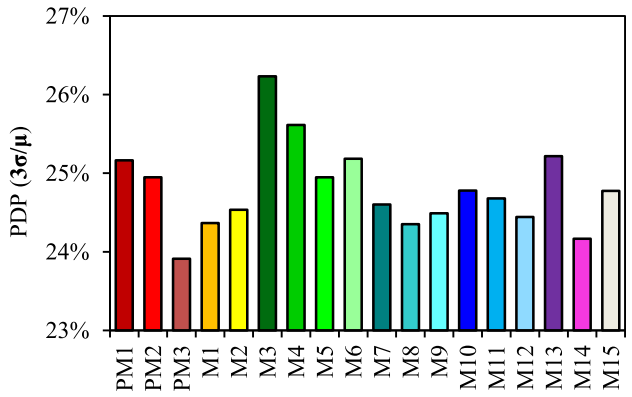


FIGURE 5. **PDP variations of the approximate multipliers.**

robustness of different multipliers, Figure 5 shows the percentages of the *PDP* variations in terms of $3\sigma/\mu$. According to the results, the proposed designs function robustly in the presence of process variations.

B. ACCURACY METRICS

In this section, to assess the accuracy of the approximate multipliers, the *NMED* and *MRED* metrics of the approximate multipliers have been calculated using MATLAB simulations by applying all of the 65536 possible input combinations. Table 6 shows the accuracy parameters. The M15, PM1, M14, and PM3 multipliers are the most accurate designs considering the *NMED* metric. In terms of the *MRED* metric, M14, PM1, M15, and PM3 take first to fourth place, respectively.

The *NMED* and *MRED* values for the first proposed approach are, on average, 83 percent and 75 percent less than the other multipliers (except M14 and M15). These improvements are, in turn, 77 percent and 47 percent for the third proposed multiplier. Although after M1 and M2, the second proposed multiplier using only the proposed 4:1 compressor has the lowest *NMED* and *MRED* metrics, it leads to an acceptable quality in neural network and image processing applications, as will be discussed in Sections IV-C and IV-D.

Figure 6 shows a comprehensive comparison of the multipliers considering the accuracy metrics (*NMED* and *MRED*), and *PDP* in Pareto diagrams. The purpose of the Pareto diagrams is to identify the most efficient designs in terms of energy–error. It is worth mentioning that to have a more legible diagram, the M1 and M2 multipliers showing the MRED values greater than four are not considered in Figure 6b.

The multipliers near the bottom-left corner of the Pareto diagrams are better designs, which have smaller *PDP*s and higher accuracies. The third proposed multiplier shows far better results in both Pareto diagrams as compared to the other multipliers. This is due to its efficient and accurate structure obtained using the combination of both proposed 5:2 and 4:1 compressors in the multiplier structure.

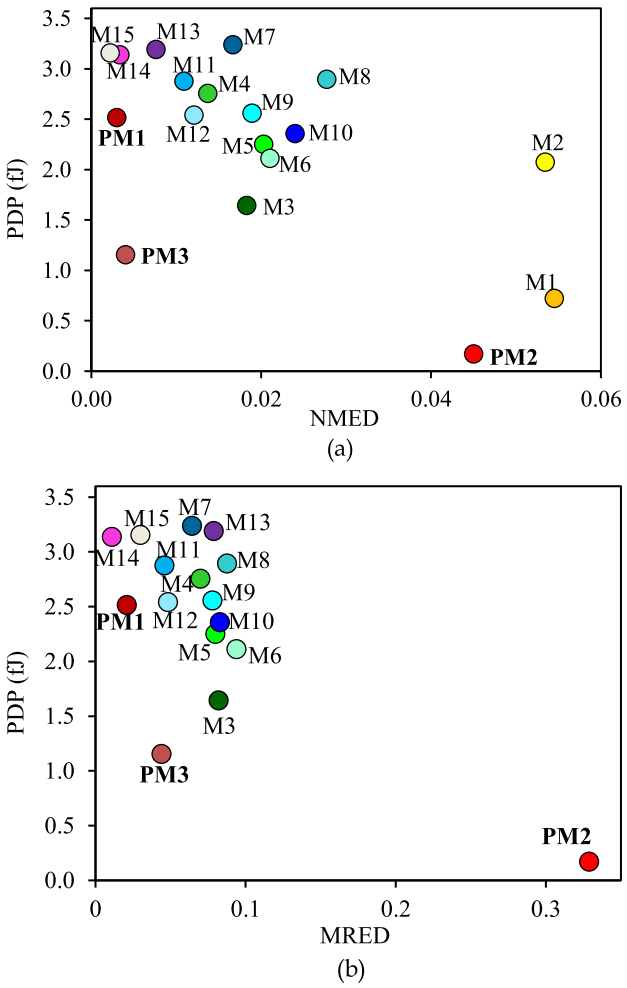


FIGURE 6. **Pareto diagrams for the approximate multipliers (a) PDP vs. NMED (b) PDP vs. MRED.**

C. NEURAL NETWORK APPLICATIONS

In this section, the efficacy of each of the approximate multipliers in the neural network (NN) applications is evaluated. Neural networks are prominent solutions for many machine learning tasks such as image classification [38]. Given that the neural networks are intrinsically error-tolerant and exploit many power-hungry blocks such as multipliers, they are suitable candidates for employing approximate computing [39].

Two types of neural networks, a multilayer perceptron (MLP) and a convolutional neural network (CNN), are considered to assess the performance of the approximate multipliers in classifications of the MNIST (Mixed National Institute of Standards and Technology) and SVHN (Street View House Numbers) datasets, respectively.

MNIST is a dataset of handwritten numbers containing 60,000 images for training and 10,000 images for testing. [40]. We use an MLP network to classify this dataset. This NN consists of 784 input neurons (each input image contains 28×28 pixels), 50 neurons in the hidden layer, and 10 output neurons. The outputs are considered as the probability of each of the classifications into ten classes (digits 0 to 9) [40].

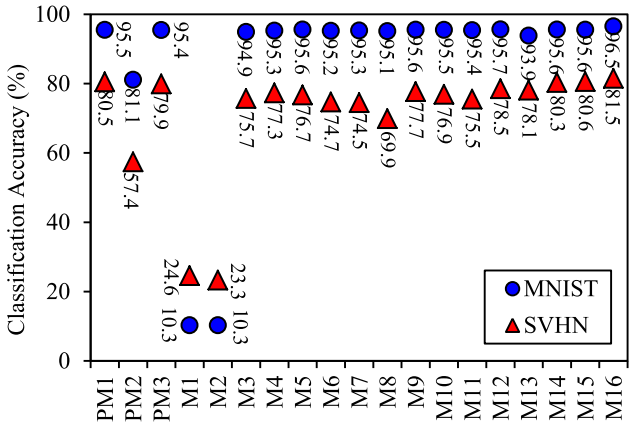


FIGURE 7. MNIST and SVHN classification accuracies using the multipliers under investigation.

This MLP uses the ReLU activation function which is a simple max(0,x) function.

SVHN is a real-world image dataset obtained from house numbers in Google Street View images. It consists of 73 257 training images and 26 032 test images. A pair of a 32 × 32 RGB image and its label represents each digit in this dataset [41]. To classify this dataset, we use the LeNet-5 network [42]. This architecture is made up of seven layers. The layer composition consists of three convolutional layers, two pooling layers, and two fully connected layers. We also use the ReLU activation function in this case. To reduce the complexity, which is one of the key parameters in approximate computing, we converted the original 32 × 32 RGB images to 32 × 32 grayscale images using the "Luma" mapping [43].

MATLAB programs have been developed to implement these two neural networks. In the training process, the exact multipliers have been used to calculate the neuron weights, and the approximate multipliers have been employed for the classification process. It is worth mentioning that in the training process, both negative and positive neuron weights are produced. Therefore, we need signed approximate multipliers to perform the multiplication. Accordingly, after the training process, we converted the neuron weights in sign-magnitude representation, and then the multiplications were performed using the discussed approximate multipliers. However, as we just convert the neuron weights one time after the training process, it does not lead to power and delay overheads in the classification process, in which we use approximate designs. Although if the outputs of the neural network should be converted to 2'complemnt, its energy overhead is also negligible as compared to the whole neural network (in the neural networks under study, there are only ten 8-bit outputs). The classification accuracy for each of the multipliers under investigation is shown in Figure 7.

For the MNIST dataset, the 8-bit exact multiplier (M16) shows an accuracy of 96.5 percent and the approximate multipliers show accuracies between 93.9 percent and 95.7 percent (up to 2.6 percent less accurate than the exact design), except M1, M2, and PM2. The accuracy of the M1 and M2 designs is 10.3 percent making them useless for this application. However, PM2 shows an accuracy of 81.1 percent, which is acceptable considering its significant hardware efficiency.

For the SVHN dataset, the accuracy of the exact 8-bit multiplier is 81.5 percent. The best approximate designs in this classification are M15, PM1, M14, and PM3, with accuracies of 80.6 percent, 80.5 percent, 80.3, and 79.9 percent, respectively (almost 1 percent less accurate than the exact design). M1 and M2 are the worst designs in this classification with accuracies of 24.6 percent and 24.3 percent. PM2 showing an accuracy of 57.4 percent is not among the best designs. However, its classification performance is acceptable in this case considering its significant hardware efficiency.

The results conclude that the approximate multipliers, especially our first and third proposed design, can be considered as efficient alternatives for the exact multiplier in neural network applications.

TABLE 7. The Average PSNR and MSSIM for the processed images in the image multiplication and image sharpening applications.

Multipliers	Image multiplication		Image sharpening	
	PSNR	MSSIM	PSNR	MSSIM
PM1	42.82	0.996	49.36	0.998
PM2	25.15	0.877	24.72	0.945
PM3	41.71	0.994	43.58	0.998
M1	24.37	0.831	13.08	0.642
M2	24.52	0.845	12.88	0.616
M3	30.54	0.969	36.60	0.992
M4	33.87	0.983	31.04	0.990
M5	29.35	0.970	27.17	0.985
M6	29.22	0.959	31.13	0.985
M7	31.71	0.974	32.09	0.986
M8	28.78	0.964	29.22	0.986
M9	29.43	0.970	27.40	0.986
M10	28.92	0.957	30.18	0.990
M11	36.08	0.994	35.54	0.994
M12	36.66	0.993	36.05	0.995
M13	36.91	0.991	32.47	0.985
M14	47.70	0.999	45.40	0.998
M15	48.85	0.999	49.20	0.998

D. IMAGE PROCESSING APPLICATIONS

Image multiplication and image sharpening, as two widely used algorithms in image processing, are considered to assess the practicality of the approximate multipliers in real applications. A program developed using MATLAB for each application. For the multiplication algorithm, the output image is computed by multiplying two images pixel by pixel. The equations utilized for the sharpening algorithm were explained in [16].

The *PSNR* and *MSSIM* metrics are calculated to evaluate the quality of the output images of the approximate multipliers in comparison with the output images computed by the exact multiplier.

Table 7 gives the average PSNR and MSSIM values for the image multiplication and image sharpening applications.

In the image multiplication, Sky*Boat, Peppers*Peppers gray, Moon*Walking bridge, and Mandrill*Mandrill gray samples are considered. In the image sharpening application, we use Sky, Pepper, Walking bridge, and Mandrill as image samples.

In the image multiplication application, M15 and M14 offer the highest PSNR and MSSIM, respectively, and the first proposed design takes the third place. The third proposed multiplier is the fourth-best design and provides 40 percent and 5 percent higher PSNR and MSSIM values than the average of the other previous designs (except the first proposed design, M14, and M15). Moreover, among the three least accurate designs (PM2, M1, and M2), our design has slightly higher PSNR and MSSIM. It is worth pointing out that it also achieves considerable improvements in hardware parameters compared to its contenders, as shown in Table 5.

In the image sharpening application, according to the results, the first proposed design leads to higher PSNR and MSSIM than the previous approximate designs. The other results almost follow the same trend as observed in the multiplication application, except for the M1 and M2 multipliers, which show very low PSNR and MSSIM values for the sharpening application. It is worth mentioning that the first proposed design improves PSNR and MSSIM, on average, by 80 percent and 9 percent, respectively, compared to the previous designs listed in Table 7.

Designing an efficient approximate multiplier depends on the design constraints, including power consumption, speed, and accuracy, imposed by the intended application. Accordingly, it is not fair to compare different multipliers, offering different characteristics just based on a specific aspect. Various figures of merit (*FOM*s), considering both hardware and quality parameters, have been defined in the literature [2], [14], [16], [17], and [20]. Each of these *FOM*s puts a greater emphasis on some parameters.

A fair *FOM* was suggested in [20]. In this *FOM*, given in Eq. (12), the power and delay savings were considered to evaluate the hardware efficiency of approximate multipliers. Besides, the quality was measured by the average of ($PSNR^2$) for the processed images due to its importance to the end-user [20]. It is worth mentioning that in this *FOM*, a higher value shows a better compromise between hardware efficiency and accuracy.

It is notable that the *MSSIM* factor, as a critical quality metric, was not considered in *FOM*1. Accordingly, we define another figure of merit expressed in Eq. (13), in which the hardware efficiency is measured by *PDP*, which incorporates the impacts of both delay and the power consumption parameters. Moreover, the quality is measured by the average of ($PSNR \times MSSIM$) for the processed images. In this *FOM*, a lower value indicates a better trade-off between hardware efficiency and accuracy.

$$FOM1 = PowerSaving \times DelaySaving \times PSNR^2 \tag{12}$$

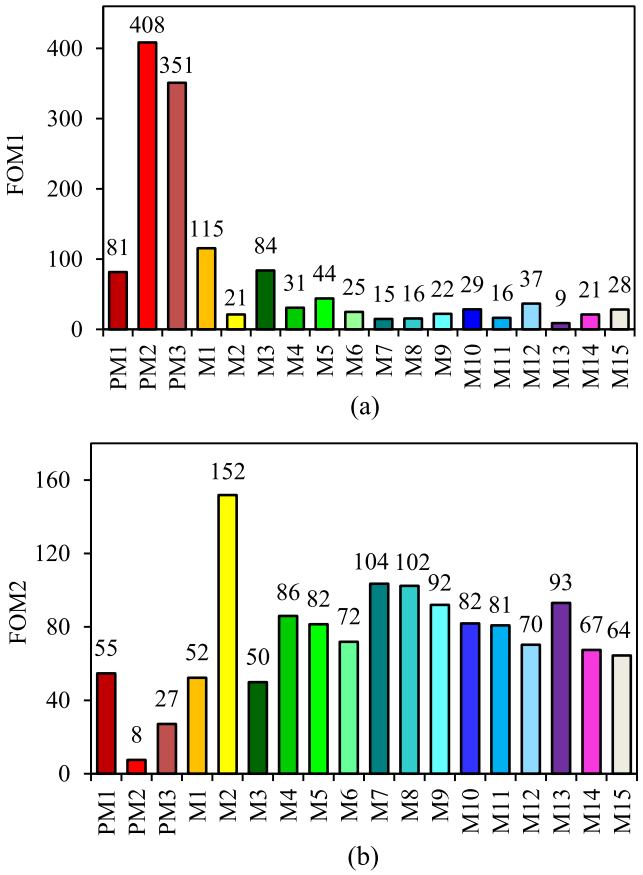


FIGURE 8. FOMs for the approximate multipliers (a) FOM1 (a higher value is desired) (b) FOM2 (a lower value is desired)

$$FOM2 = PDP/(MSSIM \times PSNR) \tag{13}$$

To have fairer comparisons between different approximate designs, we consider both of these *FOM*s. Figure 8 shows the comparisons of the *FOM*s of the approximate multipliers. According to the results, the second and the third proposed multipliers, using the approximate 5:2 and 4:1 compressors, offer better *FOM1*s than the other designs. The *FOM1* of the second proposed design is 15 times the average *FOM1* of the other approximate multipliers. In addition, the first proposed design takes the fifth-place in this factor. In *FOM2*, the second proposed multiplier, using the proposed approximate 4:1 compressor, offers the best *FOM2*, followed by the third proposed design and M3. The first proposed multiplier is also the fifth-best design. Moreover, the second proposed multiplier improves *FOM2*, on average, by 90 percent as compared to the other designs.

V. CONCLUSION

In this study, a new approximate 5:2 compressor and a new 4:1 compressor have been proposed. Moreover, three approximate multipliers have been proposed by utilizing NAND gates instead of AND gates and using different combinations of the proposed compressors. All of the proposed approximate multipliers have been simulated and compared with the state-of-the-

art designs based on the 7nm FinFET technology. According to the results, the second and the third proposed multipliers drastically improve the design parameters, including power, delay, and transistor count, as compared to the other designs. Neural network and image processing applications have been considered to assess the accuracy and quality of the approximate multipliers using MATLAB simulations. Furthermore, to have a fair judgment on the approximate multipliers, two Pareto diagrams, and two FOMs have been considered. According to the results, despite the diverse characteristics of the three proposed multipliers, they have shown better trade-offs between accuracy and performance than the previous designs.

REFERENCES

[1] W. Liu, T. Zhang, E. Clarion, M. O'Neill, P. Montuschi, and F. Lombardi, "Design and analysis of majority logic based approximate adders and multipliers," *IEEE Trans. Emerg. Top. Comput.*, early access, 2019, doi: 10.1109/TETC.2019.2929100.

[2] F. Sabetzadeh, M. H. Moaiyeri, and M. Ahmadinejad, "A majority-based imprecise multiplier for ultra-efficient approximate image multiplication," *IEEE Trans. Circuits Syst. I Reg. Papers*, vol. 66, no. 11, pp. 4200–4208, Nov. 2019.

[3] H. Jiang, F. J. H. Santiago, H. Mo, L. Liu, and J. Han, "Approximate arithmetic circuits: A survey characterization and recent applications," *Proc. IEEE*, vol. 108, no. 12, pp. 2108–2135, Dec. 2020.

[4] R. Venkatesan, A. Agarwal, K. Roy, and A. Raghunathan, "MACACO: Modeling and analysis of circuits for approximate computing," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, 2011, pp. 667–673.

[5] S. Mittal, "A survey of techniques for approximate computing," *ACM Comput. Surv.*, vol. 48, no. 4, pp. 1–33, 2016.

[6] M. Rezaei Khezeli, M. H. Moaiyeri, and A. Jalali, "Comparative analysis of simultaneous switching noise effects in MWCNT bundle and Cu power interconnects in CNTFET-based ternary circuits," *IEEE Trans . Very Large Scale Integr . Syst .*, vol. 27, no. 1, pp. 37–46, Jan. 2019.

[7] C. Xu, Y. Zheng, D. Niu, X. Zhu, S. H. Kang, and Y. Xie, "Impact of write pulse and process variation on 22 nm FinFET-based STT-RAM design: A device-architecture co-optimization approach," *IEEE Trans . Multi-Scale Comput. Syst .*, vol. 1, no. 4, pp. 195–206, Fourth Quarter 2015.

[8] S. K. Gupta and K. Roy, "Low power robust FinFET-based SRAM design in scaled technologies," in *Circuit Design for Reliability*. New York, NY, USA: Springer, 2015, pp. 223–253.

[9] B. Swahn and S. Hassoun, "Gate sizing: FinFETs vs 32nm bulk MOSFETs," in *Proc. 43rd Annu. Des. Automat. Conf.*, 2006, pp. 528–531.

[10] M. H. Moaiyeri, F. Sabetzadeh, and S. Angizi, "An efficient majority-based compressor for approximate computing in the nano era," *Microsyst. Technol.*, vol. 24, no. 3, pp. 1589–1601, Mar. 2018.

[11] H. Chang, J. Gu, M. Zhang, "Ultra low-voltage low-power CMOS 4-2 and 5-2 compressors for fast arithmetic circuits," *IEEE Trans . Circuits Syst . I Reg . Papers*, vol. 51, no. 10, pp. 1985–1997, Oct. 2004.

[12] Z. Abid, H. El-Razouk, and D.A. El-Dib, "Low power multipliers based on new hybrid full adders," *Microelectron. J.,* vol. 39, no. 12, pp. 1509–1515, Dec. 2008.

[13] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, "Design and analysis of approximate compressors for multiplication," *IEEE Trans. Comput.*, vol. 64, no. 4, pp. 984–994, Apr. 2015.

[14] M. Ahmadinejad, M. H. Moaiyeri, and F. Sabetzadeh, "Energy and area efficient imprecise compressors for approximate multiplication at nanoscale," *Int. J. Electron. Commun.*, vol. 110, pp. 4200–4208, Oct. 2019.

[15] A. Gorantla and P. Deepa, "Design of approximate compressors for multiplication," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 13, no. 3, 2017, Art. no. 44.

[16] O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, "Dual-quality 4:2 compressors for utilizing in dynamic accuracy configurable multipliers," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 25, no. 4, pp. 1352–1361, Apr. 2017.

[17] M. S. Ansari, H. Jiang, B. F. Cockburn, and J. Han, "Low-power approximate multipliers using encoded partial products and approximate compressors," *IEEE J. Emerg. Sel. Top. Circuits Syst.*, vol. 8, no. 3, pp. 404–416, Sep. 2018.

[18] S. Venkatachalam and S.-B. Ko, "Design of power and area efficient approximate multipliers," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 25, no. 5, pp. 1782–1786, May 2017.

[19] M. Ha and S. Lee, "Multipliers with approximate 4–2 compressors and error recovery modules," *IEEE Embedded Syst. Lett.*, vol. 10, no. 1, pp. 6–9, Mar. 2018.

[20] Z. Yang, J. Han, and F. Lombardi, "Approximate compressors for error resilient multiplier design," in *Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Nanotechnol. Syst.*, 2015, pp. 183–186.

[21] D. Esposito, A. G. M. Strollo, E. Napoli, D. De Caro, and N. Petra, "Approximate multipliers based on new approximate compressors," *IEEE Trans. Circuits Syst. I Reg. Papers*, vol. 65, no. 12, pp. 4169–4182, Dec. 2018.

[22] K. M. Reddy, M. H. Vasantha, Y. B. Nithin Kumar, and D. Dwivedi, "Design and analysis of multiplier using approximate 4-2 compressor," *Int. J. Electron. Commun.*, vol. 107, pp. 89–97, May 2019.

[23] Y. Guo, H. Sun, and Sh. Kimura, "Design of power and area efficient lower-part-OR approximate multiplier," in *Proc . IEEE Region 10 Conf .*, 2018, pp. 2110–2115.

[24] A. Arasteh, M. H. Moaiyeri, M. Taheri, K. Navi, and N. Bagherzadeh, "An energy and area efficient 4:2 compressor based on FinFETs," *Integration*, vol. 60, pp. 224–231, Jan. 2018.

[25] R. Marimuthu, Y. E. Rezinold, and P. Mallick, "Design and analysis of multiplier using approximate 15-4 compressor," *IEEE Access,* vol. 5, pp. 1027–1036, 2017.

[26] D. Baran, M. Aktan, and V. G. Oklobdzija, "Energy efficient implementation of parallel CMOS multipliers with improved compressors," in *Proc. ACM/IEEE Int. Symp. Low-Power Electron. Des. Conf.*, 2010, pp. 147–152.

[27] C. H. Lin and I. C. Lin, "High accuracy approximate multiplier with error correction," in *Proc. 31st IEEE Int. Conf. Comput. Des.*, 2013, pp. 33–38.

[28] J. Liang, J. Han, and F. Lombardi, "New metrics for the reliability of approximate and probabilistic adders," *IEEE Trans. Comput.*, vol. 62, no. 9, pp. 1760–1771, Sep. 2013.

[29] V. Leon, G. Zervakis, D. Soudris, and K. Pekmestzi, "Approximate hybrid high radix encoding for energy-efficient inexact multipliers," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 26, no. 3, pp. 421–430, Mar. 2018.

[30] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[31] H. Jiang, J. Han, and F. Lombardi, "A comparative review and evaluation of approximate adders," in *Proc. 25th Ed. Great Lakes Symp.*, 2015, pp. 343–348.

[32] R. Zimmermann and W. Fichtner, "Low-power logic styles: CMOS versus pass-transistor logic," *IEEE J. Solid-State Circuits,* vol. 32, no. 7, pp. 1079–1090, Jul. 1997.

[33] J. Rabaey, *Low Power Design Essentials*. New York, NY, USA: Springer, 2009.

[34] H. Afzali-Kusha, M. Vaeztourshizi, M. Kamal, and M. Pedram, "Design exploration of energy-efficient accuracy-configurable dadda multipliers with improved lifetime based on voltage overscaling," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 28, no. 5, pp. 1207–1220, May 2020.

[35] A. G. M. Strollo, E. Napoli, D. De Caro, N. Petra, and G. Di Meo, "Comparison and extension of approximate 4-2 compressors for low-power approximate multipliers," *IEEE Trans. Circuits Syst. I Reg. Papers*, early access, 2020, doi: 10.1109/TCSI.2020.2988353.

[36] L. T. Clark *et al.*, "ASAP7: A 7-nm finFET predictive process design kit," *Microelectron. J.*, vol. 53, pp. 105–115, Jul. 2016.

[37] S. Salahuddin, H. Jiao, and V. Kursun, "A novel 6T SRAM cell with asymmetrically gate underlap engineered FinFETs for enhanced read data stability and write ability," in *Proc. Int. Symp. Qual. Electron. Des.*, 2013, pp. 353–358.

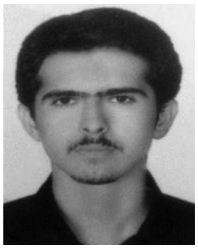
[38] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.

[39] M. S. Ansari, V. Mrazek, B. F. Cockburn, L. Sekanina, Z. Vasicek, and J. Han, "Improving the accuracy and hardware efficiency of neural networks using approximate multipliers," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 28, no. 2, pp. 317–328, Feb. 2020.

[40] Y. LeCun, C. Cortes, and C. Burges. "MNIST handwritten digit database," AT&T Labs, 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist

[41] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *Proc. Neural Inf. Process. Syst. Workshop Deep Learn. Unsupervised Feature Learn.*, 2011, Art. no. 5.

[42] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[43] V. Mrazek, S. S. Sarwar, L. Sekanina, Z. Vasicek, and K. Roy, "Design of power-efficient approximate multipliers for approximate artificial neural networks," in *Proc. 35th IEEE/ACM Int. Conf. Comput.-Aided Design*, 2016, pp. 1–7.



MOHAMMAD AHMADINEJAD received the BSc and MSc degrees in electrical engineering from Shahid Beheshti University, Tehran, Iran. His research interests include VLSI circuit design, nanoelectronics, approximate computing, and neural networks.

MOHAMMAD HOSSEIN MOAIYERI (Senior Member, IEEE) received the PhD degree in computer architecture from Shahid Beheshti University, Tehran, Iran, in 2012. He is currently an assistant professor with the Faculty of Electrical Engineering, Shahid Beheshti University. His research interests include VLSI design for beyond-CMOS nanodevices, low-power VLSI design, and mixed-signal integrated circuits design.