

Compressor based approximate multiplier architectures for media processing applications

Uppugunduru Anil Kumar, Syed Ershad Ahmed

Department of Electrical and Electronics Engineering, BITS Pilani, Hyderabad Campus, Hyderabad, India

Article Info

Article history:

Received Jul 14, 2020

Revised Dec 19, 2020

Accepted Jan 12, 2021

Keywords:

Approximate computing multiplier

Compressor

Partial product generation

Partial product reduction

ABSTRACT

Approximate computing is an attractive technique to gain substantial improvement in the area, speed, and power in applications where exact computation is not required. This paper proposes two improved multiplier designs based on a new 4:2 approximate compressor circuit to simplify the hardware at the partial product reduction stage. The proposed multiplier designs are targeted towards error-tolerant applications. Exhaustive error and hardware analysis has been carried out on the existing and proposed multiplier designs. The results prove that the proposed approximate multiplier architecture performs better than the existing architectures without significant compromise on quality metrics. Experimental results show that die-area and power consumed are reduced up to 28%, and 25.29% respectively in comparison with the existing designs without significant compromise on accuracy.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Uppugunduru Anil Kumar

Department of Electrical and Electronics Engineering

BITS Pilani, Hyderabad Campus

Jawahar Nagar, Hyderabad, Telangana 500078, India

Email: anilkumaruppugundur@gmail.com

1. INTRODUCTION

Applications involving image and video processing have inherent error resilience capability and hence can tolerate computation errors [1-3]. Since the final output is interpreted by human sensory systems, particularly in image and video processing, inaccuracy in the final result can be tolerated to a certain limit [4]. Most of these applications require a power-hungry multiplier [5, 6] to carry out computations, approximating this multiplier results in an improvement in the area, delay, and power without significant compromise on accuracy [7].

Multiplication is a ubiquitous arithmetic operation, and hence improving the energy consumption will lead to substantial improvements at the system level [8-11]. The criteria used to measure their performance in digital systems are die-area, speed of operation, and power consumed. Thus, various approximate multiplier design techniques have been proposed at the logic level, which tries to leverage error resilience and achieve an improvement in the area, latency, and power [12-14].

Various types of approximate multiplier architectures have been reported in the literature to reduce the computational complexity. Work in [15] proposed an under-designed multiplier by altering the Karnaugh map of the 2*2 sub multiplier at the partial product generation stage. Mahdiani *et al.* [16] proposed a broken-array multiplier (BAM) by pruning either vertical or horizontal partial product columns. An approximate recursive Wallace tree multiplier with simple carry-in prediction was proposed in [17].

Momeni *et al.* [18] reported a multiplier with approximation at the compressor level to achieve lower area and power; however, it suffered from low accuracy. Yang *et al.* [19] reduced the error in the

multiplier using an improved compressor design but results in higher power consumption. Minho *et al.* [20] modified the compressor designed by Yang and incorporated a simple error correction circuit to improve accuracy further. Suganti *et al.* [21] simplified the partial product reduction stage using an approximate half adder, full adder, and compressor circuit. Xilin *et al.* [22] simplified the PPR by deploying approximate compressor circuits. However, the designs [20-22] tends to occupy more area and hence consumes more power. This paper proposes two approximate multiplier designs using an efficient compressor circuit with the intent to minimize computation complexity without significant compromise on accuracy. The objectives of this work are summarized as:

- An improved approximate compressor is proposed in this work that reduces the die-area and power in the multiplier designs.
- Two multiplier architectures based on the proposed approximate compressor are implemented.

The remainder of the paper is arranged as: A description of the proposed approximate multiplier designs is presented in section 2. Section 3 provides the exhaustive error analysis and hardware synthesis of the proposed and existing designs. The impact of proposed and existing multiplier architectures on image processing application is investigated in section 4, while conclusions are provided in section 5.

2. PROPOSED WORK

The implementation of any multiplier involves three steps [23]. Partial products are generated in the first stage while their reduction to two rows is accomplished in the second step. These two rows are compressed to the final product in the final step. Among these, the partial product reduction (PPR) step consumes more power compared to the other steps as it is a computational intensive stage. The compressor forms the essential module in the partial product reduction structure, and hence optimizing it results in a simpler reduction structure. A new approximate 4:2 compressor is presented in this work that, when deployed in multiplier architectures, result in improved area and performance.

2.1. Proposed 4:2 approximate compressor

Figure 1(a) shows an exact 4:2 compressor [24] comprises of five inputs and three outputs. Intuitively, exact 4:2 compressors are obtained by cascading two 3:2 full adders, as shown in Figure 1(b). However, the exact compressor tends to consume more area and power. Since C_{in} and C_1 contribute little to the final result, the associated circuit can be removed. In this work, this idea is used to design a 4:2 approximate compressor. Since C_{in} and C_1 are ignored, the remaining four inputs B_1 , B_2 , B_3 , and B_4 , result in two outputs Sum and C_0 .

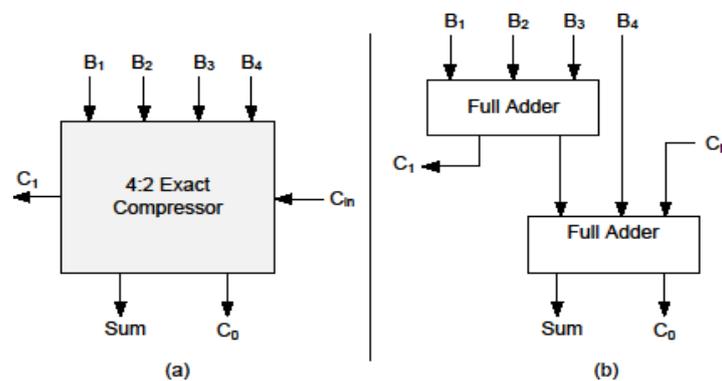


Figure 1. Proposed 4:2 compressor: (a) 4:2 exact compressor, (b) 4:2 compressor using full adder

The efficacy of the proposed design can be found by evaluating the compressor output for all possible input combinations. Accordingly, Table 1 presents the statistics about the correctness of the Sum and C_0 for different inputs. It is evident from Table 1 that the proposed design produces correct output in most of the cases, and the same is indicated in the last column in Table 1 with a difference as zero. However, the proposed compressor generates an incorrect result for the "1111" input combination. The same is highlighted in grey color in Table 1, corresponding to the "1111" input combination. Therefore, the error rate of the proposed compressor out of 256 possibilities is $1/256$, i.e., 0.0039.

Table 1. Truth table of proposed 4:2 approximate compressor for all possible input combinations

Compressor Input combinations				Value	Probability	Compressor Output		Approx. Value	Difference
B ₁	B ₂	B ₃	B ₄			Sum	C ₀		
0	0	0	0	0	81/256	0	0	0	0
0	0	0	1	1	27/256	1	0	1	0
0	0	1	0	1	27/256	1	0	1	0
0	0	1	1	2	9/256	0	1	2	0
0	1	0	0	1	27/256	1	0	1	0
0	1	0	1	2	9/256	0	1	2	0
0	1	1	0	2	9/256	0	1	2	0
0	1	1	1	3	3/256	1	1	3	0
1	0	0	0	1	27/256	1	0	1	0
1	0	0	1	2	9/256	0	1	2	0
1	0	1	0	2	9/256	0	1	2	0
1	0	1	1	3	3/256	1	1	3	0
1	1	0	0	2	9/256	0	1	2	0
1	1	0	1	3	3/256	1	1	3	0
1	1	1	0	3	3/256	1	1	3	0
1	1	1	1	4	1/256	0	1	2	-2

Error probability P(e) of the proposed approximate 4:2 compressor is calculated mathematically as follows. Let p₁, p₂, p₃, and p₄ be the probabilities of getting '1' to the input bits of the 4:2 compressor B₁, B₂, B₃, and B₄. Then the probability of occurrence error in the proposed approximate 4:2 compressor is written as (1).

$$P(e) = p_1 p_2 p_3 p_4 \tag{1}$$

The partial products generated using AND gates form the input to B₁, B₂, B₃, and B₄. The probability that an individual bit of multiplier or multiplicand is '1' is 0.5. Therefore, the probability of generating '1' as an input bit to the compressor is 0.25. Then the probability of getting error for proposed approximate 4:2 compressor is P(e)=0.25*0.25*0.25*0.25, i.e., 0.0039. The logic circuit of the proposed 4:2 compressor derived using truth table is given in Figure 2 and is described by the Boolean (2) and (3).

$$\text{Sum} = k_1 \oplus k_2 \tag{2}$$

$$C_0 = k_1 \cdot k_2 + B_1 \cdot B_2 + B_3 \cdot B_4 \tag{3}$$

where

$$k_1 = B_1 \oplus B_2$$

$$\text{and } k_2 = B_3 \oplus B_4$$

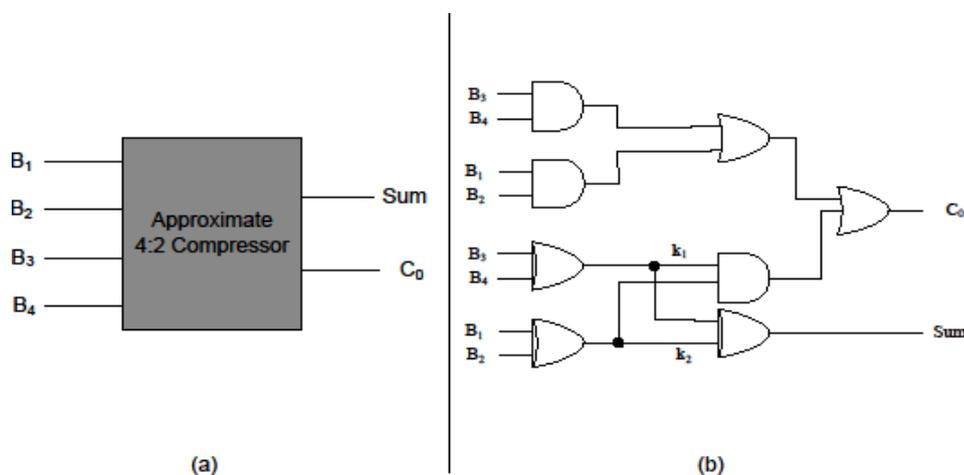


Figure 2. Logic circuit of the proposed 4:2 compressor: (a) Block diagram of proposed 4:2 approximate compressor, (b) Logic circuit of proposed 4:2 compressor

2.2. Proposed approximate multiplier architecture (D1)

The architecture of 8*8 proposed approximate multiplier design, namely D1, is depicted in Figure 3. The generated partial products are shown using solid dots. The PPR tree structure requires two levels (Level 1 and 2). It is divided into two regions: one four-bit truncation region, and the other 11-bit approximate region. The approximate region consists of proposed 4:2 approximate compressors, full and half adders. The partial product tree is reduced to two rows using the compressor logic, and these rows are accumulated to the final product using a ripple carry adder (RCA).

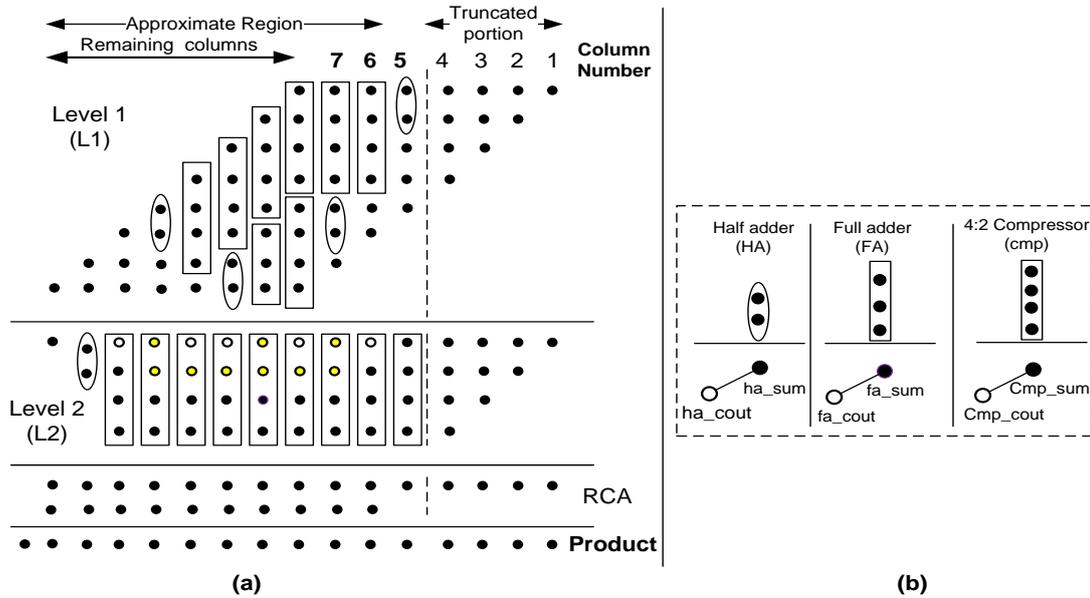


Figure 3. Proposed PPR structure with approximate compressors

Another variant of the multiplier, namely D2, is achieved by reducing the number of approximate columns in the reduction tree structure. In the first multiplier design discussed above, approximation and truncation are applied to all the partial product columns in the reduction tree, while the second multiplier design (D2) is obtained by applying the truncation in four least significant columns and approximation in N-4 columns for an N-bit multiplier. The rest of the partial products in the most significant columns are compressed by exact logic.

3. EXPERIMENTAL RESULTS

3.1. Error analysis

Detailed error analysis on various multiplier architectures, including the proposed designs, was carried out using MATLAB for all the input combinations (65535 cases), and computed results are tabulated in Table 2. Quality metrics such as mean relative error distance (MRED) and normalized mean error distance (NMED) are used to compute the accuracy of existing and proposed approximate designs. The error distance (ED) is computed as a difference of exact and approximate result and is represented in (4).

$$ED = |R' - R| \tag{4}$$

where R' is the approximate result, and R is the accurate result [25]. The average of all EDs is the mean error distance (MED). At the same time, MRED is the mean of all relative error distances (REDs) where RED is calculated by using (5).

$$RED = ED/R \tag{5}$$

From Table 2, it is evident from the proposed design D2 is more accurate since it has better NMED and MRED than existing designs. D2 achieves better accuracy since approximation and truncation are made

to least significant 8 columns in the partial product reduction tree. From Table 2, it is evident that D2 achieves an improvement upto 93.5% in MRED compared to the existing designs. Similarly, NMED of D2 is upto 98.9% better than the existing designs.

The design D1 has more NMED compared to existing designs except for Multiplier1 [21]. Also, D1 has less MRED compared to the existing designs except for Minhó [20] and Yang [19]. Though designs [18-22] have less NMED and MRED in comparison with D1, they tend to occupy more area and consume more power.

Table 2. Error metrics of various approximate multiplier designs

Multiplier Design	NMED (10^{-3})	MRED (%)
D1	3.4	1.11
D2	0.27	0.6
Multiplier1 [21]	25.98	7.9
Multiplier2 [21]	1.56	1.36
Xilin [22]	1.33	1.40
Momeni [18]	1.64	9.29
Yang [19]	0.49	0.7
Minho [20]	0.43	0.7

3.2. Synthesis results

For the sake of fair analysis, all the 8-bit multiplier [18-22] schemes, including the proposed ones, have been modeled structurally using verilog hardware description language. Simulation has been performed using the Cadence incisive unified simulator v6.1. Hardware synthesis of approximate multiplier designs has been carried out using Cadence RTL compiler v7.1 using TSMC 180 nm process node (slow-normal library) to compute area, delay, and power.

It can be observed from Table 3 that the proposed designs D1 and D2 are faster than the existing designs except for Minhó. D1 consumes less area in comparison to all the existing designs due to approximation being introduced in all the columns of the partial product reduction stage. Further, D1 achieves low power consumption compared to existing designs except for multiplier1. The percentage improvement in area and power in D1 is up to 28% and 25.29% respectively, compared to existing designs. Similarly, D2 takes less area than existing designs except for Minhó and Momeni. Also, D2 has low power consumption in comparison with existing designs except for multiplier1 [21], Xilin [22], and Momeni [18]. Though a few existing designs [18, 21, 22] perform marginally better in area and power compared to D2, they suffer from lower peak signal to noise ratio, as shown in Table 4. The percentage improvement in area and power of D2 is upto 17.21% and 14.15%, respectively, compared to existing designs.

Table 3. Synthesis of various approximate multipliers

Design	Area (μm^2)	Power (nW)	Delay (ps)
D1	818.5	26132.7	2559
D2	939.8	30029.7	2559
Xilin [22]	1017.5	29654.7	2927
Multiplier1 [21]	944.8	24625.8	2657
Multiplier2 [21]	1135.3	34981.6	2719
Yang [19]	951.1	31093	2637
Minho [20]	925	30171	2557
Momeni [18]	928.57	26594.8	2927

Table 4. PSNR and SSIM of various approximate multipliers

Parameter	Multiplier1 [21]	Multiplier2 [21]	Yang [19]	Minho [20]	Momeni [18]	Xilin [22]	D1	D2
PSNR	33.00	54.70	61.12	63.91	49.84	56.39	40.41	64.44
SSIM	0.9622	0.9995	0.9998	0.9999	0.9224	0.9995	0.9875	0.9999

Figure 4 presents the power-delay product (PDP) of various multipliers, including D1 and D2. It can be observed that D1 and D2 have less PDP compared to all existing multipliers except for multiplier 1. The percentage improvement of D1 in PDP is upto 29.69% compared to existing designs. Similarly, D2 has upto 19.2% improvement in PDP in comparison with the existing designs.

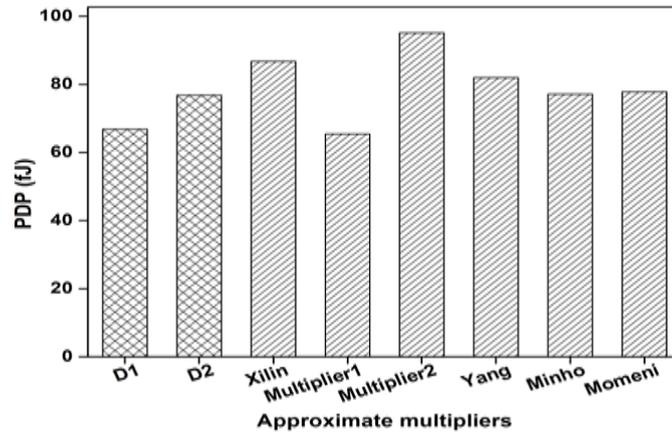


Figure 4. Comparison of PDP of various multiplier architectures

4. PERFORMANCE OF APPROXIMATE MULTIPLIERS IN IMAGE PROCESSING APPLICATIONS

As discussed in the preceding subsection, the proposed designs offer significant benefits in terms of accuracy, area, and power. In this section, proposed designs D1 and D2 are evaluated on real-time image sharpening [26] and image multiplication [27] applications.

4.1. Image multiplication

In this section, proposed designs D1 and D2 are evaluated using an image multiplication. Image multiplication is the best candidate to validate the multipliers as it uses direct multiplication. Image multiplication accepts two images and generates an output image. For example, let P1 and P2 be the input images, then output image X is expressed as (6).

$$X(i,j)=P1(i,j) * P2(i,j) \quad (6)$$

To quantify our results and measure the approximate multiplier's performance, structural similarity index (SSIM) and peak signal to noise ratio (PSNR) is used. Table 4 presents the PSNR and SSIM values of different approximate multipliers. From Table 4, it is evident that D2 achieves better PSNR and SSIM compared to existing multipliers due to its low NMED and MRED. D1 achieves better PSNR compared to Multiplier1. Existing designs [18-22] have better PSNR and SSIM than D1, however, they suffer from more area, power, and delay. Figure 5 shows the multiplication of images processed using exact and proposed multipliers. It is evident that the obtained images using an exact multiplier, D1 and D2 look almost identical.

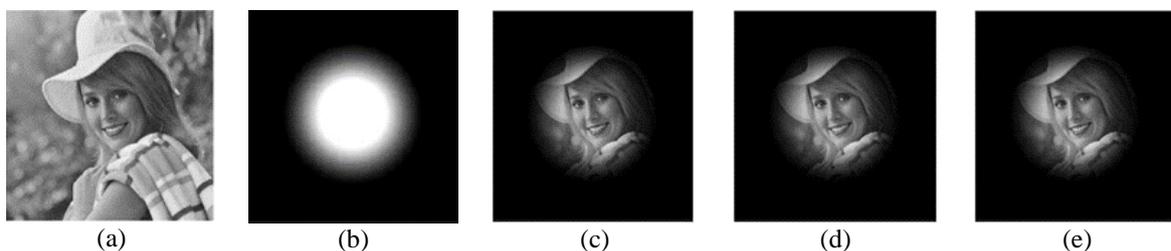


Figure 5. Images obtained by using exact and proposed multipliers, (a) Image1 [27], (b) Image2 [27], (c) Exact output, (d) D1 output, (e) D2 output

4.2. Image sharpening

Image sharpening improves the visual quality of an image. The sharpening algorithm accepts an image (P), processes it using 5*5 kernels to produce a high-quality image [26]. Let P be the input image, and then the output image Z can be expressed as (7).

$$Z(x,y) = P(x,y) - R \quad (7)$$

Where:

$$R = \frac{1}{273} \sum_{l=-2}^{l=2} \sum_{n=-2}^{n=2} H(l+3, n+3)P(x-l, y-n)$$

and H is a matrix defined as;

$$H = \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

4.3. Comparison with existing approximate multipliers for hardware and accuracy

Figure 6 shows the comparison results for 8*8 approximate multipliers obtained by considering the MRED, NMED, SSIM, and PSNR w.r.t power-delay-product (PDP). From Figure 7(a) and 7(b) (see in appendix), it is evident that D2 has lower PDP with better MRED and NMED compared to existing designs. Similarly, From Figure 7(c) and 7(d) (see in appendix), D2 has lower PDP and better SSIM and PSNR compared to existing designs. Similarly, D1 has a lower PDP than the existing multipliers except for multiplier1. Finally, it can be concluded that D2 obtains better PSNR, SSIM, MRED, and NMED that, too, with less PDP compared to existing designs except for multiplier1. D1 has a lower PDP compared to the existing designs except for multiplier1 with good SSIM and PSNR.

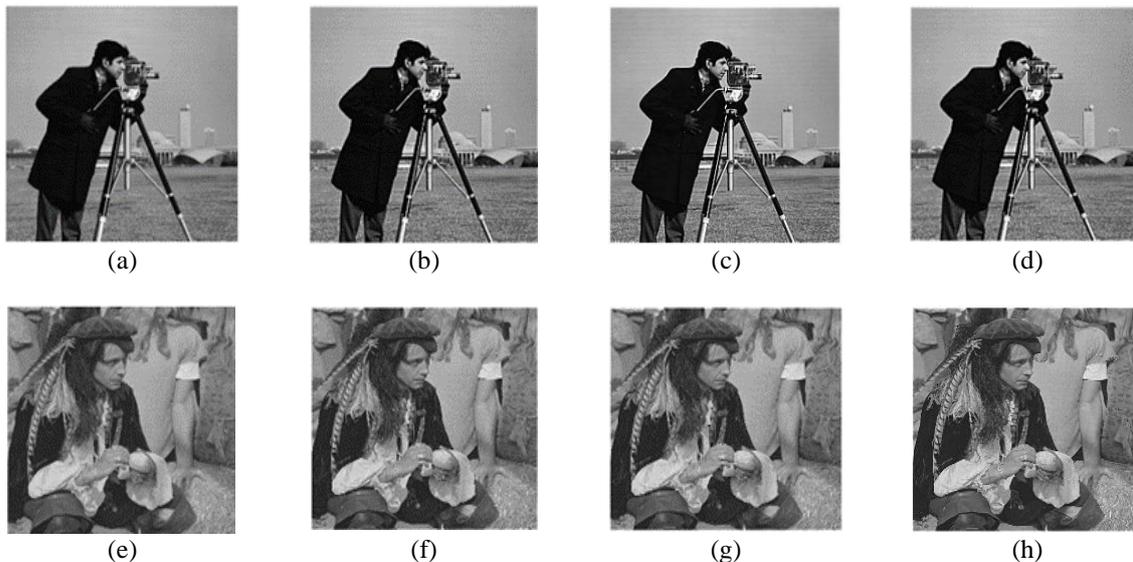


Figure 6. Images obtained by using exact and proposed multipliers after image sharpening, (a) Input image, (b) Exact output, (c) D1 output, (d) D2 output, (e) Input image, (f) Exact output, (g) D1 output, (h) D2 output

5. CONCLUSION

Approximate computing aims to leverage the error-tolerant quality in image and video processing applications. In this paper, two multiplier design techniques are presented that reduce computational complexity. The approximate multiplier designs comprise of approximate compressor that simplifies the hardware at the PPR stage, thereby improving the delay and power consumption compared to existing works. Comprehensive error analysis and synthesis results prove the efficacy of the proposed designs in comparison with existing designs. Experimental results show that die-area and power consumed are reduced up to 28% and 25.29% respectively in comparison with the existing designs. Towards the end, the impact of proposed methods on an image processing application is investigated, and the results prove that the proposed multiplier designs achieve a better computation quality effort trade-off when compared to existing designs.

APPENDIX

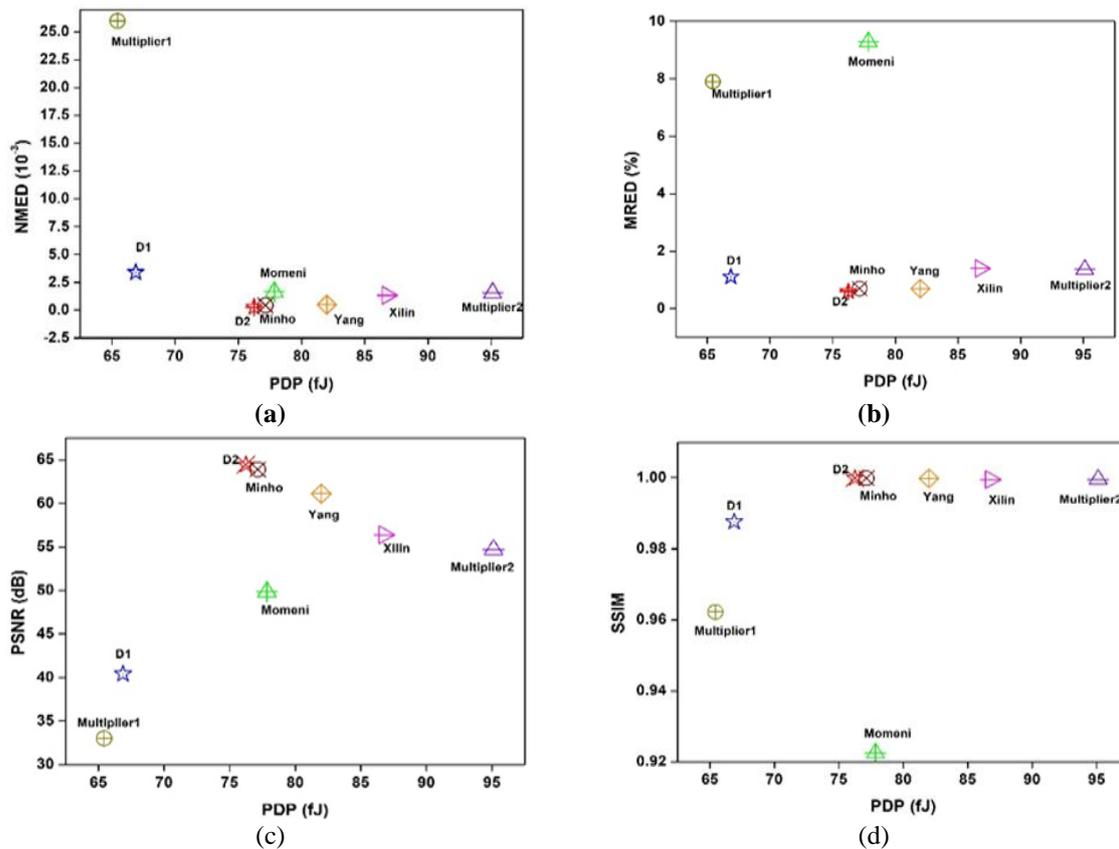


Figure 7. NMED, MRED, PSNR and SSIM comparison w.r.t PDP for various multipliers, (a) NMED vs PDP, (b) MRED vs PDP, (c) PSNR vs PDP, (d) SSIM vs PDP

REFERENCES

- [1] Chippa, V. K., Chakradhar, S. T., Roy, K., and Raghunathan, A., "Analysis and characterization of inherent application resilience for approximate computing," in *Proceedings of the 50th Annual Design Automation Conference*, 2013, pp. 1–9.
- [2] El-Harouni, W., Rehman, S., Prabakaran, B. S., Kumar, A., Hafiz, R., and Shafique, M., "Embracing approximate computing for energy-efficient motion estimation in high-efficiency video coding," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2017, pp. 1384–1389.
- [3] Venkatachalam, S., Adams, E., Lee, H. J., and Ko, S., "Design and analysis of area and power efficient approximate booth multipliers," *IEEE Transactions on Computers*, vol. 68, no. 11, pp. 1697–1703, 2019.
- [4] Mittal and Sparsh, "A survey of techniques for approximate computing," *ACM Computing Surveys (CSUR)*, vol. 48, no. 4, pp. 1–33, 2016.
- [5] Narayanamoorthy, S., Moghaddam, H. A., Liu, Z., Park, T., and Kim, N. S., "Energy-efficient approximate multiplication for digital signal processing and classification applications," *IEEE transactions on very large scale integration (VLSI) systems*, vol. 23, no. 6, pp. 1180–1184, 2014.
- [6] Jiang, H., Liu, C., Liu, L., Lombardi, F., and Han, J., "A review, classification, and comparative evaluation of approximate arithmetic circuits," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 13, no. 4, pp. 1–34, 2017.
- [7] Venkataramani, S., Chakradhar, S. T., Roy, K., and Raghunathan, A., "Computing approximately, and efficiently," in *2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2015, pp. 748–751.
- [8] Venkataramani, S., Chakradhar, S. T., Roy, K., and Raghunathan, A., "Approximate computing and the quest for computing efficiency," in *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2015, pp. 1–6.
- [9] Xu, Q., Mytkowicz, T., and Kim, N. S., "Approximate computing: A survey," *IEEE Design & Test*, vol. 33, no. 1, pp. 8–22, 2015.
- [10] Qiqieh, I., Shafik, R., Tarawneh, G., Sokolov, D., Das, S., and Yakovlev, A., "Significance-driven logic compression for energy-efficient multiplier design," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 8, no. 3, pp. 417–430, 2018.

- [11] Jiang, H., Liu, C., Maheshwari, N., Lombardi, F., and Han, J., "A comparative evaluation of approximate multipliers," in *2016 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)*, 2016, pp. 191–196.
- [12] Qiqieh, I., Shafik, R., Tarawneh, G., Sokolov, D., Das, S., and Yakovlev, A., "Energy-efficient approximate Wallace-tree multiplier using significance-driven logic compression," in *2017 IEEE International Workshop on Signal Processing Systems (SiPS)*, 2017, pp. 1–6.
- [13] Shafique, M., Hafiz, R., Rehman, S., El-Harouni, W., and Henkel, J., "Cross-layer approximate computing: From logic to architectures," in *Proceedings of the 53rd Annual Design Automation Conference*, 2016, pp. 1–6.
- [14] Stanley-Marbell, P., Alaghi, A., Carbin, M., Darulova, E., Dolecek, L., Gerstlauer, A *et al.*, "Exploiting errors for efficiency: A survey from circuits to algorithms," arXiv:1809.05859, 2018.
- [15] Kulkarni, P., Gupta, P., and Ercegovic, M., "Trading accuracy for power with an underdesigned multiplier architecture," in *2011 24th International Conference on VLSI Design*, 2011, pp. 346–351.
- [16] Mahdiani, H. R., Ahmadi, A., Fakhraie, S. M., and Lucas, C., "Bio-inspired imprecise computational blocks for efficient VLSI implementation of soft-computing applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 4, pp. 850–862, 2009.
- [17] Bhardwaj, K., Mane, P. S., and Henkel, J., "Power-and area-efficient approximate Wallace tree multiplier for error-resilient systems," in *Fifteenth International Symposium on Quality Electronic Design*, pp. 263–269, 2014.
- [18] Momeni, A., Han, J., Montuschi, P., and Lombardi, F., "Design and analysis of approximate compressors for multiplication," *IEEE Transactions on Computers*, vol. 64, no. 4, pp. 984–994, 2014.
- [19] Yang, Z., Han, J., and Lombardi, F., "Approximate compressors for error-resilient multiplier design," in *2015 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFTS)*, 2015, pp. 183–186.
- [20] Ha, M., and Lee, S., "Multipliers with approximate 4–2 compressors and error recovery modules," *IEEE Embedded Systems Letters*, vol. 10, no. 1, pp. 6–9, 2017.
- [21] Venkatachalam, S., and Seok-Bum Ko., "Design of power and area efficient approximate multipliers," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 5, pp. 1782–1786, 2017.
- [22] Yi, X., Pei, H., Zhang, Z., Zhou, H., and He, Y., "Design of an Energy-Efficient Approximate Compressor for Error-Resilient Multiplications," in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2019, pp. 1–5.
- [23] B. Parhami, "Computer arithmetic," *Oxford university press*, vol. 20, 2010.
- [24] Mokrian, P., Howard, G. M., Jullien, G., and Ahmadi, M., "On the use of 4: 2 compressors for partial product reduction," in *CCECE 2003-Canadian Conference on Electrical and Computer Engineering. Toward a Caring and Humane Technology (Cat. No. 03CH37436)*, vol. 1, pp. 121–124, 2003.
- [25] Liang, J., Han, J., and Lombardi, F., "New metrics for the reliability of approximate and probabilistic adders," *IEEE Transactions on Computers*, vol. 62, no. 9, pp. 1760–1771, 2012.
- [26] Lau, M. S., Ling, K. V., and Chu, Y. C., "Energy-aware probabilistic multiplier: design and analysis," in *Proceedings of the 2009 international conference on Compilers, architecture, and synthesis for embedded systems*, 2009, pp. 281–290.
- [27] Jiang, H., Liu, C., Lombardi, F., and Han, J., "Low-power approximate unsigned multipliers with configurable error recovery," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 1, pp. 189–202, 2018.

BIOGRAPHIES OF AUTHORS



U. Anil Kumar received an Engineering degree in electronics engineering from the JNTU University Kakinada in 2011. He is currently pursuing a Ph.D. degree with the Birla Institute of Technology and Science (BITS), Pilani, Hyderabad Campus, India. His research interests include power-efficient VLSI design, approximate computing, and video coding.



Syed Ershad Ahmed received his Ph.D. from the Birla Institute of Technology, India. He is currently an Assistant Professor at Birla Institute of Technology and Science (BITS), Pilani, Hyderabad Campus, India. His research interests include arithmetic circuits design, low power VLSI design, and approximate computing. He is an active member of IEEE and published 2 SCI journals and 15 conference articles.